Tableau Server on Linux Administrator Guide

Last Updated 11/11/2024

Contents

Tableau Server Release Notes

This topic describes what's new in the latest release. Use the viz below to explore new features in Tableau Server. Click on a feature to bring up the tooltip with a link to detailed documentation for that feature. Explore the filters to refine your search. Download the data to create a customized list.

- Use the Search by **Feature** dashboard to see a list of new features for a product or version, or explore when a feature was released. The dashboard currently defaults to Server for the latest version of Server.
- Use the **Upgrade Server** dashboard to see a list of features specific to your upgrade. The upgrade view includes filters that allow you to filter for new or changed features. Features that are listed under the **changed** status are typically features that can impact upgrade. The list features categorized as changed might be useful to prepare and test your upgrades.

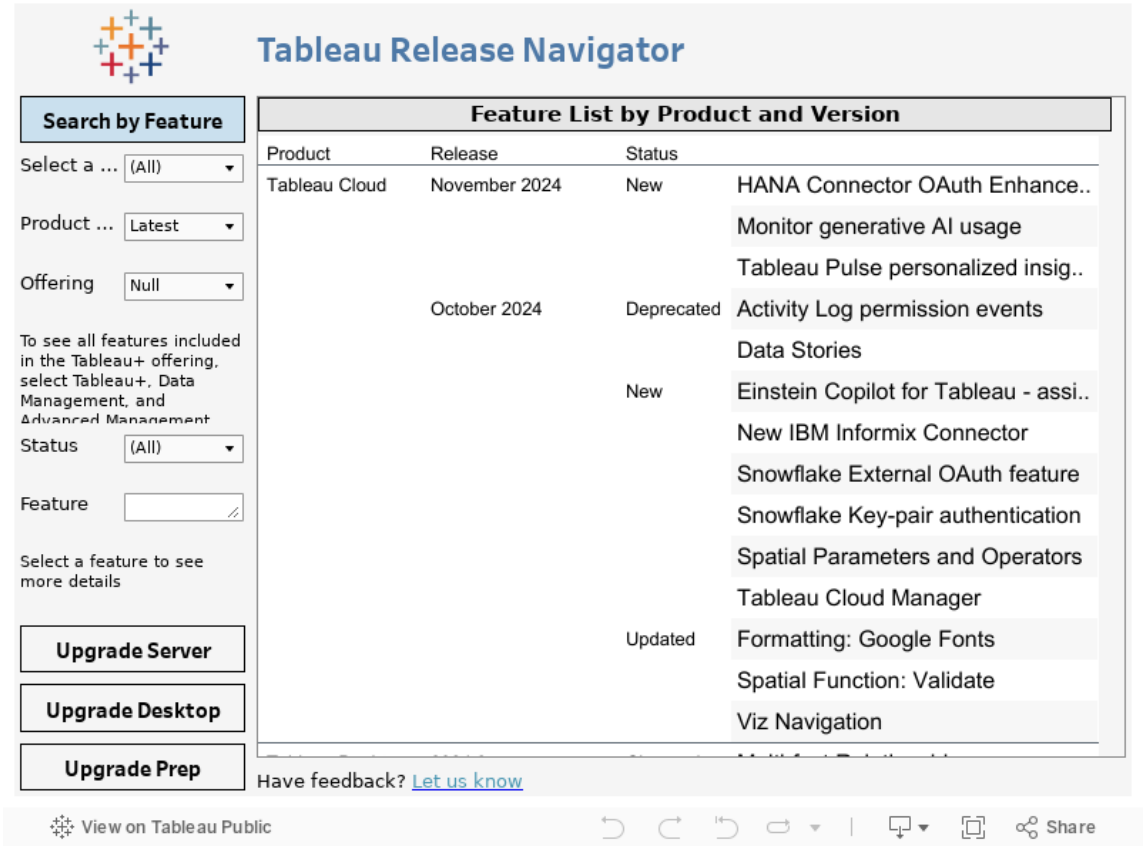


Tableau Release Navigator
Search by Feature
Select a ... (All)
Product ... Latest
Offering Null
To see all features included in the Tableau+ offering, select Tableau+, Data Management, and
Status (All)
Feature
Select a feature to see more details
Upgrade Server
Upgrade Desktop
Upgrade Prep
Feature List by Product and Version
Product Release Status
Tableau Cloud November 2024 New HANA Connector OAuth Enhance..
Monitor generative AI usage
Tableau Pulse personalized insig..
October 2024 Deprecated Activity Log permission events
Data Stories
New Einstein Copilot for Tableau - assi..
New IBM Informix Connector
Snowflake External OAuth feature
Snowflake Key-pair authentication
Spatial Parameters and Operators
Tableau Cloud Manager
Updated Formatting: Google Fonts
Spatial Function: Validate
Viz Navigation
Have feedback? Let us know
View on Tableau Public
Share

Plan Your Deployment

The articles in this section provide information on planning a Tableau Server deployment.

In addition to consulting the articles in this section, we recommend that you review Tableau Blueprint as a first step in your planning workflow. Tableau Blueprint is a step-by-step guide to becoming a data-driven organization.

Validating your server deployment plan

Before you commit to installing a new Tableau Server deployment in your organization, be sure to carefully evaluate your options. For most organizations, Tableau Cloud will provide a more reliable, performant, and cost-effective analytics solution when compared to self-hosting Tableau Server. For information about the viability of Tableau Cloud for your organization, review this blog post, Should I move my analytics to the cloud?

Already running Tableau Server and want to migrate to Tableau Cloud? See Tableau Cloud Manual Migration Guide.

If you have determined that you must self-host Tableau Server, we recommend following the prescribed Tableau deployment in the Enterprise Deployment Guide (EDG). The EDG presents a fully-tested and supported, performant, scalable, secure reference architecture based on a tiered data network. Moving forward, we are committed to investing in the EDG reference architecture to ease feature rollouts and improve upgrade scenarios.

Server Administrator Overview

Tableau Server on Linux integrates with a number of components in your IT infrastructure to provide a unique self-service data analytics culture for your users. It's important that you, as a server administrator, understand how Tableau Server fits into your IT infrastructure.

The topics in this section provide information on planning, deploying, tuning, and managing Tableau Server.

If you are deploying Tableau Server as part of a broader effort to transform your organization into a data-driven culture, see Tableau Blueprint. Tableau Blueprint is a step-by-step guide to becoming a data-driven organization, whether your organization is new to modern, self-service analytics or you've already deployed and need to broaden, deepen, and scale the use of data.

This topic provides a brief overview of how to think about Tableau Server and how it interacts with your existing IT infrastructure.

> Looking for Tableau Server on Windows? See Server Administrator Overview.

Validating your server deployment plan

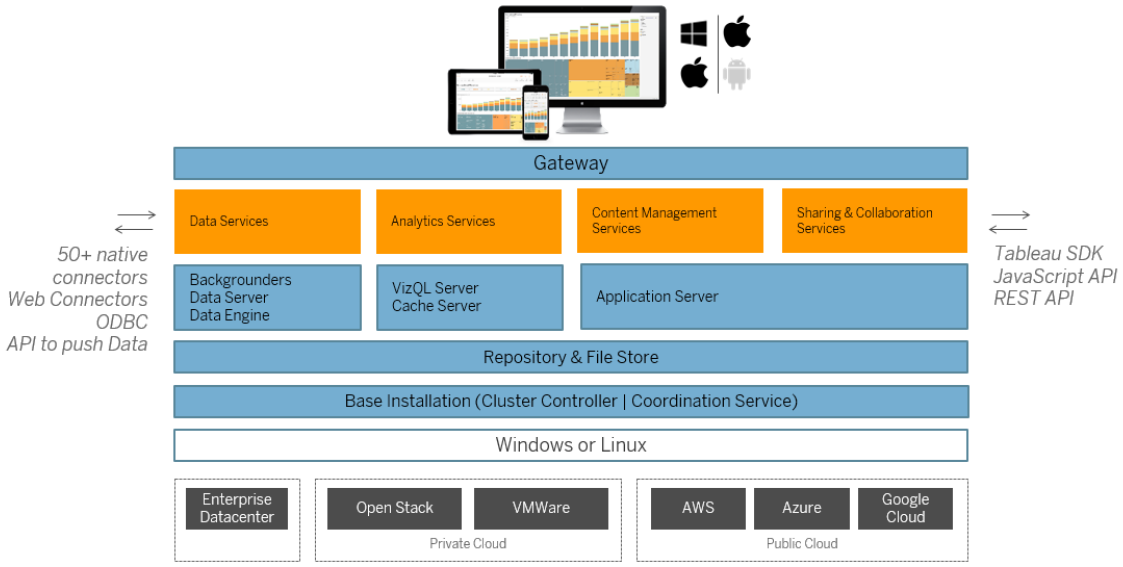
Before you commit to installing a new Tableau Server deployment in your organization, be sure to carefully evaluate your options. For most organizations, Tableau Cloud will provide a more reliable, performant, and cost-effective analytics solution when compared to self-hosting Tableau Server. For information about the viability of Tableau Cloud for your organization, review this blog post, Should I move my analytics to the cloud?

Already running Tableau Server and want to migrate to Tableau Cloud? See Tableau Cloud Manual Migration Guide.

If you have determined that you must self-host Tableau Server, we recommend following the prescribed Tableau deployment in the Enterprise Deployment Guide (EDG). The EDG presents a fully-tested and supported, performant, scalable, secure reference architecture based on a tiered data network. Moving forward, we are committed to investing in the EDG reference architecture to ease feature rollouts and improve upgrade scenarios.

Architectural overview

Tableau Server is a collection of processes that work together to provide a full self-service analytic platform for your users. The following diagram shows a high-level architectural view of Tableau Server.



Multiple server processes (shown in blue above) work together to provide services at various tiers. The Gateway process is the component that redirects traffic from all Tableau clients to the available server nodes in a cluster.

Data Services is a logical grouping of services that provide data freshness, shared meta data management, governed data sources, and in-memory data. The underlying processes that power Data Services are the Backgrounder, Data Server and Data Engine processes.

Analytics Services, composed of the VizQL and Cache Server processes, provide user-facing visualization and analytics services and caching services.

Sharing and Collaboration, and Content Management Service are powered by the Application Server process. Core Tableau Server functionality such as user login, content management (projects, sites, permissioning, etc.) and administration activities are provided by the Application Server process.

All of the above services use and rely on the Repository process, which contains structured relational data like metadata, permissions, workbooks, data extracts, user info, and other data. The File Store process enables data extract file redundancy across the cluster and ensures extracts are locally available on all cluster nodes. Under heavier loads, extract files are available locally across the cluster for faster processing and rendering.

Tableau's architecture is flexible, allowing you to run the platform just about anywhere. You can install Tableau Server on-premises, in your private cloud or data center, on Amazon EC2, on Google Cloud Platform, or on MS Azure. Tableau analytics platform can also run atop virtualization platforms. We recommend you follow the best practices for each virtualization platform to ensure the best performance from Tableau Server.

Tableau and your data

When you install Tableau Server into your organization, it becomes a core component of the analytics pipeline to the data your users need. It's important to understand how Tableau Server interacts with your business data. Specifically, Tableau Server can store extracts of data in your organization. It can also connect to live data sources. How you choose to provide the data to your Tableau users is informed by a number of variables: data source type, user scenario, performance and access requirements, and infrastructure conditions.

Tableau Server has not been architected as a data warehouse server where static, native data files are housed. In fact, using Tableau Server as a traditional data warehouse is a poor use of your investment. Rather, when it comes to data storage, we recommend hosting optimized data extracts on Tableau Server. While a data extract is often a subset of a larger data source in your organization, you can also create extracts for data sources that are overtaxed during work hours by scheduling the extract refresh for off-hours.

Extracts are also useful for modeling data or to enable highly-performant visualization authoring. For example, to improve visualization authoring and interaction performance you may optimize extracts by filtering the source data to the essential fields for a given department or project.

Tableau Server also provides direct, authorized access to live data sources, allowing users to build and run complex filtered queries against a variety of connected data sources. For this scenario, Tableau requires highly performant network access to the data sources in your organization and to those in the cloud. Tableau Server and the target data sources also need to be properly sized to handle the processing load required by high-volume, complex data operations.

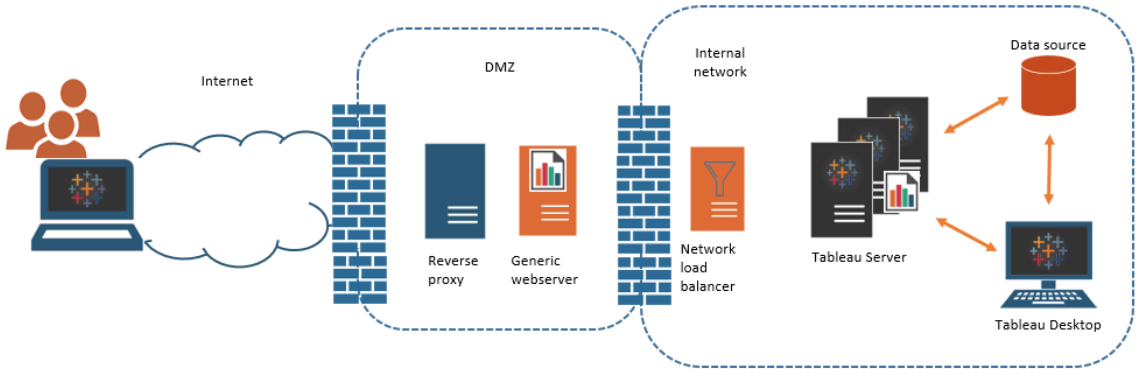
User access

Tableau Server is also a web-based collaboration platform, where users connect to share, view, and interact with data visualizations and data sources from a variety of devices. This means that Tableau Server must be accessible to Tableau users within your local protected network. You can also extend access to data visualizations to desktop, mobile, and authenticated web users outside your organization.

Tableau Server integrates with the following user authentication solutions: Active Directory, SAML, OpenId, and Kerberos.

Where should I install Tableau Server in my network?

Because of the highly-sensitive nature of most data that organizations manage with Tableau Server, and because Tableau Server requires access to internal data stores, Tableau Server must be run inside a protected network. Authenticated access from the internet is configured to connect to Tableau Server through a reverse proxy or a VPN solution.



Some organizations embed Tableau views in public webpages, or, for internal users, on generic web servers on the internal network.

Tableau Server can be configured to support such scenarios with either authenticated or anonymous access. For authorized access, where users can only view underlying data to which they have permission, you can configure trusted tickets with a generic web server. In this scenario, Tableau Server authorizes access to the underlying data in an embedded view. This scheme enables you to host interactive data visualizations on a web server in a DMZ or outside the protected network.

Anonymous access to embedded Tableau views requires that you enable "guest user" for Tableau Server. Guest user also requires that you license Tableau Server according to the number of cores you are running, rather than a named-user (interactor) model.

Sizing and scalability

Depending on the size and data usage in your organization, you can scale Tableau Server up or out. As you scale your server, you can also selectively allocate resources to meet your data needs and user needs.

When you scale up Tableau Server, you add hardware resources to a single server. For example, you might increase the memory and processing power of the computer running Tableau Server.

When you scale out Tableau Server, you add computers (or nodes). To create a highly available deployment with failover, you need at least three nodes. For example, you might run most CPU-intensive server processes on two nodes and use the third node for the gateway and coordination controller services.

Whether you scale up or scale out, you can selectively allocate resources by configuring the number and type of server processes that run. If your organization has a lot of data and creates a lot of data extracts, you can increase the number of processes that are dedicated to refreshing and storing extracts. Alternatively, if your organization wants to optimize for heavy user loads, you can increase the number of processes dedicated to responding to user

requests. Additionally, you can integrate Tableau Server into industry-standard network load balancers to further optimize your server for user requests.

Tableau Server management model

Tableau Server has been designed to support a management scheme with two high-level administrators: server administrator and site administrator. In small organizations, these roles may be assumed by the same person or team, but in larger organizations, the roles often diverge.

In this model, server administrators are IT professionals who maintain and deploy heterogeneous server solutions. Essential areas for server administrators may include networking, hardware tuning and maintenance, security and access, and managing users and directory services. The tools and documentation that we deliver with Tableau Server for the server administrator support these core server IT areas.

Site administrator, on the other hand, is an administrative role specific to Tableau Server or Tableau Cloud deployments. The Tableau site administrator is fundamentally concerned with data content. The site administrator manages users and their access to projects, workbooks, and data sources. To learn about sites and how to plan your deployment for them, see What is a site

Administrative roles

In some small organizations a single administrator may manage the entirety of Tableau Server. But for bigger enterprise organizations, Tableau Server usually requires at least three administrative roles for management at scale.

Tableau Server administrator

The Tableau Server administrator has access to administrative pages for creating and editing sites, adding users and setting roles, and many content-related tasks after the Tableau Server installation is complete. The Tableau Server administrator also creates and manages other server and site administrators, who in turn may manage sites, user groups, and projects.

For information about signing into Tableau Server as a Tableau Server administrator, see Sign in to the Tableau Server Admin Area.

TSM administrator

Tableau Services Manager (TSM) is a tool that gives server administrators command-line and web-based options for installing, upgrading, configuring, and maintaining Tableau Server. The TSM administrator installs the server and performs server-related administrative task like backing up server data, restoring backups, creating log archives, and managing multi-node clusters.

The TSM administrator must be an administrator on the local computer. See Sign in to Tableau Services Manager Web UI..

Common tasks performed by the TSM administrator include:

- Initial configuration of Tableau Server after installation
- Ongoing configuration management, including editing settings and changing the server topology
- Running administrative tasks such as backup, restore, and ziplogs

To learn more about TSM, see Tableau Services Manager Overview.

Tableau portal administrator

An important administrative role in a Tableau Server deployment is the Tableau customer portal administrator. The portal administrator manages licensing and the associated keys for the Tableau deployment. As the portal administrator, your first step is to purchase licenses on the Tableau Customer Portal. When you purchase licenses, the portal will return corresponding product keys. To renew your license, visit the Tableau renewal web page.

Tableau has a number of products (Desktop, Server, Prep Builder, and more). Each of the Tableau products require that you activate licenses by updating the Tableau software with the product keys that are purchased and stored on the Tableau Customer Portal. As the administrator who is tasked with activating Tableau licenses, it important that you understand the

relationship between licenses and keys. See Understanding License Models and Product Keys.

Management tools

Tableau Server includes a number of toolsets for managing the system:

- **Tableau Server administrator page**: This is the web-based administrative site that is installed on each Tableau Server instance. Tasks performed on the administrator page are day-to-day tasks for both server and site administrators. Server-related tasks include creating sites and site administrator accounts, optionally importing users, setting up synchronization with directory services, setting up extract refresh schedules, monitoring server performance and usage, and other global settings.

  Site-related tasks include managing content and assign permissions, running extract refreshes, create groups and projects, monitoring site activity, optionally adding users, and other content-related tasks.

  See Navigate the Admin Areas of the Tableau Web Environment.

  Permissions required for the Tableau Server administrator page are based on site roles. The site roles are generated and managed by Tableau Server.

- tsm Command Line Reference - This is the primary interface for server-wide configurations. Many configurations made with TSM CLI are rarely revisited after initial configuration. For example: SSL, subscriptions, data caching, service account, SMTP alerting, user authentication, and single-sign on configuration are all performed with TSM CLI.
- You can also Sign in to Tableau Services Manager Web UI.
- tabcmd: You can use the tabcmd command-line utility on a Windows or Linux computer to create scripts to automate administrative tasks on your Tableau Server sites. For example, use tabcmd for creating or deleting users, projects, and groups.
- REST API: With the Tableau Server REST API you can manage and change Tableau Server resources programmatically, via HTTP. The API gives you simple access to the functionality behind the data sources, projects, workbooks, site users, and sites on a Tableau server. You can use this access to create your own custom applications or to script interactions with Tableau Server resources.

Security

As an application server connecting to data that may be highly-sensitive, Tableau Server supports and implements a number of industry security standards. Our server admin documentation includes best practices and implementation for user authentication, authorization, data security, and network security. While our default installation is secure by design, we also recommend following the security hardening checklist to further lock down your deployment.

For more information about security audit compliance, vulnerability reporting, and other security resources, visit http://www.tableau.com/security.

Tableau Services Manager Overview

This article provides an overview of Tableau Services Manager (TSM), which you can use to configure and administer Tableau Server. The TSM CLI was introduced with Tableau Server on Linux, version 10.5. Beginning with version 2018.2, the TSM Web UI is available.

Functionality

TSM gives server administrators command-line and web-based options for configuring and maintaining Tableau Server, including performing administrative task like backing up server data, restoring backups, creating log archives, and managing multi-node clusters. For example, you use TSM to perform the following tasks:

- Initial configuration of Tableau Server after installation
- Ongoing configuration management, including editing settings and changing the server topology
- Running administrative tasks such as backup, restore, and ziplogs

For administrators familiar with earlier versions of Tableau Server, TSM replaces the following tools from previous versions of Tableau Server:

- Tableau Server Configuration utility
- tabadmin command line utility
- Tableau Server Monitor

Components

TSM consists of *services* (called *processes* in this documentation) and *clients*. TSM processes are administrative services which manage Tableau Server processes. TSM processes run continuously after TSM is initialized, even when the rest of Tableau Server is offline.

TSM processes that run, even when Tableau Server is stopped include:

- Administration Agent
- Administration Controller
- Client File Service
- Coordination Service (based on Apache Zookeeper™)
- Service Manager
- Licensing Service

For more information about TSM processes and Tableau Server processes, see Tableau Server Processes.

TSM Authentication

Whether you use the TSM Web UI, the command line interface, or the TSM API, you need to authenticate to Tableau Server before you can perform administrative tasks. This user account is distinct from Tableau Server user accounts, including Tableau Server administrators and site administrators.

TSM delegates authentication of users to the underlying operating system. On Linux, this means that authentication is handled using Pluggable Authentication Modules (PAM). PAM is the standard on all Linux distributions on which Tableau Server is supported. If your organization has configured PAM to authenticate with your directory service (Active Directory,

LDAP), then you can authorize any user from that directory service to access TSM. In this scenario, any authenticated PAM user that is a member of the `tsmadmin` group is authorized to access TSM.

In the 2019.1 release, TSM authentication process uses PAM directly and then falls back to an authentication scheme using *substitute user* (`su`) if PAM fails or is not configured with a directory service. If PAM is not configured with a directory service then local accounts must be managed on the Linux computer. In these cases, TSM will use the `su` method of authentication: passing the user-provided credentials to run the `true` command in the /bin directory. If that command succeeds, then authentication is verified. Therefore, if the user is a member of the `tsmadmin` group, then the authenticated user is granted access to TSM.

Custom PAM service definition

TSM uses the standard PAM *login* service to authenticate. You can further customize TSM authentication behavior by creating a `tableau` PAM service file in `/etc/pam.d`. If this file exists, then it will be consulted instead of the PAM login service.

TSM authorization group

You authenticate to TSM with a user that exists on the Tableau Server computer. The TSM user account must use password-based authentication. By default, the TSM user account must be a member of the `tsmadmin` group on the computer where Tableau Server is running. Alternatively, you can specify a different authorization group for TSM administration. To specify a different default group during the install process, see Help Output for initialize-tsm Script. To specify a different authorization group after you have installed Tableau Server, see Configure a Custom TSM Administration Group.

Connecting TSM clients

As a security measure, you can only connect to TSM with clients (CLI, Web UI, Rest API) over HTTPS. This is because TSM allows you to perform administrative tasks and to connect to TSM from other computers.

When you are connecting with a TSM client, you must connect to the Tableau Server instance running the TSM Administration Controller service.

As a security best practice, do not expose the TSM port (by default, `8850`) to the internet.

> **Note**: The TSM CLI tool does not require admin credentials in some scenarios. Specifically, if the account you are logged in as is a member of the TSM-authorized group, you do not need to provide credentials to run commands when running tsm CLI locally. For more information, see Authenticating with tsm CLI.

TSM HTTPS connections rely on a self-signed certificate generated by the Tableau Server installer. This certificate is the Tableau installation CA certificate that signs the SSL certificates Tableau creates for encrypting traffic over HTTP. The Tableau installation CA certificate must be trusted by the systems connecting to TSM Administration Controller.

The TSM CLI client validates certificate trust from a different store than the TSM Web UI uses. The TSM CLI client refers to the trusted store in the local Java keystore to validate trust for CA certificates. Since the TSM Web UI must establish connection with a web browser, trust is validated with the operating system's trusted keystore. The difference in how CA certificates are stored determines different trust configuration scenarios as outlined here:

- For TSM CLI communications on Tableau Server, the certificate trust is configured by default as part of the installation, node bootstrap, and upgrade processes. The Tableau installation CA certificate is added to the trusted store in the Java keystore. This allows you to access TSM using the CLI from any computer in the cluster without additional configuration. However, when accessing TSM Web UI, the browser will prompt you to trust the host running TSM Administration Controller service.
- For TSM CLI connections from remote computers, you will be prompted to trust the Tableau installation CA certificate the first time you connect to the Tableau Server running TSM Administration Controller. You can choose to trust the CA certificate, in which case you will not be prompted again on that computer until the certificate expires (default is 3 years). Or you can connect with a one-time trust by running your TSM command with the `--trust-admin-controller-cert` flag.

- For TSM Web UI connections from remote computers, the browser will prompt you to trust the host running TSM Administration Controller service.

Infrastructure Planning

Before you install Tableau Server, you should review the disk requirements, recommended configurations, user accounts, security, and networking requirements.

Validating your server deployment plan

Before you commit to installing a new Tableau Server deployment in your organization, be sure to carefully evaluate your options. For most organizations, Tableau Cloud will provide a more reliable, performant, and cost-effective analytics solution when compared to self-hosting Tableau Server. For information about the viability of Tableau Cloud for your organization, review this blog post, Should I move my analytics to the cloud?

Already running Tableau Server and want to migrate to Tableau Cloud? See Tableau Cloud Manual Migration Guide.

If you have determined that you must self-host Tableau Server, we recommend following the prescribed Tableau deployment in the Enterprise Deployment Guide (EDG). The EDG presents a fully-tested and supported, performant, scalable, secure reference architecture based on a tiered data network. Moving forward, we are committed to investing in the EDG reference architecture to ease feature rollouts and improve upgrade scenarios.

Before you install...

Note: You can find additional information about technical specifications for Tableau Server on the Tableau web site, here.

This topic includes requirements and recommendations that you must consider before you install Tableau Server into a production environment.

- If you are new to Tableau Server, and you want to deploy it in your organization, we encourage you to deploy Tableau Server as a single server in a test environment first. The easiest way to do a single-server installation is to follow the steps in Jump-start Installation.
- For an end-to-end procedure that describes how to deploy an enterprise-ready, four-node, reference architecture in a tiered data center, see Tableau Server Enterprise Deployment Guide.
- If you are deploying Tableau Server in a distributed cluster, review Distributed Requirements in addition to the requirements and recommendations described in this topic.
- If you are migrating from Tableau Server on Windows to Tableau Server on Linux, see Migrate Tableau Server from Windows to Linux.

Validating your server deployment plan

Before you commit to installing a new Tableau Server deployment in your organization, be sure to carefully evaluate your options. For most organizations, Tableau Cloud will provide a more reliable, performant, and cost-effective analytics solution when compared to self-hosting Tableau Server. For information about the viability of Tableau Cloud for your organization, review this blog post, Should I move my analytics to the cloud?

Already running Tableau Server and want to migrate to Tableau Cloud? See Tableau Cloud Manual Migration Guide.

If you have determined that you must self-host Tableau Server, we recommend following the prescribed Tableau deployment in the Enterprise Deployment Guide (EDG). The EDG presents a fully-tested and supported, performant, scalable, secure reference architecture based on a tiered data network. Moving forward, we are committed to investing in the EDG reference architecture to ease feature rollouts and improve upgrade scenarios.

Hardware recommendations for production installations

The following list describes the minimum hardware recommendations for a production use, single- node installation of Tableau Server:

Important: These recommendations are minimums and may not reflect the requirements for your installation and organization. For example, there are a number of factors that can impact disk space requirements, including whether or not you will be publishing extracts, flows, and the number of workbooks to Tableau Server. For more information on what might impact free disk space requirements, see Disk Space Requirements.

Install Type	*Processor*	*CPU*	*RAM*	*Free Disk Space*
Single node	• 64-bit (x86_64 chipsets) • Must support SSE4.2 and POPCNT instruction sets • ARM-based processors are not supported	8 cores (16 vCPUs), 2.0 GHz or higher	Version 2022.3 and later: • 128 GB Version 2021.4.0 to version 2022.1.x: • 64 GB Version 2021.3.x and earlier: • 32 GB	50 GB
	If you are adding Tableau Prep Conductor to your Tableau Server installation, we recommend you add a second node and dedicate this to running Tableau Server Prep Conductor. This node should have a minimum of 4 cores (8 vCPUs), and 16 GB of RAM.			
Multi-node and enterprise deployments	Contact Tableau for technical guidance. Nodes must meet or exceed the minimum hardware recommendations, except:			

Install Type	*Processor*	*CPU*	*RAM*	*Free Disk Space*
	• Dedicated Backgrounder nodes running up to two instances of backgrounder, where 4 cores may be acceptable. • Dedicated node for Tableau Prep Conductor: Minimum of 4 cores (8 vCPUs), and 16 GB of RAM. • Dedicated node for Independent Gateway: Minimum of 2 cores (4 vCPUs), 8 GB of RAM, and 100 GB free disk space.			

Important: The disk space requirement cannot be checked until you initialize TSM. If you don't have enough space, you won't be told this until after you install the Tableau Server package.

50 GB disk space available, with a minimum of 15 GB allocated to the `/opt` directory, and the remainder allocated to the `/var` directory for data storage.

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space. You may need to allocate additional disk space depending on various factors like whether you will be using extracts.

  The core Tableau Server bits must be installed in a directory with at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup. By default the install location is the `/opt` directory. You can change the installation path for Tableau Server on RHEL distros.

  If you plan to make heavy use of extracts then you may need to allocate additional disk space.You can specify a different directory for data (extract) storage during installation.

- **Network attached storage space requirements for External File Store:** If you are planning to configure Tableau Server with External File Store, you will need to estimate the amount of storage space to dedicate on your network attached storage.

Estimating the storage size: You must take into account the amount of storage needed for publishing and refreshing extracts. In addition, you must also take into account the repository backup size unless you specifically choose the option to do your repository backup separately as described in the Option 2: Back up repository separately topic.

- Extracts:
  - Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out how many extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.
  - Consider the space needed by the temp directory during an extract refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.
- Repository Backup:
  - To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base` directory.
  - To obtain the exact size of the repository data, open the backup file and use the size of the `workgroup.pg_dump` file.

- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the purposes of counting cores.
- RAM shown is the minimum recommended for a single-node installation. Your installation may function better with more RAM, depending on activity, number of users, and background jobs, for example.

To see the full list of recommendations and to see the minimum requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server. For hardware specifications Tableau uses internally for testing scalability, see Hardware recommendations for production installations.

For public cloud deployments on Amazon Web Services and Google Cloud Platform, their "vCPU" is actually a CPU hyper-thread, and not a full CPU core. When sizing cloud instances, you will need twice as many vCPU as the Tableau Server CPU core requirements given (8 vCPU required for a minimum trial installation, 16 vCPU recommended for a single-node installation).

Operating system requirements

The following distributions of Linux are supported:

	2021.-4.x	**2022.1-.0 - 2022.1-.11**	**2022.1-12+**	**2022.-3.0 - 2022.-3.3**	**2022.3-.4+**	**2023.-1.0 - 2023.-1.7**	**2023.1-.8+**	**2023.-3.0**	**2023.-3.1 - 2024.-2.x**
AlmaLinux 8.x									✓
AlmaLinux 9.x									✓
Amazon Linux 2	✓	✓	✓	✓	✓	✓	✓	✓	✓
Amazon Linux 2023									✓
CentOS 7.9+ (not 8.x)	✓	✓	✓	✓	✓	✓	✓	✓	✓
CentOS Stream 8.x									✓

	2021.-4.x	2022.1-.0 - 2022.1-.11	2022.1.-12+	2022.-3.0 - 2022.-3.3	2022.3-.4+	2023.-1.0 - 2023.-1.7	2023.1-.8+	2023.-3.0	2023.-3.1 - 2024.-2.x
CentO-S Stream 9.x									✓
Debian 9	**Note:** As of July 2022, Debian distributions are no longer supported. For more information, see this Tableau Community post.								
RHEL 7.3+	✓	✓	✓	✓	✓	✓	✓	✓	✓
RHEL 8.3+	✓	✓	✓	✓	✓	✓	✓	✓	✓
RHEL 9.x								✓	✓
Oracle Linux 7.3+ (not 8.x)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Oracle Linux 8.x									✓
Oracle Linux 9.x									✓
Rocky Linux 8.x									✓

	2021.-4.x	2022.1-.0 - 2022.1-.11	2022.1-.12+	2022.-3.0 - 2022.-3.3	2022.3-.4+	2023.-1.0 - 2023.-1.7	2023.1-.8+	2023.-3.0	2023.-3.1 - 2024.-2.x
Rocky Linux 9.x									✓
Ubuntu 16.04 LTS	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ubuntu 18.04 LTS	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ubuntu 20.04 LTS			✓		✓	✓	✓	✓	✓
Ubuntu 22.04 LTS							✓		✓

For information about Linux distribution requirements for Tableau Server in a Container, see Supported distributions for building .

Important: To ensure you have the latest security and functionality patches, Tableau strongly recommends that you use the latest supported version of the Linux distribution you are deploying on. Tableau generally tests and validates on the latest minor version of a supported distribution major version.

Additional notes on Linux distributions:

- Red Hat Enterprise Linux (RHEL), CentOS, Oracle Linux, and Amazon Linux distributions are collectively referred to in this documentation as RHEL-like.

- As of July 2022, Debian distributions are no longer supported. For more information, see this Tableau Community post.
- Non-LTS releases of Ubuntu are not supported.
- Ubuntu version 17.04 is not supported.
- Ubuntu version 20.04 support was added in Tableau Server version 2023.1.0, and in Server maintenance releases 2022.1.12 and 2022.3.4. It is not supported in earlier versions.
    - Installing Tableau Server 2023.1 and later on a physical machine running Ubuntu Linux 20.04 results in an installation error. For more information, see the Error "One or more control plane service(s) are in a non-active state" knowledge article.
- Previous versions of CentOS and Ubuntu are not supported because Tableau Server requires `systemd` for process management.
- The version of the installer with the file suffix, `.deb`, installs on both Ubuntu and Debian distributions.
- Custom kernels are not supported.

In a multi-node installation of Tableau Server, all of the computer nodes where you are installing Tableau must run Linux and the same distribution of Linux.

Installation directory

The core Tableau Server bits are installed in the `/opt` directory by default.

- The directory where you install Tableau Server must have at least 15 GB of free disk space allocated to it. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup.
- You can specify a non-default install location on RHEL-like systems, but cannot change the location on Ubuntu.

- Do not specify a symbolic link or a directory location on a Network File System (NFS) volume when specifying a non-default install location on RHEL-like systems.

Data directory

By default, Tableau Server will create a data directory for all content and extracts that are managed by Tableau. The directory is created at `/var/opt/tableau/tableau_server`.

You can specify a different directory for data (extract) storage during installation. If you plan to use a different directory, do not create the directory. Instead, let Tableau Server setup create the directory. The data directory requires specific permissions that are set during the installation process.

To change the data directory, you must pass a flag along with the data directory path when you run the `initialize-tsm` script. See Help Output for initialize-tsm Script.

If you are changing the default data directory:

- Do not specify a symbolic link or a data directory location on a Network File System (NFS) volume.
- Do not specify a data directory location with a path that includes a period or space. If there is a period or space in the path, initialization will fail.
- The data directory must be installed into a different directory than the installation directory.

Important: You cannot change the data directory location after you've run `initialize-tsm`. The data directory location will persist for the life of the deployment, including subsequent upgrades.

Tableau Prep Conductor

Tableau Prep Conductor is one of the process on Tableau Server. It runs a flow, checks connection credentials, and sends alerts if a flow fails. Tableau Prep Conductor leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to

update the flow output instead of logging into Tableau Prep Builder to manually run individual flows as your data changes.

Tableau Prep Conductor is licensed separately and is available through the Data Management license. For more information on Tableau Prep Conductor licensing, see License Data Management.

We recommend you enable Tableau Prep Conductor on a dedicated node. For more information:

- If you are installing a new Tableau Server, see Step 1 (New Install): Install Tableau Server with Tableau Prep Conductor.
- If you are adding Tableau Prep Conductor to an existing installation of Tableau Server, see Step 1 (Existing Install): Enable Tableau Prep Conductor.

Additional requirements

Make sure that your environment also meets the following additional requirements:

Hostname

- Tableau Server must be able to resolve the hosthame to an IP address either using the domain name server (DNS) or with a local host file on the computer running Tableau Server. By default, host files are stored at `/etc/hosts`.
- The hostname of the server must not change after you start Tableau Services Manager during the setup process. For example, this might happen if you use the cloud-init package to initialize a virtual machine, and you install Tableau Server on that virtual machine.
- Hostnames that include underscores (_) are not supported by Tableau Server.

Static IP address

The computer where you install Tableau Server must have a static IPv4 or IPv6 address.

Database drivers

To connect to specific data sources, the computer where you install Tableau Server must have the correct database drivers installed. For more information, see Database Drivers.

Available ports

TSM and Tableau Server each require an available TCP port in order for you to access them. TSM defaults to port 8850, and the Tableau Server Gateway service defaults to port 80. We strongly recommend that you ensure that both port 8850 and 80 are not in use on your system before installing Tableau Server. If those ports are not available, the TSM and gateway ports may be dynamically remapped to different port numbers, and there is currently no interface for displaying which port they have been remapped to.

See Tableau Services Manager Ports.

Local firewall configuration

If you are running a firewall on the computer where you will be installing Tableau Server, then you will need to open the following default ports for Tableau Server traffic. All port numbers, except 443 can be changed.

Port	TCP/UDP	Used by ...	TYPE OF INSTALLATION	
			All	Distributed / High Avail-ability
80	TCP	Gateway	X	
443	TCP	SSL. When Tableau Server is configured for SSL, the application server redirects requests to this port. Do not change this port.	X	
8850	TCP	Tableau Services Manager.	X	
8060	TCP	PostgreSQL database.	X	

Port	TCP/UDP	Used by ...	TYPE OF INSTALLATION	
			All	Distributed / High Avail-ability
8061	TCP	PostgreSQL backup verification port	X	
8000-9000	TCP	Range of ports reserved by default for dynamic mapping of Tableau processes		X
27000-27009	TCP	Range of ports used by Tableau Server for License service. This range must be open on the node running the License service and accessible from other nodes. By default, the initial node runs the License service.	X	

See Tableau Services Manager Ports and Configure Local Firewall.

System user and groups

Tableau Server on Linux uses one unprivileged user, and two groups for proper operation. Tableau will create the default account and groups during setup. Alternatively, you can specify existing accounts. See System user and groups
and TSM authorization group.

Sudo and root access

All installation tasks and administrative tasks for Tableau Server must be run as root. Often this is accomplished using the sudo command, but running the commands directly as the root user is also possible.

To install Tableau Server with the root account, you must specify a user account during installation. The account will be used for managing TSM. Specify the account by running the initialize-tsm script with the `-a` option. See Help Output for initialize-tsm Script.

Account password

The user account that you use to install and administer Tableau Server must be able to authenticate with a password. That is, the user must not use another means of authenticating (for example public key authentication).

If the account you are using to install and initialize Tableau Server does not have a password, you can set one using the `passwd` command:

```
sudo passwd $USER
```

Port access requirements

If you want to install Tableau Server remotely, for example by means of SSH, ensure that the following ports are open:

- `8850`. The port used for the Tableau Services Manager (TSM) web interface. You can use this interface to configure Tableau Server.
- `80`. The port used for the main Tableau Server web interface.

The Tableau Server installer attempts to open these ports during the installation process, but it can only open these ports for the `firewalld` firewall. If your computer runs another firewall, you must open the ports before you install.

Virtual Container environments

Beginning with version 2021.2, certain configurations of Tableau Server on Linux can be run in a container. For details on supported configurations, see Tableau Server in a Container.

Previous versions of Tableau Server on Linux and unsupported configutations have not been tested and are not supported in virtual container environments such as Docker. In these cases, Tableau Server on Linux will not function as expected if installed in these environments.

Package requirements

Systemd

Tableau Server requires `systemd`to manage services. This package is installed by default on CentOS 7 and Ubuntu 16. If you decide to test Tableau Server on a modified version of these distributions, you can run the following command to confirm that `systemd` is installed:

```
whereis systemd
```

If `systemd`is installed, the installation location is displayed. For example, you might see the following output:

```
systemd: /usr/lib/systemd /etc/systemd /us-
r/share/systemd /usr/share/man/man1/systemd.1.gz
```

If you have systemd installed but the Tableau installer is failing requirements checks for systemd, it's likely that systemd is not running. To verify that systemd is running, run the following command:

```
ls /run/systemd
```

The output will be a list of files and directories. If systemd is running, the output will include `system`. If `system` is not in the output, then systemd is not running.

Antivirus software

Antivirus software that scans directories used by Tableau Server can interfere with installation and ongoing use of Tableau Server. In some cases, this can result in installation failures, problems starting Tableau Server, or impacts to performance. If you plan to run antivirus software on the computer running Tableau Server, follow the recommendations in the Knowledge Base.

Continue to the next step: Install and Configure Tableau Server.

Disk Space Requirements

In general, when estimating the amount of additional disk space to allocate for Tableau Server for day-to-day usage, you must consider whether or not extracts will be published to Tableau Server, and consider the number of workbooks that you expect to publish to Tableau Server . If you anticipate using extracts, Tableau recommends that you begin with a few

hundred gigabytes (GB). If you do not anticipate using extracts, you may only need around 50 GB to fulfill your usage needs. To setup drive space alerts, see Configure Server Event Notification.

> Looking for Tableau Server on Windows? See Disk Space Requirements.

It is critical for Tableau Server to have adequate disk space. If you run out of disk space on any node in a Tableau Server installation, you can experience erratic performance, including not being able to access Tableau Server or the TSM Web UI. For troubleshooting steps, see the Tableau Knowledgebase.

Here are the factors that affect disk space requirements and where you might choose to install Tableau Server:

Publishing extracts to Tableau Server

Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out how many extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.

Refreshing extracts

Consider the space needed by the temp directory during an extract refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.

Creating many workbooks

If using workbooks, consider the number of workbooks that will be published to Tableau Server. Individual workbooks tend to take up a small amount of disk space. However, if you anticipate thousands of workbooks being published, you may want to allocate additional disk space to accommodate those workbooks.

Logging

To assist with daily management and troubleshooting, Tableau Server creates log files as a part of its normal operations. Depending on the level at which the logging is configured, it can significantly impact the amount of disk space necessary on the Tableau Server computer.

Backup and restore processes

The free disk space required to create a backup varies depending on the amount of data in the Tableau Server repository and file store services, and their collocation with the tabadmincontroller service. During backups, the background tasks for cleaning up old extracts are temporarily paused. This means that, for the duration of the backup, extract refreshes will leave extra files in place, adding to disk space usage. If your backup takes a long time, or if your organization uses many extracts that are regularly updated, this can result in a significant amount of temporary disk space usage. These temporary files will be removed after the backup is complete.

The following table lists the disk space requirements for backup based on whether the node hosts the repository, file store, controller, or some combination of them. In multi-node Tableau Server environments you need to estimate the required disk space on each node.

Repository	File Store	Controller	Disk Space Required
			3x repository data + 250 MB To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base` directory. To obtain the exact size of the repository data, open the backup file and use the size of the workgroup.pg_dump file.
			1.5x file store data To obtain an estimate of file store data (extracts, flows, etc.), check the size of `<data directory>/dataengine` directory.

		✓	3x repository data + 250 MB + 2.5x file store data
✓	✓		3x repository data + 250 MB + 1.5x file store data
	✓	✓	3x repository data + 250 MB + 1.5x file store data
✓		✓	3x repository data + 250 MB + 2.5x file store data
✓	✓	✓	3x repository data + 250 MB + 1.5x file store data

Restore disk space requirements

You must have adequate disk space for the database restore process to run successfully.

To restore Tableau Server:

- On controller nodes, you need free space equal to at least the size of the backup archive.
- On repository nodes, you need free space equal to at least three times the size of the repository data in the backup archive, plus 250 MB, plus the size of the pgsql data directory.
- On file store nodes, you need free space equal to at least twice the size of the dataengine folder in the backup archive.

Site Export and Import

Site exports and imports require adequate disk space, just as backup and restore processes do. At a minimum, you must have enough space for the exported file. You can use the requirements for Backup and Restore as a guideline for amount of disk space necessary when doing site exports and imports.

The export file is generated to the directory defined in the TSM `basefilepath.site_export.exports` variable. For additional details, see tsm sites export.

For more information about file paths and how to change them, see tsm File Paths.

Recommended Baseline Configurations

Determining the topology (number of nodes, number of Tableau Server processes) of your Tableau Server deployment requires you to consider these variables: your environment, sources of data and management to provide self-service data access, workload, and usage. However you may not have enough information about these variables when you deploy Tableau Server for the first time. This topic describes three baseline architectures that can be used as starting points for your Tableau Server installations.

Validating your server deployment plan

Before you commit to installing a new Tableau Server deployment in your organization, be sure to carefully evaluate your options. For most organizations, Tableau Cloud will provide a more reliable, performant, and cost-effective analytics solution when compared to self-hosting Tableau Server. For information about the viability of Tableau Cloud for your organization, review this blog post, Should I move my analytics to the cloud?

Already running Tableau Server and want to migrate to Tableau Cloud? See Tableau Cloud Manual Migration Guide.

If you have determined that you must self-host Tableau Server, we recommend following the prescribed Tableau deployment in the Enterprise Deployment Guide (EDG). The EDG presents a fully-tested and supported, performant, scalable, secure reference architecture based on a tiered data network. Moving forward, we are committed to investing in the EDG reference architecture to ease feature rollouts and improve upgrade scenarios.

Hardware recommendations for production installations

The hardware recommendations for production Tableau Server installations below are based on the hardware that the Tableau team uses to test Tableau Server scalability. We suggest that you use these recommendations as starting points for your production deployments. For Proof of Concept (PoC) deployments, we recommend you use Tableau Cloud. For more information, see Minimum installation hardware requirements.

Install Type	***Processor***	***CPU***	***RAM***	***Free Disk Space***
Single node	• 64-bit (x86_64 chipsets) • Must support SSE4.2 and POPCNT instruction sets • ARM-based processors are not supported	8 cores (16 vCPUs), 2.0 GHz or higher	Version 2022.3 and later: • 128 GB Version 2021.4.0 to version 2022.1.x: • 64 GB Version 2021.3.x and earlier: • 32 GB	500 GB - 1 TB
	If you are adding Tableau Prep Conductor to your Tableau Server installation, we recommend you add a second node and dedicate this to running Tableau Server Prep Conductor. This node should have a minimum of 4 cores (8 vCPUs), and 16 GB of RAM.			
Multi-node and enterprise deployments	Contact Tableau for technical guidance. Nodes must meet or exceed the minimum hardware recommendations, except:			

Install Type	*Processor*	*CPU*	*RAM*	*Free Disk Space*
	• Dedicated Backgrounder nodes running up to two instances of backgrounder, where 4 cores may be acceptable. • Dedicated node for Tableau Prep Conductor: Minimum of 4 cores (8 vCPUs), and 16 GB of RAM. • Dedicated node for Independent Gateway: Minimum of 2 cores (4 vCPUs), 8 GB of RAM, and 100 GB free disk space.			

Note: For deployments using virtual machines, Tableau recommends dedicated CPU affinity. If you are running Tableau Server in a virtual environment, use your VM host's best practices for vCPU allocation in relation to the number of physical CPU cores on the VM host. Typically 2 vCPUs = 1 physical core for Tableau Server. For example, for AWS installations, the 4 core minimum recommendation is equivalent of 8 AWS vCPUs. Similarly, follow the best practices provided by your virtual infrastructure provider to make sure Tableau Server has access to the appropriate compute, memory, and data resources. If you are installing Tableau Server in a virtual environment or a cloud-based deployment, see Virtual Machines and Public Cloud Deployments section later in this topic.

Estimating Disk Space

There are several factors that affect disk space requirements, including whether or not you will be publishing extracts, flows, and the number of workbooks to Tableau Server. For more information see Disk Space Requirements.

Baseline Configurations

- Single Server Installations
- Two Node Installation - Specialized for extract heavy environments

- Two Node Installation - Specialized for flow environments
- High Availability Installations (HA)

Single Server Installations

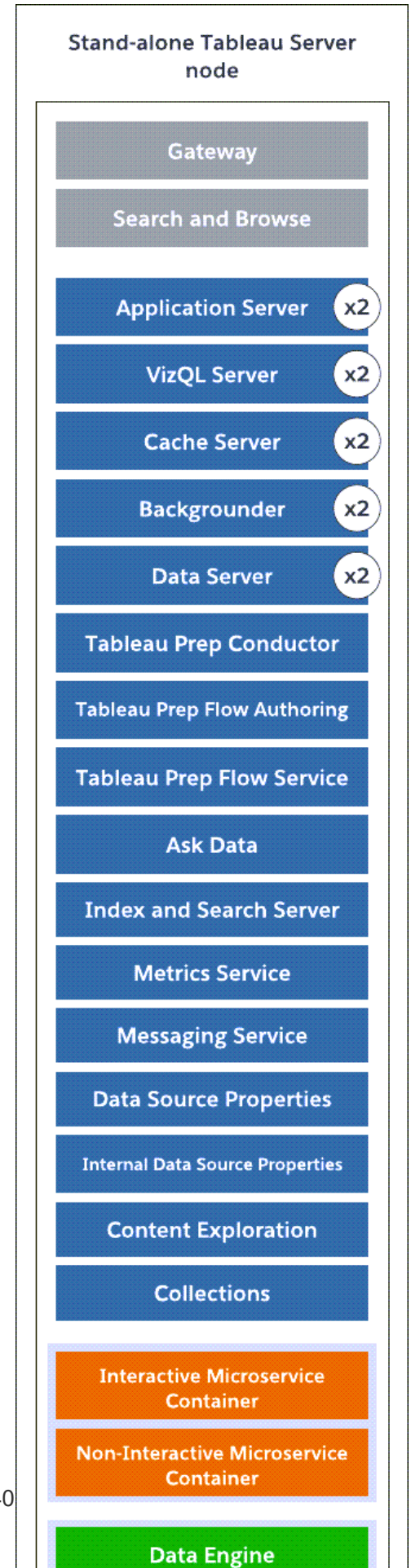
Recommendations

We recommend using a single machine to install Tableau Server for initial deployments with limited usage and are not mission critical. Single Server installations can also be expanded to multi-node installation as your workloads grow.

Here are some instances when a single server installation may not be right for you:

- If your system is considered mission critical and needs to be highly available. High availability is about minimizing the system downtime. It is achieved by eliminating single points of failure, and having a reliable failover mechanism. Tableau Server requires a minimum of a three-node configuration to provide redundancy and eliminate the single points of failure. This is one of the primary reasons to move to a multi-node configuration.
- If you have a lot of active users and a lot of extract refreshes, the two types of loads may be competing for the same resources on the machine. In such a scenario, a single server configuration may not be the right option as you may need additional specialized nodes to isolate the difference workloads.

Note: Active users represent the interactive, concurrent requests made to Tableau Server, including consuming dashboards on a laptop or mobile device, web authoring, and connecting to and querying Published Data Sources.


Stand-alone Tableau Server node
Gateway
Search and Browse
Application Server
x2
VizQL Server
x2
Cache Server
x2
Backgrounder
x2
Data Server
x2
Tableau Prep Conductor
Tableau Prep Flow Authoring
Tableau Prep Flow Service
Ask Data
Index and Search Server
Metrics Service
Messaging Service
Data Source Properties
Internal Data Source Properties
Content Exploration
Collections
Interactive Microservice Container
Non-Interactive Microservice Container
Data Engine

- Stand-alone single server node with all the processes installed on one machine.
- Below are the number of processes for an 8 core machine:
    - VizQL Server: Set to 2 instances (Number of physical cores divided by 4, up to a maximum of 4).
    - Backgrounder, Cache Server, and Data Server: Set to 2 instances.
    - All other processes, only one instance of the process is installed, regardless of hardware.

Note: One instance of Tableau Prep Conductor is automatically configured with Backgrounder, when you have the Data Management Product Key activated on your server. However, It is recommended that you have a dedicated node for Tableau Prep Conductor. If you plan to have flows on your Tableau Server, we recommended you use two or more nodes and dedicated one of these nodes to run only flows. The example configuration described above does not include Tableau Prep Conductor since it is a single node server.

Multi-Node Installations

Running Tableau Server on more than one machine is called a multi-node installation, or a cluster. There are various reasons why you might want to have a multi-node installation. For example, you may have heavy extract environments which can mean dedicating some hardware resources to Backgrounder process. For systems that have high availability requirements, you need a multi-node environment that has at least three nodes.

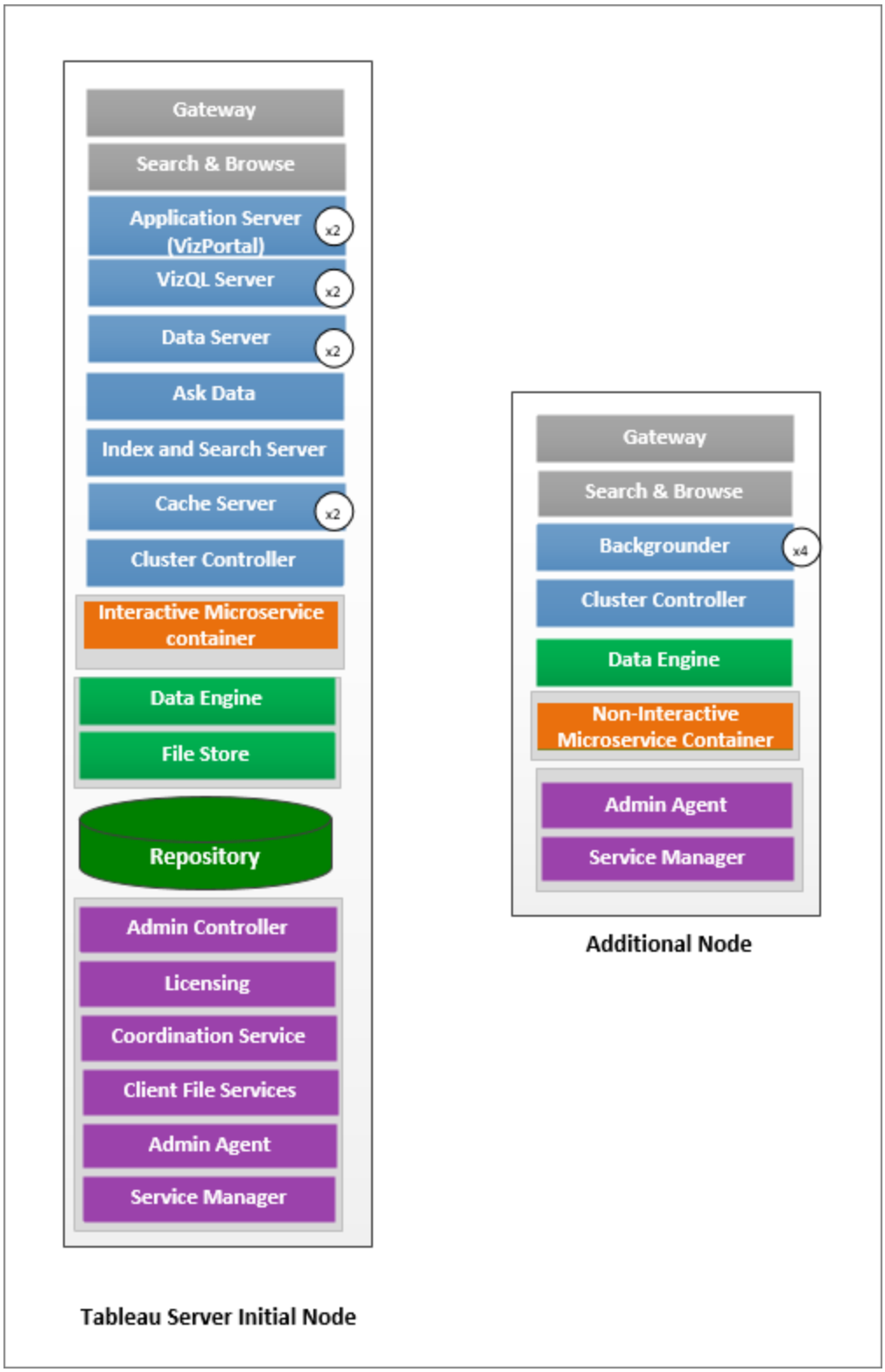
Two Node Installation - Specialized for extract heavy environments

Recommendations

Start with a two node configuration when the following conditions apply to you:

- **Extract heavy environment:** Majority of your data sources are extracts. Having just a few, extremely large extracts could put your deployment in this category, as would having very many small extracts.
- **Frequent extract refreshes:** Refreshing an extract is a CPU-intensive task. Deployments where extracts are frequently refreshed (for example, several times a day during business hours) are often helped by more emphasis on the background process, which handles refresh tasks.

Important: Two-node configurations do not meet the minimum requirements for high availability. If you need a system that is highly available, see High Availability Installations (HA).


Gateway
Search & Browse
Application Server (VizPortal)
x2
VizQL Server
x2
Data Server
x2
Ask Data
Index and Search Server
Cache Server
x2
Cluster Controller
Interactive Microservice container
Data Engine
File Store
Repository
Admin Controller
Licensing
Coordination Service
Client File Services
Admin Agent
Service Manager
Tableau Server Initial Node
Gateway
Search & Browse
Backgrounder
x4
Cluster Controller
Data Engine
Non-Interactive Microservice Container
Admin Agent
Service Manager
Additional Node

- On the initial node, install all the processes except for the backgrounder. Below is the number of instances of the processes for an 8 core machine:
    - VizQL Server: Set to 2 instances. (default calculation: Number of physical cores divided by 4, up to a maximum of 4).
    - Cache Server, and Data Server: Set to 2 instances. One instance of Ask Data is automatically configured on the node that has Data Server.
    - Index and Search Server : Index and Search Server memory can be configured to improve performance by using the `indexandsearchserver.vmopts` TSM configuration option. For more information, see tsm configuration set Options.
    - All other processes, only one instance of the process is installed, regardless of hardware. One instance of Interactive Microservice Container is installed on a node that has Application Server enabled, and one instance of Non-Interactive Microservice Container is installed on a node that has Backgrounder enabled.
- Isolate backgrounder on the additional node. To calculate the minimum number of backgrounder processes to run on this node, divide the computer's total number of physical cores by 4. To calculate the maximum number, divide the computer's total physical cores by 2. In the example shown above, both the nodes are on machines with 8 physical cores. When you install the backgrounder, Tableau Server automatically installs one instance of the Data Engine.

Note: This configuration assumes that you do not have Tableau Prep Conductor enabled on your Tableau Server. If are using Tableau Prep Conductor to schedule and manage flows, and have an extract heavy environment, we recommend that you have at least 3 nodes and use the 3 node configuration described later in this topic.

As you monitor and gather data about the performance and usage, you can fine tune and configure the number of instances for these processes. For example, on the node dedicated to running backgrounder you can initially set the number of backgrounders to the recommended

minimum (total number of cores divided by four), and increase the number of backgrounder processes later if you find that:

- Extract refreshes are taking a long time to complete
- Subscriptions and alerts are not completing on time

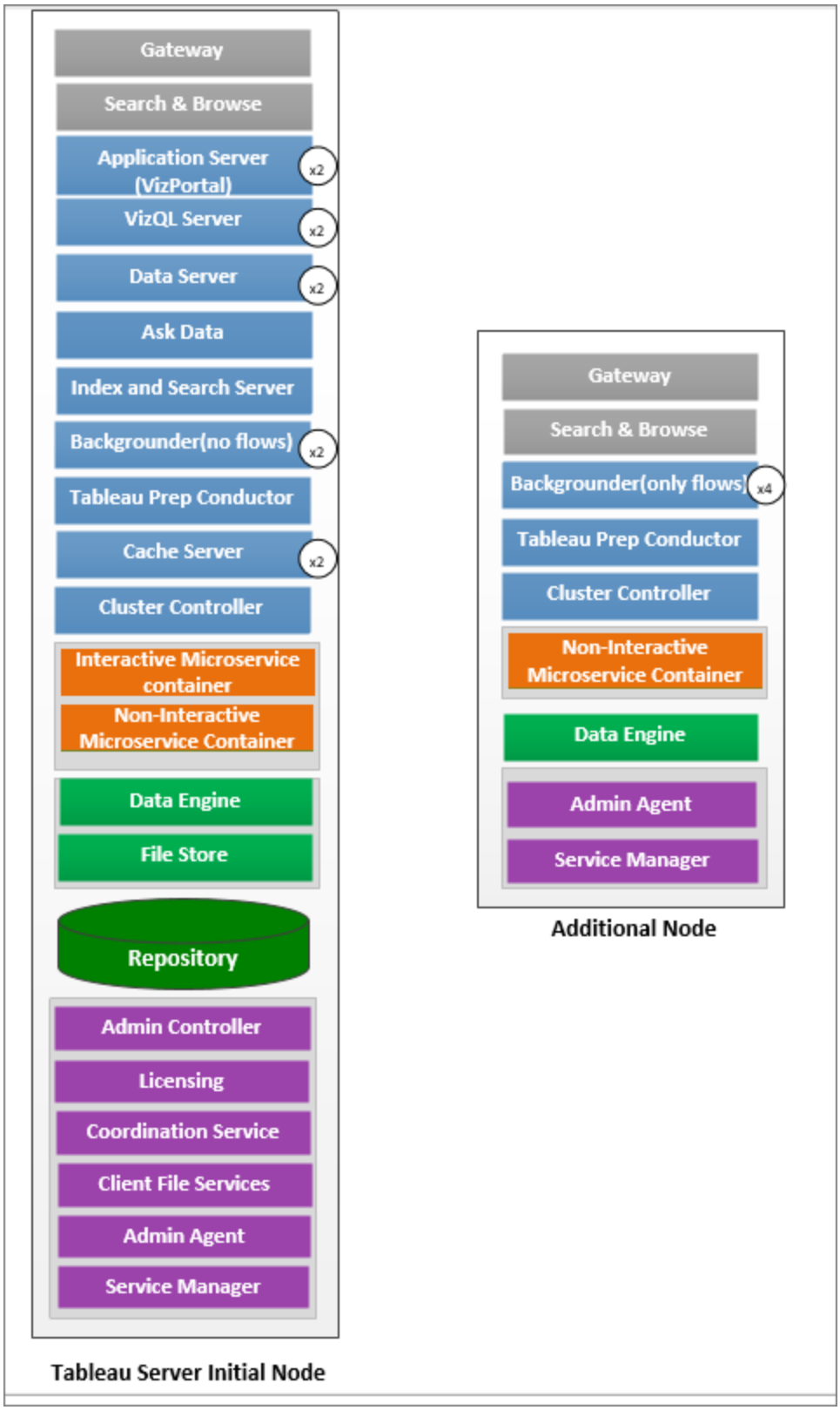
For dedicated backgrounder nodes, depending on workload and system resources, you may be able to increase the number of backgrounder instances beyond the maximum recommended above. Increasing backgrounder instances on the node can impact node functionality in positive and negative ways. You are responsible for carefully monitoring RAM and CPU resources, and other aspects of Tableau Server to determine the best configuration for your environment.

For more information on performance tuning, see Performance Tuning topic.

Two Node Installation - Specialized for flow environments

Start with a two node configuration if you are planning to publish, schedule, and manage flows on your Tableau Server.

Important: Two-node configurations do not meet the minimum requirements for high availability. If you need a system that is highly available, see High Availability Installations (HA).


Gateway
Search & Browse
Application Server (VizPortal)
x2
VizQL Server
x2
Data Server
x2
Ask Data
Index and Search Server
Backgrounder(no flows)
x2
Tableau Prep Conductor
Cache Server
x2
Cluster Controller
Interactive Microservice container
Non-Interactive Microservice Container
Data Engine
File Store
Repository
Admin Controller
Licensing
Coordination Service
Client File Services
Admin Agent
Service Manager
Tableau Server Initial Node
Gateway
Search & Browse
Backgrounder(only flows)
x4
Tableau Prep Conductor
Cluster Controller
Non-Interactive Microservice Container
Data Engine
Admin Agent
Service Manager
Additional Node

- On the initial node, install all the processes. Below is the number of instances of the processes for an 8 core machine:
    - VizQL Server: Set to 2 instances. (default calculation: Number of physical cores divided by 4, up to a maximum of 4).
    - Cache Server, and Data Server: Set to 2 instances. One instance of Ask Data is automatically configured on the node that has Data Server.
    - Backgrounder: Minimum 2, maximum 4. The diagram above shows the maximum for an 8 core node. Tableau Prep Conductor is automatically configured one the node where you have backgrounder installed. On the initial node, set the Backgrounder node role to run all job types including flows using the `tsm topology set-node-role` tsm configuration. For more information, see tsm topology set-node-role
    - Index and Search Server: Index and Search Server memory can be configured to improve performance by using the `indexandsearchserver.vmopts` TSM configuration option. For more information, see tsm configuration set Options.
    - All other processes, only one instance of the process is installed, regardless of hardware. One instance of Interactive Microservice Container is installed on a node that has Application Server enabled, and one instance of Non-Interactive Microservice Container is installed on a node that has Backgrounder enabled.
- Isolated the backgrounder on the additional node to run only flows. Use the `tsm topology set-node-role` tsm configuration to configure this setting. For more information, see tsm topology set-node-role

Note: If you have both a heavy extract environment, and schedule and manage flows on your server, we recommend that you use the 3 node configuration described below.

High Availability Installations (HA)

Recommendations

A highly available installation of Tableau Server is a distributed installation that is designed to maximize the availability of Tableau Server. High availability basically means that the system is available with minimal amount of downtime. To build in redundancy for HA related items such as repository, file redundancy, and failover, you need a **minimum of three nodes**. The tolerance for downtime will vary for each organization and depends on the SLAs you have established in your organization.

High availability is achieved by eliminating single points of failure and detecting failures and setting up a reliable failover system. HA in Tableau Server is mainly achieved by:

- File redundancy with multiple File Store/Data Engine instances.
- Active/Passive Repository across two nodes.
- Index and Search Server across all the three nodes.
- Adding an external load balancer to make sure your installation is robust to Gateway failures and make sure that requests only get routed to functioning Gateway processes.

Server Configuration

Three-node configuration:

- To build in redundancy, you need to add additional nodes to host instances of the repository and File Store/Data Engine processes. You can add instances of other processes, including multiple instances of a process on a node.

- To build redundancy for the type of backgrounder jobs, have one of the nodes (initial node in this example) run all type of jobs. Backgrounders run all types of jobs by default. On one of the additional nodes, set the backgrounder to run only flows, and the other additional node to run all jobs except for flows.
- The successful functioning of Tableau Server depends on a properly functioning Coordination Service. For server installations of three or more nodes, we recommend that you add additional instances of the Coordination Service by deploying a new Coordination Service ensemble. This provides redundancy and improved availability in the event that one instance of the Coordination Service has problems. For more information, see Deploy a Coordination Service Ensemble .
- Index and Search Server memory is added to all three nodes for redundancy and can be configured to improve performance by using the `index-andsearchserver.vmopts` TSM configuration option. For more information, see indexandsearchserver.vmopts.
- To reduce the system's vulnerability, you can run multiple gateways and additional instances of some of the server processes. The fewest number of computers required to achieve this configuration is three.
- The repository has also been moved from the initial node to one of the additional nodes, and a second, passive instance has been added to the other new node.
- One instance of Interactive Microservice Container is installed on a node that has Application Server enabled, and one instance of Non-Interactive Microservice Container is installed on a node that has Backgrounder enabled.

NOTE: In certain circumstances you may want to limit the processes running on your initial node. Reasons for doing this include wanting to run as few processes as possible on the node to limit processing requests on the node. You might also remove licensed Tableau Server processes from the node if you have a core-based license and do not

want the initial node cores to count against your core use. For more information on Tableau Server licensed processes, see Tableau Server Processes from the node.

Virtual Machines and Public Cloud Deployments

In general, the considerations and recommendations described in this topic apply to virtual environment and cloud deployments.

If you are running Tableau Server in a virtual environment, use your VM host's best practices for vCPU allocation in relation to the number of physical CPU cores on the VM host. Typically 2 vCPUs = 1 physical core for Tableau Server. For example, for AWS installations, the 4 core minimum recommendation is equivalent of 8 AWS vCPUs.

For more information on cloud-based deployments, see:

- Self-Host Tableau Server in a Public Cloud Service.

Beyond Baseline Configurations

If you are planning a system whose configuration is beyond the limits documented here, contact Tableau Professional Services.

High VizQL Peak Usage Considerations

For optimal performance for Tableau Server we recommend isolating the repository on a dedicated node in your deployment. If you have an Advanced Management license, consider running the repository as an external database.

If your organization has a peak load of more than 1000 VizQL sessions per hour, we also recommend running Tableau Server on Linux. In this scenario, VizQL sessions refer to any user actions that display or generate visualizations from Tableau Server.

For more information, see Tableau Server External Repository.

Disaster Recovery Considerations

While HA configurations reduce downtime, you may still encounter failures in case of a disaster or hardware failures. In addition to the above considerations, you should evaluate the importance of disaster recovery in your organization and plan for a deployment that helps you meet your disaster recovery goals and objectives.

When planning for disaster recovery (DR) in your Tableau environment, there are two main factors to consider:

- Recovery Time Objective (RTO), a measure of how much downtime your business can accept before a full recovery.
    - Influences how often you restore your backups to an alternative cluster and the amount of infrastructure investment.
- Recovery Point Objective (RPO), a measure of how much data loss your business can tolerate.
    - Influences how often you will need to take backups of your system.
    - For Tableau Server the RPO cannot be shorter than the time it takes to complete a full backup of your server.

The diagram below illustrates how to plan for a range of RTO requirements:

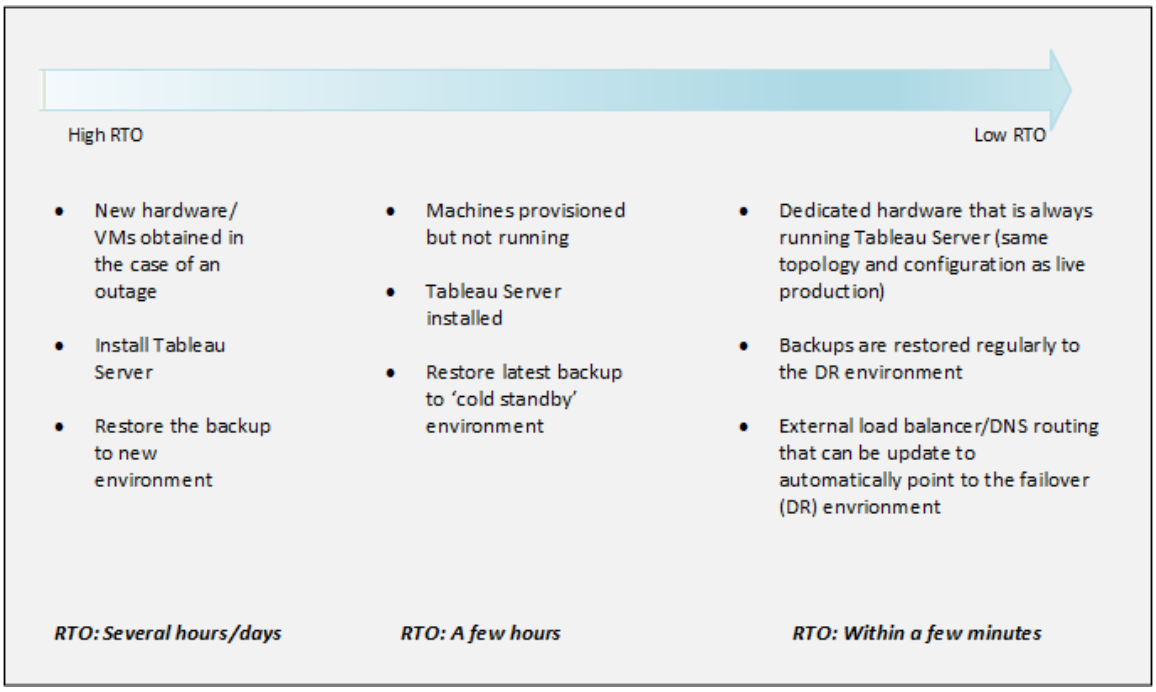
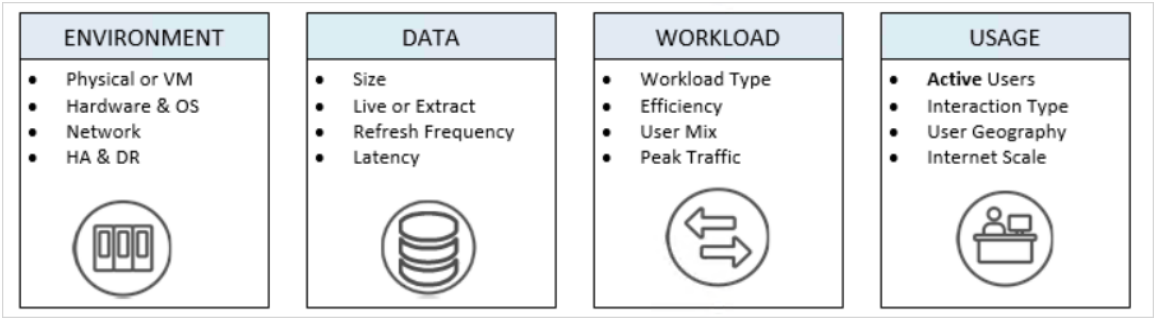


Tableau Server Scalability

These baseline configurations may not be enough as your needs change and grow, and you might need to scale your Tableau Server beyond these configurations. Like other enterprise platforms, Tableau Server scales up by adding processor, memory, and disk to existing node, and scales out by adding more nodes to a cluster. However, scalability and performance are heavily dependent on external systems and user activity. The configuration of your Tableau Server can be different depending on your requirements and variables:



For more information on Tableau Server scalability and the variables affecting scalability, see Tableau Server Scalability whitepaper.

Identity Store

Tableau Server requires an identity store to manage user and group information. There are two kinds of identity stores: local and external. When you install Tableau Server you must configure either a local identity store or an external identity store.

For information about configuration options for the identity store, see identityStore Entity and External Identity Store Configuration Reference. For more information about adding more flexibility to the single identity store model, see Provision and Authenticate Users Using Identity Pools.

Local identity store

When you configure Tableau Server with a local identity store, all user and group information is stored and managed in the Tableau Server repository. In the local identity store scenario, there is no external source for users and groups.

External identity store

When you configure Tableau Server with an external store, all user and group information is stored and managed by an external directory service. Tableau Server must synchronize with the external identity store so that local copies of the users and groups exist in the Tableau Server repository, but the external identity store is the authoritative source for all user and group data.

If you have configured the Tableau Server identity store to communicate with an external LDAP directory, then all users (including the initial admin account) that you add to Tableau Server must have an account in the directory.

When Tableau Server is configured to use an external LDAP directory, you must first import user identities from the external directory into the Tableau Server repository as system users. When users sign in to Tableau Server, their credentials are passed to the external directory, which is responsible for authenticating the user; Tableau Server does not perform this

authentication. However, the Tableau user names stored in the identity store are associated with rights and permissions for Tableau Server. Therefore, after authentication is verified, Tableau Server manages user access (authorization) for Tableau resources.

Active Directory is an example of an external user store. Tableau Server is optimized to interface with Active Directory. For example, when you install Tableau Server on an Active Directory domain-joined computer using the Configure Initial Node Settings, Setup will detect and configure most Active Directory settings. If, on the other hand, you are using TSM CLI to install Tableau Server, you must specify all the Active Directory settings. In this case, be sure to use the LDAP - Active Directory template to configure identity store.

If you are installing into Active Directory, we recommend that you review User Management in Deployments with External Identity Stores before you deploy.

For all other external stores, Tableau Server supports LDAP as a generic way to communicate the identity store. For example, OpenLDAP is one of several LDAP server implementations with a flexible schema. Tableau Server can be configured to query the OpenLDAP server. To do so, the directory administrator must provide information about the schema. During setup, you must use Configure Initial Node Settings to configure a connection to other LDAP directories.

LDAP bind

Clients that wish to query a user store using LDAP must authenticate and establish a session. This is done by binding. There are multiple ways to bind. Simple binding is authenticating with a username and password. For organizations that connect to Tableau Server with simple bind, we recommend configuring an SSL encrypted connection, otherwise the credentials are sent over the wire in plaintext. Another type of binding Tableau Server supports is GSSAPI binding. GSSAPI uses Kerberos to authenticate. In Tableau Server's case, Tableau Server is the client and the external user store is the LDAP server.

LDAP with GSSAPI (Kerberos) bind

We recommend binding to LDAP directory with GSSAPI using a keytab file to authenticate to the LDAP server. You will need a keytab file specifically for the Tableau Server service. We

also recommend encrypting the channel with the LDAP server using SSL/TLS. See Configure Encrypted Channel to LDAP External Identity Store.

If you are installing into Active Directory, and the computer where you are installing Tableau Server is already joined to the domain, then the computer may already have a configuration file and a keytab file. In this case, the Kerberos files are for the operating system functionality and authentication. Strictly speaking, you can use these files for GSSAPI bind, but we don't recommend using them. Instead, contact your Active Directory administrator and request a keytab specifically for the Tableau Server service. See Understanding Keytab Requirements.

Assuming your operating system has a properly configured keytab for authentication to the domain, then the Kerberos keyfile for GSSAPI bind is all you need for the base installation of Tableau Server. If you plan to use Kerberos authentication for users, then configure Kerberos for user authentication and Kerberos delegation to data sources after installation is complete.

LDAP over SSL

By default, LDAP with simple bind to arbitrary LDAP servers is not encrypted. User credentials that are used to establish the bind session with the LDAP server are communicated in plaintext between Tableau Server and the LDAP server. We strongly recommend that you encrypt the channel between Tableau Server and the LDAP server.

Beginning with version 2021.2, Tableau Server on Linux requires an encrypted LDAP channel when you use Active Directory as an identity store. You must install a valid SSL/TSL certificate before installing or upgrading to 2021.2 or newer. Although not recommended, you can also disable the default encrypted LDAP channel. For more information about enabling or disabling encryption for Active Directory and other LDAP servers, see Configure Encrypted Channel to LDAP External Identity Store.

System user and groups

Tableau Server on Linux uses one user, and two groups for proper operation. The user and groups can be local or from an LDAP directory service.

User

Tableau Server requires a service account. This account is an unpriviledged user with normal login privileges. By default, Tableau Server installation will created a local user, `tableau`, for the service account.

If you want to use an existing user account for the Tableau Server service account then you must disable account creation during installation.

Specifically, you will need to set the `--disable-account-creation` option when you run the initialize-tsm script. You will also need to specify the account name with the `--unprivileged-user` option. If the account that you specify does not exist, then the initialize-tsm script will create it. See Help Output for initialize-tsm Script for more details.

If you want to specify an existing account with the `--unprivileged-user` option, verify that the user account is an unprivileged user with normal login privileges. Configure the account with the following characteristics:

- Shell set to `/bin/bash`.
- For convenience, consider setting the home directory to the data directory path. The account must have ownership and write privileges to the home directory.

If you specify a different unprivileged account during setup, you must manually add that same user to the systemd-journal group. The unprivileged user must be a member of the systemd-journal group so that Tableau Server can collect logs from some services (such as Ask Data) when running the tsm maintenance ziplogs command. If the unprivileged user is not a member of the group, ziplogs will not contain logs from the affected services

Groups

Tableau Server requires two groups for operation.

In a default installation, the local `tableau` service account belongs to a primary group named `tableau`. However, if you specify an alternate unprivileged user during installation, then the

primary group for that alternate account will be used. As a convenience, any account can be added to this group to be able to read the Tableau Server log files (without becoming root).

The second group is used to authorize which users are authorized to authenticate to Tableau Services Manager (TSM). Any user in this group will be able to send commands to TSM, so it should be restricted to Tableau Server administrators. By default, this group is named `tsmadmin`.

If you are not going to use the default name, you will need to specify the group name with the `--tsm-authorized-group` option when you run initialize-tsm. See Help Output for initialize-tsm Script for more details.

Authenticating clients

Basic user authentication in Tableau Server is by username and password sign-in for both local and external user stores. In the local case, user passwords are stored as a hashed password in the repository. In the external case, Tableau Server passes the credentials to the external user store and awaits a response as to whether the credentials are valid. External user stores can also handle other kinds of authentication like Kerberos, but the concept is still the same, Tableau Server delegates the credentials or user to the external store and awaits a response.

You can configure Tableau Server such that username-password sign-in is disabled. In these scenarios other authentication methods, such as trusted authentication, OpenID, or SAML can be used. See Authentication.

In some cases, you may need to update LDAP external directories to allow bind operations with username + DN format from Tableau Server. See User binding behavior on sign in.

User Management in Deployments with External Identity Stores

This topic describes important technical details that you should be familiar with if you use an external identity store to manage users for Tableau Server. Tableau Server supports

connecting to an external directory using LDAP. In this scenario, Tableau Server imports users from the external LDAP directory into the Tableau Server repository as system users.

Arbitrary LDAP directories

The system username in Tableau is whatever attribute you set as part of LDAP configuration, for example "cn". This is true for both individual user import and group sync functionality. See External Identity Store Configuration Reference.

User binding behavior on sign in

You may need to update your LDAP configuration to allow binding with usernames appended with the DN. Specifically, you will need to update your LDAP configuration when Tableau Server is configured with an arbitrary LDAP directory. (e.g., OpenLDAP) that uses UPN or Email addresses as usernames.

Tableau Server will search for a given user based on the username that is supplied during sign in. Tableau Server will then attempt to bind with the username appended with the DN. In the case where Tableau Server has been configured with GSSAPI, then the username@REALM (domain name) will be used.

Active Directory

This content in rest of this topic assumes that you are familiar with Active Directory user management and basic Active Directory schema and domain concepts.

Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

Active Directory user authentication and Tableau Server

Tableau Server stores all user names in the Tableau Server identity store, which is managed by the repository. If Tableau Server is configured to use Active Directory for authentication, you must first import user identities from Active Directory to the identity store. When users sign in to Tableau Server, their credentials are passed to Active Directory, which is responsible for

authenticating the user; Tableau Server does not perform this authentication. (By default, NTLM is used for authentication, but you can enable Kerberos or SAML for single sign-on functionality—however, in all these cases, authentication is left to Active Directory.) However, the Tableau user names stored in the identity store are associated with rights and permissions for Tableau Server. Therefore, after authentication is verified, Tableau Server manages user access (authorization) for Tableau resources.

Active Directory user name attributes and Tableau Server

Active Directory uniquely identifies user objects using several attributes. (For details, see User Naming Attributes on the MSDN website.) Tableau Server relies on two Active Directory user naming attributes:

- `sAMAccountName`. This attribute specifies the logon name that was originally designed for use with older versions of Windows. In many organizations, this name is combined with the NetBIOS name for authentication, using a format like `example\jsmith`, where `example` is the NetBIOS name and `jsmith` is the `sAMAccountName` value. Due to the original design in Windows, the `sAMAccountName` value must be less than 20 characters.

  In the Windows **Active Directory Users and Computers** administrative console, this value is in the field labeled **User logon name (pre-Windows 2000)** on the **Account** tab of the user object.

- `userPrincipalName` (UPN). This attribute specifies a user name in the format `jsmith@example.com`, where `jsmith` is the UPN prefix and `@example.com` is the UPN suffix.

  In the Windows **Active Directory Users and Computers** administrative console, the UPN is a concatenation of two fields on the **Account** tab of the user object: the **User logon name** field, and the domain drop-down list next to it.

Adding users from Active Directory

You can add users individually from Active Directory, either by typing them in the server environment or by creating a CSV file and importing the users. You can also add Active Directory

users by creating a group via Active Directory and importing all of the group's users. The result can be different depending on which approach you're using.

Importing UPN prefix as username

You cannot import the whole UPN as a user name.

In most cases, the user name that Tableau Server will import into the identity store will be the sAMAccountName value. For more information about exceptions to this behavior, see the Importing UPN Prefix as Username in Non-Standard Scenarios with Active Directory in the Tableau knowledge base.

Adding user groups

If you import an Active Directory user group, Tableau will import all users from the group using the `sAMAccountName`.

Sync behavior when removing users from Active Directory

Users cannot be automatically removed from Tableau Server through an Active Directory sync operation. Users that are disabled, deleted, or removed from groups in Active Directory remain on Tableau Server so that you can audit and reassign the user's content before removing the user's account completely.

However, Tableau Server will act upon user objects differently based how the status of that user object changes in Active Directory. There are two scenarios: deleting/disabling users in Active Directory or removing users from synchronized groups in Active Directory.

When you delete or disable a user in Active Directory and then synchronize that user's group on Tableau Server, the following occurs:

- The user is removed from the Tableau Server group you synchronized.
- The user's role is set to "unlicensed."
- The user will still belong to the All Users group.
- The user is unable to sign in to Tableau Server.

When you remove a user from a group in Active Directory and then synchronize that group on Tableau Server, the following occurs:

- The user is removed from the Tableau Server group you synchronized.
- The users role is retained: it is not set to “unlicensed.”
- The user will still belong to the All Users group.
- The user will still have permission to the Tableau Server with access to everything that the All Users group is granted permission to use.

In both instances, to remove a user from Tableau Server, the server administrator must delete the user from the Server Users page in Tableau Server.

Domain nicknames

In Tableau Server, domain nickname is equivalent to the Windows NetBIOS domain name. In a Windows Active Directory forest, a fully qualified domain name (FQDN) can have an arbitrary NetBIOS name. The NetBIOS name is used as the domain identifier when a user logs in to Active Directory.

For example, the FQDN `west.na.corp.lan` might be configured with a NetBIOS name (nickname) of `SEATTLE`. The user `jsmith` in that domain could log on to Windows using either of the following user names:

- `west.na.corp.example.com\jsmith`
- `SEATTLE\jsmith`

If you want your users to sign in to Tableau Server with a NetBIOS name instead of the FQDN, then you'll need to verify that the nickname value for each domain where users log in is set. See editdomain for information on how to view and set the nickname value for each domain.

Support for multiple domains

You can add users and groups from a domain that's different from the domain of the Tableau Server computer in these cases:

- Two-way trust has been established between the server’s domain and the users’ domain.

- The server's domain trusts the users' domain (one-way trust). See Domain Trust Requirements for Active Directory Deployments.

The first time you add a user or group from the non-server domain, you must specify the fully qualified domain name with the user/group name. Any additional users or groups you add from that domain can be added using the domain's nickname, provided the nickname matches the NetBIOS name. If Tableau Server connects to multiple domains, you must also specify the other domains that Tableau Server connects to by setting the `wgserver.domain.whitelist` (version 2020.3 and earlier) or `wgserver.domain.accept_list` (version 2020.4 and later) option with TSM. For more information, see wgserver.domain.whitelist or wgserver.domain.accept_list.

Duplicate display names

If user display names are not unique across multiple domains, then managing users with the same display name in Tableau can be confusing. Tableau Server will display the same name for two users. For example, consider an organization with two domains, example.lan and example2.lan. If user John Smith exists in both domains, then adding that user to groups and other administrative tasks will be confusing in Tableau Server. In this scenario, consider updating the display name in Active Directory for one of the users to differentiate the accounts.

Sign in to Tableau Server with NetBIOS name

Users can sign in to Tableau Server using the domain nickname (NetBIOS name), for example, `SEATTLE\jsmith`.

Tableau Server cannot query for NetBIOS name for a given FQDN. As a result, Tableau sets the nickname of a given FQDN according to the first entry in the namespace. For example, given the FQDN `west.na.corp.lan` , Tableau sets the nickname to `west`.

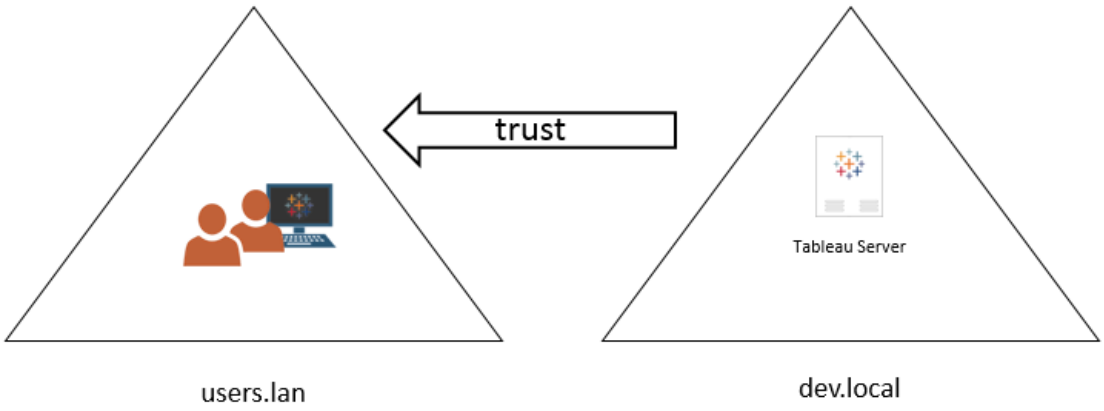
Therefore, you might need to update the domain nickname on Tableau Server before users can sign in using the nickname. If you do not update the nickname, users will have to sign in using a fully qualified domain name. For more information, see Users From New Domain Unable to Log In and Do Not Appear in User List in the Tableau Knowledge Base.

Domain Trust Requirements for Active Directory Deployments

When you run Tableau Server in an Active Directory environment across multiple domains (either in the same Active Directory forest or in different forests), some Tableau functionality is dependent on the trust relationship between the domains. For example, some administrators manage users in domains that are separate from where they deploy server applications, such as Tableau Server. In other organizations, a Tableau Server deployment might be shared with external partners or with different partners in the organization. Finally, Windows-authenticated data sources, such as SQL Server, MSAS, or Oracle, that Tableau Server connects to may also be in other domains.

If it's feasible, we recommend configuring two-way trust between all domains that interact with Tableau Server. If this is not possible, Tableau Server can be configured to support user authentication where a one-way trust has been configured. In this case, a one-way trust between domains is supported when the domain in which Tableau Server is installed is configured to trust the domain where user accounts reside.

The following illustration shows one-way trust between the domain where Tableau Server is installed and the domain where user accounts reside:



In this scenario, Tableau Server is in the dev.local domain, and users from the users.lan Active Directory domain are imported into Tableau Server. A one-way trust is required for this scenario; specifically, the dev.local domain is configured to trust the users.lan domain. Users

in the users.lan domain can access Tableau Server in the dev.local with their normal Active Directory credentials. However, you may need to update the domain nickname on Tableau Server before users log on with the nickname. Refer to the Tableau Knowledge Base for more information.

When you configure Tableau Server for this scenario, specify the primary user domain during installation. See Configure Initial Node Settings. To ensure that Tableau Server can connect to other Active Directory domains, you must also specify other domains that Tableau Server connects to by setting the `wgserver.domain.accept_list` option with TSM. For more information, see wgserver.domain.accept_list.

Duplicate bind accounts for domain trust

Tableau Server on Linux relies on JDK's LDAP implementation that uses a simple bind to authenticate with Active Directory. Simple bind is not domain-aware and as a result, does not support a cross-domain bind. When you set up the initial identity store, you must supply the bind account that you will use to authenticate to Active Directory.

To enable cross-domain trust and directory look-ups, you must duplicate this bind account in every target domain. Each bind account in each domain must use the same username (`sAMAccountName` or `dn`) and password.

Kerberos single sign-on is supported in this one-way trust scenario.

Review User Management in Deployments with External Identity Stores to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

Connecting to live data in one-way trust scenarios

In the one-way trust scenario, users connecting to Tableau Server can connect to live data that's hosted in the cloud or on any other data source on premises that does not rely on Windows authentication.

Data sources that require Windows-authentication might have additional authentication requirements that complicate the scenario, or that can even prevent Tableau Server users

from connecting. This is because Tableau Server uses a service account for authentication with such data sources. If you are running Tableau Server in a different domain than data sources that use Windows authentication, verify that the service account that is used for Tableau Server can access the data source.

Communicating with the Internet

In most enterprises, Tableau Server needs to communicate with the internet. Tableau Server was designed to operate inside a protected internal network. Do not set up Tableau Server directly on the internet or in a DMZ. Instead, communications between your network and the internet should be mediated using proxy servers. If the computer running Tableau Server cannot access the internet directly, then you may need to deploy forward proxy servers to mediate traffic from inside the network to targets on the internet. Tableau Server doesn't support pass-through or manual proxy authentication.

For inbound traffic, we recommend running Tableau Server behind reverse proxy servers.

How Tableau communicates with the internet

Tableau Server requires outbound access to the internet for these scenarios:

- Working with maps. Tableau uses map data that is hosted externally.

  Tableau Server needs to connect to the following internet locations with port 443 to use maps:

  - mapsconfig.tableau.com
  - api.mapbox.com

  If Tableau cannot make these connections, maps may fail to load.

  You can test connectivity by accessing each of those addresses in a browser: `https://mapsconfig.tableau.com/v1/config.json` and `https://api.mapbox.com/` will prompt you to download a json file.

If you use a proxy to connect to the internet and are unable to connect to `api.mapbox.com`, see Working with firewalls on the Mapbox website.

For Tableau Server version 2019.1 and earlier, see the documentation for your version: Tableau Help

- Connecting to the Tableau send-logs server.

  You can upload log files to Tableau when working with Support. See `tsm maintenance send-logs`. To successfully upload files to Tableau, your Tableau Server must be able to communicate with the send-logs server on port 443:

  - report-issue.tableau.com:443
  - crash-artifacts-747369.s3.amazonaws.com
  - s3-us-west-2-w.amazonaws.com
  - s3-w-a.us-west-2.amazonaws.com

- Sending Basic Product Data.

  The domain, prod.telemetry.tableausoftware.com, is used by Tableau to receive the Basic Product Data about process launch and shutdown. It is also used for the more general Product Usage Data.

  Traffic to this domain will occur on port 80 (for initial registration of our Product Data clients) and on port 443 (for all subsequent traffic).

  prod.telemetry.tableausoftware.com:80

  prod.telemetry.tableausoftware.com:443

- Licensing. Tableau products connect to the internet to activate product keys. Unless you activate Tableau software with the Offline Activation Tool, all Tableau products must have access to the internet to validate licenses. Specifically Tableau requires internet

access during the following licensing operations: activation, deactivation, and on the refresh maintenance date. For more information about these operations, see Manage Licenses.

Tableau Server needs to connect to the following internet locations when activating product keys, registering the product, and signing in to Tableau Cloud.

- atr.licensing.tableau.com:443
- licensing.tableau.com:443
- register.tableau.com:443
- o.ss2.us
- s.ss2.us
- crt.rootca1.amazontrust.com
- crt.sca1b.amazontrust.com
- crt.sca0a.amazontrust.com
- crt.sca1a.amazontrust.com
- crt.sca2a.amazontrust.com
- crt.sca3a.amazontrust.com
- crt.sca4a.amazontrust.com
- *.digicert.com
- ocsp.*.amazontrust.com
- crl.*.amazontrust.com
- crt.rootg2.amazontrust.com

Requests to the above domains may be on port 80 or 443. Port 80 is used for certificate validation (revocation, certificate chain, etc). Port 443 is used for SSL connections.

Requests to the `ocsp.*.amazontrust.com` and `crl.*.amazontrust.com` domains are managed by Amazon for certificate revocation information. See ACM certificate characteristics for more information. We recommend that you install the Amazon root certificates in the certificate trust store on the computer running Tableau. To download and install the Amazon root certificates, see Certificate Authorities on the Amazon Trust Services web site.

If Tableau Server cannot make a connection while attempting to activate its license, you will be prompted to do an offline activation.

To diagnose connectivity to Tableau's licensing server, paste the following URL into a browser or at a curl command prompt on the Tableau Server computer:

```
https://atr.licensing.tableau.com/_status/healthz
```

If Tableau Server is able to access the licensing server, it displays an "OK" message. Otherwise, an error such as "Can't reach this page" may be displayed. To resolve this issue, work with your networking team to unblock access to atr.licensing.tableau.com:443 on the Tableau Server computer.

- Working with external or cloud-based data.

  Tableau Server needs to connect to the following internet location for Anaplan, Box, Dropbox, Google Drive, Google Sheets, OneDrive, and Snowflake services:

  ```
  galop.connectors.tableau.com:443
  ```

- Working with Tableau dashboard extensions.

  Tableau Server needs to connect to the following internet location to use Sandboxed dashboard extensions:

  ```
  extensions.tableauusercontent.com: 443
  ```

For more information, see Manage Dashboard and Viz Extensions in Tableau Server.

- Working with Slack.

  If you are integrating Tableau with a Slack workspace, there are a number of steps you need to take, including adding specific URLs to the Tableau allowlist. These are listed here. For complete details on how to do this, see Integrate Tableau with a Slack Workspace.

Tableau Server can run without internet access. For more information about deploying Tableau Server in organizations without access to the internet, see Install Tableau Server in a Disconnected (Air-Gapped) Environment.

In many enterprises, users also need to access Tableau Server from outside the network (that is, from the internet). For example, in many enterprises, users want to be able to reach Tableau Server from their mobile devices in order to interact with views that are stored on the server. To configure access to Tableau Server from the internet or from mobile devices, you should use a reverse proxy. See Configuring Proxies and Load Balancers for Tableau Server.

As a security best practice, do not expose the TSM port (by default, `8850`) to the internet.

Configuring Proxies and Load Balancers for Tableau Server

In most enterprises, Tableau Server needs to communicate with the internet. Tableau Server was designed to operate inside a protected internal network. Do not set up Tableau Server directly on the internet or in a DMZ. Instead, communications between your network and the internet should be mediated using proxy servers. Forward proxy servers mediate traffic from inside the network to targets on the internet. Reverse proxy servers and load balancers mediate traffic from the internet to targets inside the network.

Who should read this article?

This article is for IT professionals who are experienced with general networking, load balancing, and gateway proxy solutions. The article describes how and when Tableau requires

internet access, and describes how to configure your network and Tableau to use proxy servers and load balancers for access to and from the internet. There are many third-party solutions available, so some of the content in the article is necessarily generic.

Before you configure a proxy server, see Communicating with the Internet.

Configure a forward proxy server

To enable communication from Tableau Server to the internet, deploy Tableau Server behind a forward proxy server. When Tableau Server needs access to the internet, it doesn't send the request directly to the internet. Instead, it sends the request to the forward proxy, which in turn forwards the request. Forward proxies help administrators manage traffic out to the internet for tasks such as load balancing, blocking access to sites, etc.

If you use a forward proxy, you must configure the computers that run Tableau Server inside the network to send traffic to the forward proxy. Tableau Server doesn't support pass-through or manual proxy authentication.

If you are running OpenID authentication with a forward proxy solution, additional configurations are required. See Configure Tableau Server for OpenID Connect.

Configuring Tableau Server on Linux to work with a forward proxy

We recommend configuring Tableau Server to work with your forward proxy solution as part of the installation process. Specifically, configure Tableau Server when you run `./initialize-tsm` as described in Install and Initialize TSM, or as part of Automated Installation of Tableau Server.

The procedure below describes how to create a forward proxy configuration file for Tableau Server on Linux.

The configuration file is stored in the following directory:

```
~<unprivileged_user>/.config/systemd/tableau_server.conf.d
```

By default, Tableau Server creates the unprivileged user, `tableau`. Therefore, the default path to the configuration directory is:

```
~tableau/.config/systemd/tableau_server.conf.d
```

The proxy configuration file name in this topic and in the configuration file below is referred to as `20-proxy.conf`. You can name this file according to your own convention, but it must use the .conf extension. systemd will process files stored in the `tableau_server.conf.d` directory in lexical order according to file name.

1. Run the `tsm stop` command.
2. Start a session as the unprivileged user. By default, `tableau`, is the unprivileged user created by Tableau Server during installation. Run the following command:

   ```
   sudo su -l tableau
   ```

3. Create or open the `20-proxy.conf` file in the `tableau_server.conf.d` directory. If you configured forward proxy during setup, then the `20-proxy.conf` file has already been created.

   - Create the file. Run the following command:

     ```
     touch ~tableau/.config/systemd/tableau_server.conf.d/20-proxy.conf
     ```

   - Open the `20-proxy.conf` file in a text editor.

4. Copy the *Proxy configuration file contents* into the file. If you are editing an existing file, take care not to delete the configuration. The *Proxy configuration file contents* include instructions on how to set forward proxy configurations. After you have edited and saved the file go to Step 5.

Proxy configuration file contents

```
# Always edit this file on Tableau Server as the unprivileged
user. By default, tableau, is the unprivileged user created by
Tableau Server during installation.
# Set environment variables http_proxy and https_proxy to point
to your proxy host.
# For example, to set the proxy to example-host for ports 80
and 443, run the following commands:
#
http_proxy=http://example-host:80/
https_proxy=http://example-host:443/
#
# Take care to use 'http' when you specify the URL for the
https_proxy environmental variable.
# Do not specify the 'https' protocol for the value of  the
https_proxy environmental variable.
#
# To bypass the proxy server, specify exceptions in the no_
proxy variable. Use this variable if your proxy server does not
route internal addresses.
# You must also add exceptions to this proxy configuration to
guarantee that all communications within a local Tableau Server
cluster (if you have one now or will have one later) do not
route to the proxy server.
# Enter both the host name and the IP address for each com-
puter. Additionally, include the cannonical host name (loc-
alhost) and IP address (127.0.0.1) for the local computer.
# For example, to specify exceptions for a three-node cluster:
#
no_proxy-
="lo-
ocalhost,127.0.0.1,hostname1,hostname2,hostname3,IP1,IP2,IP3"
#
# Where "hostname1" is the actual hostname of node 1, and "IP1"
is the actual IP address of node 1, etc.
```

5. Exit the Tableau shell. Run the following command:

```
exit
```

6. Restart the TSM business services. Run the following script commands:

```
sudo /opt/tableau/tableau_server-
/packages/scripts.<version>/stop-administrative-services
```

```
sudo /opt/tableau/tableau_server-
/packages/scripts.<version>/start-administrative-services
```

7. Restart TSM.

```
tsm restart
```

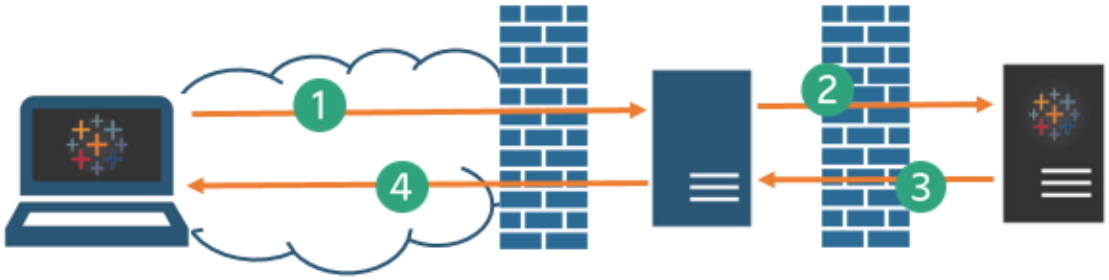
Server crash reporter

If your organization uses a proxy server to connect to the internet then you must configure Tableau Server's crash reporter to use the proxy. Even if you have already configured Tableau Server to use a proxy, you must also configure server crash reporter separately. To configure proxy for server crash reporter, see Configure Server Crash Reporter.

How a reverse proxy and a load balancer works with Tableau Server

Reverse proxies and load balancers are servers that receive requests from external (internet) clients and forwards them to Tableau Server. These solutions make Tableau Server available to the internet without having to expose the individual IP address of that particular Tableau Server to the internet. They may also act as authentication and pass-through devices, so that no data is stored where people outside the company can get to it. This requirement can be important for organizations that are subject to various privacy regulations such as PCI, HIPAA, or SOX.

The following diagram illustrates the communication path when a client makes a request to Tableau Server that is configured to work with a reverse proxy and/or load balancer (LB).



1. An external client initiates a connection to Tableau Server. The client uses the public URL that's been configured for the reverse proxy server/LB, such as `https://tableau.example.com`. (The client doesn't know that it's accessing a reverse proxy/LB.)
2. The reverse proxy maps that request in turn to a request to Tableau Server. In some scenarios, the reverse proxy may be configured to authenticate the client (using SSL/TLS) as a precondition to passing the request to Tableau Server.
3. Tableau Server gets the request and sends its response to the reverse proxy/LB.
4. The reverse proxy/LB sends the content back to the client. As far as the client is concerned, it just had an interaction with Tableau Server, and has no way to know that the communication passed through intermediary server(s).

TLS/SSL

Depending on your gateway scenario, you should consider configuring your reverse proxy and load balancing servers to use TLS/SSL for any traffic that's external to your network. This helps to ensure privacy, content integrity, and authentication. Unless you've deployed other security measures to protect traffic between your internet gateway and Tableau Server, we also recommend configuring SSL between the gateway proxy and Tableau Server. You can use internal or self-signed certificates to encrypt traffic between Tableau Servers and other internal computers.

Mobile access

Tableau Server adds an X-header to all HTTP responses for Tableau Mobile sessions. By default, most proxy solutions will preserve X-headers. If your gateway solution does not preserve X-headers, then you will need to configure your proxy server and load balancer to preserve the following header to all HTTP responses for Mobile client sessions: `X-Tableau: Tableau Server`.

If you have configured authentication at the gateway, then your proxy server/LB must respond to Tableau Mobile HTTP requests with a HTTP 302 response. The 302 must include a redirect to the identity provider login page. To view a diagram that describes the 302 authentication sequence, see Tableau Mobile Authentication Sequence in the Tableau Community.

Reverse proxy, load balancer and user authentication

Tableau Server will always authenticate users. This means that even if you are authenticating inbound connections at the gateway for your organization, Tableau Server will still authenticate the user.

However, not all clients will support user authentication with with a gateway solution:

- For supported web browsers, you can use SAML, OpenID Connect, Kerberos, Trusted Tickets or manual authentication with a reverse proxy/LB.
- Tableau Mobile supports SAML or manual authentication with a reverse proxy/LB. The iOS version of Tableau Mobile additionally supports Kerberos with a reverse proxy/LB. The same recommendation above applies.
- Tableau Prep does not support authentication with a reverse proxy or load balancer. For remote access, use a VPN solution or configure your gateway services to route traffic from Tableau Prep directly to Tableau Server for authentication.
- Tableau Desktop supports authentication with a reverse proxy provided that an authentication module is performing preauthentication on the reverse proxy before traffic is routed to Tableau Server for final authentication. For more information, see Part 5 -

Configuring Web Tier of the *Tableau Server Enterprise Deployment Guide* and Configure Authentication Module with Independent Gateway.

If your organization is authenticating with Active Directory:

- Tableau Server must be configured for reverse proxy before configuring Tableau Server for Kerberos. For more information, see Configure Kerberos.

Configure Tableau Server to work with a reverse proxy server and/or load balancer

Before you configure Tableau Server, you'll need to collect the following information about the proxy server configuration. To configure Tableau Server, you use the `tsm configuration set` command. The information you need to collect corresponds to options you'll need when you run `tsm`.

Most of the following tsm options are also used to configure Tableau Server deployments that operate behind a load balancer. For more information, see Add a Load Balancer.

Item	Description	Corresponding `tsm configuration set` option
IP address or `CNAME`	You can either enter an IP address or a CNAME for this option. The public IP address or addresses of the proxy and load balancer servers. The IP address must be in IPv4 format, such as `203.0.113.0`, and it must be a static IP. If you are unable to provide a static IP, or if you are using cloud proxies or external load balancers, you can specify the CNAME (Canonical Name) DNS value that clients will use to con-	`gateway.trusted`

Item	Description	Corresponding `tsm configuration set` option
	nect to Tableau Server. This CNAME value must be configured on your reverse proxy solution to communicate with Tableau Server.	
FQDN	The fully qualified domain name that people use to reach Tableau Server, such as `tableau.example.com`. Tableau Server doesn't support context switching for this option. For example, the following URL is not supported: `example.com/tableau`.	`gateway.public.host`
Non-FQDN	Any subdomain names for the proxy or LB servers. In the example of `tableau.example.com`, the subdomain name is `tableau`.	`gateway.trusted_hosts`
Aliases	Any public alternative names for the proxy or LB servers. In most cases, aliases are designated using CNAME values. An example would be a proxy server `bigbox.example.com` and CNAME entries of `ftp.example.com` and `www.example.com`.	`gateway.trusted_hosts`
Ports	Port numbers for traffic from the client to the reverse proxy server.	`gateway.public.port`

If you are using a distributed installation of Tableau Server, then run the following `tsm` commands on the initial node in your cluster.

1. Enter the following command to set the FQDN that clients will use to reach Tableau Server through the proxy and/or LB servers, where *name* is the FQDN:

   ```
   tsm configuration set -k gateway.public.host -v 'name'
   ```

   For example, if Tableau Server is reached by entering `https://tableau.example.com` in the browser, enter this command:

   ```
   tsm configuration set -k gateway.public.host -v 'tableau.example.com'
   ```

2. Enter the following command to set the address or the CNAME of the proxy and or LB servers, where *server_address* is the IPv4 address or CNAME value:

   ```
   tsm configuration set -k gateway.trusted -v 'server_ip_address'
   ```

   If your organization uses multiple proxy servers and/or LB servers, enter multiple IPv4 addresses, separating them with commas. IP ranges are not supported. To improve start up and initialization of Tableau Server, minimize the number of entries for `gateway.trusted`.

3. Enter the following command to specify alternate names for the proxy/LB servers, such as their fully qualified domain names, any not fully qualified domain names, and any aliases. If there's more than one name, separate the names with a comma.

   ```
   tsm configuration set -k gateway.trusted_hosts -v 'name1, name2, name3'
   ```

   For example:

   ```
   tsm configuration set -k gateway.trusted_hosts -v 'proxy1.example.com, proxy1, ftp.example.com, www.example.com'
   ```

4. If the proxy server is using SSL to communicate with the internet, run the following command, which tells Tableau that the reverse proxy server is using port 443 instead of port

80:

```
tsm configuration set -k gateway.public.port -v 443
```

> **Note**: If the proxy server is using SSL to communicate with Tableau Server, SSL must be configured and enabled on Tableau Server.

5. Enter the following command to commit the configuration change:

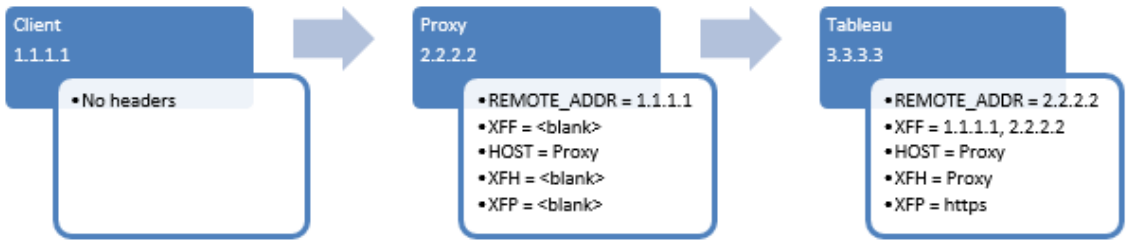
   ```
   tsm pending-changes apply
   ```

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

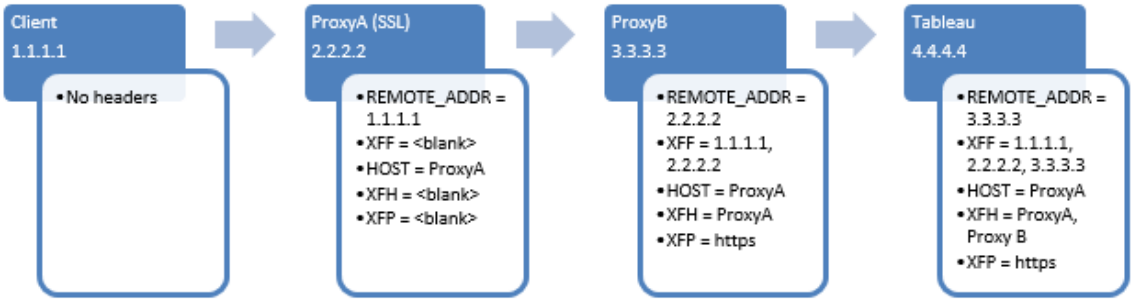
Configure the reverse proxy or load balancing server to work with Tableau Server

When a client accesses Tableau Server through a reverse proxy or load balancer, specific message headers have to be preserved (or added). Specifically, all servers in the message chain must be represented in the `gateway.trusted` and `gateway.trusted_hosts` settings.

The following graphic shows example headers for a single-hop message chain, where the proxy server is communicating directly with Tableau Server:



The following graphic shows example headers for a multiple-hop message chain, where the message traverses two proxy servers before connecting to Tableau Server:



The following table describes what these headers are and how they relate to the configuration settings on Tableau Server:

Headers	Description	Related Tableau Server settings
`REMOTE_ADDR` and `X-FORWARDED-FOR` (`XFF`)	Tableau Server needs these headers to determine the IP address of origin for requests. `X-FORWARDED-FOR` header must present IP address chain to Tableau Server in the order the connections have occurred.	The IP address that you set in`gateway.trusted` must match the IP presented in `REMOTE_ADDR`. if you sent multiple addresses in`gateway.trusted`, one of them must match the IP presented in `REMOTE_ADDR`.
`HOST` and `X-FORWARDED HOST` (`XFH`)	These headers are used to generate absolute links to Tableau Server when it replies to the client. `X-FORWARDED-HOST` header must present host names to Tableau Server in the order the connections have occurred.	The host names that are presented in `X-FORWARDED-HOST` header must be included in the host names that you specify in `gateway.trusted_hosts`.
`X-FORWARDED-PROTO` (`XFP`)	This header is required if SSL is enabled for traffic from the client to	Port configuration on

	the proxy, but not for traffic from the proxy to Tableau Server. The `X-FORWARDED-PROTO` headers are important for scenarios where HTTP or HTTPS is not maintained along each hop of the message route. For example, if the reverse proxy requires SSL for outside requests, but traffic between the reverse proxy and Tableau Server is not configured to use SSL, `X-FORWARDED-PROTO` headers are required. Some proxy solutions add the `X-FORWARDED-PROTO` headers automatically, while others do not. Finally, depending on your proxy solution, you might have to configure port forwarding to translate the request from port 443 to port 80. Related KB article: "Unable to Sign In" and "Invalid username or password" Error With SAML After Upgrading.	reverse proxy (inbound connections from client and outbound connections to Tableau Server) must be specified in the corresponding parameter: `gateway.public.port`, which is the port clients use to connect to the proxy. If the proxy server is using SSL to communicate with Tableau Server, SSL must be configured and enabled on Tableau Server.

Validate reverse proxy and load balancer configuration

To validate your gateway-to-Tableau Server configuration, publish workbooks and datasources using Tableau Server web authoring or Tableau Desktop. If you are connecting with a web browser to Tableau Server from the internet, verify that you are using a recommended browser. Publish and view workbooks that use existing datasources as well as a datasource

that you've published . Use the links below to familiarize yourself with connecting with Tableau Server as an end-user.

Task	Documentation
Overview of web authoring.	Use Tableau on the Web
Log in to Tableau Server from Tableau Desktop or a web browser.	Sign in to Tableau Server or Online
Publish a workbook to Tableau Server.	Publish a Workbook
Publish a data source.	Publish a Data Source
Open workbook from Tableau Server.	Opening Workbooks from the Server
Log out Server (with Desktop).	Sign in to Tableau Server or Online
Download workbook from a web browser.	Download Workbooks
Check to make sure tabcmd (from a non-server client) works.	tabcmd

Related topics

- Tableau Desktop Internet Access Requirements
- Add a Load Balancer

Deploy

The topics in this section provide information on installing, configuring, and upgrading Tableau Server on Linux.

> Looking for Tableau Server on Windows? See Install and Configure Tableau Server

Validating your server deployment plan

Before you commit to installing a new Tableau Server deployment in your organization, be sure to carefully evaluate your options. For most organizations, Tableau Cloud will provide a more reliable, performant, and cost-effective analytics solution when compared to self-hosting Tableau Server. For information about the viability of Tableau Cloud for your organization, review this blog post, Should I move my analytics to the cloud?

Already running Tableau Server and want to migrate to Tableau Cloud? See Tableau Cloud Manual Migration Guide.

If you have determined that you must self-host Tableau Server, we recommend following the prescribed Tableau deployment in the Enterprise Deployment Guide (EDG). The EDG presents a fully-tested and supported, performant, scalable, secure reference architecture based on a tiered data network. Moving forward, we are committed to investing in the EDG reference architecture to ease feature rollouts and improve upgrade scenarios.

Install and Configure Tableau Server

The topics referenced at the bottom of this page describe the steps to install and configure Tableau Server. If you are installing a distributed deployment (cluster), use the steps in this

topic to install the initial node, then, to install additional nodes, see Distributed and High Availability Tableau Server Installations.

After you run the installation, you must then continue setup by activating a license, registering Tableau Server, and configuring various settings including authentication.

Other installation methods

There are a few alternative methods that you can use to install Tableau Server.

- If you want a quick start procedure to install Tableau Server in a non-production environment, see Jump-start Installation.
- For an end-to-end procedure that describes how to deploy an enterprise-ready, four-node, reference architecture in a tiered data center, see Tableau Server Enterprise Deployment Guide.
- Automated Installation of Tableau Server.
- If you are installing Tableau Server in an environment without a connection to the internet, see Install Tableau Server in a Disconnected (Air-Gapped) Environment.
- You can also install Tableau Server onto various cloud platforms. See Self-Host Tableau Server in a Public Cloud Service.

Validating your server deployment plan

Before you commit to installing a new Tableau Server deployment in your organization, be sure to carefully evaluate your options. For most organizations, Tableau Cloud will provide a more reliable, performant, and cost-effective analytics solution when compared to self-hosting Tableau Server. For information about the viability of Tableau Cloud for your organization, review this blog post, Should I move my analytics to the cloud?

Already running Tableau Server and want to migrate to Tableau Cloud? See Tableau Cloud Manual Migration Guide.

If you have determined that you must self-host Tableau Server, we recommend following the prescribed Tableau deployment in the Enterprise Deployment Guide (EDG). The EDG presents a fully-tested and supported, performant, scalable, secure reference architecture based on a tiered data network. Moving forward, we are committed to investing in the EDG reference architecture to ease feature rollouts and improve upgrade scenarios.

Before you begin

To install Tableau Server you must have a computer that satisfies the hardware requirements. You will get an informational message if your computer meets the minimum requirements but does not satisfy the recommended minimum requirements. In this case, your computer hardware can handle a trial installation of Tableau but is not adequate for a production environment. For more information, see Before you install....

Installation steps

The following steps describe how to install Tableau Server on a single computer. Use the steps to install Tableau Server in a single server deployment. Use the steps to install the initial node in a multi-node Tableau Server deployment. Run the steps sequentially.

1. Install and Initialize TSM
2. Activate and Register Tableau Server
3. Configure Initial Node Settings
4. Add an Administrator Account
5. Validate Installation

Before you install...

Note: You can find additional information about technical specifications for Tableau Server on the Tableau web site, here.

This topic includes requirements and recommendations that you must consider before you install Tableau Server into a production environment.

- If you are new to Tableau Server, and you want to deploy it in your organization, we encourage you to deploy Tableau Server as a single server in a test environment first. The easiest way to do a single-server installation is to follow the steps in Jump-start Installation.
- For an end-to-end procedure that describes how to deploy an enterprise-ready, four-node, reference architecture in a tiered data center, see Tableau Server Enterprise Deployment Guide.
- If you are deploying Tableau Server in a distributed cluster, review Distributed Requirements in addition to the requirements and recommendations described in this topic.
- If you are migrating from Tableau Server on Windows to Tableau Server on Linux, see Migrate Tableau Server from Windows to Linux.

Validating your server deployment plan

Before you commit to installing a new Tableau Server deployment in your organization, be sure to carefully evaluate your options. For most organizations, Tableau Cloud will provide a more reliable, performant, and cost-effective analytics solution when compared to self-hosting Tableau Server. For information about the viability of Tableau Cloud for your organization, review this blog post, Should I move my analytics to the cloud?

Already running Tableau Server and want to migrate to Tableau Cloud? See Tableau Cloud Manual Migration Guide.

If you have determined that you must self-host Tableau Server, we recommend following the prescribed Tableau deployment in the Enterprise Deployment Guide (EDG). The EDG presents a fully-tested and supported, performant, scalable, secure reference architecture

based on a tiered data network. Moving forward, we are committed to investing in the EDG reference architecture to ease feature rollouts and improve upgrade scenarios.

Hardware recommendations for production installations

The following list describes the minimum hardware recommendations for a production use, single- node installation of Tableau Server:

Important: These recommendations are minimums and may not reflect the requirements for your installation and organization. For example, there are a number of factors that can impact disk space requirements, including whether or not you will be publishing extracts, flows, and the number of workbooks to Tableau Server. For more information on what might impact free disk space requirements, see Disk Space Requirements.

Install Type	***Processor***	***CPU***	***RAM***	***Free Disk Space***
Single node	• 64-bit (x86_64 chipsets) • Must support SSE4.2 and POPCNT instruction sets • ARM-based processors are not supported	8 cores (16 vCPUs), 2.0 GHz or higher	Version 2022.3 and later: • 128 GB Version 2021.4.0 to version 2022.1.x: • 64 GB Version 2021.3.x and earlier: • 32 GB	50 GB
	If you are adding Tableau Prep Conductor to your Tableau Server installation, we recommend you add a second node and dedicate this to running			

Install Type	*Processor*	*CPU*	*RAM*	*Free Disk Space*
	Tableau Server Prep Conductor. This node should have a minimum of 4 cores (8 vCPUs), and 16 GB of RAM.			
Multi-node and enterprise deployments	Contact Tableau for technical guidance. Nodes must meet or exceed the minimum hardware recommendations, except: • Dedicated Backgrounder nodes running up to two instances of backgrounder, where 4 cores may be acceptable. • Dedicated node for Tableau Prep Conductor: Minimum of 4 cores (8 vCPUs), and 16 GB of RAM. • Dedicated node for Independent Gateway: Minimum of 2 cores (4 vCPUs), 8 GB of RAM, and 100 GB free disk space.			

Important: The disk space requirement cannot be checked until you initialize TSM. If you don't have enough space, you won't be told this until after you install the Tableau Server package.

50 GB disk space available, with a minimum of 15 GB allocated to the `/opt` directory, and the remainder allocated to the `/var` directory for data storage.

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space. You may need to allocate additional disk space depending on various factors like whether you will be using extracts.

  The core Tableau Server bits must be installed in a directory with at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to con-

tinue with setup. By default the install location is the `/opt` directory. You can change the installation path for Tableau Server on RHEL distros.

If you plan to make heavy use of extracts then you may need to allocate additional disk space.You can specify a different directory for data (extract) storage during installation.

- **Network attached storage space requirements for External File Store:** If you are planning to configure Tableau Server with External File Store, you will need to estimate the amount of storage space to dedicate on your network attached storage.

  Estimating the storage size: You must take into account the amount of storage needed for publishing and refreshing extracts. In addition, you must also take into account the repository backup size unless you specifically choose the option to do your repository backup separately as described in the Option 2: Back up repository separately topic.

  - Extracts:
    - Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out how many extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.
    - Consider the space needed by the temp directory during an extract refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.
  - Repository Backup:
    - To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base` directory.
    - To obtain the exact size of the repository data, open the backup file and use the size of the `workgroup.pg_dump` file.

- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the

purposes of counting cores.

- RAM shown is the minimum recommended for a single-node installation. Your installation may function better with more RAM, depending on activity, number of users, and background jobs, for example.

To see the full list of recommendations and to see the minimum requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server. For hardware specifications Tableau uses internally for testing scalability, see Hardware recommendations for production installations.

For public cloud deployments on Amazon Web Services and Google Cloud Platform, their "vCPU" is actually a CPU hyper-thread, and not a full CPU core. When sizing cloud instances, you will need twice as many vCPU as the Tableau Server CPU core requirements given (8 vCPU required for a minimum trial installation, 16 vCPU recommended for a single-node installation).

Operating system requirements

The following distributions of Linux are supported:

	2021.-4.x	**2022.1-.0 - 2022.1-.11**	**2022.1.-12+**	**2022.-3.0 - 2022.-3.3**	**2022.3-.4+**	**2023.-1.0 - 2023.-1.7**	**2023.1-.8+**	**2023.-3.0**	**2023.-3.1 - 2024.-2.x**
AlmaLinux 8.x									✓
AlmaLinux 9.x									✓
Amazon Linux 2	✓	✓	✓	✓	✓	✓	✓	✓	✓
Amazon Linux									✓

	2021.-4.x	2022.1-.0 - 2022.1-.11	2022.1.-12+	2022.-3.0 - 2022.-3.3	2022.3-.4+	2023.-1.0 - 2023.-1.7	2023.1-.8+	2023.-3.0	2023.-3.1 - 2024.-2.x
2023									
CentOS 7.9+ (not 8.x)	✓	✓	✓	✓	✓	✓	✓	✓	✓
CentOS Stream 8.x									✓
CentOS Stream 9.x									✓
Debian 9	**Note:** As of July 2022, Debian distributions are no longer supported. For more information, see this Tableau Community post.								
RHEL 7.3+	✓	✓	✓	✓	✓	✓	✓	✓	✓
RHEL 8.3+	✓	✓	✓	✓	✓	✓	✓	✓	✓
RHEL 9.x								✓	✓
Oracle Linux 7.3+ (not 8.x)	✓	✓	✓	✓	✓	✓	✓	✓	✓

	2021.-4.x	2022.1-.0 - 2022.1-.11	2022.1.-12+	2022.-3.0 - 2022.-3.3	2022.3-.4+	2023.-1.0 - 2023.-1.7	2023.1-.8+	2023.-3.0	2023.-3.1 - 2024.-2.x
Oracle Linux 8.x									✓
Oracle Linux 9.x									✓
Rocky Linux 8.x									✓
Rocky Linux 9.x									✓
Ubuntu 16.04 LTS	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ubuntu 18.04 LTS	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ubuntu 20.04 LTS			✓		✓	✓	✓	✓	✓
Ubuntu 22.04 LTS							✓		✓

For information about Linux distribution requirements for Tableau Server in a Container, see Supported distributions for building .

Important: To ensure you have the latest security and functionality patches, Tableau strongly recommends that you use the latest supported version of the Linux distribution you are deploying on. Tableau generally tests and validates on the latest minor version of a supported distribution major version.

Additional notes on Linux distributions:

- Red Hat Enterprise Linux (RHEL), CentOS, Oracle Linux, and Amazon Linux distributions are collectively referred to in this documentation as RHEL-like.
- As of July 2022, Debian distributions are no longer supported. For more information, see this Tableau Community post.
- Non-LTS releases of Ubuntu are not supported.
- Ubuntu version 17.04 is not supported.
- Ubuntu version 20.04 support was added in Tableau Server version 2023.1.0, and in Server maintenance releases 2022.1.12 and 2022.3.4. It is not supported in earlier versions.
    - Installing Tableau Server 2023.1 and later on a physical machine running Ubuntu Linux 20.04 results in an installation error. For more information, see the Error "One or more control plane service(s) are in a non-active state" knowledge article.
- Previous versions of CentOS and Ubuntu are not supported because Tableau Server requires `systemd` for process management.
- The version of the installer with the file suffix, `.deb`, installs on both Ubuntu and Debian distributions.
- Custom kernels are not supported.

In a multi-node installation of Tableau Server, all of the computer nodes where you are installing Tableau must run Linux and the same distribution of Linux.

Installation directory

The core Tableau Server bits are installed in the `/opt` directory by default.

- The directory where you install Tableau Server must have at least 15 GB of free disk space allocated to it. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup.
- You can specify a non-default install location on RHEL-like systems, but cannot change the location on Ubuntu.
- Do not specify a symbolic link or a directory location on a Network File System (NFS) volume when specifying a non-default install location on RHEL-like systems.

Data directory

By default, Tableau Server will create a data directory for all content and extracts that are managed by Tableau. The directory is created at `/var/opt/tableau/tableau_server`.

You can specify a different directory for data (extract) storage during installation. If you plan to use a different directory, do not create the directory. Instead, let Tableau Server setup create the directory. The data directory requires specific permissions that are set during the installation process.

To change the data directory, you must pass a flag along with the data directory path when you run the `initialize-tsm` script. See Help Output for initialize-tsm Script.

If you are changing the default data directory:

- Do not specify a symbolic link or a data directory location on a Network File System (NFS) volume.
- Do not specify a data directory location with a path that includes a period or space. If there is a period or space in the path, initialization will fail.
- The data directory must be installed into a different directory than the installation directory.

> **Important**: You cannot change the data directory location after you've run `initialize-tsm`. The data directory location will persist for the life of the deployment, including subsequent upgrades.

Tableau Prep Conductor

Tableau Prep Conductor is one of the process on Tableau Server. It runs a flow, checks connection credentials, and sends alerts if a flow fails. Tableau Prep Conductor leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output instead of logging into Tableau Prep Builder to manually run individual flows as your data changes.

Tableau Prep Conductor is licensed separately and is available through the Data Management license. For more information on Tableau Prep Conductor licensing, see License Data Management.

We recommend you enable Tableau Prep Conductor on a dedicated node. For more information:

- If you are installing a new Tableau Server, see Step 1 (New Install): Install Tableau Server with Tableau Prep Conductor.
- If you are adding Tableau Prep Conductor to an existing installation of Tableau Server, see Step 1 (Existing Install): Enable Tableau Prep Conductor.

Additional requirements

Make sure that your environment also meets the following additional requirements:

Hostname

- Tableau Server must be able to resolve the hosthame to an IP address either using the domain name server (DNS) or with a local host file on the computer running Tableau Server. By default, host files are stored at `/etc/hosts`.

- The hostname of the server must not change after you start Tableau Services Manager during the setup process. For example, this might happen if you use the cloud-init package to initialize a virtual machine, and you install Tableau Server on that virtual machine.
- Hostnames that include underscores (_) are not supported by Tableau Server.

Static IP address

The computer where you install Tableau Server must have a static IPv4 or IPv6 address.

Database drivers

To connect to specific data sources, the computer where you install Tableau Server must have the correct database drivers installed. For more information, see Database Drivers.

Available ports

TSM and Tableau Server each require an available TCP port in order for you to access them. TSM defaults to port 8850, and the Tableau Server Gateway service defaults to port 80. We strongly recommend that you ensure that both port 8850 and 80 are not in use on your system before installing Tableau Server. If those ports are not available, the TSM and gateway ports may be dynamically remapped to different port numbers, and there is currently no interface for displaying which port they have been remapped to.

See Tableau Services Manager Ports.

Local firewall configuration

If you are running a firewall on the computer where you will be installing Tableau Server, then you will need to open the following default ports for Tableau Server traffic. All port numbers, except 443 can be changed.

Port	TCP/UDP	Used by ...	TYPE OF INSTALLATION	
			All	Distributed / High Avail-ability
80	TCP	Gateway	X	
443	TCP	SSL. When Tableau Server is configured for SSL, the application server redirects requests to this port. Do not change this port.	X	
8850	TCP	Tableau Services Manager.	X	
8060	TCP	PostgreSQL database.	X	
8061	TCP	PostgreSQL backup verification port	X	
8000-9000	TCP	Range of ports reserved by default for dynamic mapping of Tableau processes		X
27000-27009	TCP	Range of ports used by Tableau Server for License service. This range must be open on the node running the License service and accessible from other nodes. By default, the initial node runs the License service.	X	

See Tableau Services Manager Ports and Configure Local Firewall.

System user and groups

Tableau Server on Linux uses one unprivileged user, and two groups for proper operation. Tableau will create the default account and groups during setup. Alternatively, you can specify existing accounts. See System user and groups
and TSM authorization group.

Sudo and root access

All installation tasks and administrative tasks for Tableau Server must be run as root. Often this is accomplished using the sudo command, but running the commands directly as the root user is also possible.

To install Tableau Server with the root account, you must specify a user account during installation. The account will be used for managing TSM. Specify the account by running the initialize-tsm script with the `-a` option. See Help Output for initialize-tsm Script.

Account password

The user account that you use to install and administer Tableau Server must be able to authenticate with a password. That is, the user must not use another means of authenticating (for example public key authentication).

If the account you are using to install and initialize Tableau Server does not have a password, you can set one using the `passwd` command:

```
sudo passwd $USER
```

Port access requirements

If you want to install Tableau Server remotely, for example by means of SSH, ensure that the following ports are open:

- `8850`. The port used for the Tableau Services Manager (TSM) web interface. You can use this interface to configure Tableau Server.
- `80`. The port used for the main Tableau Server web interface.

The Tableau Server installer attempts to open these ports during the installation process, but it can only open these ports for the `firewalld` firewall. If your computer runs another firewall, you must open the ports before you install.

Virtual Container environments

Beginning with version 2021.2, certain configurations of Tableau Server on Linux can be run in a container. For details on supported configurations, see Tableau Server in a Container.

Previous versions of Tableau Server on Linux and unsupported configutations have not been tested and are not supported in virtual container environments such as Docker. In these cases, Tableau Server on Linux will not function as expected if installed in these environments.

Package requirements

Systemd

Tableau Server requires `systemd`to manage services. This package is installed by default on CentOS 7 and Ubuntu 16. If you decide to test Tableau Server on a modified version of these distributions, you can run the following command to confirm that `systemd` is installed:

```
whereis systemd
```

If `systemd`is installed, the installation location is displayed. For example, you might see the following output:

```
systemd: /usr/lib/systemd /etc/systemd /usr/share/systemd /usr/share/man/man1/systemd.1.gz
```

If you have systemd installed but the Tableau installer is failing requirements checks for systemd, it's likely that systemd is not running. To verify that systemd is running, run the following command:

```
ls /run/systemd
```

The output will be a list of files and directories. If systemd is running, the output will include `system`. If `system` is not in the output, then systemd is not running.

Antivirus software

Antivirus software that scans directories used by Tableau Server can interfere with installation and ongoing use of Tableau Server. In some cases, this can result in installation failures, problems starting Tableau Server, or impacts to performance. If you plan to run antivirus software

on the computer running Tableau Server, follow the recommendations in the Knowledge Base.

Continue to the next step: Install and Configure Tableau Server.

Minimum Hardware Requirements and Recommendations for Tableau Server

The following minimum hardware requirements and recommendations apply to all computers running Tableau Server, including physical hardware and virtual machines (VMs):

- **Minimum requirements for installation** reflect the minimum hardware your computer must have in order to install Tableau Server. We do not recommend you attempt run Tableau Server on servers with these values, even if you are just testing. Depending the features you have licensed and are using, you may experience poor performance and an unrealistic experience. In certain cases Tableau Server may not start without at least 20GB of memory.

  For prototyping and Proof of Concept (PoC) testing, we recommend you use Tableau Cloud. This will give you an opportunity to experience Tableau Server on appropriately sized hardware.

- **Minimum recommendations for production** go beyond minimum installation requirements, and represent the minimum hardware configuration you should use for installation on most production nodes. If your computer meets the minimum installation requirements but does not meet these recommendations, the Setup program will warn you but you can continue the installation. For certain nodes dedicated to specific tasks and processes such as backgrounder, or Prep, you may be able to use servers that do not meet this minimum recommendation.

  The minimum recommendations listed here are intended as general guidance. However the recommendations for your environment may vary. For more information, see the Hardware recommendations section of the Recommended Baseline Configurations topic.

In addition, Tableau Server should not be installed on a physical computer or on a VM instance that is also running resource-intensive applications such as databases or application servers, or on a VM instance that is using shared resources.

Note: If you install Tableau Server on a computer that meets the minimum requirements but does not have at least 8 cores and 16 GB of system memory, the default number of all processes installed is reduced to one of each process by design. For more information about processes, see Server Process Limits

Looking for Tableau Server on Windows? See Minimum Hardware Requirements and Recommendations for Tableau Server.

Minimum installation hardware requirements

We strongly recommend any Proof of Concept (PoC) testing or prototyping be done using Tableau Cloud. This guarantees you will be running on systems with adequate resources.

The computer on which you are installing or upgrading Tableau Server must meet the minimum hardware installation requirements. If the Setup program determines that your computer does not meet the following requirements, you will not be able to install Tableau Server. Meeting these requirements does not guarantee you a good experience testing Tableau Server

Minimum Hardware Requirements

These minimum requirements are for installing Tableau Server. They do not guarantee successful testing or use. For production minimum recommendations, see Minimum production hardware recommendations.

Processor	***CPU***	***RAM***	***Free Disk Space***
• 64-bit	4 cores (8	Version 2022.3 and	15 GB

Processor	*CPU*	*RAM*	*Free Disk Space*
• Must support SSE4.2 and POPCNT instruction sets • ARM-based processors are not supported	vCPUs)	later: • 64 GB - initial node • 16 GB - additional nodes Version 2022.1 and earlier: • 16 GB - all nodes	

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space. You may need to allocate additional disk space depending on various factors like whether you will be using extracts.

  The core Tableau Server bits must be installed in a directory with at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup. By default the install location is the `/opt` directory. You can change the installation path for Tableau Server on RHEL distros.

  If you plan to make heavy use of extracts then you may need to allocate additional disk space. You can specify a different directory for data (extract) storage during installation.

- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the purposes of counting cores.

Minimum production hardware recommendations

For production use, the computers on which you install or upgrade Tableau Server should, in most cases, meet or exceed the minimum hardware recommendations. These recommendations are general. Actual system needs for Tableau Server installations can vary based on many factors, including number of users and the number and size of extracts, as well as the features you have licensed. If the Setup program determines that your computer does not meet the following recommendations, you will get a warning, but you can continue with the installation process. For more information, see the Hardware recommendations for production installations.

Install Type	***Processor***	***CPU***	***RAM***	***Free Disk Space***
Single node	• 64-bit (x86_64 chipsets) • Must support SSE4.2 and POPCNT instruction sets • ARM-based processors are not supported	8 cores (16 vCPUs), 2.0 GHz or higher	Version 2022.3 and later: • 128 GB Version 2021.4.0 to version 2022.1.x: • 64 GB Version 2021.3.x and earlier: • 32 GB	50 GB
	If you are adding Tableau Prep Conductor to your Tableau Server installation, we recommend you add a second node and dedicate this to running Tableau Server Prep Conductor. This node should have a minimum of 4 cores (8 vCPUs), and 16 GB of RAM.			
Multi-node and	Contact Tableau for technical guidance.			

Install Type	*Processor*	*CPU*	*RAM*	*Free Disk Space*
enterprise deployments	Nodes must meet or exceed the minimum hardware recommendations, except: • Dedicated Backgrounder nodes running up to two instances of backgrounder, where 4 cores may be acceptable. • Dedicated node for Tableau Prep Conductor: Minimum of 4 cores (8 vCPUs), and 16 GB of RAM. • Dedicated node for Independent Gateway: Minimum of 2 cores (4 vCPUs), 8 GB of RAM, and 100 GB free disk space.			

Important: The disk space requirement cannot be checked until you initialize TSM. If you don't have enough space, you won't be told this until after you install the Tableau Server package.

50 GB disk space available, with a minimum of 15 GB allocated to the `/opt` directory, and the remainder allocated to the `/var` directory for data storage.

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space. You may need to allocate additional disk space depending on various factors like whether you will be using extracts.

  The core Tableau Server bits must be installed in a directory with at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup. By default the install location is the `/opt` directory. You can change the installation path for Tableau Server on RHEL distros.

  If you plan to make heavy use of extracts then you may need to allocate additional disk space.You can specify a different directory for data (extract) storage during installation.

- **Network attached storage space requirements for External File Store:** If you are planning to configure Tableau Server with External File Store, you will need to estimate the amount of storage space to dedicate on your network attached storage.

  Estimating the storage size: You must take into account the amount of storage needed for publishing and refreshing extracts. In addition, you must also take into account the repository backup size unless you specifically choose the option to do your repository backup separately as described in the Option 2: Back up repository separately topic.

  - Extracts:
    - Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out how many extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.
    - Consider the space needed by the temp directory during an extract refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.
  - Repository Backup:
    - To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base` directory.
    - To obtain the exact size of the repository data, open the backup file and use the size of the `workgroup.pg_dump` file.

- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the purposes of counting cores.

- RAM shown is the minimum recommended for a single-node installation. Your installation may function better with more RAM, depending on activity, number of users, and background jobs, for example.

To see the full list of recommendations and to see the minimum requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server. For hardware specifications Tableau uses internally for testing scalability, see Hardware recommendations for production installations.

For hardware recommendations for Tableau Server in the cloud, see the following:

- Selecting an AWS Instance Type and Size in the Tableau Server on Linux in the AWS Cloud Administrator Guide
- Selecting a Google Compute Engine Virtual Machine Type and Size in the Tableau Server on Linux in the Google Cloud Platform Administrator Guide
- Selecting a Microsoft Azure Virtual Machine Type and Size in the Tableau Server on Linux in Microsoft Azure Administrator Guide

Install and Initialize TSM

This topic guides you through the process of installing Tableau Server and initializing Tableau Services Manager (TSM).

> **Important:** Do not install a beta version of Tableau Server in your production environment. You should also never restore a production Tableau Server installation using a backup of a beta version.

Prerequisites

Before proceeding, review the topic, Before you install....

Review optional initialization parameters

Before you install and initialize TSM, it is critical that you review the parameters that you may optionally set as part of the initialization operation. You can only run `initialize-tsm` once, so be sure to run it with all of the options that your organization needs. Some options, such as

non-default system user and group can only be configured as part of the initialization operation. Other configurations, such as forward proxy and dynamic port settings can be manually set after you run initialization, but doing so is a much more labor-intensive process.

For a complete list of optional parameters, see Help Output for initialize-tsm Script.

Some common scenarios where optional initialization parameters are used :

- Configuring Tableau Server to work with a forward proxy server. See Optional: common initalize-tsm parameters, later in this topic, to configure Tableau Server during installation. You can also configure Tableau Server after you install, see Configuring Tableau Server on Linux to work with a forward proxy
- Specifying dynamic port mapping. By default, most ports needed by Tableau Server are assigned (mapped) dynamically from a predefined range of ports. The port assignments are made for each service or process during installation. If you want to modify port mapping, we recommend configuring this during installation, see Controlling port remapping with initialize-tsm.
- Specifying non-default system user or group This configuration change can only be performed during initialization. See System user and groups.
- Specifying a non-default data directory. This configuration change can only be performed during initialization. See Data directory.

As a security best practice, do not expose the TSM port (by default, `8850`) to the internet.

Install Tableau Server

Install Tableau Server with your distribution's package manager, then run a script to initialize TSM. The script is included with the installed package.

> **Important**: The hostname of the server must not change after you start TSM. For example, this might happen if you use the cloud-init package to initialize a virtual machine, and you install Tableau Server on that virtual machine.

Install the Tableau Server package

By default, Tableau Server is installed in the `/opt` directory. On RHEL-like distributions you can specify a different install location.

1. Log on as a user with `sudo` access to the computer where you want to install Tableau Server.

   Note: To avoid possible complications, we recommend a user account that does not include any special characters (for example, non-ASCII, "+", "-"). These may cause problems, including a failure to fully install Tableau Server, depending on how your environment is configured.

2. Download the `.rpm` or `.deb` installer package from the Tableau Server Downloads and Release Notes page.

3. Navigate to the directory where you copied the `.rpm` or `.deb` package.

4. Use the package manager to install the Tableau Server package.

   You must install the new version to the same location as the existing version. The install location must be the same on all nodes. Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

   - On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

     - **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

       ```
       sudo yum update

       sudo yum install tableau-server-<version>.x86_64.rpm
       ```

- **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

  Run the following command:

  ```
  sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
  ```

  Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

  ```
  sudo apt-get update
  sudo apt-get upgrade
  sudo apt-get -y install gdebi-core
  sudo gdebi -n tableau-server-<version>_amd64.deb
  ```

Initialize TSM

You can specify a separate location for data (extracts and extract metadata) storage, whether or not you install Tableau to the default location.

1. Navigate to the `scripts` directory:

   ```
   cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
   ```

2. Run the following script to start TSM:

```
sudo ./initialize-tsm --accepteula --activation-service --
<optional_parameters>
```

The only required parameter for the `initialize-tsm` script is `--accepteula`. You must include this parameter to accept the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

```
/opt/tableau/tableau_server/packages/docs.<version_code>/Com-
mercial_EULA.txt
```

However, review the following common parameters before running the script.

Optional: common initalize-tsm parameters

There are a number of parameters (also referred to as, *flags*) that you can set when you run `initialize-tsm`. Common flags are listed below. For a complete list, run the command, `sudo ./initialize-tsm -h`, or see Help Output for initialize-tsm Script.

- Use the `--activation-service` option to activate Tableau Server using the Tableau authorization-to-run (ATR) service. As of Tableau Server 2021.4 and later, this option is the default for Tableau Server on Microsoft Windows, Tableau Server on LInux, and Tableau Server in containers. Server ATR is for new installs only, not upgrades. This option is ideal for cloud-based or virtual environments, but is available to anyone who can activate their copy of Tableau Server online. Selecting ATR for product activation is a permanent choice that cannot be undone later. For more information, see Activate Tableau Server using the authorization-to-run (ATR) service. If you don't want to use Server ATR to activate Tableau Server, use the `--no-activation-service` option.

- The `-a` flag to specify a user to be added to the tsmadmin and tableau groups instead of the user running this script. If you are installing with the root account, the you must specify the `-a` flag.

- The `--unprivileged-user` flag to set a different service account. By default a new user called `tableau` will be created. This account is an unprivileged service account under which most Tableau process run. We recommend creating a different user only in the case where a `tableau` user account already exists on the computer.

- The `-d` flag to specify a non-default location of the "data directory" where Tableau Server stores extracts, information about extracts, and more.

  By default, Tableau Server uses the following location for the directory:

  `/var/opt/tableau/tableau_server`

  When you set this flag, the `intialize-tsm` script will create and apply permissions to the directory that you specify. There are important restrictions that apply to changing the default directory path. See Data directory.

- The `--debug` flag for troubleshooting. This option displays each command in the script as it is run and can make it easier to troubleshoot issues. Use of this option results in extensive output to the screen.

  Note: Beginning in version 2021.3 this option has been removed and the script output that would have been displayed is logged to `/var/tmp/`.

- We recommend configuring Tableau Server for a forward proxy solution during installation.

  To do so, include the `--http_proxy` and/or `--https_proxy` flags to specify the forward proxy server. Specify the URL with the port, for example:

  `--http_proxy=http://proxy.exampe.lan:80/`

```
--https_proxy=http://1.2.3.4:443/
```

Take care to use `http` when you specify the URL for the `https_proxy` environmental variable. Do not specify the `https` protocol for the value of the `https_proxy` environmental variable.

To configure Tableau Server to bypass the forward proxy, include the `--no_proxy` flag. You should also add exceptions to this proxy configuration to guarantee that all communications within a local Tableau Server cluster (if you have one now or will have one later) do not route to the proxy server. For example:

```
--no_proxy-
y=localhost,127.0.0.1,localaddress,.localdomain.com.
```

- If you want to manually manage port assignment for TSM and Tableau Server processes, you may need to use one or more port-related switches with initialize-tsm. For more information, see Controlling port remapping with initialize-tsm.

3. Log off and log on again to the terminal before you configure Tableau Server.

   When you log on again, you create a new session in which group membership changes have taken effect. The new session also has access to the environment variables added by the `initialize-tsm` script.

Next step

- Activate and Register Tableau Server

Activate and Register Tableau Server

Before you can use Tableau Server, you must activate and register it.

Tableau Server requires at least one product key that both activates the server and specifies the number of license levels you can assign to users. You can access your product keys from the Customer Portal.

If you need to activate Tableau Server on a computer that is offline, see Activate Tableau Server Offline. If you need to activate additional product keys to add capacity to an existing Tableau Server installation, see Add Capacity to Tableau Server.

Prerequisite

Before proceeding with the procedures in this topic, complete the following prerequisites as outlined in Install and Configure Tableau Server:

- Install and Initialize TSM
- TSM uses port 8850. If you are running a local firewall, open port 8850. See Configure Local Firewall.

Use the TSM web interface

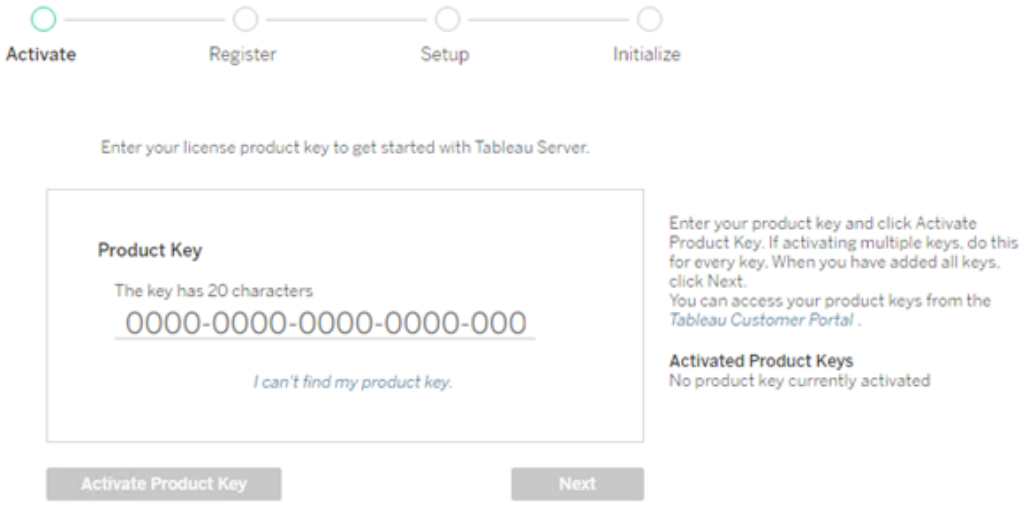
1. Sign in to Tableau Services Manager Web UI.

   The account you use must have administrative privileges on the computer where TSM is installed.

2. On the **Activate** page, Enter or paste your product key and click **Activate Product Key**.

   > **Note:** In versions prior to 2023.3.0 an option to activate a trial license displays. This option was retired (removed) starting version 2023.3.0. To learn about options for trying Tableau Server, contact your account representative.

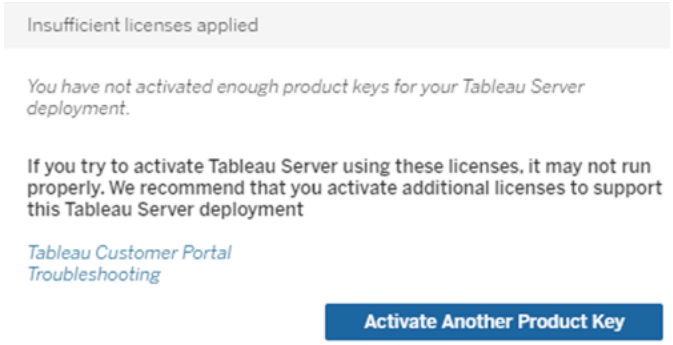
   After your product key is activated, it appears under **Activated Product Keys**.



3. To activate another product key, type over they key you just entered to add the new product key, and then click **Activate Product Key**. After your product key is activated, it appears under **Activated Product Keys**. Continue adding product keys in this manner until you're done.

4. When you’re finished activating product keys, click **Next**.

> **Note:** If the product keys you have activated don’t have the necessary capacity, such as not enough cores or only a Viewer role product key, Tableau Services Manager displays the **Insufficient licenses applied** dialog box. Click **Activate Another Product Key** to add another product key and increase capacity on your Tableau Server installation.



5. On the **Register** page, enter your information into the fields and click **Register**.

Register with Tableau. All fields are required.

Contact Information

First Name

Last Name

Phone Number

Email

Company Information

Organization

Industry

Company Size

Department

Job Role

Region Information

City

Postal Code

Country/Region

State/Province

Use the TSM CLI

Activate Tableau Server

To activate Tableau Server for production use, you must have a product key.

To activate a product key, run the following command:

```
tsm licenses activate -k <product key>
```

To activate a two-week trial, run the following command:

```
tsm licenses activate -t
```

If you are unable to activate Tableau, for example, if you get an error like this:

```
License Server not available
```

see Activate Tableau Server Offline.

What if I get an access denied error when I attempt to run TSM commands?

The account that you use to configure the rest of the installation must be a member of the `tsmadmin` group that was created during initialization. To view the user accounts in the `tsmadmin` group, run the following command:

```
grep tsmadmin /etc/group
```

If the user account is not in the group, run the following command to add the user to the `tsmadmin` group:

```
sudo usermod -G tsmadmin -a <username>
```

Register Tableau Server

After activation, you need to register Tableau Server. To do this, create a registration file and then pass it as an option with the `tsm register` command.

1. Generate a template that you can edit by running the following command:

   ```
   tsm register --template > /path/to/<registration_file>.json
   ```

2. Edit the template file to create your completed registration file.

   Here is an example of a registration file in the required format:

```
{
    "first_name" : "Andrew",
    "last_name" : "Smith",
    "phone" : "311-555-2368",
    "email" : "andrew.smith@mycompany.com",
    "company" : "My Company",
    "industry" : "Finance",
    "company_employees" : "500",
    "department" : "Engineering",
    "title" : "Senior Manager",
    "city" : "Kirkland",
    "state" : "WA",
    "zip" : "98034",
    "country" : "United States",
    "opt_in" : "true",
    "eula" : "true"
}
```

3. After saving changes to the file, pass it with the `--file` option to register Tableau Server:

```
tsm register --file /path/to/<registration_file>.json
```

   For example:

```
tsm register --file /usr/share/tableau-reg-file.json
```

If you have a product key for Data Management or Advanced Management, you must activate that key to use the additional functionality. Product keys for these licenses should only be activated after at least one capacity product key has been activated on the server.

- For license information on Tableau Data Management, see License Data Management.

- For license information on Tableau Advanced Management, see About Tableau Advanced Management on Tableau Server.

Next step

- Configure Initial Node Settings

Activate Tableau Server Using the Authorization-To-Run (ATR) Service

You can use the Server authorization-to-run (ATR) service to activate Tableau Server deployed in on-premises, cloud, container, or virtual environments without running out of license activations. The ATR service achieves this by providing short-term leases of configurable duration (ATR duration) which ties the license to the device until product key expiration date is met. ATR handles activation capacity tracking so if there is an underlying hardware change, maximum activation errors don't occur. This option is recommended for all Tableau Server installations.

If you are activating Tableau Server online (this is the default), Tableau Server connects to various internet locations for licensing purposes. For more information, see Communicating with the Internet.

If you are activating Tableau Server offline, you can still use ATR service to activate, however there are slight differences in how ATR duration works with offline activations. For more information, see ATR Duration for Offline Activations.

We recommend using the Server authorization-to-run (ATR) service to simplify server licensing. If you chose not to use Server ATR, you will use the legacy licensing technology which does not provide the dynamic management capabilities of Server ATR.

Your choice of activation type will be permanent for this installation of Tableau Server. To change this later, you must backup, remove, and then reinstall Tableau Server.

How Tableau Server ATR works

When ATR service is enabled, Tableau Server periodically contacts a Tableau-hosted authorization-to-run (ATR) service to verify that Tableau is authorized to run, based on its license and

the length of the authorization window (ATR duration or lease). As long as this communication is successful, Tableau runs without any impact to the user.

When Tableau Server is activated offline and ATR is enabled, Tableau Server can not periodically contact the ATR service to verify Tableau is authorized to run. Instead, Tableau Server internally tracks the ATR Duration from when the product key was first activated offline.

ATR Duration

By default, an instance of Tableau Server is given a 5 day ATR lease (duration) to successfully contact the ATR service for verification. This means that after the initial authorization, Tableau Server could be used for 5 days without network connection before the activation expired. The authorization checks between Tableau Server and ATR service are attempted regularly, and each time a check is successful, the ATR duration is reset to its full length.

The authorization check frequency varies, and is dependent on the ATR duration:

ATR duration	**Authorization check frequency**
< 4 hours	every 15 minutes
< 24 hours	every 1 hour
< 7 days	every 12 hours
> 7 days	every 24 hours

For example: If the ATR duration is 48 hours, Tableau Server will contact the ATR service every 12 hours to complete an authorization check and the ATR duration is reset to 48 hours after each successful authorization check. The ATR duration will then begin to count down to 0 until the next authorization check. If the Tableau Server machine is shut down or there is no internet access, Tableau Server can no longer contact ATR service. If this happens, the ATR duration will not be reset to 48 hours and will continue to count down to 0. If Tableau Server is

not started or cannot communicate with the ATR service before the ATR duration reaches 0, the license will expire and you must activate the license again.

> **Note:** You should keep your Tableau Server running as much as possible. If Tableau Server cannot successfully complete an authorization check within the ATR duration period, ATR service will reclaim the license lease and then you'll need to reactivate your license.

To view the ATR duration, see tsm licenses atr-configuration get.

Set or change the Server ATR duration

When using Server authorization-to-run (ATR) to activate Tableau Server, you can change the ATR duration and use a setting different than the default of 432000 seconds (5 days). If you're creating a test server or virtual machine (VM) that will have a short lifespan, you might want to shorten the ATR duration. Similarly, if you have a server that you plan to keep for a long time, you might want to lengthen the ATR duration.

If you start up new VMs frequently, reducing the ATR duration can allow older VMs to release their capacity, allowing it to be used by new VMs. On the other hand, if you increase the ATR duration, renewal cycles will be longer but capacity is not released as often.

To change the ATR duration, you use the `tsm licenses atr-configuration set -duration <value_in_seconds>` command. For more information, see tsm licenses atr-configuration set.

ATR duration in seconds	Minimum	Maximum	Default
Tableau Server (container)	3600 (1 hour)	2593000 (30 days)	14400 (4 hours)
Tableau Server (non-container)	3600 (1 hour)	7776000 (90 days)	432000 (5 days)

Tableau Server on Linux

To set ATR duration and manually start the Activation Service, run the following commands:

1. On the initial node, open a terminal session, and then use the following commands:
2. `tsm licenses atr-configuration set --duration <value_in_seconds>`
3. `tsm pending-changes apply`
4. `sudo su -l tableau`
5. `systemctl --user stop activationservice_0`
6. `systemctl --user start activationservice_0`
7. Verify that the Activation Service is running by using the `tsm status -v` command. Tableau Server Activation Service should be listed as "is running". If the Activation Service is not started, then run:

   `systemctl --user restart activationservice_0`

Tableau Server in a Container

To set ATR duration and manually start the Activation Service, run the following commands:

1. On the initial node, open a terminal session, and then use the following commands:
2. `tsm licenses atr-configuration set --duration <value_in_seconds>`
3. `tsm pending-changes apply`
4. `sudo su -l tableau`
5. `supervisorctl stop activationservice_0`
6. `supervisorctl start activationservice_0`
7. Verify that the Activation Service is running by using the `tsm status -v` command. Tableau Server Activation Service should be listed as "is running". If the Activation

Service is not started, then run:

```
supervisorctl restart activationservice_0
```

Note: For Tableau Server running in a container, the TTL Start and TTL End values reflect the current lease being used by Tableau Server. Container leases that are shorter than a day are refreshed hourly, but longer leases can take up to 24 hours to renew.

ATR Duration for Offline Activations

When Tableau Server is activated offline, the ATR duration is set to either one year from the day the product key was activated offline, or to the renewal date of the product key, whichever comes first. Once the ATR duration reaches the set date, the product key will become deactivated. Since Tableau can not communicate with the ATR service in offline environments, there are no authorization checks.

If you use the `tsm licenses atr-configuration get` command to view the ATR duration for offline activations, the result will be `0`, which is expected. To view ATR duration for offline activations, use the `ATRDiag -product "Tableau Server"` command instead. For more information, see ATRDiag.exe Command Line Reference. In the resulting output, the `TTL End` date is the date the ATR duration ends.

Updating ATR duration in offline environments

Because authorization checks are not possible in an offline environment, the ATR duration is set to a fixed value as described above. To avoid unplanned Server downtime, be aware of the ATR duration in your installation, and plan to update the ATR duration before it expires. How you do this depends on if you have a USL key or a non-USL key:

- **USL keys:** If you have a USL key, follow the instructions here: USL offline license entitlement updates.
- **Non-USL keys:** If you have a non-USL key:

- Deactivate the existing license key. See Deactivate Tableau Server Offline.
- Activate your new key. See Activate Tableau Server Offline

Move a Server ATR product key to another Tableau Server

If you want to stop using your product key on an existing Tableau Server and use it on a new installation of Tableau Server, you can move your Server ATR product key. You might want to move product keys when:

- Switching between development or pre-production environments.
- Moving Tableau Server to upgraded hardware.
- Moving Tableau Server to cloud infrastructure.
- Using an ephemeral Tableau Server, such as a virtual machine (VM) or container.

Deactivate a product key for reuse on another Tableau Server

Deactivate a product key in version 2021.4 and later

You can remove product key(s) that were activated using Server ATR when you want to reclaim license capacity from one Tableau Server installation and use it on another. For more information about reclaiming a product key, see Deactivate Product Key.

Deactivate a product key in version 2021.3 and earlier

When ATR service is used to activate a Tableau Server license you cannot manually deactivate the license. You can have one production and two non-production installations per license. If you have unused activations on a license, you can activate the same license on another Tableau Server. If you're out of activations, you can still activate the license after the ATR lease expires by taking the following steps:

1. Disable the existing Tableau Server from refreshing its leased activation. You can do this using any of the following methods:

- Shut down your existing Tableau Server.
- Uninstall the existing Tableau Server.
- Disconnect the existing Tableau Server from the Internet by unplugging the ethernet cable or disconnecting wifi.

2. After the ATR lease expires, you can reuse the license on another Tableau Server.
3. On the new computer, install Tableau Server.

   When prompted, activate Tableau Server using the same license.

Activate Tableau Server Offline

When you install Tableau Server, you have to activate at least one product key, but we recommend that you activate all Tableau Server licenses found in the Tableau Customer Portal. Doing this activates the server, and specifies the number of license levels you can assign to users. For offline activations, you should activate the product key listed in the **Offline Activation Id** field in the Tableau Customer Portal. For information about finding the right key, see the Find the Correct Key to Activate on Tableau Server Knowledge Article.

There are also times you may need to activate licenses after Tableau Server is installed, for example, if you add capacity to your server, or get a new product key. If you don't have your product key, you can get it from the Tableau Customer Account Center.

> **Note:** Activating any product key after Tableau Server has already started will require a Tableau Server restart for the changes to take effect.

In most cases, you can activate your key directly from Tableau Server, either during installation, or later, using the Tableau Services Manager (TSM) Licenses page, but there are some circumstances that don't allow you to do this. If your computer is not connected to the internet for example, or has a firewall that restricts access outside your intranet. In these cases you need to do an offline activation.
Tableau Server in a Container only supports license activation using Server ATR. Offline

activation using Server ATR is supported in 2023.1 and later. This functionality is available in Containers but requires extra steps and approval. If you need to run Tableau Server in a Container in an air-gapped or offline environment, contact your Account representative for more information.

Offline activation and login-based license management (LBLM)

Beginning in Tableau Server version 2023.1.0, offline activation is supported for LBLM when your server is configured to use the Authorization-to-Run (ATR) service. You can only configure Tableau Server to use the ATR service during a new install. Upgrading customers with existing server installations need to install a new instance of Tableau Server version 2023.1.0 or later and restore a backup of their existing installation to that new instance. For information on this process, see Using a Blue/Green approach for upgrading Tableau Server. For more information about ATR service, see Activate Tableau Server Using the Authorization-To-Run (ATR) Service.

Offline activation and updateable subscription licenses (USL)

Offline activation of updateable subscription licenses requires special steps. For details, see Activating USL in Offline or Disconnected Environments.

There are two scenarios in which you may need to do an offline activation:

- Offline activation during install—To complete an offline activation when you are installing Tableau Server.
- Offline activation of licenses after install—To complete an offline activation after your server is installed and running.

Offline activation overview

Offline activation of Tableau Server involves the following steps:

1. Generate an offline activation request file.
2. Copy the offline activation request file to a computer with internet access.
3. Upload the offline activation request file to the Tableau activation website.

4. Download the resulting offline activation response file from the website. You'll use this file to activate Tableau Server

Offline activation file name changes

Beginning in Tableau Server version 2023.1, the Tableau licensing system supports two underlying licensing technologies. From an administrative perspective, the only configuration difference between the two systems is the file types that are generated and consumed for offline activation. The licensing technology is determined during the initial installation of Tableau Server, and cannot be changed after install.

We refer to the legacy (and still supported) version of licensing technology as FlexNet. The latest version of the technology is referred to as Server ATR. For more information, see Activate Tableau Server Using the Authorization-To-Run (ATR) Service. The following table describes the file naming nomenclature for each technology. The table also includes the generic reference.

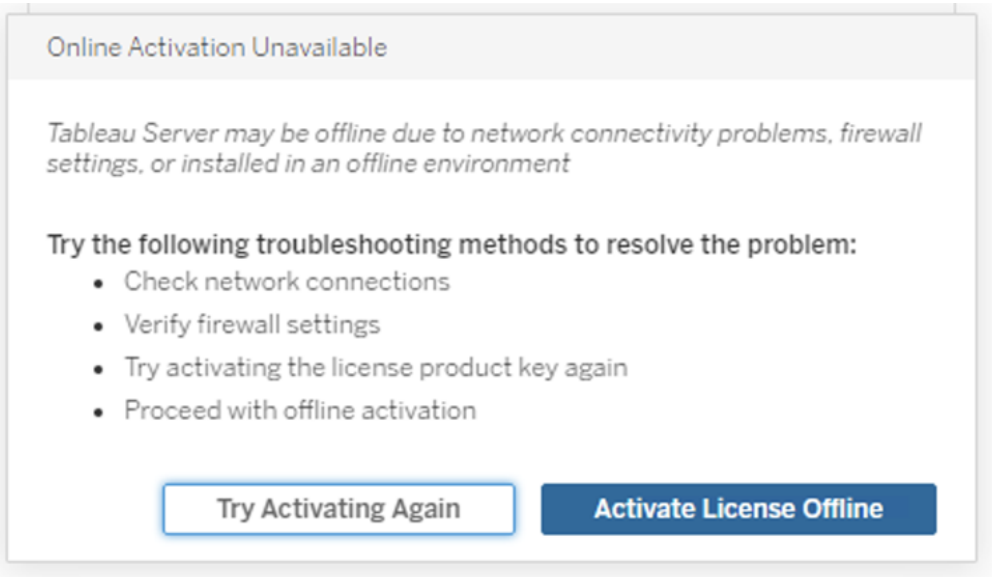
Generic file name	Server ATR file names	FlexNet file names
OfflineActivationRequest	OfflineActivationRequestFile_yyyyMMdd.hhmmss.json	TableauOfflineActivationRequest.tlq
OfflineActivationResponse	OfflineActivationLicensingAtrs.zip	activation.tlf

Note: Since this documentation supports multiple versions of Tableau Server, we will use the generic file name references (OfflineActivationRequest and OfflineActivationResponse) for the rest of this topic. You can identify the licensing technology your Tableau Server installation uses according to the file type that generated in the steps that follow.

Use the TSM web interface

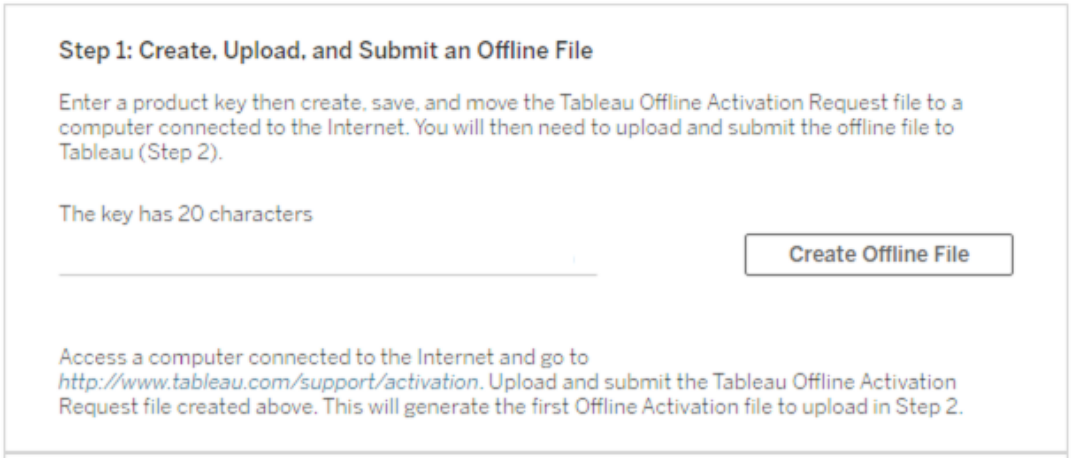
If you attempt to activate your product key from the TSM licenses page and see a dialog that says online activation is unavailable, you can activate the key offline. The offline activation process must be completed once for each product key.

1. Click **Activate License Offline**.



2. Create an offline activation request file (OfflineActivationRequest) for the product key.

   Create an OfflineActivationRequest file you will upload to the Tableau activation web-site. If your product key is not pre-filled in the form, enter your key and click **Create Off-line File** to generate an OfflineActivationRequest file on the local computer.



Copy the OfflineActivationRequest file to a computer with internet access. You need to upload this file to the Tableau activation website to generate an activation response file.

3. Upload and submit the OfflineActivationRequest file.

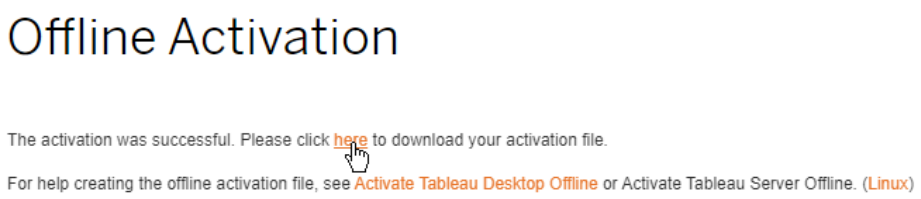
   You will upload and submit the OfflineActivationRequest file to the Tableau activation website. This automatically generates an activation response file (OfflineActivationResponse) that you can download and copy back to the Tableau Server computer.

   a. On the computer where you copied the OfflineActivationRequest file, open a browser and go to http://www.tableau.com/support/activation to open the Tableau Support Activation page.

   b. On the Offline Activation page, click **Choose File** to select the OfflineActivationRequest file.

   c. Click **Upload Activation File** to submit the file to the Tableau activation website.

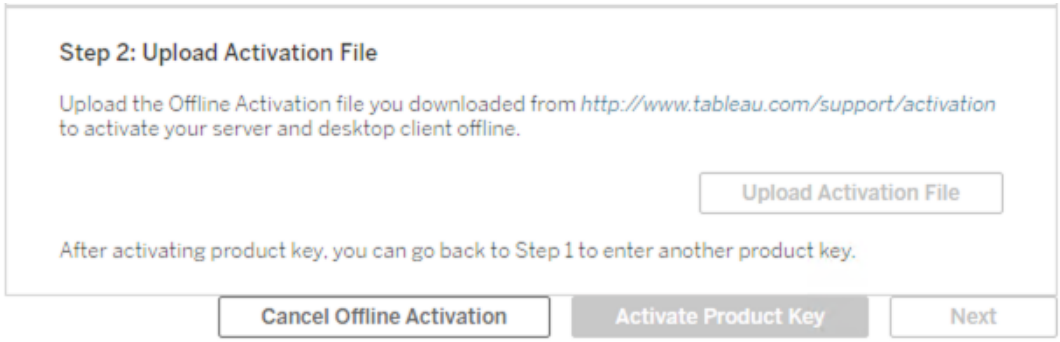
   d. Click the `here` link to download the OfflineActivationResponse file to your computer.



e. Copy the OfflineActivationResponse file to the computer where Tableau Server is installed.

4. Upload the OfflineActivationResponse file.

   On the Tableau Server computer, click **Upload Activation File** to upload the OfflineActivationResponse file to Tableau Server. When you do this successfully, the **Activate Product Key** button is enabled.



5. Click **Activate Product Key** to complete the offline activation.

6. (Skip this step if you are installing Tableau Server for the first time.)

   Restart Tableau Server for licensing changes to take effect.

Use the TSM CLI

Step 1. Log in to Tableau Services Manager

Before you can proceed you must log in to Tableau Services Manager (TSM). To log in to TSM, run the following command:

```
tsm login -u <username>
```

What if I can't log in?

The account that you use to configure the rest of the installation must be a member of the `tsmadmin` group that was created during initialization. To view the user accounts in the `tsmadmin` group, run the following command:

```
grep tsmadmin /etc/group
```

If the user account is not in the group, run the following command to add the user to the `tsmadmin` group:

```
sudo usermod -G tsmadmin -a <username>
```

After you have added the user to the `tsmadmin` group, run the `tsm login` command.

Step 2. Generate an offline activation request file

1. On the initial node, open a terminal session.
2. Type this command to get your offline activation file:

   ```
   tsm licenses get-offline-activation-file -k <product-key> -o
   <target-directory>
   ```

   You can get your product key from the Tableau Customer Portal. The target directory must already exist.
3. Copy the offline activation file (OfflineActivationRequest) from the target directory to a computer that has internet access.

Step 3. Upload the offline activation request to the Tableau activation website

1. On the computer that has internet access, go to the Tableau Product Activations page.
2. Complete the instructions to upload your OfflineActivationRequest file.

This creates an activation response file (OfflineActivationResponse).

3. Download the OfflineActivationResponse file from the Tableau activation website.

Step 4. Initialize or activate your license

1. Copy the OfflineActivationResponse file to a location accessible from the Tableau Server computer.

2. Run the following command:

   ```
   tsm licenses activate -f <path-and-activation-file>
   ```

   Note: When using ATR to activate Tableau Server, `<path-and-activation-file>` should point to the packaged OfflineActivationResponse `.zip` file. Do not unzip the OfflineActivationResponse file prior to running this command.

3. (Skip this step if you are installing Tableau Server for the first time.)

   Restart Tableau Server for licensing changes to take effect:

   ```
   tsm restart
   ```

4. (Optional) To verify that all licenses are activated, you can run this command:

   ```
   tsm licenses list
   ```

   If you have completed the steps above, you should see a success message:

   ```
   Activation successful.
   ```

   Tableau Server is activated. If you need additional assistance, contact Tableau Technical Support.

Configure Initial Node Settings

This topic describes how to configure essential server settings as part of the initial Tableau Server installation process.

Prerequisite

Before proceeding with the procedures in this topic, complete the following prerequisites as outlined in Install and Configure Tableau Server:

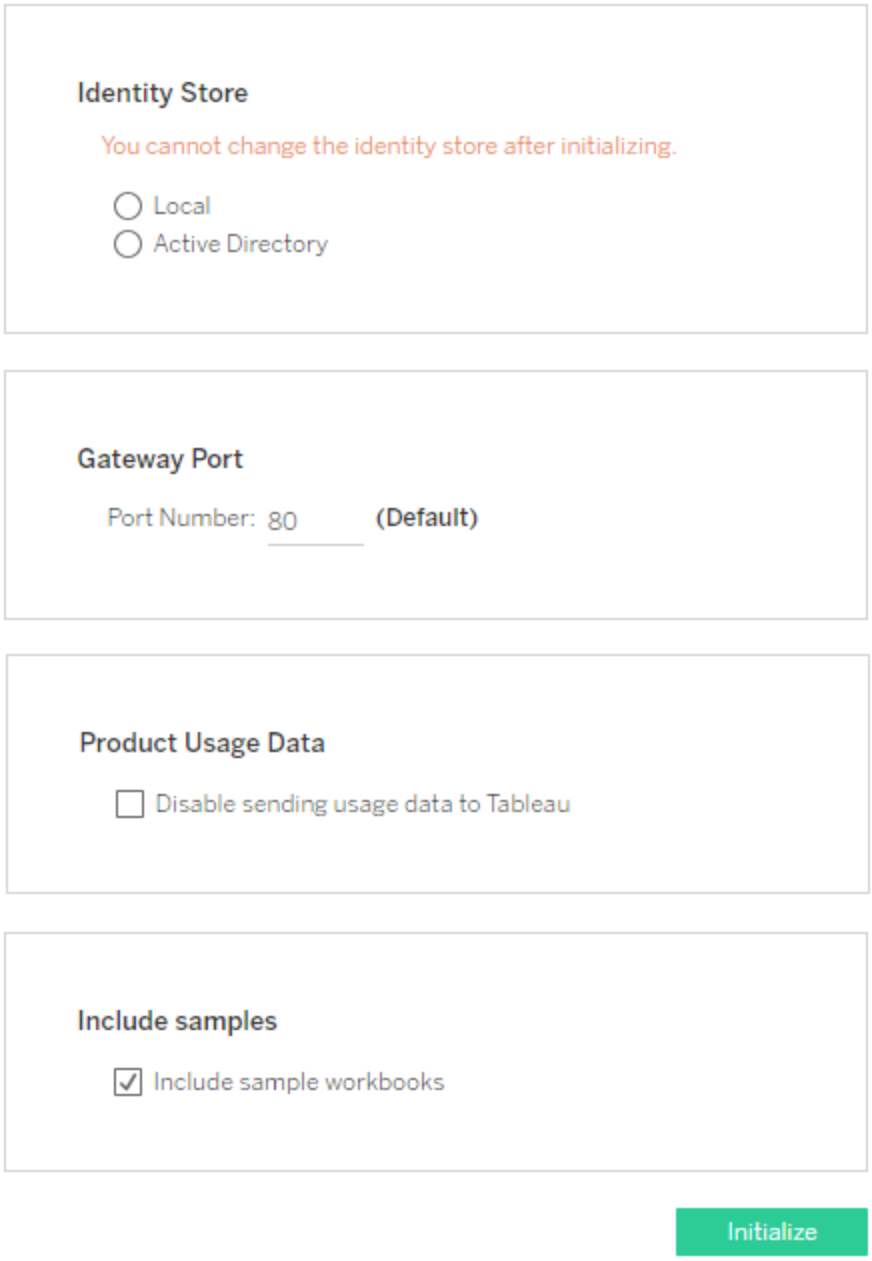
- Install and Initialize TSM
- Activate and Register Tableau Server

You may also need to configure your local firewall for Tableau Server traffic. See Configure Local Firewall.

Use the TSM web interface

After you have activated and registered Tableau Server, the installation program will display the Setup page.

Note: If you need to configure Tableau Server to connect to an LDAP directory that is not Active Directory, then you must use the TSM CLI.



Identity store settings

You must configure the identity store settings for the Tableau Server computer. The identity store manages Tableau Server accounts. You can configure the identity store to synchronize with an external directory (for example, OpenLDAP or Active Directory) or you can configure the identity store to manage and store accounts on Tableau Server. If you will be using a

single sign on solution (OpenID, SAML, Kerberos, etc) then review the following topics before configuring the identity store:

- Identity Store
- Authentication

> **Important**: After you have configured and applied settings for the identity store, it cannot be changed.

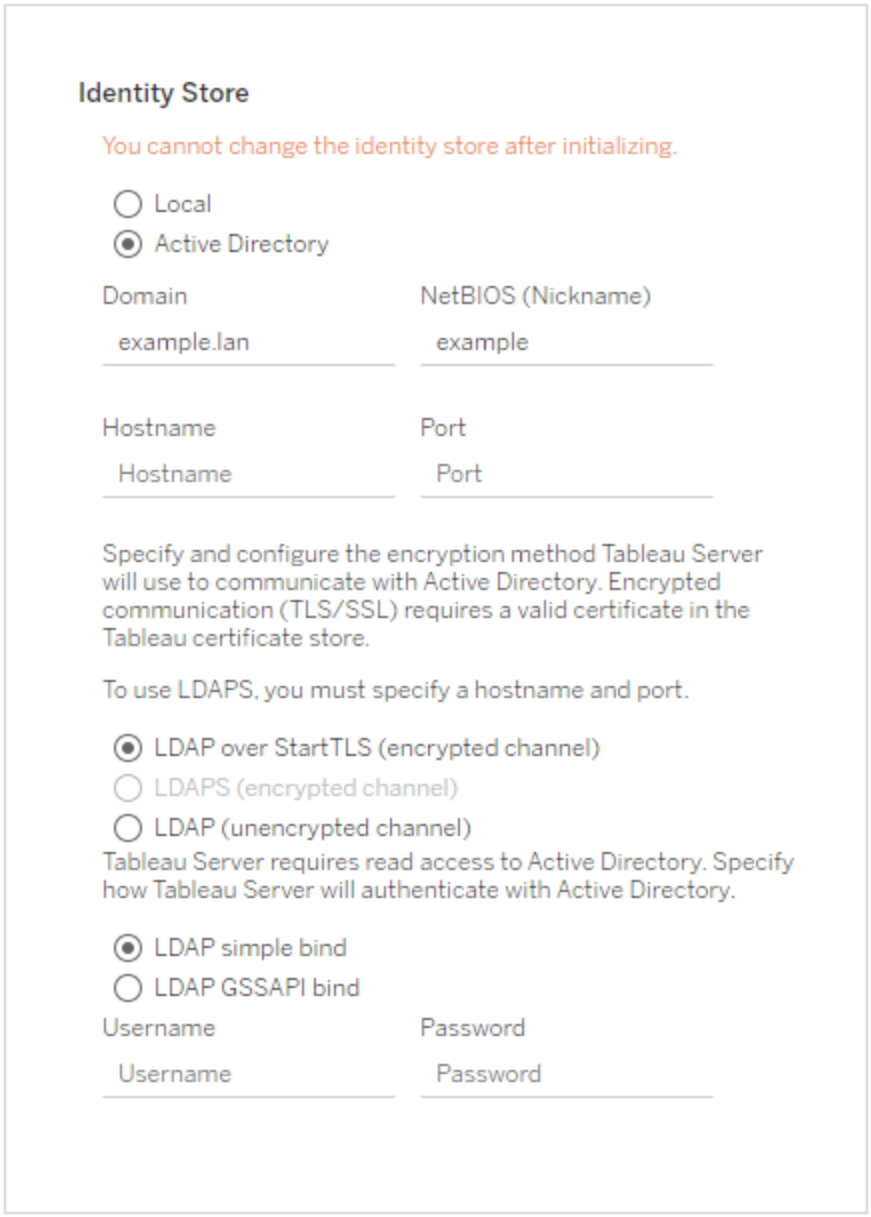
If you select **Active Directory**, Tableau Server will populate the **Domain** and **NetBIOS** fields from the computer on which you are running Setup. In some cases, Setup may not display these attributes. For more information about how Tableau Server connects and communicates with Active Directory, see User Management in Deployments with External Identity Stores.

Tableau Server requires read access to Active Directory.

You can use simple bind or GSSAPI bind to authenticate Tableau Server with Active Directory. If Tableau Server requires access to domains outside of the domain where you are installing, you will need to create duplicate bind accounts. See Duplicate bind accounts for domain trust.

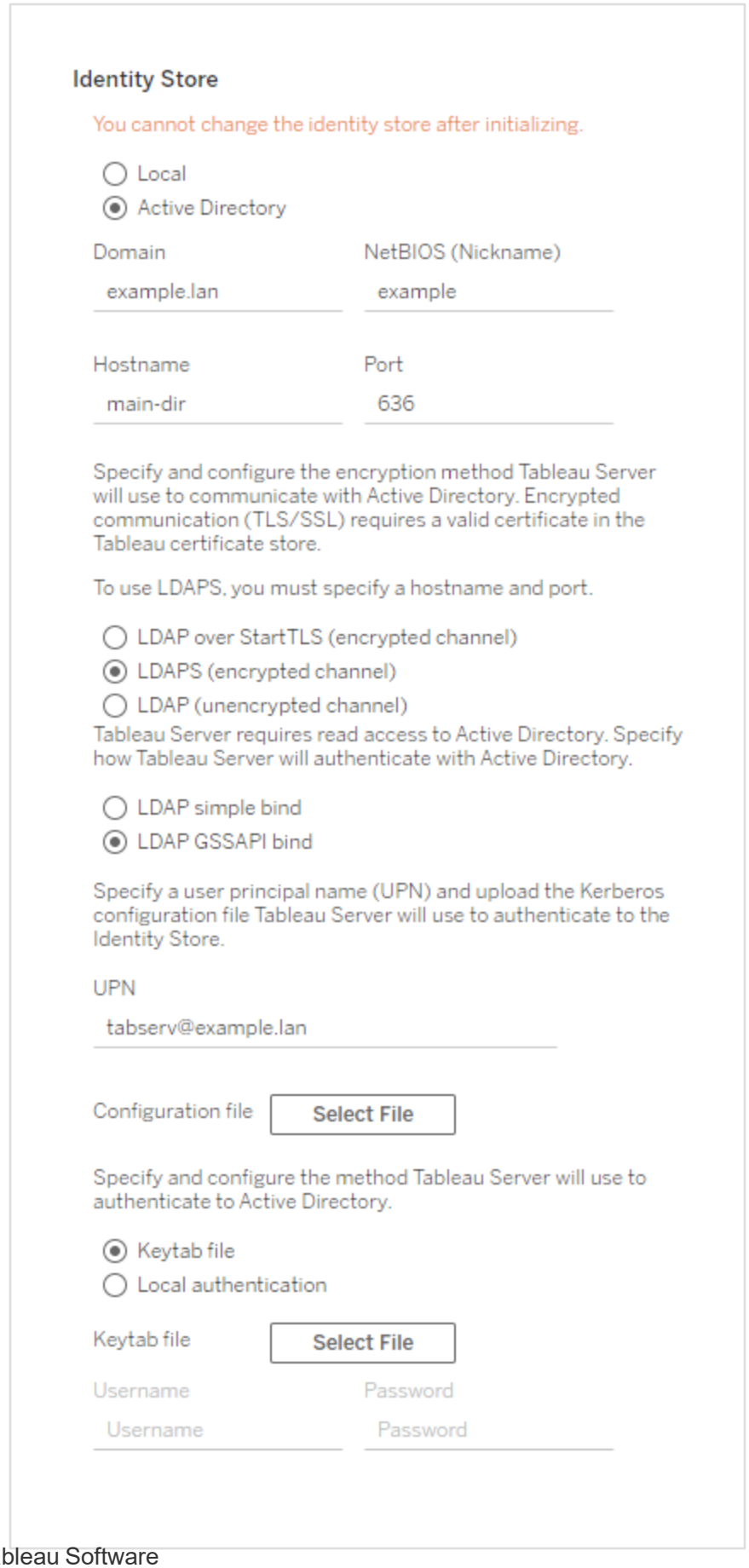
We recommend configuring an encrypted channel for LDAP. See Configure Encrypted Channel to LDAP External Identity Store.

LDAP simple bind



If you are using simple bind to authenticate with Active Directory, enter a domain account and password.

LDAP GSSAPI bind

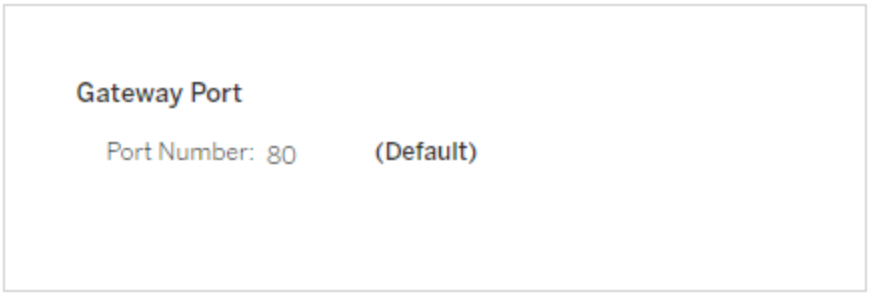

Identity Store
You cannot change the identity store after initializing.
Local
Active Directory
Domain
example.lan
NetBIOS (Nickname)
example
Hostname
main-dir
Port
636
Specify and configure the encryption method Tableau Server will use to communicate with Active Directory. Encrypted communication (TLS/SSL) requires a valid certificate in the Tableau certificate store.
To use LDAPS, you must specify a hostname and port.
LDAP over StartTLS (encrypted channel)
LDAPS (encrypted channel)
LDAP (unencrypted channel)
Tableau Server requires read access to Active Directory. Specify how Tableau Server will authenticate with Active Directory.
LDAP simple bind
LDAP GSSAPI bind
Specify a user principal name (UPN) and upload the Kerberos configuration file Tableau Server will use to authenticate to the Identity Store.
UPN
tabserv@example.lan
Configuration file
Select File
Specify and configure the method Tableau Server will use to authenticate to Active Directory.
Keytab file
Local authentication
Keytab file
Select File
Username
Password
Username
Password

To bind with GSSAPI you can bind with credentials or with a keytab file. If you are using a keytab file, we recommend creating a keytab specifically for the Tableau Server service. See Understanding Keytab Requirements.

Gateway port

The default port for web access to Tableau Server (via HTTP) is port 80. If the installation program determines that port 80 is in use when you first install Tableau Server, an alternate port (for example 8000) is used and shown in the Port number box.

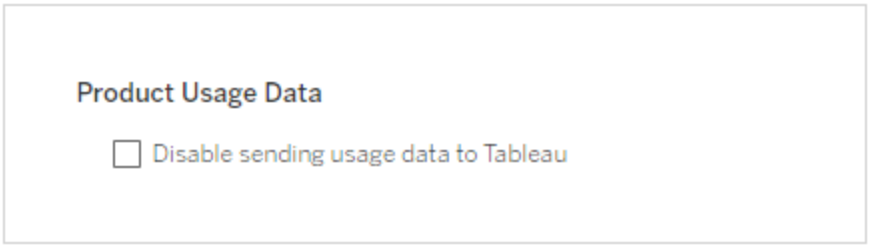
You may need to change the port for other networking needs, for example, if you have a hardware firewall or proxy in front of the Tableau Server host, this might make running a back-end system on port 80 undesirable.



Product usage data

By default, Tableau Server shares usage data with Tableau that helps us better understand how you use our products, improve your overall experience, and build highly intelligent features that make Tableau even more powerful.

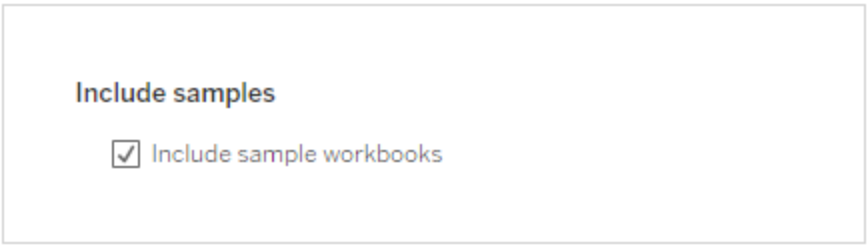
Clear this option if you do not want usage data sent to Tableau.



You can also change this setting after installation, on the TSM Maintenance tab in the TSM Web UI, or using the TSM CLI. For more information, see Server Usage Data.

Sample workbook installation

By default, Tableau Server will install sample workbooks in the Default site when you initialize the server.

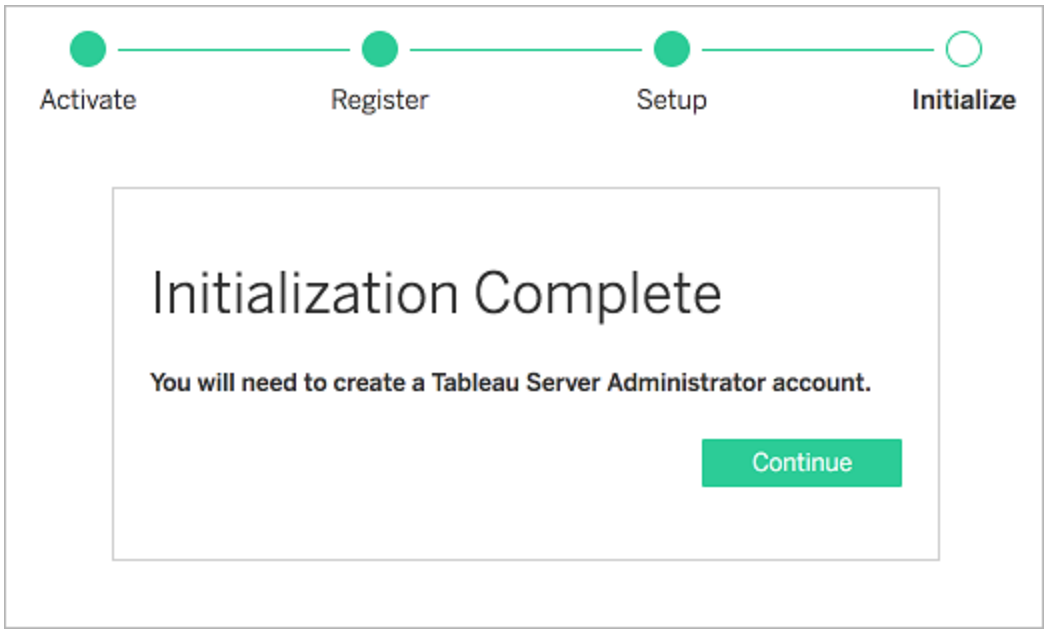


Alternatively, you can publish samples after installation by using the publishsamples tabcmd command.

Initializing install

After you have configured the options on this page, click **Initialize**.

The initialization process can take a while. When initialization is complete the following page is displayed:



Use the TSM CLI

First, configure identity store, gateway settings, and sample workbook installation. Then apply the changes, optionally verify your LDAP connection, and then initialize Tableau Server

Configure identity store settings

You must configure the identity store settings for the Tableau Server computer.

> **Important**: After you have configured and applied settings for the identity store, it cannot be changed.

Use the json template in identityStore Entity to create a json file. After you have filled in the options with the appropriate values, you can then pass the json file with this command:

```
tsm settings import -f path-to-file.json.
```

Configure gateway settings (optional)

Depending on your network requirements, you may need to configure the gateway settings for the Tableau Server computer. For example, if you are enabling SSL or configuring access to Tableau Server with a reverse proxy, you may need to configure gateway settings. See gatewaySettings Entity for more information.

Use the json template in gatewaySettings Entity to create a json file. After you have filled in the options with the appropriate values, you can then pass the json file with this command:

```
tsm settings import -f path-to-json-file.json.
```

Configure product usage data (optional)

By default, Tableau Server shares usage data with Tableau to help us better understand how you use our products. This allows us to improve your overall experience and build highly intelligent features that make Tableau even more powerful.

Tableau collects only behavioral and usage data, never any of your confidential database values, and your usage data will never be shared or sold; its sole purpose is to improve your Tableau experience.

If you do not want to share product usage data, use the json template in shareProductUsageDataSettings Entity to create a json file, and specify a value of `false`. Then pass the json file with this command:

```
tsm settings import -f path-to-json-file.json.
```

You can also change this setting after installation, on the TSM Maintenance tab or using the TSM CLI. For more information, see Server Usage Data.

Configure sample workbook installation (optional)

By default, Tableau Server will install sample workbooks in the Default site when you initialize the server.

If you do not want to install sample workbooks during installation, run the following command:

```
tsm configuration set -k install.component.samples -v false
```

You can publish samples after installation by using the publishsamples tabcmd command.

Apply pending configuration changes

Now that you've created and set initial configuration, you must apply them. When you apply configuration changes, tsm will verify the settings you've set before committing them.

To apply configuration changes to Tableau Server, run this command:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Once this command has completed, TSM processes are running, and Tableau Server is configured but is not running.

Verify LDAP configuration (Optional)

If your identity store uses LDAP, then we recommend verifying LDAP connectivity before proceeding.

To do so, run the following commands before you initialize the server:

```
tsm user-identity-store verify-user-mappings -v <user name>
```

```
tsm user-identity-store verify-group-mappings -v <group name>
```

User and group names must be valid names that exist in the LDAP server that you are connecting to. If your LDAP connection is set up correctly, then the user or group attributes will be returned to the shell. If your connection is not set up correctly, then an error will be returned.

Initialize and start Tableau Server

- To initialize and start Tableau Server, use the `--start-server` option:

  ```
  tsm initialize --start-server --request-timeout 1800
  ```

  This saves time by starting the server running after initialization.

- If you are going to reconfigure Tableau Server after initialization, leave the `--start-server` option off:

  ```
  tsm initialize --request-timeout 1800
  ```

  This stops the server after initialization.

Start Tableau Server. If you did not use the `--start-server` option during initialization and are finished configuring Tableau Server, use this command to start the server:

```
tsm start --request-timeout 900
```

Note: If you experience timeouts when installing or configuring Tableau Server, you may need to specify a longer timeout. For more information, see Install fails due to timeouts.

Next Step

After initialization is complete, create the Tableau Server administrator user account. See Add an Administrator Account.

Configuration File Example

This article provides an example of a complete JSON configuration file, with gatewaySettings and identityStore entities specified. In addition, a configuration key sets the gateway timeout to 900 seconds.

Your configuration file will look different depending on the options you need to set.

You might set multiple .json configuration files during installation. To set the values for each file in Tableau Server, you run the following command, once for each configuration file:

```
tsm settings import -f path-to-file.json
```

After you set the configuration files, run tsm pending-changes apply to apply the changes from all of the .json files you've set.

```
{
   "configEntities": {
      "gatewaySettings": {
         "_type": "gatewaySettingsType",
         "port": 80,
         "publicHost": "localhost",
         "publicPort": 80
      },
      "identityStore": {
         "_type": "identityStoreType",
         "type": "local",
         "domain": "example.lan",
         "nickname": "EXAMPLE"
      }
    },
     "configKeys": {
        "gateway.timeout": "900"
     }
}
```

Entities vs keys

As shown in the example above, there are two classes of configuration parameters: configEntities and configKeys.

configEntities

Certain types of configuration are done through entity sets that map to specific scenarios, such as the identity store and gateway configurations. When you pass a set of configEntities

with the `tsm settings import -f path-to-file.json` command, TSM validates the configuration. If values passed are invalid, TSM will provide an error. This enables you to make changes during the configuration process, rather than experience a configuration failure at initialization or run time.

Entities can be set only by including a configEntities block in a .json file.

> **Important**: All files that are referenced in configEntities must be located on the local computer. Do not specify UNC paths.

configKeys

Entities cover only a small portion of the configuration values that can be set. Hundreds of keys correspond to parameters stored in .yml files. Tableau Server uses these parameters to store all of the configuration information for all services.

You can set individual keys with the tsm configuration command. But during deployment, setting them along with other configuration scenarios in JSON files, as shown above, is more convenient.

Unlike configEntities, configKeys are not validated.

> **Note:** We do not recommend setting parameters that are not documented in tsm configuration set Options.

Server Usage Data

The Tableau Server administrator can control whether or not usage data from Tableau Server is sent to Tableau. By default this option is enabled, and can be disabled at initial install, or after installing Tableau Server, using the TSM Web UI or command line. For details about this usage data, see Tableau Product Usage Data.

In addition to product usage data, Tableau products send Basic Product Data to Tableau. This data is sent whether or not you have disabled the sending of product usage data. You can disable the sending of Basic Product Data separately. For details, see Basic Product Data.

Disable Usage Data Sharing

You can disable the sharing of usage data when you install Tableau Server, or at any time after installation.

Disabling the sharing of usage data at install

To disable the sharing of usage data with Tableau when you are installing Tableau Server, clear the option during the initial configuration of server. For details, see Product usage data.

Disabling the sharing of usage data after install

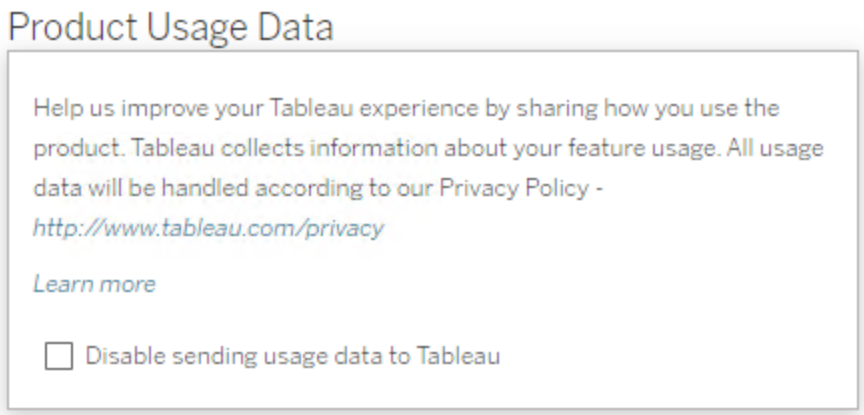
Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Maintenance** tab.

3. Under Other Maintenance Tasks, in Server Usage Data, clear **Send usage data to improve Tableau features**:



4. When you are finished, click **Pending Changes**, and then click **Apply Changes and Restart**.

Use the TSM CLI

If you do not want to share product usage data, disable the option using this tsm configuration command:

```
tsm configuration set -k shareproductusagedata.enabled -v false

tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Basic Product Data

By default Tableau products send usage data to Tableau so that we can understand how customers use our software, and gain insights into where they are successful and where they might run into problems that we can address. For example, this data can help us learn where

upgrades are commonly failing and allow us to make product changes to address those issues, or identify what portion of our user base needs informed about a security issue that applies to a specific version of Tableau Server. You can disable the sending of this data at installation time, or later. For details on how to do this, see the instructions for Tableau Desktop or Tableau Server.

Even when you disable the sending of product usage data, certain basic product data is sent to Tableau. This Basic Product Data includes information about products and their processes, including which product or process is running, when they start up, what operating system they are running on, licensing information, which machine or cluster of machines has sent the data (using unique pseudonymized identifiers), and whether the product is configured to send product usage data.

You can disable the sending of Basic Product Data at the machine level, or at the enterprise level, by blocking traffic sent to **prod.telemetry.tableausoftware.com**.

Disabling sharing of Basic Product Data on individual computers

> **Important:** This procedure involves modifying your local `hosts` file. If you do not know what this is, you should not change it. You should only make this change if you understand the implications of making changes to the file, know how to change the file, and have made a backup of the file for safety.

Modifying `hosts` files changes network behavior for computers. Detailed instructions for modifying `hosts` files are provided by operating system providers such as Microsoft, Apple or Linux Distributions.

1. Make a copy of your existing hosts file and save it to a computer that is not your Tableau computer. This is your backup, in case you need to reverse your changes. Do not start modifying the file until you have made a backup copy of it.
2. Modify your computer's `hosts` file to include these lines:

```
# Stops sending Product Usage to Tableau (prod.tele-
metry.tableausoftware.com).
# Learn more here: http:\\tableau.com\derived-data
127.0.0.1    prod.telemetry.tableausoftware.com
```

The first and second lines are comments, explaining the third line.

The third line prevents all traffic to `prod.telemetry.tableausoftware.com` (`http://-prod.telemetry.tableausoftware.com/`) from leaving your local machine by sending it to the Internal host loopback address. The data does not get sent outside the computer.

Disabling the sharing of Basic Product Data at the enterprise level

To disable sending of Basic Product Data on an enterprise level, modify your Network Firewall to prevent outbound traffic to `prod.telemetry.tableausoftware.com`.

This domain is used by Tableau to receive the Basic Product Data about process launch and shutdown. It is also used for the more general Product Usage Data. Blocking traffic to this domain it you will prevent both kinds of data from being sent.

Traffic to this domain will occur on Ports 80 (for initial registration of our Product Data clients) and on Port 443 (for all subsequent traffic). To completely prevent product data from being sent, block all traffic to this domain.

For details on how to configure your network firewall, refer to your vendor or your internal IT department. Tableau cannot provide these instructions.

Add an Administrator Account

The final step in activating Tableau Server is to add the initial administrator account. The administrator will have all access to the server including the ability to manage users, groups, and projects.

The server must be running when you create the initial admin user.

If you have configured the Tableau Server identity store to use LDAP or Active Directory, then the initial administrative user that you specify must be an account in the directory. The initial

administrative user is generally distinct from the user account on the Tableau Server computer that you use to run `tsm`.

However, these accounts can be the same if you have configured the Tableau Server identity store to use LDAP or Active Directory and the initial admin user is a member of the `tsmadmin` group on the Tableau Server computer.

Prerequisites

Before proceeding with the procedures in this topic, complete the following prerequisites as outlined in Install and Configure Tableau Server:

- Install and Initialize TSM
- Activate and Register Tableau Server
- Configure Initial Node Settings

Use web UI

After Tableau Server is finished initializing, the installation program will display a page to create the Tableau Server administrator.

- If you configured a local identity store during setup, then specify a name and password that you want to use.
- If you configured a LDAP or Active Directory identity store during setup, then you must specify a user account that is a member of the directory.
- The `username` value cannot include an at sign (@) unless the user name suffix matches Tableau Server's primary domain. For example, if Tableau Server connects to domain "myco.com", a user name of "user@example.com@myco.com" cannot be used.

If you are installing remotely, then you must sign in to TSM on the physical computer where Tableau Server is installing, or you can access the computer remotely and run the tabcmd initialuser command from a shell.

Use tabcmd CLI

You must create the initial administrative account for Tableau Server.

- If you configured a local identity store during setup, then specify a name and password that you want to use.
- If you configured a LDAP or Active Directory identity store during setup, then you must specify a user account that is a member of the directory.
- The `username` value cannot include an at sign (@) unless the user name suffix matches Tableau Server's primary domain. For example, if Tableau Server connects to domain "myco.com", a user name of "user@example.com@myco.com" cannot be used.

To create the initial user, run the following `tabcmd` command:

```
tabcmd initialuser --server http://localhost --username '<new-admin-username>'
```

For example:

```
tabcmd initialuser --server http://localhost --username 'tableau-admin'
```

If you are running the HTTP protocol on a port other than 80, specify the port after the host name, for example: `--server http://localhost:8080`.

After you run the command, the shell will prompt for an administrative password.

Next steps

After you have created the Tableau Server administrator account, continue your deployment by working through the configuration topics at Post Installation Tasks.

Important: You must install the PostgreSQL driver if you want to use the built-in administrative views. You can find driver links and installation instructions for all the supported connectors on the Driver Download page.

Validate Installation

To validate that Tableau Server is installed and running properly and to review the built-in administrative views, you must install the PostgreSQL driver.

Prerequisites

Before proceeding with the procedure in this topic, complete the following prerequisites as outlined in Install and Configure Tableau Server:

- Install and Initialize TSM
- Activate and Register Tableau Server
- Configure Initial Node Settings
- Add an Administrator Account

Install PostgresSQL driver and validate installation

To validate installation:

1. Download PostgreSQL drivers from the Driver Download page.
2. Copy the .jar file to this folder (you may have to create it manually): `/opt/tableau/tableau_driver/jdbc`.
3. Restart TSM:

   ```
   tsm restart
   ```

4. To validate that the drivers installed, navigate to the Administrative Views in Tableau Server.

Initial Node Installation Defaults

By default, the Tableau Server installer configures the number of process instances that Tableau Server runs based on the hardware detected by the installer. The default configuration applies to single-server installations and to the initial server of a multi-node installation.

You can calculate the default configuration based on the following rules for each process, where the "number of cores" refers to the number of physical processors:

Process Name	Number of Processes
VizQL Server	Equal to the number of cores divided by four, up to a maximum of four process instances.
Backgrounder	Set to two unless the number of cores is fewer than eight.
Cache Server	Set to two unless the number of cores is fewer than eight.
Data Server	Set to two unless the number of cores is fewer than eight.

For all other process types, the number of process instances is set to one, regardless of the hardware.

Here's an example default configuration for a computer with 16 cores:

Process Name	Number of Processes
VizQL Server	4
Application Server	1
Backgrounder	2
Cache Server	2

Process Name	Number of Processes
Data Server	2
File Store	1
Data Engine	1

Jump-start Installation

This topic provides all of the steps required to perform a basic, quick-start installation of Tableau Server using the command line. The purpose of the configuration described here is to provide the quickest, simplest path to a Tableau Server installation on a computer running the CentOS 7.3 (and higher) or Ubuntu distribution of Linux. Use this procedure as practice, to try out the Tableau Server installation, management, and user experience before your actual server deployment.

Important: Do not use this procedure as a stand-alone resource for deploying Tableau Server into a production environment. To deploy Tableau Server into a production environment, refer to the content at Install and Configure Tableau Server.

The procedures in this topic will install an instance of Tableau Server for Linux with the following characteristics:

- Operating system: Supported RHEL-like Linux distribution or Ubuntu. (As of July 2022, Tableau no longer supports Debian distributions. For more information, see this Tableau Community post.)
- Identity store: local authentication
- Gateway port: 80
- Tableau Server administrator account: admin

Before you begin

Review the topic, Before you install.... The procedure below assumes that you have installed Linux on conforming hardware and according to the environmental requirements specified in

that topic.

Note that the computer you install on must meet the minimum hardware requirements specified in Minimum Hardware Requirements and Recommendations for Tableau Server. The setup program will not install Tableau Server onto systems that do not meet the minimum hardware requirements.

Step 1: Install Tableau Server package and start Tableau Services Manager

Install Tableau Server with your distribution's package manager, then run a script to initialize Tableau Services Manager (TSM). Tableau Services Manager is a the management toolset used to install, configure, and manage Tableau services.

The initialize script is included with the installed package. For more details, see Install and Initialize TSM.

1. Log on as a user with `sudo` access to the computer where you want to install Tableau Server.
2. Navigate to the directory where you copied the Tableau Server installation package.
3. Use the package manager to install the Tableau Server package.

   Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

   - On RHEL-like distributions, including CentOS, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

     ```
     sudo yum update

     sudo yum install tableau-server-<version>.x86_64.rpm
     ```

> **Note:** When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately.

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. Navigate to the `scripts` directory:

```
cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
```

5. Run the following script to start TSM:

```
sudo ./initialize-tsm --accepteula
```

6. After initialization is complete, close the terminal session:

`exit` or `logout`

Step 2: Activate and register Tableau Server

Before you can configure Tableau Server you must activate a license and register. For more details, see Activate and Register Tableau Server.

1. Open a Bash session and sign in with the same account that you previously used to run `initialize-tsm`.

2. Activate the Tableau Server license. Run the following command:

```
tsm licenses activate -k <license_key>
```

3. Register Tableau Server. Generate a template that you can edit by running the following command:

   ```
   tsm register --template > /path/to/registration_file.json
   ```

4. Open a text editing program, fill in the registration file, save it, then pass it with the following command:

   ```
   tsm register --file /path/to/registration_file.json
   ```

Step 3: Configure local identity store

You must configure the identity store settings. This procedure simplifies installation setting identity store to local authentication. Workbook examples are installed by default. For more details on customizing these defaults, see Configure Initial Node Settings.

- Pass the configuration file with the following command:

  ```
  tsm settings import -f /opt/tableau/tableau_server-
  /packages/scripts.<version_code>/config.json
  ```

Step 4: Finalize installation

The final steps of installation are to apply changes, initialize and start TSM, and then to create the administration account. More details about these steps are at Configure Initial Node Settings.

1. Apply the configurations you made in the previous steps. Run the following command:

   ```
   tsm pending-changes apply
   ```

2. Initialize and start Tableau Server. Run the following command:

   ```
   tsm initialize --start-server --request-timeout 1800
   ```

3. Create the Tableau Server administrator account. Run the following command:

```
tabcmd initialuser --server 'localhost:80' --username 'admin' --password '<password>'
```

Where `'<password>'` is a strong password. Enclose the password and other arguments in single quotes.

Use the `admin` account that you created to access the Tableau Server admin web pages. See Sign in to the Tableau Server Admin Area.

Step 5: Install PostgreSQL drivers

To validate that Tableau Server is installed and running properly and to review the built-in administrative views, you must install the PostgreSQL driver.

1. Download PostgreSQL drivers from the Driver Download page.
2. Copy the .jar file to this folder (you may have to create it manually): `/opt/tableau/tableau_driver/jdbc`.
3. Restart TSM:

   ```
   tsm restart
   ```

4. To validate that the drivers installed, navigate to the Administrative Views in Tableau Server.

Configure Local Firewall

This topic explains how to configure the firewall on the computer running Tableau Server.

A local firewall should be enabled on the operating system to protect Tableau Server in single and multi-node deployments. In a distributed (multi-node) installation of Tableau Server, communication between nodes does not use secure communication. Therefore, you should enable firewalls on the computers that host Tableau Server.

We recommend that you configure the firewall so that only two ports are accessible to external traffic: the `gateway` port and the `tabadmincontroller` port. By default, these are

ports 80 and 8850 respectively. Additionally, if you are running in a distributed deployment, then you will need to open the port range, 27000-27009, so licensing can communicate across nodes.

The `gateway` port is used for HTTP connection to Tableau Server. We recommend that you use SSL for the `gateway` port. If you will be using SSL, then the port must be `443` because Tableau Server does not support other ports for SSL. The procedures below describe how to configure the firewall for the `gateway` port. Configure the Tableau Server gateway (Configure Initial Node Settings) to match the port you set here.

The examples below describe how to configure the firewall on single and multi-node deployments of Tableau Server running on RHEL/CentOS distributions. The examples use Firewalld, which is the default firewall on CentOS.

Single-node configuration

1. Open a bash shell and run the following TSM command to retrieve the port number for the `tabadmincontroller` port:

   ```
   tsm topology list-ports
   ```

   Make a note of the `tabadmincontroller` port. By default, this port is `8850`.

2. Start firewalld:

   ```
   sudo systemctl start firewalld
   ```

3. Verify that the default zone is a high-security zone, such as `public`. If it is not, we recommend changing it to a high-security zone.

   ```
   sudo firewall-cmd --get-default-zone
   ```

   ```
   sudo firewall-cmd --set-default-zone=public
   ```

4. Add ports for the `gateway` port and the `tabadmincontroller` port. In the example below we use the default ports (`80` and `8850`).

```
sudo firewall-cmd --permanent --add-port=80/tcp

sudo firewall-cmd --permanent --add-port=8850/tcp
```

5. Reload the firewall and verify the settings.

```
sudo firewall-cmd --reload

sudo firewall-cmd --list-all
```

Multi-node cluster configuration

In addition to enabling ports, configuring the firewall on a multi-node cluster requires additional steps to ensure that nodes can communicate with each other.

Before you begin

You will need the IP address for each node in the cluster. The example here uses `<node1IP>` as a placeholder for the initial node IP address, and `<node2IP>` and `<node3IP>` as placeholders for the IP addresses of two additional nodes.

Step 1: Configure initial node.

1. Open a bash shell and run the following TSM command to retrieve the port number for the `tabadmincontroller` port:

   ```
   tsm topology list-ports
   ```

   Make a note of the `tabadmincontroller` port. By default, this port is `8850`.

2. Run the following commands to determine the range of port numbers that TSM may dynamically select. You will specify this range later in this procedure. Make a note of the port range.

   ```
   tsm configuration get -k ports.range.min

   tsm configuration get -k ports.range.max
   ```

A typical range is `8000` to `9000`.

3. Start firewalld:

```
sudo systemctl start firewalld
```

4. Verify that the default zone is a high-security zone, such as `public`. If it is not, we recommend changing it to a high-security zone.

```
firewall-cmd --get-default-zone
```

```
sudo firewall-cmd --set-default-zone=public
```

5. Add ports for the `gateway` port and the `tabadmincontroller` port. In the example below we use the default ports (`80` and `8850`). You must also add a port range (`27000-27010`) to enable licensing communication between nodes.

```
sudo firewall-cmd --permanent --add-port=80/tcp
```

```
sudo firewall-cmd --permanent --add-port=8850/tcp
```

```
sudo firewall-cmd --permanent --add-port=27000-27010/tcp
```

6. Configure the firewall to allow all traffic from the other nodes in the cluster. For the ports option, specify the range you noted in Step 2. Run the command for each of the additional nodes in your cluster. For example:

```
sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4
source address=<node2IP>/32 port port=8000-9000 protocol=tcp
accept'
```

```
sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4
source address=<node3IP>/32 port port=8000-9000 protocol=tcp
accept'
```

7. Reload the firewall and verify the settings.

```
sudo firewall-cmd --reload

firewall-cmd --list-all
```

Step 2: Configure additional nodes

Each node in the cluster must be able to communicate with the initial node and with the other nodes.

Run this procedure on each additional node in the cluster. In this example, the node at IP address, `<node2IP>`, communicates with the initial node at `<node1IP>` and a third node at `<node3IP>`.

1. Start firewalld:

   ```
   sudo systemctl start firewalld
   ```

2. Verify that the default zone is a high-security zone, such as `public`. If it is not, we recommend changing it to a high-security zone.

   ```
   firewall-cmd --get-default-zone

   sudo firewall-cmd --set-default-zone=public
   ```

3. Configure the firewall to allow `gateway` and `tabadmincontroller` access from the other nodes in the cluster. For example:

   ```
   sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4
   source address=<node1IP>/32 port port=80 protocol=tcp accept'

   sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4
   source address=<node1IP>/32 port port=8000-9000 protocol=tcp
   accept'

   sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4
   source address=<node3IP>/32 port port=80 protocol=tcp accept'
   ```

```
sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4
source address=<node3IP>/32 port port=8000-9000 protocol=tcp
accept'
```

In this example, since the `tabadmincontroller` port (`8850`) is included in the port range, it is not explicitly specified in a command.

4. Reload the firewall and verify the settings.

```
sudo firewall-cmd --reload
```

```
firewall-cmd --list-all
```

Automated Installation of Tableau Server

Tableau provides an `automated-installer` script to automate an install of Tableau Server. The script is community supported. You can download the script and use it as written, or modify it for your specific needs.

Benefits of using the automated installer

- With a single command, you can install, configure and get to a working instance of Tableau Server.
- The command can be run without user input making it suitable for automation.
- The configuration can be set once and used for all your installations, making this a repeatable process.

When not to use the automated installer:

- If you are installing for the first time, we recommend that you manually test the installation before automating the process. Any issues that block installation are easier to resolve interactively, and after you have resolved these issues, you can use the automated installer.

- If you are testing or trying new configuration parameters such as authentication methods, we recommend that you manually run the installation first. TSM validates configuration entities and rejects configuration parameters that are not valid. Once you have the correct parameters identified, you can use the automated installer.
- If you are unable to or do not want to enter passwords into the secrets file, using the automated installer might not be an option for you.

Before you begin

Review the Before you install... topic to make sure you have installed Linux on a computer that meets the operating system requirements and the minimum hardware requirements for Tableau Server.

Note: If you are installing Tableau Server in a production environment, review the minimum hardware **recommendations**. The recommendations represent the minimum hardware configuration you should use for a production installation of Tableau Server.

To perform an automated installation, you have to use the automated installer package, which uses the Tableau Server install package as an input. We recommend that you download **both of these packages** before you begin as follows:

1. Download both the automated installer package and the Tableau Server installer package:
    a. Download the **automated installer package** from GitHub for the distribution you are using. The automated installer packages can be found in the packages sub directory.

    > **Note:**. The version of automated installer package you use must match the version of the Tableau Server installer package. For example, use 10.5.0 version of the automated installer package with 10.5.0 version of the Tableau Server installer package.

b. Select and download the appropriate **Tableau Server installer package** from the Tableau Server Product Downloads page. The one you choose depends on which Linux distribution you are using. For example, for RHEL like systems, `tableau-server-<version>.x86_64.rpm`.

c. Download the `config.json`, `reg_templ.json`, and the secrets templates.

2. Copy the packages and templates to a location on or accessible from the computer where you are going to install Tableau Server.

How to use the automated installer

The automated installer installs the Tableau Server installer package, creates the directories, sets the permissions required to run Tableau Server, and starts the Tableau Services Manager (TSM) setup. After the TSM setup is completed, the automated installer runs `tsm` commands to install, configure, and start Tableau Server. By default, during installation, the automated installer activates a trial license. If you have an actual product key, you can provide the product key at the command line or activate the product key after you run the script. Most of the command line options in the automated installer are the same as the options used by the tsm initialize command.

To run the automated installer without user input, you must provide the following required command line options:

Option	Description
-s <secrets-file>	The name of the secrets file. The secrets file should have the user names and passwords for TSM administrator and the Tableau Server administrator accounts. **Note:** Providing the password in the secrets file is optional. However, if passwords are not found in the secrets file, you will be prompted to provide them during installation.

	The automated installer package includes the template for the secrets file.
-f <config-file>	The name of the configuration JSON file. The automated installer package includes the template for the configuration file.
-r <registration-file>	The name of the registration file. The automated installer package includes the template for the registration file.
--accepteula	Indicates that you have accepted the End User License Agreement.
<package-file>	The rpm or deb Tableau Server installer.

Use the -h option to see a full list of all the required command line options.

Configure Tableau Server for a forward proxy

If your organization uses a forward proxy solution to access the internet, then configure Tableau Server to use the proxy server. Tableau Server must access the internet for map data and for default licensing functionality.

We recommend configuring Tableau Server for a forward proxy solution during installation.

To configure proxy server during unattended installation, include the `--http_proxy` and/or `--https_proxy` flags to specify the forward proxy server.

Specify the URL with the port, for example:

```
--http_proxy=http://proxy.exampe.lan:80/ --https_proxy-
y=http://1.2.3.4:443/
```

Take care to use `http` when you specify the URL for the `https_proxy` variable. Do not specify the `https` protocol for the value of the `https_proxy` variable.

To configure Tableau Server to bypass the forward proxy, include the `--no_proxy` flag. You should also add exceptions to this proxy configuration to guarantee that all communications

within a local Tableau Server cluster (if you have one now or will have one later) do not route to the proxy server.For example:

```
--no_proxy=localhost,127.0.0.1,localaddress,.localdomain.com.
```

If you do not configure the forward proxy during installation, then refer to Configuring Tableau Server on Linux to work with a forward proxy, after you have installed.

Perform an unattended installation

Step 1: Install the automated installer

1. Log onto the computer as a user with sudo access.
2. Use the package manager to install the script package:

    - ON RHEL-like distributions, including CentOS, run the following command:

      ```
      sudo yum install /path/to/tableau-server-automated-installer-<version>.noarch.rpm
      ```

    - On Ubuntu, run the following commands:

      ```
      sudo apt-get update

      sudo apt-get -y install gdebi-core

      sudo gdebi -n /path/to/tableau-server-automated-installer-<version>.deb
      ```

    The automated installer package download includes templates for the configuration file (config.json), registration file (reg_templ.json) and the secrets (secrets) file that you can use to modify for your requirements as described in the next step. The installer script, and the templates for the initial node configuration, Tableau Server registration, and secrets file are installed to:

```
/opt/tableau/tableau_server_automated_installer/automated-
installer.<version>
```

Step 2: Create files to provide additional configuration information required to run the automated install

Since the automated installer is meant to run without user interaction, you must provide the following additional information:

1. Run the following command to copy the templates, config.json, reg_templ.json, and secrets, to another directory like your home directory. We don't recommend that you edit the template files directly:

   ```
   cp /opt/tableau/tableau_server_automated_installer/automated-
   installer.<version>/{config.json,reg_templ.json,secrets} ~
   ```

2. Edit the configuration template, **config.json**, to provide the initial node configuration settings. You must provide identity store settings for the Tableau Server computer. Depending on your network requirements, you may need to also provide the gateway settings. The caching option is set to cache and reuse data for as long as possible. Sample workbooks are installed by default. The template includes the minimum required information, so the template is a starting point. For more information on configuration settings, see Configure Initial Node Settings.

3. Edit the registration file **reg_templ.json** to provide your unique identifying information needed to register Tableau Server in accordance with the End User License Agreement (EULA). For more information, see End User License Agreement and Activate and Register Tableau Server.

4. Edit the secrets file using the **secrets** template with the user name and password for the TSM administrator and Tableau Server administrator accounts.

   - The TSM administrator account should be the same user as the sudo admin running the script. If you do not want to specify the password in the secrets file, you

can leave it blank, and you will be prompted to provide the password during installation.

- The Tableau Server administrator account is the initial account that is created by the installer and is used to administer Tableau Server.

Step 3: Run the automated install

1. Log onto the computer as a user with sudo access.

   ON RHEL-like distributions, including CentOS, run the following command:

   ```
   sudo /opt/tableau/tableau_server_automated_installer/automated-installer.<version>/automated-installer -s /path/to/secrets -f /path/to/config.json -r /path/to/reg_templ.json --accepteula /path/to/tableau-server-<version>.x86_64.rpm
   ```

   On Ubuntu, run the following command:

   ```
   sudo /opt/tableau/tableau_server_automated_installer/automated-installer.<version>/automated-installer -s /path/to/secrets -f /path/to/config.json -r /path/to/reg_templ.json --accepteula /path/to/tableau-server-<version>_amd64.deb
   ```

   Important: You must specify `-accepteula` key to acknowledge and accept the end user license agreement (EULA) in the command that you use the run the script. The EULA is available in the following location: End User License Agreement.

   Note: If you are adding this machine as an additional node to an existing cluster, you must specify the `-b bootstrap` flag and the node configuration file from the initial server. For more information on how to generate the node configuration file, see Install and Configure Additional Nodes

Install Tableau Server in a Disconnected (Air-Gapped) Environment

You can install Tableau Server in a disconnected environment that has no outside network access of any kind. Such disconnected environments, commonly referred to as air-gapped, are used when high security is needed to prevent data breaches or to guard against hacking. Air-gapped environments have no internet access, no outside network access, no outside wireless access, etc. The only means of getting software and data into or out of an air-gapped environment is by using removable media such as USB sticks or writable optical CDs or DVDs.

Installing Tableau Server in an air-gapped environment is an advanced task for IT administrators who are familiar with the security considerations, best practices, and pitfalls of installing software in air-gapped environments.

The following Tableau Server features will be unavailable or will have reduced functionality in an air-gapped environment:

- **Maps** – Tableau Server uses externally hosted map data by default. Beginning with version 2020.4.0, you can configure Tableau Server to use offline maps. With earlier versions of Tableau, maps are unavailable in an air-gapped environment unless you also install a map server in your air-gapped environment. For more details, see Displaying Maps in an Air-Gapped Environment.
- **Licensing** – Tableau Server needs to connect to the internet in order to activate product keys. However, you can manually activate the product keys.
- **External data** – Any data located outside your air-gapped environment is unavailable.

Prerequisites

In order to install Tableau Server in an air-gapped environment, you'll need the following:

- Trusted computer with limited access to the internet that you can use to download the installation packages and resources required by Tableau Server. A trusted computer has been scanned and cleared of any viruses and malware.
- Trusted removable media that you can use to transfer the downloaded software to your air-gapped environment. Trusted removable media is removable media that is new and previously unused and comes from a reputable or known source. Trusted removable media has been scanned and verified that it does not contain any viruses or malware.
- Air-gapped environment with computers and storage that meet the requirements for installing Tableau Server.

Installing Tableau Server on an Air-Gapped Computer Running Linux

The easiest way to install Tableau Server on a computer in an air-gapped environment is to do so before the computer is placed into the air-gapped environment. If that's not possible you'll need to download the required packages to a trusted computer outside the air gap:

1. On a trusted computer outside the air gap with internet access, download the Tableau Server installation package.
2. Extract the list of dependent packages:

   On Ubuntu:

   `dpkg --field <debfile> Depends` (where `<debfile>` is the name of the .deb package you downloaded from Tableau).

   Example command:

   ```
   dpkg --field tableau-server-linux-1.deb Depends
   ```

Example output:

`ca-certificates, fontconfig, net-tools, bash-completion, ca-certificates-java, freeglut3, libegl1-mesa, libfreetype6, libgssapi-krb5-2, libxcomposite1, libxrender1, libxslt1.1, lsb-core`

On RHEL and RHEL-like Linux distributions:

`yum -q deplist <RPM file>` (where `<RPM file>` is the .rpm package you downloaded from Tableau).

Example command:

```
yum -q deplist tableau-server-linux_1.rpm
```

Example output:

```
package: tableau-server-10400.17.0703.1600.x86_64 10400-17.0703.1600
dependency: /bin/sh
provider: bash.x86_64 4.2.46-21.el7_3
dependency: bash-completion
provider: bash-completion.noarch 1:2.1-6.el7
dependency: ca-certificates
provider: ca-certificates.noarch 2017.2.14-70.1.el7_3
dependency: fontconfig
provider: fontconfig.x86_64 2.10.95-10.el7
provider: fontconfig.i686 2.10.95-10.el7
dependency: freeglut
provider: freeglut.x86_64 2.8.1-3.el7
provider: freeglut.i686 2.8.1-3.el7
dependency: freetype
provider: freetype.x86_64 2.4.11-12.el7
provider: freetype.i686 2.4.11-12.el7
dependency: krb5-libs
provider: krb5-libs.x86_64 1.14.1-27.el7_3
```

```
provider: krb5-libs.i686 1.14.1-27.el7_3
dependency: libXcomposite
provider: libXcomposite.x86_64 0.4.4-4.1.el7
provider: libXcomposite.i686 0.4.4-4.1.el7
dependency: libXrender
provider: libXrender.x86_64 0.9.8-2.1.el7
provider: libXrender.i686 0.9.8-2.1.el7
dependency: libxslt
provider: libxslt.x86_64 1.1.28-5.el7
provider: libxslt.i686 1.1.28-5.el7
dependency: mesa-libEGL
provider: mesa-libEGL.x86_64 11.2.2-2.20160614.el7
provider: mesa-libEGL.i686 11.2.2-2.20160614.el7
dependency: net-tools
provider: net-tools.x86_64 2.0-0.17.20131004git.el7
dependency: redhat-lsb-core
provider: redhat-lsb-core.x86_64 4.1-27.el7.centos.1
provider: redhat-lsb-core.i686 4.1-27.el7.centos.1
```

3. Download each of the dependent packages:

   On Ubuntu:

   ```
   apt-get download <package1> <package2>…
   ```

   On RHEL and RHEL-like Linux distributions:

   ```
   yumdownloader <package1> <package2>…
   ```

4. Transfer the packages to your removable media.

5. On your air-gapped computer, insert the removable media containing the Tableau Server installation package and dependent packages, and then run the installer.

6. After installation is complete, you can activate the Tableau Server product keys. For more information, see Activating Tableau Server in an Air-Gapped Environment.

Activating Tableau Server in an Air-Gapped Environment

Because an air-gapped computer is not connected to the internet, you'll need to perform the Tableau Server activation process manually.

Offline activation overview

Offline activation of Tableau Server involves the following steps:

1. Generate an offline activation request file.
2. Copy the offline activation request file to a computer with internet access.
3. Upload the offline activation request file to the Tableau activation website.
4. Download the resulting offline activation response file from the website. You'll use this file to activate Tableau Server

Offline activation file name changes

Beginning in Tableau Server version 2023.1, the Tableau licensing system supports two underlying licensing technologies. From an administrative perspective, the only configuration difference between the two systems is the file types that are generated and consumed for offline activation. The licensing technology is determined during the initial installation of Tableau Server, and cannot be changed after install.

We refer to the legacy (and still supported) version of licensing technology as FlexNet. The latest version of the technology is referred to as Server ATR. For more information, see Activate Tableau Server Using the Authorization-To-Run (ATR) Service. The following table describes the file naming nomenclature for each technology. The table also includes the generic reference.

Generic file name	Server ATR file names	FlexNet file names
Off-	Off-	Tableau-

lineActivationRequest	lineActivationRequestFile_ yyyyMMdd.hhmmss.json	OfflineActivationRequest.tlq
Off- lineActivationResponse	Off- lineActiv- ationLicensingAtrs.zip	activation.tlf

Note: Since this documentation supports multiple versions of Tableau Server, we will use the generic file name references (OfflineActivationRequest and Off-lineActivationResponse) for the rest of this topic. You can identify the licensing technology your Tableau Server installation uses according to the file type that generated in the steps that follow.

Step 1. Log in to Tableau Services Manager

- To log in to Tableau Services Manager (TSM), run the following command:

```
tsm login -u <username>
```

What if I can't log in?

The account that you use to configure the rest of the installation must be a member of the `tsmadmin` group that was created during initialization. To view the user accounts in the `tsmadmin` group, run the following command:

```
grep tsmadmin /etc/group
```

If the user account is not in the group, run the following command to add the user to the `tsmad-min` group:

```
sudo usermod -G tsmadmin -a <username>
```

After you have added the user to the `tsmadmin` group, run the `tsm login` command.

Step 2. Determine your Tableau Server licensing type

How you activate Tableau Server will be different depending on which licensing type you are running. Run the following command to determine the licensing type your Tableau Server deployment is configured with:

```
tsm configuration get -k serverauthorizationtorun.enabled
```

If this command returns `true`, then your deployment is configured with Server ATR licensing type.

If this command returns `false`, then your deployment is configured with legacy licensing type.

Go to the step 3 that matches your licensing type.

Step 3 (Server ATR licensing type) Generate and copy json content to Activation page.

Follow these steps if your Tableau Server deployment is configured with Server ATR licensing type. If your server is configured with the legacy licensing type, skip to the following section.

1. On your Tableau Server in the air-gapped environment, use TSM to obtain the offline activation file. At a command prompt:

   ```
   tsm licenses get-offline-activation-file -k <product-key> -o
   <target-directory>
   ```

   The `<target-directory>` must exist. You can obtain your product key in the Tableau Customer Portal.

2. Copy the JSON file contents.

3. From the internet connected computer, navigate to the Tableau Offline Activation website, select **Option B - Manually Enter Information from Activation File**, copy the

JSON contents into the requested fields, and then click **Submit**.

4. The website should say `The activation was successful. Please click here to download your activation file.`

   Download the OfflineActivationResponse file from Tableau, and proceed to step 4.

Step 3 (Legacy licensing type) Transcribe data from your air-gapped system into an activation request template.

Follow these steps if your Tableau Server deployment is configured with the legacy licensing type. If your server is configured with the Server ATR licensing type, run the above procedure.

1. On your Tableau Server in the air-gapped environment, use TSM to obtain the offline activation file. At a command prompt:

   ```
   tsm licenses get-offline-activation-file -k <product-key> -o <target-directory>
   ```

   The `<target-directory>` must exist. You can obtain your product key in the Tableau Customer Portal.

2. If you can copy the offline request file (OfflineActivationRequest) from the target directory to a computer that has Internet access, skip to step 5.

   Otherwise, if you cannot copy the file to another computer due to security reasons, continue with step 3.

3. Download and open the `server_linux.tlq` file in an XML text editor such as Notepad++ on a trusted computer that has Internet access.

   You'll need to write down the values listed in step 4 from the air-gapped computer in order to copy them to the offline template (`server_linux.tlq`).

4. Update the following XML elements in the appropriate `server_linux.tlq` file with the values for the same elements listed below from the air-gapped computer.

All the Machine / Hash values in the .tlq files are Hex values. The only valid characters are 0 - 9 and A - F. Use all caps for any letters.

Do not add any additional spaces or carriage controls and only modify the “X” characters found in the template. The format of the file must not change.

```
Line 2 - <EntitlementId>

Line 5 - <ClientVersion>

Line 5 - <RevisionType> (This value is present in the server_
linux.tlq file.)

Line 5 - <MachineIdentifier> (This value is present in the
server_linux.tlq file.)

Line 11 - <Value> (If the value is not present, remove the "X"
place holder, leaving <Value></Value>.)

Line 12 - <Value> (If the value is not present, remove the "X"
place holder, leaving <Value></Value>.)

Line 13 - <Value> (If the value is not present, remove the "X"
place holder, leaving <Value></Value>.)

Line 15 - <SequenceNumber>

Line 61 - <Hash>
```

5. Upload the offline request file (OfflineActivationRequest) or edited template file (server_linux.tlq) to the Tableau Offline Activation website.

6. The website should say `The activation was successful. Please click here to download your activation file.`

   Download the OfflineActivationResponse file and transfer it to your Tableau Server. Proceed to step 4.

Step 4. Initialize or activate your license

1. Move the OfflineActivationResponse file to your air-gapped computer using trusted removable media.
2. Run the following command:

   ```
   tsm licenses activate -f <path-and-activation-file>
   ```

   You should see the message "`Activation successful.`", which indicates that Tableau Server is activated.

Displaying Maps in an Air-Gapped Environment

In an air-gapped environment, maps in Tableau Server will be unavailable by default due to the lack of internet access. Instead, you can configure Tableau to use local maps in an air-gapped environment using the steps below.

Configuring Tableau Server to use offline maps:

1. Open a command prompt as administrator.
2. Configure Tableau to use locally available offline maps:

   ```
   tsm configuration set -k vizqlserver.force_maps_to_offline -v true
   ```

   ```
   tsm pending-changes apply
   ```

Clone Tableau Server

Beginning with version 2022.3 of Tableau Server, a new tsm command enables you to create a copy of the configuration and topology of a Tableau Server deployment and use that to create an exact replica of the original deployment. The command, `tsm settings clone`, creates a set of files (the "clone payload") includes settings, secrets, configuration, and topology, including ports being used, and information about external services.

Security considerations

The `tsm settings clone` command generates a set of files (the "clone payload") that contain all the secrets generated by Tableau Server, as well as those provided by the server administrator during configuration, including a keystore containing keypairs and certificates. It is extremely important for security that you keep the clone payload and the output location secure. To facilitate this:

- If the output directory does not exist, tsm will create it with access restricted to the user running the `tsm settings clone` command.
- If the output directory exists when you run the command, tsm will confirm that it is owned by the user running the command, and that permissions are limited to only that user. If the directory does not have the expected permissions, a message displays:

  ```
  The output directory '<path/to/directory>' exists, but must be restricted to owner only.
  ```

Limitations of the clone payload

- The version of Tableau Server being cloned must match the version being created. You cannot install a newer version of Server using a clone payload from an older version.

Using the clone command to create a copy of Tableau Server

Creating a cloned copy of Tableau Server is a multi-step process, with the two high level steps being:

1. Create a clone payload from the Tableau Server installation you want to duplicate.
2. Use the clone payload to install a second deployment of Tableau Server. You are responsible for matching the topology of the two installations, adding additional nodes to match the original deployment.

Creating the clone payload

To generate a set of files (clone payload) that contain the configuration and topology settings for Tableau Server, use the `tsm settings clone` command. The command takes a single argument, the output directory where the set of files should be saved:

```
tsm settings clone --output-directory <output_directory>
```

Once the clone payload is created, you can use this when installing a new instance of Tableau Server with the identical configuration and topology of the source installation.

Using the clone payload to create a copy of Tableau Server

1. Install the Tableau Server package on the initial node. Do not initialize Tableau Server. Install the Tableau Server package
2. Run the `initialize-tsm` script and specify the path to the clone payload created by the `tsm settings clone` command:

   ```
   sudo /opt/tableau/tableau_server-
   /packages/scripts.<version>/initialize-tsm --accepteula --
   clone-artifact-dir=<path-to-clone-directory>
   ```

   Note: Tableau Server runs as unprivileged tableau user therefore administrator must ensure that tableau user will have read access to the clone directories and files. This in most cases requires granting read permissions on clone directory content as well as execute permission (for traversal) on all parent directories to "others" .

3. (Optional) Install additional nodes to match the number of nodes on the original (cloned) installation of Tableau Server. If the original installation had additional nodes, this step is required. Only install additional nodes to match the number on the original installation.
   a. Generate the bootstrap file on the initial node:
   b. Install Tableau Server on each additional node and run the initialize-tsm script:

4. On the initial node, complete the initialization:

```
tsm initialize
```

Container

Clone functionality is convenient when using Tableau Server in a Container, especially when used with external repository and storage. The clone command allows the administrator to quickly recreate a Tableau Server environment and reattach the external services.

> **Note:** The steps to recreate Tableau Server cluster may differ depending what container technologies are being used (for example: docker, docker compose, or Kubernetes).

The below information is specific to using docker directly. When starting up the initial node container, you need to specify the location of the clone payload using the `CLONE_ARTIFACT_DIR` environment variable.

For example:

```
docker run \
-v <path-to-clone-directory>:/docker/custom-clone-path \
-e CLONE_ARTIFACT_DIR=/docker/custom-clone-path \
-e LICENSE_KEY=<key> \
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Recreating a multi-node deployment of Tableau Server in a Container

If your clone payload came from a multi-node Tableau Server deployment your initial container will wait for additional nodes to join the cluster.

Setting up additional nodes in the cluster when using clone is same as adding nodes in a regular deployment and requires:

- Mounting volumes to share the bootstrap file between the nodes
- Specifying `ALWAYS_WRITE_BOOTSTRAP_FILE` and `BOOTSTRAP_INSTALL` environment variables

Detailed steps to create a multi-node environment in a container deployment can be found here: Multi-node Tableau Server in a Container.

Note: The clone payload only needs to be mounted and used on the initial node in the cluster. You do not need to mount the clone payload on additional nodes.

Tableau Server in a Container

Introduction

Tableau Server in a Container is Tableau's first container-based server offering. Tableau Server in a Container is an all-in-one Tableau Server instance running inside of a Linux Docker container. In other words, a Tableau Server in a Container image is a docker image that runs an entire self-contained Tableau Server application. Tableau Server in a Container is our first of many steps to support Tableau Server running in container-based environments.

The easiest way to understand the concept of Tableau Server in a Container is to think of it like a VM with Tableau Server pre-installed. The image is based on a UBI 8 image (CentOS 7 for version 2022.1 and earlier) and runs `supervisord` (instead of `systemd`) inside the container. When the container starts `supervisord`, it will immediately attempt to initialize and start Tableau Server. Much of the documentation here aims to describe how to provide configuration and leverage automation so you can run Tableau Server in Docker environments.

The Tableau Server in a Container Image Setup Tool helps you create and customize container images to include custom packages and artifacts. One of the primary functions of the tool is to build the container image and install custom data connectors.

Limitations for Tableau Server in a Container

- Tableau Server in a Container only supports license activation using Server ATR. Off-line activation using Server ATR is supported in 2023.1 and later. This functionality is available in Containers but requires extra steps and approval. If you need to run Tableau Server in a Container in an air-gapped or offline environment, contact your Account representative for more information.
- Tableau Server in a Container does not currently support the Resource Monitoring Tool (RMT) agent.
- Kerberos is not supported in Tableau Server in a Container.

To test out the Tableau Server in a Container Image quickly in a proof-of-concept scenarios, see Tableau Server in a Container - Quick Start.

Basic Workflow for Tableau Server in a Container

Here is the basic workflow for using Tableau Server in a Container. You can find detailed instructions for each step in the links.

1. Use the Setup Tool to create a custom image of Tableau Server in a Container. See Tableau Server in a Container - Using the Setup Tool.
2. Run the image you created to start and use Tableau Server in a Container. See Tableau Server in a Container - Using an Image.

Tableau Server Feature Considerations

Some Tableau Server features works differently in containers. This section covers specific features that have special or different considerations in a container environment.

Active Directory

Set AD Domain Controller

If you plan on using Active Directory as an Identity Store for Tableau Server web pages and sites, there is an additional consideration to account for. Tableau Servers running in Linux environments dynamically determine which AD Domain Controller to communicate with by

examining their IP subnet. Containers can be assigned arbitrary IP addresses, and in this case Tableau Server will not necessarily be able to use its IP address to find an appropriate domain controller. For this reason it may be necessary to configure a specific domain controller / hostname for Tableau Server to communicate with. To do this follow these steps:

1. Determine which domain controller you want Tableau Server to use and get the hostname.
2. Set the configuration key `wgserver.domain.ldap.hostname` to the hostname using the standard Tableau Server Admin configuration options:
    - Set the value in the json configuration file `CONFIG_FILE`.
    - Use the TSM configuration command

      ```
      tsm configuration set -k wgserver.domain.ldap.hostname -v
      <hostname>
      ```

Import AD certificate to Tableau Server Keystore

By default Tableau Server in a container communicates with AD via StartTLS whenever simple bind is used. So when the container is run in this configuration, it is necessary to import the AD server certificate to the Tableau Server Keystore, otherwise server initialization will fail. To do this follow these steps:

1. Create a `pre-init-command` script (check Pre-initialization script section). Add the following line to add the AD certificate to tableau server keystore.

   ```
   ${INSTALL_DIR}/packages/repository.${SERVICE_
   VERSION}/jre/bin/keytool -importcert -noprompt -alias
   startTlsCert -file <mounted-certificate-path> -storetype JKS -
   storepass changeit -keystore ${DATA_DIR}/-
   config/tableauservicesmanagerca.jks
   ```

2. Mount the AD server certificate at the filepath provided for `-file` parameter in the `pre-init-command` script.

Alternatively, the default setting to communicating with AD via StartTLS can be disabled. Set `wgserver.domain.ldap.starttls.enabled` to `false` to disable the StartTLS. But it is not recommended.

Note: For an example configuration entity with AD, see LDAP - Active Directory.

Deployment Configuration Examples

Docker

Tableau Server in a Container Basic Usage

```
docker run \
-e LICENSE_KEY=<key>
-p 8080:8080
-d <Tableau Server in a Container image ID or tag>
```

Tableau Server in a Container Basic Usage with Automated Initial Admin User

```
docker run \
-e LICENSE_KEY=<key> \
-e TABLEAU_USERNAME=<myadmin> \
-e TABLEAU_PASSWORD_FILE=/etc/tableau-admin-secret \
-v <full-path-to-pw-file>:/etc/tableau-admin-secret \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

TSM only mode

```
docker run \
-e TSM_ONLY=1 \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Multi-Node Basic Usage

Initial Node

Option 1: Use this if the server configuration (`CONFIG_FILE`) specifies a multi-node topology:

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-v <full-path-to-config-file>:/docker/config/config.json:ro \
-e LICENSE_KEY=<key> \
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Option 2: Use this if you want a multi-node deployment even if server configuration does not specify multi-node topology:

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-e LICENSE_KEY=<key> -e ALWAYS_WRITE_BOOTSTRAP_FILE=1 \
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Additional node

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-e BOOTSTRAP_INSTALL=1 \
-p 8080:8080 -p 8800-9000:8800-9000 \
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Externalize Data Usage

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
-e LICENSE_KEY=<key> \
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Init-Container Basic Usage

Init Container

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
-e LICENSE_KEY=<key> \
-e INIT_CONTAINER=1 \
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Run Container

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Basic Restore from Backup Single-Node

```
docker run \
-v <full-path-to-backup-file>:/docker/config/backup/backup-file.ts-
bak \
-v <full-path-to-config-only-file>:/docker/config/config.json:ro \
-e LICENSE_KEY=<key> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Docker-Compose

```
version: '3.2'
services:
    tableau-server:
         hostname: localhost
         volumes:
              - <your-tsm-command-file>:/docker/config/tsm-com-
mands:ro
              - <your-config-file >:/docker/config/config.json:ro
         ports:
              - "8080:8080"
         image: ${IMAGE_NAME}
         environment:
              - LICENSE_KEY=<license-key>
```

Tableau Server in a Container - Using the Setup Tool

Introduction

Tableau Server in a Container is Tableau's first container-based server offering. Tableau Server in a Container is an all-in-one Tableau Server instance running inside of a Linux Docker container. In other words, a Tableau Server in a Container image is a docker image that runs an entire self-contained Tableau Server application. Tableau Server in a Container is our first of many steps to support Tableau Server running in container-based environments. The easiest way to understand the concept of Tableau Server in a Container is to think of it like a VM with Tableau Server pre-installed. The image is based on a UBI 8 image (CentOS 7.x for version 2022.1 and earlier) and runs `supervisord` (instead of `systemd`) inside the container. When the container starts `supervisord`, it will immediately attempt to initialize and start Tableau Server. Much of the documentation here aims to describe how to provide configuration and leverage automation so you can run Tableau Server in Docker environments.

The Tableau Server in a Container Image Setup Tool helps you create and customize container images to include custom packages and artifacts. One of the primary functions of the tool is to build the container image and install custom data connectors.

Limitations for Tableau Server in a Container

- Tableau Server in a Container only supports license activation using Server ATR. Offline activation using Server ATR is supported in 2023.1 and later. This functionality is available in Containers but requires extra steps and approval. If you need to run Tableau Server in a Container in an air-gapped or offline environment, contact your Account representative for more information.
- Tableau Server in a Container does not currently support the Resource Monitoring Tool (RMT) agent.
- Kerberos is not supported in Tableau Server in a Container.

Tableau Server in a Container Setup Tool

The Tableau Server in a Container Setup Tool, `build-image`, builds a custom Tableau Server in a Container image from a Tableau `.rpm` Installer and the provided configuration files.

The Setup tool takes a Tableau Server installer and your drivers and other artifacts as input and creates a Docker image. When the `build-image` tool is used properly the newly generated image will have installed the desired artifacts.

Supported distributions for building

Building the Tableau Server in a Container Docker image is only supported on a RHEL-based Linux system (RHEL, CentOS, or Amazon Linux 2). Building on any other Linux distributions may be possible but is currently untested and unsupported. Building images on macOS is not supported. The image created is based on a UBI 8 image (CentOS 7.x for version 2022.1 and earlier).

You must have Docker version 18.09 or later installed on the host in order to build the container images. In general, we recommend using the latest stable version of Docker. Some Linux distros only provide older versions of Docker in their software repositories, in which case you may need to install Docker from a different source. Docker versions earlier than version 18.09 do not include features that are required for Tableau Server in a Container.

Download the necessary files

To use the Setup Tool, you need to download both the tool and a compatible Server Installer `.rpm` file. The Installer file must be version 2021.2.0 or later. Both files can be downloaded from the Tableau Server page.

1. Download the Server installer file, `tableau-server-<version>.rpm` version 2021.2.0 or later.

2. Download the Server in a Container Setup tool, `tableau-server-container-setup-tool-<version>.tar.gz`.

Installation

The Tableau Server in a Container Setup Tool is provided as a tarball. You will need to extract the contents of the compressed file. Here is an example which assumes the Tableau Server in a Container setup tool archive is in your current directory:

```
tar -xzf tableau-server-container-setup-tool-<VERSION>.tar.gz
```

This creates a new directory, `tableau-server-container-setup-tool-<VERSION>` with the `build-image` script you use to run the tool.

Complete the registration form

Edit the registration file to provide your unique identifying information needed to register Tableau Server in accordance with the End User License Agreement. The file, `reg-info.json`, serves as a template for your required, uniquely identifiable registration information and is located in the top directory of the Tableau Server in a Container Setup Tool. This file is used to register the Tableau Server instance running in the image. Providing accurate information will ensure the registration process completes properly.

The `eula` field value is pre-filled with "accept" to indicate you are accepting our End User License Agreement (EULA). You can view the EULA in the EULA directory of the build tool. As outlined in the EULA, you must submit a uniquely identifiable user registration when activating Tableau Server. When you are done editing the registration file, the other fields should have values that reflect your unique information. This file is used to register the Tableau Server instance running in the image. Providing accurate information will ensure the registration process completes and your submission meets the requirements of the license grant.

Note: You must accept the EULA to use Tableau Server. If you do not accept the EULA, you cannot run Tableau Server.

The registration file template `reg-info.json` before editing:

```
{
"zip" : "<value>",
"country" : "<value>",
"city" : "<value>",
"last_name" : "<value>",
"industry" : "<value>",
"eula" : "accept",
"title" : "<value>",
"phone" : "<value>",
"company" : "<value>",
"state" : "<value>",
"department" : "<value>",
"first_name" : "<value>",
"email" : "<value>"
}
```

How to use the tool

There is an executable script in the setup tool called `build-image`. Running this with the `-h` option displays the help:

```
./src/build-image -h
Usage: build-image --accepteula -i [INPUT_IMAGE_NAME] -o [OUTPUT_
IMAGE_NAME] [optional arguments]
Creates new Tableau Server image with pre-installed database
drivers, configuration, and instructions.
REQUIRED

--accepteula              Indicate that you have accepted the End
User License Agreement (EULA).

The EULA is available in the EULA directory of this tool.

-i installer              Path to the Tableau Server installer.
```

```
OPTIONAL

-o output name            Tag of target generated Tableau Server
image.

-e environment file       Optional environment file to configuration
overrides to docker image at build time.

-v setup tool version     Prints the Setup tool version.


-b base image version     Prints the default base image URL.
```

Basic example usage:

```
./build-image --accepteula -i tableau-server-image.rpm
```

Running the command without providing artifacts or a setup script will work, but doesn't really do anything because it would not be copying or installing any additional resources into the original Tableau Server in a Container image.

Orchestration

Tableau only provides documentation and support for Tableau Server containers running on Linux. Tableau does not provide documentation or support for container orchestration systems like Kubernetes or Docker Swarm. Kubernetes can, however, be used to deploy Tableau Server in a Container. For resources and guidance on deploying Tableau containers in Kubernetes, refer to our community-supported GitHub project: https://github.com/tableau/tableau-server-in-kubernetes.

Customizing the image

Setting Environment Variables at Build Time

A subset of environment variables that can customize your image can only be set when the image is built. This includes user, group, and other properties that require root privilege inside the container. Root actions are not available by default at runtime. Additionally it can be useful

to bake certain environment variables into the image at build time so they do not have to be set every time the image is run. All of these environment variables can be set by passing in an environment file to the build-image script.

Environment File

The environment file can be passed to the build-image script using the `-e` argument. The file must conform to this format:

```
<KEY>=<VALUE>
<KEY>=<VALUE>
```

Example Usage:

Create an environment file with the correct format:

```
UNPRIVILEGED_TABLEAU_UID=1012
UNPRIVILEGED_TABLEAU_GID=1020
TABLEAU_USERNAME=myuser
TABLEAU_PASSWORD=pw
```

Pass the file to the image builder

```
./build-image --accepteula -i tableau-server-2020-3.x86_64.rpm -e
<path-to-env-file>
```

Environment Variables

Any environment variable can be set in the environment file. Take a look at the Environment Variables section to see the complete list.

Build time environment variables can only be set when this script is run to build the image:

Environment name	Default	Description
`BASE_IMAGE_URL`	Use the build tool command: `build-`	The default image specified in the build-image tool and Dockerfile is the only officially supported base image. This parameter can be used to either pull a copy of this specific base image

Environment name	Default	Description
	`image -b`	from a custom docker image repository or define a custom base image. If you choose to use a custom-defined base image (see Defining a Custom Base Image for more details), it is your responsibility to ensure it is based on UBI 8 (CentOS 7 or RHEL 7 for version 2022.1 and earlier) and contains the necessary resources to run Tableau Server properly.
`PRIVILEGED_ TABLEAU_GID`	`997`	The GID of the privileged tableau group.
`UNPRIVILEGED_ TABLEAU_GID`	`998`	The GID of the unprivileged tableau group.
`UNPRIVILEGED_ TABLEAU_UID`	`999`	The UID of the user that runs tableau processes (single user deployment).
`UNPRIVILEGED_ USERNAME`	`tableau`	The string name of the unprivileged user.
`UNPRIVILEGED_ GROUP_NAME`	`tableau`	The string name of the unprivileged group.
`PRIVILEGED_ GROUP_NAME`	`tsmadmin`	The string name of the privileged group.
`LANG`	`en_ US.UTF-8`	Locale setting

Drivers, Certificates, and other files

The Tableau Server image does not come with pre-installed data connectors or drivers. You will need to create a bash setup-script that will instruct the `build-image` script to install the data connectors Tableau Server will need. These are the steps you would take:

1. Make sure the Tableau Server in a Container setup tool is installed properly
2. Download the driver from the Tableau driver page: https://www.tableau.com/en-us/support/drivers
3. Copy the downloaded driver file into the customer-files directory in the Tableau Server in a Container setup tool.

   The `customer-files` directory will be copied into the docker image. It will be located in the following path inside the container: `/docker/customer-files`
4. Edit the file in the Tableau Server in a Container setup tool `customer-files/-setup-script` to tell Docker how to install the driver.

   The script is just an executable bash file that is executed when the Docker image is built. (Technically it can be used to perform any arbitrary actions in the image including environment setup, configuration, etc.)

   The script will be run inside the container so be mindful that the commands must work when executed inside the Tableau Server in a Container image.

   For example, if you want to install a driver named mydriver.rpm you would write this in `setup-script`:

   ```
   yum install -y /docker/customer-files/mydriver.rpm
   ```

Tableau Administrative Views

Tableau Administrative views require the PostgreSQL driver to be installed. If you plan on using these views you will need to follow the steps above and install the PostgreSQL driver. For more information about Administrative views, see Administrative Views

Defining a Custom Base Image

The default base image used to build the Tableau Server container is a UBI 8 image (CentOS 7.x for version 2022.1 and earlier) sourced from Docker Hub. In some cases you may want to configure the build-image tool to pull the image from a different docker image repository. For

example, your company may manage an internal docker repository and not want the `build-image` pulling from a public repository. Use the following steps to customizing the base image path:

1. Use the following command to view the current `build-image` tool's default base image name and version:

   ```
   ./build-image -b
   ```

2. Pull the exactly matching base image name and version from Docker Hub and store/cache it in your preferred image repository (per your company's image policies)

3. Come back to the build-image tool. Create or modify an existing environment file to include the `BASE_IMAGE_URL` environment key with the value set to a new docker image registry path:

   ```
   BASE_IMAGE_URL=<custom-image-registry-path>
   ```

4. Build the image with the environment file:

   ```
   ./build-image --accepteula -i <rpm> -e <path-to-env-file>
   ```

These steps enable you to specify a completely different base image. This capability is supported only for UBI 8-based images (RHEL and CentOS 7.x for version 2022.1 and earlier) and is provided to help customers create more secure images.

Using a base image other than the default specified by Tableau carries the risk of producing an image that does not start or function properly. If you choose to use a custom base image, you are responsible for ensuring that the base image enables Tableau Server to run properly. We recommend using the default base image unless it is unacceptable to your organization for some reason, such as security concerns.

The custom base image must be based on UBI 8 (RHEL 7 or CentOS 7 for version 2022.1 and earlier). Using any other distro will result in an unsupported image.

Using internal repositories for yum and pip

The Tableau Server in Container image is configured to use the default yum and pip repositories to pull dependent packages. If you need to edit, remove, or add repositories (for example, internal repositories might be used to improve security), you will need to modify one of the Image Setup Tool's initialization scripts.

Modify the `<setup_tool>/src/image/init/setup_default_environment.bash` script in the source code to use the internal repositories. Please keep any repository files required for the internal repository in the `<setup_tool>/src/image/init/` directory. That directory will be copied to the docker image.

Base Images and Security

Many customers will run container scanning tools (such as AquaScan or TwistLock) against the generated Tableau Server Docker image. These security tools will generate a report of potential security vulnerabilities (or Common Vulnerabilities and Exposures or CVE). There are two types of vulnerabilities associated with the Tableau Server in a Container image:

- CVEs associated with Tableau Server or with a library we have a dependency on.
- CVEs associated with the underlying Linux distro.

Tableau is responsible for CVEs associated directly with Tableau Server. The Security Team analyzes these reports to determine impact and severity to help prioritize the issues for resolution. The baseline remediation priority and timelines will be determined by the original CVSS severity scoring. Third party component security updates will not usually be back-ported into older releases unless there is an executable code path that exposes the vulnerability.

With the containerized distribution model, customers are faced with a different set of challenges around OS Layer vulnerabilities. Traditionally, with Tableau Server, Tableau would focus its efforts on delivering a secure application and the customer is responsible for managing the Operating Systems (Linux or Windows). However, with containerization, the OS (UBI 8 starting in version 2023.3, CentOS 7.x or RHEL 7.x for version 2022.1 and earlier) and the application are packaged together in one container. Tableau takes responsibility for the CVE associated with Tableau Server and for determining if a CVE from a third party library

would impact customers. However customers must be responsible for the OS layer issues. In order for customers to address the security issues in the OS layer customers may replace the base OS layer with their own version (UBI 8-based for version 2022.3 and later, RHEL or CentOS 7.x for version 2022.1 and earlier). In doing this you must validate that Tableau Server runs correctly. Removing a library Tableau Server is dependent on because of security issues may result in a non-functioning instance of Tableau Server. If you change the base OS layer customers are responsible for validating that it works.

Hardening Tableau Server in a Container

You can apply the standard hardening steps to Tableau Server in a Container. For more information on hardening Tableau Server, see Security Hardening Checklist.

Deployment Configuration Examples

Docker

Tableau Server in a Container Basic Usage

```
docker run \
-e LICENSE_KEY=<key>
-p 8080:8080
-d <Tableau Server in a Container image ID or tag>
```

Tableau Server in a Container Basic Usage with Automated Initial Admin User

```
docker run \
-e LICENSE_KEY=<key> \
-e TABLEAU_USERNAME=<myadmin> \
-e TABLEAU_PASSWORD_FILE=/etc/tableau-admin-secret \
-v <full-path-to-pw-file>:/etc/tableau-admin-secret \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

TSM only mode

```
docker run \
-e TSM_ONLY=1 \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Multi-Node Basic Usage

Initial Node

Option 1: Use this if the server configuration (`CONFIG_FILE`) specifies a multi-node topology:

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-v <full-path-to-config-file>:/docker/config/config.json:ro \
-e LICENSE_KEY=<key> \
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \
--hostname=<name-of-host-machine> \
-d <Tableau Server in a Container image ID or tag>
```

Option 2: Use this if you want a multi-node deployment even if server configuration does not specify multi-node topology:

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-e LICENSE_KEY=<key> -e ALWAYS_WRITE_BOOTSTRAP_FILE=1 \
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \
--hostname=<name-of-host-machine> \
-d <Tableau Server in a Container image ID or tag>
```

Additional node

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-e BOOTSTRAP_INSTALL=1 \
-p 8080:8080 -p 8800-9000:8800-9000 \
--hostname=<name-of-host-machine> \
-d <Tableau Server in a Container image ID or tag>
```

Externalize Data Usage

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
-e LICENSE_KEY=<key> \
```

```
--hostname=localhost \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Init-Container Basic Usage

Init Container

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
-e LICENSE_KEY=<key> \
-e INIT_CONTAINER=1 \
--hostname=localhost \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Run Container

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
--hostname=localhost \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Basic Restore from Backup Single-Node

```
docker run \
-v <full-path-to-backup-file>:/docker/config/backup/backup-file.ts-
bak \
-v <full-path-to-config-only-file>:/docker/config/config.json:ro \
-e LICENSE_KEY=<key> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Docker-Compose

```
version: '3.2'
services:
    tableau-server:
         hostname: localhost
         volumes:
              - <your-tsm-command-file>:/docker/config/tsm-com-
mands:ro
```

```
      - <your-config-file >:/docker/config/config.json:ro
ports:
      - "8080:8080"
image: ${IMAGE_NAME}
environment:
      - LICENSE_KEY=<license-key>
```

Tableau Server in a Container - Using an Image

Introduction

Tableau Server in a Container is Tableau's first container-based server offering. Tableau Server in a Container is an all-in-one Tableau Server instance running inside of a Linux Docker container. In other words, a Tableau Server in a Container image is a docker image that runs an entire self-contained Tableau Server application. Tableau Server in a Container is our first of many steps to support Tableau Server running in container-based environments. The easiest way to understand the concept of Tableau Server in a Container is to think of it like a virtual machine (VM) with Tableau Server pre-installed. The image is based on a UBI 8 image (CentOS 7.x for version 2022.1 and earlier) and runs `supervisord` (instead of `systemd`) inside the container. When the container starts `supervisord`, it will immediately attempt to initialize and start Tableau Server. Much of the documentation here aims to describe how to provide configuration and leverage automation so you can run Tableau Server in Docker environments.

The Tableau Server in a Container Image Setup Tool helps you create and customize container images to include custom packages and artifacts. One of the primary functions of the tool is to build the container image and install custom data connectors.

To test out the Tableau Server in a Container Image quickly in a proof-of-concept scenarios, see Tableau Server in a Container - Quick Start.

Limitations for Tableau Server in a Container

- Tableau Server in a Container only supports license activation using Server ATR. Offline activation using Server ATR is supported in 2023.1 and later. This functionality is available in Containers but requires extra steps and approval. If you need to run Tableau Server in a Container in an air-gapped or offline environment, contact your Account representative for more information.
- Tableau Server in a Container does not currently support the Resource Monitoring Tool (RMT) agent.
- Kerberos is not supported in Tableau Server in a Container.

Tableau Server in a Container Image

The Tableau Server in a Container Image is a Docker image that contains all of Tableau Server. The image is built using the Tableau Server in a Container Setup Tool. When built, the image includes Tableau Server but it is not yet initialized. The default user in a Tableau Server in a Container image is a non-root unprivileged user.

Prerequisites

Run the configure-container-host script

When Tableau Server is installed without a container, certain resource limits and coredump properties are changed as part of the installation process. This is done to help optimize the performance of Tableau Server. A Tableau Server in a Container image does not have the ability to make these changes on the host machine, so we recommend running the configure-container-host script that is provided in the Tableau Server in a Container Setup Tool on any machine that will be running Tableau Server in a Container images. This will help ensure the performance of the Tableau Server in a Container image is equivalent to that of its non-container counterpart.

To run the `configure-container-host` script:

1. Locate the script (`configure-container-host`) in the top-level directory of the Tableau Server in a Container Setup Tool.

2. Copy it to the environments in which you plan on running Tableau Server.

3. Determine the unprivileged user account/uid that will run as the default user of the Tableau Server in a Container image. This user should exist on the host machine and should match the UID set in the Tableau Server container `UNPRIVILEGED_TABLEAU_UID` environment variable. If you did not set this when creating your Docker image, the default unprivileged user id inside the container is 999. If you are using Docker user-mapping, this UID should correspond with the user that exists on the host machine.

4. Execute the script as root:

   ```
   sudo ./configure-container-host -u <uid>
   ```

Running the Image

To run a Tableau Server in a Container docker image, the simplest command to get a Tableau Server in a Container Image running is the following:

```
docker run \
-e LICENSE_KEY=<key>
-p 8080:8080
-d <Tableau Server in a Container image ID or tag>
```

This will run docker in the background and, after some time, will result in a fully installed instance of Tableau Server. Tableau Server can take 10 to 20 minutes to fully start up, depending on the hardware of the computer running the image. You can confirm that the container is running by typing the command `docker ps`. Once Tableau Server is operational, the initial Tableau Server administrator account will need to be created. This step can be automated. For more information, see Automate Initial Tableau Server Administrator.

Basic Run Arguments Summary

All the options used in the Docker run command are necessary, often times more options will be provided to leverage different functionality in the image. For now, let's take a closer look at just the arguments used in the simplest Docker run command for Tableau Server in a Container:

Argument	Description
-e LICENSE_KEY-Y=<key>	Tableau Server must be licensed. This environment variable will store the key that will be used to license the server. This is a required component of the initialization process. You can provide multiple licenses separated with comma.
-p 8080:8080	This tells docker to expose port 8080 inside the container and bind it to port 8080 on the host machine. The first 8080 value is configurable, changing this will modify the port mapped on the host. Tableau Server by default expects to receive user traffic on port 8080 inside the container, you can choose whether to expose this port on a different host port or not at all.

Automate Initial Tableau Server Administrator

When Tableau Server starts up for the first time an initial administrator user must be created before remote network connections to Tableau Server are permitted. This can be done by running the `tabcmd initialuser -s localhost:8080 -u <username> -p <password>` command inside the container. You may also set admin credentials via environment variables. `TABLEAU_USERNAME` and `TABLEAU_PASSWORD` or `TABLEAU_PASSWORD_FILE` (preferred) are the environment variables that can be set to pass in initial admin credentials. For more information on password management, see Password Management.

For more information on the `tabcmd initialuser` command see initialuser.

Example

```
docker run \
-e LICENSE_KEY=<key> \
-e TABLEAU_USERNAME=<myadmin> \
-e TABLEAU_PASSWORD_FILE=/etc/tableau-admin-secret \
-v <full-path-to-pw-file>:/etc/tableau-admin-secret \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Licensing

Licensing in Containers

Licensing of Tableau Server in a Container uses the Server Authorization-To-Run (ATR) service to activate Tableau Server deployed in the cloud, containers, or virtual environments without running out of license activations. The ATR service achieves this by providing short-term leases of configurable duration (ATR duration) until the product key expiration date is met. ATR abstracts Tableau licensing from underlying hardware changes, which is a fundamental aspect of container deployments. Since Server ATR requires the container to be able to reach the ATR service hosted by Tableau, the containers require internet access. Tableau Server in a Container does not support offline or manual activation. See Activate Tableau Server Using the Authorization-To-Run (ATR) Service for more details.

Important: You must provide either the `LICENSE_KEY` or `LICENSE_KEY_FILE` environment variables (only set one).

When upgrading Tableau Server in a container, if you've used the maximum number of activations for your license, Tableau Server cannot start until the ATR duration has elapsed (4 hours/14400 seconds by default). For more information about setting or changing the ATR duration, see Activate Tableau Server Using the Authorization-To-Run (ATR) Service.

License Environment Variable

Tableau Server in a Container supports setting license keys using an environment variable: the `LICENSE_KEY` can contain one or more keys (-e LICENSE_KEY="<key1> , <key2>") via a comma separated list.

Example

```
docker run \
-e LICENSE_KEY="<key1>, <key2>" \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

License File

Tableau Server in a Container also supports setting license keys using a file. Mount a file to the default license key file location in the container (`/docker/config/license_file`) or as otherwise specified by the environment variable `LICENSE_KEY_FILE`.

Example

```
docker run \
-v <full-path-to-license-file>:/docker/config/license_file \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Requested license lease time

You can specify the ATR license lease time in a Tableau Server container by setting the environment variable `REQUESTED_LEASE_TIME`. You must provide the lease time in seconds, with the minimum duration being 3600 seconds (or 1 hour). It is recommended that you lower the lease time when experimenting and testing Tableau Server to reduce the likelihood of reaching the max activated lease limit. For production deployments, it is strongly recommend to not set the `REQUESTED_LEASE_TIME` parameter (thus using the default value), so Tableau can determine the ideal lease time.

Example

```
docker run \
...
-e REQUESTED_LEASE_TIME=<time-in-seconds> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Running an Uninitialized Image

Tableau Server has two phases of installation, first the Tableau Service Manager (TSM) services are installed. In a typical On-Premise installation this step is a time for server admins to register their server, activate their licenses, and configure the server to behave the way they want it to. The second phase of installation is setting up and starting the Tableau Server processes which will handle end-user traffic and related business logic.

The default behavior of Tableau Server in a Container images is to automate all installation steps so the `docker run` command eventually results in a fully functional server. However, if you want to start a Tableau Server in a Container image and have it only running the TSM services (what a server administrator would expect if they just ran `initialize-tsm`), you can do this by passing the `TSM_ONLY` flag as an environment variable.

For example:

```
docker run \
-e TSM_ONLY=1 \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Interacting with the Image

When you have a Tableau Server in a Container image running, you can call TSM and tabcmd commands directly. Those tools are added directly to the environment path of the pid 1 user (which is root at this time). This means you can call TSM and tabcmd commands as folllows:

```
docker exec -it <container> tsm status -v
```

```
docker exec -it <container> tabcmd initialuser -s localhost -u
<admin> -p <secret>
```

It is also possible to open a shell in the container to perform more general operations. This is generally not recommended except for debugging purposes:

```
docker exec -it <container> bash
```

TSM Web UI and Remote CLI

The TSM Web UI and remote CLI is not accessible by default. This is because it requires a username and password to authenticate and by default the user running tableau server processes in the container is not provided a password. This is done for security reasons (we do not recommend shipping images with a default password inside them as that would allow remote access). In some cases the TSM Web UI and making remote access calls using the

TSM CLI can be useful. If you want to use these features you will need to follow the steps outlined below to create a remote access user account.

More detailed information about the TSM Web UI and CLI see Sign in to Tableau Services Manager Web UI.

Create a TSM Remote User

Specify the `TSM_REMOTE_UID` and `TSM_REMOTE_USERNAME` environment variables when you build the Tableau Server in a Container image using the Setup Tool. Creation of a TSM enabled account in the image requires privileged access in the image that isn't available at runtime, and therefor it can only be done when the Docker image is being built by the Tableau Server in a Container Setup Tool (`build-image`).

Example Tableau Server in a Container Setup Tool `environment` file:

```
TSM_REMOTE_UID=1010
TSM_REMOTE_USERNAME=myuser
```

Set the password for the TSM Remote User

The Tableau Server in a Container image requires a password for the account when the image is run. There are two ways you can set the password for this account.

Secrets File (Recommended)

Create a file named `remote-user-secret`, write the password into the file and mount it into the container at runtime. `TSM_REMOTE_PASSWORD_FILE` determines the expected location (default location is `/docker/config/remote-user-secret`) of the secrets file in the container.

Example remote-user-secret file:

```
mypassword
```

Example docker run command:

```
docker run \
-e LICENSE_KEY=<key>
-v {absolute-path}/remote-user-secret:/docker/config/remote-user-
secret
-p 8080:8080 \
-p 8850:8850 \
-d <Tableau Server in a Container image ID or tag>
```

Environment Variable

You can also simply set the `TSM_REMOTE_PASSWORD` environment variable when starting the docker image.

Example docker run command:

```
docker run \
-e LICENSE_KEY=<key>
-e TSM_REMOTE_PASSWORD=<password>
-p 8080:8080 \
-p 8850:8850 \
-d <Tableau Server in a Container image ID or tag>
```

Security Notes

- Port 8850 must be exposed to receive TSM request traffic.
- If the password is not set properly in the image at runtime the container will exit immediately.
- TSM relies on the image's Linux user account system. In this case the account is a restricted inside the image. This means the account will have a restricted shell and is limited to executing two commands: `/bin/true` and `passwd`.

How to Rotate the TSM Remote User's Password

If you want to rotate the account password of the TSM remote user, you can do this using either of these options:

Start a new Tableau Server in a Container

The account's password is set every time the container starts up. If you are persisting Tableau data outside the container then starting a fresh image with a new password will effectively rotate the password.

1. Shut down and remove the running image
2. Set a new password value in either `TSM_REMOTE_PASSWORD` or `TSM_REMOTE_PASSWORD_FILE` environment variables (see above) in your image configuration.
3. Start the image up again.

Rotate the password manually inside a running container

If you don't want to shut down the image, you can still rotate the password manually.

1. Open a shell in the running container
2. Log in as the remote user account using the `su` command
3. Run the `passwd` command to change the password.

> **Warning:** These manual rotations only persist as long as the container instance's write layer remains. If you delete the container, the manual changes will not be applied when a new container is started.

Initial Configuration Options

Configuring Tableau Server in a Container is essential for getting the Tableau Server behavior you want. Tableau Server in a Container is a clean installation of Tableau Server, so you need to provide the same information to the container as when configuring Tableau Server outside a container..

Runtime Environment Variables

The runtime environment variables below tell the Tableau Server in a Container image how to deploy Tableau Server. A subset of these will be described in greater detail.

All of these values are designed to be overridden to enable more flexibility for configuration.

Environment name	Default	Description
ACCEPT-EULA	`0`	Automatically set to 1 when an image is built using the Tableau Server in a Container Setup tool.
LICENSE_KEY		Set to the license key that will be used to license the server. Accepts multiple licenses separated with commas.
LICENSE_KEY_FILE	`/docker/config/license_file`	File path to license file. The format of the license file should be one license key per line.
REGISTRATION_FILE	`/docker/config/tableau_reg.json`	File path to registration file inside the image. By default this contains the registration information that was provided when the Tableau Server in a Container image was built. This can be overwritten at run time. For more information, see Activate and Register Tableau Server.
REGISTRATION_DATA		An alternative way to overwrite registration information at runtime. This environment variable must be set to a serialized JSON string containing the same registration information that would be found in a Tableau Server registration file. For more information, see Activate and Register Tableau Server.

Environment name	Default	Description
TABLEAU_USERNAME		This refers to the initial administrator account on Tableau Server. This is recommended but optional. If this user is not set the inital admin account for Tableau Server will need to be set using tabcmd. If this variable is set to a value then a password is also required. This is only used when Tableau Server is initialized for the first time. Setting this value will tell Tableau Server in a Container to automatically attempt to initialize the user. For more information, see Add an Administrator Account.
TABLEAU_PASSWORD		A plain text password for the tableau user. This refers to the initial administrator account on Tableau Server. This is required if `TABLEAU_USERNAME` is specified. For more information, see Add an Administrator Account.
TABLEAU_PASSWORD_FILE		A file path to a file containing only the password text for the tableau user. This refers to the initial administrator account on Tableau Server. This is required if `TABLEAU_USERNAME` is specified. For more information, see Add an Administrator Account.
CONFIG_FILE	`/docker/-config/config.json`	File path to default TSM config file. The file will be used to configure Tableau Server. For more information, see Configuration File Example. Do not set `CONFIG_DATA` if `CONFIG_FILE` used
CONFI-		This can be used as a substitute for `CONFIG_`

Environment name	Default	Description
G_DATA		FILE. If you want to provide configuration to the server without mounting an external file, set this environment variable to the equivalent serialized contents of a TSM config file. Example CONFIG_DATA="{\"configEntities\":{\"identityStore\":{\"_type\":\"identityStoreType\",\"type\":\"local\"}}}" For more information, see Configuration File Example Do not set CONFIG_FILE if CONFIG_DATA is used
IGNORE_TOPOLOGY_CONFIG	0	0 or 1. If set to 1, the container will ignore any topology related configuration present in the config file designated by CONFIG_FILE.
BACKUP_FILE	/docker/-config/backup/backup-file.tsbak	A file path to a Tableau Server backup file (.tsbak). If provided during initialization the server will attempt a restore.
INIT_CONTAINER	0	0 or 1. If set to 1 Tableau Server will only attempt to initialize TSM and initialize Tableau Server and the container will exit upon completion.
TSM_	0	0 or 1. Equivalent to installing the Tableau

Environment name	Default	Description
ONLY		Server rpm and running initialize-tsm. Only the TSM (Tableau Service Manager) services will start. ONLY works if the container is initializing for the first time (this will not work if a Tableau Server in a Container is being started up with a previously initialized server directory).
BOOTSTRAP_INSTALL	`0`	0 or 1. Indicates whether or not the server is an initial node or an additional node. If set to 1, the container will wait indefinitely until a bootstrap file exists at the location specified by `$BOOTSTRAP_FILE`
ALWAYS_WRITE_BOOTSTRAP_FILE	`0`	0 or 1. If set to 1, the container will write a bootstrap file to the location given in `BOOTSTRAP_FILE`.
WAIT_FOR_BOOTSTRAP_FILE	`1`	0 or 1. If set to 1 (default), if the container detected it is a worker installation (`BOOTSTRAP_INSTALL=1`). The container will wait indefinitely until a file is detected located at the path set in `BOOTSTRAP_FILE`. If set to 0 when the startup process is run this wait will be skipped. This can be useful in some debug cases.
BOOTSTRAP_FILE	`/docker/-config/-bootstrap/-`	File path to bootstrap file. Only applies to worker containers. This file should only point to a bootstrap file. The typical usage would be to mount the directory of the target file (default would be `/docker/config/bootstrap`) to the host.

Environment name	Default	Description
	`bootstrap.json`	
BOOTSTRAP_DATA		This can be used as a substitute for BOOTSTRAP_FILE. If you want to provide a bootstrap file without mounting an external file, set this environment variable to the equivalent serialized contents of a TSM bootstrap file. Do not set BOOTSTRAP_DATA if using BOOTSTRAP_FILE.
PORT_RANGE_MIN	`8800`	For performance reasons, we recommend exposing only 200 ports (8800-9000) instead of the Tableau Server On-Premise default 8000-9000 port range because exposing 1000 ports in docker can negatively impact the start up time of the docker image. See Exposing Licensing and TSM ports below for more information.
PORT_RANGE_MAX	`9000`	We recommend exposing only 200 ports (8800-9000) instead of the Tableau Server On-Premise default 8000-9000 port range because exposing 1000 ports in docker can negatively impact the start up time of the docker image. See Exposing Licensing and TSM ports below for more information.
HTTP_PROXY		To forward http requests to your proxy server, set this environment variable to point your proxy host. For example, to set the proxy to example-host for port 8080, `HTTP_PROXY=http://example-host:8080/`
HTTPS_PROXY		To forward https requests to your proxy server, set this environment variable to point your proxy

Environment name	Default	Description
		host. For example, to set the proxy to example-host for port 443, `HTTPS_PROXY=http://example-host:443/` Be sure to use 'http' when you specify the URL for the `HTTPS_PROXY` environmental variable.
NO_PROXY		To bypass the proxy server, specify exceptions in the no_proxy variable. Use this variable if your proxy server does not route internal addresses. You must also add exceptions to this proxy configuration to guarantee that all communications within a local Tableau Server cluster (if you have one now or will have one later) do not route to the proxy server. Enter both the host name and the IP address for each computer, and add host name of the container. Additionally, include the canonical host name (localhost) and IP address (127.0.0.1) for the local computer. For example, to specify exceptions for a three-node cluster: `NO_PROXY="localhost,127.0.0.1,hostname1,hostname2,hostname3,IP1,IP2,IP3"`
COORDINATION_SERVICE_CLIENT_PORT	Any port between `PORT_RANGE_MIN` and `PORT_RANGE_MAX`	Client port for the coordination service.

Environment name	Default	Description
COORDINATION_SERVICE_PEER_PORT	Any port between `PORT_RANGE_MIN` and `PORT_RANGE_MAX`	Peer port for the coordination service.
COORDINATION_SERVICE_LEADER_PORT	Any port between `PORT_RANGE_MIN` and `PORT_RANGE_MAX`	Leader port for the coordination service.
LICENSE_SERVICE_VENDOR_DAEMON_PORT	Any port between `PORT_RANGE_MIN` and `PORT_RANGE_MAX`	Vendor daemon port for the licensing service.
AGENT_FILE_TRANSFER_PORT	Any port between `PORT_RANGE_MIN` and `PORT_RANGE_MAX`	File transfer port for the agent service.
CONTR-	Any port between `PORT_`	https port for the controller service.

Environment name	Default	Description
OLLER_PORT	`RANGE_MIN` and `PORT_RANGE_MAX`	
REQUESTED_LEASE_TIME	Default is currently set to 4 hours.	Set the requested lease time for Server ATR activations. You need to provide the time value in seconds and the minimum duration is 14400 seconds (or 4 hours). Changing this value is generally not recommended for production deployments. However when developing or prototyping with Tableau Server in a Container you may want to set this to the minimum value so as to minimize the loss of activations.

Readonly Environment Variables

These are environment properties that describe some of the basic properties of the Tableau Server in a Container image. Overriding these values is not recommended.

Environment name	Default	Description
PRE_INIT_COMMAND_SCRIPT	`${DOCKER_CONFIG}/customer-files/pre_init_command`	Path to a user custom bash/executable file to be run in Tableau Server prior to Tableau Server initialization. **Note:** Make sure the file has execute permission for all users, otherwise run `chmod +rx <path-to-pre-init-command-file>`
POST_INIT_COMMAND_SCRIPT	`${DOCKER_CONFIG}/customer-files/post_init_command`	Path to a user custom bash/executable file to be run in Tableau Server after the

Environment name	Default	Description
		server is fully functional and running. **Note:** Make sure the file has execute permission for all users, otherwise run `chmod +rx <path-to-post-init-command-file>`
DATA_DIR	`/var/opt/tableau/tableau_server`	The data directory where Tableau Server bits should be written.
INSTALL_DIR	`/opt/tableau/tableau_server`	The installation directory where Tableau Server installation bits are written.
SERVICE_NAME	`Tableau Server`	Name of the application running in the container.
SERVICE_VERSION	N/A	Version of Tableau Server installed in the container.
DOCKER_CONFIG	`/docker`	Directory that stores Tableau specific docker configuration.
ENV_FILE	`${DOCKER_CONFIG}/customer-files/environment`	File that contains all user environment overrides.

Build-Time Environment Variables

BASE_IMAGE_URL	Use the build tool command: `build-image -b`	The default image specified in the build-image tool and Dockerfile is the only officially supported base image. This parameter can be

		used to either pull a copy of this specific base image from a custom docker image repository or define a custom base image. If you choose to use a custom-defined base image, it is your responsibility to ensure it is based on UBI 8 (CentOS or RHEL 7 for version 2022.1 and earlier) and contains the necessary resources to run Tableau Server properly. For more information on custom base images, see Tableau Server in a Container - Using an Image.
PRIVILEGED_ TABLEAU_GID	`997`	The GID of the privileged Tableau group.
UNPRIVILEGED_ TABLEAU_GID	`998`	The GID of the unprivileged Tableau group.
UNPRIVILEGED_ TABLEAU_UID	`999`	The UID of the user that runs Tableau processes (single user deployment).
UNPRIVILEGED_ USERNAME	`tableau`	The string name of the unprivileged user.
UNPRIVILEGED_ GROUP_NAME	`tableau`	The string name of the unprivileged group.
PRIVILEGED_ GROUP_NAME	`tsmadmin`	The string name of the privileged group.
LANG	`en_US.UTF-8`	Locale setting

Tableau Server Configuration Overrides

These environment variables can be overwritten by Docker to point at any file in the container. So if you want to specify a different mount point you are welcome to do so.

Tableau Server needs a configuration file to start and run:

```
CONFIG_FILE=/docker/config/config.json
```

`CONFIG_FILE` refers to a TSM configuration file. The format and usage is identical to the configuration file described in Configuration File Example.

Pre-initialization and Post-initialization Commands

Tableau Server runs an automated installation script designed to take Server from a pre-initialized state to fully running. However, sometimes you may want to add in your own automation code in the initialization process. We offer two hooks to do this, the pre-initialization script and post-initialization script.

Pre-initialization script

This script will run immediately after the base TSM processes are initialized and before any other TSM setup steps are executed. This is useful for executing TSM configuration commands before Tableau Server runs. For configuration changes made at this point, you do not need to apply the changes because the normal Tableau Server automation does this after your script completes.

Post-initialization script

This script will run after all other Tableau Server initialization and start-up automation completes. Tableau Server will be fully functional and running when this script executes. Configuration changes made at this point must be applied.

Instructions

To add a custom script to one of these hooks in your image, follow these steps:

1. Write your custom script
2. Copy the custom script into the `customer-files` directory of the Tableau Server in Containers Build Image Tool.
3. Rename the script to be either `pre_init_command` or `post_init_command` depending on when you would like the script to run (you can use both hooks independently from each other).
4. Ensure the permissions of the script are either executable by other (`chmod +rx <command-file>`) or the ownership permissions match the unprivileged user in the container.

User Configuration

Tableau Server uses an unprivileged user to run server processes. This user created inside the container when Tableau Server in a Container is initializing. By default the user is named `tableau` with a UID of 999. If you are deploying a Tableau Server in a Container that uses mounts to externally store data on the host machine you may prefer to change the UID to map a to a UID on the host machine. Using docker user namespaces is another way to achieve the same result.

Tableau Server in a Container Utilities and Tools

All Tableau Server in a Container utility and tool functions are placed under this directory:

`/docker/`

File Permission Management

When passing any configuration files to the container you will want to ensure the user running the Tableau Server process inside the container has permission to access the files. To avoid granting all users access to files being mounted to the container you can change the UID and/or the GID of the user running Tableau Server inside the container to match the user-/group owner on the host. The container user will have a UID determined by the `UNPRIVILEGED_TABLEAU_UID` environment variable (default: 999) and GID determined by `UNPRIVILEGED_TABLEAU_GID` (default: 998). These values can be changed by overriding the environment variable or you can use a Docker user namepace mapping to associate the UID/GID in the container to a different UID/GID on the host.

Password Management

Certain features and options require user credentials to be provided as a configuration setting into the container. Tableau Initial Admin credentials is an example of optional credentials that enable additional features. In these cases we always provide two methods of setting the password. First, you can provide a file containing the password and supply a file path to an environment variable. Alternatively, you can set an environment variable to store the password directly.

The recommended and more secure option is to provide the password as a file path to the container. Providing a secret in a file is a well supported pattern in Docker, Docker Swarm, Kubernetes, and other container orchestration systems. Storing passwords directly in environment variables is a common pattern so we do support it, but it typically means the password is less secure.

Examples

Let's take a look at the `TABLEAU_USERNAME` credential. You can provide the password for the user as either `TABLEAU_PASSWORD` or `TABLEAU_PASSWORD_FILE`. When running a Tableau Server in a Container image you can provide either environment variable to supply the password.

The password file environment variable expects a file path inside the container to a valid secrets file. The secrets file should be a single line containing the secret.

Example of using a secrets file

```
docker run \
...
-e TABLEAU_USERNAME=admin \
-e TABLEAU_PASSWORD_FILE=/etc/admin-secret \
-v <full-path-to-pw-file>:/etc/admin-secret \
-d <Tableau Server in a Container image ID or tag>
```

Example contents of a secrets file

```
mypassword23879172
```

Alternatively one can directly store the password in plain text in the password environment variable. This approach is considered less secure but it is more convenient and a common pattern with containers.

Example

```
docker run \
...
-e TABLEAU_USERNAME=admin \
-e TABLEAU_PASSWORD=password \
-d <Tableau Server in a Container image ID or tag>
```

Configuring Tableau Server after it is running

Once Tableau Server has been initialized and is running, the best way to interact with the server is to use the TSM CLI tool. This is the classic Tableau Server tool for performing administrative tasks. In the future we will support Tableau Server reacting to changes in the static configuration provided in the `CONFIG_FILE` environment variable in between runs. But for now, after Tableau Server is initialized, you must interact with the server using the classic tools.

For more information on the TSM command-line see tsm Command Line Reference.

Status

There are two basic status checks for Tableau Server provided in the image. These can be used to check the aliveness and readiness of the server.

Liveness Check

The liveness check indicates whether or not TSM services are running. This means it will indicate whether the orchestrated services of Tableau Server are operating and are functioning. This check is callable here:

```
/docker/alive-check
```

Another option is to expose port 8850 which the Tableau Controller service runs to provide administrative functions through a web browser. One could periodically check the health of the service by checking the health of the service through tcp health checks.

Readiness Check

The readiness check indicates whether Tableau Server is running and business services are ready to receive traffic. This can be determined using the following script:

```
/docker/server-ready-check
```

Another option is to use tcp health checks against port 8080 (or whatever port Tableau Server is bound to receive traffic). Sometimes this kind of tcp health check is more reliable than the server-ready-check, as the server-ready-check is based on service status reported to TSM which can sometimes be delayed as the service state is updated.

Persisting Data

Often times with containers we want the capability to shut a container down and then turn it back on without losing any important information. Tableau Server in a Container images support this in that you can mount certain directories outside of the container so you can completely destroy or remove container instances and still preserve your data. This data can be used to start another container instance and resume where the previous container left off.

The following sections cover the different kinds of managed state.

Tableau Server Data

Server Data is all stored in the data directory. The data directory is where all user-related data and service runtime metadata is stored. Externalizing this data means your user's data can be persisted even after the Tableau Server in a Container is completely removed.

This data is transferable and can is usable with cloud-managed block storage system, like AWS EBS volumes.

When Tableau Server in a Container is used in conjunction with External Filestore, the data directory must be on EBS. Do not use a network file system (for example, NFS) for the data directory. The External Filestore directory can be on an NFS volume.

Static Hostnames

Tableau Server does not handle dynamic hostname changes well so it is important to specify the container's internal hostname so it is consistent between container runs. The hostname inside a container is arbitrary and can be set to any value. Using the `--hostname` option allows one to specify the internal hostname of the container. **Make sure subsequent containers using the same persistent data are run using the same hostname value.**

This is not to be confused with multi-node server installations. In these, additional nodes should each be assigned a different hostname. What matters is when any single container is restarted the replacing container that will use the same persistent data for that instance must have a matching hostname.

Complete Example

Here is an example where the data directory is mounted outside the container.

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
-e LICENSE_KEY=<key> \
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Backup and Restore

Tableau Server in a Container supports Tableau Server creating backups and restoring from a backup file (.tsbak). The first step is to get a Tableau Server in a Container image running, have the backup file (.tsbak) mounted in the image and set the environment variable `BACKUP_FILE` with the file path to the backup file. Additionally you must provide the backup json configuration file in the `CONFIG_FILE` environment variable. The Tableau Server container automates the restore process even for multi-node deployments. If at any point this automation fails to fully restore the system, you can always fallback on the classic Tableau Server tools

and processes such as TSM commands to interact with Tableau Server in the same way you would with a non-container deployment.

For more information on how to perform a backup and restore of a standard Tableau Server instance see Perform a Full Backup and Restore of Tableau Server.

Backup in Tableau Server Container

1. Open shell inside the Tableau Server in a Container version A. Create repository backup, and topology and configuration backup files.

```
docker exec -it my-server bash

# Just providing filename automatically produces the backup
file at /var/opt/tableau/tableau_server-
/data/tabsvc/files/backups/
tsm maintenance backup -f <repository-backup>.tsbak -d

# Any filepath where current user(UNPRIVILEGED USER) can write.
tsm settings export -f /var/opt/tableau/tableau_server-
/data/tabsvc/files/backups/<topology-conf-backup>.json
```

2. Copy the files created in previous step into the host machine. Change the file permission to have read-all permission set for both files.

```
docker cp my-server:/var/opt/tableau/tableau_server-
/data/tabsvc/files/backups/<repository-backup>.tsbak ./<re-
pository-backup>.tsbak
docker cp my-server:/var/opt/tableau/tableau_server-
/data/tabsvc/files/backups/<topology-conf-backup>.json ./<to-
pology-conf-backup>.json
chmod a+r ./<repository-backup>.tsbak ./<topology-conf-
backup>.json
```

3. Store backup artifacts in a secure location. Follow the restore steps below when needed.

Restore inside Tableau Server Container

Backups from any supported Tableau Server version (container and non-container) can be restored inside the Tableau Server container.

Prerequisites

- Tableau Server backup file.
- Configuration json file containing both configuration and topology information.
- **Note:** You will likely need to change the backup files to have read-all permission set. Backup files are typically locked to the user that created the file, and this one will likely be different than the Tableau user running in the container.

```
docker run \
-v <full-path-to-backup-file>:/docker/config/backup/backup-file.ts-
bak \
-v <full-path-to-config-only-file>:/docker/config/config.json:ro \
-e LICENSE_KEY=<key> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Notes:

- If you are restoring a multi-node system you must also start the other nodes in order for the restore automation to work. See the Multi-node Tableau Server in a Container section of this document for more information. Only the initial node requires the backup file, backup configuration file, and license.
- The backup files only need to be provided in the first run of the container. Once the server is initialized you do not need to keep mounting in the backup files.

Migrating from Tableau Server to Tableau Server in a Container

In order to migrate from a standard Tableau Server installation to Tableau Server in a Container you must use the backup and restore technique. Backups from any supported Tableau Server version (container and non-container) can be restored inside the Tableau Server container. See the Restore inside Tableau Server Container section above for more information.

Upgrading Tableau Server Versions

There are two ways to upgrade Tableau Server. The Upgrade-Image method listed in this section is the recommended solution. However, as a fallback it is also possible to upgrade Tableau Server using Backup/Restore.

Upgrading through Upgrade-Image method

The Upgrade Image is a Docker image that can be built using the `build-upgrade-image` script from the Tableau Server in a Container setup tool. The purpose of the image is solely to upgrade the currently running Tableau Server in a Container.

Follow the below steps to do the upgrade.

1. Create an upgrade-image using the `build-upgrade-image` script. The new version's tableau server rpm is needed to build this container.
2. Shutdown the container currently running the Tableau Server.
3. Start the upgrade-image, mounting the same data directory from the container shutdown in the previous step.
4. The upgrade process takes a while, but the tableau server will be upgraded, check docker logs for upgrade process update. The container will shut down after the upgrade process.
5. Start a new Tableau Server in a Container of newer version. Mount the same directory from the previous steps.

Example:

Let's say we have a Tableau Server in a Container running Tableau Server. Here are some assumptions made in this example:

- I have valuable data, and I don't want to lose any data during the upgrade process. The data directory needs to be persisted outside the container.
- The container is named my-server. The docker image is named as tableau-server-:versionA.
- The server version my-server is currently using is version A.
- The server version I want to upgrade to is version B.

1. Get tableau server rpm for version B. Create an upgrade-image.

```
# For all the options available in the script
./build-upgrade-image -h

# Frequently used command to create a upgrade-image
./build-upgrade-image --installer=<path to the tableau server
version B> -i tableau-server:versionA -o tableau-server-
upgrade:versionAB
```

2. Stop the my-server container.

```
docker stop my-server -t 120
```

3. Start the newly created image tableau-server-upgrade:versionAB. Mount the same data directory from the previously stopped container. The container starts the upgrade process to version B.

```
docker run --name my-upgrade-server \
-v <data-dir mount from previous step>:/var/opt/tableau \
...
tableau-server-upgrade:versionAB
```

4. The container will stop once the upgrade is complete. Check the docker logs for upgrade process logs and make sure the upgrade process is a success. You can also check the exit code of the docker container, to make sure the upgrade process is completed successfully.

```
# The log file /var/opt/tableau/tableau_server/logs/upgrade-con-
sole.log is created after 3-4 mins into the start of upgrade
container. When the upgrade completes successfully, "upgrade is
complete" log will be # seen.
docker logs my-upgrade-server
...
...
Verifying licensing state.
Tableau Server has been upgraded to version near.20.0801.1050.
```

```
>> The upgraded Tableau binary directory will be added to PATH
for new shells. To get the
>> updated path, either start a new session, or for bash users
run:
>> source /etc/profile.d/tableau_server.sh
Starting service...
Starting service...
Job id is '12', timeout is 30 minutes.
Service was started successfully.
Status: RUNNING
Tableau Server is Running
upgrade is complete
```

5. Stop the my-upgrade-server container. Start the new version B of Tableau Server in a Container image and mount the data directory from the stopped my-upgrade-server container

```
# Stop the server.
docker stop my-upgrade-server -t 120


# Run the new version Hu
docker run --name my-upgraded-server \
-v <data-dir mount from previous step>:/var/opt/tableau \
...
...
tableau-server:versionB
```

Upgrading though Backup-Restore method

Follow the steps in the Backup and Restore Section of this document. The only adjustment needed to change a backup-restore operation into an upgrade operation, is to restore the backup on a new version of Tableau Server.

Multi-node Tableau Server in a Container

Multi-node Tableau Server in a Container refers to a single deployment of Tableau Server distributed across multiple nodes. Multi-node in this context is the same as Tableau Server multi-node where certain processes can be run on other nodes to increase capacity, compute power, etc. This is distinct from starting up multiple individual Tableau Server in a Container where each container is an independent server with its own distinct data.

Multi-node Tableau Server in a Container works much like a non-container Tableau Server multi-nodeinstallation, and uses the same underlying mechanism. To get an overview of setting up a non-container Tableau Server multi-node installation, see Distributed and High Availability Tableau Server Installations.

Here is an example:

Multi-Node Basic Usage

Initial node

Option 1: Use this if the server configuration (`CONFIG_FILE`) specifies a multi-node topology:

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-v <full-path-to-config-file>:/docker/config/config.json:ro \
-e LICENSE_KEY=<key> \
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Option 2: Use this if you want a multi-node deployment even if server configuration does not specify multi-node topology:

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-e LICENSE_KEY=<key> -e ALWAYS_WRITE_BOOTSTRAP_FILE=1 \
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \
```

```
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Additional node

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-e BOOTSTRAP_INSTALL=1 \
-p 8080:8080 -p 8800-9000:8800-9000 \
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Exposing Licensing and TSM ports

In order for worker nodes to communicate with the primary instance we need to open up additional ports. You will need to allow traffic from other nodes on your primary Tableau Server in a Container instance in the following port ranges:

```
Service Ports: 8800-9000
Postgres Port: 8060
Licensing Ports: 27000-27010
```

Be careful how many ports you open: We recommend exposing only 200 ports, 8800-9000, instead of the Tableau Server default port range of 8000-9000, because exposing 1000 ports in Docker can negatively impact the performance and start up time of the Docker image. You can use a smaller or larger port range depending on how complex your Tableau Server topology is. Generally we do not recommend exposing fewer than 100 ports, otherwise you run the risk of services in a cluster being unable to talk to certain services. If you specify your own port range make sure you expose port 8850 (this is implicitly included in the 8800-9000). The port range is specified by setting the `PORT_RANGE_MIN` and `PORT_RANGE_MAX` environment variables.

Additional nodes will also need to expose the Service Port range (8800-9000), but not the Licensing Port range. It is important to note that these port ranges are only to allow Tableau Server inter-process communication. These ports should not be exposed to users or any

other machines other than computers that are running Tableau Server in a Container for the same multi-node cluster.

These port rules are consistent with Tableau Server firewall documentation. For more information, see Configure Local Firewall.

Resolving Hostnames

The multiple nodes of Tableau Server in a Container need to be run with consistent hostnames because Tableau Server does not handle dynamic hostname changes. When running Tableau Server multi-node, those nodes will want to communicate with each other. Tableau Server nodes will attempt to reach each other using the hostnames that multi nodes' Tableau Server in a Container are configured to use. For example, if you run your initial node with a hostname of "initial", additional nodes will attempt to send traffic to a host called "initial". There are multiple ways you can configure images to resolve hostnames to other images. `/etc/hosts` file in each container to map the arbitrary container hostname (i.e. "initial") to the IP address that is actually running the other container.

Bootstrapping additional nodes

The initial Tableau Server container that runs as part of a cluster generates a bootstrap file that subsequent additional nodes need to use to join the cluster. After additional nodes are registered to the cluster's topology you can start assigning Tableau Server processes to run on them. This process can be fully automated. If you have provided a Tableau Server configuration file (commonly supplied by mounting a configuration file to the file path specified by `CONFIG_FILE`, default path: `/docker/config/config.json`) that specifies a Multi-node topology, the initial node will automatically wait until all additional nodes have registered. Once registered the Multi-node topology will be applied across the cluster.

Once the initial node in Tableau Server in a Container is fully running Tableau Server you can have it generate a bootstrap file for additional nodes:

```
docker exec -it <container-name> tsm topology nodes get-bootstrap-file -f $BOOTSTRAP_FILE
```

This command is automatically called for you if you set the value of `ALWAYS_WRITE_BOOTSTRAP_FILE` to 1.

Security Considerations

The bootstrap file contains server secrets that allow it to establish a TSM session with the initial node. This means if a malicious user obtained the file they could send TSM commands to the server for a period of time. The file itself also contains data that would enable the decryption of server configuration secrets. This file should be treated as sensitive and should be accessible only by services and systems directly pertaining to establishing a multi-node deployment.

Bootstrap Expiration

Bootstrap files carry a limited-time session that lasts for 2 hours. In that window the additional nodes will not need to supply credentials to the initial node in order to join as an additional node. It is possible to use a bootstrap file once the session has expired, however it would mean needing to supply credentials to the initial node.

Transferring the Bootstrap File

The bootstrap file needs to be made available to and consumed by worker nodes' Tableau Server in a Container. The bootstrap file will need to be shared with all other nodes' Tableau Server in a Container that you want to as worker nodes for this deployment. This can be done in many different ways.

Transfer the file over a secure network

Part of your automation on the initial node can involve sending the file directly to additional nodes. This should be done using some secure file transfer client/tool. This is probably more useful in scenarios where multiple bootstrap files may be generated throughout the life of the initial node (possibly to add more additional nodes at a later time).

Use a network file mount

A network file mount shared by all the containers in a given deployment is another option.

Other

The end goal is to securely transfer a file produced by one container and transfer it to a specific set of other containers. So any method that achieves this and is secure is sufficient.

Starting additional nodes

To start up a Tableau Server in a Container additional node, simply start the container with the `BOOTSTRAP_INSTALL` environment variable set to 1.

This will tell the Tableau Server in a Container instance to sleep until a bootstrap file exists at the path specified by the `BOOTSTRAP_FILE` environment variable (which is also configurable). Refer to the environment variable table to view the default file path. To clarify, if you run a Tableau Server in a Container image in "additional node mode" the container will not start `supervisord` or any other process other than a bash script running as pid 1 that checks every 5 seconds to see if the bootstrap file exists. Once the file is present the Tableau Server in a Container will proceed to initialize as an additional node.

Configuring additional nodes

Configuring additional nodes to run a specific topology works the same as it does in a normal Tableau Server deployment. It also comes with the same requirements, which means adding new processes on a node may require a cluster-wide restart. For more information, see Configure Nodes.

Tableau Server Feature Considerations

Some Tableau Server features works differently in containers. This section covers specific features that have special or different considerations in a container environment.

Active Directory

Set AD Domain Controller

If you plan on using Active Directory as an Identity Store for Tableau Server web pages and sites, there is an additional consideration to account for. Tableau Servers running in Linux environments dynamically determine which AD Domain Controller to communicate with by

examining their IP subnet. Containers can be assigned arbitrary IP addresses, and in this case Tableau Server will not necessarily be able to use its IP address to find an appropriate domain controller. For this reason it may be necessary to configure a specific domain controller / hostname for Tableau Server to communicate with. To do this follow these steps:

1. Determine which domain controller you want Tableau Server to use and get the hostname.
2. Set the configuration key `wgserver.domain.ldap.hostname` to the hostname using the standard Tableau Server Admin configuration options:
    - Set the value in the json configuration file `CONFIG_FILE`.
    - Use the TSM configuration command

      ```
      tsm configuration set -k wgserver.domain.ldap.hostname -v
      <hostname>
      ```

Import AD certificate to Tableau Server Keystore

By default Tableau Server in a container communicates with AD via StartTLS whenever simple bind is used. So when the container is run in this configuration, it is necessary to import the AD server certificate to the Tableau Server Keystore, otherwise server initialization will fail. To do this follow these steps:

1. Create a `pre-init-command` script (check Pre-initialization script section). Add the following line to add the AD certificate to tableau server keystore.

   ```
   ${INSTALL_DIR}/packages/repository.${SERVICE_VERSION}/jre/bin -
   importcert -noprompt -alias startTlsCert -file <mounted-cer-
   tificate-path> -storetype JKS -storepass changeit -keystore
   ${DATA_DIR}/config/tableauservicesmanagerca.jks
   ```

2. Mount the AD server certificate at the filepath provided for `-file` parameter in the `pre-init-command` script.

Alternatively, the default setting to communicating with AD via StartTLS can be disabled. Set `wgserver.domain.ldap.starttls.enabled` to `false` to disable the StartTLS. But it is not recommended.

Deployment Configuration Examples

Docker

Tableau Server in a Container Basic Usage

```
docker run \
-e LICENSE_KEY=<key>
-p 8080:8080
-d <Tableau Server in a Container image ID or tag>
```

Tableau Server in a Container Basic Usage with Automated Initial Admin User

```
docker run \
-e LICENSE_KEY=<key> \
-e TABLEAU_USERNAME=<myadmin> \
-e TABLEAU_PASSWORD_FILE=/etc/tableau-admin-secret \
-v <full-path-to-pw-file>:/etc/tableau-admin-secret \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

TSM only mode

```
docker run \
-e TSM_ONLY=1 \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Multi-Node Basic Usage

Initial Node

Option 1: Use this if the server configuration (`CONFIG_FILE`) specifies a multi-node topology:

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-v <full-path-to-config-file>:/docker/config/config.json:ro \
-e LICENSE_KEY=<key> \
```

```
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Option 2: Use this if you want a multi-node deployment even if server configuration does not specify multi-node topology:

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-e LICENSE_KEY=<key> -e ALWAYS_WRITE_BOOTSTRAP_FILE=1 \
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Additional node

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-e BOOTSTRAP_INSTALL=1 \
-p 8080:8080 -p 8800-9000:8800-9000 \
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Externalize Data Usage

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
-e LICENSE_KEY=<key> \
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Init-Container Basic Usage

Init Container

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
-e LICENSE_KEY=<key> \
-e INIT_CONTAINER=1 \
```

```
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Run Container

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Basic Restore from Backup Single-Node

```
docker run \
-v <full-path-to-backup-file>:/docker/config/backup/backup-file.ts-
bak \
-v <full-path-to-config-only-file>:/docker/config/config.json:ro \
-e LICENSE_KEY=<key> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Docker-Compose

```
version: '3.2'
services:
    tableau-server:
         hostname: localhost
         volumes:
              - <your-tsm-command-file>:/docker/config/tsm-com-
mands:ro
              - <your-config-file >:/docker/config/config.json:ro
         ports:
              - "8080:8080"
         image: ${IMAGE_NAME}
         environment:
              - LICENSE_KEY=<license-key>
```

Tableau Server in a Container - Troubleshooting

Introduction

Tableau Server in a Container is Tableau's first container-based server offering. Tableau Server in a Container is an all-in-one Tableau Server instance running inside of a Linux Docker container. In other words, a Tableau Server in a Container image is a docker image that runs an entire self-contained Tableau Server application. Tableau Server in a Container is our first of many steps to support Tableau Server running in container-based environments. The easiest way to understand the concept of Tableau Server in a Container is to think of it like a VM with Tableau Server pre-installed. The image is based on a UBI 8 image (CentOS 7.x for version 2022.1 and earlier) and runs `supervisord` (instead of `systemd`) inside the container. When the container starts `supervisord`, it will immediately attempt to initialize and start Tableau Server. Much of the documentation here aims to describe how to provide configuration and leverage automation so you can run Tableau Server in Docker environments.

The Tableau Server in a Container Image Setup Tool helps you create and customize container images to include custom packages and artifacts. One of the primary functions of the tool is to build the container image and install custom data connectors.

To test out the Tableau Server in a Container Image quickly in a prrof-of-concept scenarios, see Tableau Server in a Container - Quick Start.

Limitations

- Tableau Server in a Container only supports license activation using Server ATR, which requires the container to have internet access. Therefore, offline activation in an air-gapped environment is not possible.
- Tableau Server in a Container does not currently support the Resource Monitoring Tool (RMT) agent.
- Kerberos is not supported in Tableau Server in a Container

Troubleshooting

If you encounter issues when running Tableau Server, there are a number of avenues one can pursue to find a solution. This section covers general Tableau Server troubleshooting advice, such as where to find logs and what they mean. It also covers some specific known scenarios and mitigation paths.

If you are working with Tableau Support to debug an issue, it can be helpful if you provide the following:

- Tableau Server logs (gathering these logs is explained below).
- Docker container `stdout` logs.
- Dockerfile of Tableau Server (if any customizations have been made).
- Deployment configuration including:
    - `Kubeconfig` (or any equivalent deployment config).
    - Static configuration files that configure the Tableau Server Container.

Installation and initialization failures

If you are initializing Tableau Server for the first time, or you have done a fresh install inside a container, the server will not recover by simply restarting the container. Every installation attempt should use a clean data directory. This may mean deleting persistent volume data from previous container runs. If you do this make sure to save logs and information that could be useful for debugging.

Debugging failing installation

Tableau Server containers are designed to exit when an installation failure is encountered. This pattern makes it easy to automate and identify when an install failure has occurred. However, it can make debugging challenging because the container will exit and not leave any examinable runtime state. If you would like to have a debug session inside a running container that is failing during initialization, follow these steps:

1. Prepare a fresh Tableau Server in a Container deployment.
2. Configure the container to run with the `TSM_ONLY=1` environment variable. The `TSM_ONLY=1` environment variable tells Tableau Server to only initialize TSM. This is the equivalent of just running the `initialize-tsm` script in a standard, non-container installation.
3. Run the Tableau Server container.
4. Open a shell inside the container.
5. You can now run TSM commands, even though Tableau Server has not been initialized. To resume the automation that takes place normally during initialization, execute the `tsm-commands` script: `"${DOCKER_CONFIG}"/config/tsm-commands`

Tableau Support and Kubernetes

Tableau Server in a Container can be run using Kubernetes, but it is not a requirement to do so. Our expectation is that most customers will use Kubernetes or one of its associated managed cloud environments (EKS, AKS or GKS) to run and manage Tableau Server in a Container.

Kubernetes can be a complex environment to run and debug and often includes dependencies on individual companies infrastructure and setup. Because of this, Tableau Support cannot help customers resolve Kubernetes (or infrastructure deployment) issues associated with running Tableau Server in a Container. However, Tableau does support running Tableau Server in a Docker container. Therefore, if you are having issues with running Tableau Server in a Container using Kubernetes, Tableau Support can only go as far as validating that the Docker container works properly by itself.

For more information on how to run Tableau Server in a Container using Kubernetes, see this Github site: https://github.com/tableau/tableau-server-in-kubernetes.

Logs

Logs are an essential resource for finding, understanding, and solving problems in Tableau Server. They are useful for helping our support teams find the root the cause of issues you encounter. Logs may also be useful for your own debugging and troubleshooting.

Extracting All Logs

If you need to extract all logs for further debugging or to send to our support teams, there are a couple methods for retrieving this information.

Ziplogs

TSM can create a compressed archive containing all relevant server logs. You can trigger this by running the `tsm maintenance ziplogs` command. When the command completes it will report the filepath of the log archive. You will need to copy the archive using whatever file transfer method works best for your situation. For details on ziplogs, see tsm maintenance ziplogs.

Example command executed inside container:

```
tsm maintenance ziplogs
```

Manual Tar Command

If you cannot run the ziplogs command, for example, if the server does not manage to reach a consistent state, you can still retrieve the logs by executing a tar command inside the container. You will need to copy out the archive using whatever file transfer method works best for your situation.

Example command executed inside container (writes the tar to a temp directory in the container's data directory):

```
tar -zcvf /var/opt/tableau/tableau_server/temp/<archive_name>.tar.gz
\
/var/opt/tableau/tableau_server/data/tabsvc/logs/. \
/var/opt/tableau/tableau_server/supervisord/ \
/var/opt/tableau/tableau_server/data/tabsvc/config/ \
/docker/.metadata.conf \
--exclude='*keystores' --exclude='*.jks' --exclude='*.tks' \
--exclude='*asset_keys.yml' --exclude='*.ks' --exclude='*.ts' \
--exclude='*.crt' --exclude='*cacerts' --exclude='*.key'
```

Navigating Logs and Debugging Tips

There are common steps to diagnosing most problems in Tableau Server. If you are thinking of taking a look at your server logs, it can be helpful to break down what information to look for depending on where in the server life cycle the error occurred.

Starting The Container (initial / install)

If the container is crashing immediately or failing to install or initialize, check the following resources:

Container stdout

Examine `stdout` for the docker container. This is most often accessible by looking at the container output collected by your container orchestration system (for example, Kubernetes). Because Tableau Server is a multi-process system running inside a container, stdout is often not useful and will not report the root cause of the problem, unless there are catastrophic failures on startup. It is recommended you check `stdout` of the failing container before digging further into Tableau Server logs.

Example:

```
docker logs <container-name>
```

Tableau Server Container Startup Log

The Tableau Server Container startup log captures output from automation that is initializing, configuring, and starting Tableau Server. If you find your container is encountering issues while booting or running for the first time, this is the first log to check:

```
/var/opt/tableau/tableau_server/supervisord/run-tableau-server.log
```

Check the bottom of the log and see if there is a reported failure. Sometimes the error will be reported and be immediately obvious from the log. If the error is not clear from the log, it's possible the root cause is only visible in a stage-specific or service-specific log file. The logs listed below cover these possibilities.

Tableau Server Install Log

If the startup log indicates there was an issue with the automation handling the initialize TSM stage, check this log:

```
/var/opt/tableau/tableau_server/logs/app-install.log
```

Tableau Server Controller Log

If the startup log indicates there was an issue with the initializing and starting Server stage (CLI only), check the tabadmincontroller service log:

```
/var/opt/tableau/tableau_server-
/data/tabsvc/logs/tabadmincontroller/tabadmincontroller_node1-0.log
```

This log file is for a specific service called tabadmincontroller. Tabadmincontroller is responsible for orchestrating the initialization and startup functionality in Server. This log can be complex and verbose. Errors in this log file may still not point to the root cause. Sometimes the errors are caused by services that tabadmincontroller is relying on to complete a certain task. Check the Server runtime section below for more details.

Service Logs - Server Runtime

If Tableau Server encounters issues during normal runtime or issues with services failing to complete tasks or are down, you can check service logs for more information. Every service running as part of Tableau Server has a service log file. If you know which service you would like to examine, you can find that service's logs under this general directory:

```
/var/opt/tableau/tableau_server/data/tabsvc/logs/<service_name>
```

Provide the name of the service in the `<service_name>` arg of the file path. Any service can write multiple kinds of logs files. Also, if you have more than one of the same service running (more than one instance), all service logs will be written into the same service directory.

General Service-Specific Log File Classifications

This table covers the most common Service Log file names, types, and descriptions for Tableau Server services. The "Failure types" column indicates which log files are likely to be useful in a given failure scenario.

Name	Filename format	Description	Failure types	Example
Control-App	`control_<service_name>_<node_id>-<instance_id>.log`	Contains information of the control-app process which is responsible for installing and configuring a service. This is often the first log written related to a service. For service install and configure failures, look here first.	Install, Configure, Status	`control_backgrounder_node1-0.log`
Service log	`<service_name>_<node_id>-<instance_id>.log`	Primary log for a running service. Most often this log contains output from the spring/-java application layer.	Start, Runtime, Status	`backgrounder_node1-1.log`
Stdout log	`stdout_<service_name>_<instance_id>.log`	Contains `stdout` output for the service. Most services do not output much content to `stdout` and instead write to the primary log. Sometimes this log can contain useful information when a service	Start, Stop	`stdout_backgrounder_0.log`

Name	Filename format	Description	Failure types	Example
		exits.		
NativeAPI log	`nativeapi_<service_name>_<instance_id>.txt`	Some services run a native code layer. This log captures that portion of the application's runtime.	Licensing, Start, Runtime, Status	`nativeapi_backgrounder_1-1_2021_05_10_00_00_00.txt`
Tomcat log	`tomcat_<service_name>_<node_id>-<instance_id>.log`	This is only for services that run inside a tomcat container and contains tomcat logs. It rarely provides information regarding service failure. It can be useful to debug some network issues.	Network, Start	`tomcat_backgrounder_node1-0.2021-05-10.log`

Stopped Container

If the container is stopped or is otherwise difficult to execute commands in, you can still look at the logs if the data directory of the server is externalized to a mounted volume. Otherwise, only the `stdout` of the container will be examinable in the container's orchestration system, which is often not going to contain the root cause.

Failure to set authentication properties

There appears to be an issue with setting authentication properties in Tableau Server without the identity store getting set first. To work around this issue just set the identity store in the pre-initialization hook.

1. Create a file called `./customer-files/pre_init_command` in the Tableau Server Image Build Tool customer-files directory and edit it to contain:

   ```
   #!/bin/bash
   tsm configuration set -k wgserver.authenticate -v local --
   force-keys
   ```

2. Set the script to be executable.

   ```
   chmod +x ./customer-files/pre_init_command
   ```

3. Build and run the image.

Failure during fresh startup (e.g. why isn't Tableau Server starting?)

- If you are encountering issues with Tableau Server initializing or starting up there are a number of troubleshooting options that may help uncover the issue.
- If the container cannot start at all, you'll want to check the stdout from the PID 1 process using `docker logs <container-name>` command.
- If the container is running but Tableau Server does not seem to be initializing or running properly, the second place to check for errors is this file:

```
${DATA_DIR}/supervisord/run-tableau-server.log
```

Example:

```
docker exec -it <container-name> bash -c 'cat $DATA_DIR/su-
pervisord/run-tableau-server.log'
```

This log file contains all events orchestrated by the tableau container initializing service that is handling the startup of tableau server as well as executing any setup scripts or custom configuration that you may have provided in the container. Most startup errors will report issues here. Sometimes if the error is related to a TSM or Tableau Server process it will suggest another log file to look at for more detailed information.

Failure during restart or starting a container with existing data

Server Won't Start PostGRES (or other processes)

When data is persisted outside the container and you are starting another Tableau Server in a Container image instance using that old data, it is important to note that the internal hostname of the new container must match the hostname of the container that initialized the persisted data. Tableau Server does not handle dynamic hostname changes well and starting up a new container with a different internal hostname is effectively causing that scenario.

The remedy is to simply make sure the container's hostname is set to the same value as the container that was previously running with that data. This is not to be confused with Multi-node, workers can (and probably should) have different hostnames from each other. What matters is when a given container is restarted or killed the subsequent container must have the same hostname as its predecessor.

Deployment Configuration Examples

Docker

Tableau Server in a Container Basic Usage

```
docker run \
-e LICENSE_KEY=<key>
-p 8080:8080
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Tableau Server in a Container Basic Usage with Automated Initial Admin User

```
docker run \
-e LICENSE_KEY=<key> \
-e TABLEAU_USERNAME=<myadmin> \
-e TABLEAU_PASSWORD_FILE=/etc/tableau-admin-secret \
-v <full-path-to-pw-file>:/etc/tableau-admin-secret \
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

TSM only mode

```
docker run \
-e TSM_ONLY=1 \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Multi-Node Basic Usage

Initial Node

Option 1: Use this if the server configuration (`CONFIG_FILE`) specifies a multi-node topology:

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-v <full-path-to-config-file>:/docker/config/config.json:ro \
-e LICENSE_KEY=<key> \
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \
--hostname=<name-of-host-machine> \
-d <Tableau Server in a Container image ID or tag>
```

Option 2: Use this if you want a multi-node deployment even if server configuration does not specify multi-node topology:

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-e LICENSE_KEY=<key> -e ALWAYS_WRITE_BOOTSTRAP_FILE=1 \
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \
--hostname=<name-of-host-machine> \
-d <Tableau Server in a Container image ID or tag>
```

Additional node

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-e BOOTSTRAP_INSTALL=1 \
-p 8080:8080 -p 8800-9000:8800-9000 \
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Externalize Data Usage

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
-e LICENSE_KEY=<key> \
---hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Init-Container Basic Usage

Init Container

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
-e LICENSE_KEY=<key> \
-e INIT_CONTAINER=1 \
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Run Container

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Basic Restore from Backup Single-Node

```
docker run \
-v <full-path-to-backup-file>:/docker/config/backup/backup-file.ts-
bak \
-v <full-path-to-config-only-file>:/docker/config/config.json:ro \
-e LICENSE_KEY=<key> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Docker-Compose

```
version: '3.2'
services:
    tableau-server:
```

```
          hostname: localhost
          volumes:
               - <your-tsm-command-file>:/docker/config/tsm-com-
mands:ro
               - <your-config-file >:/docker/config/config.json:ro
          ports:
               - "8080:8080"
          image: ${IMAGE_NAME}
          environment:
               - LICENSE_KEY=<license-key>
```

Tableau Server in a Container - Quick Start

Introduction

Tableau Server in a Container is Tableau's first container-based server offering. Tableau Server in a Container is an all-in-one Tableau Server instance running inside of a Linux Docker container. In other words, a Tableau Server in a Container image is a docker image that runs an entire self-contained Tableau Server application. Tableau Server in a Container is our first of many steps to support Tableau Server running in container-based environments. The easiest way to understand the concept of Tableau Server in a Container is to think of it like a VM with Tableau Server pre-installed. The image is based on a UBI 8 image (CentOS 7 for version 2022.1 and earlier) and runs `supervisord` (instead of `systemd`) inside the container. When the container starts `supervisord`, it will immediately attempt to initialize and start Tableau Server. Much of the documentation here aims to describe how to provide configuration and leverage automation so you can run Tableau Server in Docker environments.

The Tableau Server in a Container Image Setup Tool helps you create and customize container images to include custom packages and artifacts. One of the primary functions of the tool is to build the container image and install custom data connectors. For detailed information on using the Setup tool, see Tableau Server in a Container - Using the Setup Tool.

Limitations for Tableau Server in a Container

- Tableau Server in a Container only supports license activation using Server ATR. Offline activation using Server ATR is supported in 2023.1 and later. This functionality is available in Containers but requires extra steps and approval. If you need to run Tableau Server in a Container in an air-gapped or offline environment, contact your Account representative for more information.
- Tableau Server in a Container does not currently support the Resource Monitoring Tool (RMT) agent.
- Kerberos is not supported in Tableau Server in a Container.

Tableau Server in a Container - Quick Start Guide

This topic provides all of the steps required to perform a basic, quick start deployment of a Tableau Server in a Container Image using the command line. The purpose of the configuration described here is to provide the quickest, simplest path to a Tableau Server on Linux installation running inside a container. For a more in-depth look at what Tableau Server in a Container has to offer, see Tableau Server in a Container Setup Tool and Tableau Server in a Container Image.

> **Important:** Do not use the following Quick Start procedure as a stand-alone resource for deploying Tableau Server in a production environment.

Before you begin

There are two basic steps required to use Tableau Server in a Container:

1. Building the Docker image

   Building the Tableau Server in a Container Docker image is only supported on a RHEL-based Linux system (RHEL, CentOS, or Amazon Linux 2). Building on any other Linux distributions may be possible but is currently untested and unsupported. Building images on macOS is not supported. The image created is based on a UBI 8 image (CentOS 7.x for version 2022.1 and earlier).

You must have Docker version 18.09 or later installed on the host in order to build the container images. In general, we recommend using the latest stable version of Docker. Some Linux distros only provide older versions of Docker in their software repositories, in which case you may need to install Docker from a different source. Docker versions earlier than version 18.09 do not include features that are required for Tableau Server in a Container.

2. Running the Docker image

   Production use of Tableau Server in a Container is only supported on Linux. For exploratory prototyping and testing work, any system that can run Linux-based Docker images should be able to run Tableau Server in a Container images (assuming it satisfies the hardware and operating systems requirements outlined in Before you install...).

   Tableau Server in a Container images require the same hardware resources as Tableau Server itself in order to run. For production deployments, follow the recommendations shown in Minimum production hardware recommendations. For exploratory work, follow the recommendations shown for Minimum installation hardware requirements. If you are deploying in a public cloud environment, see the links at the bottom of the Minimum Hardware Requirements and Recommendations for Tableau Server page for detailed recommendations on instance sizing. If using Docker Desktop on Windows or macOS for exploratory work, note that the default resource limits for containers are set far below the minimum requirements and will need to be increased to run the container successfully.

Note: Tableau Support cannot help with issues on unsupported platforms or configurations.

The steps below assume that you have a Linux system with Docker installed and you have downloaded the Tableau Server in a Container Setup Tool and a version compatible Tableau Server RPM installer.

1. Untar the Tableau Server in a Container Setup Tool

   The Tableau Server in a Container Setup Tool will be available as a tarball. To unpack the tar archive, simply use the following command:

   ```
   tar -xzf tableau-server-container-setup-tool-<VERSION>.tar.gz
   ```

2. Edit the registration file to provide your unique identifying information needed to register Tableau Server in accordance with the End User License Agreement. The file, `reg-info.json`, serves as a template for your required, uniquely identifiable registration information and is located in the top directory of the Tableau Server in a Container Setup Tool. This file is used to register the Tableau Server instance running in the image. Providing accurate information will ensure the registration process completes properly.

   The `eula` field value is pre-filled with "accept" to indicate you are accepting our End User License Agreement (EULA). You can view the EULA in the EULA directory of the build tool. As outlined in the EULA, you must submit a uniquely identifiable user registration when activating Tableau Server. When you are done editing the registration file, the other fields should have values that reflect your unique information. This file is used to register the Tableau Server instance running in the image. Providing accurate information will ensure the registration process completes and your submission meets the requirements of the license grant.

   > **Note:** You must accept the EULA to use Tableau Server. If you do not accept the EULA, you cannot run Tableau Server.

   The registration file template `reg-info.json` before editing:

   ```
   {
   "zip" : "<value>",
   "country" : "<value>",
   "city" : "<value>",
   ```

```
"last_name" : "<value>",
"industry" : "<value>",
"eula" : "accept",
"title" : "<value>",
"phone" : "<value>",
"company" : "<value>",
"state" : "<value>",
"department" : "<value>",
"first_name" : "<value>",
"email" : "<value>"
}
```

3. Build the Tableau Server in a Container Image

   Run the build-image script in the Tableau Server in a Container setup tool

   ```
   ./build-image --accepteula -i <Tableau Server Installer>.rpm
   ```

4. Run the Tableau Server in a Container Image

   a. Execute the Docker run command with all arguments filled out. You will need to provide the following information in the command line:

      - License key or license key file
      - Username and password for a new initial admin account (you will use this to sign into Tableau Server after it starts)
      - A static (internal) host name for computer
      - Name of the Tableau Server in a Container image

   b. When you have collected this information run this docker command on a minimum-spec host:

      ```
      docker run \
      -e LICENSE_KEY=<key> \
      -e TABLEAU_USERNAME=<username> \
      -e TABLEAU_PASSWORD=<password> \
      --hostname=<static (internal) name of host machine> \
      ```

```
-p 8080:8080 -d <Tableau Server in a Container image ID or
tag>
```

c. After 10-20 minutes Tableau Server will be initialized.

5. Access Tableau Server

   When Tableau Server is completely initialized, check that the server is running properly by opening a web browser and going to `http://<hostname>:8080` where `<host-name>` is the host name of the machine running the image.

You should have a running instance of Tableau Server in a Docker container at this point. For more advanced and customized deployments use the documentation below to guide you.

Deployment Configuration Examples

Docker

Tableau Server in a Container Basic Usage

```
docker run \
-e LICENSE_KEY=<key>
-p 8080:8080
-d <Tableau Server in a Container image ID or tag>
```

Tableau Server in a Container Basic Usage with Automated Initial Admin User

```
docker run \
-e LICENSE_KEY=<key> \
-e TABLEAU_USERNAME=<myadmin> \
-e TABLEAU_PASSWORD_FILE=/etc/tableau-admin-secret \
-v <full-path-to-pw-file>:/etc/tableau-admin-secret \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

TSM only mode

```
docker run \
-e TSM_ONLY=1 \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Multi-Node Basic Usage

Initial Node

Option 1: Use this if the server configuration (`CONFIG_FILE`) specifies a multi-node topology:

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-v <full-path-to-config-file>:/docker/config/config.json:ro \
-e LICENSE_KEY=<key> \
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Option 2: Use this if you want a multi-node deployment even if server configuration does not specify multi-node topology:

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-e LICENSE_KEY=<key> -e ALWAYS_WRITE_BOOTSTRAP_FILE=1 \
-p 8080:8080 -p 8800-9000:8800-9000 -p 27000-27010:27000-27010 \
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Additional node

```
docker run \
-v <network-shared-directory>:/docker/config/bootstrap \
-e BOOTSTRAP_INSTALL=1 \
-p 8080:8080 -p 8800-9000:8800-9000 \
--hostname=<static (internal) name of host machine> \
-d <Tableau Server in a Container image ID or tag>
```

Externalize Data Usage

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
-e LICENSE_KEY=<key> \
```

```
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Init-Container Basic Usage

Init Container

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
-e LICENSE_KEY=<key> \
-e INIT_CONTAINER=1 \
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Run Container

```
docker run \
-v <empty-data-dir>:/var/opt/tableau \
--hostname=<static (internal) name of host machine> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Basic Restore from Backup Single-Node

```
docker run \
-v <full-path-to-backup-file>:/docker/config/backup/backup-file.ts-
bak \
-v <full-path-to-config-only-file>:/docker/config/config.json:ro \
-e LICENSE_KEY=<key> \
-p 8080:8080 -d <Tableau Server in a Container image ID or tag>
```

Docker-Compose

```
version: '3.2'
services:
    tableau-server:
         hostname: localhost
         volumes:
              - <your-tsm-command-file>:/docker/config/tsm-com-
mands:ro
```

```
      - <your-config-file >:/docker/config/config.json:ro
ports:
      - "8080:8080"
image: ${IMAGE_NAME}
environment:
      - LICENSE_KEY=<license-key>
```

Post Installation Tasks

After you install Tableau Server, you can perform other configuration tasks, such as reviewing the security hardening checklist, configuring server event notifications, configuring the data cache for views, and configuring server crash reporting.

Security Hardening Checklist

The following list provides recommendations for improving the security ("hardening") of your Tableau Server installation.

> Looking for Tableau Server on Windows? See Security Hardening Checklist.

Installing security updates

Security updates are included in the latest versions and maintenance releases (MR) of Tableau Server. You cannot install security updates as patches. Rather, you must upgrade to a current version or MR to update Tableau Server with the latest security fixes.

Always reference the most current version of this topic after upgrading. The current version includes `/current/` in the topic URL.

For example, the US version URL is: https://help.tableau.com/current/server/en-us/security_harden.htm.

1. Update to the current version

We recommend that you always run the latest version of Tableau Server. Additionally, Tableau periodically publishes maintenance releases of Tableau Server that include fixes for known security vulnerabilities. (Information regarding known security vulnerabilities can be found on the Tableau Security Bulletins page and the Salesforce Security Advisories page.) We recommend that you review maintenance release notifications to determine whether you should install them.

To get the latest version or maintenance release of Tableau Server, visit the Customer Portal page.

2. Configure SSL/TLS with a valid, trusted certificate

Secure Sockets Layer (SSL/TLS) is essential for helping to protect the security of communications with Tableau Server. Configure Tableau Server with a valid, trusted certificate (not a self-signed certificate) so that Tableau Desktop, mobile devices, and web clients can connect to the server over a secured connection. For more information, see SSL.

3. Disable older versions of TLS

Tableau Server uses TLS to authenticate and encrypt many connections between components and with external clients. External clients, such as browsers, Tableau Desktop, Tableau Mobile connect to Tableau using TLS over HTTPS. Transport layer security (TLS) is an improved version of SSL. In fact, older versions of SSL (SSL v2 and SSL v3) are no longer considered to be adequately secure communication standards. As a result, Tableau Server does not allow external clients to use SSL v2 or SSL v3 protocols to connect.

We recommend that you allow external clients to connect to Tableau Server with TLS v1.3 and TLS v1.2.

TLS v1.2 is still regarded as a secure protocol and many clients (including Tableau Desktop) do not yet support TLS v1.3.

TLS v1.3 capable clients will negotiate TLS v1.3 even if TLS v1.2 is supported by the server.

The following tsm command enables TLS v1.2 and v1.3 (using the "all" parameter) and disables SSL v2, SSL v3, TLS v1, and TLS v1.1 (by prepending the minus [-] character to a given protocol). TLS v1.3 is not yet supported by all components of Tableau Server.

```
tsm configuration set -k ssl.protocols -v 'all -SSLv2 -SSLv3 -TLSv1
-TLSv1.1'
```

```
tsm pending-changes apply
```

To modify the protocols that govern SSL for the Tableau Server PostgreSQL repository, see pgsql.ssl.ciphersuite.

You can also modify the default list of cipher suites that Tableau Server uses for SSL/TLS sessions. For more information see the *ssl.ciphersuite* section at tsm configuration set Options.

4. Configure SSL encryption for internal traffic

Configure Tableau Server to use SSL to encrypt all traffic between the Postgres repository and other server components. By default, SSL is disabled for communications between server components and the repository. We recommend enabling internal SSL for all instances of Tableau Server, even single-server installations. Enabling internal SSL is especially important for multi-node deployments. See Configure SSL for Internal Postgres Communication.

5. Enable firewall protection

Tableau Server was designed to operate inside a protected internal network.

Important: Do not run Tableau Server, or any components of Tableau Server on the internet or in a DMZ. Tableau Server must be run within the corporate network protected

> by an internet firewall. We recommend configuring a reverse proxy solution for internet clients that need to connect to Tableau Server. See Configuring Proxies and Load Balancers for Tableau Server.

A local firewall should be enabled on the operating system to protect Tableau Server in single and multi-node deployments. In a distributed (multi-node) installation of Tableau Server, communication between nodes does not use secure communication. Therefore, you should enable firewalls on the computers that host Tableau Server. See Configure Local Firewall.

To prevent a passive attacker from observing communications between nodes, configure a segregated virtual LAN or other network layer security solution.

See Tableau Services Manager Ports to understand which ports and services Tableau Server requires.

6. Restrict access to the server computer and to important directories

Tableau Server configuration files and log files can contain information that is valuable to an attacker. Therefore, restrict physical access to the machine that is running Tableau Server. In addition, make sure that only authorized and trusted users have access to the Tableau Server files in the `/var/opt/tableau/tableau_server/` directory.

7. Generate fresh secrets and tokens

Any Tableau Server service that communicates with repository or the cache server must first authenticate with a secret token. The secret token is generated during Tableau Server setup. The encryption key that internal SSL uses to encrypt traffic to Postgres repository is also generated at during setup.

We recommend that after you install Tableau Server, you generate new encryption keys for your deployment.

These security assets can be regenerated with the `tsm security regenerate-internal-tokens` command.

Run the following commands:

```
tsm security regenerate-internal-tokens

tsm pending-changes apply
```

8. Disable services that you're not using

To minimize the attack surface of the Tableau Server, disable any connection points that are not needed.

JMX Service

JMX is disabled by default. If it's enabled but you're not using it, you should disable it by using the following:

```
tsm configuration set -k service.jmx_enabled -v false

tsm pending-changes apply
```

9. Verify session lifetime configuration

By default, Tableau Server does not have an absolute session timeout. This means that browser-based client (Web authoring) sessions can remain open indefinitely if the Tableau Server inactivity timeout is not exceeded. The default inactivity timeout is 240 minutes.

If your security policy requires it, you can set an absolute session timeout. Be sure to set your absolute session timeout in a range that allows the longest-running extract uploads or workbook publishing operations in your organization. Setting the session timeout too low may result in extract and publishing failures for long-running operations.

To set the session timeout run the following commands:

`tsm configuration set -k wgserver.session.apply_lifetime_limit -v true`

`tsm configuration set -k wgserver.session.lifetime_limit -v` *value*, where *value* is the number of minutes. The default is 1440, which is 24 hours.

`tsm configuration set -k wgserver.session.idle_limit -v` *value*, where *value* is the number of minutes. The default is 240.

`tsm pending-changes apply`

Sessions for connected clients (Tableau Desktop, Tableau Mobile, Tableau Prep Builder, Bridge, and personal access tokens) use OAuth tokens to keep users logged in by re-establishing a session. You can disable this behavior if you want all Tableau client sessions to be solely governed by the browser-based session limits controlled by the commands above. See Disable Automatic Client Authentication.

10. Configure a server allowlist for file-based data sources

As of October 2023 Tableau Server releases, default file-based access behavior has changed. Previously, Tableau Server allowed authorized Tableau Server users to build workbooks that use files on the server as file-based data sources (such as spreadsheets). With the October 2023 releases, access to files stored on Tableau or on remote shares must be specifically configured on Tableau Server using the setting described here.

This setting allows you to limit access by the `tableau` system account only to those directories that you specify.

To configure access to shared files, you must configure allowlist functionality. This lets you limit `tableau` account access to just the directory paths where you host data files.

1. On the computer running Tableau Server, identify the directories where you will host data source files.

> **Important** Make sure the file paths you specify in this setting exist and are accessible by the system account.

2. Run the following commands:

   `tsm configuration set -k native_api.allowed_paths -v "path"`, where *path* is the directory to add to the allowlist. All subdirectories of the specified path will be added to the allowlist. You must add a trailing backslash to the specified path. If you want to specify multiple paths, separate them with a semicolon, as in this example:

   ```
   tsm configuration set -k native_api.allowed_paths -v "/data-sources;/HR/data/"

   tsm pending-changes apply
   ```

11. Enable HTTP Strict Transport Security for web browser clients

HTTP Strict Transport Security (HSTS) is a policy configured on web application services, such as Tableau Server. When a conforming browser encounters a web application running HSTS, then all communications with the service must be over a secured (HTTPS) connection. HSTS is supported by major browsers.

For more information about how HSTS works and the browsers that support it, see The Open Web Application Security Project web page, HTTP Strict Transport Security Cheat Sheet.

To enable HSTS, run the following commands on Tableau Server:

```
tsm configuration set -k gateway.http.hsts -v true
```

By default, HSTS policy is set for one year (31536000 seconds). This time period specifies the amount of time in which the browser will access the server over HTTPS. You should consider setting a short max-age during initial roll-out of HSTS. To change this time period, run `tsm configuration set -k gateway.http.hsts_options -v max-`

`age=<seconds>`. For example, to set HSTS policy time period to 30 days, enter `tsm configuration set -k gateway.http.hsts_options -v max-age=2592000`.

```
tsm pending-changes apply
```

12. Disable Guest access

Core-based licenses of Tableau Server include a Guest user option, which allows any user in your organization to see and interact with Tableau views embedded in web pages.

Guest user access is enabled by default on Tableau Servers deployed with core-based licensing.

Guest access allows users to see embedded views. The Guest user cannot browse the Tableau Server interface or see server interface elements in the view, such as user name, account settings, comments, and so on.

If your organization has deployed Tableau Server with core licensing and Guest access is not required, then disable Guest access.

You can disable Guest access at the server or site level.

You must be a server administrator to disable the Guest account at either the server or the site level.

To disable Guest access at the server level:

1. In the site menu, click **Manage All Sites** and then click **Settings** > **General**.
2. For **Guest Access**, clear the **Enable Guest account** check box.
3. Click **Save**.

To disable Guest access for a site:

1. In the site menu, select a site.
2. Click **Settings**, and on the Settings page, clear the **Enable Guest account** check box.

For more information, see Guest User.

13. Set referrer-policy HTTP header to 'same-origin'

Beginning in 2019.2, Tableau Server includes the ability to configure Referrer-Policy HTTP header behavior. This policy is enabled with a default behavior that will include the origin URL for all "secure as" connections (`no-referrer-when-downgrade`), which sends origin referrer information only to like connections (HTTP to HTTP) or those that are more secure (HTTP to HTTPS).

However, we recommend setting this value to `same-origin`, which only sends referrer information to same-site origins. Requests from outside the site will not receive referrer information.

To update the referrer-policy to `same-origin`, run the following commands:

```
tsm configuration set -k gateway.http.referrer_policy -v same-origin
```

```
tsm pending-changes apply
```

For more information about configuring additional headers to improve security, see HTTP Response Headers.

14. Configure TLS for SMTP connection

Beginning in 2019.4, Tableau Server includes the ability to configure TLS for the SMTP connection. Tableau Server only supports STARTTLS (Opportunistic or Explicit TLS).

Tableau Server can be optionally configured to connect to a mail server. After configuring SMTP, Tableau Server can be configured to email server administrators about system failures, and email server users about subscribed views and data-driven alerts.

To configure TLS for SMTP:

1. Upload a compatible certificate to Tableau Server. See tsm security custom-cert add.
2. Configure TLS connection using TSM CLI.

   Run the following TSM commands to enable and force TLS connections to the SMTP server and to enable certificate verification.

   ```
   tsm configuration set -k svcmonitor.notification.smtp.ssl_
   enabled -v true
   ```

   ```
   tsm configuration set -k svcmonitor.notification.smtp.ssl_
   required -v true
   ```

   ```
   tsm configuration set -k svcmonitor.notification.smtp.ssl_
   check_server_identity -v true
   ```

   By default, Tableau Server will support TLS versions 1, 1.1, and 1.2, but we recommend that you specify the highest TLS version that the SMTP server supports.

   Run the following command to set the version. Valid values are `SSLv2Hello`, `SSLv3`, `TLSv1`, `TLSv1.1`, and `TLSv1.2`. The following example sets the TLS version to version 1.2.:

   ```
   tsm configuration set -k svcmonitor.notification.smtp.ssl_ver-
   sions -v "TLSv1.2"
   ```

   For more information about other TLS configuration options, see Configure SMTP Setup.

3. Restart Tableau Server to apply changes. Run the following command:

   ```
   tsm pending-changes apply
   ```

15. Configure SSL for LDAP

If your Tableau Server deployment is configured to use a generic LDAP external identity store, we recommend configuring SSL to protect authentication between Tableau Server and your LDAP server. See Configure Encrypted Channel to LDAP External Identity Store.

If your Tableau Server deployment is configured to use Active Directory, we recommend enabling Kerberos to protect authentication traffic. See Kerberos.

Change List

Date	Change
May 2018	Added clarification: Do not disable REST API in organizations that are running Tableau Prep.
May 2019	Added recommendation for referrer-policy HTTP header.
June 2019	Removed recommendation to disable Triple-DES. As of version 2019.3, Triple-DES is no longer a default supported cipher for SSL. See What's Changed - Things to Know Before You Upgrade.
January 2020	Added recommendation to configure TLS for SMTP.
February 2020	Added recommendation to configure SSL for LDAP server.
May 2020	Added TLS v1.3 to the disabled list of TLS ciphers. Added clarification to introduction about topic versioning.
October 2020	Added TLS v1.3 as a default supported cipher.
January 2021	Added clarification: All products enabled by the Data Management license require REST API.
February 2021	Removed recommendation to disable REST API. The API is now used internally by Tableau Server and disabling it may limit functionality.

Configure SMTP Setup

Tableau Server can email server administrators about system failures, and email server users about subscribed views and data-driven alerts. First, however, you need to configure the SMTP server that Tableau Server uses to send email. After configuring SMTP, complete the steps to configure notifications (Configure Server Event Notification), then when you start or

restart the server, it will trigger an email notification, which confirms that you have set up notifications correctly.

Configuring SMTP requires that you restart Tableau Server services.

Secure SMTP

To enable and configure TLS for SMTP, you must use the TSM CLI as described in this topic. Tableau Server only supports STARTTLS (Opportunistic or Explicit TLS).

If your organization does not use public certificates for verifying TLS connections, then you can upload a private certificate to Tableau Server to verify trusted connections. For more information, see the tsm security custom-cert add command.

You may also configure SMTP TLS for encryption-only by disabling the certificate validation process. For more information, see the section, *Configuration file reference*,in the *Use the TSM CLI* tab below.

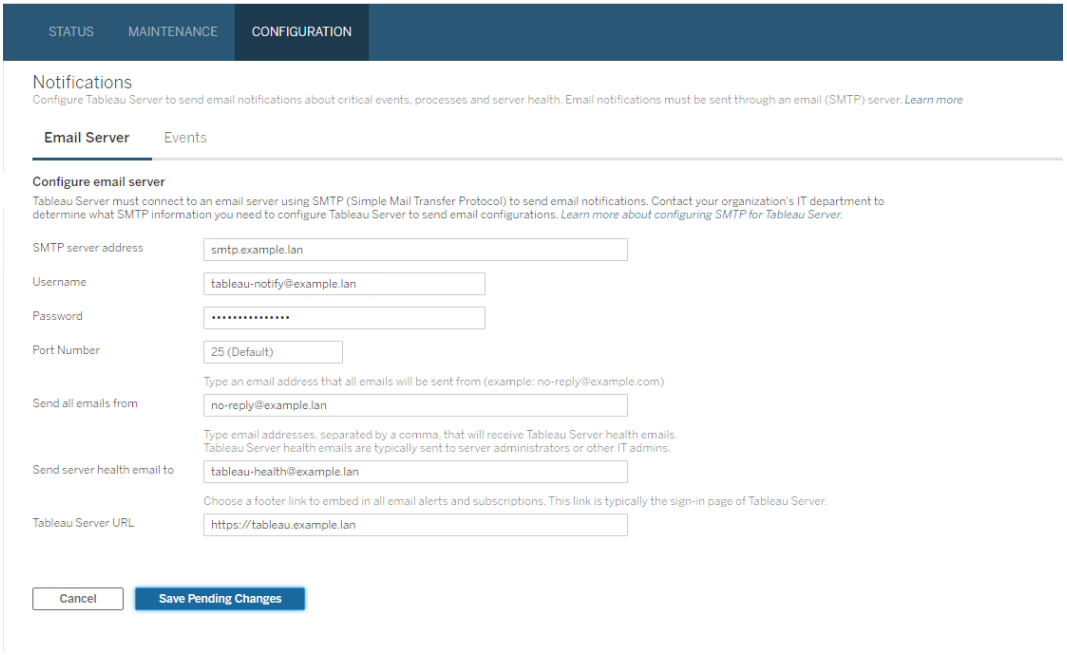
Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click **Notifications**on the **Configuration** tab and click **Email Server**.

3. Enter the SMTP configuration information for your organization:



4. Click **Save Pending Changes** after you've entered your configuration information.

5. Click **Pending Changes** at the top of the page:

6. Click **Apply Changes and Restart**.

7. Run the `tsm email test-smtp-connection` to view and verify the connection configuration. See tsm email test-smtp-connection.

Use the TSM CLI

For the initial configuration of SMTP, we recommend that you use the configuration file template below to create a json file. You can also set any single configuration key listed below with the syntax described in tsm configuration set.

1. Copy the following json template to a file.

   Important: The template below includes common options for most deployments. After you copy the template to a text file, you must edit the option values for your SMTP server requirements. You may need to remove or add options. See the reference section that follows for more information about all supported SMTP key options.

   ```
   {
   “configKeys”: {
          "svcmonitor.notification.smtp.server": "SMTP server host name",
          "svcmonitor.notification.smtp.send_account": "SMTP user name",
          "svcmonitor.notification.smtp.port": 443,
          "svcmonitor.notification.smtp.password": "SMTP user account password",
          "svcmonitor.notification.smtp.ssl_enabled": true,
          "svcmonitor.notification.smtp.from_address": "From email address",
          "svcmonitor.notification.smtp.target_addresses": "To email address1,address2",
          "svcmonitor.notification.smtp.canonical_url": "Tableau Server URL"
          }
   }
   ```

2. Run the `tsm settings import -f file.json` to pass the json file with the appropriate values to Tableau Services Manager to configure Tableau Server for SMTP. Tableau Services Manager will validate the entity values.

3. Run the `tsm pending-changes apply` command to apply the changes. See tsm pending-changes apply.

4. Run the `tsm email test-smtp-connection` to view and verify the connection configuration. See tsm email test-smtp-connection.

SMTP CLI configuration reference

This table lists all of the options that can be used to configure SMTP with TSM CLI.

Option	Description
`svc-mon-itor.no-tification.smtp.server`	Address of SMTP server. Example: `"svc-monitor.notification.smtp.server": "mail.example.com"`
`svc-mon-itor.notification.smtp.send_account`	User name for SMTP account.
`svc-mon-itor.notification.smtp.port`	Port number for SMTP server. The default is 25.
`svc-mon-itor.no-tification.smtp.password`	Password for SMTP server account. Example: `"svc-mon-itor.no-tific-ation.smtp.password":"password"`
`svc-`	Specifies whether the connection to the SMTP

Option	Description
`monitor.notification.smtp.ssl_enabled`	server is encrypted. The default is false.
`svc-monitor.notification.smtp.ssl_required`	If enabled, Tableau Server will refuse to connect to SMTP servers without using TLS. The `svc-monitor.notification.smtp.ssl_enabled` option must also be set to true. The default is false.
`svc-monitor.notification.smtp.ssl_check_server_identity`	If set to true, Tableau Server will check the SMTP server identity as specified by RFC 2595. These additional checks based on the content of the server's certificate are intended to prevent man-in-the-middle attacks. The default is false.
`svc-monitor.notification.smtp.ssl_trust_all_hosts`	When using TLS, trust certificates from all mail servers, ignoring the validity of the certificate's chain of trust. By setting this key to true, TLS will be used only to encrypt the traffic to the SMTP host. The default is false.
`svc-monitor.notification.smtp.ssl_ciphers`	The default and supported sets of cipher suites is defined by the version of JDK that is installed with Tableau Server. See the section below, TLS ciphers, for a list of supported and default ciphers.

Option	Description
	To update the cipher suites used by Tableau Server for SMTP TLS connections, enter a white space-separated list of cipher suites for this value. For example, "`TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384`".
`svc-monitor.notification.smtp.ssl_versions`	The default TLS versions enabled on this version of Tableau Server are TLSv1, TLSv1.1, TLSv1.2 and TLSv1.3. TLS version support is defined by the version of JDK that is installed with Tableau Server. Supported versions of TLS are `SSLv2Hello, SSLv3, TLSv1, TLSv1.1, TLSv1.2, TLSv1.3`. To update the versions used by Tableau Server for SMTP TLS connections, enter a white space-separated list of versions for this value. For example, "`TLSv1.2 TLSv1.3`".
`svc-monitor.notification.smtp.from_address`	Email address that will send an notification if there's a system failure. The email address must have valid syntax (for example, ITalerts@bigco.-com or noreply@mycompany), but it does not have to be an actual email account on Tableau Server. (Some SMTP servers may require an actual email account, however.)

Option	Description
	Note: You can override the system-wide email address on a per-site basis. For more information, see What is a site. Example: `"svcmonitor.notification.smtp.from_address": "donot-reply@example.com"`
`svc-mon-itor.no-tification.smtp.target_addresses`	Email address to receive notifications. If email notifications are enabled, you need to include at least one address. Separate multiple addresses with commas. Example: `"svc-monitor.notification.smtp.target_addresses": "iluvdata@example.com"`
`svc-mon-itor.no-tification.smtp.canonical_url`	URL of the Tableau Server. Enter `http://` or `https://`, followed by the name or IP address of the Tableau server. Used in the footer of subscription email. Example: `"svc-mon-itor.notification.smtp.canonical_url": "http://myserver.example.com"`

TLS ciphers

The following is a list of TLS ciphers that are supported by the JDK that is included with Tableau Server. In this version of Tableau Server, all of these ciphers are enabled by default. You can specify a custom list of ciphers for your SMTP configuration by entering a white-space separated list with the option, `svcmonitor.notification.smtp.ssl_ciphers`, as described in the table above.

TLS_RSA_WITH_AES_128_CBC_SHA256	TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384	TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA	TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256	TLS_DHE_DSS_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS_EMPTY_RENEGOTIATION_INFO_SCSV
TLS_ECDH_ECDSA_WITH_AES_256_	TLS_ECDHE_ECDSA_WITH_AES_256_

CBC_SHA384	CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256	TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA	TLS_DHE_DSS_WITH_AES_128_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_RSA_WITH_AES_128_CBC_SHA	TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	TLS_AES_256_GCM_SHA384

TLS_AES_128_GCM_SHA256	

Files and Permissions in TSM

This topic covers the permissions requirements needed for Tableau Services Manager (TSM) to access and use files. This information is intended for server administrators. This topic does *not* address permissions used for managing users and content within Tableau Server (permissions for content and users). For information related to these permissions, see Permissions.

During installation of TSM and Tableau Server, an unprivileged user (*tableau*) is created and added to a server authorized group (*tableau*). This user account enables the work done by TSM and Tableau Server processes. You can change the user and group during installation. For more information, see Identity Store.

Permissions requirements for TSM apply to both files, and to the directories in which the files are placed. When TSM creates and manages files, the files get put into specific default locations with the necessary permissions and you don't need to worry about setting permissions. When you create, copy, or move files yourself, or when you put files into non-default locations, you need to be aware of permission requirements so that TSM can properly access the files. Common cases (For information about using non-default locations, see tsm File Paths.)

General rules for permissions and TSM are:

- Files—If the *tableau* group has access to a file (if it is the group owner and has read access to the file), the users in the group have access to the file. An alternate approach is to give "other" read access.
- Directories—If the *tableau* group has read and execute access to the directory that contains a file, and any parent directories of that directory, the users in the group have access to the file.

Situations that may require you to adjust permissions include server backup files and site import archives that you copy from a different computer or to a non-default location, customization files such as logos or images, and security certificates such as SSL certificates.

For example, if you migrate from Tableau Server on Windows to Tableau Server on Linux, you use a backup created in Windows to restore data to your Linux server. Because this backup file isn't created by TSM, it may not have the correct permissions for the restore process to access it. You need to make sure the backup file and the directory structure you copy it into have the proper permissions. Similarly, if you are copying files like certificates to additional nodes in a cluster, you need to make sure the files and the directories you copy them into have the permissions the *tableau* user needs in order to access them.

Setting permissions for individual files

If you are using a file you copy to one of the default locations created by TSM, you need to make sure the ownership and permissions on the file allow TSM access by giving the *tableau* user read access. You can do this in one of two ways:

- You can give the *tableau* user read access by giving the *tableau* group (in a default installation) read and execute access to a file using the `chgrp` and `chmod` commands. For example:

  ```
  chgrp tableau <backup>.tsbak

  chmod g+rx <backup>.tsbak
  ```

- Alternately, you can give world read and execute access to the file:

  ```
  chmod o+rx <backup>.tsbak
  ```

Setting permissions for directories

In addition to setting the proper permissions on the files themselves, TSM also needs permissions for the directory that contains the file, as well as any parent directories. If you are using a non-default location for files that TSM will access, you will need to make sure per-

missions for the parent directory or directories that contain the file allow read and execute access.

You can address this issue in a couple of ways:

- Change group ownership of the directory to the *tableau* group, and add group read and execute permission to the directory. Doing this makes files in the directory more available to the *tableau* user.

  ```
  chgrp tableau <directory-name>
  ```

  ```
  chmod g+rx <directory-name>
  ```

- Alternatively, you can add world read and execute permission to the directory. This makes files in the directory more available to all users on the system. This approach may require additional steps to ensure security of other files in the directory. For example, you may want to make sure other files in the directory are not world readable so other users cannot read them.

  ```
  chmod o+rx <directory-name>
  ```

Hint: You can use `namei -mo` command to list an entire permissions tree. This can make it easier to see what directories need to have permissions adjusted to allow access by the *tableau* group. You can find more information on the internet.

Configure Server Event Notification

A Tableau Services Manager (TSM) administrator can configure Tableau Server to allow notifications for the following events:

- Content updates
  - Extract failures (enabled by default)
  - Subscription views for users (disabled by default)

- Server health monitoring
  - Server status changes (disabled by default)
  - Desktop License reporting (disabled by default)
- Drive space
  - Email alerts when disk space crosses or remains below pre-configured thresholds (disabled by default)
  - Recording usage history (enabled by default)

Note: You need to configure SMTP before you can configure subscriptions or notifications. For more information, see Configure SMTP Setup.

Use the TSM web interface

1. Open TSM in a browser:

   `https://<tsm-computer-name>:8850`.

   For more information, see Sign in to Tableau Services Manager Web UI.

2. Click **Notifications**on the **Configuration** tab and click **Events**.

3. Configure notification settings for your organization:

   - Content updates

     - **Send emails for extract refresh failures**

       When this option is enabled (the default), a server administrator can configure email notifications to be sent when extract refreshes fail. These messages are configured at the site level, so even if this option is enabled, messages are not sent unless the **Send email to data source and workbook owners when scheduled refreshes fail** option is enabled for a site (this is enabled by default). For details, see Enable Extract Refresh Scheduling and Failure Notification.

     - **Allow users to receive email for views that they have subscribed to**

When this option is enabled (by default is it disabled), a server administrator can configure a site to send subscription email. These email messages are configured at the site level and can only be configured when this option is enabled. For details, see Set Up a Site for Subscriptions.

When users subscribe to a workbook or view, a snapshot of the view is emailed to them on a scheduled basis, so they can see the latest updates without having to sign into Tableau Server.

To allow users to attach PDF renderings on subscription emails, select **Let users add attachments to subscribed views**.

- Server health monitoring
  - **Send emails for Tableau Server process events (up, down, and failover)**

    Tableau Server sends an email message when the data engine, file store, gateway, or repository server processes stop or restart, or when the initial Tableau Server node stops or restarts.

    If you are running a single-server installation (all processes on the same computer), health alerts are only sent when Tableau Server is up. No "down" alerts are sent. If you are running a distributed installation that's configured for failover, a DOWN alert means that the active repository or a data engine instance has failed and the subsequent UP alert means that the passive instance (repository) or second instance (data engine) of that process has taken over.

    Note: Tableau Server is designed to be self-correcting. If a service or process stops responding or goes down, Tableau Server attempts to restart it. This can take 15 to 30 minutes to complete. Because of this, reacting immediately to service or process alerts can be

counter-productive, especially in an installation with redundant services that can handle requests while one restarts.

- **Enable Tableau Desktop License reporting**

  License reporting data originates in Tableau Desktop and is sent to Tableau Server. When this option is enabled, Tableau Server will generate and display the administrative report for Desktop License reporting. For information on the report, see Desktop License Usage.

- Drive space

  Enable notifications (alerts) for remaining disk space on your Tableau Server.

  - **Send emails when unused drive space drops below thresholds**

    You can configure Tableau Server to send email notifications when disk space usage on any node crosses a threshold, or remains below the threshold. And you can configure how often threshold notifications are sent.

    There are two thresholds you must set, **Warning threshold** and **Critical threshold**. Thresholds are expressed in percentage of disk space remaining. The critical threshold must be less than the warning threshold.

    You also specify the **Send threshold alert every** option. This determines how often, in minutes, warning and critical notifications should be sent. The default value is 60 minutes.

- **Record disk space usage information and threshold violations for use in custom administrative views**

  When you configure Tableau Server to record disk space usage, information about free disk space is saved in the repository and you can view the usage history using the Administrative Views.

4. Click **Save Pending Changes** after you've entered your configuration information.
5. Click **Pending Changes** at the top of the page:

6. Click **Apply Changes and Restart**.

Use the TSM CLI

The various notification values described above can be set individually with the tsm configuration set command. Alternatively, you can construct a json file and pass all configuration values in one operation. Both methods are described in this section.

Set notification values individually

The following table shows the key/value pairs that map to the notification events described earlier in this topic. Use the tsm configuration set command with the following syntax to set a single key/value pair:

```
tsm configuration set -k <config.key> -v <config_value>
```

For example, to enable job failure notifications, run the following command:

```
tsm configuration set -k backgrounder.notifications_enabled -v true
```

Notification option	Key	Value

Extract failures or Flow run failures	`backgrounder.notifications_enabled`	`true` | `false`
Enable subscription views for user	`subscriptions.enabled`	`true` | `false`
Enable PDF attachments for subscriptions	`subscriptions.attachments_enabled`	`true` | `false`
Maximum attachment size (MB) for subscription notifications	`subscriptions.max_attachment_size_megabytes`	integer value, default is `150`
Server status changes	`svcmonitor.notification.smtp.enabled`	`true` | `false`
License reporting	`features.DesktopReporting`	`true` | `false`
Remaining space thresholds: enable email notifications	`storage.monitoring.email_enabled`	`true` | `false`
Remaining space thresholds: warning percentage	`storage.monitoring.warning_percent`	integer value, for example, `20`
Remaining space	`storage.monitoring.critical_percent`	integer value, for example, `15`

thresholds: critical percentage		
Set email interval	`storage.monitoring.email_interval_min`	integer value, in minutes, for example, `25`
Record usage history	`storage.monitoring.record_history_enabled`	`true` | `false`

After you are done setting values, you must run the following command:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Set all notification values with a single json file

To make all notifications settings with a single configuration, you can pass a json file.

Copy and edit the following template to create a file for your configuration.

```
{
  "configKeys": {
    "backgrounder.notifications_enabled": true,
    "subscriptions.enabled": true,
    "subscriptions.attachments_enabled": true,
    "subscriptions.max_attachment_size_megabytes": 150,
    "svcmonitor.notification.smtp.enabled": true,
    "features.DesktopReporting": true,
    "storage.monitoring.email_enabled": true,
    "storage.monitoring.warning_percent": 20,
```

```
    "storage.monitoring.critical_percent": 15,
    "storage.monitoring.email_interval_min": 25,
    "storage.monitoring.record_history_enabled": true
  }
}
```

After you have saved the file, pass it with the following command:

```
tsm settings import -f <path-to-file.json>
```

To apply changes, run the following command:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Configure Data Cache

Views published to Tableau Server are interactive and sometimes have a live connection to a database. As users interact with the views in a web browser, the data that is queried gets stored in a cache. Subsequent visits will pull the data from this cache if it is available. By default, Tableau Server will cache and reuse data for as long as possible. You can change this behavior by configuring the caching option using the `tsm data-access caching set` command.

1. Run this command:

   ```
   tsm data-access caching set -r <value>
   ```

Where <value> is one of these options:

- **low or empty string ("")**. This is the default value and indicates that Tableau Server should configure cache and always use cached data when available.
- <value>. "<value>" specifies the maximum number of minutes data should be cached.
- **always** or **0** (zero). These values indicates that Tableau Server should always get the latest data and that the cache should be refreshed each time a page is reloaded.

2. Apply changes with the `tsm pending-changes apply` command. This will restart Tableau Server.

Database Drivers

Tableau connectors require a driver to talk to the database. Before you can connect to data sources from Tableau Server, you must install drivers for the data sources you want to connect to. You can find information about supported data sources for Tableau Server on Linux on the Tableau Server tech specs page. You can find driver links and installation instructions for all the supported connectors on the Driver Download page.

Important: You must install the PostgreSQL driver if you want to use the built-in administrative views. You can find this on the Driver Download page.

Install drivers in a cluster

You need to install the drivers for your data sources on the initial node in a Tableau Server cluster. If you install Tableau Server on multiple nodes, you must also install drivers on any node that runs any of the following processes:

- Application Server (Vizportal)
- Backgrounder
- Data Server
- VizQL Server

Server Crash Reporter

The Tableau Server administrator can enable an option to allow logs and related files to be sent to Tableau when the server has an issue that results in a crash. These files are used by Tableau to identify and address issues that cause crashes. By default this option is disabled, and it should only be enabled in organizations that are not subject to regulations related to data privacy.

Important: Do not enable crash reporting if your data is subject to privacy regulations.

If Tableau Server has a problem that results in a crash, log files and dump files are generated. If the crash data upload feature is enabled, these files are automatically gathered and zipped into an encrypted package that is sent in the background, at the scheduled time. The encrypted package is sent in small pieces to limit impact to network performance. Only one crash report is packaged and uploaded at a time (a new crash report is not packaged until the previous package has been uploaded) and is sent in a "first in, first out" order. You can schedule the sending for a low-use window to further reduce any impact to your users.

The encrypted package is made up crash dump files and logs that include the following:

- Crash/core dump files
- Error log files related to the crash
- Manifest files related to the crash

The files can contain data that includes:

- Machine-specific information (for example: hardware, operating system, domain).
- A snapshot of the contents of memory at the time of the crash, including application activity details like information about data connections, actions taken by the user in Tableau, and data being worked on in Tableau.
- Tableau information including customer-identifiable information.

Configure Server Crash Reporter

Server crash reporting is disabled by default. This topic describes how to enable and configure server crash reporting. Crash reports are encrypted and sent to Tableau. See Server Crash Reporter for more information.

If your organization uses a proxy server to connect to the internet then you must configure server crash reporter to use the proxy. Even if you have already configured Tableau Server to use a proxy, you must also configure server crash reporter separately. To configure proxy for server crash reporter you must use TSM CLI procedure as described in this topic.

Important: Do not enable crash reporting if your data is subject to privacy regulations.

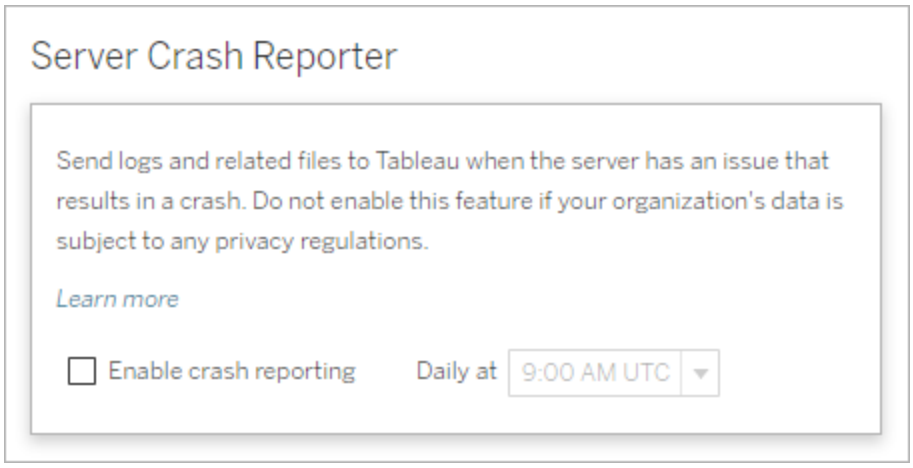
Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Maintenance** tab.

3. Under Other Maintenance Tasks, in Server Crash Reporter, select **Enable crash reporting**:



4. Specify the scheduled time of day to upload the crash reports to Tableau.
5. When you are finished, click **Pending Changes**, and then click **Apply Changes and Restart**.

Use the TSM CLI

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f path-to-file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Because the configuration file is using configKey class, the values that you pass are not validated by TSM as they are when you use configEntities class. You can verify and set individual options by using the tsm configuration commands.

Crash reporter settings

The crash reporter settings in the template below specify a range of options for configuring Tableau Server to send crash reports to Tableau.

Configuration template

Use this template to configure the gateway settings.

For more explanation about configuration files, entities, and keys see Configuration File Example.

```
{
      "configKeys": {
       "servercrashupload.enabled": "true",
       "servercrashupload.scheduled_time": "1:00:00 UTC",
       "servercrashupload.proxy_server_host": "",
       "servercrashupload.proxy_server_port": "",
       "servercrashupload.proxy_server_username": "",
       "servercrashupload.proxy_server_password": "",
       "servercrashupload.preserve_upload_packages": "false",
       "servercrashupload.delete_completed_dumps": "false"
       }
}
```

Configuration file reference

This table includes keys that you can set to configure crash reporting.

`servercrashupload.enabled`

Default: `false`.

Set to `true` to enable crash reporting.

`servercrashupload.scheduled_time`

Default: `1:00:00 UTC`

Specifies the scheduled time that crash uploads will begin. Enter time of day in 24 hour format.

`servercrashupload.proxy_server_host`

If your organization uses a proxy server to communicate with the internet, specify the host name.

`servercrashupload.proxy_server_port`

If your organization uses a proxy server to communicate with the internet, specify the port number.

`servercrashupload.proxy_server_username`

If your proxy server requires authentication, specify the user name with this key.

`servercrashupload.proxy_server_password`

If your proxy server requires authentication, specify the password with this key.

`servercrashupload.preserve_upload_packages`

Default: `false`.

To save all packages that are created for a crash reporting, set this key to `true`.

By default, packages are saved to `/var/opt/tableau/tableau_server-/data/tabsvc/clustercontroller/tabcrashreporter`.

`servercrashupload.delete_completed_dumps`

Default: `false`.

To delete all dumps after they are sent, set this key to `true`.

Navigate the Admin Areas of the Tableau Web Environment

As an administrator on Tableau Server or Tableau Cloud, you can access admin settings that aren't available to other users to configure sites, users, projects, and to do other content-related tasks.

The settings in this article refer to the Tableau web environment. Tableau Server administrators with appropriate credentials can also change server settings such as processor, caching, authentication, distributed deployment, and related configurations using the TSM web environment. For information, see Sign in to Tableau Services Manager Web UI.

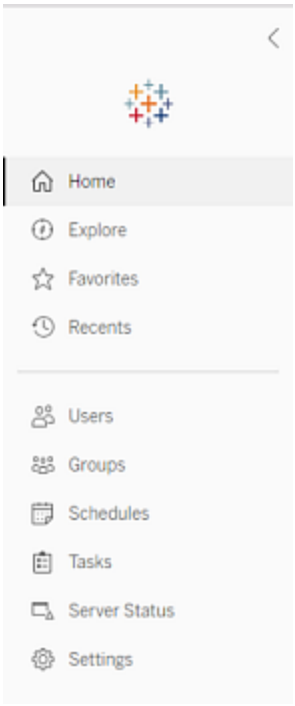
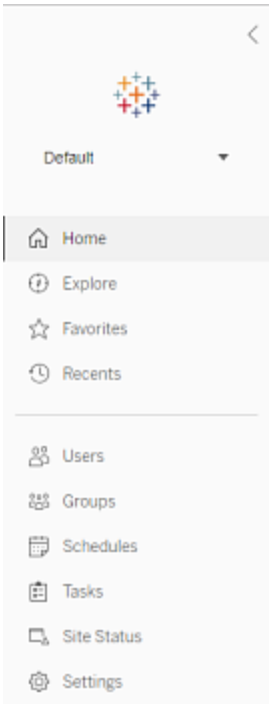
Access based on site role and number of sites

The menus you get when you sign in to Tableau Server or Tableau Cloud depend on the following conditions:

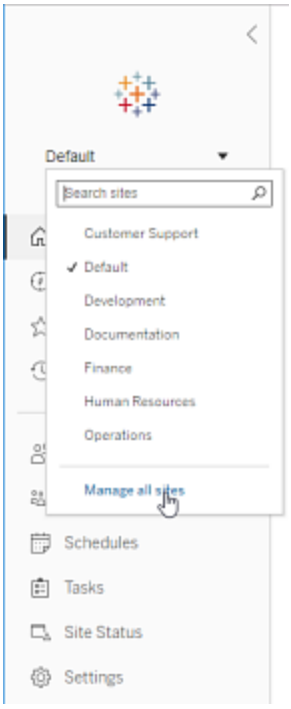
- Whether you're a site or server administrator.

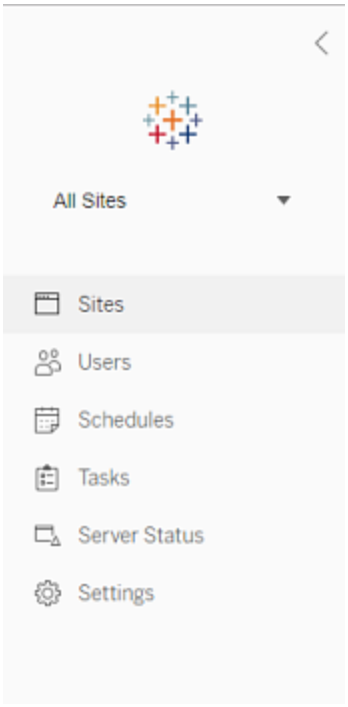
  Site administrator access is available on Tableau Cloud and Tableau Server. Server administrator access is only on Tableau Server.

- Whether you have access to only one site or to multiple sites.

On a **single-site** server, the site selector does not appear, and all other menus are the same.	
In a **multi-site** environment, menus along the left enable you to modify a specific site or all sites, and to configure users, groups, schedules, tasks, and server settings. To access server administrator settings that affect all sites, open the site menu by clicking the arrow next to the current site name, and then select **Manage all sites**. The **Content** and **Group** tabs go away, and the site menu text changes to **All Sites** to let you know you are managing server-wide settings, and options like **Server Status** reflect the server-wide view.	

To return to the site administration menus, select **All Sites**, and the select the site you want to manage.



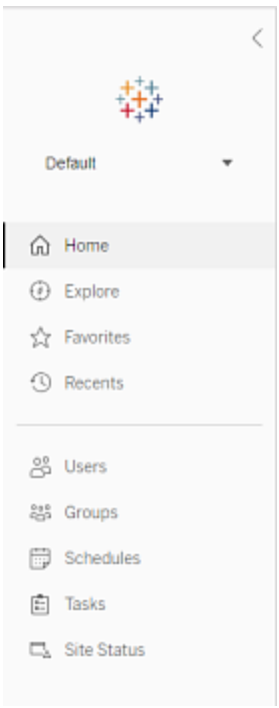


Site administrator

If you are a site administrator for Tableau Cloud or Tableau Server, and you have access to multiple sites, you'll get menus for selecting which site to manage, and for managing that site's content, users, groups, schedules, and tasks, and for monitoring its status.

The site selector displays the name of the current site. To go to another site, select the site menu, and then select the site name.

If you have access to only one site, the site selector does not appear, but all other menus are the same.



Server administrator tasks

Server administrators (available with Tableau Server Enterprise only) can do the following:

- Monitor server status and activity.
- Generate log files.
- Add sites and edit site settings. Only server administrators can add sites to the server.
- Add users to the server, and assign users to sites.
- Add and manage site groups.
- Add users to or remove users from identity pools.

To manage settings only for a specific site, you must first navigate to the site. Within each site, you can do the following:

- Administer content: Create projects, move content from one project to another, assign permissions, change ownership of a content resource, and so on.
- Manage schedules for extract refreshes and subscriptions.
- Monitor site activity and record workbook performance metrics.
- Manage storage space limits for content published by users.
- Allow web authoring.
- Enable revision history.
- Allow site administrators to add and remove users.
- Set the maximum number of licenses that site can consume for each license type (Creator, Explorer, Viewer).
- Allow users to subscribe to workbooks and views, and allow content owners to subscribe others to workbooks and views.
- Enable offline snapshots for favorites (iOS only).

Site administrator tasks

A site administrator on Tableau Cloud or Tableau Server can do the following tasks:

- Administer content: Create projects, move content from one project to another, assign permissions, change ownership of a content resource, and so on.
- View, manage, and manually run schedules for extract refreshes and subscriptions.
- Add and manage site users (if allowed by the server administrator; see Site Settings Reference).
- Add and manage site groups.
- Monitor site activity.

Move Tableau Server to Another Drive

If you need to move Tableau Server to a different drive (if a new policy requires you to not have application data on your system drive, for example, or you are running out of space on the original drive), you can do this by following the procedure below. The steps are intended as an example for moving Tableau Server from one drive to another drive on the same computer, and may not exactly reflect your installation and configuration. For instructions on moving Tableau Server to a new computer, see Migrate to New Hardware.

Before you start

Before starting, make sure you:

- Have a current backup of your data and assets from your existing installation as well as a settings export. You'll need these to restore your installation of Tableau Server on the new drive. For details on creating a backup, see Perform a Full Backup and Restore of Tableau Server.

  You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

- Have access to the Tableau Server computer with an account that is a member of the `tsmadmin` group. This is where you will be working and you need to be logged on to the computer with an account that is a member of the `tsmadmin` group.

- Have a copy of the installation program for the version of Tableau Server you are moving. You need this to install Tableau Server to the new drive.

- Know what authentication methods your current installation is using. For example, if Tableau Server is configured for SSL, SAML, or Kerberos, you will need to back up the related certificate or keytab files separately, and then copy them to the new drive after

you reinstall Tableau Server.

- Know and understand any initial node settings and configuration in your current installation.

When you are ready to move Tableau Server to another drive and have fully backed up your data and all assets and saved those files in a safe location on a different computer:

1. Open a terminal session on the initial node with an account that is a member of the `tsmadmin` group.
2. Run the `tableau-server-obliterate` script:

   ```
   sudo /opt/tableau/tableau_server/packages/scripts.<version_
   code>/tableau-server-obliterate -a -y -y -y
   ```

   If you have a multi-node (distributed) installation of Tableau Server, run the tableau-server-obliterate script on each node in the cluster.
3. Restart the computer you ran the `tableau-server-obliterate` script on.
4. Install and initialize Tableau Server in the new directory. For details, see Install and Initialize TSM.
5. Activate and Register Tableau Server.
6. (Optional). Configure Local Firewall.
7. (Optional). Verify LDAP.
8. Initialize Tableau Server. See Configure Initial Node Settings.
9. Copy your backup file *.tsbak* to the folder location specified by the `basefilepath.backuprestore` variable. By default this is: `/var/-opt/tableau/tableau_server/data/tabsvc/files/backups/`

> **Note:** You can change the location for the backup file. For more information, see tsm File Paths.

10. Stop Tableau Server:

    ```
    tsm stop
    ```

11. Restore your Tableau Server data backup and assets you saved earlier. This includes restoring your database, and any assets that need to be restored manually.

    For details, see Restoring core Tableau Server functionality.

12. Start Tableau Server:

    ```
    tsm start
    ```

Tableau Server product key activation

You can activate the same Tableau Server product key up to three times. This allows you to test Tableau Server (in a sandbox or QA environments, for example), as well as use Tableau in production. To maximize your activations, you should deactivate your product key when you remove Tableau Server from a computer, unless you will be reinstalling Tableau on the same computer. Doing this gives you the opportunity to use the activation on a different computer. For example, if you move Tableau Server from one computer or VM to another, deactivate the product key, then remove Tableau from the original computer. When you install Tableau on the new computer, you can activate the key there without any conflict. If you use role-based licensing, be sure to activate a Creator or Explorer key or you may lose administrator access to Tableau. If you are removing Tableau Server to reinstall it on the same computer, you don't need to deactivate the key. Tableau will use the key when reinstalled. For example, if you are moving Tableau from one drive on a computer to a different drive on the same computer. For information on how to deactivate a product key, see tsm licenses deactivate.

When you remove Tableau Server using the obliterate script, you have the option to have the script deactivate and delete the Tableau product key information by using a "`-l`" switch. This

will deactivate and remove all Tableau license information, including Tableau Desktop if you have that installed. If you know you want to deactivate the key, we recommend you deactivate the key *before* running the script. For more information, see Remove Tableau Server from Your Computer.

Distributed and High Availability Tableau Server Installations

This topic describes different installation deployment types for Tableau Server.

Validating your server deployment plan

Before you commit to installing a new Tableau Server deployment in your organization, be sure to carefully evaluate your options. For most organizations, Tableau Cloud will provide a more reliable, performant, and cost-effective analytics solution when compared to self-hosting Tableau Server. For information about the viability of Tableau Cloud for your organization, review this blog post, Should I move my analytics to the cloud?

Already running Tableau Server and want to migrate to Tableau Cloud? See Tableau Cloud Manual Migration Guide.

If you have determined that you must self-host Tableau Server, we recommend following the prescribed Tableau deployment in the Enterprise Deployment Guide (EDG). The EDG presents a fully-tested and supported, performant, scalable, secure reference architecture based on a tiered data network. Moving forward, we are committed to investing in the EDG reference architecture to ease feature rollouts and improve upgrade scenarios.

Installation types

The most basic way to run Tableau Server is to install a single node. With this type of installation you have a fully functional Tableau Server, with all Tableau Services Manager (TSM) and Tableau Server processes running on that single node, but this may not be the optimal way to use Tableau Server. You can decide how to install Tableau based on your

organization's needs, and your resources, adding additional nodes and configuring Tableau for high availability. Your installation options include:

- **Single-node installation**—This type of installation is reasonable for testing, running trials, and for environments that can handle occasional downtime and system availability due to lack of redundancy. All server processes are running on a single machine. There is less redundancy and fewer safeguards in the event of a problem with one of the server processes. You also need to make sure the computer you install Tableau Server on has adequate resources to handle the processes and the demands of users and data.

- **Distributed installation**—This type of installation is also called a multi-node installation and requires multiple computers so you can install and run server processes on those distributed nodes. Spreading the server processes out over multiple nodes can extend the reliability and efficiency of Tableau Server by providing redundancy and additional computing power. With the right configuration, a distributed installation can also provide you with automatic repository failover. For more information on failover, see Repository Failover .

- **Highly available (HA) installation**—An HA installation of Tableau Server is a special type of multi-node installation with a minimum of three nodes and multiple instances of key processes (the Repository, File Store/Data Engine (Hyper), Coordination Service, and Client File Service) on different computers. With an HA installation, there is built-in redundancy of those key processes, including multiple File Stores, and automatic Repository failover. The goal is to minimize system downtime by eliminating single points of failure, and enabling detection of failures with failover where possible.

  Downtime is still possible in the event of an initial node failure, or when a node running Application Server (VizPortal) is recovering from a failure. Dashboards and views may load more slowly than expected, and timeouts are possible, depending on how your system is configured and being used. For more information about initial node failure, see If an initial node fails below.

The first computer you install Tableau on, the "initial node," has some unique characteristics. Three processes run only on the initial node and cannot be moved to any other node except in a failure situation, the License Service (License Manager), Activation Service, and TSM Controller (Administration Controller). Tableau Server includes a script that automates moving these processes to one of your other existing nodes so you can get complete access back to TSM and keep Tableau Server running.

Two other processes are initially included on the initial node but can be added or moved to additional nodes, the CFS (Client File Service) and the Coordination Service. Depending on how your installation was configured with CFS and Coordination Service, you may also need to take steps to redeploy these.

For information about moving the License service and TSM Controller from the initial node to another node, see Recover from an initial node failure below.

External repository

For optimal performance for Tableau Server we recommend isolating the repository on a dedicated node in your deployment. If you have an Advanced Management license, consider running the repository as an external database.

If your organization has a peak load of more than 1000 VizQL sessions per hour, we also recommend running Tableau Server on Linux. In this scenario, VizQL sessions refer to any user actions that display or generate visualizations from Tableau Server.

For more information, see Tableau Server External Repository.

Prerequisite

These instructions assume that your cluster meets the Distributed Requirements.

All nodes in a multi-node cluster must have the same type of operating system and the same major version of that operating system. For example, all RHEL 9 nodes.

You cannot install a multi-node instance of Tableau Server on a combination of Linux and Windows nodes.

Licensing

You must have a valid Tableau Server product key. The type of license you have may determine how many nodes you can install Tableau on. For more information on licensing, see Licensing Overview.

Creating a distributed Tableau Server installation

These are the general steps you follow to create a distributed installation of Tableau Server:

1. Begin by installing Tableau Server on your initial node.

   For details, see Install and Configure Tableau Server.

2. Generate a node configuration (bootstrap) file on the initial node.

   For details, see Generate the node bootstrap file.

3. Install Tableau Server on an additional node using the node bootstrap file.

   For more information, see Install and initialize an additional node.

4. Configure your additional node with the processes you want to run on it.

   For more information, see Configure the additional node.

5. Repeat Steps 3 and 4 for any additional nodes you want to install.

6. Deploy a new Coordination Service ensemble.

   For more information, see Deploy a Coordination Service Ensemble .

7. Add Client File Service to every node that is running the Coordination Service.

   For more information, see Configure Client File Service .

Creating a highly available (HA) Tableau Server installation

A high availability Tableau Server installation is a special type of distributed installation, designed to accommodate failure in key server components without loss of complete server functionality. To create an HA installation, follow the same steps you take to create a distributed deployment but include additional steps to make the deployment highly available. These additional steps include adding at least two additional nodes (for a minimum of three nodes in the cluster), adding a second instance of the repository, and second instances of the data engine/file store, adding additional gateway processes, and deploying a Coordination Service ensemble. You can also add a load balancer to distribute requests among the gateways.

At a high level, these are the steps you follow to create a highly available installation of Tableau Server:

1. Begin by installing Tableau Server on your initial node.

   For details, see Install and Configure Tableau Server.

2. Generate a node configuration (bootstrap) file on the initial node.

   For details, see Generate the node bootstrap file.

3. Install Tableau Server on at least two additional nodes using the node bootstrap file.

   For more information, see Install and initialize an additional node.

4. Configure each additional node with the processes you want to run on it. These must include a second copy of the Tableau Server repository, and a second copy of the data engine and file store, as well as additional instances of the gateway.

   For more information, see Configure the additional node.

5. Deploy a Coordination Service ensemble.

   For more information, see Deploy a Coordination Service Ensemble .

6. Add Client File Service to every node that is running the Coordination Service.

   For more information, see Configure Client File Service .

7. (Optional) Configure a load balancer.

   For more information, see Add a Load Balancer.

For details on how to create a three-node HA installation, see Example: Install and Configure a Three-Node HA Cluster.

If an initial node fails

If there is a problem with the initial node and you have redundant processes on your other nodes, there is no guarantee that Tableau Server will continue to run.

- Tableau Server can continue to run for up to 72 hours after an initial node failure, before the lack of the licensing service impacts other processes. If so, your users *may* be able to continue to sign in and see and use their content after the initial node fails, but you will not be able to reconfigure Tableau Server because you won't have access to the Administration Controller.
- If you are running a version of Tableau Server 2021.4.2 (or older) that is configured for ATR, then problems with the initial node will render all server functionality unavailable. This is true whether the node has a problem or if you intentionally stop it (for instance, to do a system-level patch).

Even when configured with redundant processes, *it is possible that Tableau Server may not continue to function after the initial node fails*. This is true even when an installation is configured for high availability. This means you should make a point of moving the two unique processes to another of your running nodes as soon as possible. If your initial node fails for reasons that are recoverable in a relatively short amount of time (for example, a hardware failure you can correct), you should first attempt to bring the node back up without using the procedure below.

Recover from an initial node failure

With a Tableau Server installation, the initial node includes two services that are only installed on that node, the License service, and the TSM Controller. If there is a problem with the initial node, Tableau Server may not continue to function, even when configured for high availability. To recover from a situation where the initial node fails, you can move the TSM Controller and the License service to one of your already configured nodes. This allows you to recover from the failure while using resources you already have in the cluster. You do not have to configure a standby initial node in case the initial node fails.

For details on how to recover from a failure on the initial node, see Recover from an Initial Node Failure.

Configure Coordination Service ensemble on additional nodes

Configuring a Coordination Service on multiple nodes provides additional duplication of processes and so reduces the possibility of server downtime due to an issue with one of the Coordination Service nodes. For details on how to deploy a Coordination Service ensemble on your cluster, see Deploy a Coordination Service Ensemble .

Add Client File Service (CFS) to additional nodes

Tableau Server requires at least one instance of Client File Service (CFS). Adding additional instances of CFS to other nodes provides additional duplication of processes and so reduces the possibility of server downtime due to an issue with one of the CFS nodes. We recommend that you configure an instance of CFS on each of the nodes where you deploy the Coordination Service. For details on how to configure CFS on other nodes, see Configure Client File Service .

Tableau Server service license check

A number of processes are installed when you install Tableau Server. Some of these processes are dependent on the existence of a valid Tableau Server license while other installed processes are not. The subset of Tableau Server that require a valid Tableau Server license are considered "licensed processes."

When a licensed process starts or restarts, the process checks with the Tableau Server License Manager service on the initial node to verify there is a valid license. When the License Manager validates the license, the process is fully functional and able to respond to requests from other Tableau Server processes. Once a licensed process has received confirmation from the License Manager, the process does not need to reconfirm the license for 72 hours, or until the process restarts. If the process is not able to verify that it is licensed (if the primary node is unavailable, for example) it cannot run, but it continues to check for a valid license until it confirms the license. To see when the last licensing check occurred, look at the log files in the `/var/opt/tableau/tableau_server/data/tabsvc/logs/licenseservice` directory. For more information about licensed processes, see Licensed processes.

Distributed Requirements

Before you start to configure a Tableau Server cluster, make sure you meet the following requirements.

Hardware

While the computers you use in your cluster must meet the requirements described in Before you install..., they do not need to be identical.

Hardware Guidelines for High Availability

Here are some guidelines for the systems you use for failover and high availability:

- **Failover—three computers**: To configure a cluster that provides failover support for the file store and repository processes, you need at least three computers or VMs: one for the initial Tableau Server node and two for additional nodes.
- **Multiple gateways—three computers and a load balancer**: Adding multiple Gateway processes to your Tableau Server installation and using a load balancer to automatically distribute requests to those gateways enhances the reliability of Tableau further. To configure a cluster that provides failover support and multiple gateways, you need to add a load balancer to front your three-node cluster.
- **Failover & multiple gateway support—three computers and a load balancer**: To configure a cluster that provides the above plus support for multiple gateways, you need at least three computers or VMs, and a load balancer to front the cluster.
- **High availability—three computers and a load balancer**: To configure for high availability, you need the resources described above.
- **Initial computers**: If you configure for high availability, the initial Tableau Server node may be running few or no Tableau Server processes. Therefore, the computer that serves as the initial node does not need as many cores as the ones running your additional nodes. You will, however, need adequate disk space for backups because the initial computer is used during the backup and restore processes. In addition to the amount of space needed for the backup file, you need temporary disk space. For details on disk space requirements, see Disk Space Usage for Backup and Restore.

Software

- All nodes in a multi-node installation must be running the same version of Tableau Server.
- All nodes in a multi-node cluster must have the same type of operating system and the same major version of that operating system. For example, all RHEL 9 nodes.

You cannot install a multi-node instance of Tableau Server on a combination of Linux and Windows nodes.

Installation location

Keep in mind the following requirements and limitations:

- The installation and data directory locations for Tableau Server must be the same on all nodes in a cluster.
- You can specify a non-default install location on RHEL-like distributions, but cannot change the location on Ubuntu. For more information, see Installation directory.
- When you initialize Tableau you can specify a non-default location for the data directory. For more information, see Data directory.

Networking and Ports

- **Ports**: As with any distributed system, the computers or VMs you use need to be able to communicate with one another. For information on how Tableau Services Manager handles port mapping, see Tableau Services Manager Ports.

- **Latency**: Network latency between server nodes can impact Tableau Server performance. Be aware of possible latency issues, especially if you run into performance problems. To reduce network latency, you can take steps such as locating your gateways and data sources in proximity to Tableau Server.

- **Static IP addresses**: Any computer running Tableau Server, whether it's a single server installation or part of a cluster, must have a static IP address.

- **Discoverable**: Each node in the cluster must be discoverable from other node computers using DNS or a local host file.

- **Time zone and time**: Each node in the cluster must be in the same timezone, with their system clocks synchronized. This may happen automatically. For example, if your nodes are all in the Active Directory domain, the domain controller usually handles this. If you are not sure your cluster meets this requirement, consult with your internal IT experts.

Best Practices

Here are some things to keep in mind before you start to install and configure:

- **IP addresses or computer names**: As mentioned above, each computer in the cluster must use a static IP address.
- **Backup**: It's a best practice to create a backup prior to making significant system changes. See Back up Tableau Server Data for steps.

SSL

If you are planning to configure SSL for a highly available Tableau Server cluster with multiple gateways and a load balancer (learn more), make sure that the SSL certificate you use was issued for the load balancer's host name. See Configure SSL for External HTTP Traffic to and from Tableau Server for other details.

Distributed Installation Recommendations

When you add nodes to a Tableau Server installation, you must decide how many processes to run on each computer. This page provides some general recommendations that are intended only as a starting point.

In addition to these general recommendations, you should also:

- Understand how your organization uses Tableau Server and tune your configuration for your use case—for example, whether you want to optimize for user response or for extract refreshes.
- Perform thorough performance testing to identify the best places to adjust process configuration.

For more information on tailoring a Tableau Server installation to your organization's needs, see Performance Tuning.

For more information on the requirements for a distributed installation and for information on configuring additional nodes, see Distributed and High Availability Tableau Server Installations.

Recommendations for all installations

Although the computers that make up a Tableau Server cluster do not need to have identical hardware, they must all meet the same minimum system requirements. The recommendations on this page assume that the computers where you install Tableau Server have eight cores or more.

The following recommendations apply to all server configurations:

- Run Backgrounder processes on a dedicated computer if you plan on refreshing extracts frequently. Backgrounder processes are generally the most CPU intensive and can slow down other processes on the same computer.
- If you plan to refresh extracts frequently or if you plan to refresh large extracts, increase the number of processes for Backgrounder processes.
- Run VizQL processes on a different computer than Backgrounder processes. Having them on the same machine means that extract refreshes can affect user views.
- The instance of Data Engine installed on the node where File Store is installed is used for querying data for view requests. Consider separating the File Store process from the backgrounder processes to help minimize the backgrounder tasks from affecting user views.
- **Optimizing with topology configurations:**
    - Co-locating File Store on the same node as the Administration Controller can reduce the length of time it takes to back up Tableau Server by reducing or eliminating the need to transfer data between nodes during the backup process. This is especially true if your organization uses many extracts.
    - Co-locating the repository (pgsql) with the Administration Controller node can also help to reduce back up time, but the time savings is less significant than that of the File Store.

The Administration Controller is usually on the initial node, unless you have had an initial node failure and moved the controller to another node.

Note: In a distributed installation with three or more nodes, you can have a maximum of two repository instances (active and passive). You can also run Tableau Server with one repository, but doing this means there is no failover available for the repository. For more information, see Tableau Server Repository.

Install and Configure Additional Nodes

After you install Tableau Server on one computer (or node), the server is functional and ready for use, but it has no redundancy. If there is a problem with a process or a problem with the computer itself, Tableau Server may be unavailable. In addition, all processes are running on a single computer, so there can be contention for resources on that computer.

You can extend your Tableau Server installation by adding Tableau to additional nodes, creating a distributed installation. This article describes the general steps for installing Tableau Server on additional nodes and assumes you have already installed Tableau on an initial node. For more information on installing Tableau on the initial node, see Install and Initialize TSM.

If you are installing Tableau Server on multiple nodes, you should install and configure one node at a time. This makes it easier to troubleshoot any issues you might run into.

Important: You should add and configure additional nodes when you can fully complete the process by applying pending changes. Adding a node without finishing by applying pending changes can result in users being unable to log into Tableau Server.

Installation location

Keep in mind the following requirements and limitations:

- The installation and data directory locations for Tableau Server must be the same on all nodes in a cluster.
- You can specify a non-default install location on RHEL-like distributions, but cannot change the location on Ubuntu. For more information, see Installation directory.
- When you initialize Tableau you can specify a non-default location for the data directory. For more information, see Data directory.

Use the TSM web interface

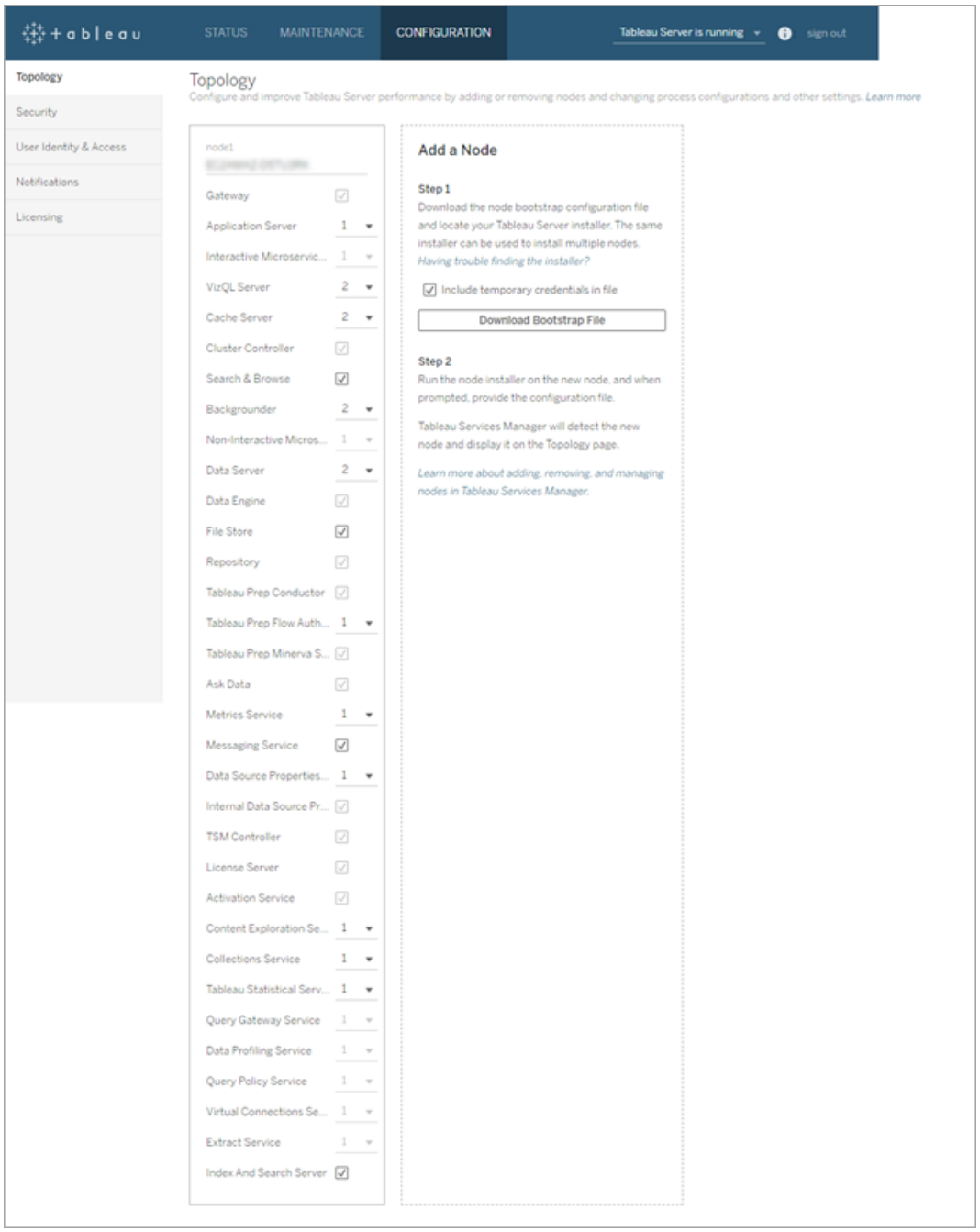
Generate the node bootstrap file

1. Open TSM in a browser:

   `https://<tsm-computer-name>:8850`

   For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Configuration** tab, and in the Add a Node box, click **Download Bootstrap File**.



The bootstrap file is created and copied to your local computer.

Embedded credentials are included in the bootstrap file by default. If you don't want credentials embedded in the bootstrap file, clear the **Include temporary credentials in**

file option. If you want to completely disable the ability to include embedded credentials in node bootstrap files, you can set a configuration option for the server. See features.PasswordlessBootstrapInit for more details.

Install and initialize an additional node

Before you begin, verify that your node bootstrap file is recent. For example, if you have run `tsm security regenerate-internal-tokens` after you generated a bootstrap file, then initialization will fail.

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the new computer.

2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see Configure Local Firewall.

3. On the new node, run the Tableau Server Setup program:

   Use the package manager to install the Tableau Server package.

   You must install the new version to the same location as the existing version. The install location must be the same on all nodes. Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

   - On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

     - **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

       ```
       sudo yum update

       sudo yum install tableau-server-<version>.x86_64.rpm
       ```

- **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

  Run the following command:

  ```
  sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
  ```

  Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

  ```
  sudo apt-get update
  sudo apt-get upgrade
  sudo apt-get -y install gdebi-core
  sudo gdebi -n tableau-server-<version>_amd64.deb
  ```

4. To initialize communications between your new node and the initial node, run the initialize-tsm script that is installed when you install Tableau Server.

   Navigate to the `scripts` directory:

   ```
   cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
   ```

5. Run the initialize-tsm script:

   ```
   sudo ./initialize-tsm -b /path/to/<bootstrap>.json --accepteula
   ```

- Use the `-b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in Securing secrets for import and export operations.
- If the bootstrap file was generated without embedded credentials, use the `-u` flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see tsm topology nodes get-bootstrap-file.

  ```
  sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u
  <admin-user-on-first-node> --accepteula
  ```

  Important: You must enter the credentials of the same user that you used during the installation process on the initial node.

- The `--accepteula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location: End User License Agreement.

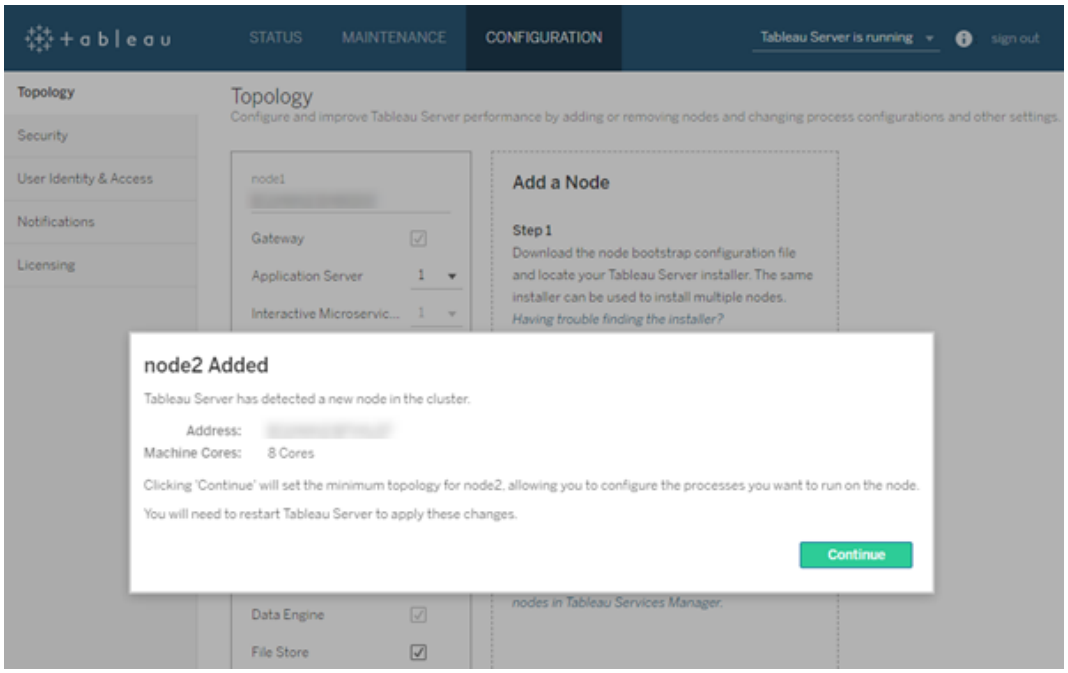
Configure the additional node with Cluster Controller

1. Open TSM in a browser:

   ```
   https://<tsm-computer-name>:8850
   ```

   For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Configuration** tab. A message should tell you that the new node was added.



Click **Continue** to dismiss the message.

3. Cluster Controller is part of the minimum topology and is already selected. Each node must run Cluster Controller.

   If you only want to set the minimum topology for the node, go to Step 4 below. You might do this if you will be adding additional nodes and are not ready to configure this node beyond the minimum.

   If you want to add additional processes to the minimum topology, specify the processes that should run on the node. Do this by selecting the processes you want, or specifying the number of processes that should run on the node.

   For example, to add a Gateway and two instances of the Backgrounder on the node:

   a. Select **Gateway**.

   b. Set the **Backgrounder** count to 2.

Adding Backgrounder to a node will also add an instance of Data Engine if one is not already on the node.

The specific processes and process counts you set will depend on your organizational environment and needs. Some processes are added automatically when you add another process. For more information, see Configure Nodes and Tableau Server Processes.

Note: The TSM Web UI limits you to a maximum of 8 instances of processes that allow you to select the number of instances. To configure more instances than this, use the command line and the `TSM topology set-process` command. For more information, see tsm topology set-process.

4. Click **Pending Changes** at the top of the page:

If you are configuring a cluster with three or more nodes, a Coordination Service ensemble warning displays. You can continue. You will deploy a Coordination Service ensemble in a separate step.

5. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

After Tableau Server restarts, the node is included with the minimum topology necessary and any additional processes you configured.

Use the TSM CLI

Generate the node bootstrap file

To install Tableau Server on additional computers you use the same installer you did for the initial node, along with a "bootstrap" file you generate on the initial node.

Important: The bootstrap file contains a copy of the master keystore file used for encrypting the configuration secrets. The file can also embedded credentials which are valid for a pre-determined amount of time (see tabadmincontroller.auth.expiration.minutes) and serve as a session cookie. We strongly recommend that you take additional measures to secure the bootstrap file.

The following command set provides an example method to encrypt the bootstrap file output. This method is similar to the encryption process described in more detail at Securing secrets for import and export operations.

Note however, the method here must be passed as separate arguments with trailing `&& \` operators as follows:

```
mkfifo -m 600 /tmp/secure1 && \

tsm topology nodes get-bootstrap-file --file /tmp/secure1 && \

gpg --symmetric --batch --yes --passphrase-file ~/.secret-
s/pgppassphrase.txt --cipher-algo AES256 --output encrypted.enc <
/tmp/secure1 && \

rm /tmp/secure1
```

1. After installing Tableau Server on the initial node, generate the node bootstrap file.
2. On the initial node, open a terminal session.
3. Type this command to generate the bootstrap file:

   ```
   tsm topology nodes get-bootstrap-file --file <path\file>.json
   ```

   Embedded credentials are included in the bootstrap file by default. If you don't want the bootstrap file to include credentials, use the `-nec` or `--no-embedded-credentials` option:

   ```
   tsm topology nodes get-bootstrap-file --file <path\file>.json -
   -no-embedded-credentials.
   ```

If you want to completely disable the ability to include embedded credentials in node bootstrap files, you can set a configuration option for the server. See features.PasswordlessBootstrapInit for more details.

Install and initialize an additional node

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the new computer.

2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see Configure Local Firewall.

3. On the new node, run the Tableau Server Setup program:

   Use the package manager to install the Tableau Server package.

   You must install the new version to the same location as the existing version. The install location must be the same on all nodes. Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

   - On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

     - **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

       ```
       sudo yum update

       sudo yum install tableau-server-<version>.x86_64.rpm
       ```

     - **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note

below.

Run the following command:

```
sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
```

> **Note:** When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. To initialize communications between your new node and the initial node, run the initialize-tsm script that is installed when you install Tableau Server.

   On the new node:

   Navigate to the `scripts` directory:

   ```
   cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
   ```

5. Run the initialize-tsm script:

   ```
   sudo ./initialize-tsm -b /path/to/<bootstrap>.json --accepteula
   ```

- Use the `-b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in Securing secrets for import and export operations.
- If the bootstrap file was generated without embedded credentials, use the `-u` flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see tsm topology nodes get-bootstrap-file.

  ```
  sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u
  <admin-user-on-first-node> --accepteula
  ```

  Important: You must enter the credentials of the same user that you used during the installation process on the initial node.

- The `--accepteula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

  ```
  /opt/tableau/tableau_server/packages/docs.<version_
  code>/EULA.rtf
  ```

Configure the additional node

Note: This basic example illustrates how to set the topology on a node. For a more detailed, working multi-node example, see Example: Install and Configure a Three-Node HA Cluster.

On the initial (original) node, set the topology for the newly added node. The topology specifies which processes should run on the node, and how many instances of each process should run. The topology for the node will depend on your environment and organizational needs. The below are just examples of setting the topology.

1. On the initial (original) node, open a terminal session.

2. Get the node-id for the new node:

   ```
   tsm topology list-nodes -v
   ```

   The `-v` option lists the nodes and the processes they are currently running. You can identify the newly added node because it will not have any processes on it.

3. Specify the individual processes that should run on this node.

   Do this with the following command:

   ```
   tsm topology set-process -n <nodeID> -pr <processname> -c <n>
   ```

   You must add an instance of the Cluster Controller to each node.

   For example, to add the Cluster Controller, two instances of the Backgrounder, and a Gateway to node2:

   ```
   tsm topology set-process -n node2 -pr clustercontroller -c 1
   tsm topology set-process -n node2 -pr backgrounder -c 2
   tsm topology set-process -n node2 -pr gateway -c 1
   ```

   The specific processes and process counts you set will depend on your organizational environment and needs. Some processes are added automatically when you add another process. For more information, see Configure Nodes and Tableau Server Processes.

4. Apply the node configuration. If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

```
tsm pending-changes apply
```

A warning about deploying a Coordination Service ensemble displays if you have deployed a three- or five-node cluster. If this is the only warning, you can safely override it using the `--ignore-warnings` option to apply the configuration changes in spite of the warning.

```
tsm pending-changes apply --ignore-warnings
```

Install drivers

You need to install drivers so that Tableau Server can connect to data and run extracts. Install these drivers on nodes that are running any of the following processes:

- VizQL Server (vizqlserver)
- Application Server (vizportal)
- Data Server (dataserver)
- Backgrounder (backgrounder)

Drivers and administrative views

If you want to use the built-in administrative views in Tableau Server, you also need to install the PostgreSQL driver on any nodes running any of the above processes.

For more information, see Database Drivers.

Database Drivers

Tableau connectors require a driver to talk to the database. Before you can connect to data sources from Tableau Server, you must install drivers for the data sources you want to connect to. You can find information about supported data sources for Tableau Server on Linux on the Tableau Server tech specs page. You can find driver links and installation instructions for all the supported connectors on the Driver Download page.

Important: You must install the PostgreSQL driver if you want to use the built-in administrative views. You can find this on the Driver Download page.

Install drivers in a cluster

You need to install the drivers for your data sources on the initial node in a Tableau Server cluster. If you install Tableau Server on multiple nodes, you must also install drivers on any node that runs any of the following processes:

- Application Server (Vizportal)
- Backgrounder
- Data Server
- VizQL Server

Example: Install and Configure a Three-Node HA Cluster

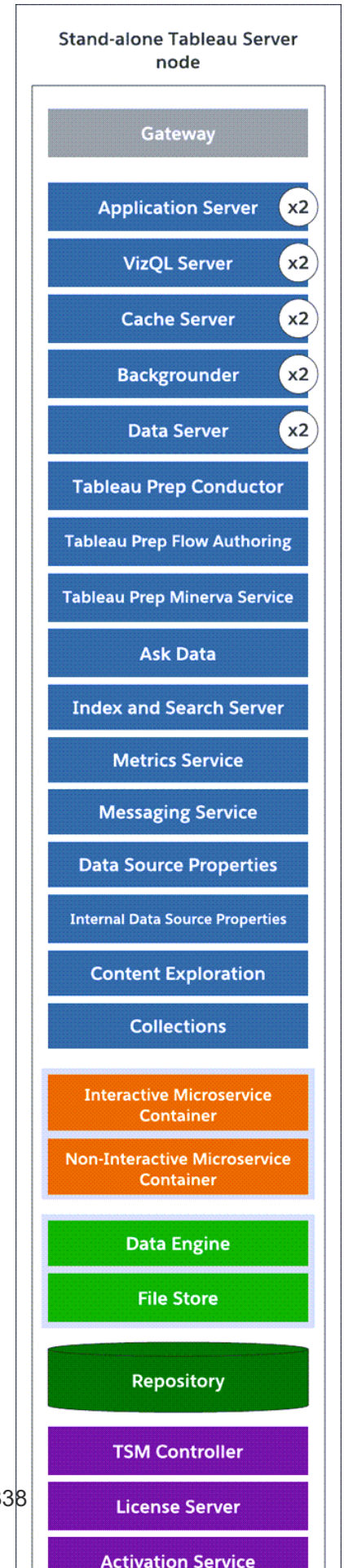
A multi-node high availability installation of Tableau Server can help to maximize the efficiency and availability of Tableau Server. When configuring a multi-node Tableau Server deployment, the steps you perform are designed to build in redundancy, helping to reduce your potential downtime. In addition to simply improving efficiency by moving or adding server processes to additional nodes, you can create a highly available (HA) installation of Tableau Server by satisfying these requirements:

- Add additional nodes for a total of at least three nodes
- Deploy a Coordination Service ensemble on at least three nodes
- Add an instance of Client File Service (CFS) on at least one additional node (we recommend adding an instance on each node running the Coordination Service)
- Add a second instance of the File Store on one of the additional nodes (Data Engine will be installed automatically, if it is not already on the node)
- Add a second instance of the Repository (pgsql) on one of the additional nodes

A Tableau Server installation that includes these additions will have built-in redundancy and can support failover in the event of a problem with the repository. This example shows how to do this, and more.

A Single Server System

After installing Tableau Server on an initial node, you have a system that is running everything it needs to function. It has at least one instance of all server processes and is the most basic configuration of Tableau Server. It has no redundancy. The server topology looks like this (some TSM-specific processes are not shown):


Stand-alone Tableau Server node
Gateway
Application Server x2
VizQL Server x2
Cache Server x2
Backgrounder x2
Data Server x2
Tableau Prep Conductor
Tableau Prep Flow Authoring
Tableau Prep Minerva Service
Ask Data
Index and Search Server
Metrics Service
Messaging Service
Data Source Properties
Internal Data Source Properties
Content Exploration
Collections
Interactive Microservice Container
Non-Interactive Microservice Container
Data Engine
File Store
Repository
TSM Controller
License Server
Activation Service

A Three-Node System

To build in redundancy, you need to add additional nodes to host instances of the repository and the data engine and file store processes. You can add instances of other processes, including multiple instances of a process on a node. In addition, to reduce the system's vulnerability, you can run multiple gateways and additional instances of some of the server processes. The fewest number of computers required to achieve this configuration is three.

In the diagram below, the file store process has been added to both additional nodes. A second, passive instance of the repository has also been added to one of the other new nodes. Finally, the server processes (shown in blue) have been added to the additional nodes to provide redundancy.

Configuration steps

This procedure describes how to configure a three-node HA Tableau Server cluster with two repository instances and two file store/data engine instances on the additional nodes as pictured above.

Before you begin

Before you install Tableau Server on any additional nodes, ensure that each additional node meets the distributed requirements. See Distributed Requirements for details.

Use the TSM web interface

Note: This operation includes steps that you may need to perform using the TSM command line.

Step 1: Install the initial node

See Install and Configure Tableau Server.

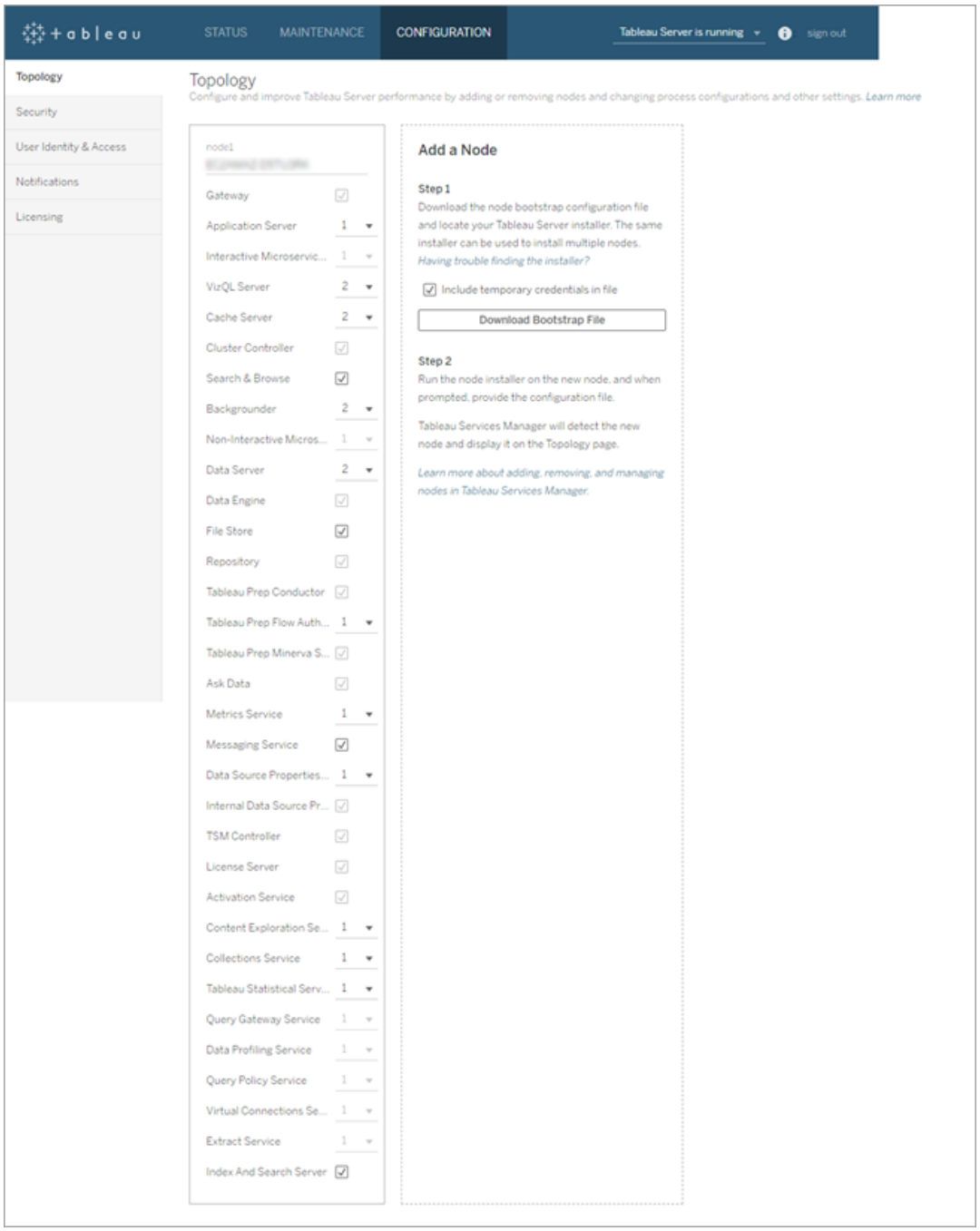
Step 2: Generate a bootstrap file for the additional nodes

1. Open TSM in a browser:

   `https://<tsm-computer-name>:8850`

   For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Configuration** tab, and in the Add a Node box, click **Download Bootstrap File**.



The bootstrap file is created and copied to your local computer.

Embedded credentials are included in the bootstrap file by default. If you don't want credentials embedded in the bootstrap file, clear the **Include temporary credentials in**

file option. If you want to completely disable the ability to include embedded credentials in node bootstrap files, you can set a configuration option for the server. See features.PasswordlessBootstrapInit for more details.

Step 3: Install and initialize node 2

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the new computer.

2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see Configure Local Firewall.

3. On the new node, run the Tableau Server Setup program:

   Use the package manager to install the Tableau Server package.

   You must install the new version to the same location as the existing version. The install location must be the same on all nodes. Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

   - On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

     - **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

       ```
       sudo yum update

       sudo yum install tableau-server-<version>.x86_64.rpm
       ```

     - **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the

note below.

Run the following command:

```
sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
```

> **Note:** When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. To initialize communications between your new node and the initial node, run the initialize-tsm script that is installed when you install Tableau Server.

   Navigate to the `scripts` directory:

   ```
   cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
   ```

5. Run the initialize-tsm script:

   ```
   sudo ./initialize-tsm -b /path/to/<bootstrap>.json --accepteula
   ```

- Use the `-b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in Securing secrets for import and export operations.
- If the bootstrap file was generated without embedded credentials, use the `-u` flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see tsm topology nodes get-bootstrap-file.

  ```
  sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u
  <admin-user-on-first-node> --accepteula
  ```

  Important: You must enter the credentials of the same user that you used during the installation process on the initial node.

- The `--accepteula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location: End User License Agreement.

Step 4: Install and initialize node 3

Repeat Step 3 above.

Step 5: Configure the new node with a minimum topology

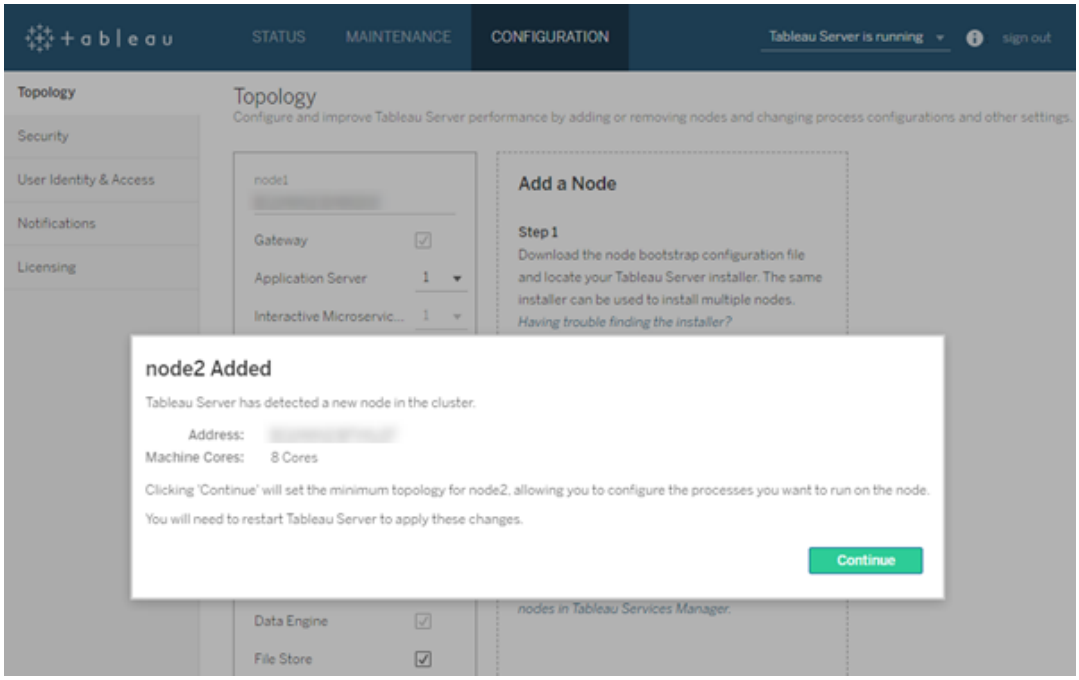
To complete the process of adding new nodes to your cluster, you need to configure them with a minimum topology. With a minimum topology, the only pending change will be the addition of Cluster Controller, which is required on each node. If you want other processes on the nodes you can add most of them at the same time. This example only configures the nodes with Cluster Controller.

1. Open TSM in a browser:

   ```
   https://<tsm-computer-name>:8850
   ```

For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Configuration** tab. A message should tell you that new nodes were added.



Click **Continue** to dismiss the message.

3. Click **Pending Changes** at the top of the page:

If you are configuring a cluster with three or more nodes, a Coordination Service ensemble warning displays. You can continue. You will deploy a Coordination Service ensemble in a separate step.

4. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

When Tableau Server restarts, the nodes are included with the minimum topology necessary.

Step 6: Deploy a Coordination Service ensemble

If you install a total of three or more nodes, you should also deploy a Coordination Service ensemble. If you do not, you will get a warning message every time you make changes to the server configuration or topology. You can ignore this message, but as a best practice you should deploy a multi-node Coordination Service ensemble.

When you install Tableau Server, a single instance of the Coordination Service is installed on the initial node. TSM and Tableau Server depend on the Coordination Service to function properly, so to provide redundancy and ensure availability on multi-node installations, configure additional instances of the Coordination Service by deploying a Coordination Service ensemble. Coordination Service ensembles are installed with one, three, or five instances of the Coordination Service. In a three-node installation of Tableau Server, the recommended number of Coordination Service instances is three, one on each node.

Do not attempt to deploy a Coordination Service ensemble if there are other changes pending. Discard or apply any pending changes before deploying a new Coordination Service ensemble.

Note: This operation includes steps that you may need to perform using the TSM command line.

The following steps illustrate how to deploy a new Coordination Service ensemble on an existing three-node Tableau Server cluster and clean up the old ensemble.

1. On the initial node, open a terminal session.
2. Stop Tableau Server:

   ```
   tsm stop
   ```

   If prompted, sign in as a TSM administrator.

Some TSM processes will continue to run, including the Administration Controller and Administration Agent.

3. Confirm there are no pending changes:

   ```
   tsm pending-changes list
   ```

   If there are pending changes, you need to either discard the changes or apply them. Applying pending changes will take some time:

   - Discard the changes

     ```
     tsm pending-changes discard
     ```

     or

   - Apply the changes:

     ```
     tsm pending-changes apply
     ```

   Wait until the command completes and you are returned to the system prompt.

4. Get the node IDs for each node in the cluster:

   ```
   tsm topology list-nodes -v
   ```

5. Use the `tsm topology deploy-coordination-service` command to add a new Coordination Service ensemble by adding the Coordination Service to specified nodes. You must specify the node(s) that the Coordination Service should be added to, using the actual node ID to identify each node. The command also makes the new ensemble the "production" ensemble (the ensemble in use) and removes the old ensemble, unless the deployment fails. If this happens, see step 6 below.

   Note: A "y/n" prompt displays confirming that a server restart will take place. To run the command without input, include the `--ignore-prompt` option.

For example, deploy the Coordination Service to all three nodes of a three-node cluster, where the nodes are node1, node2, and node3:

```
tsm topology deploy-coordination-service -n node1,node2,node3
```

Wait until the command completes and you are returned to the system prompt.

6. (Optional) If the deployment fails, you need to run the `tsm topology cleanup-coordination-service` command to remove the unsuccessfully deployed ensemble. For details on running the command, see tsm topology cleanup-coordination-service.

7. Start Tableau Server:

```
tsm start
```

For more information and details on deploying a new Coordination Service ensemble, see Deploy a Coordination Service Ensemble .

Step 7: Configure Client File Services (CFS) on additional nodes

Add CFS to additional nodes. We recommend you add CFS to every node running the Coordination Service.

Note: This operation includes steps that you may need to perform using the TSM command line.

1. On the initial node, open a terminal session.

2. Find the node ID for the node you are adding CFS to:

```
tsm topology list-nodes -v
```

3. Add CFS on the node by specifying the node, the process, and a single instance.

For example, this command adds an instance of CFS to node2:

```
tsm topology set-process -n node2 -pr clientfileservice -c 1
```

If you attempt to add an instance of CFS to a node that already is configured with CFS, an error message will let you know there is already an instance on the node.

To add CFS to additional nodes, repeat this step for each node.

4. Apply the changes:

```
tsm pending-changes apply
```

Step 8: Configure processes for node 2

1. Open TSM in a browser:

```
https://<tsm-computer-name>:8850
```

For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Configuration** tab.

3. Specify the processes and number of instances that should run on the first additional node.

In this example:

a. Select **Gateway**.

b. Set the **Application Server** (vizportal) count to 2.

c. Set the **VizQL Server** count to 2.

d. Set the **Cache Server** count to 2.

e. Set the **Backgrounder** count to 2.

Adding Backgrounder to a node will also add an instance of Data Engine if one is not already on the node.

f. Set the **Data Server** count to 2.

g. Select **File Store**.

h. Select **Repository** (pgsql).

i. Select **Metrics Service**. (The legacy Metrics feature was retired in February 2024 for Tableau Cloud and in Tableau Server version 2024.2. For more information, see Create and Troubleshoot Metrics (Retired).)

j. Select **Messaging Service**.

k. Select **Data Source Properties**.

l. Select **Content Exploration Service**.

m. Select **Collections Service**.

n. Select **Search and Index Server**.

The specific processes and process counts you set will depend on your organizational environment and needs. Some processes are added automatically when you add another process. For more information, see Configure Nodes and Tableau Server Processes.

> **Note:** The TSM Web UI limits you to a maximum of 8 instances of processes that allow you to select the number of instances. To configure more instances than this, use the command line and the `TSM topology set-process` command. For more information, see tsm topology set-process.

Step 9: Configure processes for node 3

1. In TSM, on the **Configuration** tab, specify the processes and number of instances that should run on the second additional node.

   In this example:

a. Select **Gateway**.

b. Set the **Application Server** (vizportal) count to 2.

c. Set the **VizQL Server** count to 2.

d. Set the **Cache Server** count to 2.

e. Set the **Backgrounder** count to 2.

   Adding Backgrounder to a node will also add an instance of Data Engine if one is not already on the node.

f. Set the **Data Server** count to 2.

g. Select **File Store**.

h. Select **Metrics Service**. (The legacy Metrics feature was retired in February 2024 for Tableau Cloud and in Tableau Server version 2024.2. For more information, see Create and Troubleshoot Metrics (Retired).)

i. Select **Data Source Properties**.

j. Select **Content Exploration Service**.

k. Select **Collections Service**.

l. Select **Search and Index Server**.

2. Click **Pending Changes** at the top of the page:

   The Pending Changes list displays.

3. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

Use the TSM CLI

Step 1: Install the initial node

See Install and Configure Tableau Server.

Step 2: Generate a bootstrap file for the additional nodes

To install Tableau Server on additional computers you use the same installer you did for the initial node, along with a "bootstrap" file you generate on the initial node.

Important: The bootstrap file contains a copy of the master keystore file used for encrypting the configuration secrets. The file can also embedded credentials which are valid for a pre-determined amount of time (see tabadmincontroller.auth.expiration.minutes) and serve as a session cookie. We strongly recommend that you take additional measures to secure the bootstrap file.

The following command set provides an example method to encrypt the bootstrap file output. This method is similar to the encryption process described in more detail at Securing secrets for import and export operations.

Note however, the method here must be passed as separate arguments with trailing `&& \` operators as follows:

```
mkfifo -m 600 /tmp/secure1 && \

tsm topology nodes get-bootstrap-file --file /tmp/secure1 && \

gpg --symmetric --batch --yes --passphrase-file ~/.secret-
s/pgppassphrase.txt --cipher-algo AES256 --output encrypted.enc <
/tmp/secure1 && \

rm /tmp/secure1
```

1. After installing Tableau Server on the initial node, generate the node bootstrap file.
2. On the initial node, open a terminal session.

3. Type this command to generate the bootstrap file:

   ```
   tsm topology nodes get-bootstrap-file --file <path\file>.json
   ```

   Embedded credentials are included in the bootstrap file by default. If you don't want the bootstrap file to include credentials, use the `-nec` or `--no-embedded-credentials` option:

   ```
   tsm topology nodes get-bootstrap-file --file <path\file>.json --no-embedded-credentials.
   ```

   If you want to completely disable the ability to include embedded credentials in node bootstrap files, you can set a configuration option for the server. See features.PasswordlessBootstrapInit for more details.

Step 3: Install and initialize node 2

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the new computer.

2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see Configure Local Firewall.

3. On the new node, run the Tableau Server Setup program:

   Use the package manager to install the Tableau Server package.

   You must install the new version to the same location as the existing version. The install location must be the same on all nodes. Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

   - On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

- **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

  ```
  sudo yum update

  sudo yum install tableau-server-<version>.x86_64.rpm
  ```

- **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

  Run the following command:

  ```
  sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
  ```

  Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

  ```
  sudo apt-get update
  sudo apt-get upgrade
  sudo apt-get -y install gdebi-core
  sudo gdebi -n tableau-server-<version>_amd64.deb
  ```

4. To initialize communications between your new node and the initial node, run the initialize-tsm script that is installed when you install Tableau Server.

   On the new node:

   Navigate to the `scripts` directory:

   ```
   cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
   ```

5. Run the initialize-tsm script:

   ```
   sudo ./initialize-tsm -b /path/to/<bootstrap>.json --accepteula
   ```

   - Use the `-b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in Securing secrets for import and export operations.
   - If the bootstrap file was generated without embedded credentials, use the `-u` flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see tsm topology nodes get-bootstrap-file.

     ```
     sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u
     <admin-user-on-first-node> --accepteula
     ```

     Important: You must enter the credentials of the same user that you used during the installation process on the initial node.

   - The `--accepteula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

     ```
     /opt/tableau/tableau_server/packages/docs.<version_
     code>/EULA.rtf
     ```

Step 4: Install and initialize node 3

Install Tableau Server on node 3:

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the new computer.

2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see Configure Local Firewall.

3. On the new node, run the Tableau Server Setup program:

   Use the package manager to install the Tableau Server package.

   You must install the new version to the same location as the existing version. The install location must be the same on all nodes. Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

   - On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

     - **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

       ```
       sudo yum update

       sudo yum install tableau-server-<version>.x86_64.rpm
       ```

     - **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

       Run the following command:

       ```
       sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
       ```

> **Note:** When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. To initialize communications between your new node and the initial node, run the initialize-tsm script that is installed when you install Tableau Server.

   On the new node:

   Navigate to the `scripts` directory:

   ```
   cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
   ```

5. Run the initialize-tsm script:

   ```
   sudo ./initialize-tsm -b /path/to/<bootstrap>.json --accepteula
   ```

   - Use the `-b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in Securing secrets for import and export operations.

- If the bootstrap file was generated without embedded credentials, use the `-u` flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see tsm topology nodes get-bootstrap-file.

  ```
  sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u
  <admin-user-on-first-node> --accepteula
  ```

  Important: You must enter the credentials of the same user that you used during the installation process on the initial node.

- The `--accepteula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

  ```
  /opt/tableau/tableau_server/packages/docs.<version_
  code>/EULA.rtf
  ```

Step 5: Add a process to the additional nodes

1. On the initial node, configure a cluster controller instance on each additional node:

   ```
   tsm topology set-process -n <nodeID_second-node> -pr cluster-
   controller -c 1
   ```

   ```
   tsm topology set-process -n <nodeID_third-node> -pr cluster-
   controller -c 1
   ```

2. Apply the node configuration changes:

   ```
   tsm pending-changes apply --ignore-warnings
   ```

   A warning about deploying a Coordination Service ensemble displays because you have deployed a three-node cluster. Use the `--ignore-warnings` option to apply the configuration changes in spite of the warning. You will deploy a new Coordination Service ensemble in the next step.

Step 6: Deploy a Coordination Service ensemble

If you install a total of three or more nodes, you should also deploy a Coordination Service ensemble. If you do not, you will get a warning message every time you make changes to the server configuration or topology. You can ignore this message, but as a best practice you should deploy a multi-node Coordination Service ensemble.

When you install Tableau Server, a single instance of the Coordination Service is installed on the initial node. TSM and Tableau Server depend on the Coordination Service to function properly, so to provide redundancy and ensure availability on multi-node installations, configure additional instances of the Coordination Service by deploying a Coordination Service ensemble. Coordination Service ensembles are installed with one, three, or five instances of the Coordination Service. In a three-node installation of Tableau Server, the recommended number of Coordination Service instances is three, one on each node.

Do not attempt to deploy a Coordination Service ensemble if there are other changes pending. Discard or apply any pending changes before deploying a new Coordination Service ensemble.

> **Note:** This operation includes steps that you may need to perform using the TSM command line.

The following steps illustrate how to deploy a new Coordination Service ensemble on an existing three-node Tableau Server cluster and clean up the old ensemble.

1. On the initial node, open a terminal session.
2. Stop Tableau Server:

   ```
   tsm stop
   ```

   If prompted, sign in as a TSM administrator.

Some TSM processes will continue to run, including the Administration Controller and Administration Agent.

3. Confirm there are no pending changes:

   ```
   tsm pending-changes list
   ```

   If there are pending changes, you need to either discard the changes or apply them. Applying pending changes will take some time:

   - Discard the changes

     ```
     tsm pending-changes discard
     ```

     or

   - Apply the changes:

     ```
     tsm pending-changes apply
     ```

   Wait until the command completes and you are returned to the system prompt.

4. Get the node IDs for each node in the cluster:

   ```
   tsm topology list-nodes -v
   ```

5. Use the `tsm topology deploy-coordination-service` command to add a new Coordination Service ensemble by adding the Coordination Service to specified nodes. You must specify the node(s) that the Coordination Service should be added to, using the actual node ID to identify each node. The command also makes the new ensemble the "production" ensemble (the ensemble in use) and removes the old ensemble, unless the deployment fails. If this happens, see step 6 below.

   Note: A "y/n" prompt displays confirming that a server restart will take place. To run the command without input, include the `--ignore-prompt` option.

For example, deploy the Coordination Service to all three nodes of a three-node cluster, where the nodes are node1, node2, and node3:

```
tsm topology deploy-coordination-service -n node1,node2,node3
```

Wait until the command completes and you are returned to the system prompt.

6. (Optional) If the deployment fails, you need to run the `tsm topology cleanup-coordination-service` command to remove the unsuccessfully deployed ensemble. For details on running the command, see tsm topology cleanup-coordination-service.

7. Start Tableau Server:

   ```
   tsm start
   ```

For more information and details on deploying a new Coordination Service ensemble, see Deploy a Coordination Service Ensemble .

Step 7: Configure Client File Services (CFS) on additional nodes

Add CFS to additional nodes. We recommend you add CFS to every node running the Coordination Service.

Note: This operation includes steps that you may need to perform using the TSM command line.

1. On the initial node, open a terminal session.

2. Find the node ID for the node you are adding CFS to:

   ```
   tsm topology list-nodes -v
   ```

3. Add CFS on the node by specifying the node, the process, and a single instance.

   For example, this command adds an instance of CFS to node2:

```
tsm topology set-process -n node2 -pr clientfileservice -c 1
```

If you attempt to add an instance of CFS to a node that already is configured with CFS, an error message will let you know there is already an instance on the node.

To add CFS to additional nodes, repeat this step for each node.

4. Apply the changes:

```
tsm pending-changes apply
```

Step 8: Configure processes for node 2

1. On the initial node, set the topology for node 2. The topology specifies which processes should run on the node, and how many instances of each process should run. For details about configuring nodes, see Configure Nodes. Data engine will be added automatically when you add other processes. For details on when data engine is added, see Tableau Server Processes.
   a. Get the node-id for the new node: `tsm topology list-nodes -v`
   b. Specify individual processes that will run on the new node:

```
tsm topology set-process -n <node-id> -pr gateway -c 1

tsm topology set-process -n <node-id> -pr vizqlserver -c 2

tsm topology set-process -n <node-id> -pr vizportal -c 2

tsm topology set-process -n <node-id> -pr backgrounder -c
2

tsm topology set-process -n <node-id> -pr cacheserver -c 2

tsm topology set-process -n <node-id> -pr index-
andsearchserver -c 1

tsm topology set-process -n <node-id> -pr dataserver -c 2
```

```
tsm topology set-process -n <node-id> -pr filestore -c 1

tsm topology set-process -n <node-id> -pr pgsql -c 1

tsm topology set-process -n <node-id> -pr metrics -c 1

tsm topology set-process -n <node-id> -pr activemqserver -c
1

tsm topology set-process -n <node-id> -pr tdsservice -c 1

tsm topology set-process -n <node-id> -pr con-
tentexploration -c 1

tsm topology set-process -n <node-id> -pr collections -c 1
```

2. Apply the node configuration changes. You will be prompted with a message that Tableau Server will restart.

   ```
   tsm pending-changes apply
   ```

Step 9: Configure processes for node 3

On the initial node, set the topology for node 3. The topology specifies which processes should run on the node, and how many instances of each process should run. Data engine will be added automatically when you add other processes. For details on when data engine is added, see Tableau Server Processes.

1. Get the node-id for the new node:

   ```
   tsm topology list-nodes -v
   ```

2. Specify individual processes that will run on the new node:

   ```
   tsm topology set-process -n <node-id> -pr gateway -c 1

   tsm topology set-process -n <node-id> -pr vizqlserver -c 2

   tsm topology set-process -n <node-id> -pr vizportal -c 2
   ```

```
tsm topology set-process -n <node-id> -pr backgrounder -c 2

tsm topology set-process -n <node-id> -pr cacheserver -c 2

tsm topology set-process -n <node-id> -pr indexandsearchserver
-c 1

tsm topology set-process -n <node-id> -pr dataserver -c 2

tsm topology set-process -n <node-id> -pr filestore -c 1

tsm topology set-process -n <node-id> -pr metrics -c 1

tsm topology set-process -n <node-id> -pr tdsservice -c 1

tsm topology set-process -n <node-id> -pr contentexploration -c
1

tsm topology set-process -n <node-id> -pr collections -c 1
```

3. Apply the node configuration. You will be prompted with a message that Tableau Server will restart.

   ```
   tsm pending-changes apply
   ```

4. Start the server:

   ```
   tsm start
   ```

Step 10: Configure firewall rules (optional)

If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see Local firewall configuration and Configure Local Firewall.

Add a Load Balancer

At this point, all three nodes have gateways, which are used to route requests to available server processes. All gateways are active, but to further reduce the potential for downtime in the cluster, you can configure a load balancer. For more information, see Add a Load Balancer.

Add a Load Balancer

You can enhance the reliability of Tableau Server by running gateways on multiple nodes, and configuring a load balancer to distribute requests across the gateways. Unlike the repository process, which can be active or passive, all gateway processes are active. If one gateway in a cluster becomes unavailable, the load balancer stops sending requests to it. The load balancer algorithm you choose determines how the gateways will route client requests.

- **Kerberos**: If you will be using Kerberos authentication, you need to configure Tableau Server for your load balancer before you configure Tableau Server for Kerberos. For more information, see Configure Kerberos.
- **Tested load balancers**: Tableau Server clusters with multiple gateways have been tested with Apache and F5 load balancers.

  If you are using an Apache load balancer and creating custom administrative views, you need to connect directly to the Tableau Server repository. You cannot connect through the load balancer.
- **Tableau Server URL**: When a load balancer is in front of a Tableau Server cluster, the URL that's accessed by Tableau Server users belongs to the load balancer, not the initial Tableau Server node.
- **Single load balancer endpoint**: You must configure your load balancers for a single URL endpoint. You cannot configure different endpoint hosts to redirect to the same Tableau Server deployment. The single external URL is defined in `gateway.public.host` when you configure Tableau Server, as described in Configuring

Proxies and Load Balancers for Tableau Server.

- **Trusted host settings**: The computer running the load balancer must be identified to Tableau Server as a trusted host, as described in Configuring Proxies and Load Balancers for Tableau Server.

Configure Tableau Server to work with a load balancer

The settings used to identify a load balancer to Tableau Server are the same ones that are used to identify a reverse proxy server. If your Tableau Server cluster requires both a proxy server and a load balancer, both must use a single external URL defined in `gateway.public.host` and all proxy servers and load balancers must be specified in `gateway.trusted` and `gateway.trusted_hosts`. See Configuring Proxies and Load Balancers for Tableau Server.

Deploy a Coordination Service Ensemble

The Coordination Service is built on Apache ZooKeeper, an open-source project, and coordinates activities on the server, guaranteeing a quorum in the event of a failure, and serving as the source of "truth" regarding the server topology, configuration, and state. The service is installed automatically on the initial Tableau Server node, but no additional instances are installed as you add additional nodes. Because the successful functioning of Tableau Server depends on a properly functioning Coordination Service, we recommend that for server installations of three or more nodes, you add additional instances of the Coordination Service by deploying a new Coordination Service ensemble. This provides redundancy and improved availability in the event that one instance of the Coordination Service has problems.

Important: The process to deploy a Coordination Service ensemble changed as of version 2020.1.0. If you are running an earlier version of Tableau Server, see the documentation for that version. You can find documentation for all supported versions here: Tableau Help

- Hardware requirements
- The Coordination Service Quorum
- Deploy a new Coordination Service ensemble

Hardware requirements

The hardware you use for Tableau Server can have an effect on how well the Coordination Service runs. In particular:

- **Memory**. The Coordination Service maintains state information in memory. By design, the memory footprint is small, and is typically not a factor in overall server performance.
- **Disk speed**. Because the service stores state information on disk, it benefits from fast disk speed on the individual node computers.
- **Connection speed** between nodes. The service communicates continuously between cluster nodes; a fast connection speeds between nodes helps with efficient synchronization.

Because the Coordination Service is I/O intensive, if you are running Tableau Server on computers that meet or just exceed the minimum hardware requirements, you may want to configure a Coordination Service ensemble that puts the service on nodes that are not being used for other server processes. This reduces the chance of delays due to I/O contention between server processes. For information on how to deploy an ensemble on dedicated Coordination Service-only nodes, see Configure Tableau Server for High Availability with Coordination Service-Only Nodes.

The Coordination Service Quorum

To ensure that the Coordination Service can work properly, the service requires a *quorum*—a minimum number of instances of the service. This means that the number of nodes in your installation impacts how many instances of the Coordination Service you want to configure in your ensemble.

Number of Coordination Service instances to use

The maximum number of Coordination Service instances you can have in an ensemble on Tableau Server depends on how many Tableau Server nodes you have in your deployment. Configure a Coordination Service ensemble based on these guidelines:

Total number of server nodes	**Recommended number of Coordination Service nodes in ensemble (must be 1, 3, or 5)**	**Notes**
1-2 nodes	1 node	This is the default and requires no changes unless you want to move the Coordination Service off your initial node and onto your additional node.
3-4 nodes	3 nodes	
5 or more nodes	3 nodes or 5 nodes	Five is the maximum number of Coordination Service instances you can install. A 3-node Coordination Service ensemble allows for one of the ensemble nodes to fail without causing Tableau Server to fail. A 5-node ensemble allows for two of the ensemble nodes to fail without causing Tableau Server to fail. For most installations, three Coordination Service nodes are adequate, and because of the I/O-intensive nature of the Coordination Service, this is the most performant configuration. If high availability is your absolute priority, you may want to consider deploying a 5-node Coordination Service ensemble. This provides the most redundancy in the event that one or more nodes fail but will require more system resources. A maximum of two of the ensemble nodes can fail without impacting

Total number of server nodes	Recommended number of Coordination Service nodes in ensemble (must be 1, 3, or 5)	Notes
		Tableau Server (as long as any other services on the node also exist on still-functioning nodes). To reduce performance impact, locate the Coordination Service on nodes that are running fewer other services or consider using Coordination Service-only nodes. For details, see Configure Tableau Server for High Availability with Coordination Service-Only Nodes.

Deploy a new Coordination Service ensemble

If you install a total of three or more nodes, you should also deploy a Coordination Service ensemble. If you do not, you will get a warning message every time you make changes to the server configuration or topology. You can ignore this message, but as a best practice you should deploy a multi-node Coordination Service ensemble.

When you install Tableau Server, a single instance of the Coordination Service is installed on the initial node. TSM and Tableau Server depend on the Coordination Service to function properly, so to provide redundancy and ensure availability on multi-node installations, configure additional instances of the Coordination Service by deploying a Coordination Service ensemble. Coordination Service ensembles are installed with one, three, or five instances of the Coordination Service. In a three-node installation of Tableau Server, the recommended number of Coordination Service instances is three, one on each node.

Do not attempt to deploy a Coordination Service ensemble if there are other changes pending. Discard or apply any pending changes before deploying a new Coordination Service ensemble.

Note: This operation includes steps that you may need to perform using the TSM command line.

The following steps illustrate how to deploy a new Coordination Service ensemble on an existing three-node Tableau Server cluster and clean up the old ensemble.

1. On the initial node, open a terminal session.
2. Stop Tableau Server:

   ```
   tsm stop
   ```

   If prompted, sign in as a TSM administrator.

   Some TSM processes will continue to run, including the Administration Controller and Administration Agent.
3. Confirm there are no pending changes:

   ```
   tsm pending-changes list
   ```

   If there are pending changes, you need to either discard the changes or apply them. Applying pending changes will take some time:

   - Discard the changes

     ```
     tsm pending-changes discard
     ```

     or

- Apply the changes:

  ```
  tsm pending-changes apply
  ```

Wait until the command completes and you are returned to the system prompt.

4. Get the node IDs for each node in the cluster:

   ```
   tsm topology list-nodes -v
   ```

5. Use the `tsm topology deploy-coordination-service` command to add a new Coordination Service ensemble by adding the Coordination Service to specified nodes. You must specify the node(s) that the Coordination Service should be added to, using the actual node ID to identify each node. The command also makes the new ensemble the "production" ensemble (the ensemble in use) and removes the old ensemble, unless the deployment fails. If this happens, see step 6 below.

   Note: A "y/n" prompt displays confirming that a server restart will take place. To run the command without input, include the `--ignore-prompt` option.

   For example, deploy the Coordination Service to all three nodes of a three-node cluster, where the nodes are node1, node2, and node3:

   ```
   tsm topology deploy-coordination-service -n node1,node2,node3
   ```

   Wait until the command completes and you are returned to the system prompt.

6. (Optional) If the deployment fails, you need to run the `tsm topology cleanup-coordination-service` command to remove the unsuccessfully deployed ensemble. For details on running the command, see tsm topology cleanup-coordination-service.

7. Start Tableau Server:

   ```
   tsm start
   ```

Configure Client File Service

The Client File Service (CFS) manages most shared files in a multinode cluster. For example, authentication related certificates, keys, and files (OpenID, mutual SSL, SAML, and Kerberos), and customization files are managed by CFS.

A single instance of CFS is configured on the initial node when you first install Tableau Services Manager (TSM) and Tableau Server. You can and should configure additional instances on other nodes.

In a cluster, if a node that is running your only instance of CFS fails, any files being managed by CFS will be lost, and you will need to repopulate CFS those files by reimporting certs and custom images, and making any related configuration changes. For a list of files managed by CFS, see Tableau Server Client File Service.

Keep in mind these limitations and recommendations:

- There must be at least one instance of CFS for any Tableau Server installation. You cannot remove all instances of CFS.
- We recommend you do not configure more than one instance on any one node. Additional instances would not provide any benefit and would be a waste of resources.
- We recommend you configure a CFS instance on each node where you deploy the Coordination Service. This provides redundancy and helps safeguard Tableau Server from unexpected errors in case a node fails.

This article explains how to add or remove an instance of CFS. You cannot configure CFS from the Web UI. You must use the CLI to add or remove CFS. The procedure is the same one you use for adding a process to a node.

Configure CFS on additional nodes

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are adding CFS to:

```
tsm topology list-nodes -v
```

3. Add CFS on the node by specifying the node, the process, and a single instance.

   For example, this command adds an instance of CFS to node2:

   ```
   tsm topology set-process -n node2 -pr clientfileservice -c 1
   ```

   If you attempt to add an instance of CFS to a node that already is configured with CFS, an error message will let you know there is already an instance on the node.

   To add CFS to additional nodes, repeat this step for each node.

4. Apply the changes:

   ```
   tsm pending-changes apply
   ```

Repository Failover

In a Tableau Server installation, the repository (pgsql) database is one of the key required processes. The Tableau Server repository stores information about Tableau Server users, groups and group assignments, permissions, projects, data sources, and extract metadata and refresh information. Because it is critical to the server functioning, Tableau Server has a built-in automatic "failover" for the repository when server is installed in a distributed environment that meets certain requirements.

Automatic repository failover

Automatic repository failover means that if there is a problem with the active Tableau Server repository, the server will automatically switch to using the passive repository. This does not happen immediately, to protect against ephemeral issues with the repository that don't justify a switch, but if the repository is unavailable for more than 1-5 minutes, failover occurs. For example, if the underlying PostgresSQL service fails, then failover will occur in about a minute. However other configuration problems may not trigger a failure for up to 5 minutes.

For automatic repository failover to work, your Tableau Server installation needs:

- A minimum of three nodes
- Two instances of the repository installed

  Optional but highly recommended:

- A multi-node Coordination Service ensemble deployed

With these conditions satisfied, repository failover will occur if the active repository becomes unavailable, either due to a problem with the process, or a problem with the node the process is running on. If the original repository becomes available again (if, for example, the node is restarted and all processes come up properly), it is made the passive repository, available for failover if necessary.

Manual repository failover

There may be reasons you want to shift back to the original repository after failover occurs. One reason would be if that instance of the repository is installed on a computer with more resources. To do this, use the `tsm topology failover-repository` command to manually switch back to the original repository. For more information, see tsm topology failover-repository.

Preferred active repository

When you configure Tableau Server you have the option to specify a node as the preferred active repository. When Tableau Server is configured for repository failover, the preferred active repository node is the one used for the active repository. This is an optional step, and if you do not specify a preferred active repository node, Tableau Server will select the active repository node on startup.

To configure the preferred active repository, use the `tsm configuration set` command to configure the `pgsql.preferred_host` option:

```
tsm configuration set -k pgsql.preferred_host -v "<host_name>"
```

Note: The `host_name` is case-sensitive and must match the node name shown in the output of `tsm status -v`.

Configure a preferred active repository node if you want Tableau Server to select a specific node on startup. You might want to do this if you have a particular server you want to use for your active repository (a computer with more disk space or memory for example), or if you are using custom administrative views. Custom administrative views have embedded connection information that refers to the repository for which you created the views. For more information on connecting to the Tableau Server repository, see Collect Data with the Tableau Server Repository

Recover from an Initial Node Failure

The first computer you install Tableau on, the "initial node," has some unique characteristics. Three processes run only on the initial node and cannot be moved to any other node except in a failure situation, the License Service (License Manager), Activation Service, and TSM Controller (Administration Controller). Tableau Server includes a script that automates moving these processes to one of your other existing nodes so you can get complete access back to TSM and keep Tableau Server running.

Two other processes are initially included on the initial node but can be added or moved to additional nodes, the CFS (Client File Service) and the Coordination Service. Depending on how your installation was configured with CFS and Coordination Service, you may also need to take steps to redeploy these.

If an initial node fails

If there is a problem with the initial node and you have redundant processes on your other nodes, there is no guarantee that Tableau Server will continue to run.

- Tableau Server can continue to run for up to 72 hours after an initial node failure, before the lack of the licensing service impacts other processes. If so, your users *may* be able to continue to sign in and see and use their content after the initial node fails, but you will

not be able to reconfigure Tableau Server because you won't have access to the Administration Controller.
- If you are running a version of Tableau Server 2021.4.2 (or older) that is configured for ATR, then problems with the initial node will render all server functionality unavailable. This is true whether the node has a problem or if you intentionally stop it (for instance, to do a system-level patch).

Even when configured with redundant processes, *it is possible that Tableau Server may not continue to function after the initial node fails*. This is true even when an installation is configured for high availability. This means you should make a point of moving the two unique processes to another of your running nodes as soon as possible. If your initial node fails for reasons that are recoverable in a relatively short amount of time (for example, a hardware failure you can correct), you should first attempt to bring the node back up without using the procedure below.

Note: The steps in this article require server downtime and can be disruptive, and should only be used in the event of a catastrophic failure of the initial node. If you are unable to get your initial node running again, use the following steps to move key TSM processes to another node in your cluster.

General requirements

The 2021.1 version of Tableau Server has been updated with improved recovery functionality. The procedure in this topic has been written for Tableau Server 2021.1.

If you are attempting to recover a failed node from an earlier version of Tableau Server, you must follow the procedure for that version. To view archived versions of Tableau help, see Tableau Help.

- As part of the process for setting up a multi-node Tableau Server installation you should have deployed a Coordination Service ensemble. The process below assumes there was a Coordination Ensemble deployed before there was a problem with the initial node. For more information about deploying a Coordination Service ensemble, see Deploy a Coordination Service Ensemble .

- This process assumes that you have configured instances of Client File Service (CFS) on every node that is running the Coordination Service. If you did not add additional instances of CFS, your only instance was on the initial node, and you will need to add at least one instance of CFS to another node. You will also need to repopulate CFS. Tableau Server requires at least one instance of the CFS. For more information, see Configure Client File Service and Tableau Server Client File Service.

Note: This operation includes steps that you may need to perform using the TSM command line.

Move the TSM Controller, License Service, and Activation Service to another node

If there is a problem with the initial node, the TSM Controller, the Licensing Service, and Activation Service need to be started on another node. Follow these steps to use the provided `move-tsm-controller` script and get the TSM Controller, Licensing Service, and Activation Service working on another node.

1. On a node that is still working, run the Controller recovery script. At a terminal prompt on a working node, type the following command:

   ```
   sudo /opt/tableau/tableau_server/packages/scripts.<version_
   code>/move-tsm-controller -n <nodeID>
   ```

   where "nodeID" is the ID for the node you want the TSM Controller to run on. For example:

   ```
   sudo /opt/tableau/tableau_server-
   /packages/scripts.10400.17.0802.1319/move-tsm-controller -n
   node2
   ```

2. Verify the Administration Controller is running on the node:

   ```
   tsm status -v
   ```

3. Stop Tableau Server.

   The remainder of this procedure includes some commands with the `--ignore-node-status` option. When a command is run with the `--ignore-node-status` option, the command will run without consideration of the status of the specified node. To use `--ignore-node-status` , specify the failed node:

   ```
   tsm stop --ignore-node-status <nodeID>
   ```

   For example, if node1 has failed, run the command as follows:

   ```
   tsm stop --ignore-node-status node1
   ```

4. Add the License Service to the node:

   ```
   tsm topology set-process -pr licenseservice -n <nodeID> -c 1
   ```

5. Remove the old License Service from the original node, where "nodeID" is the initial node that has failed:

   ```
   tsm topology set-process -pr licenseservice -n <nodeID> -c 0
   ```

6. If you are running one of the following versions

   - 2023.3.0 or later
   - 2023.1.3 or later
   - 2022.3.7 or later
   - 2022.1.15 or later

   or you are running an earlier version and using ATR,add the Activation Service to the new node:

   ```
   tsm topology set-process -pr activationservice -n <nodeID> -c 1
   ```

7. If you are running one of these versions or later

- 2023.3.0 or later
- 2023.1.3 or later
- 2022.3.7 or later
- 2022.1.15 or later

or you are running an earlier version and using ATR, remove the old Activation Service from the original node, where "nodeID" is the initial node that has failed:

```
tsm topology set-process -pr activationservice -n <nodeID> -c 0
```

Important: In a cluster, if a node that is running your only instance of CFS fails, any files being managed by CFS will be lost, and you will need to repopulate CFS those files by reimporting certs and custom images, and making any related configuration changes. For a list of files managed by CFS, see Tableau Server Client File Service.

8. If the initial node had been running the Messaging Service, add the Messaging Service to this node:

   ```
   tsm topology set-process -pr activemqserver -n node2 -c 1
   ```

9. (Optional) You can also add other processes that had been running on the initial node but are not running on this node. For example, to add an cache server:

   ```
   tsm topology set-process -pr cacheserver -n node2 -c 1
   ```

10. Apply the changes:

    ```
    tsm pending-changes apply --ignore-node-status <nodeID>
    ```

    If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays

even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

11. Restart the TSM Administration Controller (as *tableau* system account):

    ```
    sudo su -l tableau -c "systemctl --user restart tabadmincontroller_0.service"
    ```

    Note: It may take a few minutes for tabadmincontroller to restart. If you attempt to apply pending changes in the next step before the controller has fully restarted, TSM will not be able to connect to the controller. You can verify that the controller is running by using the `tsm status -v` command. Tableau Server Administration Controller should be listed as "is running".

12. Apply pending changes (there may not appear to be any, but this step is required):

    ```
    tsm pending-changes apply --ignore-node-status <nodeID>
    ```

13. Activate the Tableau Server license on the new Controller node:

    ```
    tsm licenses activate -k <product-key>
    ```

14. Verify the license is properly activated:

    ```
    tsm licenses list
    ```

15. If the initial node was running the Coordination Service, you need to deploy a new Coordination Service ensemble that does not include that node. If you have a three node cluster and the initial node was running the Coordination Service, you must deploy a new, single-instance Coordination Service ensemble on a different node and clean up the old ensemble. In this example, a single instance of the Coordination

Service is being deployed to the second node:

```
tsm topology deploy-coordination-service -n node2 --ignore-node-status node1
```

16. If the initial node was running a File Store instance, you need to remove that instance:

    ```
    tsm topology filestore decommission -n <nodeID> --delete-filestore
    ```

    Where `nodeID` is the initial node that has failed.

17. Apply pending changes, using the `--ignore-warnings` flag if the new Coordination Service ensemble you deployed above is a single node ensemble:

    ```
    tsm pending-changes apply --ignore-node-status node1 --ignore-warnings
    ```

18. Remove the initial node, where `nodeID` is the initial node that has failed:

    ```
    tsm topology remove-nodes -n <nodeID>
    ```

19. Apply pending changes, using the `--ignore-warnings` flag if the new Coordination Service ensemble you deployed above is a single node ensemble:

    ```
    tsm pending-changes apply --ignore-warnings
    ```

20. Start Tableau Server:

    ```
    tsm start
    ```

    At this point your server should start, and you will be able to use TSM to configure it. The next step is to replace your initial node so your cluster has the original number of nodes. How you do this depends on whether or not you want to reuse the node that failed. We recommend that you only reuse that node if you are able to identify the reason it failed, and take steps to keep the failure from recurring.

21. If you plan to reuse the original node, you first need to completely remove Tableau from it. Do this by running the `tableau-server-obliterate` script. For details on doing this, see Remove Tableau Server from Your Computer.

22. On a fresh computer, or on your original computer after completely removing Tableau, install Tableau using your original Setup program and a bootstrap file generated from the node that is now running the Administration Controller and Licensing Service. This creates an additional node you can configure as part of your cluster. For details on how to add the node, see Install and Configure Additional Nodes.

    A best practice is to configure any processes you lost when the original node failed, to make sure your cluster is fully redundant. You may want to move processes from your new initial node to the newly added additional node to duplicate your original configuration. For example, if your initial node was only running gateway and File Store, you may want to configure the new initial node the same way.

23. You should also redeploy a new Coordination Service ensemble, once you have your nodes up and running the way you want. For details, see Deploy a Coordination Service Ensemble .

24. Finally, if you have not already done this, add an instance of CFS to every node that is running the Coordination Service. For more information, see Configure Client File Service

    In a cluster, if a node that is running your only instance of CFS fails, any files being managed by CFS will be lost, and you will need to repopulate CFS those files by reimporting certs and custom images, and making any related configuration changes. For a list of files managed by CFS, see Tableau Server Client File Service.

Recover from a Node Failure

If there is a problem with one of your server nodes, and you have redundant processes on your other nodes, Tableau Server can continue to run. Your users can continue to sign in and

see and use their content after the node fails, but they may experience performance degradation as a result of the failed node. In addition, your server will be at greater risk of catastrophic failure if the bad node was running processes that are no longer redundant. This means you should make a point of removing the bad node and replacing it as soon as you can. If your node fails for reasons that are recoverable in a relatively short amount of time (for example, a hardware failure you can correct), you should first attempt to bring the node back up without using the procedure below.

Note: If the failed node is your *initial* node, there are larger implications for your Tableau Server installations. For details on how to recover from the failure of an initial node, see Recover from an Initial Node Failure.

General requirements

The 2020.1 version of Tableau Server has been updated with improved recovery functionality. The procedure in this topic has been written for Tableau Server 2020.1.

If you are attempting to recover a failed node from an earlier version of Tableau Server, you must follow the procedure for that version. To view archived versions of Tableau help, see Tableau Help.

- There is at least one functioning node with an instance of the File Store on it.
- There is at least one functioning node with a Repository on it.
- There is at least one functioning node with the Client File Service (CFS) on it.

Note: This operation includes steps that you may need to perform using the TSM command line.

Removing a Failed Node

To remove a failed node from a Tableau Server cluster:

1. Identify the failed node:

   ```
   tsm status -v
   ```

   The failed node will have a status of "ERROR" and processes will show as unavailable. The node ID is listed as "node<n>" with the machine name following it. For example, `node3`:

   ```
   node3: WIN-OO915SFASVH
                                                    Status: ERROR
                                                'Tableau Server Gateway 0' status is un
   ```

2. Stop Tableau Server.

   The remainder of this procedure includes some commands with the `--ignore-node-status` option. When a command is run with the `--ignore-node-status` option, the command will run without consideration of the status of the specified node. To use `--ignore-node-status` , specify the failed node:

   ```
   tsm stop --ignore-node-status <nodeID>
   ```

   For example, if node3 has failed, run the command as follows:

   ```
   tsm stop --ignore-node-status node3
   ```

3. Determine any key processes that were running on the node:

   - If the failed node was running the Messaging Service, you need to remove the service from the failed node and add it to a working node.

     Remove it from the failed node:

     ```
     tsm topology set-process -pr activemqserver -n <nodeID> -c
     0
     ```

     Add it to a working node:

     ```
     tsm topology set-process -pr activemqserver -n <nodeID> -c
     1
     ```

- If the failed node was running the Coordination Service, you need to deploy a new ensemble before you can remove the node:

  ```
  tsm topology deploy-coordination-service -n <good_nodeID> --ignore-node-status <failed_nodeID>
  ```

- If the failed node was running the only instance of Client File Service (CFS), you need to configure a new instance of CFS on a working node. We recommend that you configure CFS on every node that is running the Coordination Service. For detail steps, see Configure Client File Service .

- If the failed node was running File Store, you need to force-decommission File Store and remove it before you can remove the node.

  ```
  tsm topology filestore decommission -n <nodeID> --delete-filestore
  ```

  Apply pending changes (use `--ignore-warnings` option if you had a three node cluster and a single Coordination Service instance):

  ```
  tsm pending-changes apply --ignore-warnings --ignore-node-status <nodeID>
  ```

4. If the cluster was a three-node cluster and there are repositories on the remaining working nodes, you need to either remove one repository, or add a new node. This is because you are limited to a single instance of the repository when you have fewer than three nodes.

   To remove one repository:

   ```
   tsm topology set-process -n <nodeID> -pr pgsql -c 0
   ```

5. Run the command to remove the failed node. This adds the change to the pending changes list:

   ```
   tsm topology remove-nodes -n <nodeID>
   ```

6. Verify the node removal is pending:

```
tsm pending-changes list
```

7. Apply pending changes to remove the node:

   ```
   tsm pending-changes apply
   ```

8. Start Tableau Server:

   ```
   tsm start
   ```

9. Install Tableau Server on a new node and configure the node with the processes that the old, failed node had been running.

10. On a fresh computer, or on your original computer after completely removing Tableau, install Tableau using your original Setup program and a bootstrap file generated from the initial node. For details on how to do this, see Install and Configure Additional Nodes.

    A best practice is to configure any processes you lost when the original node failed, to make sure your cluster is fully redundant.

11. You should also redeploy a new Coordination Service ensemble, once you have your nodes up and running the way you want. For details, see Deploy a Coordination Service Ensemble .

12. Finally, if you have not already done this, add an instance of CFS to every node that is running the Coordination Service. For more information, see Configure Client File Service

Configure Nodes

Use the Tableau Services Manager (TSM) Web UI or CLI commands to configure the topology of a node. The initial node is configured with a default that includes all the processes used by TSM and Tableau Server. When you add additional nodes you need to specify which processes will run on those nodes, and how many instances of those processes will run. You

may also want to change the topology of the initial node, either adding instances of existing processes, or moving some of those processes to your additional nodes (this is common when setting up a distributed installation of Tableau Server).

> **Note:** You cannot remove the File Store or Repository (pgsql) if this is the only instance of that process in your cluster. You must have at least one instance of each of the processes in your Tableau Server installation, and you must add the second instance and allow it to synchronize with the first before you remove the process on the original node. For more information see Move the Repository Process and Move the File Store Process.

Use the TSM web interface

In most cases, you can make multiple changes to your server configuration. Exceptions are if you are moving or removing a File Store instance or the Repository.

To configure your Tableau Server nodes using the TSM web interface, do the following:

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850

   For more information, see Sign in to Tableau Services Manager Web UI.

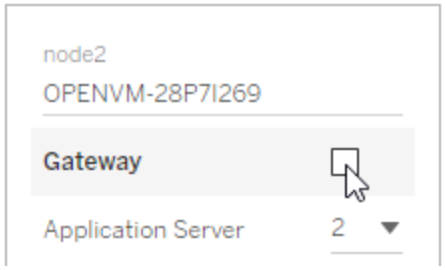
2. Click the **Configuration** tab.

   Your next steps depend on the configuration changes you want to make.

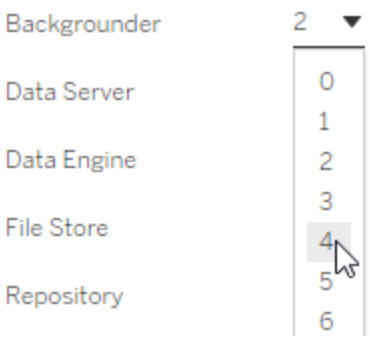
   - **Add processes**—Add processes to a node by specifying the number of instances you want on the node, or selecting the box for the process.

     For example, to add four instances of the Backgrounder and a Gateway to a node:

a. Select the **Gateway** box:



b. Set the **Backgrounder** count to 4:



Adding Backgrounder to a node will also add an instance of Data Engine if one is not already on the node.

> **Note:** The TSM Web UI limits you to a maximum of 8 instances of processes that allow you to select the number of instances. To configure more instances than this, use the command line and the `TSM topology set-process` command. For more information, see tsm topology set-process.

- **Change process count**—Change the number of processes on a node by specifying the new number, or selecting the box for the process.

- **Remove a process completely**—Remove all instances of a process from a node by clearing the box for the process, or setting the count to 0 (zero).

In most cases you move a process from one node to another by setting the process instance count on the first node to 0 (zero), and setting the count to a non-zero value on the second node.

If you are attempting to make a configuration that is not allowed (if, for example, you try to remove a File Store that has not been decommissioned), a message displays to let you know this.

3. Click **Pending Changes** at the top right, and **Apply Changes and Restart** to commit the changes and restart Tableau Server.

Use the TSM CLI

- Adding processes to a node
- Changing the number of processes on a node
- Removing all instances of a process from a node
- Moving all instances of a process from one node to another node

To configure nodes, run commands from the initial node and use the node ID to specify which node you are configuring. To determine the node ID, use the `tsm topology list-nodes` command. Use the `tsm topology set-process` command to add, update or remove a process on a node. You need to specify the node you are configuring, the process you are adding, updating, or removing, and the number of instances of the process. After setting the topology for a node you need to apply the changes to Tableau Server.

Apply changes using the `tsm pending-changes apply` command. After the changes are applied, Tableau Server is returned to the state it was in before the command was run. This means that if it was running, it will be restarted, and if it was stopped it will remain stopped after pending changes have been applied. In most cases, if Tableau Server is running when you

apply pending changes, the server is stopped so that changes can be applied, and then restarted. The exception is if you are changing the number of instances of Backgrounder, or VizQL Server on an existing node. With changes to those processes on an existing node, Tableau Server does not have to be stopped if it is running.

You need the node ID for a node in order to configure the node. To determine the node ID, use this command:

```
tsm topology list-nodes -v
```

> **Note:** Examples here show some process names. For a complete list, see Tableau Server Processes.

Adding processes to a node

Use the `tsm topology set-process` command to add a process to a node. You need to specify the node you are configuring, the process you are adding, and the number of instances of the process.

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are changing:

   ```
   tsm topology list-nodes -v
   ```

3. Add processes on the node by specifying the process and the number of instances.

   For example, this command adds two instances of backgrounder to node1:

   ```
   tsm topology set-process -n node1 -pr backgrounder -c 2
   ```

4. Apply the changes:

   ```
   tsm pending-changes apply
   ```

Changing the number of processes on a node

Change the number of processes on a node by specifying an already configured process and providing a new value for the number of instances.

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are changing:

   ```
   tsm topology list-nodes -v
   ```

3. Change the number of processes on the node by specifying an already configured process and providing a new value for the number of instances.

   For example, on a node (node1) that is already running backgrounder, this command changes the number of instances to four:

   ```
   tsm topology set-process -n node1 -pr backgrounder -c 4
   ```

4. Apply the changes:

   ```
   tsm pending-changes apply
   ```

Removing all instances of a process from a node

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are changing:

   ```
   tsm topology list-nodes -v
   ```

3. Remove a process from a node by specifying a count of 0 instances for that process on the node.

   For example, this command removes the backgrounder process from node1:

   ```
   tsm topology set-process -n node1 -pr backgrounder -c 0
   ```

4. Apply the changes:

   ```
   tsm pending-changes apply
   ```

Moving all instances of a process from one node to another node

In most cases you move a process from one node to another by setting the process instance count on the first node to zero, and setting the count to a non-zero value on the second node.

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are changing:

   ```
   tsm topology list-nodes -v
   ```

3. Move a process from one node to another node by specifying a count of 0 instances for that process on the first node and specifying a count of 1 or greater to the second node.

   For example, these commands remove Backgrounder from node1 and add two instances of it to node2:

   ```
   tsm topology set-process -n node1 -pr backgrounder -c 0

   tsm topology set-process -n node2 -pr backgrounder -c 2
   ```

4. Apply the changes:

   ```
   tsm pending-changes apply
   ```

Workload Management through Node Roles

Using node roles, you can configure where certain types of workloads are processed on your Tableau Server installation. The node roles features allows you to dedicate and scale resources to specific workloads. You can configure node roles for Backgrounder and File Store.

The Backgrounder node role specifies the type of background tasks that should run on a node, whereas the File Store node role specifies the type extract workload that should run on

a node. Both node roles are specified at a node level. Although these node roles can work independently to optimize selected workload, the two node roles in combination can be used to specialize server nodes to preferentially execute selected workloads to optimize performance extract heavy workloads. This combination is discussed in more detail later in the File Store node roles section.

Backgrounder node roles

The Backgrounder process runs Tableau Server tasks, including extract refreshes, subscriptions, flow tasks, 'Run Now' tasks, and tasks initiated from *tabcmd*. Running all these tasks can use a lot of machine resources. If you have more than one Backgrounder node in your cluster, you can manage your Backgrounder workload by specifying the type of tasks a Backgrounder can run on a node using the Backgrounder node role feature.

This configuration option is currently available only through TSM CLI commands and is only useful on multi-node clusters. If you have only one node, the Backgrounder is set to run all tasks by default and that cannot be changed.

Using Backgrounder node roles

The Backgrounder node role feature is intended to give you more control and governance over where certain type of Backgrounder workloads are processed in your Tableau Server installation and allows you to dedicate and scale resources to specific workloads.

For example, if your deployment is heavy on extract and users are running a lot of extract refreshes or encryption jobs, it could be beneficial to dedicate a node to extract refreshes. Similarly, in the case of subscriptions, if your Tableau Server installation processes a lot of subscriptions and you want to ensure that other jobs do not take resources from subscriptions, then you can dedicate a node to subscriptions. In these cases, you would also want to dedicate other backgrounder nodes to workloads other than extract refreshes or subscriptions.

To support high availability, Tableau recommends having multiple nodes that are dedicated towards a specific workload. For example, if you dedicate a node to extract refreshes, you should also configure a second node to process extract refresh workload. This way if a node

dedicated to extract refreshes becomes unavailable, extract refreshes can still be processed by the other node.

Configuration options

Configuration	Jobs
all-jobs (default)	All Tableau Server jobs
flows	Flow run jobs.
no-flows	All jobs except flows.
extract-refreshes	Jobs that are created for: Incremental refreshes, full refreshes, encryption and decryption of all extracts including extracts that flow outputs create.
subscriptions	Subscription jobs
system	System maintenance jobs that interact with other Tableau Server processes. For example, cleaning crashed jobs, reaping database events, and synching Active Directory.
extract-refreshes-and-subscriptions	Extract-refreshes, encryption and decryption of all extracts including extracts that flow outputs generate, and subscription jobs.
no-extract-refreshes	All jobs except extract-refreshes, extract encryption and decryption of all extracts including extracts created from flow outputs.
no-subscriptions	All jobs except subscriptions.
no-extract-refreshes-and-subscriptions	All jobs except extract-refreshes, encryption and decryption of all extracts including extracts created from flow outputs, and subscriptions.
no-system	All jobs except system maintenance jobs.

For more information on how to use the tsm commands to set the node role, see tsm topology.

> **Note:** Making configurations to node roles require a restart of the server and will require some downtime. For more information, see tsm pending-changes.

License requirements

Configuring a node to do only a specific type of tasks, like, flows, extract refreshes, and subscriptions, you must have one of the following licenses activated on your Tableau Server:

- To configure a node to run flows, you must have a valid Data Management license activated on your server, and have Tableau Prep Conductor running on that node. To learn more about Tableau Prep Conductor, see Tableau Prep Conductor.
- To configure a node to run extract refreshes, subscriptions, and any combination related to extract refreshes and subscriptions you must haveAdvanced Management capabilities enabled on your Tableau Server. If the license expires or is deactivated, you will see an error any time you make a change to the Server configuration. For more information on Advanced Management. see About Tableau Advanced Management on Tableau Server.

Important!
While flows, extract refreshes, and subscriptions can be expensive and resource heavy, they are not the only jobs that may require dedicated resources. In the **all jobs** group, there are a variety of System jobs that the Backgrounder executes, such as thumbnail generation for workbooks. Make sure that the nodes that run jobs other than extract refreshes, subscriptions, or flows have enough machine resources.

For more information on configuring node roles using TSM commands, see tsm topology set-node-role.

Considerations

There are some rules you must consider when configuring Backgrounder node roles, which are listed below:

- Only one node role configuration can be set for a node at a time. You cannot configure multiple node roles on a node.
- To configure a node role, there must be at least one Backgrounder process on that node.
- If you have only one Backgrounder node, you must configure this node to run all jobs. This is the default configuration and does not require additional licensing.
- If you have more than one Backgrounder node, combined, they must be configured to handle all jobs. This can be achieved in the following ways:
    - Configure one of the nodes to run all jobs using the all jobs option. This is the easiest and most straightforward way.
    - Using one of the exception configurations on one of the nodes:
        - no-flows
        - no-subscriptions
        - no-extract-refreshes
        - no-extract-refreshes-and-subscriptions

For example, in a cluster where there are three backgrounders, you could have one node configured to run flows, one to run subscriptions and extract refreshes, and one to run all jobs except flows, subscription and extract refreshes.

Note: The ability to specify node roles to run flows, or run all jobs except flows, or run all jobs was introduced in 2019.1.

File Store node roles

The Tableau Server File Store controls the storage of extracts. There are three broad categories of workloads that are extract dependent.

Extract Workload	Execution Service
Refresh	Backgrounder
Query	Data Engine
Backup/Restore	Backup/Restore

File Store node role management in combination with Backgrounder node role management gives server admins the ability to specialize server nodes to preferentially execute selected workloads to optimize performance of all categories of extract heavy workloads.

It is possible to specialize a node to execute extract query workloads through a topology that has only stand-alone Data Engine nodes. For more information, see Optimize for Extract Query-Heavy Environments. However, this is at the expense of extract refresh workloads, which are executed by Backgrounder nodes. With the topology-based isolation approach, extract refresh heavy Backgrounder workloads can get slower as none of the Backgrounder nodes have a File Store and thus all extract refresh traffic goes over the network.

With the File Store Node Role configuration option, it is possible to designate certain server nodes that process extract queries to be preferentially selected from the list of server nodes that can do so. This helps speed up workloads such as backup and extract refreshes by allowing server admins to enable File Store on Backgrounder server nodes, which prevents extract queries from running on these nodes. This feature is useful if you have an extract-heavy query workload and an extract-heavy refresh workload and want to achieve optimal extract query and refresh performance.

Guidelines to optimize for extract refresh and backup or restore workloads.

Start from a topology with specialized Data Engine nodes (see Optimize for Extract Query-Heavy Environments).

Note: In the below diagram and procedure, node 1 is Initial Node, node 2 is Additional Node 1, node 3 is Additional Node 2, and node 4 is Additional Node 3.

Process	Initial Node	Additional Node 1	Additional Node 2	Additional Node 3
Cluster Controller	✓	✓	✓	✓
Gateway	✓	✓		
Application Server	✓	✓		
VizQL Server	✓ ✓	✓ ✓		
Cache Server	✓ ✓	✓ ✓		
Search & Browse	✓	✓		
Backgrounder	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓		
Data Server	✓ ✓	✓ ✓		
Data Engine	✓	✓	✓	✓
File Store			✓	✓
Repository	✓	✓		

Topology 1 - Dedicated Data Engine Nodes

1. Add File Store to Node 1.

   ```
   tsm topology set-process -n node1 -pr filestore -c 1
   ```

2. Designate Node 3 and Node 4 to preferentially execute extract-query workloads

   ```
   tsm topology set-node-role -n node3, node4 -r extract-queries
   ```

3. Designate Node 1 to preferentially execute extract-refresh workloads.

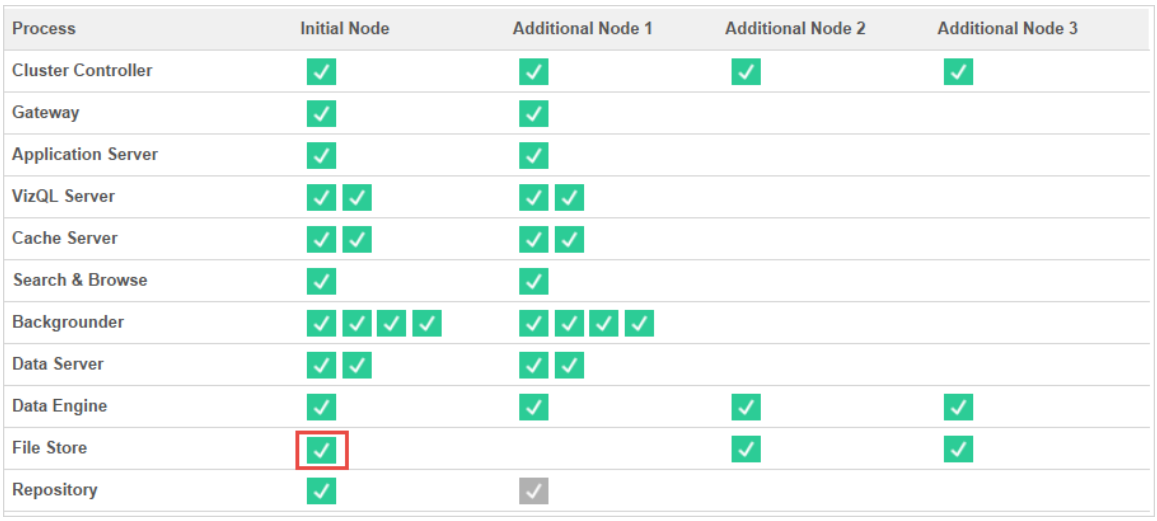
   ```
   tsm topology set-node-role -n node1 -r extract-refreshes
   ```

4. Designate Node 2 to preferentially execute non-extract-refresh workloads.

   ```
   tsm topology set-node-role -n node2 -r no-extract-refreshes
   ```

5. Apply pending changes.

   ```
   tsm pending-changes apply
   ```



Process	Initial Node	Additional Node 1	Additional Node 2	Additional Node 3
Cluster Controller	✓	✓	✓	✓
Gateway	✓	✓		
Application Server	✓	✓		
VizQL Server	✓ ✓	✓ ✓		
Cache Server	✓ ✓	✓ ✓		
Search & Browse	✓	✓		
Backgrounder	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓		
Data Server	✓ ✓	✓ ✓		
Data Engine	✓	✓	✓	✓
File Store	✓		✓	✓
Repository	✓	✓		

Topology 2 - Extra File Store Node

> **Note:** In your Tableau Server deployment, adding File Store roles to existing nodes will temporarily increase network I/O between all File Store nodes while the new File Store is being synchronized. The duration of this operation is dependent on the volume of data on the File Store and the network bandwidth capacity. The status of synchronization can be monitored using the TSM Web GUI. If you are adding more than one File Store to your deployment, it is recommended to add them consecutively and wait for the initial synchronization to complete in between each File Store addition.

Fine tune extract query workload management

When extract queries for email subscriptions and metric alerts are running at the same time that users are interactively viewing extract-based visualizations, users may experience slower than normal viz load times. Use the following node roles to fine tune how these workloads are prioritized.

Node role to use	Type of extract query workload	Example
`extract-queries`	scheduled	email subscriptions and metric alerts

`extract-queries-interactive`	interactive	users viewing an extract-based visualization

If your server deployment is seeing growth in email subscriptions and metric alerts, you can add nodes and assign the `extract-queries` node role, which makes them more available to handle subscriptions and alerts.

If your server deployment is seeing growth in users viewing extract-based visualizations, you can add nodes and assign the `extract-queries-interactive` node role, which makes them prioritize interactive extract queries to reduce extract-based viz load times. The `extract-queries-interactive` node role is a preference and not strict isolation. This means that queries will be routed to nodes that have the `extract-queries-interactive` node role assigned. If you have multiple nodes with the `extract-queries-interactive` role, queries will be routed based on node health.

For example, add a node and designate it to preferentially execute `extract-queries-interactive` workloads.

- `tsm topology set-node-role -n node4 -r extract-queries-interactive`

Configuration options

Configuration	**Jobs**
all-jobs (default)	All Tableau Server jobs
extract-queries	Jobs that are created for extract queries. The nodes selected will run as all-jobs and will prioritize the processing of extract queries.
extract-queries-interactive	Jobs that are created for extract queries. The nodes selected will run as all-jobs and will prioritize the processing of interactive extract queries, such as those that run when a user is looking at their screen and waiting for an extract-based dashboard to load. This is an advanced setting and it should only be used if the cluster has a heavy subscription and alert job workload that causes users to experience

	degraded performance on viz load times that run around the same time as scheduled loads.

For more information on configuring node roles using TSM commands, see tsm topology set-node-role.

License requirements

To configure a node to run extract queries you must have a valid Advanced Management license activated on your Tableau Server.

How to see node roles

Use the following command to see what node roles are currently configured on Tableau Server:

```
tsm topology list-nodes -v
```

Who can do this

Tableau Server Administrators can configure node roles and activate any required product keys.

Install Tableau Server on a Two-Node Cluster

When you install Tableau Server on a two-node cluster, you can install server processes on one or both nodes. A two-node cluster can improve the performance of Tableau Server, because the work is spread across multiple machines.

Note the following about two-node clusters:

- A two-node cluster does not provide failover or support for high availability.
- You can't install more than one instance of the repository on a two-node cluster, and the repository must be on the initial node.

If you need failover or high availability, or want a second instance of the repository, you must install Tableau Server on a cluster of at least three computers. In a cluster that includes at least three nodes, you can configure two instances of the repository, which gives your cluster failover capability.

Restart Multi-Node Tableau Server Computers

Restarting the computers running a multi-node installation of Tableau Server requires that you follow a few specific steps.

To restart computers running a multi-node installation of Tableau Server:

1. Stop Tableau Server. To do this:
    a. On the initial node, open a terminal session.
    b. Run this tsm command:

        ```
        tsm stop
        ```

    c. Wait until Tableau Server is stopped.
2. Restart the additional node computers. These are all the other computers *except* for the one running TSM Controller.
3. Wait until each of the additional node computers has completely restarted, including the Tableau Services Manager (TSM) processes installed there.
4. Restart the initial node computer (the computer running TSM Controller). When it has completely restarted and Tableau Server is running, connections to each additional node should be restored.

Maintain a Distributed Environment

After you set up an initial node and one or more additional nodes for a distributed installation, you can perform all subsequent configuration and updates from the initial node, using the TSM CLI, or from any computer using a browser and the TSM Web interface.

When you install additional nodes, they are added by computer name. If the computer name of a node changes, you will need to remove and reinstall the node. For details on removing a node, see Remove a Node.

You can monitor the status of the Tableau Server cluster on the TSM Status page. See View Server Process Status for details.

Additional actions you may need to maintain your distributed environment include:

Move the Repository Process

Tableau Server relies on the PostgreSQL repository to store server data. There must always be at least one active instance of the repository in any Tableau Server installation, and you can have a maximum of two instances (one active, one passive) if you have at least three nodes in your installation. You cannot remove a repository instance if it is the only instance.

This means that if you want to move the only instance of your repository from one node to another node, you need to add a second instance and synchronize the new repository with the old one before you remove the old one. Synchronize repository instances by starting server. If you've added a new repository, it will automatically synchronize with the existing instance.

If you are deleting a node from your server cluster and that node is hosting the only instance of the repository, you must add a second instance of the repository and synchronize the instances before removing the node.

If you are also moving the file store, you can move the repository at the same time. See Move the File Store Process.

Before making a change to the repository, create a full backup of Tableau Server. For more information, see tsm maintenance backup.

Important: You cannot add a second repository instance and remove the first one in the same step. You must have both running so the contents of the first is synchronized with the second, before you remove the original instance.

The steps for moving the repository are:

1. Add a new instance of the repository to another node, start server, and wait for it to synchronize with the first repository.
2. Remove the instance of the repository from the original node.

Use the TSM web interface

To move a repository you need to first add a second instance on a second node, and then after the two instances have synchronized all the data in the original repository, remove the original instance. These steps must be done separately to allow for the synchronization of content between the two instances.

Add a new instance of the repository.

1. Open TSM in a browser:

   `https://<tsm-computer-name>:8850`

   For more information, see Sign in to Tableau Services Manager Web UI.
2. Click the **Configuration** tab.
3. For the node you are adding the repository to:

   Select **Repository** (pgsql).
4. Click **Pending Changes** at the top of the page:

The Pending Changes list displays.

If you are configuring a three- or five-node cluster and have not deployed a Coordination Service ensemble, a warning will display. You can continue, and deploy a Coordination Service ensemble in a separate step. For details on deploying a Coordination Service ensemble, see Deploy a Coordination Service Ensemble .

5. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.
6. After Tableau Server has restarted, on the **Status** tab, verify that all processes are active.

Remove an instance of the repository.

1. In TSM, on the **Status** tab, verify that all processes are active. When both repositories show as Active, you can remove the first one.
2. Click the **Configuration** tab.
3. For the node you're removing the repository from, clear the **Repository** box.
4. Click **Pending Changes** at the top of the page.

   If you are configuring a three- or five-node cluster and have not deployed a Coordination Service ensemble, a warning will display. You can continue, and deploy a Coordination Service ensemble in a separate step. For details on deploying a Coordination Service ensemble, see Deploy a Coordination Service Ensemble .

5. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

Use the TSM CLI

To move a repository you need to first add a second instance on a second node, and then after the two instances have synchronized all the data in the original repository, remove the original

instance. These steps must be done separately to allow for the synchronization of content between the two instances.

Add a new instance of the repository.

1. Add the repository (pgsql) to another node:

   ```
   tsm topology set-process -n <nodeID> -pr pgsql -c 1
   ```

2. Apply the changes. If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

   ```
   tsm pending-changes apply
   ```

3. Wait for the new repository on the second node to synchronize with the repository on the first node.

   ```
   tsm status -v
   ```

   Wait until the new repository status shows as "passive".

Remove an instance of the repository.

Once the new instance of the repository is fully synchronized and shows as "passive" you can remove the original instance:

1. Remove the repository from the first node by setting the process count to 0 (zero):

   ```
   tsm topology set-process -n <nodeID> -pr pgsql -c 0
   ```

2. Apply the change. If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not

change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

```
tsm pending-changes apply
```

Move the File Store Process

This topic describes File Store process when configured to run locally on Tableau Server. However, File Store can be run locally as well as external to Tableau Server. For more information on Tableau Server External File Store, see Tableau Server External File Store.

You cannot remove an instance of File Store if it is the only instance on the server. You cannot run Tableau Server without at least one instance of File Store. This means if you need to move the File Store, or if you are deleting a server node that is hosting the only instance of the File Store, you must first move File Store to another node.

Moving the File Store is a two-part process:

- Adding a second instance of File Store (if there is not an existing second instance).
- Decommissioning and removing the original instance of File Store.

This article assumes you have installed Tableau Server on an initial node and at least one additional node. For more information on adding nodes to Tableau Server, see Install and Configure Additional Nodes.

Use the TSM web interface

Adding a second instance of File Store

This procedure assumes you have added an additional node. For more information on adding nodes to Tableau Server, see Install and Configure Additional Nodes.

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850

   For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Configuration** tab.
3. For the node you want to add an instance of File Store on, select **File Store**.

   Adding File Store to a node will also add an instance of Data Engine if one is not already on the node.
4. Click **Pending Changes** at the top of the page:

   The Pending Changes list displays.

   A Coordination Service ensemble warning displays because you are configuring a three-node cluster. You can continue. You will deploy a Coordination Service ensemble in an upcoming step.
5. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

Decommissioning and removing an instance of file store

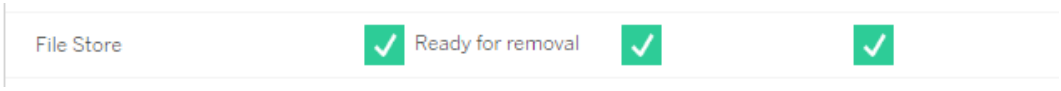
1. In TSM, on the **Status** tab, verify that all processes are active.
2. Click the **Configuration** tab.
3. For the node you're removing File Store from, clear the **File Store** box.

   This will generate a warning about decommissioning the File Store before removing it. You cannot remove an instance of File Store unless it has been decommissioned. Click **OK** to decommission and remove the file store. Data Engine will also be removed unless an instance of one of these processes is installed on the node: VizQL Server, Application Server (Vizportal), Data Server, or Backgrounder.
4. If you are removing File Store from a node that also has the TSM Administrative Controller (usually the initial node), a warning will caution you about impact to server

backup performance.

Click **Continue** to decommission File Store.

5. Click the **Status** tab to see the status of the decommission. When the instance of File Store is marked "Ready for removal" you can continue.



6. Click **Pending Changes** at the top of the page.

   Note: If you are configuring a three- or five-node cluster, a Coordination Service ensemble warning will display. You can continue to apply pending changes and deploy a Coordination Service ensemble in a separate step. For information on deploying a Coordination Service ensemble, see Deploy a Coordination Service Ensemble .

7. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

Use the TSM CLI

Adding a second instance of file store

1. Create a full backup of Tableau Server. For more information, see Back up Tableau Server Data.

2. Add the File Store to a second node.

   ```
   tsm topology set-process -n <nodeID> -pr filestore -c 1
   ```

   The File Store is automatically added. Data engine is also added if it is not already on the node.

   Apply the configuration changes:

   ```
   tsm pending-changes apply
   ```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

3. Check the status of the new file store instance.

   ```
   tsm status -v
   ```

   Wait for the new File Store to synchronize with the File Store on the first node. When synchronization is complete the new File Store has a status of "running" instead of "synchronizing".

Decommissioning and removing an instance of File Store

Once you have a second instance of File Store installed and synchronized you can decommission and remove the original instance. You must decommission the original instance before you remove it. Doing this guarantees that any unique files on the File Store node are duplicated to another file store node.

1. Decommission the original File Store:

   ```
   tsm topology filestore decommission -n <nodeID> --override
   ```

2. When the decommission command completes, remove the File Store from the node by applying the pending configuration changes. The File Store is automatically removed. Data Engine is also removed unless an instance of one of these processes is installed on the node: VizQL Server, Application Server (Vizportal), Data Server, or Backgrounder.

   ```
   tsm pending-changes apply
   ```

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays

even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Move the Messaging Service Process

This article assumes you have installed Tableau Server on an initial node and at least one additional node. For more information on adding nodes to Tableau Server, see Install and Configure Additional Nodes.

You cannot remove an instance of the Messaging Service if it is the only instance on the server. You cannot run Tableau Server without one instance of the Messaging Service. This means if you need to move the Messaging Service, or if you are deleting a server node that is hosting the Messaging Service, you must first move the service to another node.

Moving the Messaging Service is straightforward process of three steps:

- Remove the original instance of the Messaging Service.
- Add a new instance of the Messaging Service.
- Apply the pending changes.

Use the TSM web interface

Moving the Messaging Service

1. In TSM, on the **Status** tab, verify that all processes are active.
2. Click the **Configuration** tab.
3. For the node you're removing the Messaging Service from, clear the **Messaging Service** box.

This will activate the **Pending Changes** button, but in Pending Changes an error will tell you that the Messaging Service (activemqserver) is not on any node. Until you add it to another node, you cannot apply pending changes.

4. For the node you are adding the Messaging Service to, select the **Messaging Service** box.
5. Click **Pending Changes** at the top of the page.
6. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

Use the TSM CLI

Moving the Messaging Service

1. On the initial node, open a terminal session.
2. Find the node ID for the nodes you are changing:

   ```
   tsm topology list-nodes -v
   ```

3. Remove the Messaging Service from one node:

   ```
   tsm topology set-process -n <nodeID> -pr activemqserver -c 0
   ```

4. Add the Messaging Service to another node:

   ```
   tsm topology set-process -n <nodeID> -pr activemqserver -c 1
   ```

5. Apply the configuration changes:

   ```
   tsm pending-changes apply
   ```

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart

behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Remove a Node

If your Tableau Server installation includes a node you no longer need, you can remove it to simplify your installation, and to free up the hardware resources on that node.

Prerequisites for removing a node

There are prerequisites you must satisfy before you can remove a node from your Tableau Server cluster. If your node has one of a number of limitations, you need to address these before you can remove the node. The limitations include confirm that the node has been added with at least one process, and that the node does not include any process that is not also installed on another node.

If one of the following configuration limitations applies, you must take action before you can remove the node:

- If the node was just added using the Web UI, you need to apply pending changes before you can remove it. If you added it using the command line, you need to configure it with at least one process before you can remove it.
- If the node includes the only Repository instance, you need to move the Repository to another node. See Move the Repository Process.
- If the node is running the only instance of the File Store, you need to move the File Store to another node. See Move the File Store Process.
- If the node is running an instance of the Coordination Service, you must deploy a new Coordination Service ensemble that does not include the node. See Deploy a Coordination Service Ensemble .
- If the node is running the Messaging Service, you need to move the Messaging Service to another node. See Move the Messaging Service Process.

Important: Do not use the `tableau-server-obliterate` script to remove a node. First remove the node using the TSM UI or the tsm topology remove-nodes command. This leaves the rest of the cluster in a good state. Later, if you want to re-add the node to the cluster, run the `tableau-server-obliterate` script on the node to completely remove Tableau. After removing Tableau from the node, restart the computer and then re-add the node using the normal steps for adding and configuring a node. For details on running the script, see Remove Tableau Server from Your Computer.

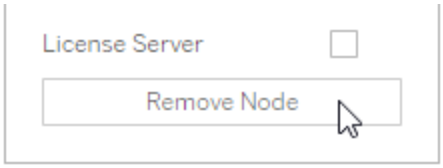
Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850

   For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Configuration** tab and, under the node you want to remove, click **Remove Node**:

   If a configuration limitation does not allow you to remove the node (if, for example, it includes a File Store that must first be decommissioned), a message will display letting you know this. See **Prerequisites** above for more information.

   > **Note:** If you just added the node, and have not configured it, you must first apply pending changes before you can remove the node.

3. Click **Pending Changes** at the top right, and **Apply Changes and Restart**.

Use the TSM CLI

Use the `tsm topology remove-nodes` command to remove a node from a cluster.

To remove a node from a cluster it must have been configured with a process at some point in the past. If you added a node using the CLI and want to remove it but have not configured any processes, you must add a process on it, run the `tsm pending-changes apply` command, and then remove the node. For example, you might add one instance of Cluster Controller to the node: `tsm topology set-process -n <nodeID> -pr clustercontroller -c 1`.

If you are removing a node you added using the Web UI, the Cluster Controller process is automatically added so you do not need to add it before removing the node.

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are changing:

   ```
   tsm topology list-nodes -v
   ```

3. Remove a node using the remove-nodes command.

   For example, to remove `node2` from an existing cluster:

   ```
   tsm topology remove-nodes --node-names "node2"
   ```

   If a configuration limitation does not allow you to remove the node (if, for example, it includes a File Store that must first be decommissioned), a message will display letting you know this. See **Prerequisites** above for more information.

4. Apply the changes:

   ```
   tsm pending-changes apply
   ```

Configure Tableau Server for High Availability with Coordination Service-Only Nodes

The Coordination Service is built on Apache ZooKeeper, an open-source project, and coordinates activities on the server, guaranteeing a quorum in the event of a failure, and serving as the source of "truth" regarding the server topology, configuration, and state. The service is installed automatically on the initial Tableau Server node, but no additional instances are installed as you add additional nodes. Because the successful functioning of Tableau Server depends on a properly functioning Coordination Service, we recommend that for server installations of three or more nodes, you add additional instances of the Coordination Service by deploying a new Coordination Service ensemble. This provides redundancy and improved availability in the event that one instance of the Coordination Service has problems.

The Coordination Service can generate a large amount of I/O as it communicates with other components of the server, so if you are running Tableau Server on computers that meet or just exceed the minimum hardware requirements, you may want to install Tableau Server in a configuration that uses Coordination Service-only nodes. This means installing Coordination Service on nodes that run no other server processes, and removing Coordination Service from any nodes that are running other server processes. This procedure explains how to do this. You can also run the Coordination Service ensemble on the same nodes running other Tableau Server processes. For details on how to do that, see Deploy a Coordination Service Ensemble .

Important: The process to deploy a Coordination Service ensemble changed as of version 2020.1.0. If you are running an earlier version of Tableau Server, see the documentation for that version. You can find documentation for all supported versions here: Tableau Help

Prerequisite

Before proceeding with the procedures in this topic, complete the following prerequisites:

- Install and Configure Tableau Server - Install Tableau on your initial node.
- Install and Configure Additional Nodes - Install Tableau on at least two additional nodes.

Note: This operation includes steps that you may need to perform using the TSM command line.

Deploy an ensemble on Coordination Service-only nodes

One way to accommodate the high I/O impact of the Coordination Service is to deploy an ensemble on nodes that only run the Coordination Service and the Cluster Controller. The following steps illustrate how to deploy a Coordination Service ensemble on an existing multi-node Tableau Server cluster.

Note: For a core-based Tableau Server license, Coordination Service-only nodes do not count against the total count of licensed cores.

1. Add additional nodes to your cluster.

   See Install and Configure Additional Nodes.

2. If you added the new nodes using the TSM CLI, you need to configure the nodes with Cluster Controller (this step is not necessary if you added the nodes using the TSM Web UI because Cluster Controller is automatically added when you add a node with the Web UI).

   On the initial node, open a terminal session.

3. From the initial node of the cluster, configure the new nodes with an instance of the Cluster Controller:

   ```
   tsm topology set-process -pr clustercontroller -n <node4> -c 1

   tsm topology set-process -pr clustercontroller -n <node5> -c 1
   ```

```
tsm topology set-process -pr clustercontroller -n <node6> -c 1
```

If prompted, sign in as a TSM administrator.

4. Apply the configuration changes. If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

   ```
   tsm pending-changes apply
   ```

   A warning about deploying a Coordination Service ensemble displays because you have deployed a multi-node cluster. If this is the only warning, you can safely override it using the `--ignore-warnings` option to apply the configuration changes in spite of the warning.

   ```
   tsm pending-changes apply --ignore-warnings
   ```

5. Confirm that all nodes are up and running:

   ```
   tsm status -v
   ```

6. On the initial node of the cluster, open a terminal session and type this command to stop Tableau Server:

   ```
   tsm stop
   ```

7. Get the node IDs for each node in the cluster:

   ```
   tsm topology list-nodes -v
   ```

8. Use the `tsm topology deploy-coordination-service` command to add a new Coordination Service ensemble by adding the Coordination Service to specified

nodes. You must specify the node(s) that the Coordination Service should be added to. The command also makes the new ensemble the "production" ensemble (the ensemble in use) and removes the old ensemble.

Note: A "y/n" prompt displays confirming that a server restart will take place. To run the command without input, include the `--ignore-prompt` option.

For example, deploy the Coordination Service to three nodes of a six-node cluster:

```
tsm topology deploy-coordination-service -n <node4,node5,node6>
```

Wait until the command completes and you are returned to the system prompt.

9. Start Tableau Server:

```
tsm start
```

Add a Load Balancer

You can enhance the reliability of Tableau Server by running gateways on multiple nodes, and configuring a load balancer to distribute requests across the gateways. Unlike the repository process, which can be active or passive, all gateway processes are active. If one gateway in a cluster becomes unavailable, the load balancer stops sending requests to it. The load balancer algorithm you choose determines how the gateways will route client requests.

- **Kerberos**: If you will be using Kerberos authentication, you need to configure Tableau Server for your load balancer before you configure Tableau Server for Kerberos. For more information, see Configure Kerberos.
- **Tested load balancers**: Tableau Server clusters with multiple gateways have been tested with Apache and F5 load balancers.

  If you are using an Apache load balancer and creating custom administrative views, you need to connect directly to the Tableau Server repository. You cannot connect through the load balancer.

- **Tableau Server URL**: When a load balancer is in front of a Tableau Server cluster, the URL that's accessed by Tableau Server users belongs to the load balancer, not the initial Tableau Server node.
- **Single load balancer endpoint**: You must configure your load balancers for a single URL endpoint. You cannot configure different endpoint hosts to redirect to the same Tableau Server deployment. The single external URL is defined in `gateway.public.host` when you configure Tableau Server, as described in Configuring Proxies and Load Balancers for Tableau Server.
- **Trusted host settings**: The computer running the load balancer must be identified to Tableau Server as a trusted host, as described in Configuring Proxies and Load Balancers for Tableau Server.

Configure Tableau Server to work with a load balancer

The settings used to identify a load balancer to Tableau Server are the same ones that are used to identify a reverse proxy server. If your Tableau Server cluster requires both a proxy server and a load balancer, both must use a single external URL defined in `gateway.public.host` and all proxy servers and load balancers must be specified in `gateway.trusted` and `gateway.trusted_hosts`. See Configuring Proxies and Load Balancers for Tableau Server.

Upgrade Tableau Server Overview

The topics in this section help you upgrade an existing installation of Tableau Server on Linux. They describe the recommended steps of planning and testing *before* actually performing the upgrade. There's information about best practices, and when you're ready to actually perform your upgrade, steps for upgrading a single node server and a multi-node installation. Where possible, we call out possible pitfalls and help you to avoid these.

Note: Use Tableau Release Navigator to search for features in Tableau Desktop, Server, and Prep, or to see a full list of features when comparing your current version of Tableau to a later one.

Looking for Tableau Server on Windows? See Upgrade Tableau Server Overview.

Choose your upgrade path

Blue/Green upgrades

Blue/green upgrades are a special type of upgrade approach that provide minimal downtime, but require a knowledgeable IT team and resources to implement. These are not for every organization, but for those who are equipped to handle them, they can be the right upgrade solution. For more information, see Using a Blue/Green approach for upgrading Tableau Server.

Upgrading in place

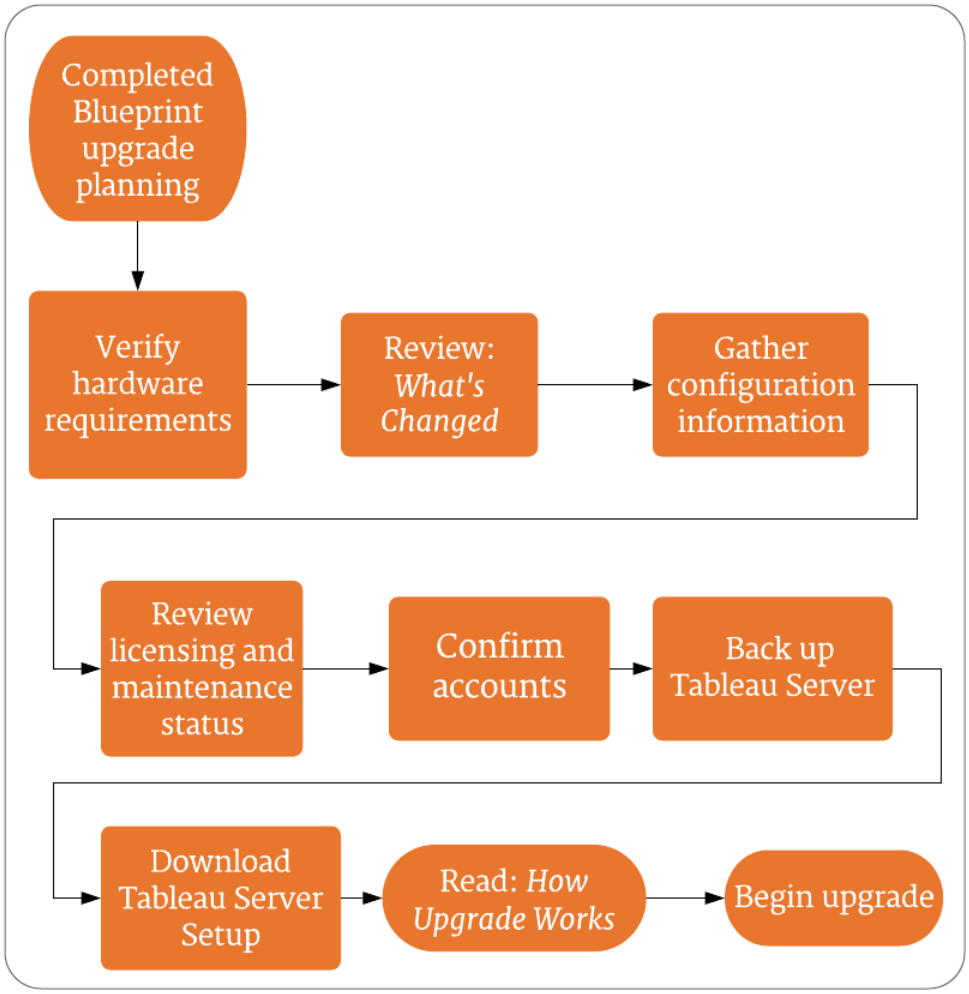
Important: Your Tableau Server on Linux upgrade steps depend on which version you are upgrading from. When you are ready to actually upgrade, be sure you follow the procedure that applies to your installation:

- Upgrading from Tableau Server on Linux version 2018.1 or later is more straightforward. If you are upgrading from version 2018.1.0 or later, follow the steps in Upgrading from 2018.1 and Later (Linux).
- Upgrading from Tableau Server on Linux version 10.5? See Upgrade Tableau Server on Linux from 10.5 in the 2021.4 Server Help.

For instructions on how to determine your version of Tableau Server, see View Server Version.

Preparing for Upgrade

Follow the flow chart and the linked topics to prepare your environment for Tableau Server upgrade.



Release Navigator

Note: Use Tableau Release Navigator to search for features in Tableau Desktop, Server, and Prep, or to see a full list of features when comparing your current version of Tableau

to a later one.

Server Upgrade - Minimum Hardware Recommendations

For production use, the computer on which you upgrade Tableau Server should meet or exceed the minimum hardware recommendations. These recommendations are general. Actual system needs for Tableau Server installations can vary based on many factors, including number of users and the number and size of extracts. If the Setup program determines that your computer does not meet the following recommendations, you will get a warning, but you can continue with the setup process. The minimum recommendations listed here are intended as general guidance. However the recommendations for your environment may vary. For more information, see the Hardware recommendations section of the Recommended Baseline Configurations topic.

Install Type	***Processor***	***CPU***	***RAM***	***Free Disk Space***
Single node	• 64-bit (x86_64 chipsets) • Must support SSE4.2 and POPCNT instruction sets • ARM-based processors are not supported	8 cores (16 vCPUs), 2.0 GHz or higher	Version 2022.3 and later: • 128 GB Version 2021.4.0 to version 2022.1.x: • 64 GB Version 2021.3.x and earlier: • 32 GB	50 GB
	If you are adding Tableau Prep Conductor to your Tableau Server install-			

Install Type	*Processor*	*CPU*	*RAM*	*Free Disk Space*
	ation, we recommend you add a second node and dedicate this to running Tableau Server Prep Conductor. This node should have a minimum of 4 cores (8 vCPUs), and 16 GB of RAM.			
Multi-node and enterprise deployments	Contact Tableau for technical guidance. Nodes must meet or exceed the minimum hardware recommendations, except: • Dedicated Backgrounder nodes running up to two instances of backgrounder, where 4 cores may be acceptable. • Dedicated node for Tableau Prep Conductor: Minimum of 4 cores (8 vCPUs), and 16 GB of RAM. • Dedicated node for Independent Gateway: Minimum of 2 cores (4 vCPUs), 8 GB of RAM, and 100 GB free disk space.			

Important: The disk space requirement cannot be checked until you initialize TSM. If you don't have enough space, you won't be told this until after you install the Tableau Server package.

50 GB disk space available, with a minimum of 15 GB allocated to the `/opt` directory, and the remainder allocated to the `/var` directory for data storage.

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space. You may need to allocate additional disk space depending on various factors like whether you will be using extracts.

  The core Tableau Server bits must be installed in a directory with at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to con-

tinue with setup. By default the install location is the `/opt` directory. You can change the installation path for Tableau Server on RHEL distros.

If you plan to make heavy use of extracts then you may need to allocate additional disk space.You can specify a different directory for data (extract) storage during installation.

- **Network attached storage space requirements for External File Store:** If you are planning to configure Tableau Server with External File Store, you will need to estimate the amount of storage space to dedicate on your network attached storage.

  Estimating the storage size: You must take into account the amount of storage needed for publishing and refreshing extracts. In addition, you must also take into account the repository backup size unless you specifically choose the option to do your repository backup separately as described in the Option 2: Back up repository separately topic.

  - Extracts:
    - Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out how many extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.
    - Consider the space needed by the temp directory during an extract refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.
  - Repository Backup:
    - To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base` directory.
    - To obtain the exact size of the repository data, open the backup file and use the size of the `workgroup.pg_dump` file.

- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the

purposes of counting cores.

- RAM shown is the minimum recommended for a single-node installation. Your installation may function better with more RAM, depending on activity, number of users, and background jobs, for example.

To see the full list of recommendations and to see the minimum requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server. For hardware specifications Tableau uses internally for testing scalability, see Hardware recommendations for production installations.

For hardware recommendations for Tableau Server in the cloud, see the following:

- Selecting an AWS Instance Type and Size in the Tableau Server on Linux in the AWS Cloud Administrator Guide
- Selecting a Google Compute Engine Virtual Machine Type and Size in the Tableau Server on Linux in the Google Cloud Platform Administrator Guide
- Selecting a Microsoft Azure Virtual Machine Type and Size in the Tableau Server on Linux in Microsoft Azure Administrator Guide

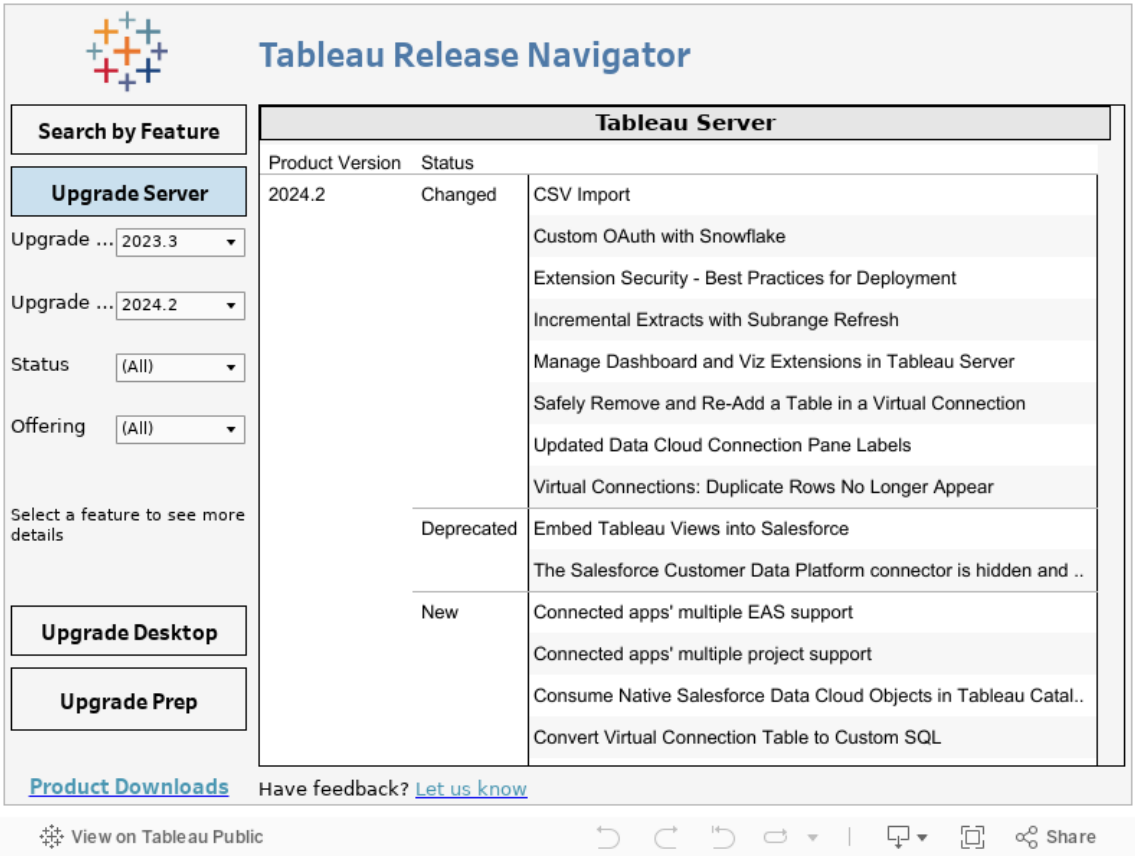
Continue to Server Upgrade - Review What's Changed.

Server Upgrade - Review What's Changed

This topic includes a list of important product changes in Tableau Server, beginning with version 2019.1. The changes listed in this topic may impact the upgrade process itself, or they may impact functionality after you have upgraded. Read these changes carefully and make note of the changes and remediation steps that you'll need to take. Include these remediation steps as part of your upgrade process or post-upgrade configuration plan.

This list is cumulative, so if you are upgrading from an early version (for example, 2019.1), read the list of changes for every version between your version and the version you are upgrading to. You can also download the workbook with the list and use it create your own customized checklist.

To develop a more robust verification and testing plan, we recommend that you also review new features before you upgrade. To see a full list of all new and changed features, select everything listed in the **Status** filter.



Continue to Server Upgrade - Gather Configuration Details.

Server Upgrade - Gather Configuration Details

If you are running an in-place upgrade (you're not updating hardware as part of your upgrade), then nearly all the configuration data is preserved. Strictly-speaking, you do not need to gather all of the configuration information in this case. However, we recommend gathering the information as detailed in this topic. In the worse case scenario, should upgrade fail, then you will

have a record of all configuration information should you need to restore. In any case, you can use the configuration details you collect to verify the upgrade when it's complete.

Take screen shots

A relatively quick way to capture the basics of your configuration is to take screen shots of the TSM web interface pages and the Tableau Server admin area.

Click through all visible pages and take screen shots:

- Sign in to Tableau Services Manager Web UI
- Sign in to the Tableau Server Admin Area

Record object counts

When you are in Tableau Server admin area, count and record the following in each site:

- Projects, workbooks, views, data sources
- Users and groups

Record firewall configuration

If you have configured a local firewall for Tableau Server on Linux, then it's a good idea to copy the configuration for your records.

Our setup documentation describes how to use Firewalld to configure the firewall on single and multi-node deployments of Tableau Server running on RHEL/CentOS distributions. See Configure Local Firewall.

Run the following command to retrieve the firewall configuration:

```
sudo firewall-cmd --list-all
```

Verify TSM Controller certificate expiry

Verify the certificate for the TSM Controller is still valid.

To verify your TSM Controller SSL certificate's expiration date:

1. Open a terminal session.

2. Type the following commands to display the dates when the certificate is valid:

   ```
   openssl s_client -connect <tsm_servername>:8850
   ```

   ```
   echo | openssl s_client -connect <tsm_servername>:8850
   2>/dev/null | openssl x509 -noout -dates
   ```

3. If the certificate is expired, open a case with our Support team, and they can provide guidance.

Gather asset files

Many of the supporting files (certificates, IdP metadata, logos, etc) that you upload to Tableau Server are not accessible with TSM after you upload them. Specifically, files that are uploaded and managed by the Client File Service are renamed and obfuscated before they are distributed across the deployment. This process also parametrizes the file attributes that are required by Tableau services. As a result, files are not mapped to a single file location on the file system for the following files. If you have uploaded any of the following files, be sure to have copies of them saved off of the computer that is running Tableau Server:

- SAML certificate file
- SAML key file
- SAML IdP metadata file
- OpenID.static.file
- Kerberos.keytab file
- LDAP Kerberos keytab file
- LDAP Kerberos conf file
- Mutual SSL certificate file
- Mutual SSL revocation file
- Customization header logo file
- Customization sign-in logo file
- Customization compact logo file

Gather custom configuration information

Some configuration information is not displayed in the TSM or Tableau Server web pages. This section includes configuration details that you may need to gather depending on how

you've customized your Tableau deployment.

Secure SMTP

If you have configured TLS for Tableau Server, then you will need to record the TLS-related configurations, which are not included in the **Email Server** configuration of the TSM Web UI.

To gather the TLS-related configurations, you must run `tsm configuration get` with the following key values:

- `svcmonitor.notification.smtp.ssl_enabled`
- `svcmonitor.notification.smtp.ssl_required`
- `svcmonitor.notification.smtp.ssl_check_server_identity`
- `svcmonitor.notification.smtp.ssl_trust_all_hosts`
- `svcmonitor.notification.smtp.ssl_ciphers`
- `svcmonitor.notification.smtp.ssl_versions`

For example, to retrieve the list of ciphers that are configured for SMTP TLS, run the following command:

```
tsm configuration get -k svcmonitor.notification.smtp.ssl_ciphers
```

You can find more information about each of the keys above in the TSM CLI section of Configure SMTP Setup.

Analytics extensions

If you have configured analytics extensions (formerly referred to as "external services), you will need to record your configuration information. Upgrading to Tableau Server 2020.2 or later will remove all configuration for this feature.

To retrieve the analytics extensions configuration from Tableau Server versions 2019.1 through 2020.1, run the following command:

```
tsm security vizql-extsvc-ssl list
```

To retrieve the password that is stored for the analytics extensions connection (if any), run the following tsm command:

```
tsm configuration get -k vizqlserver.rserve.password
```

You should also have a copy of the certificate for analytics extensions if you've configured SSL.

External Repository

This applies only if you are using the External Repository configuration with Tableau Server. If you are not sure if this applies to you, see Tableau Server External Repository.

If you are using an External Repository, you may need to take additional steps when upgrading.

- **No version change**—If there is no version change in PostgreSQL, there are no special actions required.
- **Minor version change**—If there is a minor version change in PostgreSQL, you need to upgrade your external repository before upgrading Tableau Server. In most cases there are in-place methods for doing so. The method you use depends on the location of your repository and is beyond the scope of this documentation.

- **Major version change**—If there is a major version change in PostgreSQL, you need to follow the steps described in Upgrade Tableau Server with External Repository for a New Major Version of PostgreSQL.

  Steps include:

  1. Creating a new instance of PostgreSQL DB. For more information, see:
     - Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS)
     - Create a Azure Database PostgreSQL Instance on Azure
     - Create a PostgreSQL Instance on Google Cloud
     - Create a PostgreSQL Database as a Stand-alone Installation
  2. Creating a configuration file and download the SSl certificate file for the new instance that you created in Step 1.

  During upgrade, you will need to point Tableau Server to the new instance using the configuration file. The upgrade process will migrate the content from your current external repository to your new instance. For more details, see Upgrade Tableau Server with External Repository for a New Major Version of PostgreSQL.

External File Store

There are no special steps, actions, or configuration required when upgrading Tableau Server configured with External File Store. You can upgrade Tableau Server using the normal procedure.

Port customization

If you have changed the dynamic port range or have configured ports manually for Tableau Server, record the changes you have made.

Run the following command:

```
tsm topology list-ports
```

For more information see Tableau Services Manager Ports.

Continue to .Server Upgrade - Verify Licensing Status

Server Upgrade - Verify Licensing Status

Prior to upgrading Tableau Server, review the product keys that are currently installed and compare them to the product keys, maintenance expiration, and license expiration end dates that are listed in the Tableau Customer Portal.

It is important that all expiration and end dates associate with product keys are up-to-date before you upgrade.

Step 1: View license expiration information.

Run the following command to view all licenses that are installed on your Tableau Server installation:

```
tsm licenses list
```

The tsm licenses list command returns all licenses, each with a number of fields. For upgrade, you need to make note of two fields:

- For each subscription license, make note of the date in the LIC EXP field.
  LIC EXP displays the date that the license expires and Tableau Server will stop working.
- For each legacy perpetual key, make note of the MAINT EXP field. MAINT EXP displays the date that the maintenance contract for the Tableau Server deployment expires.

Step 2: Compare license expiration dates from TSM to your licenses in Tableau Customer Portal**.**

Step 3: If required, update licenses.

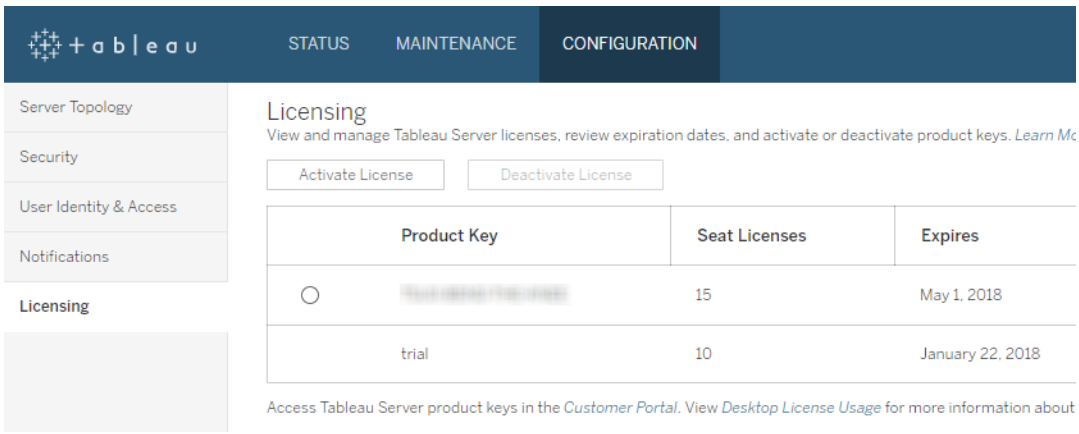
If your TSM maintenance date is not current or the expiration date will occur soon or is not the same as the date listed in the Tableau Customer Portal, refresh the license.

To refresh the license:

1. Open TSM in a browser:

   http://<tsm-computer-name>:8850

2. Click **Configuration** and **Licensing** and click **Refresh All**:



If this does not update the maintenance or expiration end date but the Tableau Customer Portal does show more current end dates, please contact Tableau Technical Support.

Reactivating the product key will be part of the upgrade process.

Continue to Server Upgrade - Verify Accounts.

Server Upgrade - Verify Accounts

Before you upgrade Tableau Server, verify that the account you will use for running Tableau Server Setup has `sudo` access.

If your organization uses an external identity store (LDAP or Active Directory), then it's a best practice to have the credentials or the keytab file for the account that is used to bind with LDAP.

Continue to Server Upgrade - Back Up Tableau Server.

Server Upgrade - Back Up Tableau Server

We *strongly* recommend that you make a backup of your installation of Tableau Server before beginning the upgrade process. A backup made before you start the upgrade provides data that you'll need to set up a test version of the upgraded environment, and it also gives you the ability recover if the upgrade process fails. Backups are not an include step during upgrades, except when the upgrade includes a minor version upgrade to PostGRES and then a pg-only upgrade is created for internal use during upgrade.

Notes:

- We recommend you disable subscriptions and scheduling in your production environment immediately before taking the backup, and reenable them after the backup is complete. Doing this will help avoid having your users receive duplicate subscriptions and email messages when you restore your backup in your test environment.
- The full backup can take a while if you have a large installation or a lot of extracts.
- Any changes made between the time you took the backup and the time you do the upgrade are lost because they aren't included in the backup.
- Remove Unneeded Files.

To back up server configuration data, use the `tsm settings` command.

When you use the `tsm maintenance backup` command, the current date is appended to the backup file:

```
tsm maintenance backup -f <backup_file> -d
```

For more information, see tsm maintenance backup.

Continue to Server Upgrade - Download Setup.

Server Upgrade - Download Setup

You must download and copy the Tableau Server Setup program to the computer running Tableau Server. If you are running a multi-node deployment of Tableau Server, copy the Setup program to each node in the cluster.

To get the latest version of Tableau Server software, go to the Customer Portal. When you purchase Tableau, you get a user name and password to sign in to the Customer Portal.

> **Important**: Always download and install the latest maintenance release for the version you are upgrading to.

Continue to How Tableau Server Upgrade Works.

How Tableau Server Upgrade Works

When upgrading from Tableau Server 2018.2 or later, you do not uninstall the previous version. The upgrade process is designed to install a new version side-by-side with your existing version, and then switch from the earlier version to the later one.

When doing a TSM-to-TSM upgrade (version 2018.2.x to later version), you must leave your existing version in place and running until just before you upgrade using the upgrade script. If you uninstall your existing version before upgrading, you will not be able to upgrade. For your end-users, this has the impact of reducing downtime because you install your new version

while the existing version is running. The only time the server is not running is during the period the upgrade script is actually upgrading to the new version.

After installing the new version and upgrading, you can uninstall the previous version when you choose to do so. You can leave the older version installed indefinitely, though one reason you might choose to uninstall it is to free up disk space used by files specific to the older version. To understand the difference between uninstalling Tableau Server and completely removing all aspects of Tableau Server, see Uninstall Tableau Server.

Upgrading Tableau Server requires a stop and start the server as part of the upgrade process. During this stop/restart Tableau Server is unavailable.

What's Changed - Things to Know Before You Upgrade

Beginning with version 2020.2, this topic has been merged with What's New in Tableau Server, to provide a single location for new and changed features and behavior in Tableau Server. See Tableau Server Release Notes for both new features and for changes introduced in this and earlier versions.

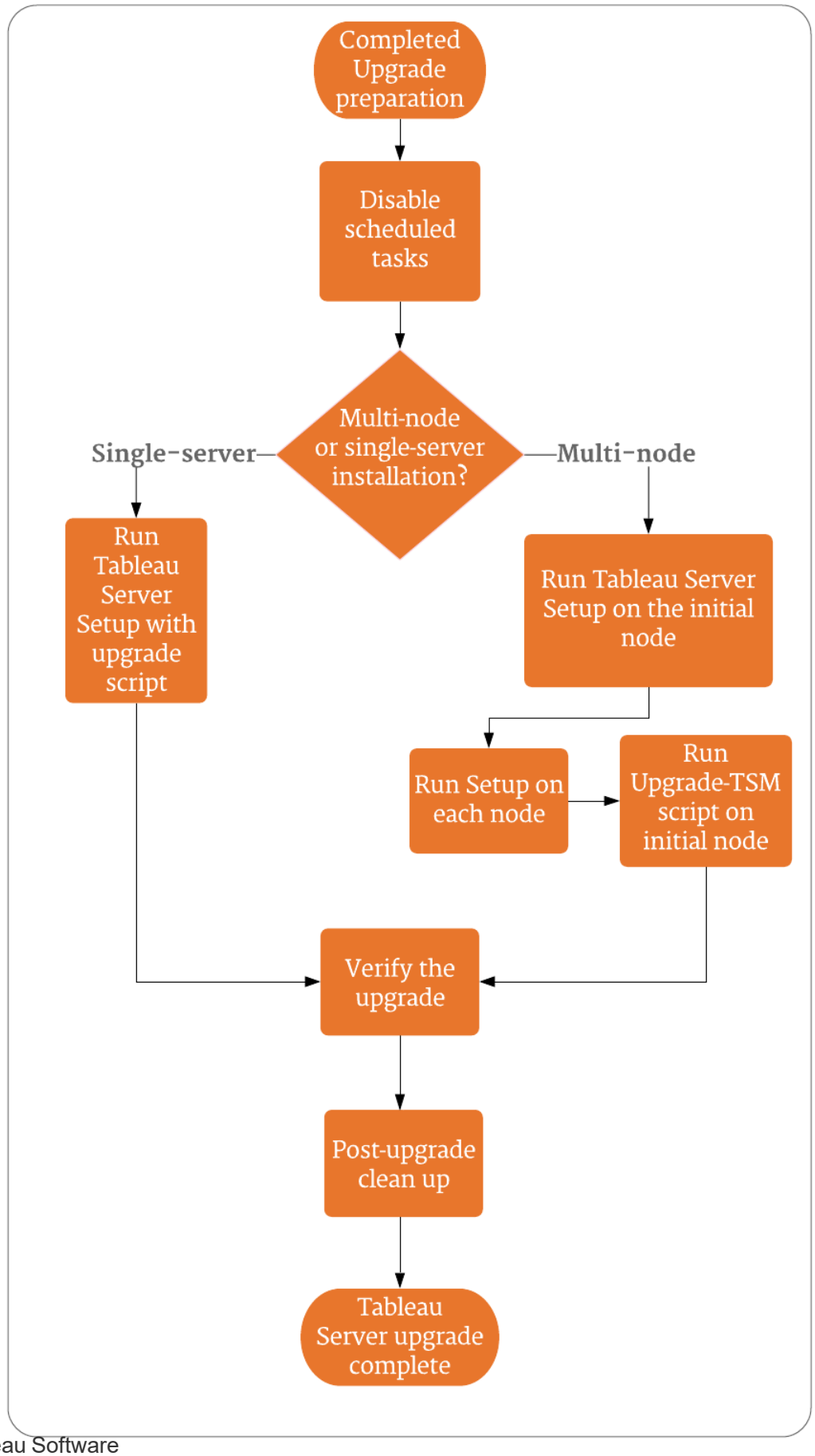
> **Note:** Use Tableau Release Navigator to search for features in Tableau Desktop, Server, and Prep, or to see a full list of features when comparing your current version of Tableau to a later one.

Upgrading from 2018.1 and Later (Linux)

Follow the flow chart below and the linked topics to perform an in-place upgrade of Tableau Server.

Before you start the upgrade, make sure you have completed the Upgrade Preparation, For more details, see Preparing for Upgrade.

If you are migrating Tableau Server to new hardware as part of your upgrade, see Migrate to New Hardware.


Completed Upgrade preparation
Disable scheduled tasks
Multi-node or single-server installation?
Single-server
Multi-node
Run Tableau Server Setup with upgrade script
Run Tableau Server Setup on the initial node
Run Setup on each node
Run Upgrade-TSM script on initial node
Verify the upgrade
Post-upgrade clean up
Tableau Server upgrade complete

Support and services to help with Tableau Server upgrades

Tableau Server upgrades require planning and testing. Upgrades are generally done during times when users are not on Tableau Server. If you plan an upgrade over a weekend and anticipate needing Tableau Technical Support, see the Technical Support Programs information on the Tableau web site. Availability depends on the level of support you have. If you would like Tableau to assist with the planning and upgrading of your Tableau Server installation, see our Tableau Global Services Server Upgrade page for details.

Server Upgrade - Disable Scheduled Tasks

Important: We strongly recommend disabling scheduled tasks before you perform an upgrade. This includes all updates to data content and should be done before you create your pre-upgrade backup. This may involve disabling jobs that are triggered outside of Tableau Server, such as those initiated through REST API-based extract refreshes or using tabcmd.

Run the following procedure to disable all scheduled extract refreshes, flows, and subscriptions.

1. Sign in to the Tableau Server Admin Area as the Tableau Server administrator.
2. Go to the server-wide Schedules page:
    - On a Tableau Server where only a single (Default) site exists, click **Schedules** in the left pane.
    - On a Tableau Server with more than one site, click **Manage all sites** in the drop-down menu on the top of the left pane, and then click **Schedules**.
3. On the Schedules page, click **Select All**.
4. On the **Actions** menu, select **Disable**, and then click **Disable** in the resulting prompt.

Continue to: Single-Server Upgrade -- Run Setup, or Multi-node Upgrade -- Run Setup

Single-Server Upgrade -- Run Setup

Run Setup

Follow these steps to upgrade a single-node installation of Tableau Server version 2018.2 or later.

1. Log on as a user with sudo access to the computer you are upgrading.
2. Navigate to the directory where you copied the `.rpm` or `.deb` Tableau Server package.
3. Use the package manager to install the Tableau Server package.

   You must install the new version to the same location as the existing version. The install location must be the same on all nodes. Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

   - On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

     - **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

       ```
       sudo yum update

       sudo yum install tableau-server-<version>.x86_64.rpm
       ```

     - **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

       Run the following command:

       ```
       sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
       ```

> **Note:** When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. Run the upgrade script.

   If Tableau Server is not stopped when you run the script, the script will let you know, and will offer to stop the server. You can also choose to stop the server before running the script using the `tsm stop` command. Tableau Server must be stopped to complete the upgrade.

   The options you need to include depend on the version you are upgrading to:

   - Version 2019.3 or later:

     ```
     sudo /opt/tableau/tableau_server-
     /packages/scripts.<version_code>/upgrade-tsm --accepteula
     ```

     where `<version_code>` is the long form of new version you are upgrading to, for example `scripts.20183.18.1128.2033`.

Starting with version 2019.3.0, when you upgrade from 2019.2.x or later, the script runs using the account you are logged in with. If you are prompted, enter your password. For more information, see What's Changed - Things to Know Before You Upgrade. You can specify a different user with administrative permissions using the `-u` option and specifying a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

- Version 2018.1 through version 2019.2.x:

  ```
  sudo /opt/tableau/tableau_server/packages/scripts.<version_code>/upgrade-tsm -u <system_admin> --accepteula
  ```

  where `<version_code>` is the long form of new version you are upgrading to, for example `scripts.20183.18.1128.2033`, and `<system_admin>` is a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

  The `-u` option was added as of 2018.1. For more information, see What's Changed - Things to Know Before You Upgrade.

To see all the options available for the `upgrade-tsm` script, use the `-h` option. For example:

```
sudo /opt/tableau/tableau_server/packages/scripts.<version_code>/upgrade-tsm -h
```

5. After the upgrade is completed, exit the terminal session and log in again. This ensures that your session will be using the updated TSM version.

6. Start Tableau Server:

   ```
   tsm start
   ```

Continue to Verify Tableau Server Upgrade.

Multi-node Upgrade -- Run Setup

Run Setup

1. Log on as a user with sudo access to the initial node in the cluster.
2. Navigate to the directory where you copied the `.rpm` or `.deb`Tableau Server package.
3. Use the package manager to install the Tableau Server package.

   You must install the new version to the same location as the existing version. The install location must be the same on all nodes. Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume.

   - On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

     - **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

       ```
       sudo yum update

       sudo yum install tableau-server-<version>.x86_64.rpm
       ```

     - **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

       Run the following command:

       ```
       sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
       ```

   Note: When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location,

> or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- On Ubuntu, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

Continue to Multi-node Upgrade -- Run Setup on Each Node.

Multi-node Upgrade -- Run Setup on Each Node

Run Setup

On each additional node, navigate to the folder where you copied the Tableau Server Setup program and run the Setup program as you did on the initial node. This will install the new version of Tableau Server on each node, side-by-side with your existing, running version.

Continue to Multi-node Upgrade -- Run Upgrade Script.

Multi-node Upgrade -- Run Upgrade Script

Run Upgrade script

1. After you have installed the new version on *every node in the cluster*, run the upgrade script on the initial node.

   If Tableau Server is not stopped when you run the script, the script will let you know, and will offer to stop the server. You can also choose to stop the server before running the

script using the `tsm stop` command. Tableau Server must be stopped to complete the upgrade.

The options you need to include depend on the version you are upgrading to:

- Version 2019.3 or later:

  ```
  sudo /opt/tableau/tableau_server-
  /packages/scripts.<version_code>/upgrade-tsm --accepteula
  ```

  where `<version_code>` is the long form of new version you are upgrading to, for example `scripts.20183.18.1128.2033`.

  Starting with version 2019.3.0, when you upgrade from 2019.2.x or later, the script runs using the account you are logged in with. If you are prompted, enter your password. For more information, see What's Changed - Things to Know Before You Upgrade. You can specify a different user with administrative permissions using the `-u` option and specifying a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

- Version 2018.1 through version 2019.2.x:

  ```
  sudo /opt/tableau/tableau_server-
  /packages/scripts.<version_code>/upgrade-tsm -u <system_
  admin> --accepteula
  ```

  where `<version_code>` is the long form of new version you are upgrading to, for example `scripts.20183.18.1128.2033`, and <`system_admin`> is a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

  The `-u` option was added as of 2018.1. For more information, see What's Changed - Things to Know Before You Upgrade.

To see all the options available for the `upgrade-tsm` script, use the `-h` option. For example:

```
sudo /opt/tableau/tableau_server/packages/scripts.<version_
code>/upgrade-tsm -h
```

2. After the upgrade is completed, exit the terminal session and log in again. This ensures that your session will be using the updated TSM version.

3. Start Tableau Server:

   ```
   tsm start
   ```

Continue to Verify Tableau Server Upgrade.

Verify Tableau Server Upgrade

Work through the sections of this topic to verify if server upgrade was successful.

Verify Tableau Service processes

Start your verification process by signing into TSM with the command line and verifying the status of Tableau Server processes.

1. Open a command prompt with an account that is a member of the `tsmadmin` group.

   For more information on using the tsm command line, see Using the tsm CLI.

2. Run the following command to view the status of all the Tableau Server processes: `tsm status -v`.

What if I get an access denied error when I attempt to run TSM commands?

The account that you use to configure the rest of the installation must be a member of the `tsmadmin` group that was created during initialization. To view the user accounts in the `tsmadmin` group, run the following command:

```
grep tsmadmin /etc/group
```

If the user account is not in the group, run the following command to add the user to the `tsmadmin` group:

```
sudo usermod -G tsmadmin -a <username>
```

Verify TSM global settings

As part of the preparation for upgrade, you should have taken screen shots of the TSM web UI settings. Sign in to Tableau Services Manager Web UI and compare the settings with the screen shots.

Enable subscriptions and scheduling

1. Sign in to the Tableau Server Admin Area as the Tableau Server administrator.
2. Go to the server-wide Schedules page:
    - On a Tableau Server where only a single (Default) site exists, click **Schedules** in the left pane.
    - On a Tableau Server with more than one site, click **Manage all sites** in the drop-down menu on the top of the left pane, and then click **Schedules**.
3. On the Schedules page, click **Select All**.
4. On the **Actions** menu, select **Enable**, and then click **Enable** in the resulting prompt.

Verify user access

With a user account, sign in to Tableau Server with Tableau Desktop and a browser.

If Tableau Server is available from outside your organization, verify connectivity from the internet.

If your organization supports mobile connectivity to Tableau Server, sign in to server with a mobile device.

View published workbooks

With a user account, sign in to Tableau Server and browse published workbooks. Open workbooks to view them in a browser

Verify publishing workbooks and data sources

Using an account with a Creator license, verify web authoring functionality to connect to data, and to create and publish workbooks and data sources:

- Connect to Data on the Web
- Publish a Data Source on the Web
- Upload Workbooks to a Tableau Site

Verify Tableau Prep Builder

If your organization is running Tableau Prep Builder, verify that you can connect to data, open a flow, and other Prep tasks. See Tableau Prep: Connect to Data.

Verify count of Tableau objects

As part of upgrade preparation, you should have noted how many projects, workbooks, views, data sources, users, and groups are in your deployment. Verify that all objects still appear.

Sign in to the Tableau Server Admin Area as the Tableau Server administrator.

- Verify count of projects, workbooks, views, data sources
- Verify count of users and groups

Verify API functionality

If your organization has deployed or developed content with Tableau APIs, verify that these are operational.

REST API

Verify REST API access and authentication by running the sample cURL from Get Started Tutorial: Introduction and Set Up:

```
curl "https://MY-SERVER/api/3.8/auth/signin" -X POST -d @signin.xml
```

Content of signin.xml:

```
<tsRequest>
  <personalAccessTokenName="MY_PAT_NAME"
```

```
      <credentials name="username" password="password" >
       <site contentUrl="MarketingSite" />
  </credentials>
</tsRequest>
```

Example response:

```
<tsResponse version-and-namespace-settings>
  <credentials token="12ab34cd56ef78ab90cd12ef34ab56cd">
       <site id="9a8b7c6d-5e4f-3a2b-1c0d-9e8f7a6b5c4d"
       contentUrl="MarketingSite"/>
  </credentials>
</tsResponse>
```

Compatibility testing

For other developed features, perform compatibility testing to verify that the upgraded version of Tableau Server operates as expected with your existing solutions:

- Dashboard Extensions
- JavaScript API for embedded Tableau
- Connectors (Web Data Connector, Tacos)

Go to Post Upgrade Cleanup.

Post Upgrade Cleanup

As discussed in How Tableau Server Upgrade Works, the Tableau Server upgrade process installs a new version side-by-side with your existing version. Now that upgrade is complete and verified, you can remove the older version of Tableau Server to free up disk space. This is an optional step.

Uninstall previous version

Use this procedure to free up disk space by uninstalling packages for previous Tableau Server versions after you have upgraded to a newer version of Tableau Server.

1. Look at the environment.bash file to confirm which version of Tableau Server is currently in use. At a command prompt, type:

   ```
   grep TABLEAU_SERVER_DATA_DIR_VERSION /etc/opt/tableau/tableau_server/environment.bash
   ```

2. Determine which versions of the Tableau Server package are installed on your computer.

   - On RHEL-like distributions, including CentOS, run the following command:

     ```
     yum list installed tableau-server"*"
     ```

   - On Ubuntu, run the following command:

     ```
     apt list --installed tableau-server"*"
     ```

3. Remove the Tableau Server package with your package manager.

   - On RHEL-like distributions, including CentOS, run the following command:

     ```
     sudo yum remove tableau-server-<version>.x86_64
     ```

   - On Ubuntu, run the following commands:

     ```
     sudo apt-get purge tableau-server-<version>
     ```

Using a Blue/Green approach for upgrading Tableau Server

While many organizations are well-served by performing an in-place upgrade of Tableau Server, other organizations with mission-critical use of Tableau may want a more robust, enterprise-grade upgrade approach which uses investments of additional effort and resources to reduce risk and impact. In these cases Tableau recommends a “Blue/Green” upgrade approach to achieve that outcome, characterized by:

- Reliable deployment of applications into a freshly-configured host non-production environment (versus an in-place upgrade)
- Performing pre-production testing of the new version in your non-production environment to validate business-critical functionality, including testing common use cases to detect changes in behavior that may impact the organization

- Testing in production-identical environments to validate integrations and resource requirements
- The ability to roll back to the previous version rapidly

This high-level guidance describes a Blue/Green approach to upgrades that has been successfully used by many of our biggest customers to upgrade Tableau Server (and other mission-critical applications) with confidence. Blue/Green upgrades are a long-established industry practice; there are many variations and the specific steps below are one possible path. If you're seeking specific guidance on customizing this approach for your organization's next upgrade, talk to your Premium Support Technical Account Manager, or discuss a services engagement with your account executive or delivery partner.

The Blue/Green approach involves installing a new version of Tableau Server in a non-production environment (your "Green" installation) that mirrors the configuration of your existing production environment (your "Blue" installation). For purposes of your Tableau license, note that Tableau grants a standard right to two (2) non-production environments to support the single production environment in your deployment, as further described in the governing license agreement. These non-production environments should be able to be taken down and used for this method consistent with the intended use of a test environment as described in this Tableau Knowledge Base article: Licensing a Tableau Server Test Environment.

By copying your content from the production environment to the Green installation in your non-production environment (for example, your test environment), you can create an upgraded instance of Tableau Server with minimum downtime and the safety net of having the original production installation to fall back to if necessary. Once you determine all your critical content in the Green environment is functional, switch your users over to the new environment (Green). Your Green environment becomes your production environment, and your original installation (Blue environment) can be maintained as a non-production environment to support your next upgrade.

What follows is a high-level outline of the steps necessary to use Blue/Green as an approach to upgrade Tableau Server. These assume you have a load balancer or DNS that allows you to redirect user traffic from one installation to the other, and that you have downloaded the new version of Tableau you want to upgrade to.

Important: Your Blue/Green installations should be treated as production environments. They should not serve as disaster recovery or general testing environments. For more details on Disaster Recovery please see Disaster Recovery in the Tableau Blueprint.

1. Create a second installation of servers that mirrors your production installation of Tableau Server as closely as possible. This second environment is your Green installation, a non-production environment. It should have a similar capacity, resources, and configuration as your production environment ("Blue") because your Green instance will become your production instance after the upgrade.
2. Create an initial backup and settings export of your Tableau Server Blue installation. You'll use these for initial user acceptance testing (UAT) in Green. Later, you'll create final, up-to-date backup and export to use when actually switching from Blue to Green.
    a. Backup: For details on creating backups, see Back up Tableau Server Data.
    b. Settings export: Along with a backup, you'll also need to export the settings from the Blue installation. Some settings will need to be manually recreated in the Green environment. For more information, see Backup assets that require a manual process
3. Install the new version of Tableau Server in Green. You should specify the same Identity Store type and Run As service account as you use in Blue.
4. Use the initial Blue backup and settings export to update the Green installation.
    a. Restore the backup from the Blue environment in the Green installation.
    b. Import the settings you exported from the Blue installation. You may have to manually recreate some of the settings in your Green environment. See Restoring core Tableau Server functionality.

5. Disable any scheduled tasks in Green to avoid sending duplicate messages while testing, and to avoid overloading database resources. For details, see Server Upgrade - Disable Scheduled Tasks.

6. Test the Green installation to confirm it is functioning as expected. Perform User Acceptance Testing (UAT), and simulate any load testing if desired. If you have scripts or API integration with Tableau, you should test these as well. The more complete your testing, the better prepared you will be to address any discrepancies between your original Blue installation and your new Green installation.

7. Prepare the Blue installation so you can create a backup and export settings you'll use for production in Green. This might mean restricting access to Tableau, or sending a message instructing users not to make any changes to their content and warning them that any changes made after you create your backup will be lost.

8. Create a final backup of Blue. This is the backup you will use to bring Green up to the most recent production content. If you do regular backups, you can use your latest production backup, keeping in mind that any changes or updates done in Blue after the backup is created will be lost. We recommend a "change freeze" on your production environment after this backup is taken, so no new workbooks or data sources are added, and there are no changes to existing content.

9. Export the settings in the Blue environment, paying attention to those assets you may need to handle manually. For more information, see Restoring core Tableau Server functionality.

10. Restore the final Blue backup and import the Blue settings file to Green. Make any manual configuration changes necessary.

11. Perform UAT/sanity checks of Green content.

12. Reenable schedules for any jobs, subscriptions, and notifications you disabled in Step 5 above.

13. Switch your users to Green using a load balancer or DNS, and block access to Blue.

14. Verify that the Green server is functional as desired and monitor for production load issues. If you have Advanced Management, you can use the Resource Monitoring Tool to monitor performance.

15. (Optional) If you're not keeping your Blue environment for future upgrades, deactivate the Blue environment product keys:

    - For Tableau Server 2021.4 and later activated with ATR, see Deactivate Product Key.
    - For any version of Tableau Server not activated with ATR, see Deactivate Product Key.
    - For Tableau Server 2021.3 and earlier activated with ATR, see Move a Server ATR License to Another Tableau Server.

16. (Optional) If you're not keeping your Blue environment for future upgrades, stop the Blue environment and prepare it for decommissioning. For details about decommissioning, see Remove Tableau Server from Your Computer.

Upgrade Tableau Server on Linux from 10.5

Important: Beginning with version 2020.4.0, if you are running version 10.5 of Tableau Server on Linux, you cannot upgrade directly to the latest version. You must upgrade to a version between 2018.1 and 2020.3 before upgrading to 2020.4 or later. Support for version 10.5 ended in July 2020 so direct upgrades to version 2020.4 or later are not supported. For information about supported versions, see the Tableau web site.

When you upgrade Tableau Server on Linux from version 10.5, you need to take several unique steps to complete the upgrade. These are necessary because of a change made after version 10.5.0 released, related to sudo privileges. For more information, see System User, sudo Privileges, and systemd. You only need to do these extra steps once, during the upgrade to 2018.1 or later. This topic describes how to upgrade from version 10.5.0 or 10.5.x (10.5.1 or later) to version 2018.1 or later.

If you attempt to upgrade from 10.5.0 or 10.5.x without following these instructions, warnings are displayed and the upgrade is canceled. You will not break your existing Tableau Server installation, but you cannot continue the upgrade.

To identify the version of your installation, see View Server Version.

Follow these steps to upgrade from 10.5.0:

1. Upgrade to 10.5.x—If you are running version 10.5.0, you must first upgrade to 10.5.x (10.5.1 or higher) by installing 10.5.x and running the `upgrade-tsm` script in the 10.5.x scripts directory on your initial node.
2. Install 2018.x or later, up to 2020.3.x—With 10.5.x installed and running as expected, install 2018.x or later, but do not upgrade to this version yet.
3. Run TSM commands—Use TSM to stop the server and run three additional commands.
4. Migrate 10.5.x to single user—Run the migration script in the new version (2018.x or later) scripts directory. Do this on every node in your cluster.
5. Upgrade to 2018.x or later, up to 2020.3.x— Upgrade Tableau Server by running the `upgrade-tsm` script from the new version scripts directory on your initial node.
6. Upgrading from 2018.2 and Later— After upgrading to a version between 2018.x and 2020.3.x, you can upgrade Tableau Server to 2020.4 or later by following the instructions for here: Upgrading from 2018.1 and Later (Linux).

Upgrade to 10.5.x

If you are running version 10.5.0 of Tableau Server on Linux, the first step you need to take is to upgrade to a later version of 10.5. Beginning with version 10.5.1 changes were made that are needed in order to upgrade to 2018.1 or later. (If you are already on a version of 10.5 that is higher than 10.5.0, you can skip to the Install 2018.x or later, up to 2020.3.x step.)

To upgrade from 10.5.0 to a later version of 10.5:

1. On each node in your cluster:

   a. Copy the Tableau Server version 10.5.x .rpm or .deb package to location accessible from the computer you are upgrading.

      If you are upgrading a distributed deployment of Tableau Server, copy the .rpm or .deb package to each node in the cluster or to a location accessible from each node.

   b. Log on as a user with sudo access to the computer you are upgrading.

   c. Navigate to the directory where you copied the `.rpm` or `.deb`Tableau Server package.

   d. Use the package manager to install the Tableau Server package.

      - On RHEL-like distributions, including CentOS, run the following command:

        ```
        sudo yum install tableau-server-<version>.x86_64.rpm
        ```

      - On Ubuntu, run the following commands:

        ```
        sudo gdebi -n tableau-server-<version>_amd64.deb
        ```

2. Stop Tableau Server. If you are upgrading a cluster, do this after you have installed the new package on every node in your cluster.

   ```
   tsm stop
   ```

3. With Tableau Server stopped, run the following command on your initial node. Do not run this command on any additional nodes:

   ```
   sudo /opt/tableau/tableau_server/packages/scripts.<version_
   code>/upgrade-tsm --accepteula
   ```

   where `<version>` is the 10.5.x version you are upgrading to.

To see all the options available for the `upgrade-tsm` script, use the `-h` option. For example:

```
upgrade-tsm -h
```

4. After the upgrade is completed, ensure your session is using the updated TSM version by doing one of the following:

   - Use the source command:

     ```
     source /etc/profile.d/tableau_server.sh
     ```

   - Exit the terminal session on the initial node and log in again.

5. Start Tableau Server:

   ```
   tsm start
   ```

Install 2018.x or later, up to 2020.3.x

Install the new Tableau Server package but do not upgrade to this version yet. Before you do so, you need to run several commands and a migration script. You can install the new version package without stopping the server. When you install the new package you are copying the software to your computer but not changing anything about the currently running version.

To install the new version package, on each node in your cluster:

1. Copy the Tableau Server .rpm or .deb package to location accessible from the computer you are upgrading.

   If you are upgrading a distributed deployment of Tableau Server, then copy the .rpm or .deb package to each node in the cluster or to a location accessible from each node.

2. Log on as a user with sudo access to the computer you are upgrading.

3. Navigate to the directory where you copied the `.rpm` or `.deb` Tableau Server package.

4. Use the package manager to install the Tableau Server package.

- On RHEL-like distributions, including CentOS, run the following command:

  ```
  sudo yum install tableau-server-<version>.x86_64.rpm
  ```

- On Ubuntu, run the following commands:

  ```
  sudo gdebi -n tableau-server-<version>_amd64.deb
  ```

Run TSM commands

Using version 10.5.x of Tableau Server that is installed and running:

1. Stop the server:

   ```
   tsm stop
   ```

2. Run these three commands:

   ```
   tsm configuration set -k service.linux.privileged_user -v
   'tableau'

   tsm configuration set -k install.username -v 'tableau'

   tsm pending-changes apply
   ```

   where '`tableau`' is the user name you specified with the `initialize-tsm --unprivileged-user` option when you first installed 10.5.x. If you did not specify a user, the default is '`tableau`'.

Migrate 10.5.x to single user

Run this script from the 2018.x or later scripts directory:

```
sudo /opt/tableau/tableau_server/packages/scripts.<version_
code>/migrate-to-single-user
```

where <version_code> is the long form of your new version number.

Important: If you have a multi-node installation, you must run this script on every node in your cluster.

At this point Tableau Server is running 10.5.x but configured to work with a single user. This is an interim stage. You should complete the upgrade to version 2018.x or later before using Tableau.

Upgrade to 2018.x or later, up to 2020.3.x

After completing the above steps:

1. With Tableau Server stopped, run the upgrade script on the initial node. Do not run the script on any additional nodes. The options you need to include depend on the version you are upgrading to:

    - Version 2019.3 or later:

      ```
      sudo /opt/tableau/tableau_server-
      /packages/scripts.<version_code>/upgrade-tsm --accepteula
      ```

      where `<version_code>` is the long form of new version you are upgrading to, for example `scripts.20183.18.1128.2033`.

      Starting with version 2019.3.0, when you upgrade from 2019.2.x or later, the script runs using the account you are logged in with. If you are prompted, enter your password. For more information, see What's Changed - Things to Know Before You Upgrade. You can specify a different user with administrative permissions using the `-u` option and specifying a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

    - Version 2018.1 through version 2019.2.x:

```
sudo /opt/tableau/tableau_server/packages/scripts.<version_
code>/upgrade-tsm -u <system_admin> --accepteula
```

where `<version_code>` is the long form of new version you are upgrading to, for example `scripts.20183.18.1128.2033`, and <`system_admin`> is a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

The `-u` option was added as of 2018.1. For more information, see What's Changed - Things to Know Before You Upgrade.

To see all the options available for the `upgrade-tsm` script, use the `-h` option. For example:

```
sudo /opt/tableau/tableau_server/packages/scripts.<version_
code>/upgrade-tsm -h
```

2. After the upgrade is completed, ensure your session is using the updated TSM version by doing one of the following:

   - Use the source command:

     ```
     source /etc/profile.d/tableau_server.sh
     ```

   - Exit the terminal session on the initial node and log in again.

3. Start Tableau Server:

   ```
   tsm start
   ```

When desired, you can remove Tableau Server on Linux version 10.5 from your server. Unlike most other programs that run on Linux, previous Tableau Server versions are not automatically removed as part of a successful upgrade. To learn more, see Remove Tableau Server from Your Computer.

Related topics

- Common Tableau Server Upgrade Issues

Test the Upgrade

The best way to learn what impact a Tableau Server upgrade will have to your current environment is to test it. Knowing how an upgrade will affect your users and your server helps you plan and communicate before the actual upgrade, ensuring that your users will not be caught by surprise.

If you have a Tableau Server test environment this is a great place to test out the upgrade.

We recommend the following sequence for testing a Tableau Server upgrade:

1. Prepare a test environment
2. Upgrade the test environment
3. Confirm that existing functionality works
4. Performance and user acceptance testing
5. Test new features
6. Communicate about the upgrade

Prepare a test environment

To start, create a test environment that mirrors your production environment as closely as possible. The closer your test environment is to the actual environment you will be upgrading, the more accurate a representation you will have of how the upgrade will impact you. This includes identical or similar hardware and operating systems, as well as the same authentication options and network access.

When you've got a test computer or virtual machine ready, follow these steps for creating a test environment.

1. On the existing production environment, create a backup of Tableau Server using the `tsm maintenance backup` command.

For more information, see Create a pre-upgrade backup.

2. On your test environment, install a copy of the same version of Tableau Server as you have in your production environment.

   Note: You can download the setup program for your current version from the Alternate Downloads Site.

3. Restore your existing database data using the `tsm maintenance restore` command.

   For more information, see Restore from a Backup.

4. Manually replicate your existing Tableau Server configuration.

   You need to manually configure certain aspects of your environment because when you restore the Tableau database it doesn't include configuration details and customizations.

Upgrade the test environment

Follow the appropriate steps for upgrading the test environment, based on your environment:

- Upgrading from 2018.1 and Later (Linux)
- Upgrade Tableau Server on Linux from 10.5

Confirm that everything works as expected

After you have the new version of Tableau Server installed and configured in your test environment, you are ready to test. You should test basic functionality, along with any special aspects of server that your organization relies on. For example, if there are key subscriptions that your organization relies on, make sure that you test those.

These are some areas of testing to consider:

- **Server processes**. Sign in to Tableau Server as a server administrator, and then open the Server Status page to confirm that all services and processes are running as expected (including on all additional nodes if this is a distributed installation).
- **User access**. Confirm that Tableau Server users can sign in. Test your normal user sign in process. Have some of your users participate in the testing to make sure they are able to sign in as expected, and that they can get to the same content that they have access to in your production environment.
- **Publishing workbooks and data sources**. Have users publish workbooks and data sources from Tableau Desktop to make sure this goes as you expect.
- **Viewing published workbooks**.Have users who are familiar with the content try to view published workbooks to make sure they appear as expected. Test views embedded in web pages (for example, in SharePoint pages).
- **Subscriptions and extract refreshes**. Manually run some extract refreshes to confirm that they complete successfully. Run some key scheduled extract refreshes to confirm that they complete as expected.
- **Permissions**. Confirm that permissions are still set as expected for users and content.
- **Command-line utilities and APIs**. If applicable, test the command line utilities (tsm and tabcmd) and programmatic access via APIs.

Performance and user acceptance testing

Use tools like Tabjolt, Replayer, and Scout to do performance and user acceptance testing on your test environment. For more information about these and other performance testing tools, see Performance Monitoring Tools .

Test new features

Take a look at the new features that come with the version you are upgrading to, and at any features that were added between the version you currently have and the new version. Think

about how to help your users understand the benefits of the features that apply to your environment.

For more information on new features, see What's New in the Tableau Server Help.

Communicate about the upgrade

The best way to make an upgrade go smoothly is by letting your organization know ahead of time about the upgrade and how it might impact them. If you've had users help test, take advantage of their experience by having them help communicate the changes they saw while testing. You can also provide user access to the test environment if there are key people who should see the upgraded version before the actual upgrade.

Troubleshoot Tableau Server Install and Upgrade

Follow the suggestions in this topic to resolve common issues with Tableau Server. For additional troubleshooting steps based on process status viewed on the Status page, see Troubleshoot Server Processes.

General Troubleshooting Steps

Many Tableau Server issues can be addressed with some basic steps:

1. Make sure there is enough disk space on each computer running Tableau Server. Limited disk space can cause a failure to install, a failure to upgrade, or problems running Tableau Server.
2. Restart Tableau Server. Issues related to processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau Server, use the `tsm restart` command. This will stop all the processes associated with Tableau Server and then restart them.

3. Reindex Tableau Server. Issues related to indexing can be resolved by reindexing Tableau Server. To reindex Tableau Server, use the `tsm maintenance reindex-search` command. For more information, see Reindexing Tableau Server Search & Browse below.

4. Restart the computer on which Tableau Server is running. Some issues, such as those related to data source connectivity, can be resolved by restarting the server computer.

Common Tableau Server Install Issues

Installation logs location

The install log, `app-install.log`, is located in `/var/opt/tableau/tableau_server-/logs`.

The upgrade log, `app-upgrade.log`, is located in `/var/opt/tableau/tableau_server/logs`.

Multiple install attempts fail

If you attempt to install Tableau Server and the install fails, any subsequent installation attempts are likely to fail unless you run the `tableau-server-obliterate` script to clean Tableau off the computer.

A failed install attempt can leave the computer in a state that causes subsequent attempts to also fail with errors that don't seem directly related to a previous install attempt. One possible error is:

```
Enabling and starting all services
+ services=(appzookeeper* tabadmincontroller* tabsvc* licenseser-
vice* fnplicenseservice* tabadminagent* clientfileservice*)
+ systemctl_user enable appzookeeper_0.service 'tabad-
mincontroller*' 'tabsvc*' 'licenseservice*' fnplicenseservice_0.ser-
vice 'tabadminagent*' 'clientfileservice*'
++ id -ru a_tabadminpoc
```

```
+ local unprivileged_uid=222954
+ su -l a_tabadminpoc -c 'XDG_RUNTIME_DIR=/run/user/222954 systemctl
--user enable appzookeeper_0.service tabadmincontroller* tabsvc*
licenseservice* fnplicenseservice_0.service tabadminagent* cli-
entfileservice*'
Failed to execute operation: No such file or directory
```

To fix this problem, run the `tableau-server-obliterate` script to clean up any left over remnants of the previous install attempt and then restart the computer. For more information, see Running the tableau-server-obliterate script .

> **Important:** If you created a backup of Tableau (`<file>.tsbak`) you want to keep (for example, to restore to your new installation), copy that file to a safe location on another computer to guarantee it is not removed when you clean up your Tableau computer.

Install fails due to hardware requirements

Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. The requirements apply to all computers on which you are installing Tableau Server. For details on minimum hardware requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server.

Install or upgrade fails due to CPU requirements

Beginning in version 2020.4.0 Tableau Server requires CPUs that support SSE4.2 and POPCNT instruction sets. You cannot install or upgrade Tableau Server 2020.4.0 or oater on computers that have CPUs which do not support these instruction sets.

You may see this error message when installing a new installation, or in preparation for upgrading an existing installation:

```
Your computer’s processor doesn’t meet the minimum requirements that
Tableau requires to install the software. If you are using a VM,
make sure Processor compatibility mode is off.
```

The SSE4.2 and POPCNT instruction sets have been common for more than 10 years and most newer CPUs support them, but if you get an error related to processor minimum requirements when attempting to install or upgrade Tableau Server on a Virtual Machine (VM), Processor compatibility mode may be enabled on the VM. To successfully install or upgrade Tableau on a VM, make sure the Processor compatibility mode is turned off.

Common Tableau Server Upgrade Issues

Upgrade logs location

The upgrade log, `app-upgrade.log`, is located in `/var/opt/tableau/tableau_server/logs`.

Maps do not display or display incompletely after upgrading

Beginning with Tableau version 2019.2, the internet access requirements changed for maps. If you are upgrading from version 2019.1.x or earlier to version 2019.2.x or later, and maps are not displaying as expected, confirm that your environment is configured to allow access on port 443 to `mapsconfig.tableau.com` and `api.mapbox.com`.

In version 2019.1.x or earlier, access was necessary to `maps.tableausoftware.com`.

For more details on internet access requirements, see Communicating with the Internet.

Upgrade script error: "Tableau Server Version change validation failed."

When upgrading, if you run the `upgrade-tsm` script from the `scripts.<version_code>` directory for the earlier version, the upgrade will fail with an error:

```
Tableau Server Version change validation failed.
Tableau Server <version> is already installed.
```

If you get this error, change to the `scripts.<version_code>` directory for the version you just installed and run the script from there.

Upgrade multi-node, initializing additional node fails with "Enter your credentials again" error

If you attempt to initialize an additional node when upgrading Tableau Server and see this error:

```
Enter your credentials again. The credentials you enter must provide administrative access to the computer where you generated the configuration file.
```

this is an indication that the node is unable to connect to or communicate with the initial node. This can happen for multiple reasons:

- The credentials you entered are not valid or you mistyped them. The credentials must be for a user who has administrative permissions on the computer where Tableau Server was first installed. You do not need to use the credentials of the user who created the bootstrap file but doing so will ensure you are using valid credentials.
- The local firewall of the computer you are trying to add is not allowing communication to the initial node. For more information, see Local firewall configuration.

Upgrading fails due to lack of disk space

If there is not enough disk space for the Tableau Server Setup program to run and do the upgrade, the installation will fail. The amount of disk space required will depend on the size of your repository database and the number and size of your extracts.

To free up disk space:

1. Create a log archive snapshot using the `tsm maintenance ziplogs` command.

   After you create the ziplogs file, save it to a safe location that is not part of your Tableau Server installation.

2. Clean up unnecessary files using the `tsm maintenance cleanup` command. For more information, see Remove Unneeded Files.

Upgrade fails on RebuildSearchIndex job

Beginning with version 2020.1.x, the final step in an upgrade is to rebuild the search index. At this point all services have been upgraded, so if this job fails, you can manually reset the search server by running the `tsm maintenance reset-searchserver` command. You do not need to obliterate and start over.

The error will be:

```
An error occurred while rebuilding search index.
```

To reset the search server :

1. On the initial node, open a terminal session.

   This must be a new terminal session because the upgrade script updates system environment for the new version.

2. Rebuild the search index using the tsm maintenance reset-searchserver command.

Upgrade fails on 2022.1 and later

After upgrading Tableau Server 2022.1 (or later), restoring a Tableau Server backup as part of your upgrade process can cause the following error:

"The backup cannot be restored because Tableau Server uses the new identity service tables by default."

This issue occurs because Tableau Server 2022.1 (and later) uses an identity schema that is different from the identity schema used by the backup. To resolve this issue, see Troubleshoot Issues with the Identity Migration.

Upgrade fails on 2020.4.0 or later

Beginning with version 2020.4.0, the Checkpoint Upgrade feature allows you to retry a failed upgrade. In general, this is most useful for experienced server administrators and IT professionals who are comfortable with Tableau Server log files and are willing to search through them. But the feature can help in all failed upgrades because it allows you to rerun the

upgrade-tsm script, and the script is run from the last successful step, saving time. For those with experience, it may be possible to identify problems like disk space problems, or permissions issues, correct them, and rerun the upgrade.

If you are upgrading to version 2020.4.0 or later and the upgrade fails, the following steps may help you to complete the upgrade:

- Rerun the `upgrade-tsm` script. Upgrade failures are sometimes a result of timeouts during the upgrade process, and rerunning the script can allow the upgrade to get beyond intermittent or occasional timing issues. This is also a step that is safe to do, and easy. Rerunning the script will do no harm, and at worst, the upgrade will fail again at the same point, but without needing to go through any previous steps.

  The script is located in the `\scripts` directory:

  `opt/tableau/tableau_server/packages/scripts.<version_code>/upgrade-tsm`

  If your Tableau Server upgrade isn't successful when you rerun the upgrade-tsm script, and you are comfortable with Tableau Server logs, you can take these additional troubleshooting steps:

- Look at the output of the script in the command window. Useful error messages may help you identify the cause of the upgrade failure and give you some ideas for how to correct the issue.

- Look in the `app-upgrade.log` file. Any errors that are displayed at the command line will also appear in the `app-upgrade.log` file, often with more details.

- Look in the `tabadmincontroller.log` file. Upgrade problems that aren't easily identifiable in the above two instances are likely the result of an issue in a job. The `tabadmincontroller.log` file may have more information that helps you diagnose the issue.

Note: For information about log file locations, see Tableau Server Logs and Log File Locations.

Upgrade fails due to permission problems with the backup/restore file location

With versions of Tableau Server before 2022.1.0, If the file location for the backup/restore file does not have the correct permissions, the upgrade script will fail with an error about not being able to read the backup file or not being able to restore the repository.

Beginning with version 2022.1, the upgrade script confirms the permissions of file location for the backup/restore file before starting the upgrade so the file can be written to and read from the location during the upgrade to the new version of Tableau Server.

The errors will be similar to these:

```
The tableau user does not have permission to read the backup file:
<backup/restore basefilepath>.

Repository restore failed.
An error occurred during installation.
An error occurred while restoring repository.
```

The location used by TSM for backup and restore is defined by the `basefilepath.backuprestore` configuration key and has a default that the installation program sets up with correct permissions, but these may be impacted by organization IT rules or if you change the location to one you have created yourself. A new command available starting in 2022.1 allows you to check the permissions on the backup/restore file location immediately after creating it, to avoid any permission-related problems. For details about that command, see tsm maintenance validate-backup-basefilepath.

For details about the backup/restore file path, see tsm File Paths.

Upgrade succeeds but published data sources cannot be accessed

In limited, specific scenarios, after upgrading Tableau Server from version 2021.3 to early versions of 2023.1 or 2023.3, attempts to connect to or refresh existing published data sources

fail with this error:

```
java.io.FileNotFoundException: Unable to fetch data from any other
host. This may indicate a lost or invalid folder.
```

This could happen if:

1. You upgrade a Tableau Server installation that was version 2021.3.x at any point (you could be running 2021.3 or have upgraded from 2021.3 to a 2022.x version)

   and

2. You upgrade that installation *to* early versions of 2023.1 or 2023.3

No impact

There are no problems in the following situations:

- In all other upgrade paths *from* 2021.3
- In all other upgrade paths *to* 2023.1 or 2023.3
- In all fresh installations of 2023.1 and 2023.3

More information

As of September 16, 2024, all problematic versions have been removed from the download site. If you need to upgrade to version 2023.1.x or 2023.3.x, upgrade to maintenance release versions 2023.1.16 or higher, or 2023.3.9 or higher.

For more information about this issue, see the Known Issue.

Common Settings Import Issues

Import of settings file causes "not present on any node" validation error due to missing services

If you are upgrading by installing a new version of Tableau Server and importing a settings file from an earlier version, you may encounter topology validation errors when running the `tsm settings import` command.

This can happen when you export a settings file from an older version of Tableau Server and import it into a new version, and new services have been added to Tableau between the two versions.

Errors will be similar to this (the specific service may be different):

```
>tsm settings import -f 20183-export.json

Pending topology set.
There are 1 topology validation errors/warnings.

Service 'elasticserver' is not present on any node in the cluster.
Service: Elastic Server
```

To resolve this issue, add any missing services to Tableau Server:

1. For any service that generated a validation error, add the service with an instance count of 1.

   For example, if the Elastic Server is not present in the cluster, set the process instance count to 1 using the service name that appears in the first line of the validation error message:

   ```
   tsm topology set-process -n node1 -pr elasticserver -c 1
   ```

   Repeat this step for each service that results in an error.

2. When you have no more warnings or errors, apply the pending changes:

   ```
   tsm pending-changes apply
   ```

Your settings should be imported successfully.

Import of settings file causes "configuration value you specified does not match" error

If you are installing a new version of Tableau Server and import a settings file from an earlier version, you may encounter configuration validation errors when running the `tsm settings`

`import` command. These can occur when a settings file includes a configuration value that has since been removed from Tableau.

The error will look similar to this (the configuration key may be different):

```
>tsm settings import -f 20183-export.json
Configuration error: At least one configuration value you specified
does not match a known configuration key. This applies to the fol-
lowing keys: '[features.TsmConfigFileService]'
Use this parameter to override unknown key error: --force-keys
```

To resolve this issue, edit the settings file you are importing to remove the reference to the configuration key or keys in the error:

1. Copy the JSON settings file and save the copy for backup.
2. Open the JSON settings file in a plain text editor.
3. Locate and delete the entire line that includes the key. In this example, `features.TsmConfigFileService`:

   ```
   "configKeys" : {
     "config.version" : 19,
     "tabadmincontroller.port" : "8850",
     "endpoints.enabled" : false,
     "endpoints.health.enabled" : true,
     "features.TsmConfigFileService" : true,
     "tableau_projects.language" : "en",
   ```

   The above is an example of a small section of an exported settings file and is not intended to represent the entire contents of the file.

4. Save the settings file and import it again.

You may encounter additional errors related to topology validation. For information about solving those errors, see Import of settings file causes "not present on any node" validation error due to missing services above.

"You cannot directly modify instances of the Coordination Service" error

This error can occur in two situations:

- When you import a Tableau Server settings file into an installation that has a different Coordination Service topology than the settings file does
- When you attempt to configure the Coordination Service using the `tsm topology set-process` command

If you see this error after importing a settings file:

The Tableau Server settings file has a different Coordination Service topology than the target server does. This can happen if you are upgrading Tableau Server by installing a new version and importing a settings file from an earlier version. If you have not explicitly deployed a Coordination Service ensemble on the target server, it has a single instance of Coordination Service, on the initial node.

To correct this error you can take either correct the mismatch from the command line, or by editing the settings import file. You can also discard all pending changes, deploy the Coordination Service on the target computer to match the settings in the import file, and reimport the settings file.

To correct the mismatch from the command line, for each node that generates an error, use the tsm topology set-process command to revert the instance count of Coordination Service.

1. Run the `tsm pending-changes list` command. The output shows you which nodes have changes.

2. Find the node or nodes where the Coordination Service count is changed.

   For example, if the settings file had a Coordination Service instance on node2, but the target system did not have any Coordination Service instance on that node, the count for node 2 would show as changed from 0 to 1 by the import of the settings file:

   ```
   C:\Windows\system32>tsm pending-changes list
   Configuration
   There are no pending configuration changes.
   ```

```
Topology
node2:
            Coordination Service
                                New Instance Count:1
                                Old Instance Count:0
```

3. Use the `tsm topology set-process` command to set the count back to the "Old Instance" value.

   For the example above:

   ```
   tsm topology set-process -n node2 -c 0 -pr "Coordination Service"
   ```

4. Once you have reset any Coordination Service instance count that was changed, apply pending changes:

   ```
   tsm pending-changes apply
   ```

If you see the error when setting the process count for Coordination Service manually:

This error can also occur if you attempt to update the Coordination Service directly, using the `tsm topology set-process` command instead of the `tsm topology` commands for managing the Coordination Service. If you tried this:

1. Use the `tsm pending-changes discard` command to discard the pending changes.
2. Use the correct commands for configuring the Coordination Service. For more information, see Deploy a Coordination Service Ensemble .

Starting Tableau Server

Tableau Server cannot determine if it fully started

In some instances Tableau Server may report that it could not determine if all components started properly on startup. A message displays: "Unable to determine if all components of the service started properly."

If you see this message after starting, verify that Tableau Server is running as expected by using a `tsm status -v` command.

If the status shows as running ("Status: RUNNING"), then the server successfully started and you can ignore the message. If the status is DEGRADED or STOPPED, see "Tableau Server doesn't start" in the next section.

Tableau Server doesn't start

If Tableau Server does not start or is running in a degraded state, run the `tsm restart` command from a command prompt. This will shut down any processes that are running, and restart Tableau Server.

Reindexing Tableau Server Search & Browse

Problems that can be solved by rebuilding Search & Browse index

Symptoms of an index that needs to be rebuilt include:

- A blank list of sites when a user attempts to log in
- A blank list of projects when a user tries to select a project
- Missing content (workbooks, views, dashboards)
- Unexpected or inaccurate alerts (for example, an "refresh failed" alert on a workbook that does not include an extract)

If you see any of these behaviors, reset and rebuild the Search & Browse index using the `tsm maintenance reset-searchserver` command.

Activating Tableau Server

Tableau Server license activation fails

In some instances Tableau Server license activation may fail. Error messages can range from a very generic one:

- `An error has occurred`

To more specific messages:

- `Function flxActCommonLicSpcPopulateFromTS returned error 50030, 71521,`
- `No license found for 'Tableau Server'`

To resolve this issue, try these solutions in the order listed:

Confirm you can access the licensing server

The Tableau licensing service was moved to a new data center on October 6, 2018. This means any environments that required special configuration (static IP safe listing for example) to access licensing.tableau.com or licensing.tableau.com will need to be updated before you can activate, refresh, or deactivate a Tableau product key.

To test access, type the URL and the port of the licensing server in a browser:

`https://licensing.tableau.com:443`

and:

`https://atr.licensing.tableau.com/_status/healthz`

If you are able to access the server, a "Test success" message displays for the first server, and an "OK" message displays for the second.

Tableau Server needs to make a connection to the following internet locations for licensing purposes:

- atr.licensing.tableau.com:443
- licensing.tableau.com:443
- register.tableau.com:443
- o.ss2.us
- s.ss2.us
- crt.rootca1.amazontrust.com

- crt.sca1b.amazontrust.com
- crt.sca0a.amazontrust.com
- crt.sca1a.amazontrust.com
- crt.sca2a.amazontrust.com
- crt.sca3a.amazontrust.com
- crt.sca4a.amazontrust.com
- *.digicert.com
- ocsp.*.amazontrust.com
- crl.*.amazontrust.com
- crt.rootg2.amazontrust.com

Requests to the above domains may be on port 80 or 443. Port 80 is used for certificate validation (revocation, certificate chain, etc). Port 443 is used for SSL connections.

Requests to the `ocsp.*.amazontrust.com` and `crl.*.amazontrust.com` domains are managed by Amazon for certificate revocation information. See ACM certificate characteristics for more information. We recommend that you install the Amazon root certificates in the certificate trust store on the computer running Tableau. To download and install the Amazon root certificates, see Certificate Authorities on the Amazon Trust Services web site.

Verify the date and time

Verify the date and time on the initial Tableau Server computer is correct. If the clock is set to a time and date earlier than the current date, Tableau Server cannot be activated.

Force the product key to be read again

1. On the initial Tableau Server computer, log on as a user with sudo access.
2. Change to the Tableau Server bin directory. By default this is:

```
/opt/tableau/tableau_server/packages/bin.<version_code>/
```

3. Type the following commands:

```
tsm stop

./lmreread
tsm start
```

Send the contents of trusted storage to Tableau Support

If FlexNet Licensing Services is installed and running but you're still seeing an error, there might be a problem with the Tableau product key information. To resolve this issue, complete the following steps to create a file of the key information located in trusted storage.

1. On the initial Tableau Server computer, log on as a user with sudo access.
2. Type the following command:

   ```
   serveractutil -view > <machine_name>-LicResults.txt
   ```

   This creates the `<machine_name>-LicResults.txt` file in your current directory. If you don't have write permissions for that location and see an error, change to a location where you do have permission to create a file and run the command again.
3. Contact Tableau Support (http://www.tableau.com/support/request) and include the `<machine_name>-LicResults.txt` file that you created.

tabcmd Installation Problems

Installing tabcmd separately

tabcmd is automatically installed on the initial Tableau Server node when you install Tableau Server, but if you want to run it on another computer, you need to download and install tabcmd separately. For details, see Install tabcmd.

Problems installing tabcmd on Linux

tabcmd requires Java 11 to run properly. On RHEL-like systems, this will be installed as a dependency when installing tabcmd. On Debian-like systems, you need to install Java 11 separately if it is not already installed.

As of July 2022, Debian distributions are no longer supported. For more information, see this Tableau Community post.

Java is not installed

If you see errors similar to this when installing tabcmd, confirm that Java 11 is installed on your Linux computer:

```
Cannot find 'java' in your PATH. Install 'java' and make sure it is
in your PATH to continue.
```

Incorrect version of Java is installed

If you see errors similar to these, confirm that Java 11 is installed:

```
Exception in thread "main" java.lang.UnsupportedClassVersionError:
com/tableausoftware/tabcmd/Tabcmd : Unsupported major.minor version
52.0
```

or.

```
*** Uncaught exception NoClassDefFoundError: javax/xm-
l/bind/JAXBException
*** See the logs for the stacktrace.
```

Uninstall Tableau Server

Do not uninstall Tableau before upgrading. For details on upgrading, see Upgrading from 2018.1 and Later (Linux).

You can have multiple versions of Tableau Server installed at the same time. This allows you to run most of an upgrade while an existing version is running, and reduces downtime and

impact to users. Once you have upgraded, you can uninstall your previous version. Doing this frees up disk space. You do not have to uninstall the previous version.

This article explains how to uninstall previous versions, after you've upgraded to a newer version.

Uninstalling and completely removing Tableau Server

There are two primary "uninstall" scenarios that Tableau Server on Linux supports:

- **Uninstall Tableau Server:** *After you upgrade* to a new version of Tableau Server you can uninstall your previous version to free up disk space. Continue reading for information about uninstalling Tableau.
- **Remove Tableau Server:** If you want to complete remove Tableau Server from a computer, you can use a script provided by Tableau to remove Tableau Server and all related files. *This removes all data as well as server components, so should only be done if you know you want to reset the computer to a pre-Tableau state.* You might need to do this if Technical Support recommends this step when troubleshooting an installation problem. We recommend you create a backup of your data before removing Tableau. Save the backup file to a safe location on a computer that is not part of your Tableau installation. Completely remove Tableau Server without uninstalling any version first. The script will uninstall all existing versions found on the computer. If you have already uninstalled your existing version and now want to completely remove Tableau, you can find the script to do so in a temporary location. For more details, see Remove Tableau Server from Your Computer.

Uninstall a Tableau Server package

Use this procedure to free up disk space by uninstalling packages for previous Tableau Server versions after you have upgraded to a newer version of Tableau Server.

1. Look at the environment.bash file to confirm which version of Tableau Server is currently in use. At a command prompt, type:

   ```
   grep TABLEAU_SERVER_DATA_DIR_VERSION /etc/opt/tableau/tableau_
   server/environment.bash
   ```

2. Determine which versions of the Tableau Server package are installed on your computer.

   - On RHEL-like distributions, including CentOS, run the following command:

     ```
     yum list installed tableau-server"*"
     ```

   - On Ubuntu, run the following command:

     ```
     apt list --installed tableau-server"*"
     ```

3. Remove the Tableau Server package with your package manager.

   - On RHEL-like distributions, including CentOS, run the following command:

     ```
     sudo yum remove tableau-server-<version>.x86_64
     ```

   - On Ubuntu, run the following commands:

     ```
     sudo apt-get purge tableau-server-<version>
     ```

Reinstall a Tableau Server package that was accidentally uninstalled

Do *not* uninstall the package for your currently running version of Tableau Server. Doing so will make the server unusable. To completely remove Tableau Server and all its files, see Remove Tableau Server from Your Computer.

When you uninstall the Tableau Server package for the current instance of Tableau Server the following operations run:

- All files under /opt/tableau/tableau_server are removed. These files are the unmodified installation files.
- Tableau Server services are stopped and disabled

- Service files for all Tableau Server services are persisted
- Data files are left in place

If you accidentally uninstall the package for your currently running version of Tableau Server, follow this procedure to correct the situation.

To reinstall after uninstalling the running instance of Tableau Server:

1. Reinstall the package for the version you accidentally uninstalled.
2. Run `initialize-tsm`.

Remove Tableau Server from Your Computer

Warning: The steps below *completely remove* Tableau Server on Linux, and delete users and groups created by `initialize-tsm`, all related data, and configuration information. This includes any files in `/tmp` or `/var/tmp` that are owned by users configured in `/etc/opt/tableau/tableau_server/environment.bash` as privileged and unprivileged users (by default, `tsmagent` and `tableau`). Tableau Server licenses are also deactivated, unless you omit the `-l` option when running the command shown below and the computer is connected to the internet.

If you want to uninstall a particular Tableau Server package to free up disk space (after upgrading, for example), see Uninstall Tableau Server.

As part of the regular installation of Tableau Server, a script is installed that provides you a way to completely remove Tableau and all associated files from your computer. This is something you would only do if you did not care about your Tableau data, configuration, or log files, or if you are working with Tableau Technical Support and need to reinstall Tableau Server after an installation or upgrade attempt fails. The obliterate script will not remove any drivers you installed separately, even those you installed to use with Tableau Server.

The `tableau-server-obliterate` script is intended for when you want to completely remove Tableau Server from your computer. You might want to do this for a couple of different reasons:

- You no longer want Tableau Server installed on the computer. Use the `tableau-server-obliterate` script to remove Tableau Server completely. If this is the case, and the computer is connected to the internet, you can include a `-l` option to deactivate the Tableau Server license.
- Troubleshooting Tableau Server installation problems—If you run into issues installing Tableau, you may need to use the `tableau-server-obliterate` script to completely remove Tableau Server from your computer before reinstalling. Doing this will clean up any older settings or states (such as the `/etc/opt/tableau/tableau_server/environment.bash` file) and allow you to reinstall on a "clean" computer. If you are doing this, you can leave off the `-l` option to preserve licensing information on the computer. When you omit the `-l` option, you will not need to activate your license when you reinstall Tableau Server.

What tableau-server-obliterate does

The intent of the `tableau-server-obliterate` script is to completely remove Tableau Server from your computer. This includes files, system settings and configurations, and, if you specify, licensing information.

When you run `tableau-server-obliterate`, the following steps are taken:

- Uninstall is run for all installed versions of Tableau Server (yum erase or apt-get remove).
- Most contents of the data directory is removed (by default `/var/opt/tableau/tableau_server`). Backup and log files are preserved by default. See the section below, "Preserving Tableau Server backup and log files."
- All semaphores and shared memory segments are deleted.
- All temp files owned by the "tableau" user are deleted from `/tmp` and `/var/tmp`.
- All users and groups created during install are deleted.
- `/etc/opt/tableau` is deleted.

- Trusted certificates are removed from `/etc/pki/ca-trust/source/anchors/TableauServer` and `/usr/share/ca-certificates/tableau`
- Configuration files are removed from:
  - /etc/sysctl.d/99-tableau-server.conf
  - /etc/profile.d/tableau_server*
  - /etc/security/limits.d/99-tableau_server*
  - /etc/systemd/logind.conf.d/tableau_server*
  - /usr/share/bash-completion/completions/tsm
  - etc/bash_completion.d/tsm
  - /usr/share/bash-completion/completions/tabcmd
  - /etc/bash_completion.d/tabcmd
  - /run/tableau
  - /usr/lib/tmpfiles.d/tableau-server.conf
- All server licenses are deactivated if you use the `-l` option and the computer is connected to the internet. This option does not work in offline situations.

Preserving Tableau Server backup and log files

Prior to version 2020.1 of Tableau Server, running the `tableau-server-obliterate` script deleted all content from the Data directory. Beginning with version 2020.1, the default behavior of the `tableau-server-obliterate` script has changed: the script copies and saves Tableau Server backup and log files to the `logs-temp` directory. The default location for the logs-temp directory is at `/var/opt/tableau/logs-temp`. You can set options on the script to disable this new functionality.

To change the default behavior and to remove backup or log files, include on of the following options when running the `tableau-server-obliterate` script:

- `-k` Do not copy backups to logs-temp directory.
- `-g` Do not copy logs to logs-temp directory.
- `-a` Do not copy anything to logs-temp directory.

Running the tableau-server-obliterate script

You can completely remove Tableau Server from a computer, either preserving the licensing information, or removing the licensing information along with all other aspects of Tableau

Server. You might want to preserve licensing if you are going to reinstall Tableau Server on the same computer.

An older version of `tableau-server-obliterate` may miss files from new versions of Tableau Server. Always run the obliterate script for the version of Tableau Server that is installed on the computer.

To completely remove Tableau Server without removing server licensing

The example script in this procedure also includes the `-a` option to remove Tableau Server backup and log files.

> **Note:** If you plan to reinstall Tableau Server and Activate Tableau Server Using the Authorization-To-Run (ATR) Service, we recommend that you remove licensing information before reinstalling and activating Tableau Server using Server ATR.

1. On the initial node, open a terminal session.
2. Run the `tableau-server-obliterate` script:

   ```
   sudo /opt/tableau/tableau_server/packages/scripts.<version_
   code>/tableau-server-obliterate -a -y -y -y
   ```

   > **Note:** If you have already uninstalled Tableau Server and now you want to remove it, run the copy of the `tableau-server-obliterate` script located in the `/var/tmp` directory. If the script is not in that directory, reinstall the Tableau Server package and then run the `tableau-server-obliterate` script.

   If you have a multi-node (distributed) installation of Tableau Server, run the `tableau-server-obliterate` script on each node in the cluster.
3. Restart each computer you ran the `tableau-server-obliterate` script on.

To completely remove Tableau Server and licensing

The example script in this procedure also includes the `-a` option to remove Tableau Server backup and log files.

1. On the initial node, open a terminal session.
2. Deactivate any active product keys.

   ```
   tsm licenses deactivate -k <product_key>
   ```

3. Run the `tableau-server-obliterate` script:

   ```
   sudo /opt/tableau/tableau_server/packages/scripts.<version_
   code>/tableau-server-obliterate -a -y -y -y -l
   ```

   The `-l` option removes all licensing files from the computer if the computer is connected to the internet. The script first attempts to deactivate any active licenses, but it will remove all licensing information whether or not deactivation was successful. This is why we recommend you run the `tsm licenses deactivate` command before running the obliterate script.

   > **Note:** If you have already uninstalled Tableau Server and now you want to remove it, run the copy of the `tableau-server-obliterate` script located in the `/var/tmp` directory. If the script is not in that directory, reinstall the Tableau Server package and then run the `tableau-server-obliterate` script.

   If you have a multi-node (distributed) installation of Tableau Server, run the `tableau-server-obliterate` script on each node in the cluster. You do not need to deactivate licenses on any additional nodes.

4. Restart each computer you ran the `tableau-server-obliterate` script on.

Help Output for tableau-server-obliterate Script

The following help content is the output when you run the following command:

```
sudo ./tableau-server-obliterate -h
```

The ./tableau-server-obliterate script is installed to `/opt/tableau/tableau_server-/packages/scripts.<version_code>/`.

Output

```
Remove Tableau Server from this computer.

This script will stop and remove all Tableau Services from this
computer. It also removes data and configuration files. It leaves
licensing in place. It also preserves logs and backup files, which
are moved to a temp directory under the Tableau data folder. You
can
force removal of these files, and licensing, using optional para-
meters.

This script is destructive and not reversible. It should only
be used to clean Tableau Server from a computer. For multi-node
installations, you must run the script separately on each node.

This script must be run as the root user.


-y              Required. Yes, remove Tableau Server from this com-
puter.
                Must be specified three times to confirm.
-l              Optional. Delete licensing files and data. This com-
mand
                will attempt to deactivate licenses before deleting
                licensing data. Internet access is required for license
                deactivation. Offline deactivation is not supported.
                To deactivate license before removing Tableau Server,
```

```
                run 'tsm licenses deactivate' before running this script.
  -k            Optional. Do not copy backups to logs_temp directory.
  -g            Optional. Do not copy logs to logs_temp directory.
  -a            Optional. Do not copy anything to logs_temp directory.
```

Migrate

The topics in this section describe how to migrate from Tableau Server to Tableau Cloud. Topics here also cover Tableau Server to Tableau Server migration scenarios.

> Looking for Tableau Server on Windows? See Migrate

Migrate Tableau Server to Tableau Cloud

For information about migrating small (less than 100 users) Tableau Server to cloud, see Tableau Cloud Manual Migration Guide.

For information about planning a move to Tableau Cloud, see Tableau Blueprint: Move Tableau to Cloud.

Technical Considerations for Migrating from Tableau Server to Tableau Cloud

This topic covers most of the technical considerations that may arise in a migration from Tableau Server to Tableau Cloud. It's written for the administrators that currently manage Tableau Server and who are considering a migration to Tableau Cloud. We recommend that you review this topic as a primary step in your migration plan. Additionally, work with your account team to understand if a migration makes sense for your organization.

The topic reflects the product capabilities in Tableau Server 2023.3.

For additional information and to initiate a migration to Tableau Cloud, visit the Tableau Cloud Migration site on Tableau.com.

Summary outline

This topic is organized into four major areas that are familiar to Tableau administrators. The following tables show a summary feature comparison between Tableau Cloud and Tableau Server within each major area. For more information on a given feature, see the corresponding section later in this topic.

Security, administration, and governance		
	Tableau Cloud	**Tableau Server**
Compliance	See Salesforce Compliance portal. Access to this site may require permission from your account manager.	Managed by customer
Hosting and upgrades	Managed by Tableau	Managed by customer
Site level administration	Single site default	Managed by customer
Telemetry data	Limited	Managed by customer
Observability data	Admin Insights with Activity Log	Managed by customer with repository access and Activity Log
Availability	99.9% with SLA for Premium Support customers	Managed by customer

Data Connectivity		
	Tableau Cloud	**Tableau Server**
Files	Supported	Supported
Applications and databases	Supported, with database-	Supported

	specific limitations	
Custom connectivity	Limited support	Supported
Cube data sources	Not supported	Supported

User management and licensing		
	Tableau Cloud	**Tableau Server**
Licensing	Internal: role-based External: role and usage-based	Internal: role and core-based External: role and core-based
SSO	Supported	Supported
Active Directory, Kerberos, LDAP directory	Not supported	Supported
Automated user and group management	Okta, OneLogin, Azure AD, and custom tooling	Active Directory, LDAP

Extensibility and external integrations		
	Tableau Cloud	**Tableau Server**
Automated tooling through APIs	Supported	Supported
R and Python integration	Supported	Supported
Embedding	Supported	Supported
Customization	Limited support	Supported

Security, administration, and governance

Security and compliance

Make sure Tableau Cloud security certifications meet your organization's requirements. Tableau Cloud is compliant with ISO 27001/27017/27018 and SOC 2/3 and adheres to data privacy requirements such as those outlined in GDPR. To see a complete and up-to-date list of certifications go to the Salesforce Compliance portal. Access to this site may require permission from your account manager.

Additionally, Tableau Cloud and Salesforce adheres to the strictest standards for regional data security and privacy. Customer data never leaves the region in which it is hosted. Salesforce adopts a Shared Responsibility Model. In this model, we work with you to ensure that all the proper security and compliance controls are enabled by Salesforce and your organization.

Hosting and upgrades

Tableau hosts and manages Tableau Cloud for customers. Tableau Cloud is always running the latest version of Tableau. Major upgrades are completed with zero downtime. Routine maintenance (activities like upgrades, backups, performance tuning, etc.) is performed to minimize downtime. In the case where downtime is required, maintenance occurs during pre-communicated windows. For more information about the Tableau maintenance schedule and how to sign up for maintenance notifications from Tableau Trust, see Tableau Cloud System Maintenance.

Tableau Cloud is run in Amazon Web Services. You can choose to deploy your site in any of the AWS Regions listed here.

Tableau takes daily backups of the environment for Disaster Recovery. However, these backups are for the purpose of restoring the system as a whole. Customers are responsible for maintaining backups of any files they want to recover if they’re removed or deleted from Tableau Cloud.

Site level administration

By default, Tableau Cloud deployments provide a single Site for deploying Tableau to your users. A single site model simplifies administration of the environment while allowing comprehensive governance scenarios as recommended in Blueprint.

To ensure consistent performance across the platform, Tableau Cloud restricts sites with site capacity limits. See Tableau Cloud Site Capacity for a list of capacity types and associated allowances. The capacity restrictions include limits for overall storage and for tasks that you may perform on the site. You can evaluate if any of your use cases would be impacted by these limits by connecting to the Tableau Server repository and comparing usage to the site capacity limits. For more information about connecting to the repository, see Collect Data with the Tableau Server Repository.

Observability data

In Tableau Cloud, you can't connect directly to the Tableau Server repository database. Instead, Site Admins have access to a Project called Admin Insights that contain pre-built observability dashboards and data sources. See Use Admin Insights to Create Custom Views. Admin Insights help you to understand usage, performance, licensing, user management and more. You can also use the provided data sources to build custom reports or take advantage of Accelerators built on this data. For an example, see the Dashboard Load Times Accelerator. Default data retention in Admin Insights is 90 days. With Advanced Management you receive an extended retention period of 365 days and get access to Activity Log for more detailed observability data.

Availability

Tableau Cloud offers a service level agreement (SLA) that guarantees monthly service availability of 99.9%. We provide this level of service to all Tableau Cloud customers. However, in the unlikely event availability falls short of 99.9%, only those customers subscribed to Premium Support for Tableau Cloud are eligible for a service credit on their account.

Tableau Cloud customers who subscribe to Standard Support or Extended Support benefit from the SLA because Tableau maintains 99.9% availability across all Tableau Cloud

deployments. However, Standard and Extended Support customers can't request service credits if Tableau Cloud fails to meet 99.9%.

Learn more about Technical Support Services and Premium Support for Tableau Cloud.

Data connectivity

Where your data resides determines which of the two connectivity options you use with Tableau Cloud:

- Tableau Cloud can directly connect to the cloud data sources listed below. For direct connections, some databases require authorizing Tableau Cloud to access these data sources before you connect. See Authorize Access to Cloud Data Published to Tableau Cloud.

Supported Tableau Cloud connectors

Alibaba AnalyticsDB for MySQL‡

Alibaba Data Lake Analytics‡

Amazon Athena‡

Amazon Aurora for MySQL‡

Amazon EMR Hadoop Hive‡

Amazon Redshift‡

Apache Drill‡

Azure Data Lake Storage Gen2‡

Azure Synapse Analytics

Dropbox*‡

Esri Connector‡

Exasol‡

Google BigQuery*‡

Google Cloud SQL (MySQL compatible)‡§

Google Drive‡

Hortonworks Hadoop Hive

Impala‡

Kyvos‡

OData‡

OneDrive*‡

Oracle‡

Pivotal Greenplum Database‡

PostgreSQL‡

Presto‡

Qubole Presto‡

Salesforce‡

SAP HANA (for virtual connections only)‡

(SQL Server compatible)

Box‡

Cloudera Hadoop‡

Databricks‡

Datorama by Salesforce‡

Denodo‡

Dremio by Dremio‡

MariaDB‡

Microsoft Azure SQL Database‡

Microsoft Azure Synapse Analytics‡

Microsoft SQL Server‡

MongoDB BI Connector‡

MySQL‡

SharePoint Lists‡

SingleStore (formerly MemSQL)‡

Snowflake‡

Spark SQL‡

Teradata**‡

Vertica‡

*For more information about using OAuth 2.0 standard for Google BigQuery, OneDrive, and Dropbox connections in Tableau Cloud, see OAuth Connections.

**Teradata web authoring currently doesn't support query banding functionality. See Teradata for details.

‡Supports virtual connections if you have Data Management. See About Virtual Connections and Data Policies in the Tableau Cloud help for details.

§Tableau Cloud doesn't support SSL using Google Cloud SQL.

- For data that is stored on-premises or in a private cloud network, you can deploy Tableau Bridge. For a list of all the connectors that Tableau Bridge supports and doesn't support, see Connectivity with Bridge.

If your workbooks contain embedded data source extracts that take more than 10 minutes to refresh, we recommend publishing each data source separately from the workbook and then refreshing them individually. This approach avoids timeouts for long-running queries. For more information, see Optimize Bridge Refresh Performance and Publish a Data Source.

Files

Tableau Cloud supports various file types. Files hosted on cloud platforms like Google Drive, Dropbox, Box and S3 can use Tableau Cloud direct connectors to support both live and extract connections. Tableau Bridge supports file-based data connections for extracts, but doesn't support live connectivity to file-based data.

Applications and databases

Tableau Cloud supports most of the same Connectors that Tableau Server supports, but there are some differences. Some databases require specific configuration to enable direct Tableau Cloud connectivity. See Allow Direct Connections to Data Hosted on a Cloud Platform.

Tableau Cloud and Tableau Bridge don't support authentication to a database using Kerberos.

Custom connectivity

Tableau Cloud is designed to use the same database Connector for every Tableau Cloud customer and therefore doesn't support customer-specific connectivity solutions. The two notable exceptions are the Hyper API and Tableau Bridge, which can be used to extend or customize your connectivity. The Hyper API can create extracts that can be published to Tableau Cloud. Tableau Bridge can be used to extend some connectivity options, for example, ODBC, JDBC, and Web Data Connectors. See Connectivity with Bridge.

You can't customize connectivity with Connector SDK (.taco) on Tableau Cloud or with Tableau Bridge.

If you're using Virtual Connections as a part of the Data Management offering there may be differences specific to your desired connector. Verify that Tableau Cloud or Tableau Bridge supports your scenario.

Cube data sources

Tableau Cloud doesn't support Cube Data Sources. Instead, we recommend that you connect directly to the underlying database that the cube is built on top of for greater flexibility in your analysis.

Data prep

Tableau Bridge doesn't support running Tableau Prep Flows with Prep Conductor. Instead, follow the process in the KB article, How to Run Tableau Prep Conductor Flows with On-Premises Data in Tableau Cloud, to automate data preparation.

Licensing and user management

Licensing

Tableau Cloud supports licensing with subscription role-based plans. Learn more at the Tableau Pricing page. To eliminate key management tasks, Tableau Cloud defaults to licensing users with login-based license management (LBLM). Legacy perpetual licenses and server core-based licensing aren't supported.

Tableau Cloud doesn't offer a "Guest" user. However, connected applications can be used to provide content for broader usage within your organization. See Use Tableau Connected Apps for Application Integration.

If you're licensing Tableau Cloud for users outside of your organization, usage-based licensing for those external use cases is available. See Raise Revenue and Lower Costs with Usage-Based Licensing for Tableau Embedded Analytics.

Single sign-on

Tableau Cloud supports single sign-on (SSO) through both SAML and OIDC. Direct integrations with Okta, Azure AD, One Login, Salesforce, PingOne, and more are supported.

All users accessing Tableau Cloud are required to utilize multi-factor authentication (MFA) to authenticate into Tableau Cloud.

If you aren't using an SSO provider, you can use Tableau Cloud's built-in authentication type, TableauID with MFA. You can also create custom solutions with SAML or use Connected Applications for embedded solutions.

For more information, see the Authentication section in Tableau Cloud help.

Active Directory, Kerberos, and LDAP

Tableau Cloud doesn't support direct integration with Active Directory. However you can configure SAML with Azure AD. to integrate with Tableau Cloud.

Additionally, most IdPs have an Active Directory integration that would suffice for authentication with Tableau Cloud. For an example, see Active Directory integration (Okta).

Tableau Cloud doesn't support Kerberos authentication or using LDAP as an Identity Store.

Automated user and group management

Tableau Cloud uses System for Cross-domain Identity Management (SCIM) for automating the exchange of user identity information. There are several IdP-specific configurations available in Tableau Cloud. See Automate User Provisioning and Group Synchronization through an External Identity Provider.

The Tableau REST API and tabcmd 2.0 supports many user and group automation tasks.

Tableau Cloud doesn't support direct integration with Active Directory for user or group provisioning.

Extensibility and external integrations

Automated tooling

Tableau Cloud supports REST APIs, Javascript APIs, Metadata APIs, Dashboard APIs, Webhooks, and embedding in HTML pages for authenticated users. See Tableau Developer Tools for more information.

We recommend that you verify your use cases prior to a migration.

There are some differences to be aware of as you plan your automated tooling strategy for Tableau Cloud:

- Tableau Server supports REST APIs for server-level administration. Tableau Cloud supports site-level administration.

- Tableau Server supports a global list of Web Data Connectors. Tableau Cloud requires using Bridge to run extract refreshes for Web Data Connectors.
- The connectors on the Tableau Exchange are only supported by Tableau Server.

APIs supported by Tableau Cloud are optimized for authentication with Tableau personal access tokens (PATs). Plan on refactoring your tooling to use PAT authentication as part of your migration to Tableau Cloud. See Personal Access Tokens.

Extensibility

Tableau Cloud supports analytics extensions with both R and Python. See Configure Connections with Analytics Extensions. However, Tableau Bridge doesn't support analytics extensions, so these extensions are only compatible with data sources that don't require Tableau Bridge.

Tableau Cloud doesn't support R or Python script steps in a Prep flow authored or published to Tableau Cloud.

Embedding

Tableau Cloud supports embedding Tableau into other applications. It supports the Embedding API, various SSO options, and Tableau REST API for user and content management.

There are two main differences for embedding with Tableau Cloud vs Tableau Server:

- Tableau Cloud supports both a subscription role-based licensing model or a usage-based licensing model for embedding scenarios.
- Tableau Cloud uses a single-tenant model. To ensure data separation, you can use a number of different methods. The data separation strategy you use will be determined by your business needs and embedding solution. The following Tableau Cloud help topics provide more information:
    - Use Projects to Manage Content Access
    - Use Tableau Connected Apps for Application Integration
    - Manage Site User Visibility
    - Permissions

Customizations

Tableau Cloud allows you to customize some aspects of the user experience like language, locale, custom logos and custom project images. See Customize the Site and Content Settings.

Tableau Cloud doesn't support custom URLs, custom fonts, or custom welcome banners.

Server to Server Migrations

The topics in this section provide information on migrating between hardware, operating system, and public cloud platforms for Tableau Server.

To migrate a site from one Tableau Server to another, see Export or Import a Site.

To copy or migrate content between Tableau Server projects using the Tableau Content Migration Tool, see About Tableau Content Migration Tool.

Migrate to New Hardware

Use the following procedure to migrate Tableau Server from one computer to another. You might do this if you are upgrading the computer hardware that Tableau is running on.

Specifically, these steps describe how to move Tableau Server data from your in-production computer to a new computer where Tableau Server is installed. Before you start, make sure you have followed the steps in Preparing for Upgrade, including creating a backup and gathering any assets that require manual actions. For details, see Perform a Full Backup and Restore of Tableau Server. You'll need these to restore your Tableau Server data and configuration.

Important: If you perform Blue/Green upgrades or manually upgrade Tableau Server 2021.4 (or earlier) using the tsm maintenance (backup and restore) method, you must enable `legacy-identity-mode` before you can restore to Tableau Server 2022.1 (or later). For more information, see Troubleshoot Issues with the Identity Migration.

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

Note: This operation includes steps that you may need to perform using the TSM command line.

1. Deactivate your product key on your original installation of Tableau Server. You need to do this so you can activate it on the new computer. For more information, see "Tableau Server product key activation" below.

2. Install Tableau Server on the new computer, activate your license, configure initial settings, and create your admin user. For details, see Install and Configure Tableau Server.

3. Copy your backup file *.tsbak* to the folder location specified by the `basefilepath.backuprestore` variable. By default this is: `/var/opt/tableau/tableau_server/data/tabsvc/files/backups/`.

   Note: You can change the location for the backup file. For more information, see tsm File Paths.

4. Next, stop Tableau Server:

   ```
   tsm stop
   ```

5. Restore your in-production data to your new Tableau Server installation using the `tsm maintenance restore` command:

   ```
   tsm maintenance restore -f <filename>
   ```

   where `<filename>` is the name of the *.tsbak* file you copied in step 3.

For more information about restoring Tableau Server data, see Restore from a Backup.

6. Start the server:

```
tsm start
```

7. **Distributed installations only**: Install Tableau Server on the new computers you want to add to your new Tableau Server cluster. See Install and Configure Additional Nodes for steps.

8. If you have not deactivated your product key on the old computer, do that after you test your new Tableau Server installation and confirm that it's ready for production. For details, see "Tableau Server product key activation" below.

   Note: If you do not have an internet connection, you are prompted to create an off-line activation file to complete the deactivation process. See Activate Tableau Server Offline for steps.

Tableau Server product key activation

You can activate the same Tableau Server product key up to three times. This allows you to test Tableau Server (in a sandbox or QA environments, for example), as well as use Tableau in production. To maximize your activations, you should deactivate your product key when you remove Tableau Server from a computer, unless you will be reinstalling Tableau on the same computer. Doing this gives you the opportunity to use the activation on a different computer. For example, if you move Tableau Server from one computer or VM to another, deactivate the product key, then remove Tableau from the original computer. When you install Tableau on the new computer, you can activate the key there without any conflict. If you use role-based licensing, be sure to activate a Creator or Explorer key or you may lose administrator access to Tableau. If you are removing Tableau Server to reinstall it on the same computer, you don't need to deactivate the key. Tableau will use the key when reinstalled. For example, if you are moving Tableau from one drive on a computer to a different drive on the same computer. For information on how to deactivate a product key, see tsm licenses deactivate.

Migrate Tableau Server from Windows to Linux

Customers running Tableau Server on Windows can migrate to Linux by taking a backup of their existing Tableau installation and restoring it to a fresh installation on Linux. This topic describes the steps necessary to do this migration. You cannot migrate or upgrade from a beta version of Tableau Server to an officially released version.

The basic steps to migrate from Tableau Server on Windows to Tableau Server on Linux include:

1. Step 1: Plan your migration—Plan for your migration, including gathering all the information you'll need to be successful. During this step you should familiarize yourself with potential differences between Tableau Server on Windows and Tableau Server on Linux.

2. Step 2: Create a backup—Create a backup of Tableau Server on Windows.

3. Step 3: Install Tableau Server on Linux and restore the Windows backup—Install a fresh instance of Tableau Server on Linux in a test environment so you can test out the migration, then restore your Windows backup. The restore of your Windows backup will restore the Tableau content (users, projects, sites, workbooks and data sources), but will not restore customizations, so you will need to spend some time configuring Tableau Server on Linux to match the expectations in your organization. If you have a multi-node installation you will need to add nodes and configure them separately.

4. Step 4: Test Tableau Server on Linux—Try Tableau Server on Linux to make sure content is there as you expect, and users are able to perform all the actions they do on Windows. Look specifically at any changes identified as potential differences between Tableau Server on Windows and Tableau Server on Linux. Include key stakeholders in the testing both to leverage their knowledge and to help communicate the upcoming changes.

5. Step 5: Install Tableau Server on Linux in your production environment and restore the Windows backup—Once you're satisfied that Tableau Server on Linux gives you the

functionality you need, install Tableau in your production Linux environment and restore the Windows backup.

Step 1: Plan your migration

A successful migration from Windows to Linux requires some preparation beforehand. You will need to satisfy the following requirements:

- **Identity store**: You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication. To change the identity store from a local identity store to an external identity store, see Changing the Identity Store .
- **Server administrator**: You must have password for at least one of the user accounts with Server Administrator privileges from the Tableau Server source (Windows) deployment. You must use the same user as the Server Administrator in the migrated deployment to Linux.
- **Clean installation of Tableau Server for Linux**: When you install Tableau Server on Linux later in this topic, be sure to use the same identity store type as you are using on Windows, and do not create users or content. When you restore the Windows backup file to the Linux deployment of Tableau Server, all user data and content will be replaced from the Windows back up file.
- **Differences between Windows and Linux**: Review the differences between Tableau Server on Windows and on Linux so you are aware of them and can investigate in your test environment.
    - If you are migrating from a version of Server on Windows prior to 10.5, approach the migration like an upgrade and familiarize yourself with any changes between your existing version and 10.5 by reading What's Changed in the Server on Windows help.

- Any custom fonts you use may need to be installed on your Linux computer, and may render differently there than on Windows.
- Connection options for Linux are a subset of those available for Tableau Server on Windows. Review the connection types that are available and make sure the ones you need are supported.

Step 2: Create a backup

Use TSM to create the backup.

If you are migrating from Tableau Server on Windows:

1. Log on to the computer running Tableau Server on Windows.
2. Open a command prompt as an administrator.
3. Run the following command:

   ```
   tsm maintenance backup -f <filename> -d
   ```

   Include the `-d` flag to include the date in the backup file name.

For more information, see Back up Tableau Server data in the Tableau Server on Windows help.

Step 3: Install Tableau Server on Linux and restore the Windows backup

In a test environment, install Tableau Server on Linux:

- Install Tableau Server for Linux according to the procedure, Install and Configure Tableau Server. Use the same identity store as on your Windows deployment, and do not create users or content.

Restore the Tableau Server on Windows backup:

1. Copy the Windows backup file to the computer running Tableau Server on Linux. By default the restore process will look for the file in this location:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/backups/
```

You can change the location. For more information, see tsm File Paths.

2. Run the following command to stop Tableau Server:

   ```
   tsm stop
   ```

3. Run the following command to restore from the backup file:

   ```
   tsm maintenance restore -f <filename.tsbak>
   ```

   Where `<filename.tsbak>` is the name of your backup file, for example, `tab-server-2017-10-20.tsbak`.

   For more information, see tsm maintenance restore

4. Run the following command to start Tableau Server:

   ```
   tsm start
   ```

After restoring your Tableau content, you may need to configure Tableau Server. For example, if you are migrating from a multi-node installation, you will need to add and configure the additional nodes. You will also need to configure any customizations you made on Windows, including copying over image or logo files if applicable.

Step 4: Test Tableau Server on Linux

Once you've installed Tableau Server on Linux and restored your Windows backup, you can test the new version of Tableau Server. Because the restore process only restores content to Tableau, you may need to update configuration, topology (adding additional nodes for example), and any customizations you have on your Windows installation.

You'll want to be familiar with the known differences between Windows and Linux, as well as any potential problem areas. Test basic functionality, along with any special aspects of server that your organization relies on. For example, there may be key data sources that your organization uses with Tableau. Test these to make sure you're seeing what you expect.

These are some areas of testing to consider:

- **User access**. Confirm that Tableau Server users, including administrators, can sign in. Test your normal user sign in process. Have some of your users participate in the testing to make sure they are able to sign in as expected, and that they can get to the same content that they have access to in your production environment.

- **Viewing built-in administrative views**. With this release of Tableau Server on Linux, you must install PostgreSQL drivers manually, and the administrative views depend on these. Confirm that you have installed the drivers necessary by accessing the built-in administrative views. For more information, see Administrative Views .

- **Data source availability**. Tableau Server on Linux supports a subset of the data sources on Tableau Server on Windows. You need to confirm that the data sources used by your organization are supported on Linux, and install any drivers required. For details on which data sources are supported, see the Tableau Server tech specs. For information about installing drivers, see Database Drivers.

- **Access to file-based data sources on shared drives**. Data sources such as Excel files on network drives will require special actions in order to be accessible from Linux. You or your IT department will need to mount the drives and update any workbooks using these data sources. For more information on connecting to network drives, see the Tableau Community.

- **Viewing published workbooks**. Have users who are familiar with the content try to view published workbooks to make sure they appear as expected. Test views embedded in web pages (for example, in SharePoint pages). Be especially aware of how fonts may differ between Windows and Linux, and some custom fonts may need to be added to your Linux computer, or replaced with other fonts if they are not available on Linux. Dashboard layouts may appear different as well, due to differences in fonts.

- **Server processes**. Sign in to Tableau Server as a server administrator, and then open the Server Status page to confirm that all services and processes are running as expected. Be aware of intentional changes due to changes in version 10.5. For more information, see Tableau Server Data Engine.

- **Publishing workbooks and data sources**. Have users publish workbooks and data sources from Tableau Desktop to make sure this goes as you expect. You may need to install drivers to support the data source connections you are using. See Database Drivers.
- **Subscriptions and extract refreshes**. Manually run some extract refreshes to confirm that they complete successfully. Run some key scheduled extract refreshes to confirm that they complete as expected.
- **Permissions**. Confirm that permissions are still set as expected for users and content.
- **Command-line utilities and APIs**. If applicable, test the command line utilities (tsm and tabcmd) and programmatic access via APIs.

Step 5: Install Tableau Server on Linux in your production environment and restore the Windows backup

When you have completed testing and have identified those areas that require additional changes on your part, or communication to your users, you are ready to install Tableau Server on Linux in your production environment and restore the Windows backup. To do this, follow the same steps described above.

Migrate from Tabadmin to the TSM CLI

The Tableau Services Manager (TSM) command-line interface (CLI) replaces the tabadmin CLI in Tableau Server on Linux, and in Tableau Server on Windows version 2018.2. This page maps tabadmin commands to TSM commands to help you to migrate to the TSM CLI.

To learn more about the TSM CLI, see tsm Command Line Reference.

> Looking for tabadmin commands for Tableau Server on Windows version 2018.1 and earlier? See tabadmin Commands.

Tabadmin commands with a corresponding TSM CLI command

The following table shows which tabadmin commands correspond to commands available in the TSM CLI.

Command Description	Tabadmin Command(s)	Comparable TSM CLI Command
Activate a license	`tabadmin activate --activate`	`tsm licenses activate`
Deactivate licenses	`tabadmin activate --return`	`tsm licenses deactivate`
Create a backup of the data managed by Tableau Server	`tabadmin backup`	`tsm maintenance backup` A backup created using TSM does not include any server configuration data. There is no option to include server configuration data.
Clear the server cache	`tabadmin clearcache`	`tsm maintenance cleanup -r`
Clean up temporary files and old log files	`tabadmin cleanup`	`tsm maintenance cleanup` **Note**: This command was added in version 10.5.1
Update the server configuration with any changes you've made	`tabadmin configure`	`tsm pending-changes apply`
Customize the server name and logos	`tabadmin customize`	`tsm customize`

Enable access to the repository	`tabadmin dbpass`	`tsm data-access repository-access enable`
Disable access to the repository	`tabadmin dbpass --disable`	`tsm data-access repository-access disable`
Set a file store instance to read-only mode	`tabadmin decommission`	`tsm topology filestore decommission`
Delete one or more Web Data Connectors (WDCs) from Tableau Server	`tabadmin delete_web-dataconnector`	`tsm data-access web-data-connectors delete` To learn more, see Web Data Connectors in Tableau Server.

Add a Web Data Connector (WDC) to Tableau Server	`tabadmin import_web-dataconnector`
	
	and
	
	`tabadmin whitelist_web-dataconnector`
		`tsm data-access web-data-connectors add`
		
		Note: TSM does not support importing WDCs, instead it lets you add WDCs to an allowlist. To learn more, see Web Data Connectors in Tableau Server.

List Web Data Connectors (WDCs) used by Tableau Server	`tabadmin list_web-dataconnectors`	`tsm data-access web-data-connectors list` To learn more, see Web Data Connectors in Tableau Server.
Export a site from Tableau Server	`tabadmin exportsite`	`tsm sites export`
Initiate a repository failover	`tabadmin failoverrepository`	`tsm topology failover-repository`

Get a configuration option	`tabadmin get`	`tsm configuration get`
Get the OpenID redirect URL	`tabadmin get_openid_redirect_url`	`tsm authentication openid get-redirect-url`
Import site .csv files into Tableau Server	`tabadmin importsite`	`tsm sites import`
Import a site into Tableau Server using .csv files	`tabadmin importsite_verified`	`tsm sites import-verified`
Display license information for Tableau Server	`tabadmin licenses`	`tsm licenses list` **Note:** For more information about the output of this command, see View Server Licenses.
Move a file store from read-only mode to an active read/write state	`tabadmin recommission`	`tsm topology filestore recommission`
Regenerate internal security tokens	`tabadmin regenerate_internal_tokens`	`tsm security regenerate-internal-tokens`
Register Tableau Server	`tabadmin register`	`tsm register`
Rebuild the search index for Tableau Server	`tabadmin reindex`	`tsm maintenance reindex-search`

Reset the Tableau Server administrator account	`tabadmin reset`	`tsm reset` **Note:** Added in version 2018.1.
Stop and restart all Tableau Server processes	`tabadmin restart`	`tsm restart`
Restore from a Tableau Server backup file	`tabadmin restore`	`tsm maintenance restore` The restore command does not restore any server configuration data. This is true whether you are using a backup created with TSM or a backup created with tabadmin.
Set a configuration option	`tabadmin set`	`tsm configuration set`
Activate or suspend a site	`tabadmin sitestate`	`tsm sites unlock`
Start all Tableau Server processes	`tabadmin start`	`tsm start`
Get the status of Tableau Server and server processes	`tabadmin status`	`tsm status`
Stop all Tableau Server	`tabadmin stop`	`tsm stop`

processes		
Create an archive (.zip) file with Tableau Server log files	`tabadmin ziplogs`	`tsm maintenance ziplogs` The default behavior of the ziplogs command has changed: with tsm, the command collects up to the last two days of log files by default. The tabadmin ziplogs command collected up to seven days of log files. For more information, see tsm maintenance ziplogs.

Tabadmin commands with no corresponding TSM CLI command

The following table lists the tabadmin commands for which a comparable TSM CLI command is not available.

Command Description	Tabadmin Command	Notes
Add or remove a user from the system administrator group	`tabadmin administrator`	You can use the Tableau Server REST API Add User to Group and Remove User from Group methods to add or remove a user from the system administrator group.
Create a new key to encrypt sensitive information stored in the repository	`tabadmin assetkeys`	Use the `tsm security regenerate-internal-tokens` command to create or regenerate secrets and master keys.
Specify whether Tableau Server starts at system	`tabadmin autostart`	Tableau Server returns to the state it was in prior to a system restart. If it was running, it will

start-up time		restart. If it was stopped it will be stopped after the system starts.
Identify a second server node for backup	`tabadmin failoverprimary`	TSM does not have primary nodes, so a TSM equivalent to this command is not needed.
Manage credentials for delegated data access on Tableau Server	`tabadmin manage_global_credentials`	We recommend that you use Kerberos delegation to Apache Impala for global credential management. To learn more, see Kerberos and Enable Kerberos Delegation for Hive/Impala in the Tableau Community.
Reset the password for a Tableau Server account	`tabadmin passwd`	If your server uses local authentication, you can use the Tableau Server REST API Update User method to reset the password for a user account.
Reset binding between Tableau Server user ID and Open ID Connect identity provider	`tabadmin reset_openid_sub`	
Determine whether your environment meets the minimum requirements to run Tableau Server	`tabadmin validate`	

Verify that a backup of the Tableau Server repository will restore successfully	`tabadmin verify_database`	The `tsm maintenance backup` command automatically verifies that a backup will restore correctly unless you use the `--skip-verification` parameter.
Prepare VizQL processes for fast load times after a Tableau Server restart	`tabadmin warmup`	The `tabadmin warmup` command is no longer necessary, as Tableau Server is now optimized to automatically provide fast load times after a server restart.

Migrate Tableau Server from an On-Premises Computer to a VM in the Cloud

You can migrate Tableau Server from a computer in your data center to a virtual machine (VM) in the cloud. As a part of this migration, you'll need to move various Tableau Server data and configuration settings from your on-premises computer to a VM in the cloud where Tableau Server is installed.

Note: This operation includes steps that you may need to perform using the TSM command line.

Prerequisites

Before you migrate Tableau Server from an on-premises computer to a VM in the cloud, be sure to review the following prerequisites:

- Have an account and an environment set up on your cloud provider. For more information, see:
  - Install Tableau Server on Amazon Web Services
  - Install Tableau Server on the Google Cloud Platform
  - Install Tableau Server on Microsoft Azure
  - Install Tableau Server in the Alibaba Cloud
- Read Preparing for Upgrade and Back up Tableau Server data to prepare for the migration.

To migrate Tableau Server to a VM in the cloud

1. Ensure that there are no background tasks running, such as, extract refreshes, workbook publishing, etc. When Tableau Server is idle, note any data source connection types in use so that you can install the appropriate drivers on the new Tableau Server on the VM in the cloud.
2. Stop Tableau Server. At a command prompt, type:

   ```
   tsm stop
   ```

3. After Tableau Server is stopped, make a complete backup, following the steps in Preparing for Upgrade and Back up Tableau Server data. The backup is what you will use to restore your Tableau Server data. You'll use this backup to restore your Tableau Server data on the VM in the cloud.
4. Move the backup off the server and store the backup securely. For example, you can store your backup on cloud storage such as Amazon S3, Google Drive, etc.
5. If you are using SSL on the on-premises server, make a copy of the SSL folder. Be sure to note any security rules, for example, firewall rules, ports, etc. to avoid communication issues between different elements of your infrastructure and your new Tableau Server on the VM in the cloud.
6. Remove any DNS entries using the IP address of the on-premises Tableau Server, so you can reassign them to the Tableau Server on the VM in the cloud.
7. Launch a VM into the cloud that meets the Tableau Server recommendations.

8. Install Tableau Server on the VM in the cloud. For more information, see:
   - Install Tableau Server on Amazon Web Services
   - Install Tableau Server on the Google Cloud Platform
   - Install Tableau Server on Microsoft Azure
   - Install Tableau Server in the Alibaba Cloud
9. After installation is complete, copy any SSL files to Tableau Server on the VM in the cloud, and restore the backup on your new server.
10. Configure Tableau Server on the VM in the cloud and install all SSH and port related stuff.

    Note: If you change the IP address/port of the database, you might need to update all connection information.

11. Copy your backup file *.tsbak* to the folder location specified by the `base-filepath.backuprestore` variable. By default this is: `/var/-opt/tableau/tableau_server/data/tabsvc/files/backups/`.

    Note: You can change the location for the backup file. For more information, see tsm File Paths.

12. Next, stop Tableau Server. At a command prompt, type:

    ```
    tsm stop
    ```

13. Restore your in-production data without configuration information to your new Tableau Server installation. At a command prompt, type:

    ```
    tsm maintenance restore -f <filename>
    ```

    where `<filename>` is the name of the *.tsbak* file you copied in step 11.

For more information about restoring Tableau Server data, see Restore from a Backup.

14. Start the server. At a command prompt, type:

```
tsm start
```

15. **Distributed installations only**: Install Tableau Server on the new VMs you want to add to your new Tableau Server cluster in the cloud. For more information, see:

    - Install and Configure Additional Nodes
    - Self-Deploy Tableau Server on AWS in a Distributed Environment
    - Self-Deploy Tableau Server on the Google Cloud Platform in a Distributed Environment
    - Self-Deploy Tableau Server on Microsoft Azure in a Distributed Environment
    - Self-Deploy Tableau Server on Alibaba Cloud in a Distributed Environment

16. Test your Tableau Server on your VM in the cloud to ensure that it works as expected. If your Tableau Server is working fine in the cloud, you can deactivate your on-premises Tableau Server product key and use this same product key to activate your Tableau Server on the VM in the cloud.

    Note: If you do not have an internet connection, you are prompted to create an offline activation file to complete the deactivation process. For more information, see Deactivate Tableau Server Offline.

17. The same Tableau Server product key can be activated three times: once for a production environment, once for a test environment, and once for a QA environment. After you have tested your new Tableau Server installation and confirmed that it's ready for production, you must deactivate your earlier production version of Tableau Server, and then you must uninstall it. To deactivate the earlier version, see tsm licenses deactivate.

Note: If you do not have an internet connection, you are prompted to create an off-line activation file to complete the deactivation process. See Deactivate Tableau Server Offline for steps.

Changing the Identity Store

Infrastructure or business changes may require you to change the identity store on Tableau Server. There are two kinds of identity stores: local and external. When you installed Tableau Server you configured either a local identity store or an external identity store.

When you configure Tableau Server with a local identity store, all user and group information is stored and managed in the Tableau Server repository. In the local identity store scenario, there is no external source for users and groups.

When you configure Tableau Server with an external store, all user and group information is stored and managed by an external directory service. Tableau Server must synchronize with the external identity store so that local copies of the users and groups exist in the Tableau Server repository, but the external identity store is the authoritative source for all user and group data. Examples of external identity stores are OpenLDAP and Active Directory.

For more information about the Tableau identity store, see Identity Store.

You can change from local store to an external store, or you can change from an external store to a local store. In either case, to change the identity store type, you complete these steps:

1. Uninstall and then reinstall of Tableau Server. The procedure for full uninstall and clean install are at the end of this topic.
2. Restore content and permissions.

   In these steps the term "restore" does not refer to using the `TSM maintenance restore` command to restore the backup you are making. You cannot restore a backup (`.tsbak`) created on a Tableau Server instance that uses a different identity

store than the target Tableau Server. The backup is a best practice safeguard, in case you need to go back to your original Tableau Server configuration.

Warning

Changing the installation type on Tableau Server can be a complicated and time-consuming process. To avoid data loss or orphaning of content or users, you'll need to plan this process carefully. In all cases, user filters applied to workbooks and data sources will need to be updated manually after the change.

Most importantly, determine how you will transition content and permissions to the new identity store after you reinstall Tableau Server.

Methods for restoring content and permissions

The following list describes two methods for restoring content and permissions after you reinstall Tableau Server. Select the method that best fits with your environmental requirements.

- **Method 1: Use site export and import**—In this method, you start by exporting each site in your existing deployment. Then, you install the new server and configure it for the new identity store type. You then create new users in the default site on the new server. Finally, you import all the original sites. During the import stage, you can map the original identities to the new users that you created in the default site.

  Note: When migrating sites between instances of Tableau Server, the target site must be on a version of Tableau Server that is the equal to or later than the version of Tableau Server for the source site. Both the source and target sites must be from supported versions of Tableau Server.

  Because this method exports all content and permissions at each site, it is the best method for organizations that require a high fidelity replica of the content and

permissions after the identity store change is complete. Some organizations require an identity store change as the result of an authentication change. In these cases, a different user name syntax is a often a requirement in the new model. This method, which includes a process of mapping original user names to new names, provides flexibility for such scenarios.

- **Method 2: Fresh installation; users republish content**—In this method, you install a new version of Tableau Server and select the new identity store type during setup. You also create new sites. You then create users and give them access, and they republish their workbooks and data sources. Unlike the other method, in this one, you do not reuse any of your existing Tableau Server infrastructure.

  This method is most appropriate for smaller deployments with fairly autonomous and data savvy users. From an administrative perspective, this method is the simplest, since you're not actively porting over content. However, because you rely entirely on users to republish content, this method may not be successful for large organizations or for those where centralized oversight of content is required.

User filters

User filters are domain-specific. Therefore, when the domain of Tableau Server changes or authentication type changes, filters no longer function as expected. Although the user filters are generated by Tableau Server, after they are set by the user, the filters are stored in the workbooks and data sources. Neither of these methods for changing the identity store modifies the contents of the workbooks or data sources.

As you plan the identity store change, you must also include a final task to correct user filtering in all workbooks and data sources with Tableau Desktop.

User names and the Tableau Identity store

If you are using Method 1, it's helpful to understand how Tableau Server stores user names in the Tableau identity store. Tableau stores all user identities in the repository, which coordinates content permissions and site membership with various services in Tableau Server. Gen-

erally, an identity store configured for Active Directory store user names in the format, `domain\username`. Some organizations use a UPN (`jsmith@domain.lan`).

On the other hand, organizations that configure Tableau Server with local identity store usually create standard, truncated user names, such as `jsmith`.

In all cases, these user names are literal strings that must be unique in the Tableau identity store. If you are changing from one identity store type to another, then your target authentication, SSO, or user provisioning solution may require a specific user name format.

Therefore, to maintain all permissions, content, and user viability, one of the following must be true after you change the identity store type:

- The new user names must match the original user names, or
- The original user names must be updated to match a new format.

If an authentication change is driving the identity store change, then the target authentication scheme will likely impose a user name syntax that is different than your original user names. Method 1 includes a process where you can map original user names to new user names.

It's possible that the original user name format will work with the new authentication type. For example, if you used UPN names in a local identity store deployment, you might be able to use the same user names in an Active Directory deployment. You could also use the `domain\username` format for local identity store, as long as users continue to use that format to sign in to Tableau Server.

If you are changing from local identity store to an external Active Directory store, review the topic, User Management in Deployments with External Identity Stores, as part of your planning process.

Method 1: Use site export and import

You must use the same version of Tableau Server for the export and import operations.

1. Export all sites on your server. See Export or Import a Site.
2. Back up, remove, and then reinstall .

3. Create new users on Tableau Server. You should have a new user that corresponds to each user on the original server.
4. Import the sites that you exported in Step 1. See Export or Import a Site. During import, you will be prompted to map the new users to the original users.

Method 2: Fresh installation–users republish content

Even if you do not plan to port content as part of your identity store change, we recommend that you back up the server.

1. Back up, remove, and then reinstall .
2. Create users, sites, and groups.
3. Inform your users of the new Tableau Server, provide them with credentials, and allow them to republish their content.

Back up, remove, and then reinstall

Both methods include the following steps:

1. Back up Tableau Server
2. Remove Tableau Server.
3. Reinstall Tableau Server with the new identity store type.

Step 1: Back up Tableau Server

As a best practice, you should back up the server before proceeding.

Follow the procedure, Create a backup using the TSM command line interface (CLI). Run the `backup` command with the `-d` option. The `-d` option adds the datestamp.

When you are finished, copy the backup file (`.tsbak`) to a safe location that is not a part of your Tableau Server installation.

Step 2: Remove Tableau Server

You must completely remove Tableau Server from the computer. See Remove Tableau Server from Your Computer.

Step 3: Reinstall Tableau Server with new authentication type

1. Go to the Tableau Customer Portal, sign in with your Tableau user name and password, and then download Tableau Server.
2. Install Tableau Server. See Install and Configure Tableau Server more information. During installation, you will select the new identity store type. See Configure Initial Node Settings.

Manage Individual Sites

In addition to planning your sites in Tableau, you can manage users and groups, manage projects and control content access, manage your site data, and interact with views on the web.

What is a site

You might be used to using the term *site* to mean "a collection of connected computers," or perhaps as the short form of "website." In Tableau-speak, we use site to mean a collection of users, groups, and content (workbooks, data sources) that's walled off from any other groups and content on the same instance of Tableau Server. Another way to say this is that Tableau Server supports multi-tenancy by allowing server administrators to create sites on the server for multiple sets of users and content.

All server content is published, accessed, and managed on a per-site basis. Each site has its own URL and its own set of users (although each server user can be added to multiple sites). Each site's content (projects, workbooks, and data sources) is completely segregated from content on other sites.

If you are a server administrator on your Tableau Server deployment, you can learn more about sites, when to use them (vs. projects), and more in Sites Overview, in the **Manage Server** section.

Note: This article pertains to configuring sites on Tableau Server deployments. For Tableau Cloud, see Site Administrator Role and Tasks.

Site administrator tasks

Where the Server Administrator site role gives a user unrestricted access to the entire Tableau Server deployment, the Site Administrator site roles give a user unrestricted or minimally restricted access at the site level. The differences between Site Administrator Creator and Site Administrator Explorer are in the level of data connection and publishing access. Both site roles allow administering the site itself and managing site users. For more information, see Set Users' Site Roles.

Although a server administrator can work at both the server and site levels, we make a distinction between the two levels of task. The site administrator is typically in charge of creating and maintaining the framework that enables Tableau users in the organization to publish, share, manage, and connect to data sources and workbooks. In this vein, site administrator tasks include any of the following (and both site roles allow this level of access):

- Creating project hierarchies to organize the site's data sources and workbooks.

  This can include delegating project-level management to project leaders.

- Creating groups and assigning permissions that allow users to access only the content they need.

- Adding and removing users, assigning their site roles.

  This is allowed by default on a site; however, a server administrator can restrict this access to the server level only.

- Managing the site's extract and subscription schedules.

- Monitoring site activity.

For more information about the distinction between server administrator and site administrator, see Administrator-level access to sites, in the **Manage Server** section.

Steps for setting up your site

The table below shows a loose sequence of steps for setting up a site, along with links to topics where you can get more information. You can complete the steps in any order that makes sense for you.

However, before you perform the steps to configure the site, we recommend spending some time with the articles in this section, learning about site authentication, site roles, projects, and permissions. Ideally you would document a plan for your projects, groups, and overall permissions strategy. Then set up a few projects and add a preliminary set of users, to test the plan and resolve issues before you add the remaining users. You can change many site settings after your users are working with the site, but try to go in with the intention of minimizing post-production changes.

Plan	To supplement the recommendations above this table, get an overview of how the site components work together in Planning a Site.
Configure access	Work with the server administrator to determine how users sign in to the site, and configure the site appropriately. For example, if the server is configured for single sign-on using SAML, you might configure SAML authentication at the site level as well.
Create projects and the permissions structure	Projects help you organize content, delegate project-level content management, and manage permissions effectively. To get started, see Use Projects to Manage Content Access.

Add users	Determine the users who can sign in to the site. See Add Users to a Site.
Get your data to Tableau Server	After you create your projects and permissions structure, designate approved users for publishing and managing vetted data sources to the appropriate projects on the site. In some organizations, people serve in multiple Tableau roles. Site administrators commonly also are data stewards. By that, we mean they create, publish, and manage the Tableau data connections. If this is you, make sure you are assigned the Site Administrator Creator site role. After content is published to the site, you can maintain connection information (credentials, access tokens) and refresh schedules. For more information, see Refresh Data on a Schedule.
Analyze site usage and performance	Monitor usage, performance, and other metrics. See Administrative Views .

Planning a Site

Before you add users and content to a site, we recommend that you plan the following aspects of the site.

- Projects
- Users and groups

- Site roles and permissions
- Extract refresh schedules

The subsequent sections go over these site components, assuming that you are familiar with

> **Note:** This article and section apply only to self-managed Tableau Server deployments on-premises or in the cloud. If you use Tableau Cloud, see Manage Content Access.

Projects

You can create projects on a site, which act as containers in which you can organize related content assets (such as data sources and workbooks). For example, you might set up a project to contain all of the certified data sources and workbooks your organization uses for mission-critical decisions. Or you might set up projects by department.

Projects are also useful for managing permissions. Once you know how your users need to access content, it's usually easier to create projects based on those the type of content, and maintain permissions at the project level.

Every site has a default project named **Default**. When you create projects, the new projects get their initial set of permissions from the default project. In effect, the default project is a template for new projects. As we explain in related articles, for most environments, we recommend that you use the Default project only as a permissions template, and not as a container for published content.

For more information, see Use Projects to Manage Content Access.

Users and groups

Any user who will publish content or access published content on a site must be able to sign in to the site. If the user already has an account on the server, you'll need to add that user to the appropriate site. You can add a user to more than one site as well. If the user doesn't already

exist, you need to create a user account. Either way, make a list of the users who will need to be able to sign in to each site.

> **Note:** The server license might restrict how many users you can have, or what level of access they can have. Check with the server administrator to make sure that you'll be able to have an account for all your users.

In general, we recommend that you create groups on the server and then add users to the groups. This helps to make permissions much easier to manage. You can assign permissions on groups, to give those permissions to all users in the group. (See the next section.)

A typical strategy is to create groups for users who use content in similar ways. For example, you might create a group named SalesWBPublishers for all the users in the Sales department who publish workbooks, and a separate group named SalesDSPublishers for people in Sales who publish data sources. Each of these sets of users needs its own set of capabilities, so it makes sense to have a group for each for these needs.

Site roles and permissions

Each user has a *site role* that determines the maximum permissions that they can have on the site. As part of your site planning, you need to decide each user's site role. A user with a site role that's too restrictive might not be able to do the work they need. By the same token, a security best practice is to limit users' capabilities to only those that they need to do their work. This is referred to as following the principle of *least privilege*.

You or a site administrator you delegate this task to must also determine the permissions a user needs to work with content. Each content asset (workbook, data source, project) supports a set of *capabilities*. For example, you can **View** or **Add Comments** to a workbook. Before a user can perform tasks on a workbook, their permissions must allow those capabilities. A recommended practice is to sketch out a mapping of permissions to users outside of Tableau before you try to set this up on the server.

Permissions determine what a user can do *within the context of the site role*. A user whose site role is **Viewer** can never publish to the site, regardless of the permissions you grant them. A user whose site role is **Creator** can publish a workbook to the site, but only if that user has permission to save and view workbooks.

Extract refresh schedules

If users publish data sources or workbooks that include extracts, you usually want to make sure that the extracts are refreshed, so that they contain the latest data. Users can manually refresh an extract, but this isn't always a good idea if the extract is large, and the refresh takes a long time. Instead, you can set up schedules for when an extract should be refreshed. Another planning task for a site administrator, therefore, is to think about when extracts should be refreshed and to work out schedules.

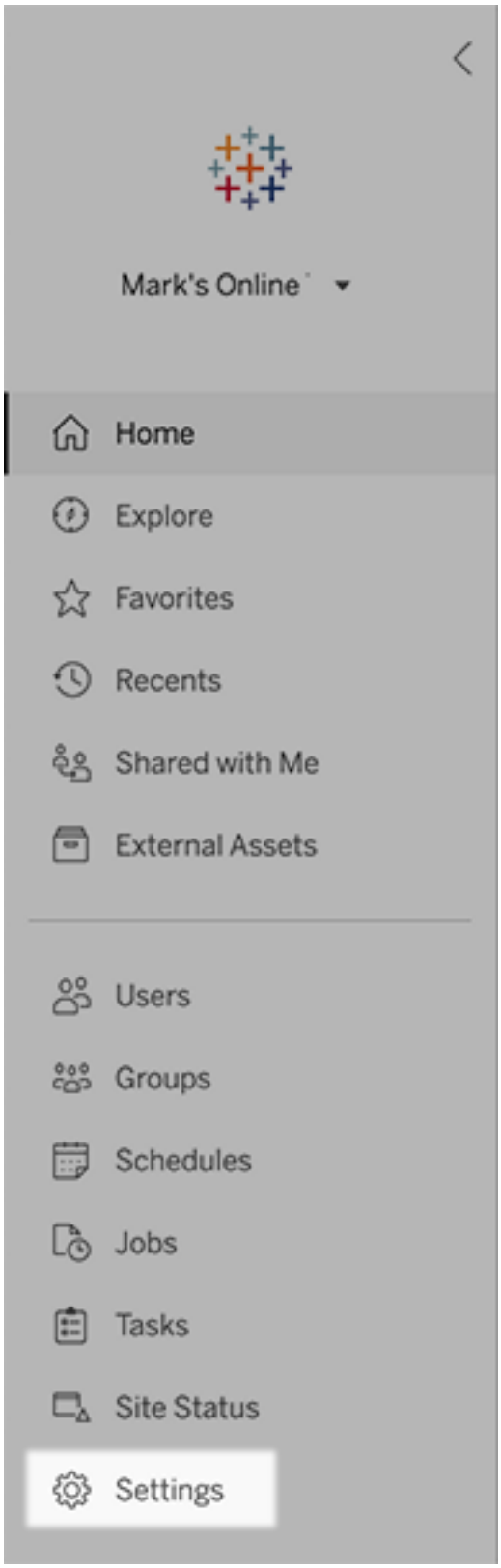
Site Settings Reference

Customize a site for your organization using the settings below. To view and edit site settings, you must be a Site Administrator on Tableau Cloud or Server Administrator on Tableau Server.

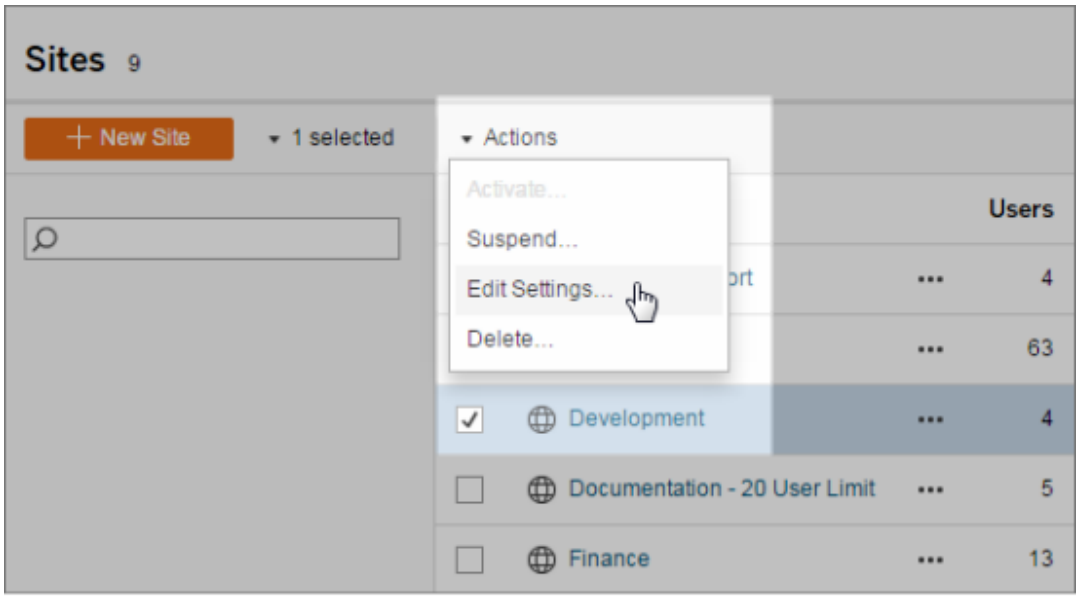
Site settings are displayed differently in Tableau Cloud and Tableau Server. The settings available to you depend on your site configuration and whether you use Tableau Cloud or Tableau Server. To easily find a specific setting below, press Ctrl+F (Windows) or Command+F (macOS) and search for the setting.

Accessing site settings

- (Tableau Cloud) From the Home page, expand the side pane, and click **Settings** at bottom.


Mark's Online
Home
Explore
Favorites
Recents
Shared with Me
External Assets
Users
Groups
Schedules
Jobs
Tasks
Site Status
Settings

- (Tableau Server) If you're editing an existing site, select it on the Sites page, and then select **Edit Settings**. Or, in a single-site deployment, click **Settings** at the bottom of the side pane.



General tab

Setting	Description
Site Name and ID (Tableau Server Administrators only)	Specifies the site name seen in the user interface and the ID seen in the site URL. (If you are editing the Default site, you cannot change the ID.) You can't change the "#/site" portion of the URL (for example, http://-localhost/#/site/sales). In multi-site server environments, these segments appear in the URL for sites other than the Default site.
Storage (Tableau Server Administrators only)	Select either **Server Limit** or **GB**, and for the latter enter the number of gigabytes you want as a limit for storage space for published workbooks, extracts, and other data sources. If you set a server limit and the site exceeds it, publishers will be pre-

	vented from uploading new content until the site is under the limit again. Server administrators can track where the site is relative to its limit using the Max Storage and Storage Used columns on the Sites page.
Revision History (Tableau Server Administrators only)	Specifies the number of previous versions of workbooks, flows, and data sources that are stored on the server.
Tableau Prep Conductor	Controls whether users with appropriate permissions can schedule and monitor flows. Tableau Prep Conductor is part of Data Management. For more information, see Tableau Prep Conductor.
Web Authoring	Controls whether browser-based authoring is enabled for the site. When web authoring for workbooks is disabled, users can't create or edit published workbooks from the server web environment but instead must use Tableau Desktop to republish the workbook. When web authoring for flows is disabled, users can't create or edit published flows from the server web environment but instead must use Tableau Prep Builder to re-publish the flow. For more information, see Set a Site's Web Authoring Access and Functions in Tableau Cloud Help.
Managing Users (Tableau Server Administrators only)	Determines whether only server administrators can add and remove users and change their site roles, or whether site administrators can too. If you allow site administrators to manage users, specify how many users they can add to the site by selecting one of the following: • Server Limit adds the number of available server seat licenses. For a server with core-based licensing, there is no limit. • Site Limit lets site administrators add users up to a limit you specify. • Site Role Limit lets site administrators add users of each site

	role up to the license limit you specify for the site. For more information, see View Server Licenses.
Guest Access (Tableau Server Administrators only)	Lets people who lack a Tableau Server account see views that have guest access permissions. **Note:** If you use Tableau Server, your administrator can disable Guest Access.
Tableau Catalog	Turns off Catalog capabilities when Tableau Server or a Tableau Cloud site is licensed with Data Management. For more information, see Disable Catalog.
Workbook Performance after a Scheduled Refresh (Tableau Server Administrators only)	Pre-computes recently viewed workbooks with scheduled refreshes to open them faster. For more information, see Configure Workbook Performance after a Scheduled Refresh.
Workbook Performance Metrics (Tableau Server Administrators only)	Lets site users collect metrics on how workbooks perform, such as how quickly they load To initiate recording, users must add a parameter to the workbook's URL. For more information, see Create a Performance Recording.
Managed Keychain Clean Up (Tableau Server Administrators only)	Lets site administrators manage saved credential keychains for OAuth connections on the site. For more information, see OAuth Connections.
Automatically Suspend Extract Refresh Tasks	To save resources, Tableau can automatically suspend extract refresh tasks for inactive workbooks. This feature applies only to refresh schedules that run weekly or more often. For more information, see Automatically Suspend Extract Refreshes for Inactive Workbooks in Tableau Cloud Help.

Linked Tasks (Tableau Server and Site Administrators only)	Lets Server administrators enable users to schedule flow tasks to run one after the other. They can also enable users to trigger the scheduled flow tasks to run using **Run Now**. This setting can be applied at the server level to include all sites on Tableau Server. The setting can be disabled at the site level to include only specific sites. If the setting is turned off after linked tasks are scheduled, any tasks that are running will complete and the scheduled linked tasks are hidden and no longer show on the **Scheduled Tasks** tab. For more information, see Schedule Linked Tasks.
Email Settings (Tableau Server Administrators only)	Specifies the From address and message footer seen in automatic emails for alerts and subscriptions.
Site Invite Notification (Tableau Cloud only)	For sites with single-sign-on authentication, sends an invite email when new users are added to the site.
Site Logo (Tableau Cloud only)	Specifies the image that appears with the site name.
Start Page	Controls which site page appears when users sign in. By default, the Home page appears, but you can instead open All Projects, All Workbooks, or other pages. For more information, see Set the Default Start Page in Tableau Cloud Help. **Note:** If you use Tableau Server, your administrator can override this site setting.
Tableau Pulse Deployment	Controls whether Tableau Pulse is available for all users, a group of users, or no users. For more information, see Set Up Your Site for

	Tableau Pulse.
AI in Tableau (Tableau Cloud only)	Controls whether generative AI functionality is enabled for Tableau features. For example, Tableau Pulse can use generative AI to summarize key metric insights using natural language so they are easier to understand. Some generative AI features require Tableau+ and a connection to a Salesforce org with Einstein generative AI set up. For more information about how to turn on AI in Tableau features, see Turn On AI in Tableau for Your Site. To learn more about Tableau AI, see AI in Tableau.
Personalized Insight Ranking (Tableau Cloud only)	Controls whether users can provide thumbs-up or thumbs-down feedback on individual insights. When this setting is turned on and users provide feedback, that feedback is used by the Tableau Pulse insights platform to further personalize and rank the types of insights it shows to a user. This setting is independent of the setting to deploy Tableau Pulse. When Personalized Insight Ranking is turned off, users won't be able to submit thumbs-up or thumbs-down feedback on individual insights. For more information, see "Turn off Personalized Insight Ranking" in Set Up Your Site for Tableau Pulse.
User Visibility	Controls what user and group names are visible to other users. For more information, see Manage User Visibility in Tableau Cloud Help.
Availability of Ask Data	Controls whether Ask Data lenses are enabled or disabled for data sources. Ask Data lets users query data using conversational language and automatically see visualizations. For more information, see Automatically Build Views with Ask Data in Tableau user Help.

Availability of Explain Data	Controls whether site users with the appropriate permissions can run Explain Data and authors can access Explain Data Settings. For more information, see Control Access to Explain Data. To learn more about Explain Data, see Discover Insights Faster with Explain Data.
Automatic Access to Metadata about Databases and Tables	Automatically grants users certain capabilities to external assets using derived permissions. For more information, see Turn off derived permissions in Tableau Cloud Help.
Sensitive Lineage Data	Specifies whether sensitive lineage data should be obfuscated or filtered when users don't have the appropriate permissions to related metadata. For more information, see Sensitive lineage data.
Cross-Database Joins	Determines where the join process happens when joining data from multiple sources. For more information, see Combine Tables from Different Databases in Tableau user Help.
Extract Encryption at Rest (Tableau Server Administrators only)	Lets you encrypt .hyper extracts while they are stored on Tableau Server. Server administrators can enforce encryption of all extracts on their site or allow users to encrypt all extracts associated with particular published workbooks or data sources. For more information, see Extract Encryption at Rest.
Tableau Support Access (Tableau Cloud only)	Allows Tableau Support technicians access to the site to help troubleshoot support cases. By default, this feature is disabled. For more information, see Enable Support Access.
Sharing	Allows users to share items directly with other users. When an item is shared, the recipients get a notification and the item is added to their Shared with Me page. If this is not enabled, users can only copy a link to share. For more information, see Share Web Content in Tableau user Help.
Comments	Controls whether users can add remarks in a Comments side pane for each view and @mention other Tableau users to notify them via

	email. For more information, see Comment on Views in Tableau user Help.
Data-Driven Alerts	Lets users automatically receive emails when data reaches key thresholds. For more information, see Send Data-Driven Alerts in Tableau user Help.
Subscriptions	Lets site users subscribe to views and receive regular emails of them. On Tableau Server, these options are available only if you first configure subscription settings.
High-Visibility Data Labels in View and Workbook Subscriptions	Controls whether subscriptions include relevant upstream high visibility data quality warnings and sensitivity labels in the email. On Tableau Server, these options are available only if you first turn on and configure subscriptions. For more information on data quality warnings, see Set a Data Quality Warning. For more information on sensitivity labels, see Sensitivity Labels. Previously titled **Data Quality Warnings in Subscriptions**. **Note:** Data quality warnings and sensitivity labels are a feature of Tableau Catalog, which is part of Data Management.
Tagging	Specifies the number of tags that users can add to items. The default limit is 50 tags, and the maximum is 200. For more information, see Use Tags.
Recommendations for Views	Controls whether recommendations show on the site and whether the names of users who have looked at recommended items show on recommendation tooltips. **Note:** If you use Tableau Server, your administrator can disable Recommendations.

Request Access	Lets users send access requests to content or project owners. For more information, see Let Site Users Request Access to Content in Tableau Cloud Help.
Metrics Content Type	Controls whether metrics are available on the site. When you turn metrics on, users can create metrics from views and metrics appear as a content type. When turned off, metrics won't appear on the site or continue to sync. If you turn on metrics again, pre-existing metrics will reappear and resume refreshing. For more information, see "Set Up for Metrics" in Tableau Cloud Help or Tableau Server Help. **Retirement of the legacy metrics feature** Tableau's legacy metrics feature was retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2. In October 2023, Tableau retired the ability to embed legacy metrics in Tableau Cloud and in Tableau Server version 2023.3. With Tableau Pulse, we've developed an improved experience to track metrics and ask questions of your data. For more information, see Create Metrics with Tableau Pulse to learn about the new experience and Create and Troubleshoot Metrics (Retired) for the retired feature.
Web Page Objects and Web Images	Controls whether these Web Page and Image objects can display target URLs. For more information, see Security for Web Page objects in Tableau user Help.
Personal Space	Allows Creator and Explorer site users to create and save content to a private Personal Space. When Personal Space is turned on, you can set user storage limits. For more information, see Create and Edit Private Content in Personal Space.
Collections	Controls whether collections are available on the site. When you turn on collections, users can create collections to organize content and browse collections made available by other users. For more information, see Organize Items in a Collection.

Site Time Zone for Extracts	The default time zone for extract-based data sources in a site is Coordinated Universal Time (UTC). Site administrators can set a different time zone. For more information, see Set the Site Time Zone for Extracts in Tableau Server Help or Set the Site Time Zone for Extracts in Tableau Cloud Help.
Extract Quota Limit Notifications	Sends email alerts to all site administrators when extract refresh jobs are canceled because of extract job capacity issues.
Flow Parameters (Tableau Server and Site Administrators only)	Enables users to schedule and run flows that include parameters. Administrators can also enable flow parameters to accept any value. If this option is enabled, any flow user can enter any value in a parameter, potentially exposing data that the user should not have access to. Parameters can be entered in an input step for file name and path, table name, or when using custom SQL queries, in an output step for file name and path and table name, and in any step type for filters or calculated values. Flow parameter settings can be applied at the server level to include all sites on Tableau Server. The settings can be disabled at the site level to include only specific sites. For more information about using parameters, see Create and Use Parameters in Flows in the Tableau Prep help.
Run Now	Controls who can run jobs manually using the Run Now option from the web, Rest API, and Tabcmd. By default, this option is selected to allow users to run jobs manually. Clear the check box if only administrators should be allowed to run jobs manually. **Note:** If you use Tableau Server, your administrator can dis-

	able this site setting.
Manage Notifications	Controls how site users can receive notifications for events such as extract jobs, flow runs, when another user shares content with them or mentions them in a comment. Notifications can be seen in their Tableau site via the notification center, sent by email, or sent to a Slack workspace. When a notification is enabled, users can configure their notification preferences on their Account Settings page. **Note:** If you use Tableau Server, your server administrator can disable this site setting.
Customize Email Notifications (Tableau Cloud only)	Controls whether email notifications for data-driven alerts and subscriptions to workbooks and views are sent using the Tableau email server or your own SMTP server. When you use your own SMTP server, you can customize the email sender's name as well as the domain used in the sender's email address and the domain for links in the notifications emails. Added for Tableau Cloud in February 2024. Separately from the SMTP settings, you can control whether the email notifications sent to users for data-driven alerts and subscriptions include links. These links direct users to your Tableau site to see the content and manage the alert or subscription. A link to unsubscribe is always included in notification emails, regardless of whether this setting is on. Added for Tableau Cloud in June 2023.
Flow Subscriptions	Controls whether flow owners can schedule and send emails with flow output data to themselves and others. When you allow flow subscriptions, you can control whether flow output data is included in the subscription email and whether flow output files are attached

	to the email. For more information, see Notify Users of Successful Flow Runs
OAuth Clients Registry	For a subset of connectors, you can register a custom OAuth client for the site to override an OAuth client that has been configured for the server. By registering a custom OAuth client, you enable new and existing connections to use the site-level OAuth client instead of the server-wide OAuth client. For more information, see Configure Custom OAuth.
View Acceleration	Controls whether Creator and Explorer site users can accelerate the views in their workbooks for faster loading times. When you allow view acceleration, you can set a maximum number of views to be accelerated, and you can choose to automatically suspend acceleration for views that repeatedly fail the acceleration task. For more information, see View Acceleration.
Assertions for Group Membership	Enables local group membership to be controlled and managed by your SSO IdP or through a Tableau connected app by dynamically asserting group membership when a user authenticates to Tableau Server. Requires additional configuration in the SAML assertion, OIDC assertion, or JSON web token (JWT). For more information, see Dynamic group membership using assertions. **Important**: This site-level setting can be enabled only when the server-wide setting has been enabled first.
Group Sets	Enables the **Group Sets** page and the ability to create group sets. Group sets can be used by certain users (server admins,site admins, project owners, and content owners) to apply permission rules that require users to be members of all groups in the group set to access content whose permissions are dependent on the group set. For more information, see Work with Group Sets.

Authentication tab (Tableau Cloud)

Setting	Description
Authentication Types	Specifies how users can sign in to the site, and how they access it after signing in the first time. Authentication verifies a user's identity. For more information, see Authentication.
Default Authentication Type for Embedded Views	Specifies how users can sign in to embedded views. By default, Tableau authentication is selected.
Control User Access in Authentication Workflows	Enables user attribute functions used in embedded content to accept the passing of user attributes from a JSON Web Token (JWT). The user attributes are passed to Tableau to customize and control the data that can be shown to a user at runtime. For more information, see Embedding API v3 Help.
Automatic Provisioning and Group Synchronization (SCIM)	Allows you to manage users on the site through a third-party identity provider (IdP). When enabled, the Base URL and Secret boxes are populated with values to use in the IdP SCIM configuration. For more information, see Automate User Provisioning and Group Synchronization through an External Identity Provider.
Connected Clients	Allows Tableau clients such as Tableau Mobile, Tableau Bridge, and others to stay authenticated to the server after a user provides sign-in credentials the first time. When turned off, users are required to sign in explicitly each time they visit Tableau Cloud. For more information, see Access Sites from Connected Clients.

Bridge tab (Tableau Cloud)

Setting	Description
Client Not Running Notifications	Sends email alerts to data source owners when a client appears to be disconnected from the site.

Pooling	Distributes live queries and refresh jobs across all clients in Bridge pools. For more information, see Configure and Manage the Bridge Client Pool in the Tableau Cloud Help.
Private Network Allowlist	Add and manage domains that enable dedicated Bridge pool access to private network data on behalf of Tableau Cloud.

Extensions tab

Setting	Description
Dashboard and Viz Extensions	Manage and control dashboard and viz extensions. Dashboard extensions are web applications that run in custom dashboard zones and can interact with the rest of the dashboard. Viz extensions are web applications that support new viz types. For more information, see "Manage Dashboard and Viz Extensions" in Tableau Cloud Help or Tableau Server Help.
Analytics Extensions	Enables a set of functions that your users can use to pass expressions to analytics extensions for integration with R and Python. For more information, see "Configure Connection with Analytics Extensions" in Tableau Cloud Help or Tableau Server Help.
Tableau Prep Extension	When authoring flows on the web, enables users to apply Einstein Discovery-powered models to their flows to bulk score predictions for their data. For more information, see Configure Einstein Discovery Integration in the Tableau Server help.

Integrations tab

Setting	Description
Slack Connectivity	Displays connections between a Slack workspace and the Tableau site. When connected, Tableau site users can see

	their Tableau notifications in the connected Slack workspace. **Note**: In Tableau Server, a Slack administrator must create a private Slack app and install it to a Slack workspace before a Tableau server administrator can add an OAuth Client and connect to Slack. In Tableau Server, you can add OAuth client information for a private Slack application, then select Connect to Slack to finalize the connection. For more information, see Integrate Tableau with a Slack Workspace.
Analytics Extensions	Enables a set of functions that your users can use to pass expressions to analytics extensions for integration with R and Python. For more information, see "Configure Connection with Analytics Extensions" in Tableau Cloud Help or Tableau Server Help.
Publish to Salesforce (Beta on Tableau Cloud and Tableau Server)	Allows site users to publish views to a Salesforce app. When a view is published to Salesforce, anyone with access to the selected app can see that the content exists. However, only those signed in with existing Tableau permissions can see the view. For more information, see Publish Views to Salesforce (Beta).

Connected Apps tab

Setting	Description
Connected Apps	Create and manage Tableau connected apps, or explicit direct trust or OAuth 2.0 trust relationships between

	Tableau Server (server-wide and site-level) and custom applications and programmatically authorize access to the Tableau REST API on users' behalf using JSON web tokens (JWTs). For more information, see Use Tableau Connected Apps for Application Integration in the Tableau Cloud Help.

Mobile tab

Setting	Description
App Lock	Requires a biometric method or device passcode for users to open this site on Tableau Mobile. For more information, see Enable App Lock for Added Security in the Tableau Mobile Deployment Guide.
Offline Previews	Controls whether offline previews are generated for display when users access the site on Tableau Mobile. For more information, see Manage Tableau Mobile Data on Devices in the Tableau Mobile Deployment Guide.
Mobile Security Policies	Some security policies are enabled automatically and cannot be disabled. Mobile security policies are not available for MAM versions of Tableau Mobile.
Jailbreak Detection	Controls whether a Tableau Mobile app user with a device that has been "jailbroken" or "rooted" is allowed to access content on Tableau, and what level of response occurs when a jailbroken or rooted device is detected. For more information, see Tableau Mobile App Security Settings.
Malware Detection (Android only)	Controls whether malware detection is enabled for mobile devices, and what level of response occurs when malware is detected. For more information, see Tableau Mobile App

	Security Settings.
Maximum Days Offline Without Policy Refresh	Controls whether there is a maximum number of days a mobile device can be offline and still use the app. For more information, see Tableau Mobile App Security Settings.
Prevent Debugging	Controls whether debuggers are prevented on mobile devices. For more information, see Tableau Mobile App Security Settings.
Screen Sharing and Screenshots (Android only)	Controls whether a Tableau Mobile user is able to take screenshots or use screen sharing while in the app. For more information, see Tableau Mobile App Security Settings.

Manage Users and Groups

You can add users to your Tableau sites and set their site roles, which determines each user's level of access. In addition, you can create groups of users, and enable guest access to your sites.

Add Users to a Site

Everyone who needs to access Tableau Server—whether to browse, publish, edit content or administer the site—must be added as a user. Administrators have the following options for adding users:

- Add a local user account or a user account from Active Directory, described later in this article.

  You can also add users by importing an Active Directory group. See Create Groups via Active Directory.

  For Tableau Server on Linux, all external directory communication is configured and managed with a LDAP identity store. In the context of user and group synchronization,

Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

- Import Users via a CSV file that you create using the CSV Import File Guidelines.

Site administrator access to user management

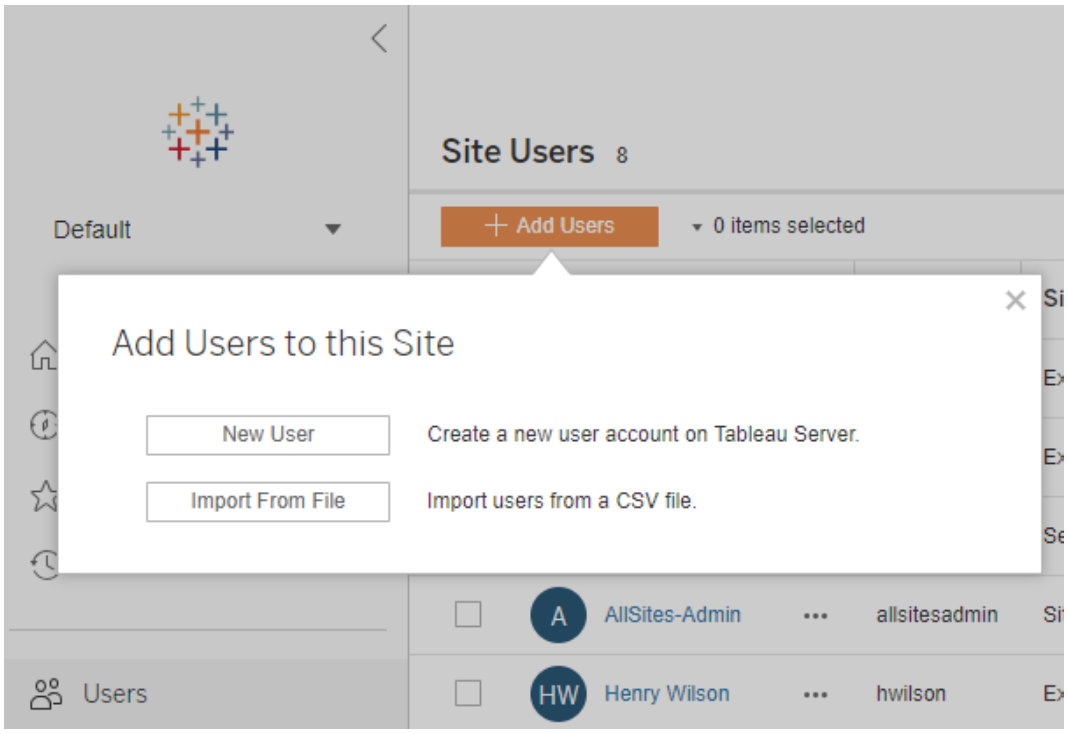
By default site administrators can add and remove users on a site. On the site's Settings page, server administrators can revoke that capability, so that only server administrators can manage the site's users.

A site administrator can edit an existing local user account only if the administrator has access to all of the sites the user is a member of. For example, if User1 is a member of sites A and B, an administrator of only site B cannot edit User1's full name or reset the password.

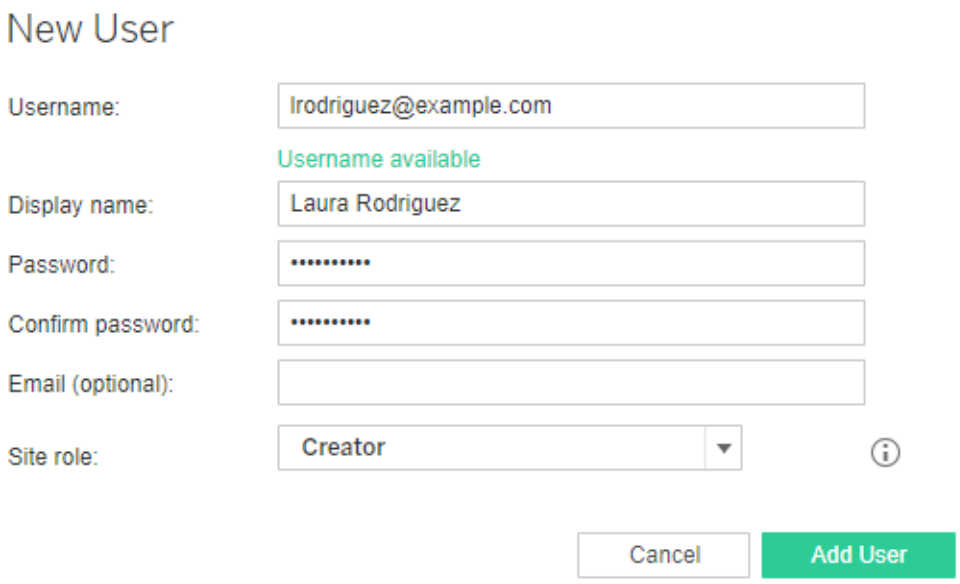
Note: When using local authentication, if a site administrator is able to add and remove users, they will be able to tell if a username is configured as a user on any site of Tableau Server

Add local users to a site

1. Sign in to Tableau Server as an administrator, and if applicable select the site.
2. Select **Users**. On the Users page, click **Add Users**, and then click **New User**.



3. Enter a user name. With local authentication, using an email address for the user name is the best way to avoid user name collisions (for example, *lrodriguez@example.com* instead of *lrodriguez*).



User names are not case sensitive. Characters not allowed in user names include the semi-colon (;) and colon (,).

Also enter information in the following fields:

- **Display Name**—Type a display name for the user (e.g., *Laura Rodriguez*).
- **Password**—Type a password for the user.
- **Confirm password**—Retype the password.
- **Email**—This is optional and can be added at a later time in the user profile settings.

4. Select a site role.

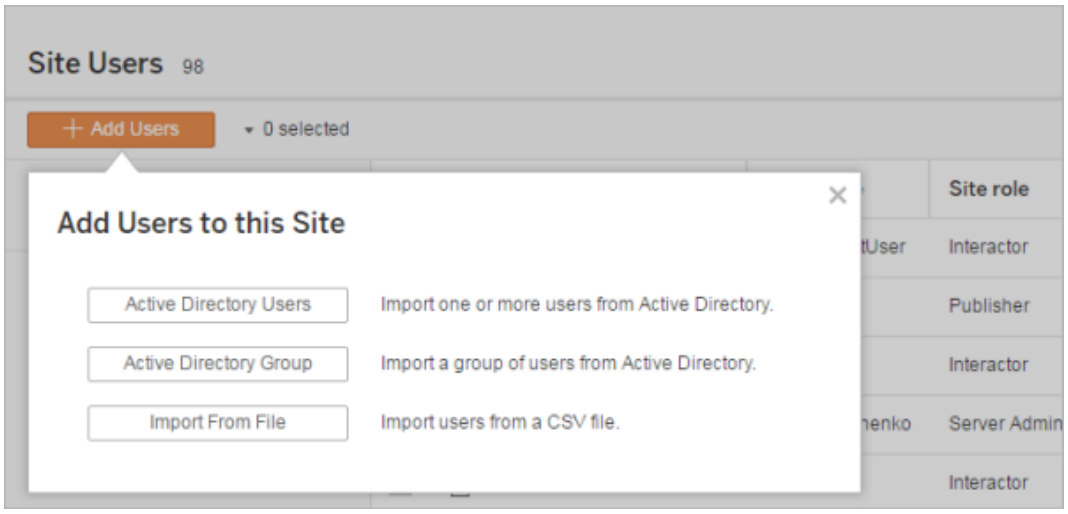
   For site role definitions, see Set Users' Site Roles.

5. Click **Add User**.

Add Active Directory users to a site

Before adding users to a site, be sure to review User Management in Deployments with External Identity Stores to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

1. In a site, click **Users**, and then click **Add Users**, and then click **Active Directory User**.



2. Enter one or more user names (separated by semicolons).

   For example, *tdavis; jjohnson; hwilson*

   If you are adding a user that is from the same Active Directory domain that the server is running on, you can type the AD user name without the domain. The server's domain will be assumed.

   > **Note:** Do not enter the user's full name in this field; it can cause errors during the importing process.

3. Select a site role.

For site role definitions, see Set Users' Site Roles.

4. Click **Import Users**.

Remove local users

1. Sign in to Tableau Server as an administrator, select the site, and open the **Users** page.
2. Select the check boxes next to the users' names, and on the **Actions** menu, select **Remove**.

   If a user is a member only of the current site, and they do not own any content, the user is removed from the server.

   If a user you remove from the current site is a member of other sites on the server, their status remains unchanged on those sites.

Related information

- You can also Add Users to Tableau Server, without site affiliation.
- Upgrading Tableau Server to version 2018.1 or later from a pre-2018.1 version, without activating user-based licenses, affects users who were assigned the **Viewer** site role in the pre-2018.1 server version.

  To learn more, see the section "User-based licenses" in the Licensing Overview and see Set Users' Site Roles.

Set Users' Site Roles

When you add users to a site on Tableau Server, independent of their license type, you must apply a *site role* to them. The site role signifies the maximum level of access a user can have on the site. Along with content permissions, the site role determines who can publish, interact with or only view published content, or who can manage the site's users and administer the site itself.

Looking for Tableau Server on Windows? See Set Users' Site Roles.

How user licenses, site roles, and content permissions work together

The intersection of a user's license type, site role, and content permissions determines the level of access a user has on the Tableau site.

Note: The license level count at the top of the header on the **Server Users** tab may differ from the count under the **Max User Role** filter due to some users having different roles across sites.

1. The license type is associated with the user. The site role you want to assign to the user determines the license type they require.

   In a multi-site environment on Tableau Server, a user's license applies to all sites the user is a member of.

2. The site role is also set at the user level. In a multi-site environment, you assign site roles on each site. For example, the same user can have the Site Administrator Creator site role on one site, and Viewer site role on another site.

   The site role defines the maximum capabilities the user can have.

3. Whether the site role's maximum capabilities are available to the user depends on the permissions set on the content resources (projects, data sources, workbooks).

For example, let's say that a user has the following access on a site:

- Creator license (due to their access on another site)
- Explorer site role (on this site)
- Save permission capability on a project (on this site)

In this scenario, the license allows connecting to and creating new data sources in the web editing environment or Tableau Desktop, and a permission rule allows them to save in a project. However, their site role prevents them from being able to save, so their effective permissions don't include the save capability. Therefore, the user can't publish content to the site.

Even if a user has a creator license and a creator site role, if they don't have the save capability on at least one project, they can't publish anything to the site.

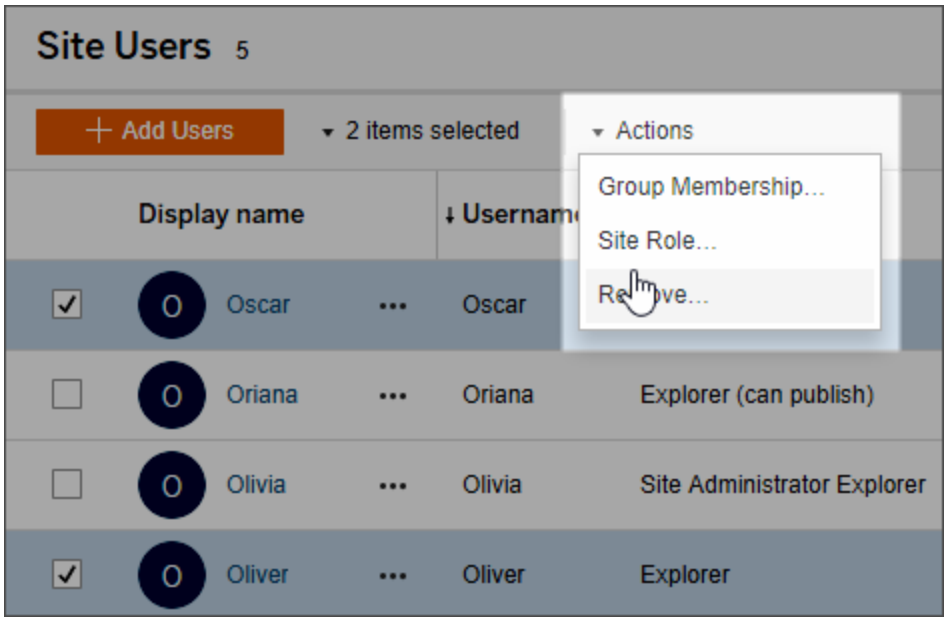
For more information, see Permissions.

Change a user's site role

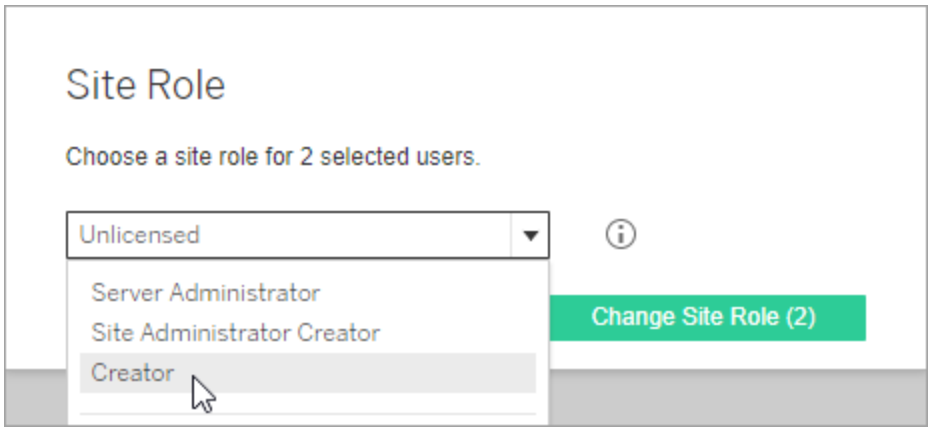
1. Sign in to the site as a server or site administrator, and go to the **Users** area.

   If you are a site administrator and don't see the **Users** area, check with your server administrator on whether they have denied user management capabilities to site administrators.

2. Select the users, and then select **Actions** > **Site Role**.



3. Select the new site role, and then click **Change Site Role**.



You can hover the pointer over the information icon to display a matrix that shows the maximum level of general capabilities each site role allows. For more information, continue to General capabilities allowed with each site role.

General capabilities allowed with each site role

The following table lists the license types as of version 2018.1, the highest level of site role allowed with each, how each site role maps to its pre-2018.1 equivalent; and summarizes the maximum capabilities each site role allows.

What this article covers and where to find what's not covered here

- This information focuses on *site* roles and is more generalized. For a list of common specific tasks available per *license* role, see the matrix on the For Teams & Organizations tab on the Tableau pricing page.
- This information describes site roles as of version 2018.1. To learn more about how core-based licensing relates to user-based licensing, how licenses transfer, or other specific licensing transition scenarios, start with the following topics:

  Migrate from Core-Based to Role-Based Licensing

Tableau site roles as of version 2018.1

Site role name as of version 2018.1	Previous site role name	Maximum capabilities this site role allows
Site roles that use a Creator license		
—Users with these site roles have access to Tableau clients such as Tableau Prep, Tableau Desktop, Tableau Bridge, and Tableau Mobile.		
Server Administrator	Server Administrator	Available on Tableau Server only; not applicable to Tableau Cloud. This site role always occupies the highest license activated on the server between Creator and Explorer. It allows unrestricted access to the configuration settings for the Tableau Server browser environment, all sites on the server, users and groups, and all content assets, such as flows, projects, data sources (including connection information), and workbooks. Connect to Tableau published data sources or external data, from the browser, Tableau Desktop, or Tableau Prep; create and publish new data sources; author and publish workbooks.
Site Administrator Creator	--	This is the highest level of access for Tableau Cloud. Unrestricted access to content as described above,

Site role name as of version 2018.1	Previous site role name	Maximum capabilities this site role allows
		but at the site level. Connect to Tableau or external data in the browser, Tableau Desktop, or Tableau Prep; create new data sources; build and publish content. On Tableau Server, server administrators can determine whether or not to allow site administrators to manage users and assign site roles and site membership. By default, on Tableau Server, and always on Tableau Cloud, site administrators are allowed these capabilities.
Creator	--	This is similar to the former Publisher site role, but allows new features. This site role offers non-administrators the maximum level of *content* access. Connect to Tableau or external data in the browser, build and publish flows, data sources and workbooks, have access to Dashboard Starters, and use interaction features on published views. Can also connect to data from Tableau Prep or Tableau Desktop, publish (upload/save) and download flows, workbooks and data sources.
Site roles that use an Explorer license		
—Users with these site roles can access the server from the browser or Tableau Mobile.		
Server Administrator	N/A	Tableau Server only; not applicable to Tableau Cloud. If Explorer is the highest license type activated on the

Site role name as of version 2018.1	Previous site role name	Maximum capabilities this site role allows
		server when a new server administrator user is created, the user's site role is Server Administrator. However, the user won't have the full connecting and publishing capabilities that come only with the Creator license. With the Explorer license a Server Administrator has unrestricted access to the configuration settings for the Tableau Server browser environment, all sites on the server, users and groups, and all content assets, such as projects, flows, data sources (including connection information), and workbooks. However, with the Explorer license, a Server Administrator can't connect to external data from the browser to create a new data source. They can author or publish workbooks and data sources from Tableau Desktop. With regards to publishing, they have the same capabilities that the Explorer (can publish) site role does. They can't publish Tableau Prep flows.
Site Administrator Explorer	Site Administrator	Same access to site and user configuration as Site Administrator Creator, but can't connect to external data or virtual connections from the web editing environment. Can connect to Tableau published data sources to create new workbooks, and edit and save existing workbooks. Can't publish Tableau Prep flows.

Site role name as of version 2018.1	Previous site role name	Maximum capabilities this site role allows
Explorer (can publish)	Publisher	Can author or publish workbooks and data sources from Tableau Desktop. Can also publish workbooks from the web using existing data sources, browse and interact with published views, and use all interaction features. In the web editing environment, can edit and save existing workbooks. Can't save new standalone data sources from data connections embedded in workbooks, and can’t connect to external data or virtual connections, or create new data sources.
Explorer	Interactor	Can browse and interact with published views. Can subscribe to content, create data driven alerts, connect to Tableau published data sources and open workbooks in the web authoring environment for ad-hoc queries, but they can’t save their work. Can’t connect to a virtual connection. Can't publish Tableau Prep flows.
Read Only	Viewer	This site role is available only in version 2018.1, for transitioning users to the user-based Viewer (or other) license and site role. Any users in the Read Only site role prior to upgrading to version 2018.2 or later are reassigned to the Viewer site role. In 2018.1 versions, Read Only users can see and subscribe to published views others have created. Can’t use other interaction features or save custom views.

Site role name as of version 2018.1	Previous site role name	Maximum capabilities this site role allows
Site roles that use a Viewer license		
Viewer	N/A	Can see published views others have created and use most interaction features. Can subscribe to views and download as images or summary data. Can't connect to data, create, edit, or publish content, or set data alerts. For a list of specific capabilities, see the **Viewer** column in the matrix on the Tableau pricing page. **Note:** Although the Viewer site role existed in previous versions, the new Viewer site role has additional capabilities.
Other site roles		
Unlicensed	Unlicensed	Unlicensed users can't sign in to Tableau Server or Tableau Cloud. Users are assigned the Unlicensed role in the following circumstances: • You import users from a CSV file and their license level is set to unlicensed. • The number of available licenses is reached at the time you add or import users. • You remove a user who owns content on the site. The user will still own the content but not be able to do anything with it. • A product key(s) has expired. See Refresh Expiration Date and Attributes for the Product

Site role name as of version 2018.1	Previous site role name	Maximum capabilities this site role allows
		Key.

Who can publish content

The following site roles allow the specified level of publishing access.

- **Server Administrator** (Tableau Server only); **Site Administrator Creator**; and **Creator** allow full connecting and publishing access.

  This includes connecting to data and publishing new flows, new workbooks and new data sources from Tableau Desktop and the web editing environment. The site roles also allow editing and saving existing published workbooks, or publishing updates to existing data sources.

- **Explorer (Can Publish)** and **Site Administrator Explorer** have limited publishing capabilities, as described in General capabilities allowed with each site role.

- **Explorer**, **Viewer**, **Read Only**, and **Unlicensed** don't allow publishing.

Site roles and Active Directory import and synchronization

When you import users from an external directory like Active Directory, you can specify the site role. If a user is not yet a member of any site on the server, the user is added to the site with the assigned role. When you synchronize groups from an external directory, the site role is applied through the **Minimum Site Role** setting on the **Groups - Details** page.

Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

If a user already exists in a Tableau Server site, the site role assigned during the import or sync process will be applied if it gives the user more access in a site. Importing or synchronizing AD users and groups can promote a user's site role, but does not demote a user's site role.

If a user already has the ability to publish, that ability is maintained.

The matrix below shows the rules applied for site roles on import.

Note: The **Import Site Role** row abbreviated headers indicate the site role specified for import. The **Current Site Role** column headers represent the current user site role. The table values represent the abbreviated resulting site role.

- Site Administrator: SA
- Site Administrator Creator: SC
- Site Administrator Explorer: SE
- Creator: C
- Explorer: E
- Explorer (Can Publish): EP
- Viewer: V
- Unlicensed: U

	Current Site Role						
Import Site Role	**SC**	**C**	**SE**	**EP**	**E**	**V**	**U**
Site Administrator Creator (SC)	SC	SC	SC	SC	SC	SC	SC
Site Administrator Explorer (SE)	SC	SE	SE	SE	SE	SE	SE

	Current Site Role						
Import Site Role	**SC**	**C**	**SE**	**EP**	**E**	**V**	**U**
Creator (C)	SC	C	SE	C	C	C	C
Explorer (Can Publish) (EP)	SC	C	SE	EP	EP	EP	EP
Explorer (E)	SC	C	SE	EP	E	E	E
Viewer (V)	SC	C	SE	EP	E	V	V
Unlicensed (U)	SC	C	SE	EP	E	V	U

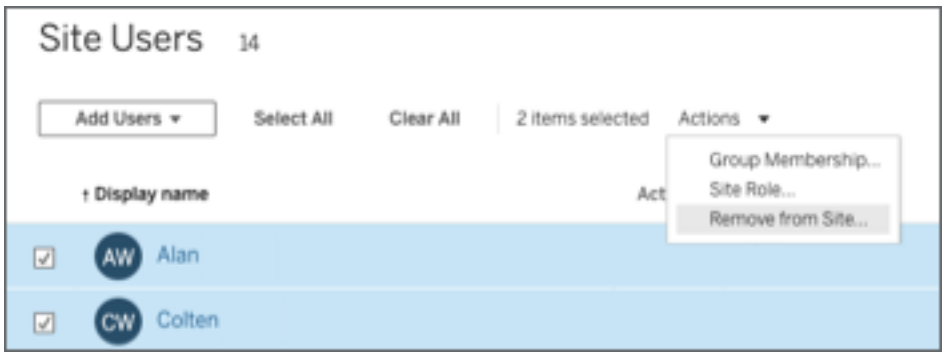
View, Manage, or Remove Users

Administrators can manage a site's users such as adding and removing users, setting the groups they're members of, setting their site roles, and so on. Server administrators can manage users on multiple sites at a time on the **All Sites** page.

View and manage users on a site

Sign in to a site as an administrator, and then select **Users**. On this page you can do any of the following to manage users:

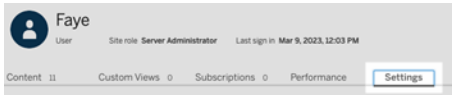
- Set group membership, set site role, or remove the user from the site. If you've configured the site for SAML single sign-on, you can set the selected users' authentication type. If your server is configured for password policies and account access lockout, you can unlock access to a user's locked out account. If you server is configured for one or more identity pools, you can add or remove users.



- Select a user name to see details about them, such as content they own, views they subscribe to, and their account settings.

  The user **Settings** page is available when the following conditions are true:

  - The user is a member only of sites that the site administrator controls
  - Site administrators can manage users by default. Tableau Server administrators can change this access for site admins.



  If the server is configured to use the internal user management system (Local Authentication), you can edit the **Display Name**, **Email**, and **Password** for users after they have been added. If you are making many changes, you may find it easier to import the changes from a CSV file. For details, see Import Users and CSV Import File Guidelines.

View and edit server users

Sign in to Tableau Server as a server administrator. On the site menu, select **Manage All Sites**, and then select **Users**.

Manage users' site membership

By default, server and site administrators can manage users at the individual site level. Server administrators can also manage users and their site roles on multiple sites. You do this at the **All Sites** level (at the server level).

1. In the site menu, select **Manage All Sites**, and then select **Users**.
2. On the Server Users page, select the check boxes next to the users, and then select **Actions** > **Site Membership**.
3. Select one or more sites, and a site role for each site, and then click **Save**.

Search for users (or groups or sites)

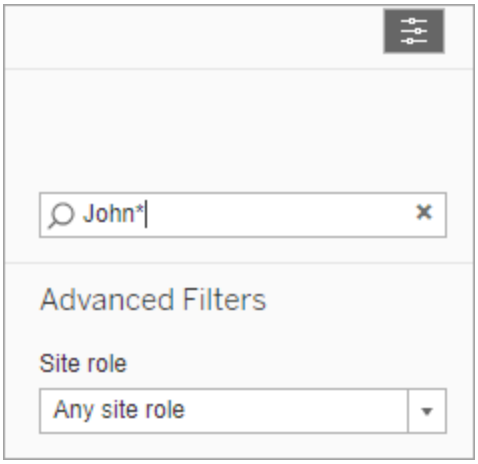
To search for a specific user (or group or site), use the filter toggle in the upper right to display the search box and site role filter. Then use the search box or filters to find the users (or group or site) you want. The search operation checks the display name and user name attributes.

The search box supports the wildcard (*) character. For example, searching for `John*` will return all names that start with *John*.

In addition:

- Starting in Tableau Server 2021.4.1, you can use the wildcard character (*) with a special character to search for names that contain special characters. For example, `sync-*` or `*sync-*`.
- Starting in Tableau Server 2022.1.13, when you search for names with diacritics, names must be entered with exact diacritics return relevant results. For example, to search for *José*, enter `José`. Searching for `Jose` will not return results.

- Starting in Tableau Server 2022.3.5, you can use the wildcard character (*) with AND or OR conditions when filtering users. For example, searching for `*aw* AND John*` returns all users whose names contain *aw* and whose names start with *John*.



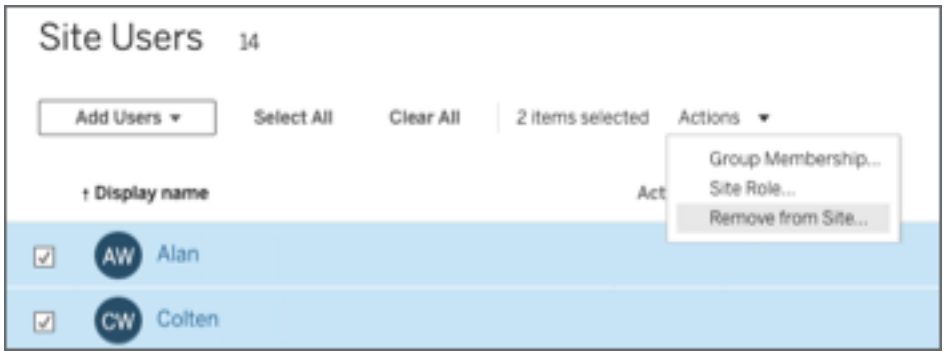
Remove users from a site

You can remove a user only if the user does not own any content (projects, workbooks, views, data sources, collections, or data alerts for example). If you attempt to remove a user who owns content, the user site role will be set to **Unlicensed**, but the user will not be removed.

If the default All Users group has enabled Grant site role on sign in, that user's content must be reassigned to another user or removed before they can be unlicensed or removed. For more information on Grant role on sign in, see Removing users affected by Grant role on sign in. For more information on changing content ownership, see Manage Content Ownership

Note: On Tableau Server, when an administrator removes a user from a site (and the user belongs only to that one site), the user is also deleted from the server.

1. Sign in to a site as an administrator, and go to the **Users** area. Select one or more users to remove, and then select **Actions** > **Remove**.



2. Click **Remove** in the confirmation dialog.

Remove users from the server

You can remove a user only if the user does not own any content (projects, workbooks, views, or data sources). If you attempt to remove a user who owns content, the user site role will be set to Unlicensed, but the user will not removed.

If a user is a member of multiple sites, and they own content on any of those sites, they are removed from the sites on which they don't own content. The user remains a member on sites where they own content, but demoted to the Unlicensed site role.

1. In the site menu, click **Manage All Sites**, and then click **Users**. In a single-site environment, click **Users**.

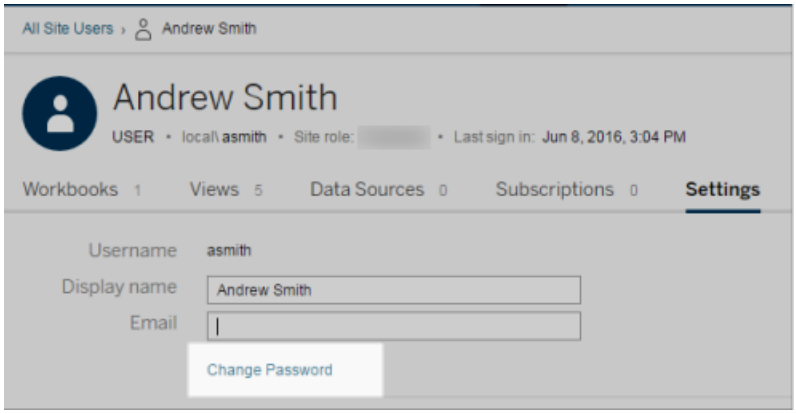
   Select one or more users to delete, and then click **Actions** > **Delete**.

2. Click **Delete** in the confirmation dialog box.

Change passwords for users of a single site

To change the password for a user with membership to a single site, sign in to Tableau Server as a site administrator or a server administrator.

1. Ensure that the correct site is selected in the menu.
2. Click **Users**.
3. Click the display name of a user.
4. Click **Settings**.
5. Click the **Change Password** link, edit the password, and then click **Save Password**.

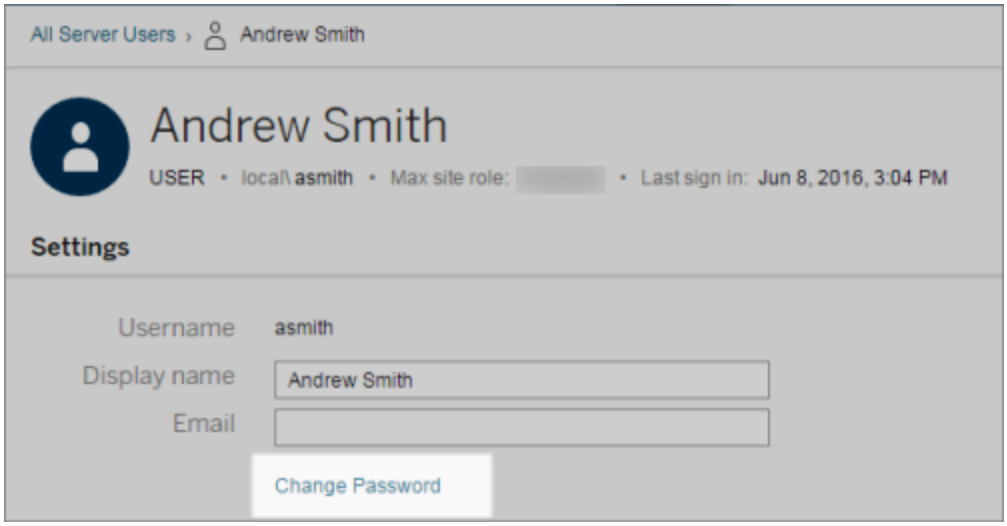


Change passwords for users of multiple sites

To change the password of a user with membership to multiple sites, sign in to Tableau Server as a server administrator.

1. In the site menu, click **Manage All Sites**.
2. Click **Users**.
3. Click the display name of a user.

4. Click the **Change Password** link, edit the password, and then click **Save Password**.

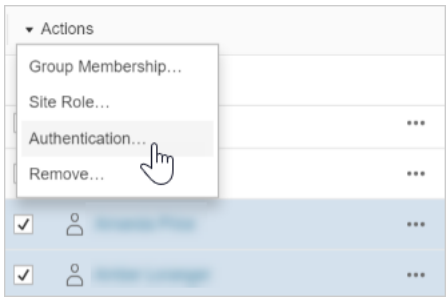


Set the User Authentication Type for SAML

On a site that has been configured for site-specific SAML, administrators can specify users' authentication type. For example, if Tableau Server was configured for site-specific SAML and server-wide SAML, administrators can specify which users authenticate with site-specific SAML and which users authenticate with server-wide SAML.

You can assign authentication type at the time you add users to Tableau Server, as well as any time afterward.

1. When you're signed in to the Tableau Server site, select **Users**.
2. On the **Site Users** page, select the check boxes next to the users you want to assign an authentication type.
3. On the **Actions** menu, select **Authentication**.



4. In the Authentication dialog box, select **Site SAML** or **Server Default**.

Notes

- Users that authenticate with site-specific SAML can only belong to one site. If a user needs to belong to multiple sites, set their authentication type to the server default. Depending on how site-specific SAML was configured by the server administrator, the server default is either local authentication or server-wide SAML.
- If you change users' authentication to site-specific SAML, the next time they sign in, they will be directed to your identity provider's site to provide their credentials.

Import Users

To automate the process of adding users to a site, you can create a CSV file that contains user information, and then import the file.

Site administrators can import users to a particular site; server administrators (Tableau Server only) can import users at the server level, to later add them to multiple sites.

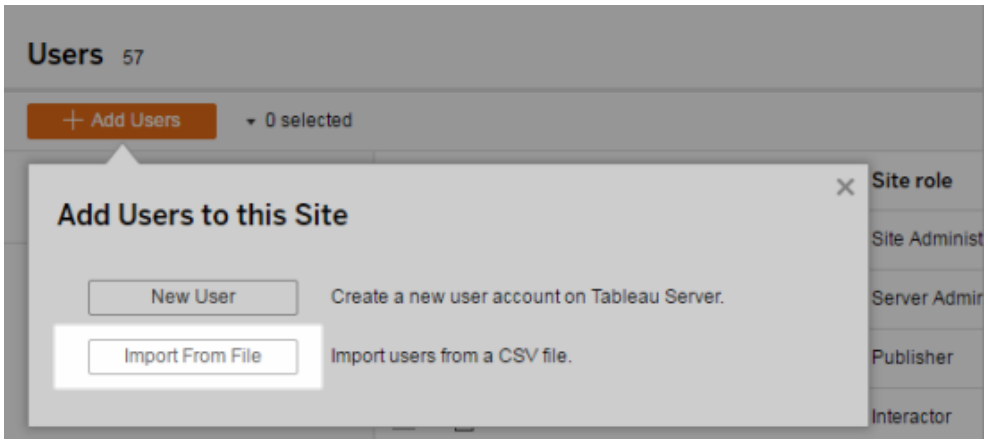
> **Note:** This topic contains the steps for importing, assuming that you have already created the CSV file. If you have not created the file yet, see CSV Import File Guidelines for a list of file format requirements and import options.

Add users from a CSV file

The following steps describe how to add users to a site or to the server. The images reflect adding users at the site level.

1. Do one of the following:

   - To add users at the site level, select **Users**, and then **Add Users**.



   - To add users at the server level on a **single-site** server, select **Users**, and then **Add Users**.

   - To add users at the server level on a **multi-site** server, open the list of sites, and select **Manage All Sites**. Select **Users**, and then **Add Users**.

2. Click **Import From File**, click **Browse** and navigate to the file, and then click **Import Users**.

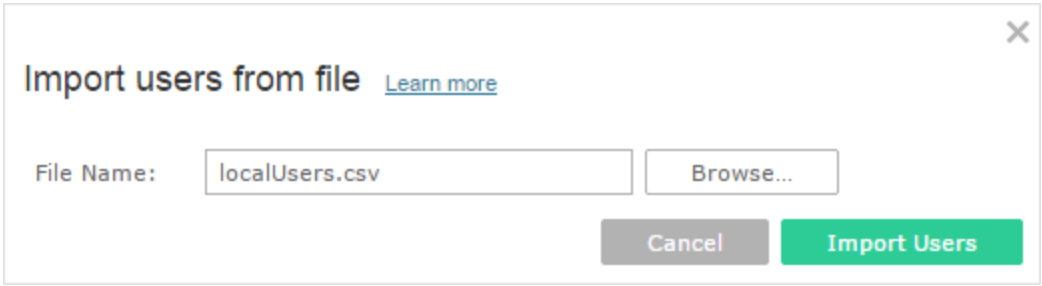
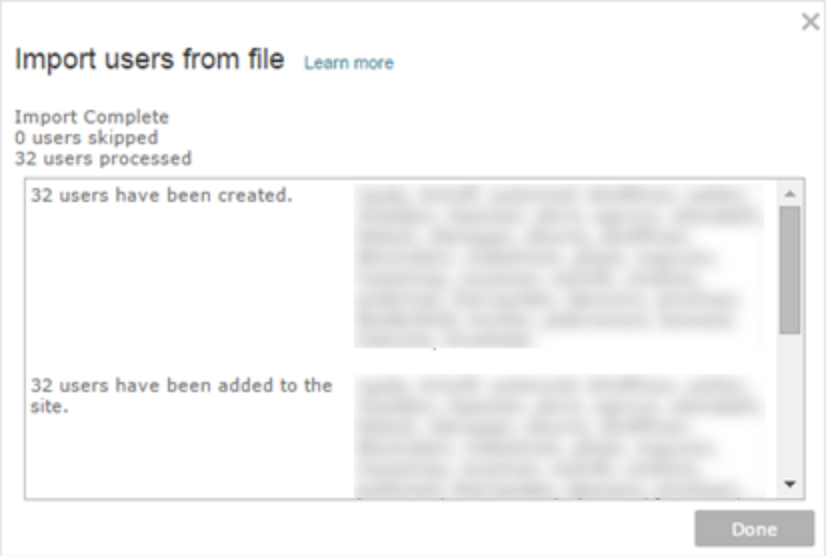


Tableau displays the results of the import process (names in this image are blurred).



3. Click **Done**.

How users' site roles are assigned or maintained

When you import at the site level or on a single-site server using `tabcmd`, you can specify the site role for all users in the CSV file. If a user already exists in the Tableau Server site, the site role assigned during the import process will be applied, even if it is more restrictive than users' existing site role. The exception is that you cannot affect a server administrator's site role.

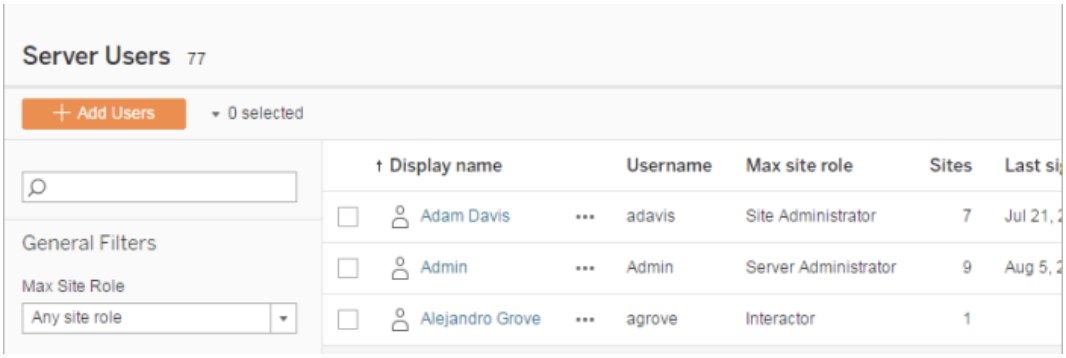
For a multi-site server, when you import users in the Server Users page, you create server users with no site affiliation. Because these users do not belong to a site, they cannot have a site role. The only site role a user can have at the server level is **Unlicensed** or **Server Administrator**.

You can also specify the user's site role when you assign site membership to a user. For information, see Manage users' site membership.

Importing at the server level in multi-site environments

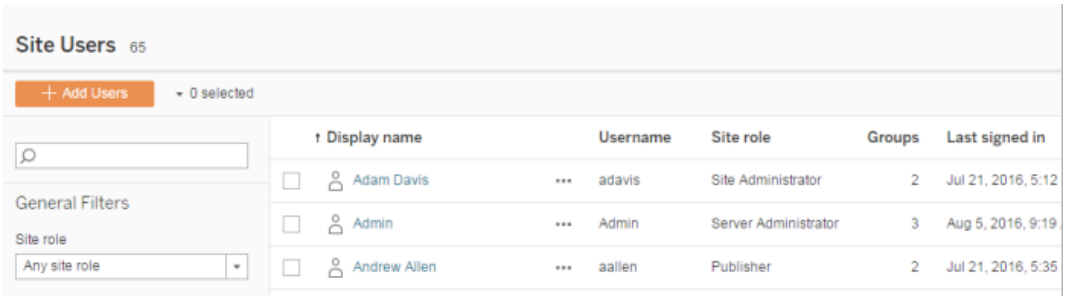
If the server is running multiple sites and you are a server administrator, you can import a CSV file from two locations. Where existing user accounts are concerned, each location has different capabilities.

- The **Server Users** page appears in a multi-site environment. Only server administrators can access this page.



  You can import the CSV file from here if you want to update existing user accounts in addition to adding new ones. For example, if you import a file that has a new password for each existing user, their passwords will be reset.

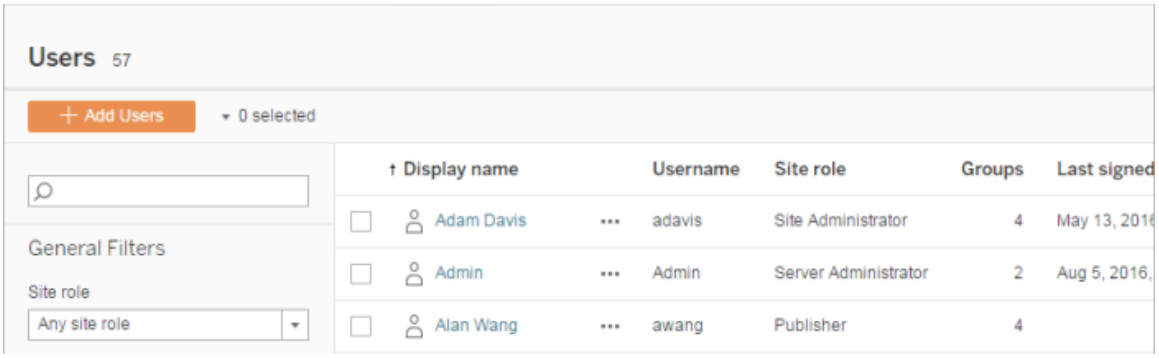
- The **Site Users** page.



  Server administrators can add new user accounts with CSV imports. If the CSV file includes existing users, the **Password** and **Display Name** fields must either match the existing or be left blank. If new passwords or full names are used, the import will fail.

Importing to a single-site environment

Server and site administrators on a single-site server perform CSV user imports from the **Users** page in a site.



Multi-site versus single-site import

Users can belong to more than one site on the same server, but they must use the same credentials for each site. This becomes important when you're adding users to a site and those users might already be members of a different site. If you try to import a user who already exists, and if the user's credentials in the CSV file don't match the existing credentials, the import fails for that user.

If you're importing users to a site and think the users might already exist on the server, you can try leaving the **Password** field in the CSV file blank (while including the delimiters for the field). If a user who is defined in the CSV already exists in another site, the user is added to the site where you're importing. However, if the user *doesn't* already exist on the server, the user is created, and the CSV import window alerts you that the new user doesn't have a password. You can then use the server administrator pages to assign a password to any user who doesn't have one.

Note: If the server is configured to use Active Directory for authentication, user passwords are managed by Active Directory, and you can leave the password field in the CSV file blank.

CSV Import File Guidelines

You can automate adding users by creating a comma-separated values (CSV) file with user information and then importing the file. You can include attributes in the CSV file, such as license level and the publishing access, to apply to the users at the same time you import them.

To import users, you can use the server or site administration pages or the `tabcmd` utility. Using `tabcmd` provides an option for assigning a site role to all users in the CSV file. For information, see Import Users or createsiteusers filename.csv.

You can import users at the site or server level. If you import users to the server (not to a specific site), the users aren't assigned to a site and are imported as Unlicensed.

Note: Unless otherwise noted, the guidelines specified in this topic apply to Tableau Server when configured with or without identity pools.

CSV file format requirements

When you create the CSV file for importing users, make sure that the file meets the following formatting requirements:

- The file does not include column headings. Tableau Server assumes that every line in the file represents a user.
- The file is in UTF-8 format, and includes the byte-order mark (BOM).
- Character encodings such as BIG-5 have been converted to UTF-8. You can do this by opening the file in a text editor and using the **Save As** command.

- If a user name includes an @ character that represents anything other than a domain separator, you need to refer to the symbol using the hexadecimal format: `\0x40`

  For example, `user@fremont@mycompany.com` should be `user-\0x40fremont@mycompany.com`

Required columns in the CSV file

The following fields are required for each user:

- Username. The user name. If the server is configured to use Active Directory, this value must match a user defined in Active Directory. If the user name is not unique across domains, you must include the domain as part of the user name (for example, `example\Adam` or `adam@example`).

  If adding users to an identity pool, make sure of the following:

  - If adding a user to an identity pool that uses AD as its identity store, make sure to use the AD sAMAccountName value for user name.
  - If adding a user to an identity pool that uses LDAP as its identity store, make sure to use the LDAP username value for user name.

- Password. A password for the user.

  - If the server is configured to use Active Directory, this value is not used however, there must be a password column and column itself should be empty.

  - If the server is using local authentication, you must provide passwords for new users.

  Note: Enforcement of the required password field started in Tableau Server 2024.2. For more information, see Unexpected “errorCode=134” occurs when attempting to add users via tabcmd in Tableau Server 2024.2 knowledge article.

Additional import file options

The CSV file can contain the following fields in addition to the fields listed above, in the order shown here:

- Display name. The display name is part of the information used to identify a user on the server. If the user's display name is already in use, Tableau Server updates the existing user information with the settings in the CSV file. If the server is configured using Active Directory, this value is not used.
- License level. This can be **Creator**, **Explorer**, **Viewer**, or **Unlicensed**. If you specify **Creator** for a particular user account, then you must also set the Publishing capability to **True**.
- Administrator level (**System**, **Site**, or **None**). This setting determines whether the user is imported as an administrator.

  If you are using the web UI to import users, you can set the administrator site role to **System** only if you import the file at the server (All Sites) level. If you are signed in to a specific site, and if the administrator column for a user in the CSV file is set to **System**, Tableau Server imports the user as a site administrator.
- Publishing capability (**yes/true/1** or **no/false/0**). If you are using the web UI, the publisher setting is used only if you import while signed in to a specific site.
- Email address. The email address is part of the information used to identify a user on the server. If the email address is already in use, Tableau Server updates the existing user information with the settings in the CSV file.

If adding users to an identity pool, the following values are needed in addition to the above:

- Identity pool name. The name of the identity pool that you want to add the user to.
- Identifier. The identifier for the user you want to add. Identifiers are only used for identity matching purposes. For more information, see Usernames and identifiers in

Tableau. **Note:** The identifier is required if adding a user to an identity pool that uses Active Directory (or LDAP) identity store. The identifier is optional if adding a user to an identity pool that uses the local identity store.

Notes:

- If you are adding users to an identity pool and you don't specify the identity pool name, users are added to the initial pool (TSM configured), which is the set of users who were provisioned in TSM during Tableau Server setup.
- For the user name value, make sure of the following:
  - If adding a user to an identity pool that uses AD as its identity store, make sure to use the AD sAMAccountName value for user name.
  - If adding a user to an identity pool that uses LDAP as its identity store, make sure to use the LDAP username value for user name.
- You can use the CSV import process to:
  - Bulk add users to additional identity pools. **Note:** You cannot use the CSV import process to replace the identity pool that a user already belongs to with another identity pool. If you add an existing user with a different identity pool value, they will be added to that additional identity pool.
  - Bulk add identifiers for users who don’t already have them. **Note:** If you add a different identifier for a user in the same pool, it will not replace the existing identifier for that user. Instead, a new identifier record will be created for that user.

Important: The order of the columns is significant. The first column is treated as the user name, the second as the password, the third as display name, and so on, regardless of the content in the columns. If you omit values for a field, you must still include the field’s comma delimiter.

Improve performance for large CSV files passed through tabcmd

> **Note:** These settings apply to Tableau Server version 2022.1 and earlier. The search and index service they affect was deprecated starting in version 2022.3 and retired (removed completely) in 2023.3.

A server administrator can enable server settings that help to improve performance for importing large CSV files through tabcmd commands. You can do this using the `tsm configuration set` command with the following options:

- `vizportal.csv_user_mgmt.index_site_users`
- `vizportal.csv_user_mgmt.bulk_index_users`
- `searchserver.index.bulk_query_user_groups`

Essentially, these options index users after the CSV file is processed, instead of one-by-one as they are added to the server's database. This reduces the number of calls to the database and the memory required to process the file. These `tsm configuration set` options apply to the `tabcmd createsiteusers`, `deletesiteusers`, `addusers`, and `removeusers` commands.

For descriptions for these settings, see tsm configuration set Options.

Notes

- If you are not signed in to a specific site and are importing users at the server level, you can assign only the Server Administrator and Unlicensed site roles.
- If you have a user-based server installation, and if adding users would exceed the number of users allowed by your license, the users are added as unlicensed users.
- If you use `tabcmd` and specify the license, but importing users would exceed your license limits, users are imported as Unlicensed.

CSV settings and site roles

The license level, administrator, and publishing settings for a user determine how the user's site role is set during the import process. The following table shows how the settings are converted to site roles.

CSV settings	Site role
License level=(any) Administrator=System Publisher=true	Server Administrator. This setting applies to Tableau Server only, and it is valid only if you are importing users while managing the server (that is, not signed in to a specific site). The Server Administrator site role always takes a Creator license if one is available. If no Creator license is available, see Troubleshoot Licensing to learn about the way Tableau Server handles this.
License level=Creator or Explorer Administrator=Site Publisher=true	Site Administrator Creator or Site Administrator Explorer. This setting is valid only if you are importing users while signed in to a specific site.
License level=Creator Administrator=None Publisher=true	Creator
License level=Explorer Administrator=None Publisher=true	Explorer (Can Publish)
License level=Explorer	Explorer

CSV settings	Site role
Administrator=None Publisher=false	
License level=Viewer Administrator=None Publisher=false	Viewer
License level=Unlicensed Administrator=None Publisher=false	Unlicensed

CSV import examples for Tableau Server

The following example shows a CSV file that contains information for several users.

```
henryw,henrypassword,Henry Wilson,Creat-
or,None,yes,henryw@example.com
freds,fredpassword,Fred Suzuki,Viewer,None,no,freds@example.com
alanw,alanpassword,Alan Wang,Explorer,Site,yes,alanw@example.com
michellek,michellepassword,Michelle Kim,Creat-
or,System,yes,michellek@example.com
```

If you import this file while managing a site, four users are added to that site. The `Admin-istrator` setting for user Michelle is `System`. However, because you are importing the users into a site, Tableau Server give Michelle the Site Administrator Creator site role. Three of the users are allowed to publish.

If you import this file while managing the server, four users are added to the server, but they are not added to any site. Only one user is imported as a server administrator; the rest are set to Unlicensed.

Identity pools examples

The following example shows a CSV file that contains information for two users added to an identity pool.

```
henryw,henrypassword,Henry Wilson,View-
er,None,yes,hwilson@myco.com,General Contractors,hwilson
freds,fredpassword,Fred Suzuki,Creat-
or,None,no,fsuzuki@myco.com,General Contractors,fsuzuki
```

The following example shows a CSV file that contains information for two users added to an additional identity pool.

```
henryw,henrypassword,Henry Wilson,View-
er,None,yes,hwilson@myco.com,General Contractors 2,hwilson
freds,fredpassword,Fred Suzuki,Creat-
or,None,no,fsuzuki@myco.com,General Contractors 2,fsuzuki
```

The following example shows a CSV file that contains information for two users without existing identifiers.

```
janes,janepassword,Jane Smith,View-
er,None,yes,jsmith@myco.com,General Contractors,jwang
laurar,laurapassword,Laura Rodrig-
uez,Creator,None,no,lrodriguez@myco.com,General Con-
tractors,jrodriguez
```

Manage Site User Visibility

By default, all site users can see aliases, project ownership and comments by other users when permissions allow. The User Visibility setting lets administrators manage if users with Viewer and Explorer site roles see other users and groups on the site, which can be important for sites that are used by multiple clients. To learn more about site roles, see Set Users' Site Roles.

Limit user visibility

Setting User Visibility to **Limited** impacts certain collaboration tools and hides user information in Tableau Cloud and Tableau Server. Limited User Visibility either disables the feature for Viewers and Explorers (excluding Site Administrator Explorers), or removes user information from other areas. Note that Creators and administrators will still see user information when User Visibility is set to Limited.

To limit user visibility for Explorers and Viewers (excluding Site Administrator Explorers):

- Navigate to the site's **Settings** page
- Select **Limited** in the **User Visibility** setting

The following is a list of site areas impacted when User Visibility is set to Limited. Unless noted that the feature is disabled for all users, only non-administrator Explorers or Viewers are impacted.

Area	Impact
Search	User information not displayed
Content owners	User information not displayed (Explorers and Viewers can't see themselves, but can see their content in My Content)
Profile pictures	User information not displayed
Subscriptions	User information not displayed
Recommendations	Similar users not displayed (all users)
Add/Edit Tags	Explorers and Viewers can see tags but cannot delete or modify them
"Who has seen this view?"	Disabled
Ask Data usage ana-	Disabled

lytics	
Permissions dialogs	Disabled
Named sharing	Disabled (all users)
Alerts	Disabled (all users)
	Existing alerts paused
Comments	Disabled (all users)
Public Custom Views	Disabled (all users)
	Existing public custom views appear as private
Request Access	Disabled (all users)
Tableau Desktop	Publishing workbooks disabled from Desktop
	User information not displayed on user filters
Tableau Pulse	The button to see and manage followers doesn't appear on metrics
Tableau Catalog (with Data Management)	User information not displayed

When User Visibility is set to Limited, Tableau Server REST API and Metdata API calls behave as described in the table above.

Users on a site can interact with views and modify them, such as applying filters. If that user shares their modified view with others, or if the user creates something from that modified view (like a metric or a private custom view), then that user's name appears in the URL. Make sure that the URL for this modified view is only distributed to users who are permitted to see that person's name.

Note: If a user is a member of multiple sites, entering an email on the sign in page for Tableau Cloud will return the names of all sites the user is a member of.

Best practices for limiting user visibility

Administrators can also check that user and group information is not visible in these ways:

- Configure permissions to only provide content to appropriate parties. For more information, see Permissions.
    - Limited User Visibility hides user identification information from search, but might return content that the user published, including when searching by owner name, if the person searching has viewing permission to that content.
    - A user publishing a workbook with a duplicate title in the same project might see a warning that a workbook with that title already exists.
- Apply row-level security when necessary.
- Check that metadata within dashboards does not contain user information.
- Check that calculations accessible to users don't contain user metadata (e.g., user filters).

Restore Full User Visibility

When administrators set User Visibility back to Full, features disabled for all users by Limited User Visibility (such as comments and alerts) remain off. Administrators can re-enable these features through the site's Settings page.

Any previous feature settings are not retained when User Visibility is set to Full, and affected features are not automatically turned on.

Guest User

Core-based licenses of Tableau Server include a Guest user option, which you can use to let people access Tableau views without an account on the server.

Guest user access is enabled by default when Tableau Server is installed with a core-based license. It is not available with user-based licensing. If you do not intend to use Guest user access, you should disable it.

Guest access allows users only to see and interact with Tableau views. The Guest user cannot browse the Tableau Server interface or see server interface elements in the view, such as user name, account settings, comments, and so on. For more information about licenses, see Manage Licenses.

Tip: To share views with Guest users, either provide URL links or embed views into web pages. For more information, see Tableau User Help.

Guest user permissions

A Guest user can have the following maximum capabilities:

- **Workbooks and views**: View, Export Image, Summary Data, View Comments, Filter, Full Data, Web Edit, Download (to save a local copy)
- **Data sources**: View and Download

If you add the Guest user to a group that has a higher level of access to a content resource, the Guest user's access does not exceed the capabilities listed above. However, the Guest user account will comply with more restrictive permissions settings.

The Guest user can only access data with embedded credentials or where credentials are not required. A data source that prompts the user for credentials cannot be accessed by the Guest user. See Data access for published Tableau data sources.

Enable or disable Guest access

You must be a server administrator to change Guest account settings at either the server or the site level.

Note: Enabling the Guest user for a site can increase the number of potential simultaneous viewers beyond the user list you might be expecting. The administrative view **Status** > **Traffic to Views** can help you gauge the activity.

1. In the site menu, click **Manage All Sites** and then click **Settings** > **General**.
2. For **Guest Access**, select or clear **Enable guest access**.
3. Click **Save**.

This enables the Guest user on all sites. You can then go to the same setting for a specific site. To disallow Guest access for a site:

1. In the site menu, select a site.
2. Click **Settings**, and on the General tab, clear the **Enable guest access for this site** check box.

If the Guest account is enabled on some or all sites, and you turn off Guest access at the server level, it is turned off for all sites as well.

Note: You can turn off Guest user access at the server or site level; however, you aren't able to remove the user. So, although no one can access data or views without signing in to the server, the Guest user still appears in the Site Users list and group lists for groups you've added the Guest user to.

Additional Guest account characteristics

The Guest user is unique in the following additional ways:

- As a single user account, it represents all unauthenticated users accessing Tableau views.
- When enabled, it is a member of the All Users group.
- You can add it as a member of other groups on a site.
- You cannot edit it or select it as the owner of a content resource.
- The account is not allowed to save custom views.
- Guest cannot be used in a user filter.

- You cannot delete the account; however, you can turn off access to it by clearing the check box described in the steps above.

Work with Group Sets

Beginning in Tableau Server 2024.2, you can create a container for your groups using group sets. A group set can contain one or more groups and be used to apply more granular rules for content permissions that are dependent on the group set. When enabling capabilities based on a group set, users in the groups that belong to the group set must be members of all the groups for the capability to be evaluated. In this way, group sets enforce AND logic.

Benefits of group sets:

- You can mix and match synchronized groups with local groups in permission rules to enable more dynamic access control scenarios.
- Use AND logic for groups in permission rules, which can simplify access control in some scenarios.

Notes:

- Group set permission rules are evaluated after user and group rules. For more information about those rules, see Evaluate permission rules.
- Group sets can only be created by server administrators.

Turn on group sets

Before group sets can be used for permissions, group sets settings must be enabled.

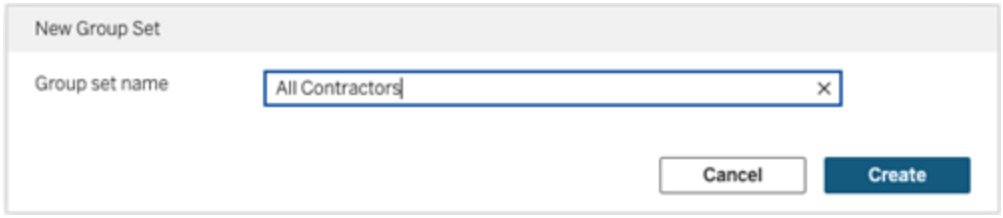
1. Sign in to Tableau Server as server administrator.
2. Navigate to the **Settings** page.
3. Under the Group Sets section, select the **Allow group sets** check box.

After enabling group sets, a dedicated Group Sets page displays in the navigation pane.

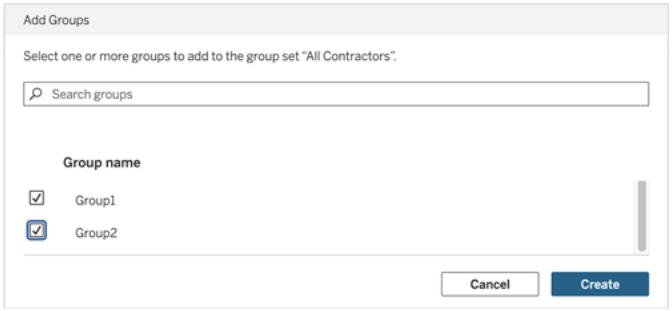
Create group sets

To create a group set, navigate to the Group Sets page and create a group set as you would a group.

1. Sign in to Tableau Server as server administrator.
2. Navigate to the Group Sets page and click the **New Group Sets** button.
3. Enter a name for the group set and click **Create**.



4. In the Group Sets table, click the name of the group set you just created and click the **Add Groups** button.
5. From the list of available groups, select the groups you want to add to the group set and click the **Add** button.



Set permissions on group sets

To use group sets, as a site admin, project leader, or content owner, add or edit the permissions of the content to use the group set.

For example, suppose you are the owner of the "Batters" workbook. To apply permissions based on the group set, do the following:

1. Go to the workbook and select **Permissions** from the actions menu.
2. In the Permissions dialog box, click the **Add Group/User Rule** button, and do the following:
    a. In the text box, enter the group set name, for example "All Contractors."
    b. Select the desired capabilities in the template.
    c. Click **Save**.

When permissions are applied using the group set model, you can enforce a more fine-grained access control.

For example, you might restrict access to different "Batters" workbook views based on a user's regional group affiliation:

- North region view:
    - Group set is called North Region
    - Groups in the group set: All Regions and North Region
- South region view:
    - Group set is called South region
    - Groups in the group set: All Regions and South Region
- East region view:
    - Group set is called East Region
    - Groups in group set: All Regions & East Region
- West region view:
    - Group set is called West Region
    - Groups in the group set: All Regions and West Region

For more information about permissions, see Configure Projects, Groups, Group Sets, and Permissions for Managed Self-Service.

Groups

You can create and delete user groups, add users to a group, and synchronize groups with Active Directory.

Add Users to a Group

You can organize Tableau Server users into groups to make it easier to manage multiple users. You can create groups on the server or import groups from Active Directory.

If you're managing users with an External identity store, such as Active Directory, add users to a group through the external identity store itself. Once users are added to a group in the external identity store, Tableau Server is able to update those users by synchronizing the group of users in the external identity store with the group of users on Tableau Server.

For example, to keep Active Directory group membership up to date, we recommend you review the following:

- Site administrators can synchronize selected groups on demand in a site. For more information, see Synchronize Active Directory Groups on a Site.
- Server administrators can synchronize all Active Directory groups on the server based on a schedule or on-demand. For more information, see Synchronize All Active Directory Groups on the Server.

Notes:

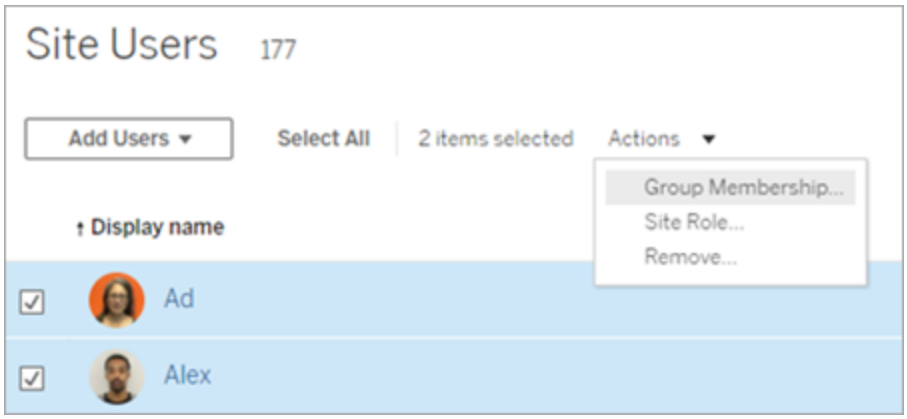
- In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions. When identity pools are configured, you cannot add groups to an identity pool.
- Adding groups to an identity pool is not supported. Only individual users can be added to an identity pool.

If you're managing users with a Local identity store, use the procedures described below to add users to a group.

To add a user to a group, the group must already exist.

Add users to a group (Users page)

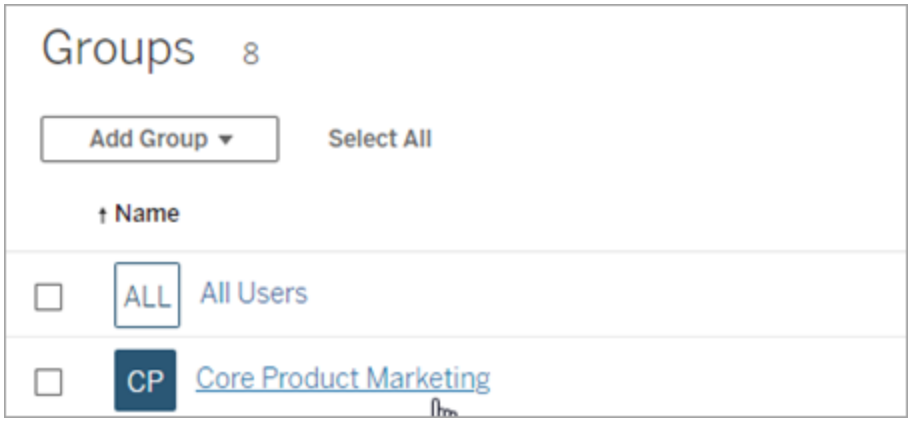
1. In a site, click **Users**.
2. Select the users you want to add to a group, and then click **Actions** > **Group Membership**.



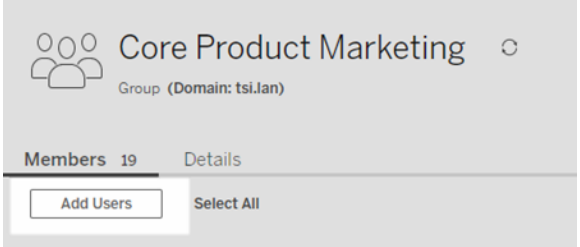
3. In the Group Membership dialog box, select the groups and then click **Save**.

Add users to a group (Groups page)

1. In a site, click **Groups**, and then click the name of the group.



2. In the group's page, click **Add Users**.

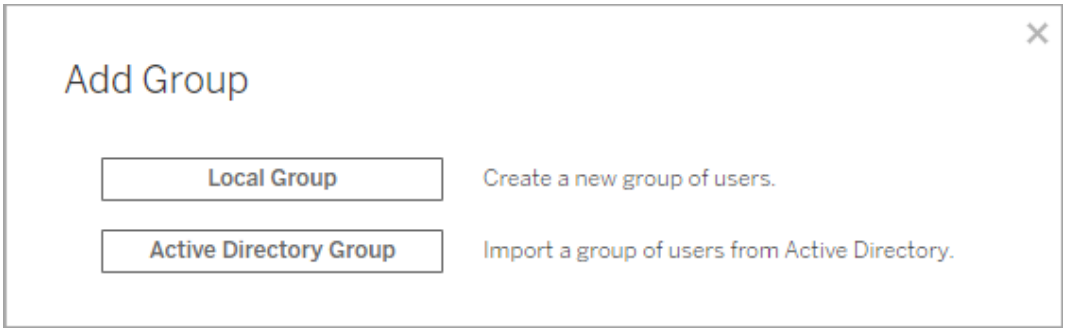


3. In the Add Users dialog box, select the users to be added, and then click **Add Users**.

Create a Local Group

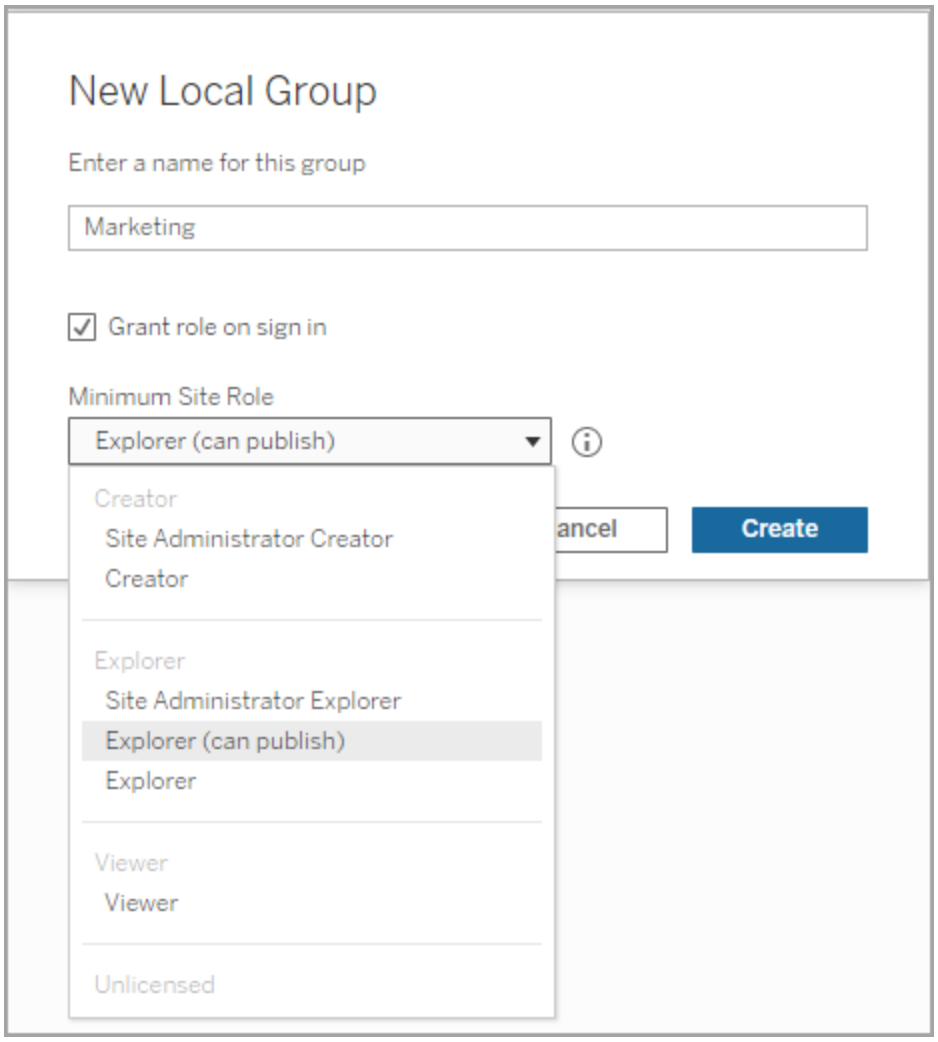
Local groups are created using the Tableau Server internal user management system. After you create a group you can add and remove users, as well as set a minimum site role to grant to users in the group when they sign in.

1. In a site, click **Groups**, and then click **Local Group**.



2. Type a name for the group.

3. To set a minimum site role for the group, select **Grant site role on sign in** and select a minimum site role from the drop-down list.



4. Click **Create**.

Dynamic group membership using assertions

Beginning in Tableau Server 2024.2, if you have SAML authentication configured or use Tableau connected apps for embedding workflows, you can dynamically control group membership through assertions. When configured, at runtime during user authentication, Tableau receives the assertion and then evaluates membership in groups and thus the content whose permissions are dependent on those groups.

The process to dynamically control group membership through assertions requires 1) enabling the setting and 2) ensuring the group membership claims are included in the assertions.

Step 1: Turn on the setting

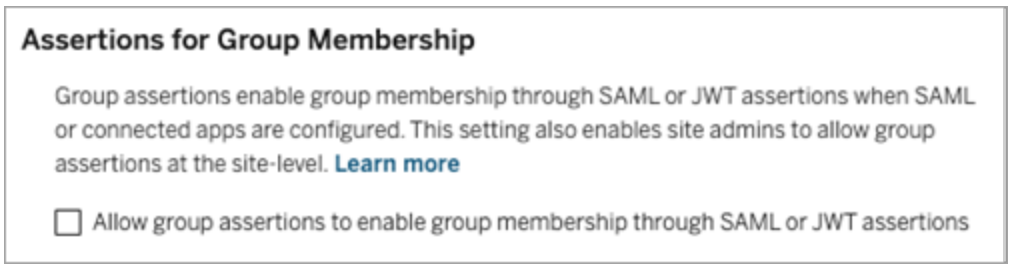
This capability has two settings, a server-wide setting and site-level setting. The site-level setting can only be turned on if the server-wide setting has been enabled first. Consider turning on the site-level setting if you have site SAML or connected apps configured.

For security purposes, group membership is only validated in an authentication workflow if the setting is turned on.

1. Sign in to Tableau Server and navigate to the **Settings** page.

   Note: For a multi-site server, navigate to the **Settings** page for all sites.

2. Under Assertions for Group Membership heading, select the **Allow group assertions to enable group membership through SAML or JWT assertions** check box.



3. (Optional) If you have site SAML or connected apps configured at the site level, navigate to the site, go to the Settings page, and then under Assertions for Group Membership heading, select the **Allow group assertions to enable group membership through SAML or JWT assertions** check box.

For more information about the settings, see one of the following topics:

- Server-wide - Server Settings (General and Customization)
- Site-level - Site Settings Reference

Step 2: Ensure group membership claims are included in the assertion

Two custom group membership claims must be included in the respective SAML, OIDC, or JWT assertion to specify group membership. The two custom group membership claims are:

- Group: `https://tableau.com/groups`
- Group names: These names should match local group names in Tableau Server exactly.

For example assertions, refer to one of the following sections:

- Dynamic group membership using SAML assertions:
- Connected apps - direct trust: Dynamic group membership (embedding workflows only)
- Connected apps - OAuth 2.0 trust: Dynamic group membership (embedding workflows only)

Create Groups via Active Directory

You can import Active Directory (AD) groups to create matching groups on Tableau Server, as well as a user for each member of an AD group that is not already on the server.

Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

Each user is assigned a site role as part of the import process. If any of the users to be imported exist in Tableau Server, the site role assigned during the import process is applied only if it gives the user more access to the server. Importing users does not demote site roles.

Before you begin

Before importing groups, review User Management in Deployments with External Identity Stores to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

Import from AD to add a group

As part of adding a user group to Tableau Server, you import a group from Active Directory (AD). When importing a group, you first enter the group name to search for the group.

Entering a group name, by default, causes Tableau Server to perform a wildcard query (for example, *marketing*) against AD (or LDAP) to maximize the search results. If you're working against a large AD (or LDAP) identity store, you might experience a timeout before you can successfully select the group to import.

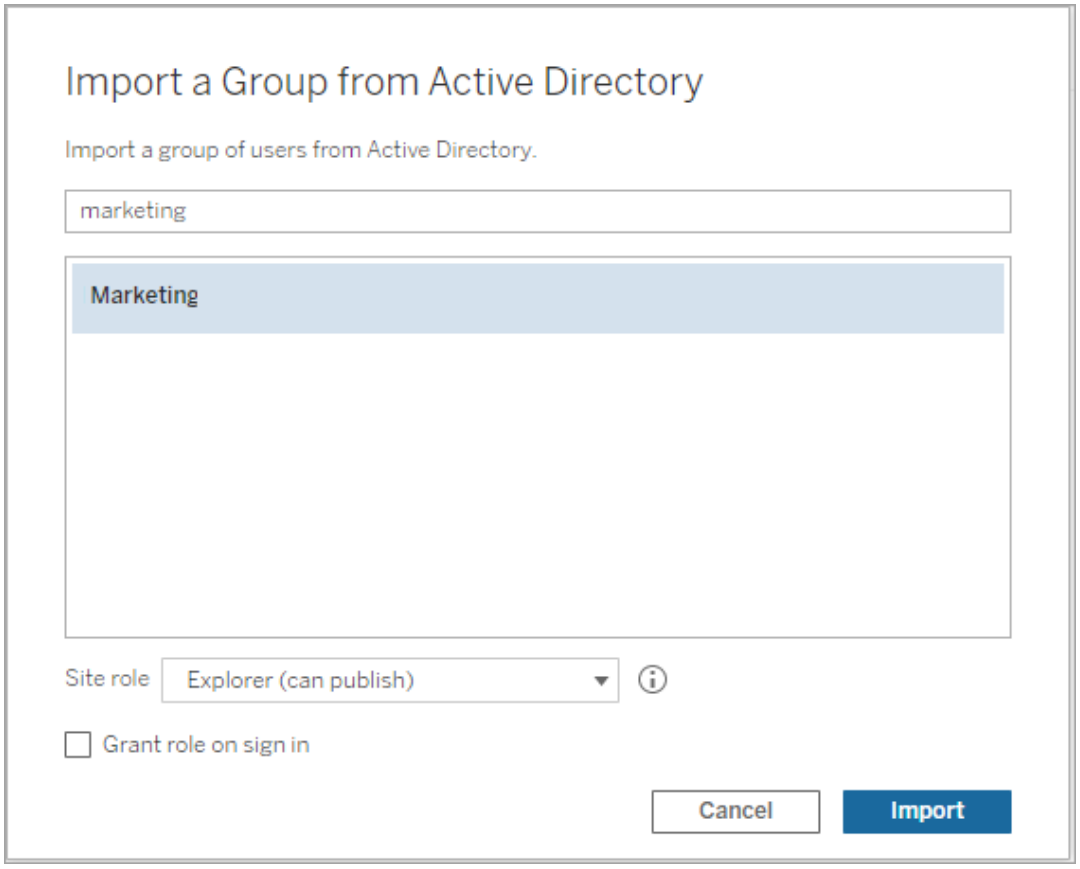
To help save time and avoid potential timeout issues during the import process, consider searching a group name using one of the following methods:

- **Exact match:** The most efficient method for searching a group name, enter the exact group name by including quotation marks (") preceding and following the string you enter. For example, **"marketing"**.
- **Partial match:** Enter a part of the group name and include an asterisk (*) preceding and following the string you enter. For example, ***ket***.
- **Begins with:** Enter the beginning portion of the group name followed by an asterisk (*). For example, **market***.
- **Ends with:** Enter an asterisk (*) followed by the ending portion of the group name. For example, ***ing**.

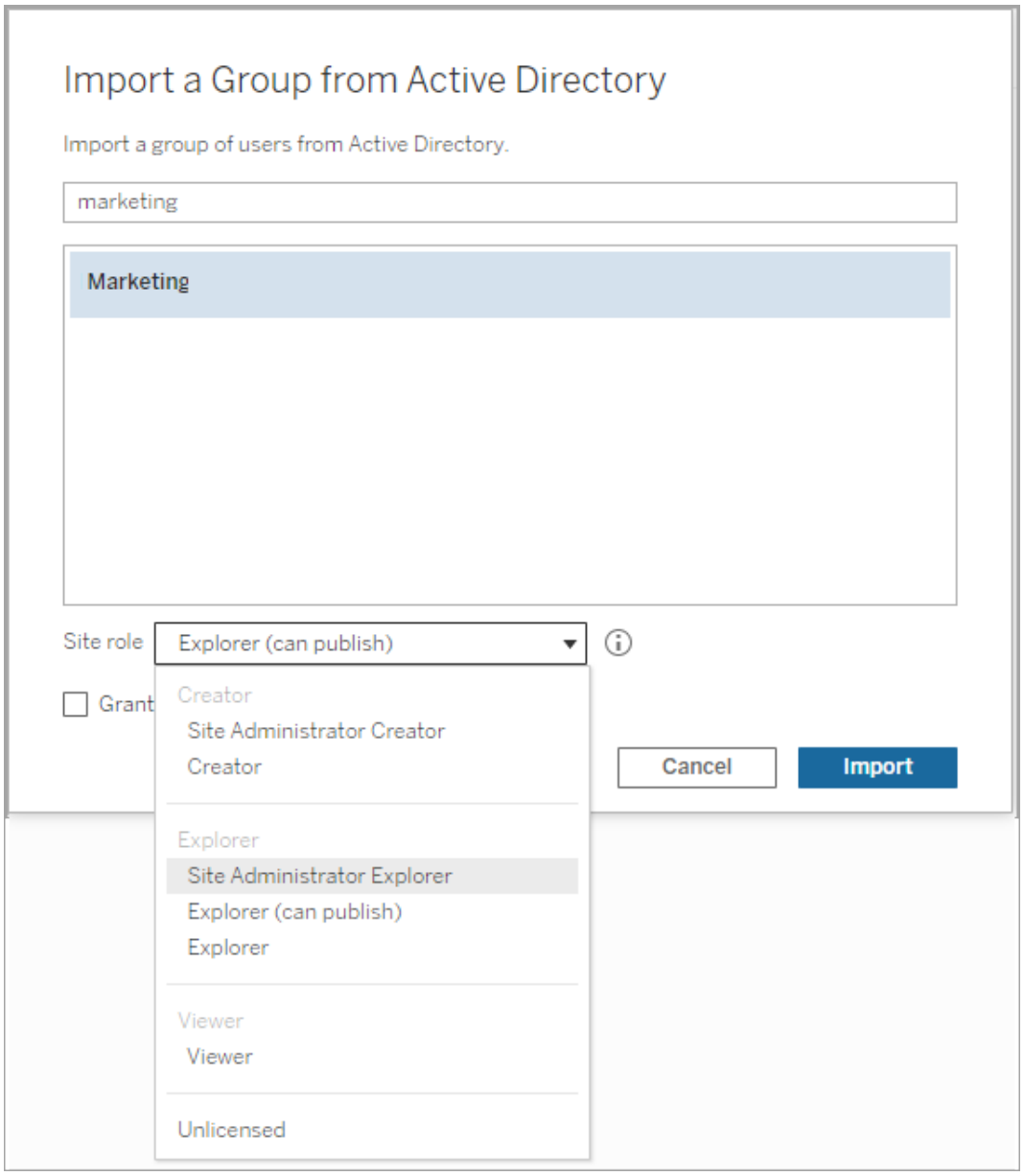
Note: These methods also apply to how Tableau Server searches for users.

1. In a site, click **Groups**, and then click **Add Groups**.
2. Type the name of the Active Directory group you want to import, and then select the group name in the resulting list. Use one of the filtering methods above to improve performance.

If you're importing a group from the same AD domain that the server is running on, you can type the AD group name without the domain. The server's domain will be assumed.



3. Select the minimum site role for the users.



4. (Optional) Select **Grant role on sign in** to provision new site roles and licenses when group users sign in. For more information, see Grant License on Sign In.

5. Click the **Import** button.

Note: You cannot change the name of groups imported from Active Directory. The group name can only be changed in Active Directory.

Synchronize External Directory Groups in a Site

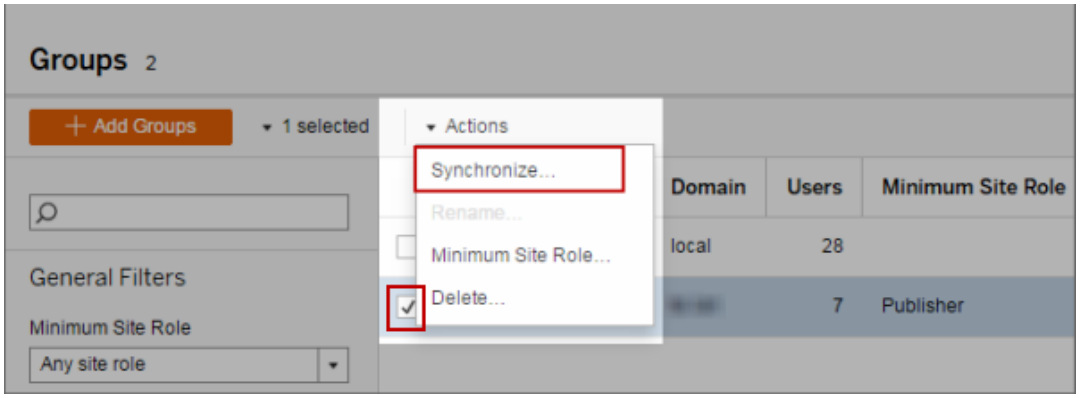
At any time, you can synchronize an external directory (such as Active Directory) group with Tableau Server to ensure new users in the external directory are also added in Tableau Server. You can synchronize individual groups or multiple groups at once.

Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

1. In a site, click **Groups**.

   On the Groups page, select one or more groups.

2. Click **Actions** > **Synchronize**.



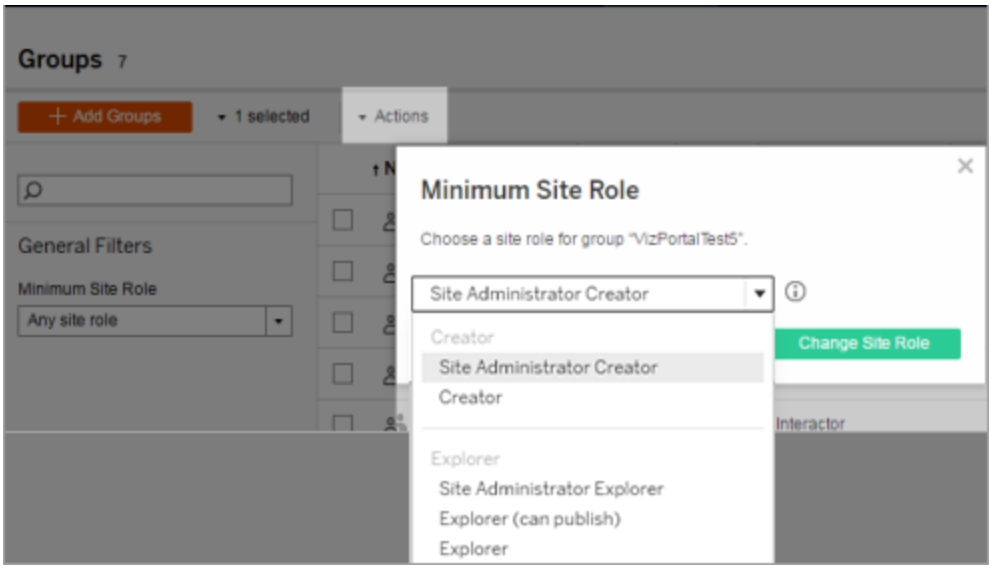
Set the minimum site role for users in an external directory group

In the **Groups - Details** page, administrators can set the minimum site role for group users to apply during synchronization.

This setting does not run synchronization; it sets the minimum site role applied to the group every time synchronization runs. When you synchronize external directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role will be applied if it gives the user more access in a site. If you don't set a minimum site role, new users are added as **Unlicensed** by default.

Note: A user's site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see Site roles and Active Directory import and synchronization.

1. In a site, click **Groups**.
2. On the Groups page, select a group, and then select **Actions** > **Minimum Site Role**.
3. Select the minimum site role, and then click **Change Site Role**.



What happens when users are removed in the source external directory?

Users cannot be automatically removed from the Tableau Server through an external directory sync operation. Users that are disabled, deleted, or removed from groups in the

external directory remain on Tableau Server so that administrators can audit and reassign the user's content before removing the user's account completely. For more information, see Sync behavior when removing users from Active Directory.

What happens when a user name changes in the source external directory

By default, Tableau Server will not synchronize changes to the user display name or email address after the initial synchronization when the corresponding account is created in Tableau Server. For example, if the user name jsmith is used for the display name John Smith, changing the display name in external directory to Joe Smith will not synchronize to the corresponding jsmith user in Tableau Server. Similarly, if the user's email changes in the external directory, Tableau Server will not synchronize changes.

You can configure Tableau Server to update the name and email properties when they change in the source external directory by setting `vizportal.adsync.update_system_user` to `true`.

To change this behavior run the following tsm commands:

```
tsm configuration set -k vizportal.adsync.update_system_user -v true

tsm pending-changes apply
```

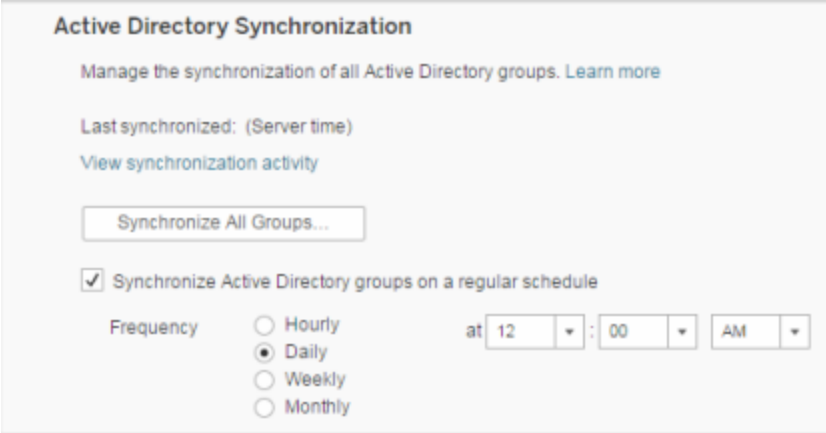
If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

What happens when an external directory group is removed from Tableau Server?

Many Tableau administrators use external directory groups to import and create users. After the users are imported into Tableau Server, administrators will then delete the group in Tableau Server. Deleting a group does not delete the users in it.

Synchronize External Directory Groups on the Server

As a server administrator, you can synchronize all external directory (such as Active Directory) groups (that have been configured on Tableau Server) on a regular schedule or on-demand on the **General** tab of the **Settings** page for the server.



Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

Before you begin

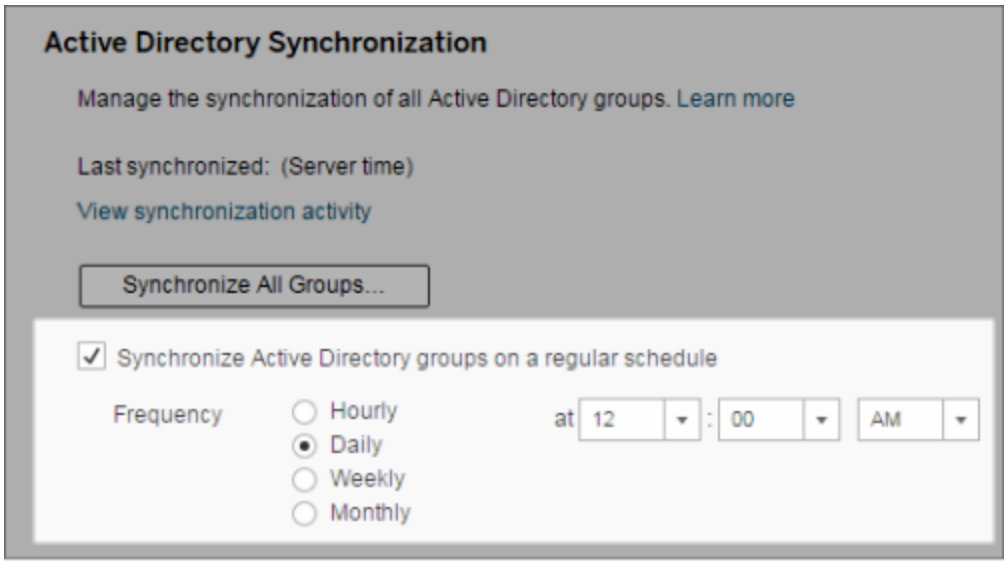
Before synchronizing groups as described in this topic, you must first import the external directory group into Tableau Server. See Create Groups via Active Directory.

Synchronize external directory groups on a schedule

1. **Single-site**: Click **Settings**> **General**.

   Multisite: In the site menu, click **Manage All Sites** and then click **Settings**> **General**.

2. Scroll down the page to **Active Directory Synchronization**, and then select **Synchronize Active Directory groups on a regular schedule**.



3. Select the frequency and time of synchronization.
4. Click **Save**.

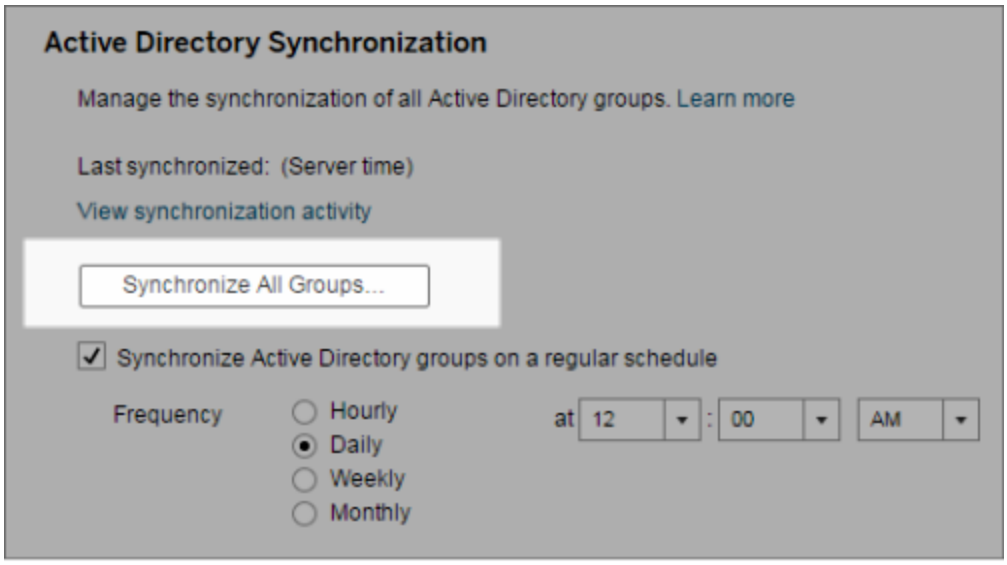
Note: Beginning in versions 2021.1.23, 2021.2.21, 2021.3.20, 2021.4.15, 2022.1.11, 2022.3.3, 2023.1, a default time limit of 4 hours limits how long a scheduled group synchronization can take before it is canceled. A server administrator can change this time limit if your scheduled synchronization is of very large groups, or taking longer than the default. For more information, see Synchronize All Active Directory Groups on a Schedule and backgrounder.timeout.sync_ad_group.

Synchronize all external directory groups on demand

At any time, you can synchronize external directory (such as Active Directory) groups with Tableau Server to ensure that new users and changes in the external directory are reflected in all external directory groups on Tableau Server.

1. **Single-site**: Click **Settings**> **General**.

   Multisite: In the site menu, click **Manage All Sites**, and then click **Settings**> **General**.



2. Under **Active Directory Synchronization**, click **Synchronize All Groups**.

View synchronization activity

You can view the results of synchronization jobs in the **Background Tasks for Non Extracts** administrative view. **Queue Active Directory Groups Sync** is the task that queues and indicates the number of **Sync Active Directory Group** tasks to be run.

1. **Single-site**: Click **Status**.

   Multisite: In the site menu, click **Manage All Sites** and then click **Status**.

2. Click the **Background Tasks for Non Extracts** link.

3. Set the **Task** filter to include **Queue Active Directory Groups Sync** and **Sync Active Directory Group**.

You can quickly navigate to this administrative view by clicking the **View synchronization activity** link in the **Settings** page for the server.

Set the minimum site role for users in an external directory group

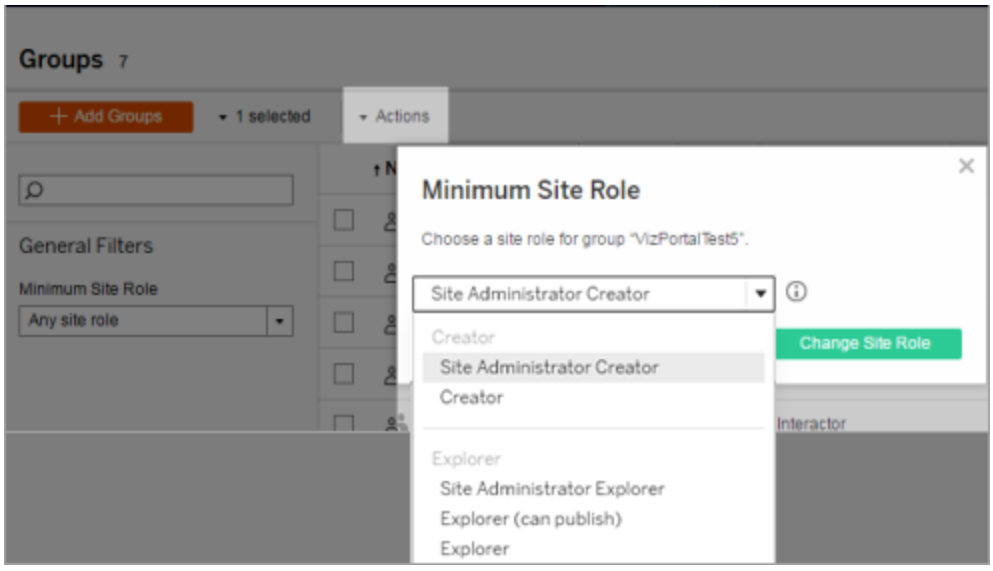
In the **Groups - Details** page, you can set the minimum site role for group users to be applied during Active Directory synchronization.

This setting does not run synchronization; instead, it sets the minimum site role to applied to the group every time synchronization runs. The result is that when you synchronize external directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role is applied if it gives the user more access in a site. If you don't set a minimum site role, new users are added as **Unlicensed** by default.

Note: A user's site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see Site roles and Active Directory import and synchronization.

1. In a site, click **Groups**.
2. On the Groups page, select a group.

   Click **Actions** > **Minimum Site Role**.
3. Select the minimum site role, and then click **Change Site Role**.



What happens when users are removed in the source external directory?

Users cannot be automatically removed from the Tableau Server through an external directory sync operation. Users that are disabled, deleted, or removed from groups in the external directory remain on Tableau Server so that administrators can audit and reassign the user's content before removing the user's account completely. For more information, see Sync behavior when removing users from Active Directory.

Improving group synchronization performance

External directory synchronization is performed by the backgrounder process. The Backgrounder process is the same process that is used for managing and creating extracts, and is also used to generate subscription content. In large organizations with dynamic group membership and heavy extract usage, the external directory group synchronization process may be disruptive. We recommend running group synchronization during non-business hours.

By default, the Backgrounder process performs synchronization in a serial operation. This means that each group is synchronized, one after the other, in a single Backgrounder process. If you are running multiple instances of Backgrounder processes either on a single Tableau Server or across a distributed deployment, consider enabling parallel processing for external directory synchronization. When parallel Backgrounder processing is enabled, the group synchronization is distributed across multiple Backgrounder processes for better performance.

To enable parallel backgrounder processing for group synchronization, open TSM CLI and enter the following commands:

```
tsm configuration set -k backgrounder.enable_parallel_adsync -v
true
```

```
tsm pending-changes apply
```

Synchronize All Active Directory Groups on a Schedule

After you import Active Directory groups in Tableau Server, you can make sure they stay synchronized in Tableau Server by setting up a schedule. You can also synchronize all Active

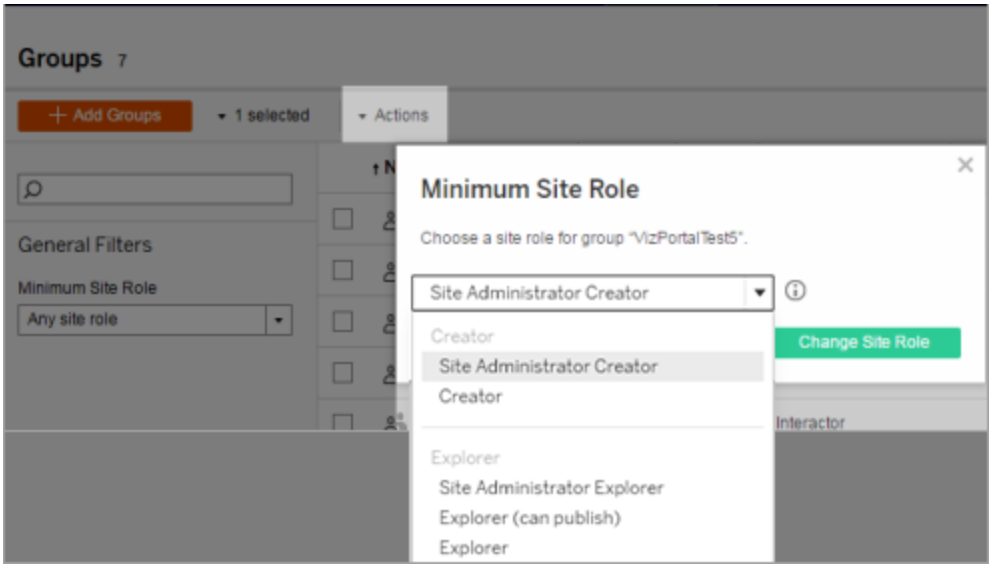
Directory groups on the server on-demand, at any time. The minimum site role setting for the group is applied when users are synchronized.

Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

> Beginning in versions 2021.1.23, 2021.2.21, 2021.3.20, 2021.4.15, 2022.1.11, 2022.3.3, 2023.1, you can set a maximum time limit for how long a scheduled group synchronization can take before it gets canceled. The default time limit is 4 hours. For more information, see backgrounder.timeout.sync_ad_group.

1 Set a minimum site role for synchronization

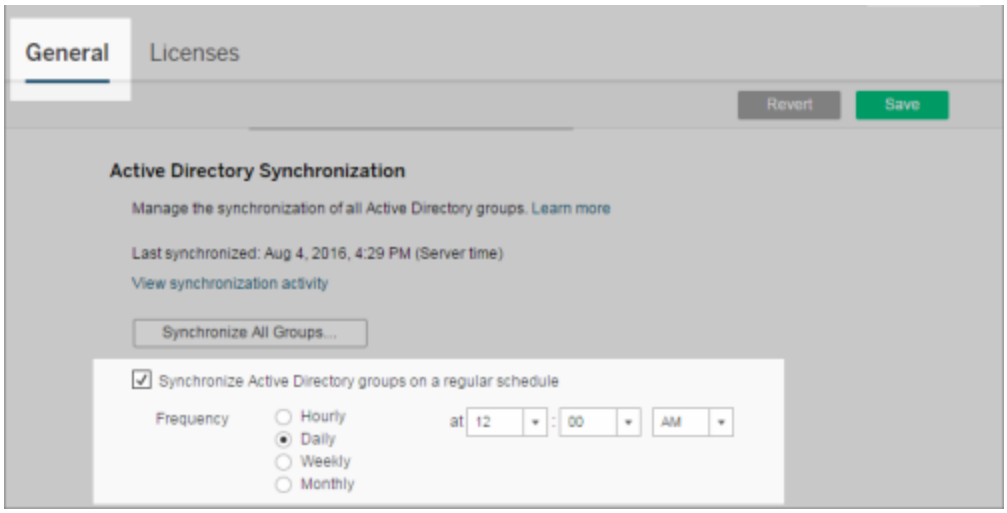
In a site, click **Groups**. Select a group, and then click **Actions**> **Minimum Site Role**. Select the minimum site role, and then click **Change Site Role**. Server and site administrators can set the minimum site role for group users to be applied during Active Directory synchronization. If you don't set a minimum site role, new users are added as **Unlicensed**.



Synchronizing can promote a user's site role, but will never demote a user's site role.

2 Set the schedule

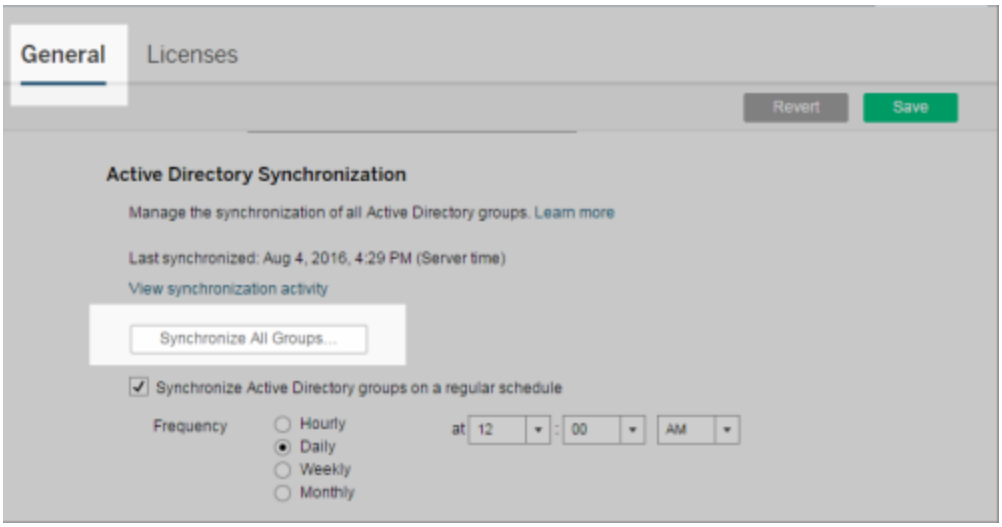
Server administrators can enable synchronization for all Active Directory groups on the **General** tab of the **Settings** page for the server. Enable synchronization, select the frequency settings, and then click **Save**.



All Active Directory groups on the server are synchronized according to the same schedule.

3 Run synchronization on-demand (optional)

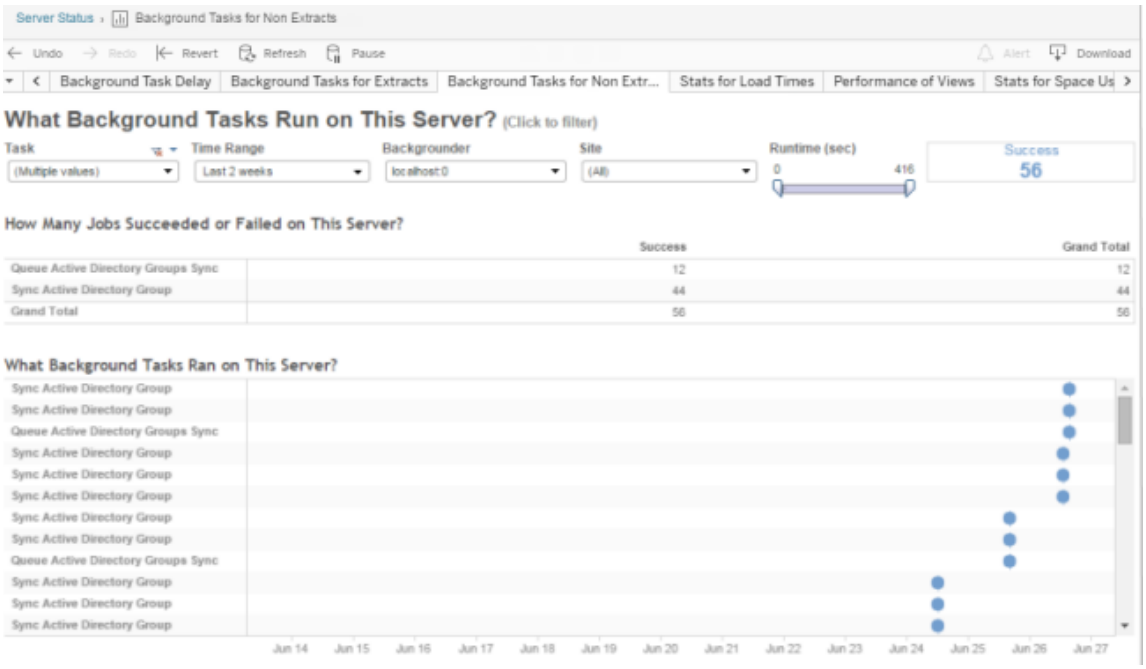
On the **General** tab of the **Settings** page, click **Synchronize All Groups** to synchronize all Active Directory groups on Tableau Server immediately. Click this button at any time to ensure new users and changes are reflected in all Active Directory groups on the server.



Click **Synchronize All Groups** to synchronize all Active Directory groups on the server outside of a schedule.

4 View the status of synchronization tasks

Server and site administrators can view the results of Active Directory synchronization jobs in the **Background Tasks for Non Extracts** administrative view. On the server or in a site, click **Status**. Under **Analysis**, click **Background Tasks for Non Extracts** and filter on the **Queue Active Directory Groups Sync** and **Sync Active Directory Group** tasks.



Queue Active Directory Groups Sync queues the **Sync Active Directory Group** tasks to be run.

Grant License on Sign In

Grant license on sign in (Grant role on sign in) lets unlicensed users in specific groups become licensed when they sign into a Tableau site. This streamlines license provisioning for administrators and removes the user's need to request a license before using Tableau.

For more information about site role capabilities and minimum site roles, see Set Users' Site Roles.

For example, imagine an Active Directory group called Marketing with 100 users, but only 25 users need to access Tableau Server. A site or server administrator can import all users in the Marketing Active Directory group, set the group's minimum site role to Explorer, and select **Grant role on sign in**. When any of the Tableau users in Marketing sign into their Tableau site, they'll be granted Explorer licenses. Users who don't need Tableau Server remain unlicensed unless they sign in.

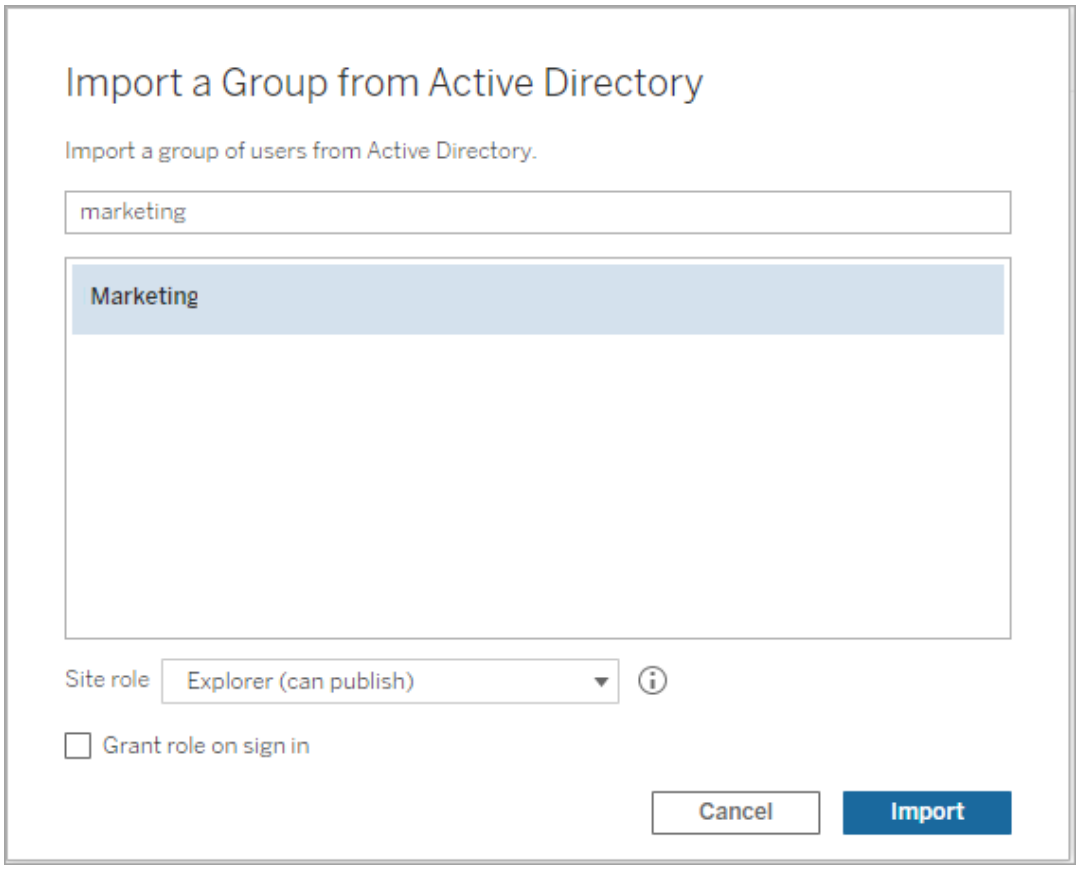
Note: For more information about benefits and best practices, see Grant Role on Sign In in Tableau Blueprint, Tableau's planning tool for data-driven organizations.

Activate Grant role on sign in

You can enable Grant role on sign in on new or existing groups. The following steps walk through how to use Grant role on sign in to add new users that are eligible for a license but may not consume one. This may be the case when your company has a lot of eligible users, but limited Tableau licenses.

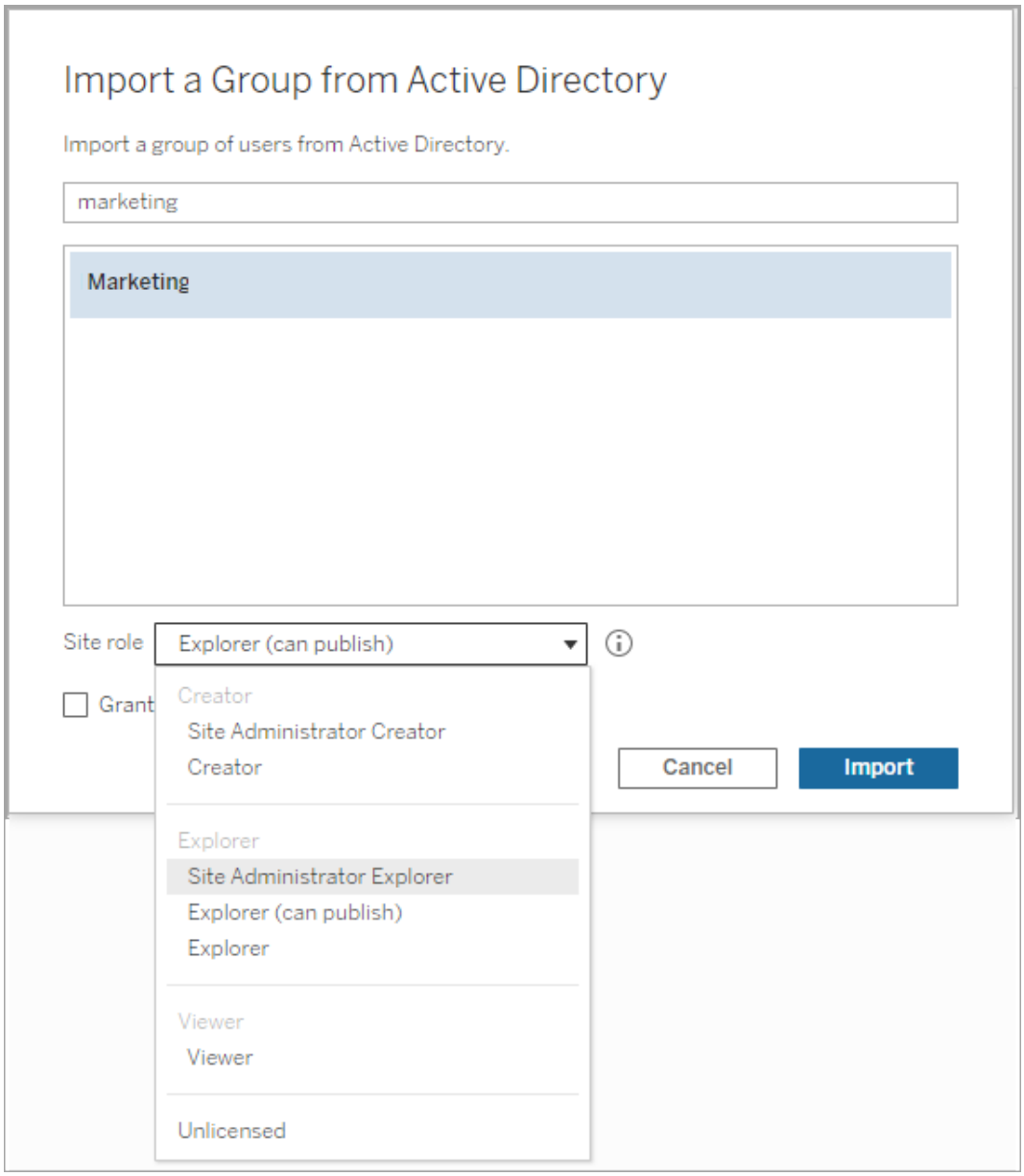
1. On a site, click **Groups**, and then click **Add Group**.

   Add new users by importing an Active Directory group. Type the name of the group you want to import, and then select the group name in the resulting list.



2. Select the minimum site role for the users, and select **Grant role on sign in**.

   All users in the selected Active Directory group will be imported as unlicensed users. The minimum site role set for the group will only be provisioned to group users who sign into Tableau Server.



3. Click **Import**.

> **Note:** Grant Site Role on Sign In can also be activated in local groups to provision minimum site roles to group members when they sign in to Tableau Server. For more information, see Create a Local Group.

Modifying user roles with Grant role on sign in

If a user is part of a group using Grant role on sign in, then that user role can't be set to unlicensed or downgraded to a role lower than the minimum site role set for the group, whether or not they sign in. Administrators can upgrade a user's site role manually, however.

To downgrade a user's site role, or unlicense the user from the site, remove the user from the group(s) that have Grant role on sign in enabled.

In accordance with the terms of the End User License Agreement, licenses granted on an Authorized User basis may be permanently reassigned to new users. Users may only be downgraded to a lower site role (including Unlicensed) when they will permanently discontinue access to Server Software at the higher role.

Removing users affected by Grant role on sign in

You can remove a user from a site only if the user does not own content. If you attempt to remove a user who owns content, the user site role will be set to Unlicensed and removed from all groups, but the user will not be removed from the site. To remove content owners, remove owners from group with Grant site role enabled or reassign content ownership to another user. For more information, see Remove users from a site in the View, Manage, or Remove Users help topic.

If the default All Users group has Grant site role enabled, users who own content can't be removed from the site or unlicensed. To remove or unlicense these users, reassign content ownership to another user, then remove or unlicense the user.

Tableau REST API can be used to reassign content ownership of a workbook. For more information, see Update Workbook method in the REST API Help. REST API can also be used to

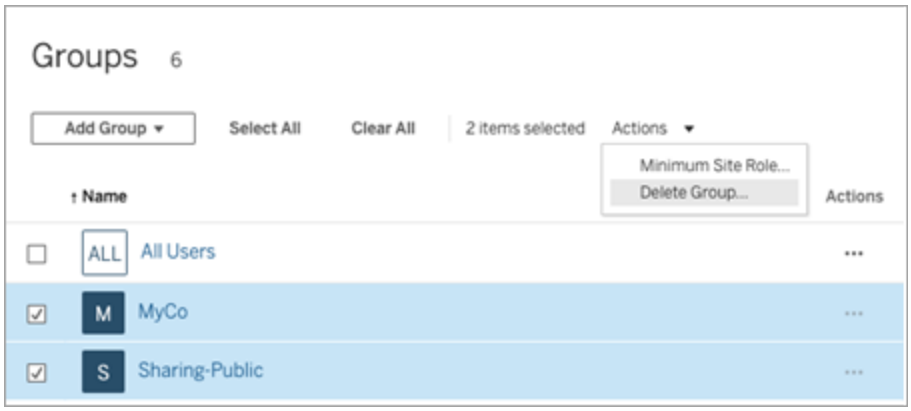
remove users from the site and transfer content ownership to another user. For more information, see Remove User from Site method in the REST API Help.

For more information on changing content ownership in Tableau Server, see Manage Content Ownership.

Delete Groups

You can delete any group from the server with the exception of the **All Users** group. When you delete a group, the users are removed from the group but they are not deleted from the server.

1. Sign in to Tableau Server site as an administrator.
2. From the left navigation pane, click **Groups**.
3. On the Groups page, select one or more groups to delete.
4. Select **Actions** > **Delete**.



Effects of deleting groups

Groups in group sets

Beginning in Tableau Server 2024.2), groups can be added to group sets. When content permissions are dependent on a group set, content capabilities are evaluated when users belong to all groups in the group set. If a group that belongs to a group set is deleted then it can

change user access to Tableau content when content permissions are dependent on the group set.

Work with Group Sets

Beginning in Tableau Server 2024.2, you can create a container for your groups using group sets. A group set can contain one or more groups and be used to apply more granular rules for content permissions that are dependent on the group set. When enabling capabilities based on a group set, users in the groups that belong to the group set must be members of all the groups for the capability to be evaluated. In this way, group sets enforce AND logic.

Benefits of group sets:

- You can mix and match synchronized groups with local groups in permission rules to enable more dynamic access control scenarios.
- Use AND logic for groups in permission rules, which can simplify access control in some scenarios.

Notes:

- Group set permission rules are evaluated after user and group rules. For more information about those rules, see Evaluate permission rules.
- Group sets can only be created by server administrators.

Turn on group sets

Before group sets can be used for permissions, group sets settings must be enabled.

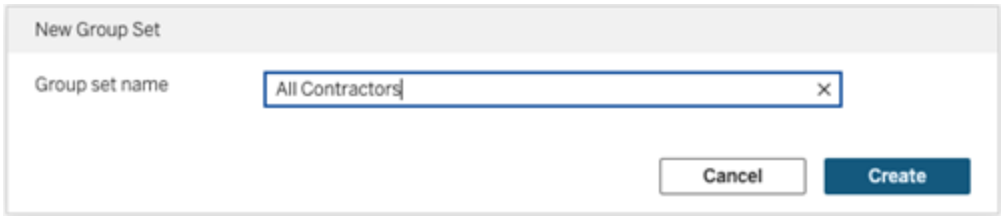
1. Sign in to Tableau Server as server administrator.
2. Navigate to the **Settings** page.
3. Under the Group Sets section, select the **Allow group sets** check box.

After enabling group sets, a dedicated Group Sets page displays in the navigation pane.

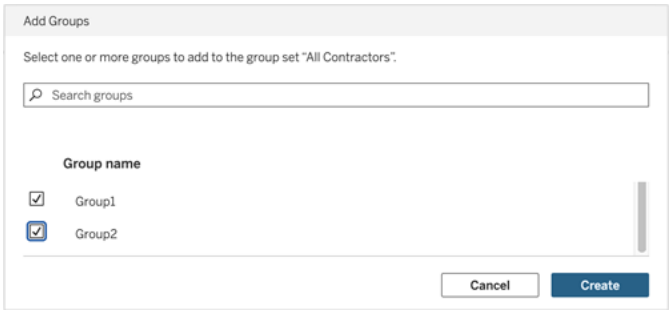
Create group sets

To create a group set, navigate to the Group Sets page and create a group set as you would a group.

1. Sign in to Tableau Server as server administrator.
2. Navigate to the Group Sets page and click the **New Group Sets** button.
3. Enter a name for the group set and click **Create**.



4. In the Group Sets table, click the name of the group set you just created and click the **Add Groups** button.
5. From the list of available groups, select the groups you want to add to the group set and click the **Add** button.



Set permissions on group sets

To use group sets, as a site admin, project leader, or content owner, add or edit the permissions of the content to use the group set.

For example, suppose you are the owner of the "Batters" workbook. To apply permissions based on the group set, do the following:

1. Go to the workbook and select **Permissions** from the actions menu.
2. In the Permissions dialog box, click the **Add Group/User Rule** button, and do the following:
    a. In the text box, enter the group set name, for example "All Contractors."
    b. Select the desired capabilities in the template.
    c. Click **Save**.

When permissions are applied using the group set model, you can enforce a more fine-grained access control.

For example, you might restrict access to different "Batters" workbook views based on a user's regional group affiliation:

- North region view:
    - Group set is called North Region
    - Groups in the group set: All Regions and North Region
- South region view:
    - Group set is called South region
    - Groups in the group set: All Regions and South Region
- East region view:
    - Group set is called East Region
    - Groups in group set: All Regions & East Region
- West region view:
    - Group set is called West Region
    - Groups in the group set: All Regions and West Region

For more information about permissions, see Configure Projects, Groups, Group Sets, and Permissions for Managed Self-Service.

Dashboard-based Custom Portals

Note: This overview was inspired by the work of Tableau Visionary Mark Jackson. For more details about the process, check out Mark's blog.

The standard Tableau Cloud or Tableau Server interface works great for many organizations, but if you want to create a company-branded experience and don't have API developer skills, consider creating a custom portal based on a Tableau dashboard. A custom portal lets you organize content around specific departments or job roles, and you can even incorporate training that helps people interact with data more fully. As the volume of Tableau content grows, a custom portal guides your users directly to the data they need, while providing ready access to related views and Tableau Server search so they can easily explore further.

Sketch out a portal design

Begin outside of Tableau, on paper or in a wireframing application. Consider the structure of your organization and the number of Tableau views that apply to each group of users. Do you simply need one level of navigation that points immediately to content? Or might you need to start with a main page that provides navigational links for separate groups of users or report types, followed by a second level with related dashboards and views?

After you get buy-in on your design from key stakeholders and data consumers, you're ready to move on to the next phase.

Gather images for logos and navigation elements

Start thinking about images while refining your initial mockup, and then gather them from libraries of clip-art or approved brand graphics, or create them from scratch in an application like Photoshop or SnagIt. PowerPoint can also be a good source if you plan to create navigation thumbnails of common chart types.

Lay out text, images, and selected sheets on a dashboard

Create a workbook with a dashboard for the portal, using a tiled layout for more predictable positioning and scaling of elements on different devices. Then start adding Text and Image objects, as well as any sheets for data views you want to highlight right up front in your portal. To refine spacing between these dashboard elements, insert Blank objects and adjust their size. For more information, see Create a Dashboard.

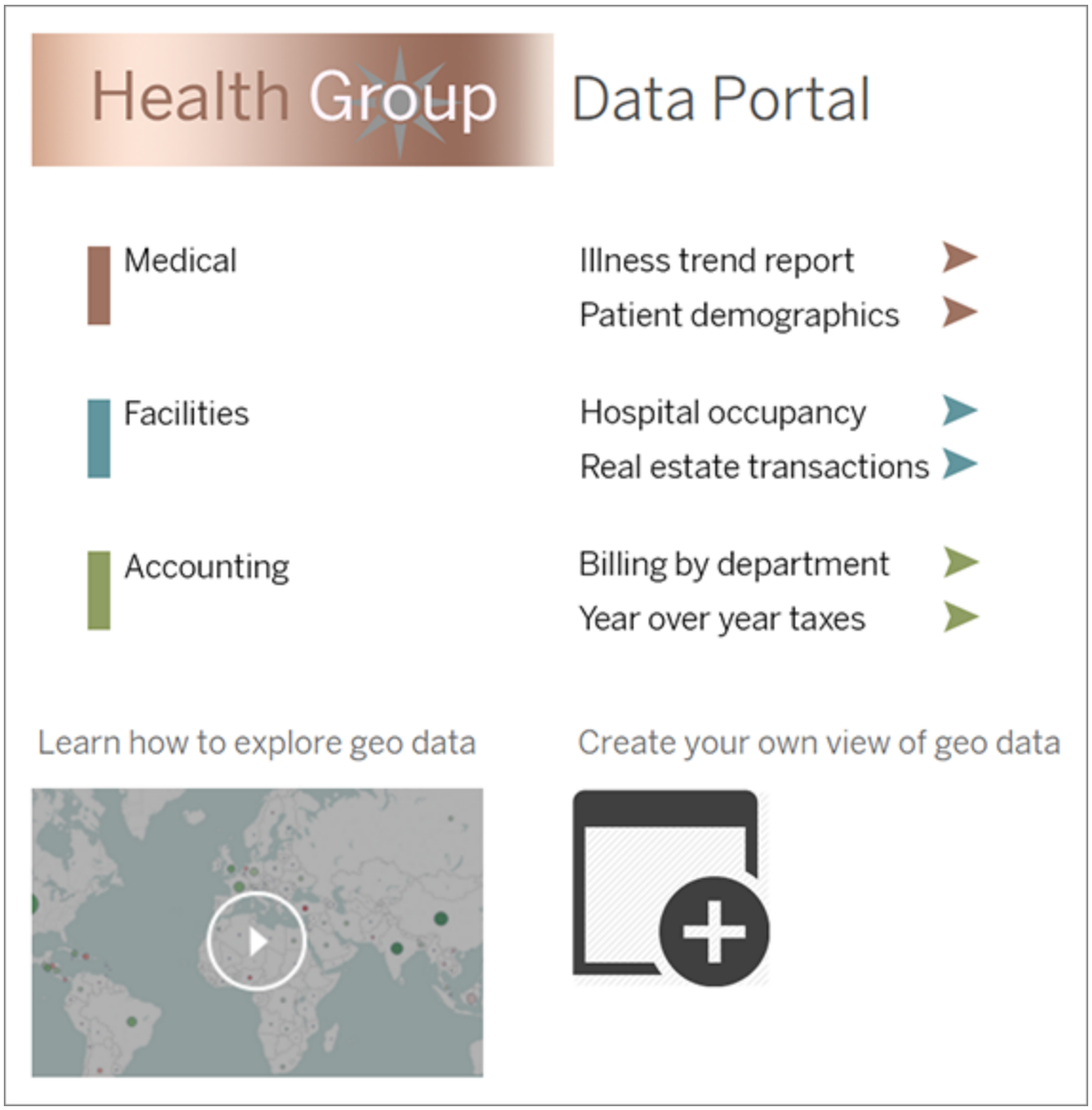
> **Tip:** For dynamic elements that show popular or recently created content, include sheets based on custom administrative views.

Link dashboard elements to content

If you have a second level of navigation in your portal, use filter actions to point to a secondary dashboard from the main one. To create links that directly open data views, right-click Image objects and choose Set URL. (In our example below, each colored arrow and accompanying text is an image that links to a view URL.) You can even point to empty views with preloaded

data sources, encouraging users to create new Tableau content in the web-authoring workspace.

Tip: To add hyperlinks to text objects, include the full URL (for example, `http://www.tableau.com`).



Publish, test, and refine your portal

Publish the workbook to your server and distribute the dashboard URL to your users. Portal design, like data analysis itself, is a cyclical process. Now that your portal is out in the wild, start gathering user feedback so you can continuously improve the experience

> **Tip:** As a finishing touch, hide the Tableau toolbar to give your portal a custom feel. After the question mark at the end of the dashboard URL, add `:embed=y&:toolbar=n`

Manage Content Access

You can manage who can access content on your site and set the permissions that govern content ownership.

Set a Site's Web Authoring Access and Functions

Tableau Server administrators can specify at the site level whether to allow users to edit published views in the web environment and configure other web authoring functionality.

By default web authoring functionality is enabled for all sites. Users with the **Web Edit** capability can create and edit workbooks directly on the server. Turn off web authoring if you want users to be able to view and interact with published workbooks but not make any changes to the core information.

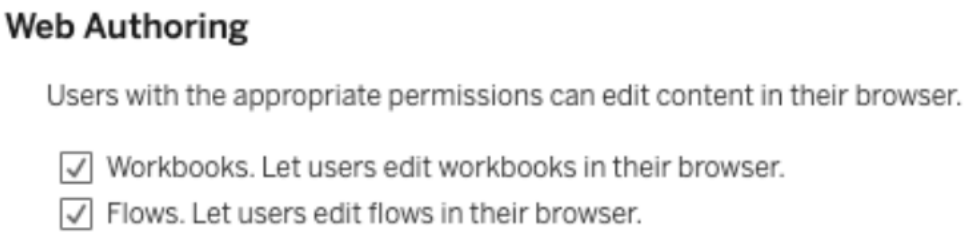
The steps below describe how to set web authoring and other associated functionality for an entire site. For more granular control over which users can use web editing, you can use projects, groups, and permissions. See Set Web Edit, Save, and Download Access on Content.

For information about how to enable authoring for flows on the web, see Create and Interact with Flows on the Web.

Turn web authoring on or off for a site

1. In a web browser, sign in to the server as an administrator and go to the site in which you want web authoring to be enabled. In that site, click **Settings**.
2. In the **Web Authoring** section, select **Workbooks. Let users edit workbooks in their browser.** to enable the functionality.

   Clear the check box to turn off web authoring for that site.

3. If your site is already in production, and you want the change to take effect immediately, restart the server.

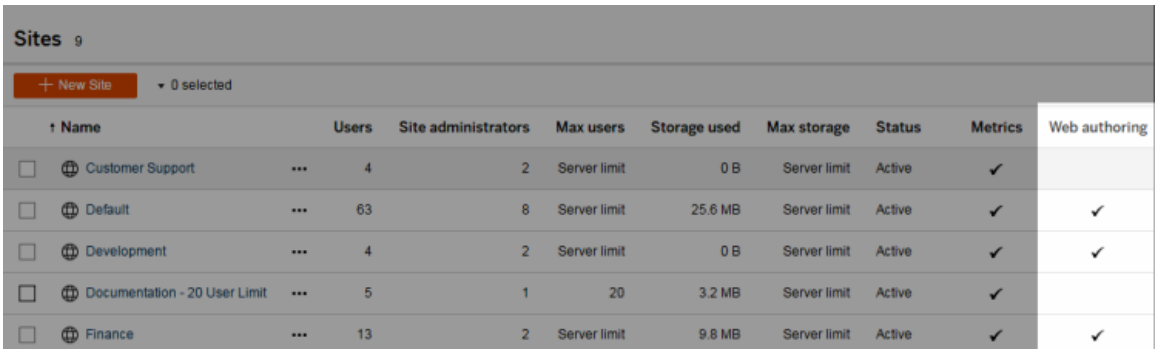
   Otherwise, the change takes effect after server session caching expires or the next time users sign in after signing out.

Notes

- When you enable web authoring, make sure that, on the appropriate workbooks or views, the permission rule for a user or group allows the **Web Edit** capability.
- If you turn off web authoring on a production site and do not complete the last step to restart the server, users might continue to have authoring access until their session caches expire or they sign out.

See which sites allow web authoring

To confirm which sites allow web authoring, on the site-selection menu at the top, select **Manage All Sites**, and then go to the **Sites** page.



About cross-database joins

To improve performance for cross-database joins, Tableau will now default to deciding whether it should perform joins within Tableau using Hyper, or move data into the connected live database as a temporary table and perform joins there.

The option in **Settings** for each site to configure cross-database joins is still visible, but it can no longer be changed from the default.

For more information, see Improve performance for cross-database joins.

Set Web Edit, Save, and Download Access on Content

If you're enabling web authoring functionality on your site, you can configure more precisely which users on the site have access to this functionality. Using site roles and permissions rules at the content level, you can grant or deny **Web edit**, **Save**, or **Download** capabilities on projects, workbooks, and data sources.

Note: This document strives to use the phrase *Web edit* to specify the name of the capability in permissions rules, and *web authoring* to refer to the general functionality of creating and modifying workbooks on the server. However, you might otherwise see these two phrases used interchangably.

Why allow users to work on the site directly

As an administrator, your initial thought about allowing people to populate a site with content, seemingly indiscriminately, might be one of skepticism. However, with a few controls, you can limit where this is done, while providing important benefits that centralized content management offers both you and your users.

Web authoring pros and cons

For publishers and business users, some benefits of web authoring include the following:

- It provides analyst teams who work collaboratively with a central location in which to provide input.
- It enables people who do not have Tableau Desktop to connect to data sources and create workbooks.
- It enables people to access content when they are away from their Tableau Desktop computer or VPN, whether on a computer or a hand-held device.
- It can provide a framework for enabling consistency across Tableau reports.(By making template workbooks available on the site, analysts can download or create new workbooks with data connections, branding, and formatting already in place.

For administrators, benefits can include the following:

- Fewer Tableau Desktop deployments to manage and support.
- Fewer computers that need to have database drivers installed.
- Capacity to govern content.
- More accurate monitoring of what people are doing with Tableau.

Some disadvantages to web editing include the following:

- For analysts, web editing functionality is not as extensive as in Tableau Desktop (although it continues to evolve toward that parity).

- For administrators, more people working on the server might mean upgrading systems.
- Without publishing guidelines, content proliferation on the site is expected.
  This can confuse the people who rely on published Tableau dashboards and data sources, degrade server performance and data quality, and potentially affect data security.

Managing permissions to help users avoid content proliferation

To help users to avoid content proliferation on the site, many Tableau administrators use projects to allow varying levels of access to content. For example, one project can be configured to allow all users to edit and save workbooks; another can allow only approved publishers to save new content.

To get a better idea how this works, see the following resources:

- Configure Projects, Groups, Group Sets, and Permissions for Managed Self-Service
- Governed Self-Service at Scale, a Tableau whitepaper by Rupali Jain.
  To view the PDF, you might need to provide your Tableau website credentials. These are the same ones you use for the community forums or to submit support cases.

Coordinate edit and save capabilities with site roles for the appropriate level of access

To edit, save, and download workbooks, users must have a site role that allows those actions, along with the capabilities—defined in permissions rules—that grant or deny editing-related access.

Site role access

- When the appropriate permissions are set at the content level, the **Creator** or **Explorer (can publish)** site role allows both **Save** (overwrite) and **Save As/Download**.

  Note that **File > Save** is only available to the workbook owner. When the **Save** permission capability has been granted at the project and workbook level, a non-owner user can overwrite the existing workbook in web authoring by selecting **File > Save As** and using the same workbook name. This overwrites the existing content and they become the owner and gain full access to the content.

- The **Explorer** site role can be granted the **Web Edit** and **Save As/Download** capabilities, but they will not be able to save (neither overwriting existing nor saving changes to a new workbook).

For more information, see Web Editing and Web Authoring.

Configure Projects, Groups, Group Sets, and Permissions for Managed Self-Service

Publishing to Tableau Cloud and Tableau Server is easy. For some organizations, it might be a little *too* easy. There is value in creating a controlled framework before letting creators publish their own content.

To keep things tidy and to make sure people can find and access the right content, it may be useful to configure your site for managed self-service. This means having guidelines and settings in place to ensure content is organized, discoverable, and secure without having bottlenecks in the publishing process.

This article lays out a possible path for you as a site administrator to set up your site for managed self-service:

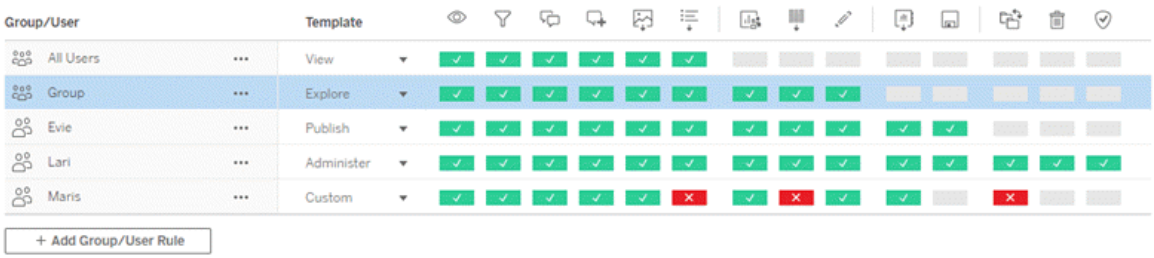
1. Identify the types of groups and projects you'll need
2. Create groups and group sets
3. Remove permissions that will cause ambiguities and establish default permission patterns
4. Create projects
5. Lock project permissions

Note: The information provided here is adapted and simplified from practices of Tableau Visionaries and customers who have shared their experiences.

Plan your strategy

Permissions in Tableau consist of rules that are applied to content (projects, workbooks, etc.) for a group or user. These permission rules are built by allowing or denying specific

capabilities.



Having a comprehensive plan for your projects, groups, and permission rules is useful whether you're starting new or making changes. The details are up to you, but there are two important practices that we recommend for all environments:

- Manage permissions on projects, not individual pieces of content.
- Assign permissions for groups, not individual users.

Setting permissions at the individual user level and on individual content assets becomes unmanageable quickly.

Use a closed permissions model

General models for setting permissions are open or closed. In an open model, users get a high level of access, and you explicitly deny capabilities. In a closed model, users get only the access they need to do their jobs. This is the model security professionals advocate. The examples in this topic follow a closed model.

For more information on how Tableau permissions are evaluated, see Effective permissions.

Identify the types of projects and groups you'll need

Designing a structure to accommodate content (in projects) and categories of users (as groups) or categories of groups (in group sets) can be the most challenging part of setting up a site, but it makes ongoing management much easier.

Projects: Projects function both as a unit for managing permissions and as an organizational and navigational framework. Try to create a project structure that balances how people expect to find content and allows for logical permissioning.

Groups or group sets: Before you create groups it can be useful to find common themes in how people interact with content. Try to identify patterns you can use to create groups or group sets and avoid one-off permissions for individual users.

Example 1: Project and group structure

For example, let's imagine an environment where there is company-wide content that everyone should be able to access, as well as some HR content that needs to be restricted.

Projects include:

- **Acme Corp Conference**. This will include data sources and workbooks for ticket sales, dashboards for content strategy, and project plans for the company conference.
- **Employee Success**. This will include anonymized data sources and workbooks for the internal employee survey
- **Human Resources**. This will include HR data sources and workbooks that should only be available to members of the HR team.

Then, groups should match what people need to do:

- **Core Content Creators**. This group is for users who can publish to top-level projects and have broad access to data sources, but who don't need to be able to move or otherwise manage content.
- **HR Content Creators.** This group is for users who have access to HR data sources and can publish to the HR project.
- **Business Users.** This group is for users who should be able to access the content created by the Core Content Creators, but shouldn't even know the HR content exists.
- **HR Users.** This group is for users who should be able to access content in the HR project but don't have rights to create or publish content.

- **Core Project Leaders.** This group is for users who should be given project leader status on the projects that aren't HR.

Example 2: Group and group set structure

Beginning in Tableau Server 2024.2, you can use group sets to further control the capabilities granted (or denied) to users by enabling permissions at the group set-level. When permissions are set at the group set-level, users must belong to all groups in the group sets to be evaluated.

Note: Group set permission rules are evaluated after user and group rules.

For example, suppose you've created the groups to match what people need from Example 1 above. You can create the following group set to further lock down HR access:

- **HR Leaders.** This group set consists of HR Content Creators and Core Project Leaders. Only if the users in this group set belong to both groups are they given project leader status, ability to access sensitive HR data sources, and publish to the HR project.

Consider site roles

Remember that permissions are tied to content, not groups or users. This means that you can't give a group **Explore** permissions in a vacuum. Rather, the group can be given **Explore** permissions for a project and its content. Site roles, however, are given to specific users and may define or limit the permissions they can have. For more information on how licenses, site roles, and permissions tie together, see Permissions, Site Roles, and Licenses.

Create the groups and group sets

While it might be tempting to create the groups and projects as soon as you identify what you need, it's important to do things in a certain order.

Projects: Projects shouldn't be created until after the Default project has been properly configured (see the next section). This is because top-level projects use the Default project as a template for their permission rules.

Groups: Groups need to be created before they can be used to build permission rules. Users do not need to be added to the groups yet, but they can be. For more information about creating groups and adding users to them, see Groups and Add Users to a Group.

Group sets: Groups need to be created before they can be used to build permission rules. Users do not need to be added to the groups yet, but they can be. For more information, see Work with Group Sets.

Tip: Creating multiple groups and projects and setting permissions manually can get a little tedious. To automate these processes and make them repeatable for future updates, you can perform these tasks using REST API commands. You can use tabcmd commands for tasks such as adding or deleting a single project or group and adding users, but not for setting permissions.

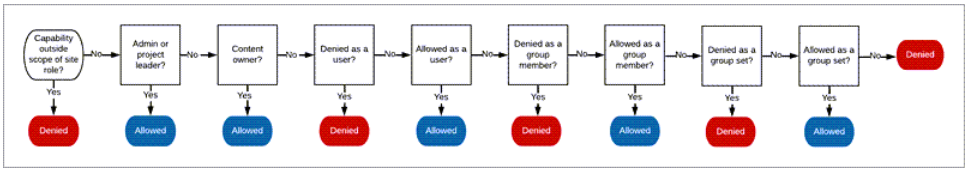
Membership in multiple groups

It's possible to include the users in the HR Content Creators and HR Users groups in the Business Users group. This would make it easy to assign permissions to Core Content Users versus Business Users for the majority of content. However, in that scenario, the Business Users group couldn't be denied any capabilities in the Human Resources folder without denying the HR users as well. Instead, the Business Users group would have to be left as unspecified, and the specific HR Content Creators and HR Users groups would be given their applicable capabilities.

This is because Tableau permissions are restrictive. If the Business Users group was denied certain capabilities, that Deny would override the Allow of another permission rule for users in both groups.

Impact of group sets

If assigned permissions are enabled at the group set-level, permissions for every group in the group set must not be specified or not be denied to allow the capability.



When deciding how group membership should be assigned it's important to understand how permission rules are evaluated. For more information, see Effective Permissions.

Remove permissions that will cause ambiguities and establish default permission patterns

Every site has an **All Users** group and a **Default** project .

All Users group: Any user added to the site becomes a member of the All Users group automatically. To avoid any confusion with permission rules set on multiple groups, it's best to remove the permissions from the All Users group.

Default project: The Default project works as a template for new projects in the site. All new top-level projects will take their permission rules from the Default project. Establishing baseline permission patterns on the Default project means you will have a predictable starting point for new projects. (Note that nested projects inherit the permission rules from their parent project, not the Default project.)

Remove the permission rule for the All Users group on the Default project

1. Select **Explore** to see the top-level projects on the site.
2. On the **Default** project's **Action** (…) menu, select **Permissions**.
3. Next to the **All Users** group name, select **…**, and then select **Delete Rule…**.

This lets you establish permission rules for the groups that you have full control over without any conflicting permissions assigned to All Users. For more information on how multiple rules are evaluated to determine effective permissions, see Effective Permissions.

Create permission rules

Now you can set up the basic permission patterns for the Default project that all new top-level projects will inherit. You may choose to keep the Default project's permission rules empty and build permissions for each new top-level project individually. However, if there are any permission rules that should apply to the majority of projects, it can be helpful to set them on the Default project.

Remember that the permissions dialog for a project contains tabs for each type of content. **You must set permissions for each type of content at the project level** or users will be denied access to that content type. (A capability is only granted to a user if they are expressly allowed it. Leaving a capability as Unspecified will result in it being denied. For more information, see Effective Permissions.)

Tip: Every time you create a permission rule at the project level, make sure you look through all the content type tabs.

Create permission rules as desired:

1. Click **+ Add Group/User Rule** and start typing to search for a group name.
2. For each tab, choose an existing template from the drop-down or create a custom rule by clicking the capabilities.
3. When finished, click Save.

For more information on setting permissions, see Set Permissions.

Example: Project level permissions for each content type

For our example, the majority of projects should be available to most people. For the default project, we'll use the permission rules templates to give the core content creators publishing rights and everyone else the ability to interact with workbooks and not much else.

Group	Projects	Workbooks	Data Sources	(Other content)
Core Content Creators	Publish	Publish	Publish	View
HR Content Creators	View	Explore	View	None
Business Users	View	Explore	View	None
HR Users	View	Explore	View	None
Core Project Leaders	Set as project leader	n/a	n/a	n/a

This pattern follows a closed model and limits permissions to basic usage for most content for most users. As new top-level projects are created, these rules are what will be inherited by default, but the permission rules can be modified per project as needed. Remember that the **Human Resources** project should have these permissions removed and its own pattern established.

Create projects and adjust permissions

After the Default project is set with your custom permissions templates, you can create the rest of your projects. For each project, you can adjust the default permissions as appropriate.

To create a project

1. Select **Explore** to see the top-level projects on the site.
2. From the **New** dropdown, select **Project**.

3. Name the project and, if desired, give it a description.

It can be useful to establish a naming convention. For example, a basic structure might be <DepartmentPrefix><Team> - <ContentUse>; such as DevOps - Monitoring.

The description appears when you hover over a project thumbnail and on the **Project details** page. A good description can help users know they're in the right place.

4. Adjust permissions as necessary.
    a. Open the new project.
    b. From the Action menu (...), select Permissions
    c. Modify any permission rules as desired. *Remember to check all the content tabs*.

Lock content permissions

In addition to permission rules, projects have a content permission setting. This setting can be configured in two ways, either **Locked** (recommended) or **Customizable**.

Locking a project is a way of maintaining consistency and ensuring that all content in the project has uniform permissions (per content type). A customizable project permits authorized users set individual permission rules on pieces of content. For more information, see Lock content permissions.

Regardless of the content permission setting, permissions are always enforced on content.

Possible project structures

Some organizations find it useful to have projects that serve specific purposes. Here are some example projects and their intended uses. Note that these are example templates and you should always test the configuration in your environment.

For information about what capabilities are included with each content type's permission rule templates, see Permission capabilities.

Examples: permission settings for specific purposes

Workbooks shared for open collaboration on the server

Anyone in the department can publish to the open-collaboration project while their content is in development. Colleagues can collaborate using web editing on the server. Some people call this a sandbox, some call it staging, and so on. On this project you can allow web editing, saving, downloading, and so on.

Here you want not only to enable collaboration, but also to enable people who don't have Tableau Desktop to contribute and provide feedback.

Group	Projects	Workbooks	Data Sources	(Other content)
Data Stewards	Publish	Publish	Publish	*TBD*
Analysts	Publish	Publish	Explore	*TBD*
Business Users	Publish	Publish	Explore	*TBD*

Remember that some capabilities in the Publish template (such as Overwrite) may be prevented by a user's site role even if they are allowed that capability.

Note: "*TBD*" indicates these permission rules aren't easily determined by the scenario and can be set however makes sense for a given environment.

Shared reports that cannot be edited

This could be a project that people who create workbooks and data sources (Analysts and Data Stewards) could publish to when they want to make content available to business users for viewing, with confidence that their work cannot be "borrowed" or modified.

For this type of project, you would deny all capabilities that allow editing or getting the data off of the server for reuse. You would allow viewing capabilities.

Group	Projects	Workbooks	Data Sources	(Other content)

Data Stewards	Publish	*TBD*	Publish	*TBD*
Analysts	Publish	Publish	View	*TBD*
Business Users	View	View	None	None

Vetted data sources for Analysts to connect to

This would be where Data Stewards publish the data sources that meet all of your data requirements and become the “source of truth” for your organization. Project leaders on this project can certify these data sources, so that they rank higher in search results and are included in recommended data sources.

You would allow authorized Analysts to connect their workbooks to data sources in this project, but not download or edit them. You would deny the view capability to the Business Users group for this project, so those users would not even see this project.

Group	Projects	Workbooks	Data Sources	(Other content)
Data Stewards	Publish	*TBD*	Publish	*TBD*
Analysts	View	None	View	None
Business Users	None	None	None	None

Inactive content

Another possibility is to segregate workbooks and data sources that the site’s administrative views show haven’t been used for a period of time. You could give content owners a time limit before their content is removed from the server.

Whether you do this or delete directly from the working projects is up to your organization. In an active environment, don’t be afraid to be intentional about removing content that is not being used.

Group	Projects	Workbooks	Data Sources	(Other content)
Data Stewards	None	None	None	None
Analysts	View	View	*TBD*	*TBD*
Business Users	None	None	None	None

Source for workbook templates

This is a project that people can download from but not publish or save to, where authorized publishers or project leaders make template workbooks available. Templates that have your organization's approved fonts, colors, images, and even data connections built in can save authors a lot of time and keep your reports looking consistent.

Group	Projects	Workbooks	Data Sources	(Other content)
Authorized Author	Publish	Publish	Publish	*TBD*
Data Stewards	None	None	None	None
Analysts	View	*Template*: Explore + *Capability*: Download Workbook/Save a Copy	View	None
Business Users	None	None	None	None

Next steps

Besides projects, groups, and permissions, other data governance themes include:

User education

Help *all* of your Tableau users become good data stewards. The most successful Tableau organizations create Tableau user groups, have regular training sessions, and so on.

For a common approach to orienting users to the site, see Dashboard-based Custom Portals.

For publishing and data certification tips, see the following topics:

- Use Certification to Help Users Find Trusted Data
- Prepare for Publishing a Workbook (links to Tableau Help)
- Best Practices for Published Data Sources (links to Tableau Help)

Optimize extract refresh and subscription activity

If you use Tableau Server, create policies for extract refresh and subscription schedules, to avoid them dominating the site's resources. The TC customer presentations by Wells Fargo and Sprint address this subject in detail. In addition, see the topics under Performance Tuning.

If you use Tableau Cloud, see the following topics to become familiar with the ways people can refresh extracts:

- Keep Data Fresh
- Use Tableau Bridge to Expand Data Freshness Options

Monitoring

Use administrative views to keep an eye on the site's performance and content use.

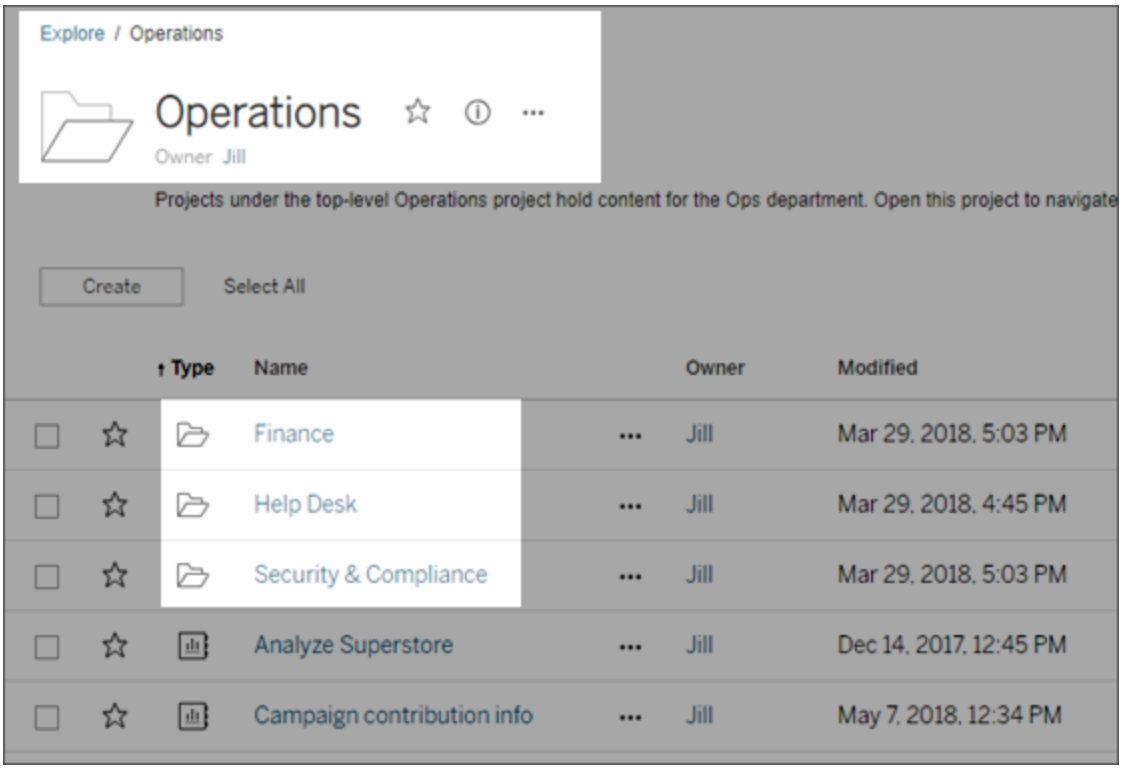
Administrative Views

Use Projects to Manage Content Access

When Tableau Desktop users publish content to a site on Tableau Server, they can select a *project* to publish it to.

Projects can be used for navigation, organization, and access management for assets like workbooks, data sources, lenses, and nested projects. Starting in Tableau Cloud October 2022 / Server 2022.3, if Data Management is licensed and Catalog is enabled, a project can also contain external assets like databases.

The following image shows content within the top-level Operations project in the web authoring environment. The Operations project contains a few nested projects (highlighted) and published workbooks. A project can also contain other asset types.



Why use projects

Projects help you to create a scalable process for managing access to the content published to Tableau Server. Advantages they have include:

- They enable administrators to delegate content management to project leaders who work with the content more closely, without having to give them administrator access to site or server settings.
    - Project leaders can create nested projects under their top-level project, enabling them to maintain their team's content within a single hierarchy.
    - **Note:** Project owners can delete top-level projects they own. Project leaders cannot delete top-level projects.
- They can make the site easier to navigate for self-service users.
    - They segment the Tableau Server site into areas that give users access based on how they use the data published to those areas, or on the Tableau user group they work with.
    - You can hide projects from groups who don't need to use them, create a distinguishable project-naming scheme, and take advantage of project descriptions to clarify how to use the project.
- They enable you to track permissions effectively.
    - You can create groups based on the level of content access users in the group need, and set default permissions on projects. This enables you to know exactly which capabilities new users get by default, and likewise which capabilities all users get when a new project is created.

When to create project hierarchies (example)

Many organizations have several or more distinct groups of Tableau users, each with its own priorities and leaders. These groups might share some organization-wide content (or even draw from an org-wide pool of data sources), but primarily they use data and reports that are specific to their team. In this or similar scenario, an example for using project hierarchies might look as follows:

1. You, as a site or server administrator, can create top-level projects for each of your distinct Tableau teams.
2. On each top-level project, you assign the Project Leader status to team leads, and change project ownership. Project leaders effectively are the content administrators,

so it's important that they understand how permissions work in Tableau, along with Tableau content management best practices.

3. Each project leader can manage their project, creating the structure within the project that works for their team. That is, they can create child projects they need, based on how their team members collaborate and share data and reports.

The benefit to you as the site administrator is that you can focus on system health. The benefit to your Tableau users is that people who know the best practices for working with Tableau and data can manage these things for their teams, without having to submit IT requests to change permissions or add projects.

Why not use sites?

Sites work well when content can remain completely separate during all phases, and there is little to no user overlap. A good (and common) example for using multiple sites is to create a site for each of multiple external clients, whose published content you manage as a consultant or vendor.

Projects allow the flexibility you need to administer shared data and reports, and users who might belong to multiple groups. Projects work better than sites for evolving content from development to staging to production.

Project-level administration

For more information about administering projects, see Manage Permissions with Projects.

Add Projects and Move Content Into Them

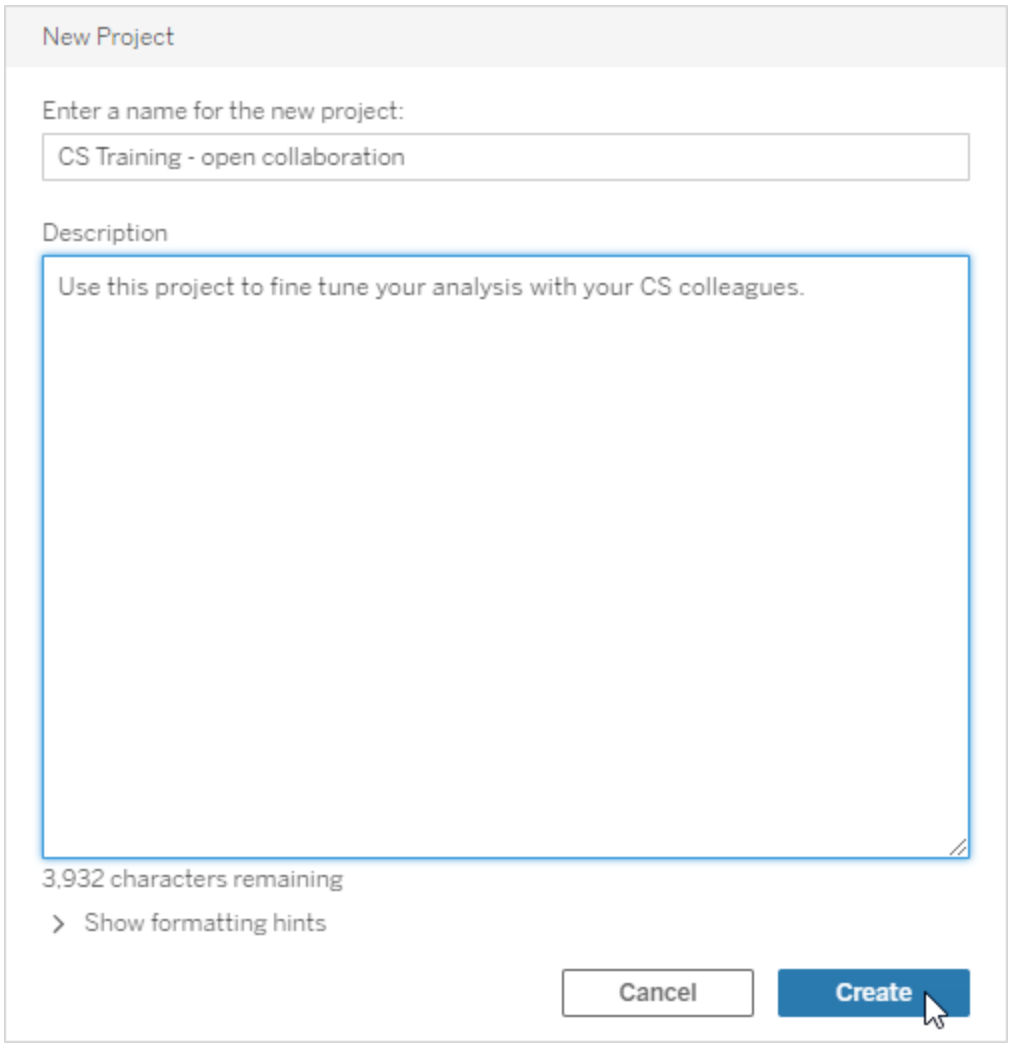
Tableau content (such as workbooks or data sources) must be in a project. Starting in Tableau Server 2022.3 and Tableau Cloud October 2022, if Data Management is licensed and Catalog is enabled, external assets (such as databases and tables) can also be in projects. Server and site administrators can add or remove top-level projects on a site, and move published content from one project to another. Project leaders with appropriate site roles can add or remove child projects and move content between projects on which they have Project Leader access.

This article contains the steps for creating and moving projects. We recommend becoming familiar with the following related content as well:

- To learn about projects and when or why to use them, see Use Projects to Manage Content Access.
- Before you create project hierarchies, become familiar with Permissions.
- To see the specific site roles that allow full Project Leader access, see Project-level administration.

Add a top-level or child (nested) project

1. While you're signed in to Tableau Server as an administrator or project leader, select **Explore**, and then do one of the following:
    - Select **New** > **Project** to create a new top-level project (only administrators can do this).
    - Navigate to and open the project in which you want to create a sub-project, and then select **New** > **Project**. If you're not sure where to find the child project, select **All Projects** from the drop-down menu next to **Explore**, or use the filters in the upper right.
2. Enter a name and description for the project, and then click **Create**.



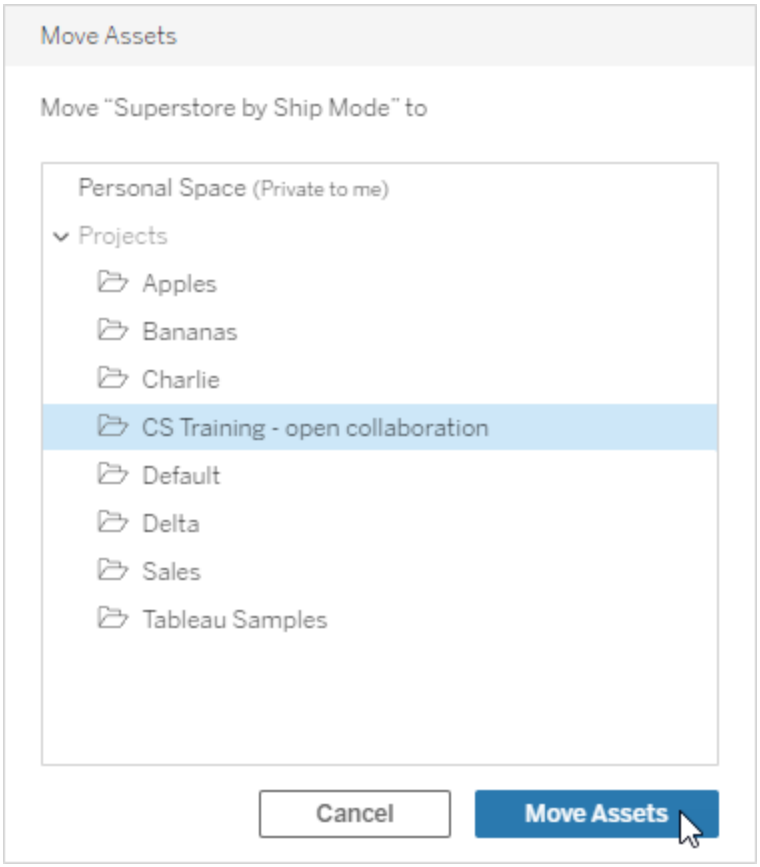
You can include formatting and hyperlinks in the project description. Select **Show formatting hints** for syntax.

When entering a project description, include a space between capital letters and parentheses to display content inside them, such as "PROJECT (a)". Omitting this space will cause display issues in the project description.

Note: To edit a project description later, select it to open it, select the information icon next to its name, and then click **Edit**.

Move an asset to another project

1. In the **Explore** section, find the asset you want to move. You can use the filters in the upper right to search, or you can navigate through the project hierarchy.
2. On the workbook's **Actions**(**…**) menu, select **Move**.
3. Select the new project for the workbook, and then click **Move Content**.



Moving a project includes moving everything in it, including child projects and their assets.

How moving projects affect permissions

When you move a project, Project Leader permissions adapt to the new project environment.

- When the target project hierarchy is **Locked**, previous Project Leader permissions are removed, and new Project Leader permissions are granted according to those set at the top-level of the target hierarchy.
- When the target project hierarchy is **Customizable**, previous implicitly granted Project Leader permissions are removed, explicitly set Project Leader permissions are retained, and new Project Leader permissions are granted according to those set at the top-level of the target hierarchy.

When you move a project and assets, permissions may be impacted. For more information, see Permissions.

Delete a project

When you delete a project, all of the Tableau content in the project is also deleted. If you want to delete a project but not its content, move the content to another project, and then delete the project.

External assets, such as databases and tables, are not deleted, but are moved to the **External Assets Default Project**. (In Tableau Server 2022.3 and earlier, the assets can be found in **External Assets**.)

Important

- You cannot undo deleting a project.
- Deleting a project deletes all Tableau content in it, including child projects and their content, but not external assets.
- You cannot delete the **Default** project or the **External Assets Default Project**.

To delete a project:

1. In the **Explore** section, find the project you want to remove. If you're not sure where to find the project, select **All Projects** from the drop-down menu next to **Explore**, or use the filters in the upper right.

2. On the project's **Actions** (**…**) menu, select **Delete**.
3. Confirm that you want to delete the project.

Requirements for moving assets

Moving an asset is effectively like removing it from one project and publishing it to another. For non-administrators, the permissions needed on the source project are different from those needed on the destination project.

Required site role

To move assets, users must have one of the following site roles:

- Server Administrator (Tableau Server only)
- Site Administrator Creator or Site Administrator Explorer
- Creator or Explorer (can publish)

Users with a Server Administrator or Site Administrator site role do not need any additional capabilities.

Required permissions for the project that users move content *to*

Non-administrators must have the **Publish** permission capability for the destination project.

Required permissions for the project that users move content *from*

Non-administrator users must

- Be the project owner, project leader, or content owner for the original project

  OR

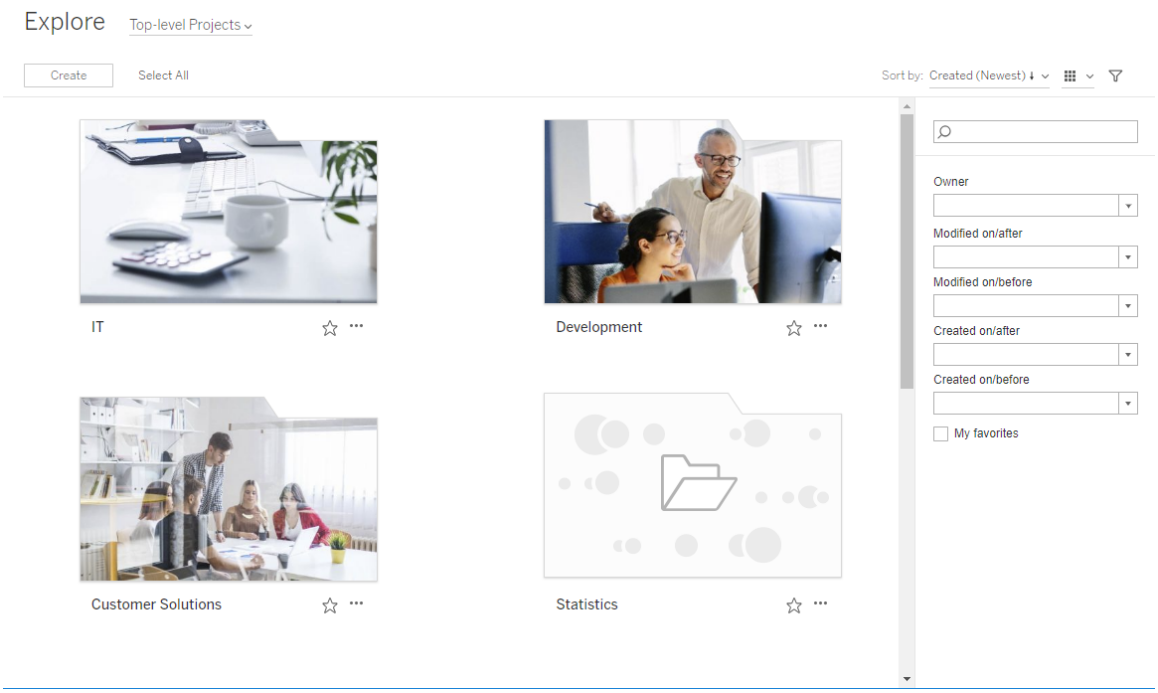
- Have the **Move** permission capability for the content (or, for data sources, be the data source owner). When moving a database with its tables, the user must have the Move capability for both the database *and* its tables.

For more information on moving assets, see Move content.

Add a Project Image

To help distinguish projects you manage on Tableau Server (and help your users find them), you can add an image that appears in the thumbnail. Your image must meet the following requirements:

- The image must be accessible using HTTPS protocol. Shared network directory and related protocols (UNC, SMB, AFP, NFS, etc) are not supported. HTTP protocol for project images is not supported by Google Chrome.
- All users who access the project must have, at a minimum, "read-only" permission on the target image.
- The image must be common internet format: .jpg, png, or gif.

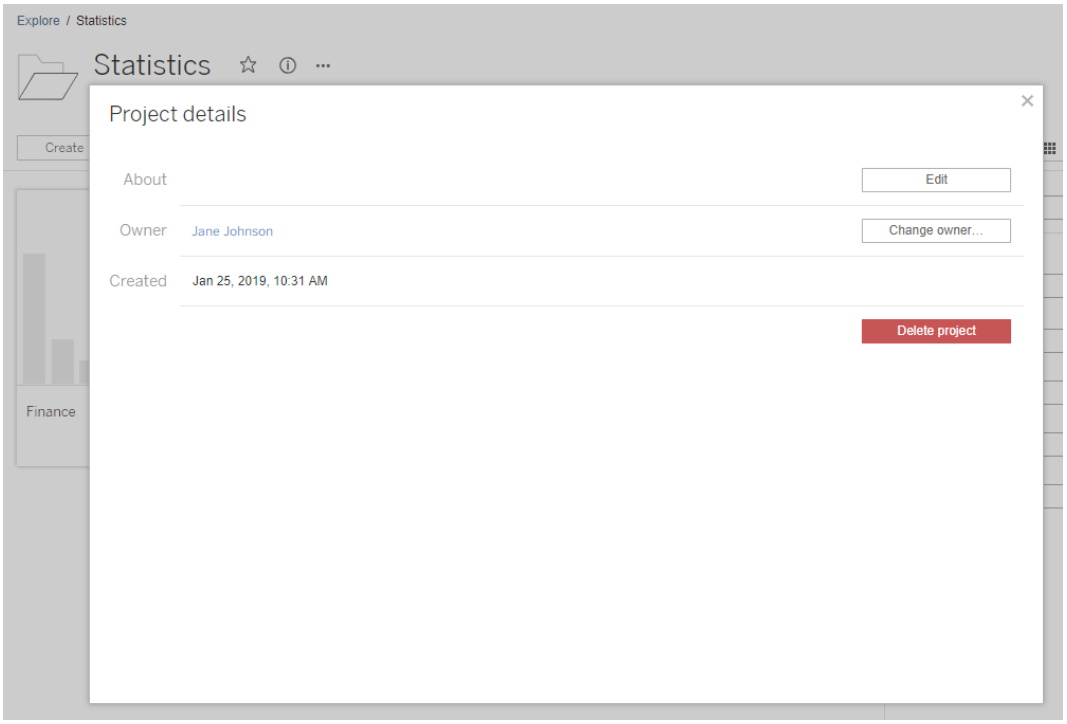


Set a project image

1. Sign in to a site on Tableau Server. In the list of **Top-level Projects** you have access to, select or navigate to the project you want to update. In this example, we'll add an image to the Statistics project folder.

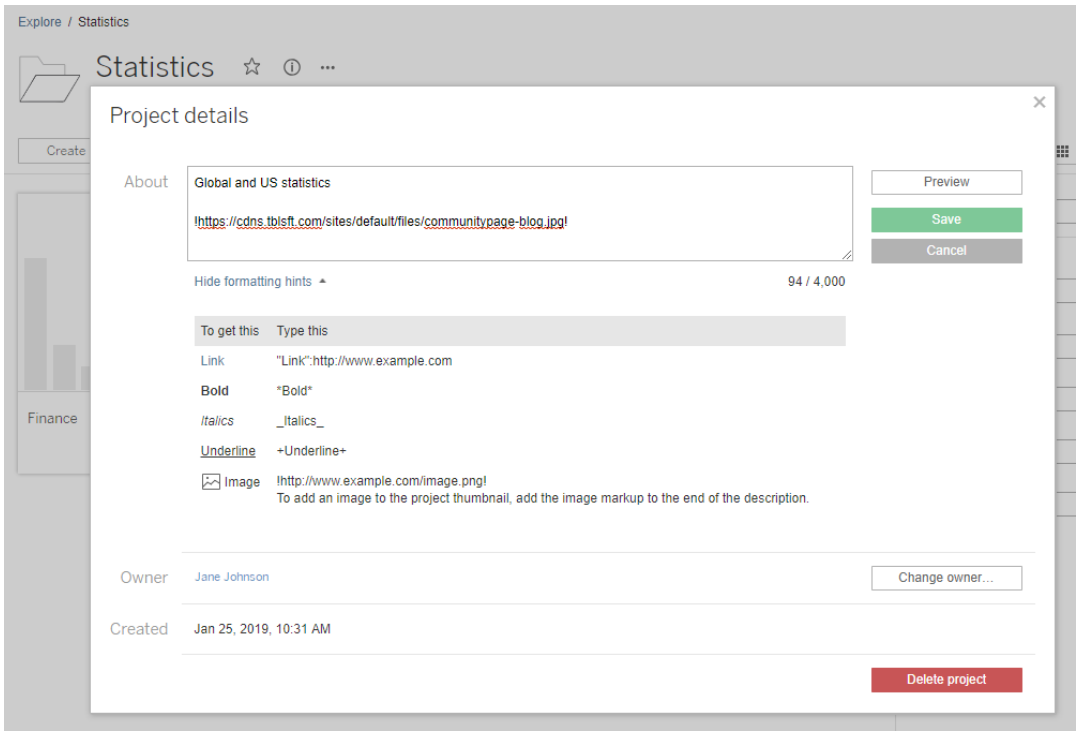
If you're not sure where to find a child project, use the **Explore** drop-down list and select **All Projects**.

2. Click the **Details** icon (i), to open the **Project details** dialog box, and then click **Edit**.



3. In the **About** field, you can enter a description for your project (optional), for example "Global and US statistics." At the end of the project description, add the URL for your image using the following syntax:

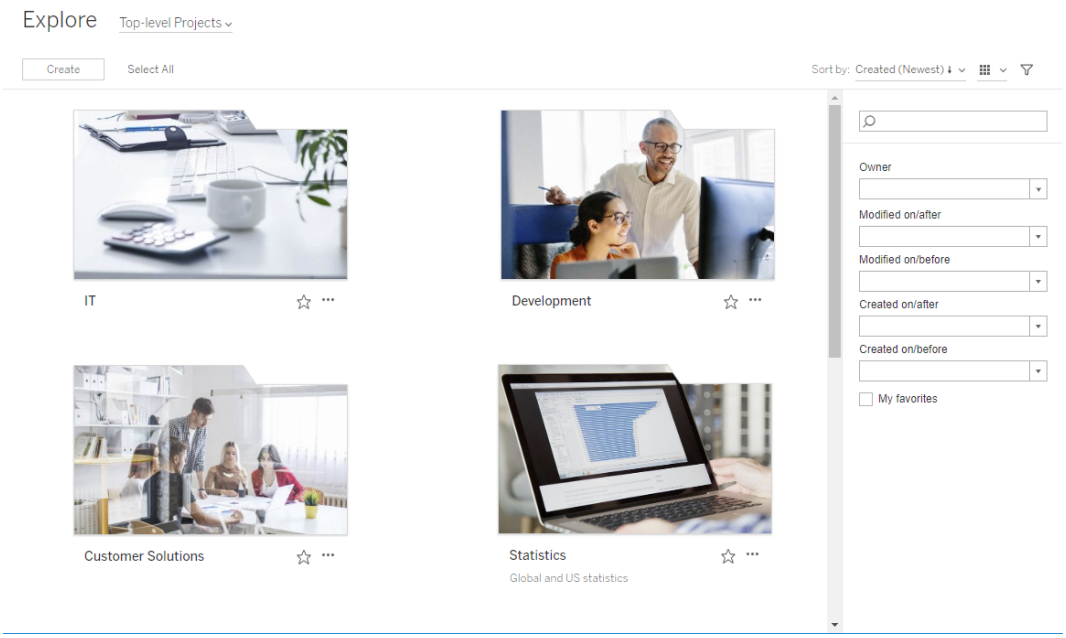
```
!http://www.example.com/image.png!
```



Select **Show formatting hints** to see how you can format description text.

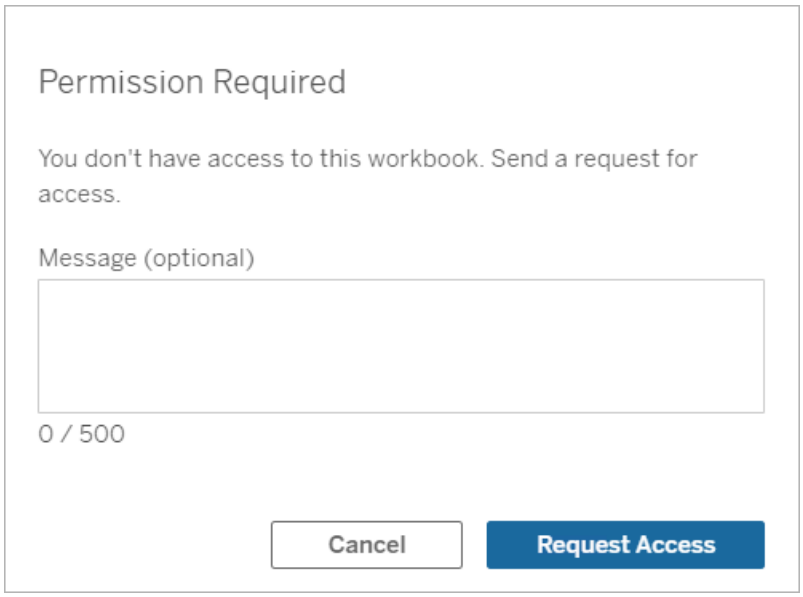
> **Note:** Images embedded in project descriptions cannot be resized or positioned. Recommended size is (300 x 184 pixels). Images that are not 300 x 184 pixels may be stretched, shrunk, or cropped to fit the width of the thumbnail. In addition, they must be added at the end of the project description and be enclosed in **!** (exclamation marks), otherwise they will not be displayed as the thumbnail.

4. Click **Save**.



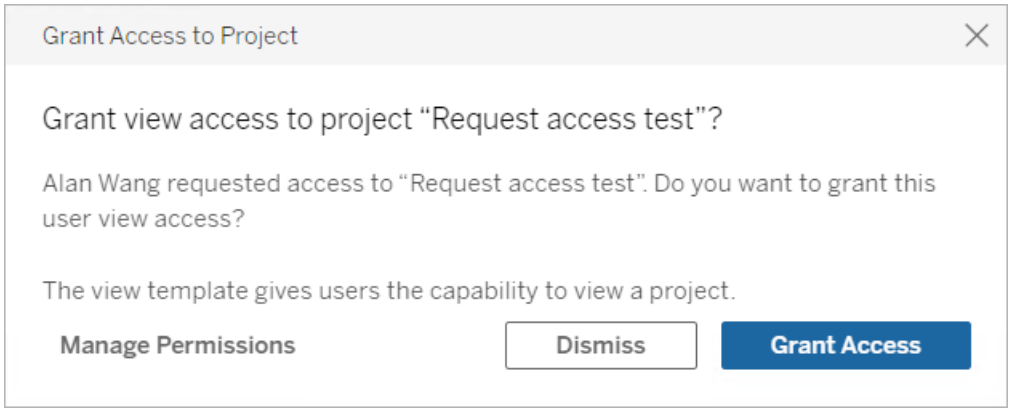
Let Site Users Request Access to Content

Permissions determine if a user has viewing access to a workbook, view, or other content inside a project. If a user clicks on content or a project they don't have access to, they can send a request for access to the owner who controls permissions for that content.



When someone requests access, the owner who controls permissions for that content (either at the project or workbook level) receives an email with the name and email of the requester, the content or project requested, and a link to grant access to the content. In Tableau Server version 2022.3 and earlier, the owner receives a link to the content to manage permissions instead of a link to directly grant access.

1. On the email notification, select **Grant Access**.
2. On the dialog that appears, to grant the view permissions template, select **Grant Access**. To grant permissions other than the view template, select **Manage Permissions**.



If a user requests access to a workbook and content permissions are locked to the project, then the project owner receives the request. Likewise, if a user requests access to a workbook and project permissions are managed by the workbook owner, then the workbook owner receives the request.

After permission is granted, the owner can email the requester to let them know they have view capability to the project or workbook.

Default settings

The Request Access setting is enabled by default on a new site. To enable the setting if it's been disabled:

1. Go to the General tab of the Settings page for your site.
    - If you have a single site, on the side navigation, click **Settings** and **General**.
    - If you have multiple sites, select the site you want to configure and click **Settings** and **General**.
2. On the General tab, scroll down to Request Access and select **Let users request access to projects, workbooks, and views**.
3. Click **Save**.

Configure project permissions

You can control who receives the access request by adjusting the project's content permissions. If content permissions are:

- Locked to the project: the project owner receives the request.
- Managed by the owner: The workbook owner receives the request.

To manage content access using projects, see Use Projects to Manage Content Access and Permissions.

For more information about how permission rules are evaluated, see Permissions: Evaluate permission rules.

Change project permissions

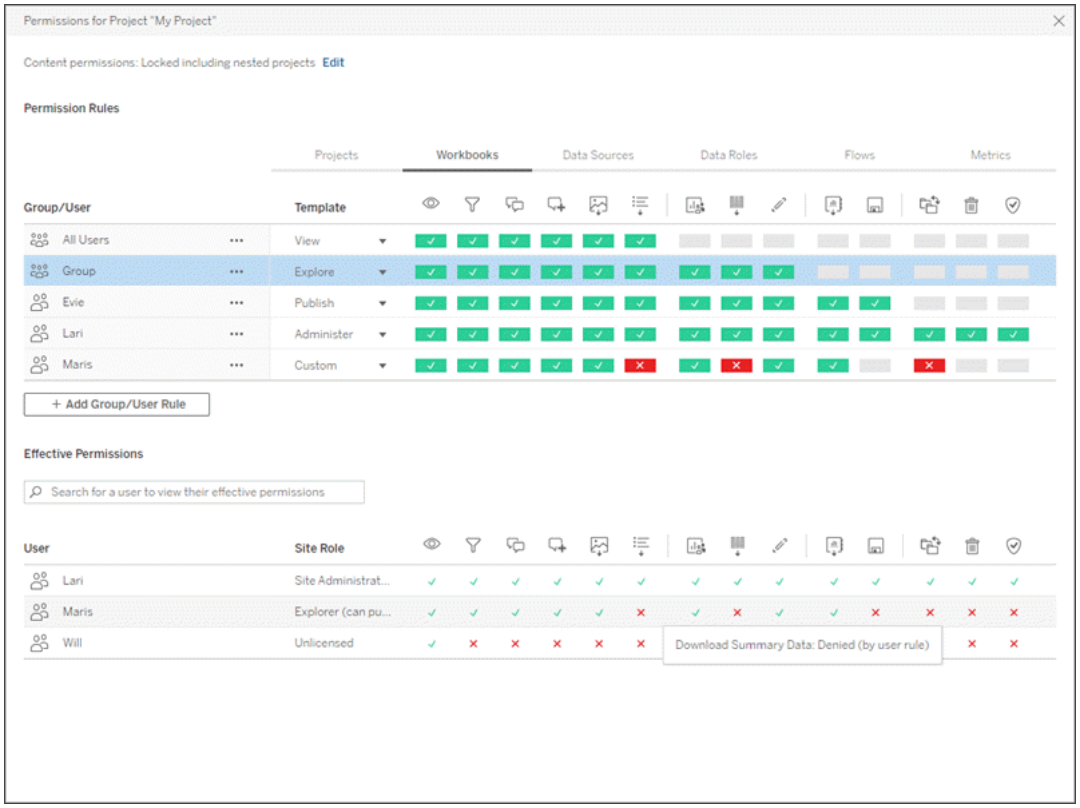
For administrators and project leaders

Permissions can be set at the project level for both the project itself and for any content in the project. For example, if workbook permissions are configured at the project level, all workbooks published into that project inherit those default permissions. However, the Creator can choose to change the permissions during publishing, or certain users can change the permissions on published content. To enforce the permissions established at the project level, **Content Permissions** can be locked to the project. For more information, see Lock asset permissions.

To set permissions at the project level:

1. Navigate to the project
2. Open the Actions menu (...) and click **Permissions**. The permissions dialog box opens.

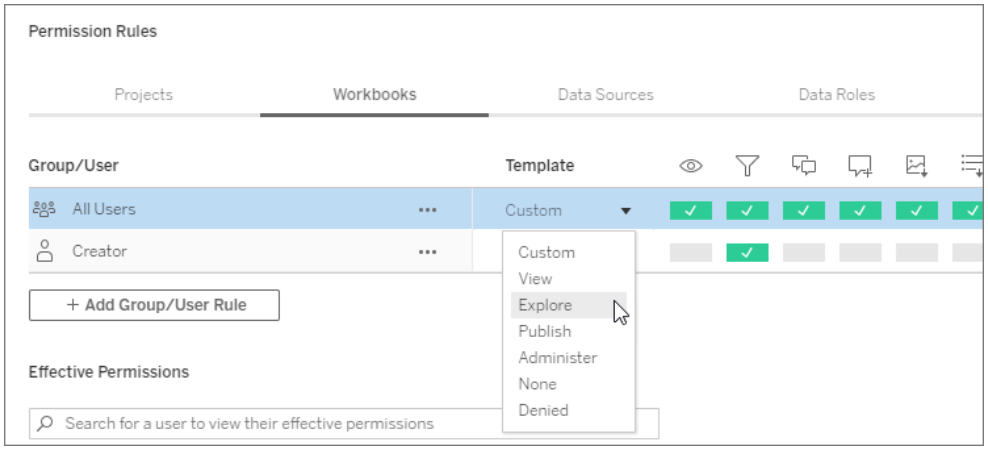
   This dialog box has two main areas: permission rules at the top and the effective permissions grid below. Use the tabs to navigate between types of content.



   With a row selected at the top, the effective permissions grid populates. Use this to verify permissions. Hovering over a capability indicator provides information about why the capability is allowed or denied for that specific user.

3. To modify an existing permission rule, select the rule and click the capability boxes to toggle through allowed/denied/unspecified.
4. To create a new rule,
   a. Select **+ Add Group/User Rule**.
   b. Select a group or user from the drop-down box. This creates a row where you can configure the permission rule.

5. In the row for the permission rule
   a. choose an existing permission role template from the drop-down box for each content type tab.



   b. Or create a custom rule by navigating to a content type tab and clicking the capabilities. One click sets the capability to **Allowed**, two clicks sets it to **Denied**, and a third click clears the selection (**Unspecified**).
6. When finished, click **Save**.

Change content permissions

For administrators, project leaders, and content owners

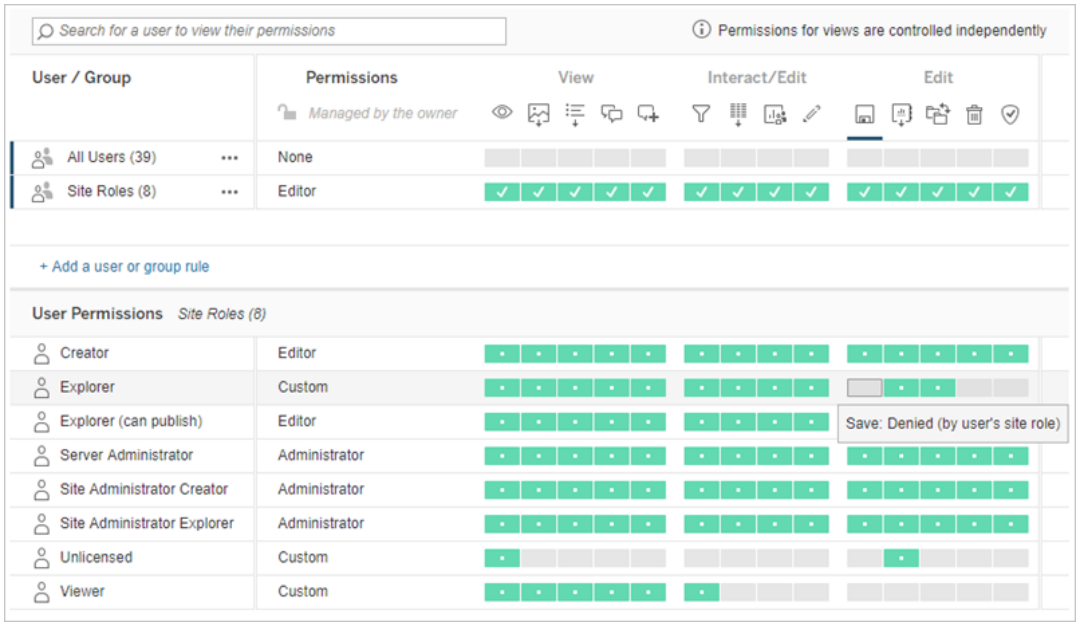
If project permissions are not locked, permissions for individual pieces of content can be modified.

Warning: Tableau recommends managing permissions at the project level within the Tableau site. These steps are relevant only for content in projects where permissions are managed by the owner.

Set permissions on content

1. Navigate to the content (workbook, data source, flow, data role)
2. Open the Actions menu (...) and click **Permissions**. The permissions dialog box opens.

This dialog box has two main areas: permission rules at the top and the effective permissions grid below.



With a row selected at the top, the effective permissions grid populates. Use this to verify permissions. Hovering over a capability square provides information about why the capability is allowed or denied for that specific user.

3. To modify an existing permission rule, open the Actions menu (...) for that row and click **Edit**.
4. To create a new rule,
    a. Select **+ Add a user or group rule**.
    b. If necessary, use the drop-down box on the right to change between groups and users.
    c. Select a group or user from the drop-down box. This creates a row where you can configure the permission rule.
5. In the row for the permission rule, choose an existing permissions role template from the drop-down box or create a custom rule by clicking the capabilities.

    One click sets the capability to **Allowed**, two clicks sets it to **Denied**, and a third click clears the selection (**Unspecified**).

6. When finished, click **Save**.

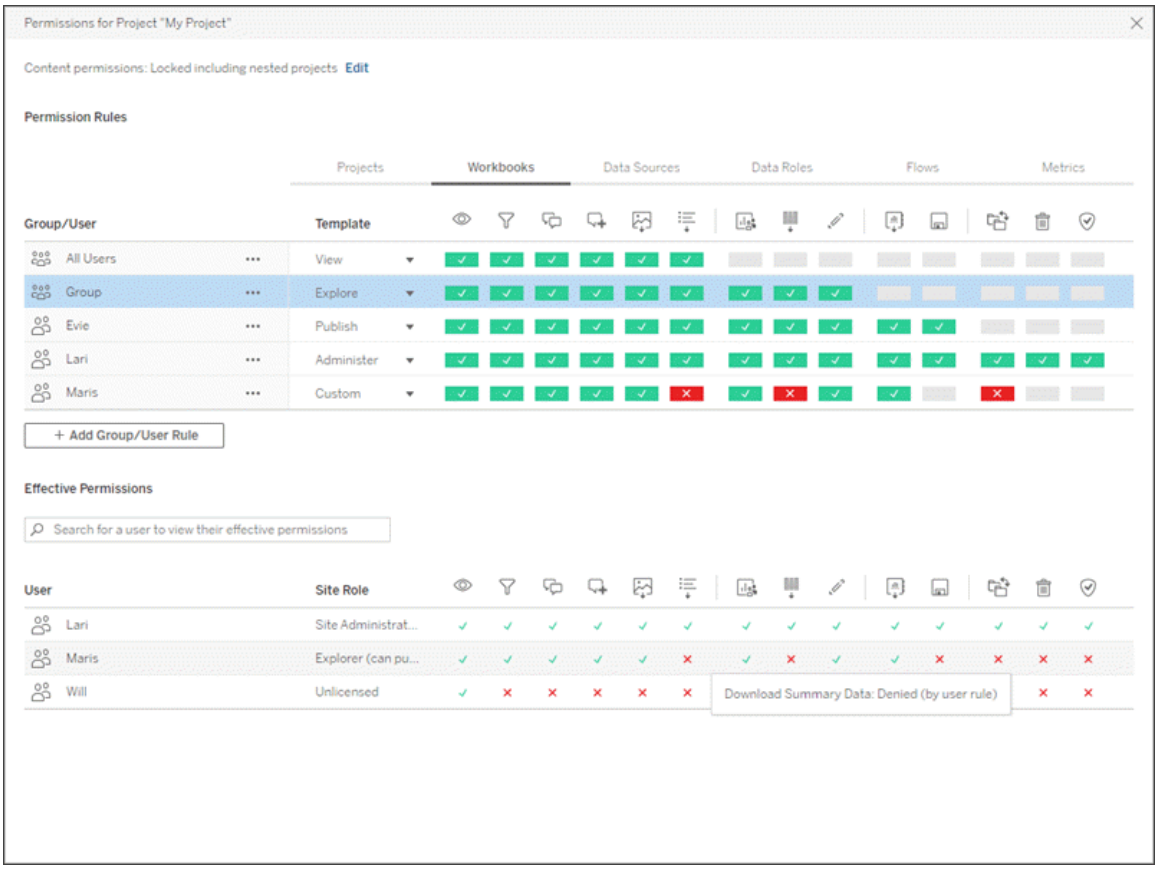
Set permissions on a view

In some situations, it may be valuable to specify permissions on a view independently from the workbook that contains it. To set permissions on a published view, navigate to the view within a published workbook and follow steps above.

Warning: While it is possible to set view-level permissions within a workbook, we strongly recommend managing permissions at the project (or workbook) level as much as possible. For views to inherit permissions, the project must be locked or the workbook must be published with **Show Sheets as Tabs**. See Let Site Users Request Access to Content for more information.

Permissions

Permissions determine how users can interact with content such as workbooks and data sources. Permissions are set in the permission dialog or via the REST API. At the top of the dialog, permission rules configure capabilities for groups or users. Below, the permissions grid displays the effective permissions for users.



There are several interrelated topics that discuss how to think about, set, and manage permissions. The main topics are:

- This topic, which covers the fundamentals, how to set permission rules for projects and other content, and permission considerations for specific scenarios.
- Permission Capabilities and Templates, which covers in detail the various capabilities that are used to build permission rules.
- Manage Permissions with Projects, which covers using projects to manage permissions and how nested and locked projects impact permissions.
- Effective permissions, which covers how permission rules are evaluated and how final permissions are determined.
- Permissions, Site Roles, and Licenses, which covers how permissions interact with site roles and licenses to determine what a user can do on a site.

Additionally, if Data Management is licensed, permissions for external assets have additional considerations. For more information, see Manage Permissions for External Assets.

Permissions fundamentals

Projects and groups

Tableau sites use *projects* to organize content and *groups* to organize users. Managing permissions is easier when permission rules are:

- Set at the project level instead of on individual pieces of content.
- Established for groups instead of individuals.

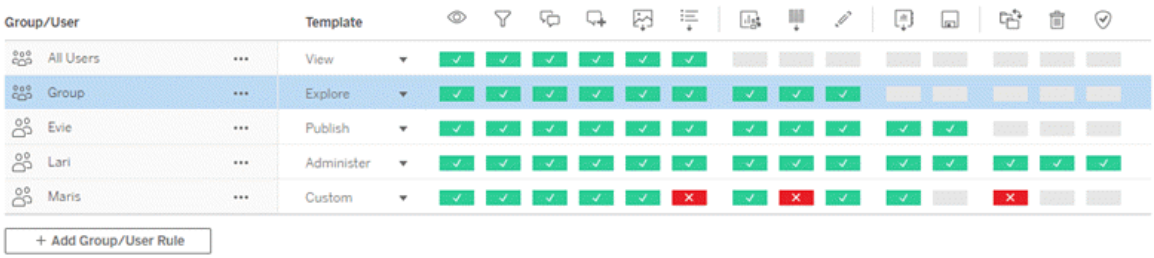
Permissions can only be established for users, groups, projects, or assets that already exist. For more information about creating users and groups, creating projects, and publishing content, see Manage Users and Groups, Use Projects to Manage Content Access , and Publish Data Sources and Workbooks.

Capabilities and permission rules

Permissions are made up of *capabilities*—the ability to perform actions like view content, web edit, download data sources, or delete content. *Permission rules* establish what capabilities are allowed or denied for a user or group on an asset.

For more information about capabilities and permission rule templates, see Permission Capabilities and Templates.

Note: When talking about permissions in general, it's common to see a phrase like "a user must have the delete *permission*." This is easy to understand in a broad context. However, when working with permissions at a technical level like in this article, it's more accurate to say "the delete *capability.*" In this topic we'll use the more precise term *capability*, but you should be aware that you might see *permission* in other places.



For a breakdown of the capability icons and their meanings, see Permission Capabilities and Templates.

The interplay between license level, site role, and potentially multiple permission rules factor into the final determination of what a user can or can't do. For each user this becomes their *effective permissions*. For more information, see Effective permissions.Some tasks such as creating new workbooks from a browser (web authoring) or moving content might require specific configurations of several capabilities rather than being captured in a single capability. For more information, see Permission settings for specific scenarios.

Set permissions

Permission rules are set differently at the project level, at the content level, or when publishing content from Tableau Desktop.

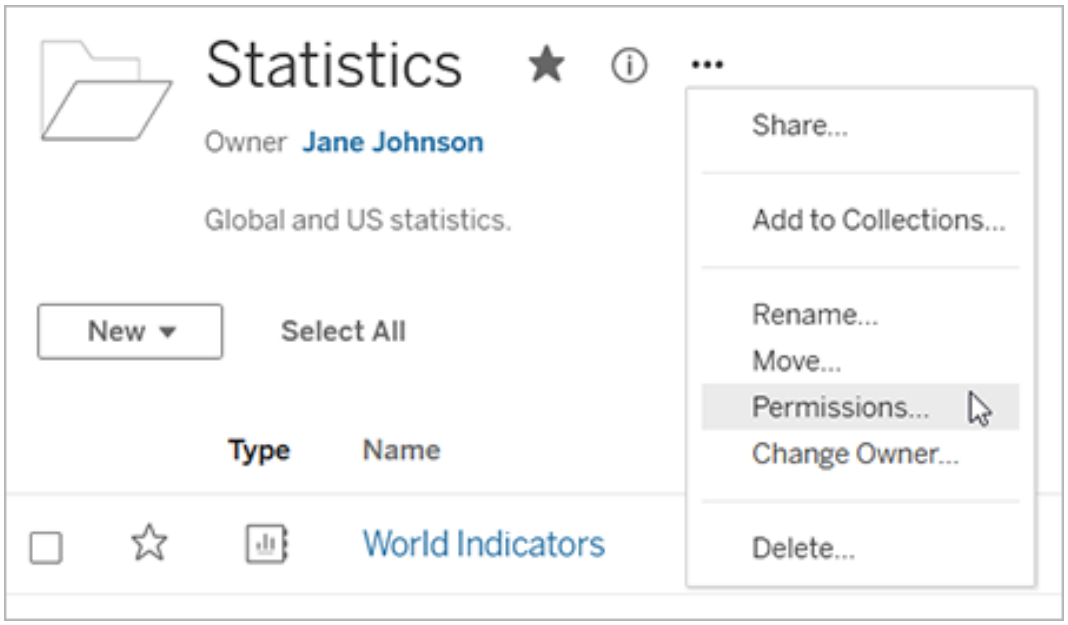
> **Note**: The phrase "project permissions" can have two meanings. There are the permission capabilities for a project itself—View and Publish—that control how a user can interact with a project. There is also the concept of project-level permission rules for other content types. In this article "project-level permissions" means permission rules for workbooks, data sources, and the other assets that are configured in the permission dialog for a project. This is in contrast to "content-level" permission rules that can be set on a specific workbook, data source, etc.

Project-level permissions

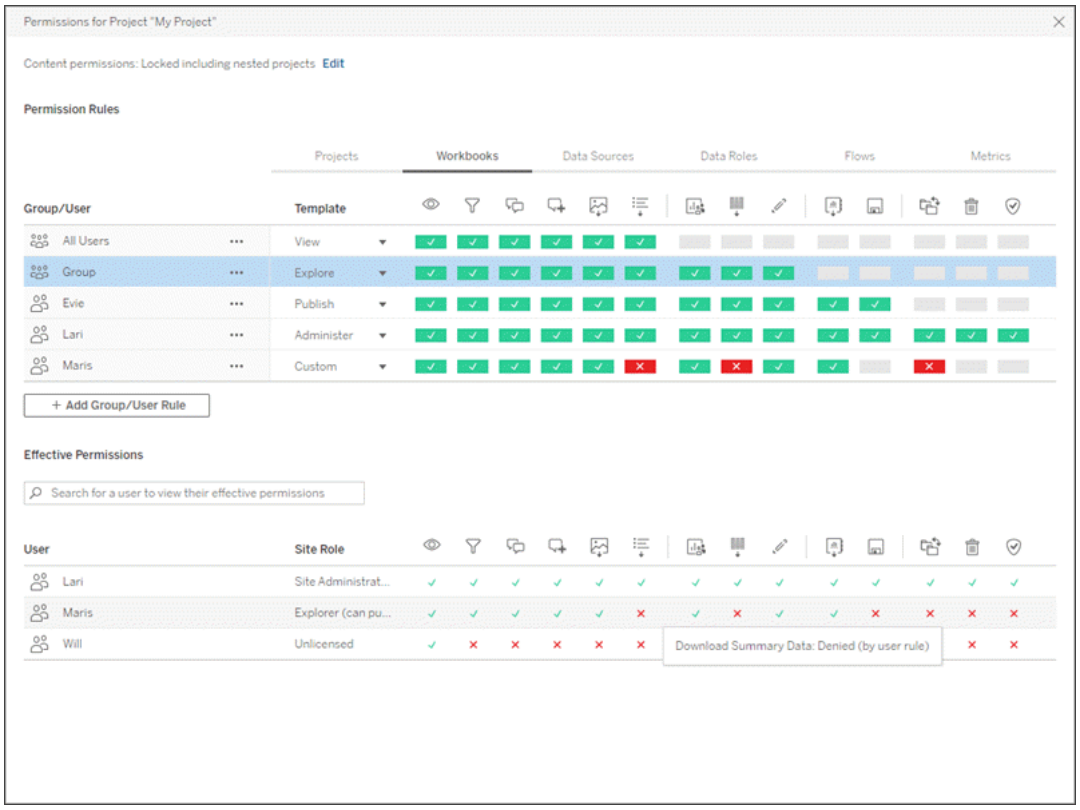
For administrators, project owners, and project leaders

To set permissions at the project level:

1. Navigate to the project
2. Open the Actions menu (...) and click **Permissions**.



The permissions dialog opens. This dialog has two main areas: permission rules at the top and the effective permissions grid below. Each content type has a tab. The image below shows the Workbook tab.



With a row selected at the top, the effective permissions grid populates. Use this to verify permissions. Hovering provides information about why the capability is allowed or denied for that specific user.

3. To modify an existing permission rule, select the appropriate tab for that content type and click a capability.
4. To create a rule, click **+ Add Group/User Rule** and start typing to search for a group or user. For each tab, choose an existing template from the dropdown box or create a custom rule by clicking the capabilities.

   One click sets the capability to **Allowed**, two clicks sets it to **Denied**, and a third click clears the selection (**Unspecified**).

5. When finished, click **Save**.
   - If the "None" template is selected, the button will say "Delete Rule".

Set project permissions for all content types

Remember that the permissions dialog for a project contains tabs for each type of content. **You must set permissions for each type of content at the project level or users will be denied access to that content type**. A capability is only granted to a user if they're expressly allowed it. Leaving a capability as Unspecified will result in it being denied.

Tip: Every time you create a permission rule at the project level, make sure you look through all the content type tabs.

Configure the asset permissions setting

Permission rules set at the project level act as a default for content saved in that project and any nested projects it contains. Whether those project-level default rules are kept uniform or are able to be edited depends on the **Asset permissions** setting. This setting can be configured in two ways, either **Locked** or **Customizable**. For more information, see Lock asset permissions.

Content-level permissions

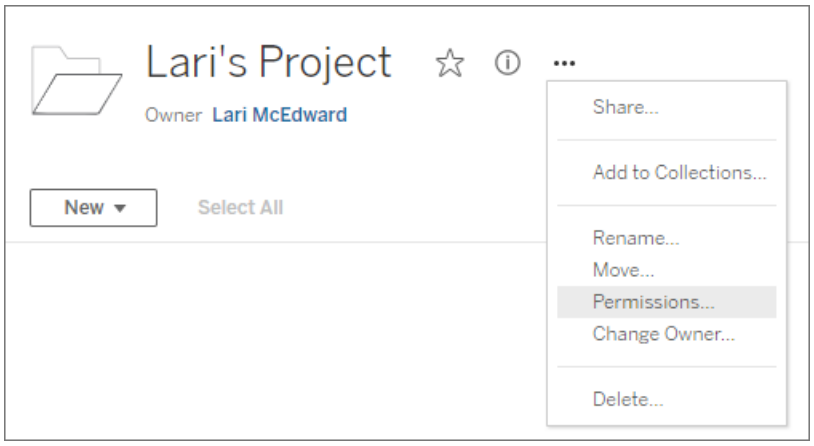
For administrators, project leaders, and content owners

If project **Asset permissions** are **Customizable**, permissions for individual assets can be modified. The information below isn't relevant to assets in locked projects. For more information, see Lock asset permissions.

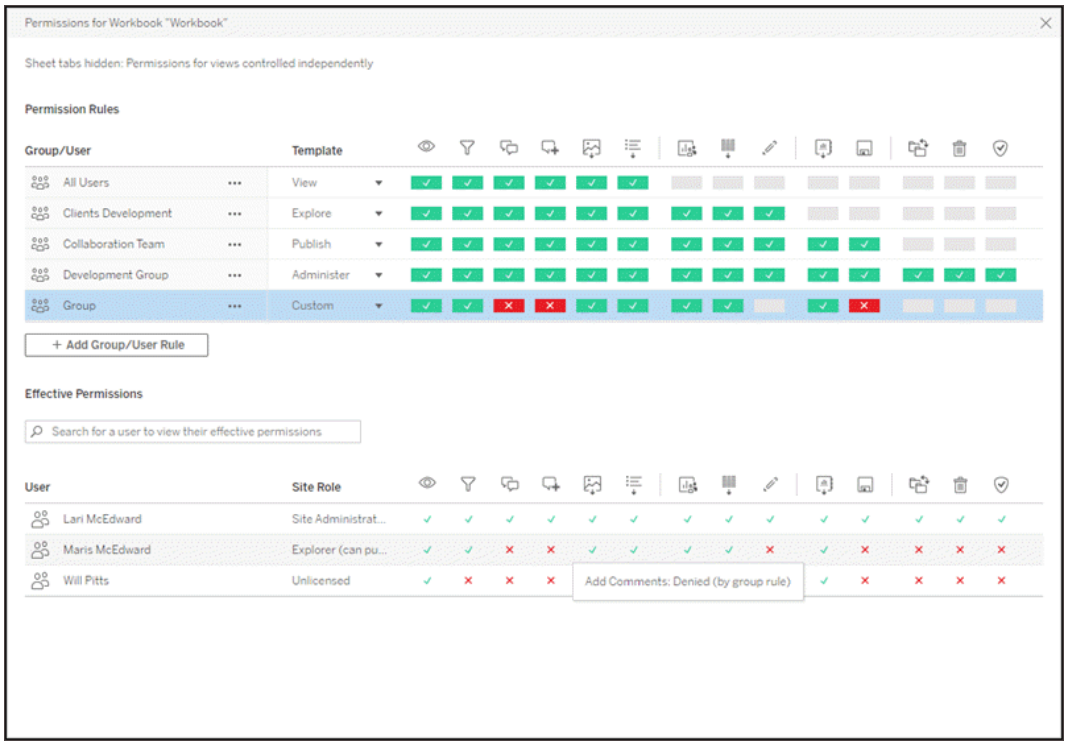
Tip: While it is possible to set permissions on individual assets in **Customizable** projects, we recommend managing permissions at the project level.

Set permissions on assets

1. Navigate to the asset (such as a workbook, data source, or flow)
2. Open the Actions menu (...) and click **Permissions**.



The permissions dialog opens. This dialog has two main areas: permission rules at the top and the effective permissions grid below. (Note the lack of tabs across the top—an asset-level permissions dialog has no tabs.)



With a row selected at the top, the effective permissions grid populates. Use this to verify permissions. Hovering over a capability square provides information about why the capability is allowed or denied for that specific user.

3. To modify an existing permission rule, click a capability.
4. To create a rule, click **+ Add Group/User Rule** and start typing to search for a group or user. Choose an existing template from the dropdown or create a custom rule by clicking the capabilities.

   One click sets the capability to **Allowed**, two clicks sets it to **Denied**, and a third click clears the selection (**Unspecified**).

5. When finished, click **Save**.
   - If the "None" template is selected, the button will say "Delete Rule".

Set permissions on a view

Tip: While it's possible to set view-level permissions within a workbook, we strongly recommend managing permissions at the project (or, if necessary, workbook) level.

If a workbook is published with **Show Sheets as Tabs** checked, the views in that workbook will inherit all permissions set for the workbook. The permission dialog for a view will be read-only.

In some situations, it may be valuable to specify permissions on a view independently from the workbook that contains it. If the workbook is published with **Show Sheets as Tabs** unchecked (sheet tabs hidden), the views will start with the workbook permissions but will be independent thereafter and can be set independently. Note that this means if the permission rules are modified for the workbook, those changes won't be applied to the views—each view's permissions will need to be managed individually.

See Show or Hide Sheet Tabs for more information.

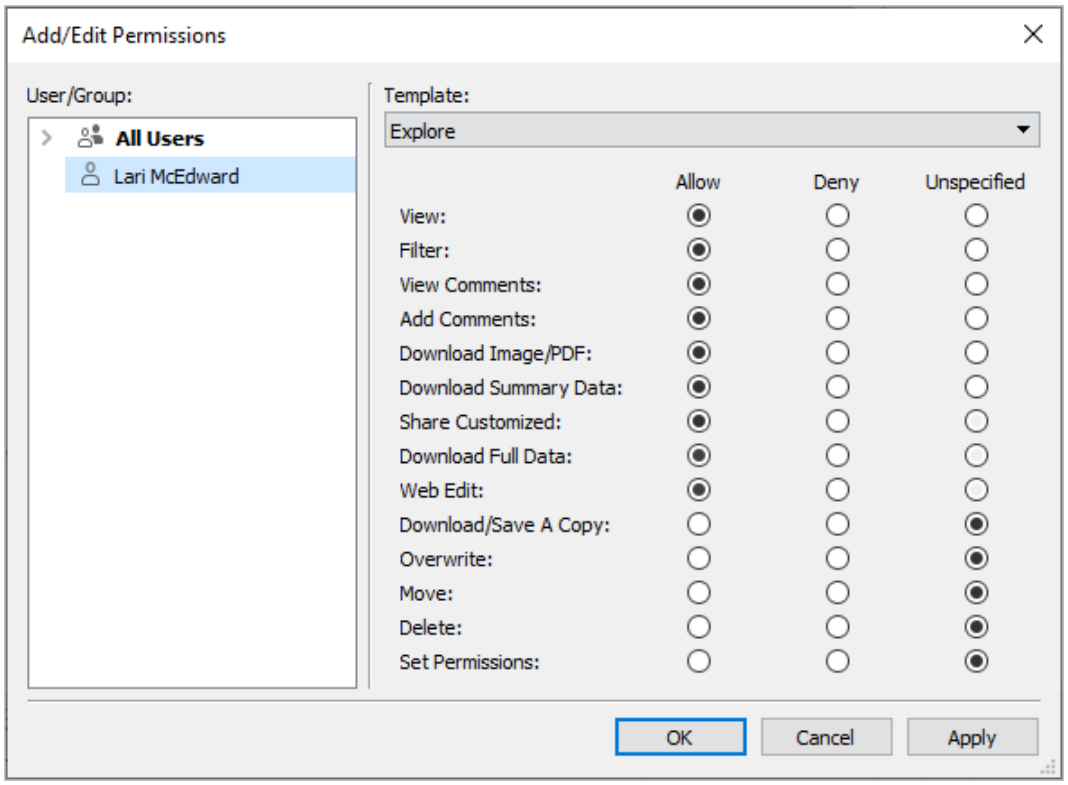
Set permissions at publish

For content publishers

If project **Asset permissions** are **Customizable**, permissions for individual assets can be set when publishing from Tableau Desktop. The information below isn't relevant for content in locked projects. For more information, see Lock asset permissions.

Tip: While it's possible to set permissions on individual assets in **Customizable** projects, we recommend managing permissions at the project level.

1. From the publishing dialog , click the Edit link for **Permissions**.
   If the Edit link is unavailable, permissions are locked to the project and can't be modified except by the project owner, project leader, or an administrator.
2. The Add/Edit Permissions dialog shows any existing permission rules. Click **Add** to add a permission rule or **Edit** to modify an existing permission rule
   a. Select the group or user from the left pane. You can expand a group to see which users it contains.
   b. Use the selector at the top of the right pane to choose an existing template, or use the radio buttons to create a custom rule.



Note that effective permissions can't be inspected from the publishing dialog.

3. When finished, click **OK** and resume publishing.

Note: Permissions can't be set while publishing flows from Tableau Prep Builder. To set permissions on a flow, refer to the steps for Project-level permissions or Content-level permissions.

Clean up the All Users group

By default, all users are added to an "All Users" group that has basic permissions for content. To start with a clean slate when building your own permission rules, we recommend that you delete the rule entirely or edit the rule for All Users to remove any permissions (set the permission role template to None). This helps prevent any ambiguity down the road by reducing the number of rules that applies to any given user and therefore making effective permissions easier to understand.

Permission settings for specific scenarios

Certain actions require combinations of permission capabilities and possibly site roles. The following are some common scenarios and their necessary permission configurations

Saving, publishing, and overwriting

In the context of permissions, saving is essentially publishing. As such, the **Overwrite** and **Save a Copy** capabilities can only be given to users with a site role that allows publishing: Administrator, Creator, or Explorer (can publish). Explorer or Viewer site roles can't publish, overwrite, or save a copy.

- The **Publish** capability for a project allows a user to publish content into that project.
- The **Overwrite** capability allows a user to save over an existing piece of content. By saving over the content, the user becomes the owner of that content. The Overwrite capability also allows users to edit minor aspects of existing pieces of content, such as the description for a metric or the synonyms for a data role. Editing the existing content in this way doesn't change the owner of the content.
- The **Save a Copy** capability allows a user to save a new copy of the content. This is usually done in conjunction with web authoring and means the user can save their modifications.

It's important to note that users aren't able to Save or Save As a piece of content unless they've the **Publish** capability for at least one project, because all content must be published into a project. Without the **Publish** capability at the project level, the content can't be published.

In web editing, the **Save** option in the File menu only appears to the content owner. If a user who isn't the owner has the **Overwrite** capability (allowing them to save the content), they must use **File > Save As** and name the workbook the exact same name. This prompts a warning that they're about to overwrite the existing content, which they can do. Conversely, a user with only the **Save a Copy** capability trying to use the same name gets an error stating they don't have permission to overwrite the existing content.

If a user who isn't the content owner overwrites content, they become the owner, with all the permissions that entails. The original owner's access to the content is then determined by their permissions as a user rather than the owner.

> **Note**: **Download Workbook/Save a Copy** is a joint capability for workbooks. Explorers can be given this capability but they're only able to download the workbook, not save a copy. Giving the capability to Explorer (can publish), Creator, or Administrator site roles gives them both the ability to download workbooks *and* save a copy.

Web Editing and Web Authoring

Web editing and web authoring allows users to edit or create workbooks directly in the browser.

The permission capability is called *Web Edit* and the site setting is called *Web Authoring*. This section refers to any web-based editing or publishing action as *web authoring*.

To enable this functionality, there are several requirements.

- **Site setting**: Web authoring must be turned on for the entire Tableau site. See Set a Site's Web Authoring Access. Without this setting enabled, no users can create workbooks or edit existing workbooks from the browser, *even if they have the web edit capability*.
- **User site role**: The user must have the appropriate site role.
  - Viewers can never web edit.
  - Explorers can be given the web edit capability but can't publish. Essentially, they can use web editing to answer deeper questions based on existing content on the fly, but can't save their edits.
  - Explorers (can publish) or Site Administrator Explorers can publish, but they can only use data that is already published to the site.
  - Creators, Site Administrator Creators, and Server Administrators can publish and create data sources.
- **Permission capabilities**: The user must have the necessary permission capabilities based on the desired functionality.

Required Permission Capability Settings

Desired functionality	Minimum Site Role	Web Edit	Download/ Save a Copy	Overwrite *(workbook)*	Publish *(project)*	Connect (*data source*)
Web author without being able to save	*Explorer*	Allow	Deny	Deny	Optional	Allow
Web author and save as new content	*Explorer (can publish)*	Allow	Allow	Deny	Allow	Allow
Web author and save (overwrite) content	*Explorer (can publish)*	Allow	Allow	Allow	Allow	Allow
Web author with new data and save new content	*Creator*	Allow	Optional	Optional	Allow	Optional

Optional indicates this capability isn't involved in the desired functionality

Data access for published Tableau data sources

Data sources published to a Tableau site can have native authentication as well as permissions within the Tableau environment.

When the data source is published to the Tableau site, the publisher can choose how to Set Credentials for Accessing Your Published Data, which addresses how data source credentials are handled (such as requiring users to log into a database or enter their credentials for

Google Sheets). This authentication is controlled by whatever technology holds the data. This can be embedded when the data source is published, or the data source publisher can choose to prompt the user for their credentials to the data source. For more information, see Publish a Data Source.

There are also data source capabilities that allow or deny users the ability to see (**View**) and connect to the published data source (**Connect**) in the context of Tableau. These capabilities are set like any other permissions in Tableau.

When a workbook is published that uses a published data source, the author can control how the Tableau authentication behaves for someone consuming the workbook. The author sets the workbook's access to the published data source, either as **Embed password** (using the author's Connect access to the data source) or **Prompt users** (using the Connect access of the person viewing the workbook), which may require data source authentication as well.

- When the workbook is set to **Embed password**, anyone who looks at the workbook sees the data based on the author's access to the data source.
- If the workbook is set to **Prompt users**, the Tableau-controlled access is checked for the data source. The person consuming the workbook must have the Connect capability for the published data source to see the data. If the published data source is also set to Prompt user, the viewer must also enter their credentials for the data source itself.

Workbook authentication to the data source	Data source authentication to the data	How data access is evaluated for someone consuming the workbook
Embed password	Embed password	User sees the data as if they were the workbook author
Embed password	Prompt user	User sees the data as if they were the workbook author. (The author is prompted for data source authentication, not the user.)
Prompt user	Embed password	User must have their own **Connect** capability to the published data source

Prompt user	Prompt user	User must have their own **Connect** capability to the published data source and are prompted for their credentials to the underlying data

Note that this applies to consuming a workbook, not web editing. To web edit, the user must have their own Connect capability.

For information on embedding passwords when you publish Tableau content such as a data source or workbook that uses a virtual connection, see Virtual connections in the Tableau Server help.

Move content

To move an item, open its Action menu (...) and click **Move**. Select the new project for the item, then click **Move Assets**. If **Move** is unavailable or there are no available destination projects, verify the appropriate conditions are met:

- Administrators can always move assets and projects to any location.
- Project leaders and project owners can move assets and nested projects among their projects.
    - Note that non-administrators can't move projects to become top-level projects
- Other users can move assets only if all three of the following requirements are met:
    - Creator or Explorer (Can Publish) site role.
    - Publishing rights (**View** and **Publish** capabilities) for the destination project
    - Owner of the content, or—for workbooks and flows—having the **Move** capability.

When moving a database with its tables, the user must have the **Move** capability for both the database *and* its tables.

For information about how permissions are handled when moving content and projects, see Move projects and content.

Metrics

Retirement of the legacy metrics feature

Tableau's legacy metrics feature was retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2. In October 2023, Tableau retired the ability to embed legacy metrics in Tableau Cloud and in Tableau Server version 2023.3. With Tableau Pulse, we've developed an improved experience to track metrics and ask questions of your data. For more information, see Create Metrics with Tableau Pulse to learn about the new experience and Create and Troubleshoot Metrics (Retired) for the retired feature.

Metrics are created from views in published workbooks. Users can create metrics if they:

- Are a Creator or Explorer (can publish) site role
- Have the **Publish** capability on a project
- Have the **Create/Refresh Metric** capability for the relevant workbook

For more information, see Create and Troubleshoot Metrics (Retired) and Set Up for Metrics.

> **Note:** Prior to 2021.3, the ability to create a metric on a view was controlled by the Download Full Data capability.

Because metrics are independent assets, it's important to note that the permissions for metrics are managed independently from the view they were created from. (This is unlike data-driven alerts and subscriptions, where the content of the alert or subscription can only be seen if the user has the correct permissions for the view itself.)

Although the capabilities for metrics are straightforward, the **View** capability should be considered carefully. It may be possible for a workbook with restricted permissions to be the basis for a metric with more open permissions. To protect sensitive data, you might want to deny metric creation for specific workbooks.

Metrics display data from their owner's perspective

When you create a metric, you capture your perspective of the data from that view. This means that any users who can access your metric will see the data as it appears to you. If the data in the view is filtered based on your credentials, the data you see might be different from

what other users see when they access the same view. Limit the **View** capability for your metric if you're concerned about exposing your perspective of the data.

Explain Data

When Explain Data is available, a user can select a mark in a view and click Run Explain Data in the mark's Tooltip menu. A combination of settings must be enabled to make Explain Data available in editing mode and viewing mode.

Requirements for authors to run Explain Data or edit Explain Data settings in editing mode:

- Site setting: **Availability of Explain Data** set to **Enable**. Enabled by default.
- Site role: Creator, Explorer (can publish)
- Permissions: **Run Explain Data** capability set to **Allowed**. Unspecified by default. If you open a workbook (Tableau version 2022.1 or earlier) that used this permission in Tableau version 2022.2 or later, you must reset the Run Explain Data capability to Allowed.

Note: The **Download Full Data** capability for a Creator or Explorer (can publish) controls whether they see the View Full Data option in Extreme Values explanations. Viewers are always denied the Download Full Data capability. However, all users can see record-level details when the Extreme Values explanation type is enabled in Explain Data settings.

Requirements for all users to run Explain Data in viewing mode:

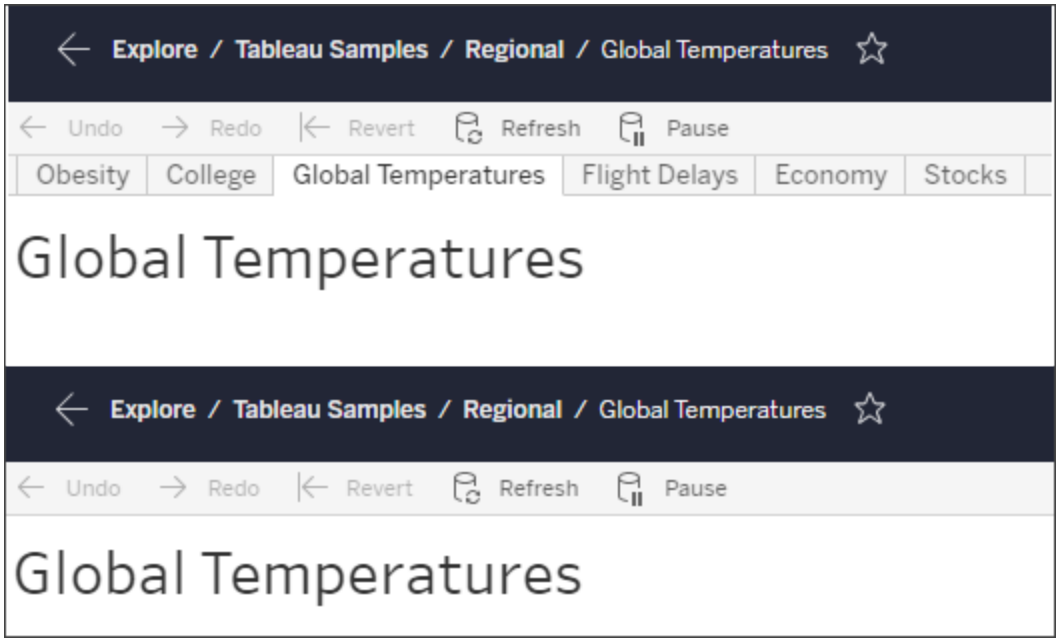
- Site setting: **Availability of Explain Data** set to **Enable**. Enabled by default.
- Site role: Creator, Explorer, or Viewer
- Permissions: **Run Explain Data** capability set to **Allowed**. Unspecified by default. If you open a workbook (Tableau version 2022.1 or earlier) that used this permission in Tableau version 2022.2 or later, you'll need to reset the Run Explain Data capability to Allowed.

Show or Hide Sheet Tabs

In the context of published content, sheet tabs (also referred to as tabbed views) is a distinct concept from sheet tabs in Tableau Desktop. Showing and hiding sheet tabs in Tableau

Desktop refers to hiding sheets in the authoring environment. For more information, see Manage Sheets in Dashboards and Stories.

Showing and hiding sheet tabs (turning tabbed views on or off) for published content refers to navigation in a published workbook. When sheet tabs are shown, published content has navigational sheet tabs along the top of each view.



This setting also impacts how permissions function and may have security implications (see note).

Note: It's possible to have the **View** capability for a view without the **View** capability for the workbook or project that contain it. Normally if a user lacks the View capability for a project and workbook, they wouldn't know those assets exist. If they have the View capability for a view, however, a user may be able to see the project and workbook name when looking at the view, such as in the navigational breadcrumb. This is expected and accepted behavior.

Turn off tabbed views to allow independent view permissions

Although it isn't recommended as a general practice, there are times when it can be useful to set permissions on views independently of the workbook that contains them. To do so, three conditions must be met:

1. The workbook must be published—there's no way to set view permissions during publishing.
2. The workbook must be in a customizable project.
3. The workbook can't show sheets as tabs (tabbed views must be hidden).

When a workbook shows sheets as tabs, all views inherit the workbook permissions and any changes to the workbook permissions affect all of its views. **When a workbook in a customizable project doesn't show tabbed views, all views assume the workbook permissions upon publication, but any subsequent changes to the workbook's permission rules *won't be inherited by the views*.**

Changing the configuration of sheets as tabs on a published workbook will also impact the permission model. Show Tabs overrides any existing view-level permissions and reinstate the workbook-level permissions for all views. Hide Tabs breaks the relationship between the workbook and its views.

- To configure sheets as tabs on a published workbook, open the Actions menu (...) for the workbook and select **Tabbed Views**. Choose **Show Tabs** or **Hide Tabs** as desired.
- To configure sheets as tabs during publishing, refer to Show sheets as tabs.
- To set view-level permissions, see Set permissions on assets.

Important: In a customizable project, any modifications to the workbook-level permissions won't be applied if navigational sheet tabs are hidden (aka tabbed views are off). Changes to permissions must be made on individual views.

Collections

For information on managing permissions in Collections, refer to Collections.

Permission Capabilities and Templates

Permissions are made up of capabilities, or the ability to perform a given action on a piece of content, such as view, filter, download, or delete. Each row in the Permission Rules area of the dialog is a *permission rule*. Permission rules are the setting for each capability (allowed, denied, or unspecified) for the group or user in that row. Permission rules have *templates* available that make it easier to assign capabilities quickly. Permission rules can also be copied and pasted.

Note: In the permission dialog for projects, there are tabs for each content type: **Projects**, **Workbooks**, **Data Sources**, **Data Roles**, **Flows**, **Ask Data Lenses**, **Metrics** and—if you have the Data Management—**Virtual Connections**, **Databases**, and **Tables**. (Virtual connections were added in Tableau Server 2021.4 and Tableau Cloud December 2021. Databases and tables were added in Tableau Server 2022.3 and Tableau Cloud October 2022.) When a permission rule is added, the default for all capabilities across all content types is Unspecified. To allow or deny capabilities for each content type, you must go to each tab in turn. In the permission dialog for a specific piece of content, there are no tabs and the permission rules only apply to that piece of content.

Templates

Templates group sets of capabilities that are often assigned together based on common user scenarios, **View**, **Explore**, **Publish**, and **Administer**. When you assign a template, its included capabilities are set to **Allowed**, with the rest left as **Unspecified**. The templates are cumulative, so the Explore template includes everything from the View template plus additional capabilities. All content also has a template for **None** (which sets all capabilities to unspecified) and **Denied** (which sets all capabilities to denied).

Templates are meant to be a starting point and can be adjusted after they are applied. Capabilities can also be granted or denied without using a template at all. In both cases, the template column then shows **Custom**.

Copy and paste permissions

If there is a permission rule that needs to be assigned to multiple groups or users, you can copy and paste from one rule to another. You can't copy from or paste onto a rule that involves Project Leader status.

1. Open the action menu (...) for the existing rule you want to copy from and select **Copy Permissions**. This is available only when the rule is not in edit mode.
2. Select an existing rule you want to paste over. You can also create a new rule by clicking **+ Add Group/User Rule** and selecting a group or user.
3. Open the action menu (...) and select **Paste Permissions**.

Capabilities

Each content type has specific capabilities:

Projects

Projects have only two capabilities and two templates. For more information about project leaders and how to assign them, see Project administration.

View template

View lets a user see the project. If a user hasn't been granted the view capability, the project won't be visible to them. Granting the view capability for a project does not mean a user can see any content in the project, just the existence of the project itself.

Publish template

Publish lets a user publish content to the project from Tableau Desktop or Tableau Prep Builder. The publish capability is also required to move content into the project or save content to the project from web authoring.

Workbooks

View template

View lets a user see the workbook or view. If a user hasn't been granted the view capability, the workbook won't be visible to them.

Filter lets a user interact with filters in the view, including keep only and exclude filters. Users lacking this capability won't see filter controls in the view.

View Comments lets a user view the comments associated with the views in a workbook.

Add Comments lets a user add comments to views in a workbook.

Download Image/PDF lets a user download each view as a PNG, PDF, or PowerPoint.

Download Summary Data lets a user view the aggregated data in a view, or in the marks they've selected, and download that data (as a CSV).

Explore template

Share Customized lets users add their custom views to the list of "Other Views" visible on a workbook.

- When this capability is denied, users won't see the "Make visible to others" option when they create a custom view. For more information, see Use Custom Views. This capability doesn't impact the ability to share a custom view with the share dialog or by copying the link.

Download Full Data lets a user view the underlying data in a view, or in the marks they've selected, and download that data (as a CSV).

Web Edit lets a user edit the view in a browser-based authoring environment.

- Note that creating new content in the browser or saving views from the web edit interface requires a specific combination of capabilities. For more information, see Web Editing and Web Authoring.
- The Web Editing feature must also be enabled for the entire site or even users with this capability allowed won't be able to web edit. For more information, see Set a Site's Web Authoring Access.

 **Run Explain Data** lets a user run Explain Data on marks in editing and viewing mode.

- Note that for Explain Data to be displayed as an option when a user selects a mark in a workbook, the feature must also be enabled as a site setting. To make Explain Data available in viewing mode, the feature must also be allowed by the author from within a workbook in Explain Data settings. For more information, see Control Access to Explain Data.

Publish template

 **Download Workbook/Save a Copy** lets a user download a packaged workbook (as a TWBX). Lets a user save (publish) a copy from the web edit interface as a new workbook.

 **Overwrite** lets a user overwrite (save) the content or asset on the server.

- When allowed, the user can re-publish a workbook, data source, or flow, or save a workbook or flow in web authoring, thereby becoming the owner and gaining access to all permissions. After this change in ownership, the original owner's access to the workbook is determined by their permissions just like any other user.

 **Create/Refresh Metrics** lets a user create metrics on the views in a workbook and lets any metrics that a user creates from those views refresh. The legacy Metrics feature was retired in February 2024 for Tableau Cloud and in Tableau Server version 2024.2. For more information, see Create and Troubleshoot Metrics (Retired).

Administer template

Move lets a user move workbooks between projects. For more information, see Move content.

Delete lets a user delete the workbook.

Set Permissions lets a user create permission rules for the workbook.

Views

In a workbook that is not in a locked project and does not show sheets as tabs for navigation, views (sheets, dashboards, stories) inherit the workbook permissions at publication, but any changes to permission rules must be made on individual views. View capabilities are the same as those for workbooks, except for **Overwrite**, **Download Workbook/Save a Copy**, and **Move** which are only available at the workbook level.

We recommend showing navigational sheet tabs whenever possible so views continue to inherit their permissions from the workbook.

Data Sources

View template

View lets a user see the data source on the server.

Connect lets a user connect to a data source in Tableau Desktop, Tableau Prep Builder, Ask Data, or web editing.

- If a workbook author embeds their credentials to a published data source in a published workbook, they are essentially embedding their **Connect** capability. Therefore, users can see the data in the workbook regardless of their own **Connect** capability for that data source. If the workbook author doesn't embed their credentials to the published data source, the user needs their own **Connect** capability to the data source to consume the workbook. For more information, see Data access for published Tableau data sources.

- A user must have the **Connect** capability for a data source to use Ask Data and to create Ask Data lenses. For more information, see Enable Ask Data for Sites and Data Sources.

Explore template

Download Data Source lets a user download the data source from the server (as a TDSX).

- Cube data sources, like those for Microsoft Analysis Services or Oracle Essbase connections, must be used locally. To download the published data source to Tableau Desktop, the user must have the Download capability. For more information, see Cube Data Sources.

Publish template

Overwrite lets a user publish a data source to the server and overwrite the data source on the server.

API Access lets a user query the data source with the VizQL Data Service. For more information, see VizQL Data Service.

Administer template

Delete lets a user delete the data source.

Set Permissions lets a user create and edit permission rules for the data source.

Other types of assets

	View template	Explore template	Publish template	Administer template
Flows	**View** lets a user view the flow.	**Download flow** lets a user	**Run** lets a user run the	**Move** lets a user move

		download the flow (as a TFLX).	flow.  **Overwrite** lets a user publish a flow and overwrite the published flow.	assets between projects. For more information, see Move content.  **Delete** lets a user delete the asset.
Data Roles	 **View** lets a user view data roles.	n/a	 **Overwrite** lets a user publish data roles, overwrite published data roles, and edit published data roles' synonyms.	 **Set Permissions** lets a user create permission rules for the asset.
Metrics (retired)	 **View** lets a user view metrics.	n/a	 **Overwrite** lets a user overwrite a metric and edit a metric's details.	
Ask Data Lenses	 **View** lets a user see the lens.	n/a	 **Overwrite** lets a user edit the lens.	
Virtual Connections	 **View** lets a user see the virtual connection.  **Connect*** lets a user con-	n/a	 **Overwrite** lets a user edit the virtual connection.	

	nect to data using a virtual connection.			
Databases	**View** lets a user see the database.	n/a	**Overwrite** lets a user edit the metadata for the data-base.	
Tables	**View** lets a user see the table.	n/a	**Overwrite** lets a user edit the metadata for the table.	
Collections	**View** lets a user view col-lections.	n/a	n/a	n/a

*By default, virtual connections have a Custom template that sets the View capability to Allowed but not the Connect capability. Be sure to set the Connect capability to Allowed so users can connect using the virtual connection.

Manage Permissions with Projects

Projects can simplify permission management with features such as nested projects, project visibility, non-admin project leaders, and locking permissions.

Tip: How permissions are set at the project level is important, especially for the Default project. When a new top-level project is created, it inherits its default permission rules (for all content types) from the Default project. When a new project is created nested inside another project, the child project inherits its default permission rules from the parent project.

Project administration

Projects are containers used to organize and manage access to content. By giving non-administrators privileges to manage projects, certain content administration tasks can be handled at the project level.

Project Leaders: Projects can have project leaders, users who have been set as a **project leader**. This setting automatically grants a user their maximum capabilities—depending on their site role—for that project and all content in that project. Project leaders with site role of Explorer (can publish) and above have all capabilities. Project leaders are essentially local admins for the project without access to site or server settings.

Hierarchy: Only administrators can create top-level projects. Project owners and project leaders can create nested projects inside their projects.

Project owners and leaders have full administrative access to the project and its content, as well as any nested projects it contains. In a hierarchy, project leaders are implicitly given project leader access to all child content. To remove project leader access, you must do so at the level in the hierarchy where the role was explicitly assigned.

Ownership: A project can have multiple project leaders, but each project has exactly one owner. By default, a project is owned by the user who created it.

A project’s owner can be changed by the existing owner or an administrator. (Project leaders can't change project ownership, only content ownership). Projects can be owned by users with a site role of Explorer (can publish), Creator, or administrator. Project ownership can be changed even if a project is locked.

Deleting: Most content can only exist inside a project. Only administrators can create and delete top-level projects, but project leaders can create or delete nested projects.

Deleting projects also deletes all the Tableau content and nested projects they contain. To delete a project without losing its content, move the content to another project first. Deleting projects can’t be undone.

External assets are handled differently. They don't have to be in a project. External assets aren't deleted if their project is deleted and continue to appear in **External Assets**. See External assets that aren't in projects for more information.

For a deeper dive into project administration, see Use Projects to Manage Content Access and Add Projects and Move Content Into Them.

Special projects

Default: The project named "Default" is a special project. When other top-level projects are created, they use the Default project as a template, and copy all their permissions rules from it (but not the **Asset permissions** setting). The **Default** project can't be deleted, moved, or renamed, but its description can be changed. It has no owner by default, but one can be assigned.

External Assets Default Project: In Tableau Cloud and Tableau Server 2023.1 and later, if you have a Data Management license with Catalog enabled, the project named "External Assets Default Project" appears when Catalog needs to move new or existing external assets to it. Catalog puts new external assets and external assets from deleted projects in the **External Assets Default Project**. The project has no permissions rules by default, so server administrators and site administrators are the only users who can see it unless permissions are added. It can't be deleted, moved, or renamed, but its description can be changed. It has no owner by default, but one can be assigned.

Set a project leader

Project leaders are users who have administrator-like access for a specific project or project hierarchy.

To assign project leader status to a group or user

1. Open the permission dialog for the appropriate project.
2. Select an existing permission rule, or click **+ Add Group/User Rule** and chose the desired group or user.
3. Open the action menu (...) for that permission rule and select **Set Project Leader...**.

Note: If the action menu includes an option for **Enable "Set Project Leader"**, this needs to be selected before the group or user can be set as a project leader. This option only appears when that group or user was denied the Project Leader capability (prior to 2020.1). That denied capability needs to be removed before they can be set as a project leader.

After a permission rule establishes a project leader, the templates and capabilities can't be edited because all capabilities are allowed for project leaders. If a project leader is established on a project that contains nested projects, they have inherited project leader status on all nested projects and their content.

Project leader status is always applied downward through the entire project hierarchy and can only be removed from the level where it was set. To remove project leader status, follow the same steps but select **Remove as Project Leader** from the action menu. After a group or user has been removed as project leader, that permission rule has all capabilities set to Unspecified. This may mean their access to and capabilities for that project is removed if there’s no other permission rule giving them permissions to the content. To keep their access to the project and its content, they need to have capabilities set like any other group or user.

Note: Project leaders can refresh extracts in their projects in most circumstances. They can't refresh extracts if they're only the project leader of a nested project (instead of a top-level project) and the top-level project is *locked (including nested projects)*.

Lock asset permissions

Permission rules set at the project level act as a default for content saved in that project and any nested projects it contains. Whether those project-level default rules are enforced or only preliminary depends on the **Asset permissions** setting. This setting can be configured in two ways, either **Locked** (recommended) or **Customizable**. Locking a project removes the ability for content owners to modify the permission rules on their content. Locking permissions can be applied to nested projects or just to the parent project itself.

- When **Asset permissions** is **Locked** (including nested projects), permission rules set at the project level are enforced for all assets in the project and all nested projects.
- When **Asset permissions** is **Locked** (*not* including nested projects), permission rules set at the project level are enforced for assets in the project. Nested projects can be configured independently with their own permission rules and set as locked or customizable.
- When **Asset permissions** is **Customizable**, permission rules set at the project level are applied to all assets in the project by default. However, permission rules can be modified for individual assets during or after publishing.

Note: Whether permission rules are locked or customizable, the permissions on content are always applied. *Locked* and *customizable* refer only to how project-level permissions are inherited by content in the project and who can change them. Even in a project with customizable permissions, only specific users can modify permissions (content or project owner, project leader, admins, or those with the Set Permission capability).

In a locked project:

- The project permission rules per content type are applied to all assets.
- Only administrators, project owners, and project leaders can modify permissions.
- Content owners lose the Set Permission capability but retain all other capabilities on their content.
- Permissions are predictable for all content in the project.

In a customizable project:

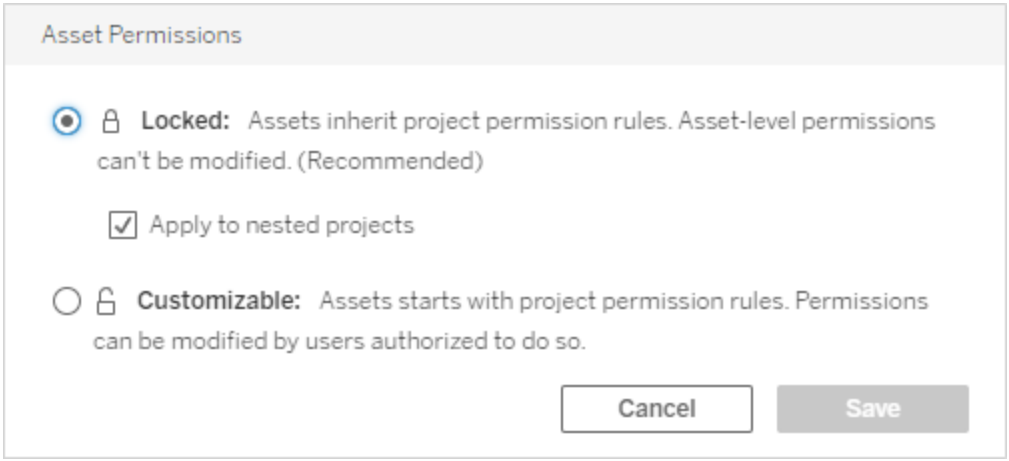
- The project permission rules are applied by default when content is published into the project or nested projects are created, but permissions can be modified during publication or after the content is created.
- Any user with the Set Permissions capability can modify permission rules for that content.
- Content owners have all capabilities on their content.
- Permissions can be different across content in the project.

Set asset permissions (lock a project)

New top-level projects inherit all initial permission rules from the Default project but not the **Asset permissions** setting, which is set to **Customizable**. This can be changed to **Locked** if desired.

To configure **Asset permissions**:

1. You must be logged into the site as an administrator, project owner, or project leader
2. Open the permissions dialog for a project
3. Next to **Asset permissions** in the upper left, click the **Edit** link and select the desired option in the **Asset permissions** dialog



Note: If the upper left corner doesn't show an **Edit** link in step 3 above, you may be on the permissions dialog for a (a) nested project or a piece of content in a locked project, in which case the link should bring you to the managing project, (b) piece of content in a customizable project, which won't show anything, or (c) view, which will indicate how the view permissions are tied to the workbook. For more information on the interplay of permissions for views and workbooks, see Show or Hide Sheet Tabs.

Change asset permissions

When the **Asset permissions** setting for a project is changed, the outcome depends on the new setting. Changes to permission rules in a locked hierarchy must be done at the level of

the managing project.

Changing from	Changing to	Outcome
Locked (including nested projects)	Locked	Doesn't modify existing permission rules. Any nested projects become customizable.
	Customizable	Doesn't modify existing permission rules, though they become customizable. Any nested projects become customizable.
Locked	Locked (including nested projects)	Overwrites existing custom permission rules for all nested projects and their content. This can't be undone.
	Customizable	Doesn't modify existing permission rules, though they become customizable. Any nested projects retain their content permission settings and permission rules.
Customizable	Locked (including nested projects)	Overwrites existing custom permission rules for content in the project, and all nested projects and their content. This can't be undone.
	Locked	Overwrites existing custom permission rules for content in the project. This can't be undone. Any nested projects retain their permission rules and remain customizable.

Move Tableau content and external assets

When *Tableau content* or *external assets* are moved between projects with different permission settings, **Asset permissions** settings determine the logic of how permissions are applied.

- Moving assets into a locked project overrides the existing permission rules and enforce the destination's permissions.
- **Moving assets into a customizable project maintains the existing permission rules on the asset.**

Note: Prior to Tableau Server 2022.3 and Tableau Cloud June 2022, external assets couldn't be in projects, and permissions on tables were managed through the **Table permissions** setting of the parent database. Beginning with Tableau Server 2022.3 and Tableau Cloud June 2022, external assets can be in projects. If a database or a table is moved into a project, older settings to control table permissions through the database are ignored, and the database or table permissions follow the logic of other assets.

Move projects

When a *project* is moved into another project, the permissions settings on the item being moved are maintained unless the destination project is scoped to include nested projects. (Project permissions in this case mean the View and Publish capabilities for the project itself.)

- If the destination project is set to **locked (including nested projects)**, the permissions for the project being moved *and its content* are overwritten.
- If the destination project is set to **locked** (not including nested projects), the permissions for the project being moved aren't overwritten. Whether the moved project is locked or customizable is preserved from its original setting.
- If the destination project is set to **customizable**, the permissions for the project being moved aren't overwritten but they're now editable.

If the project being moved was previously nested under a parent that was *locked (including nested projects)*, when moved, the project takes on the setting of *locked (including nested projects)* and becomes the managing project for any projects it contains.
Note: This is the same outcome if a project is moved to become a top-level project.

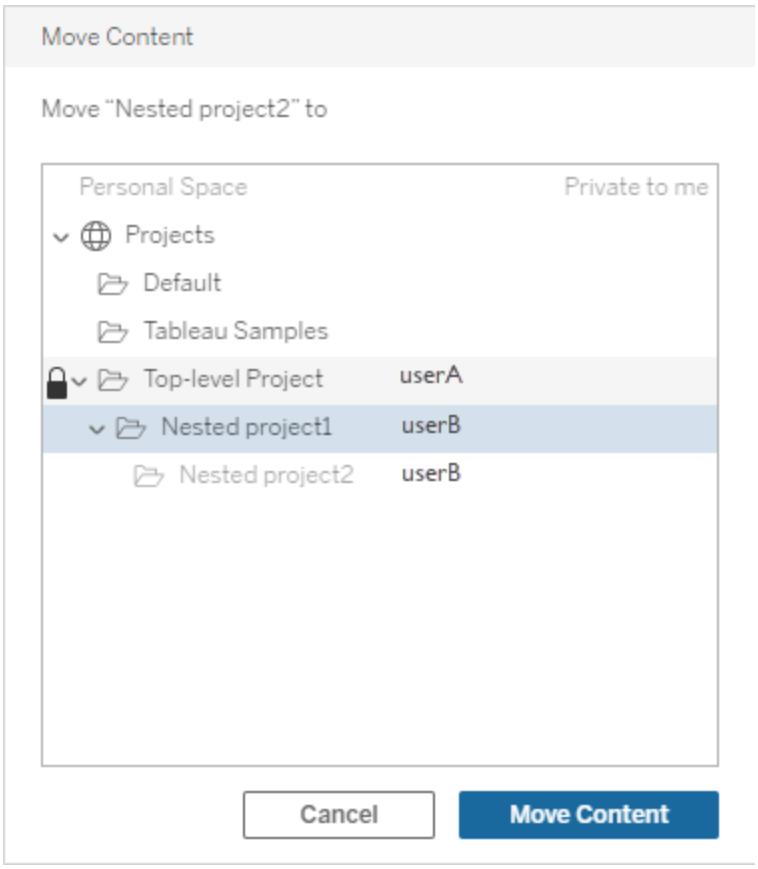
Use care moving locked nested projects

Moving nested projects inside *locked (including nested projects)* environments can be tricky. A project can be moved into situation that prevents the user from moving it out again.

If a nested project is owned by a different user than the managing project, and the managing project is set to *locked (including nested projects)*, a nested project can wind up unable to be moved by anyone except an admin.

For example, consider a locked (including nested projects) top-level project owned by userA, and two nested projects owned by userB. If userB moves one nested project inside the other, they aren't then able to move it back out—and neither is userA.

- UserB can't move **Nested project2** because they don't have rights to move rights on **Top-level Project** as a destination.
- UserA can't move **Nested project2** because they don't have move rights on it.
- A project leader on **Top-level Project** can't move it even though project leader trickles down to nested projects.
- Only an admin can move **Nested project2** in this setup.



Collections

Unlike projects, which contain content, a collection can be thought of as a list of links to content. Project permissions can be inherited by the content in the project, but permissions for a collection have no effect on the content added to the collection. This means that different users might see different numbers of items in a collection, depending on which items they have permission to view. To make sure that users can see all items in a collection, adjust the permissions for those items individually.

Permissions for a collection can be changed either by using the permissions dialog or by granting access upon sharing a collection, if you're an administrator or the collection owner. For more information, see Manage Collection Permissions.

Private collections

When a collection is created, it's private by default. A private collection appears on the owner's My Collections page, but it doesn't appear in the list of all collections on a site. Private collections are simply collections with no permission rules added. Unlike other types of content, collections don't have the "All Users" group added by default. When you add permission rules to a collection, it's no longer flagged as private. To return a collection to a private state, remove the permission rules.

Private collections can be viewed by the collection owner as well as by administrators, whose site role gives them effective permissions to view all collections.

Effective permissions

A permission rule establishes who is impacted (a group set, group, or user) and what Capabilities they are **Allowed**, **Denied**, or **Unspecified**. While it seems straightforward to simply set a permission rule and have that be the whole story, whether a user has a capability may be unclear because of membership in multiple groups and the interplay of site roles and ownership with permission rules.

Multiple factors are evaluated in a specific order, yielding *effective permissions* on a piece of content.

Tip: To help keep things as straightforward as possible, we recommend (1) setting permission rules for groups instead of users, (2) managing permissions locked at the project level instead of setting permissions on individual content, and (3) deleting the All User group's permission rule or setting all capabilities to None.

A capability is allowed for a user if and only if the following three conditions are all met:

- The capability is within the scope of their site role.
- They have that capability:
    - based on a specific user scenario (such as being the content owner or a project leader, or they're an administrator site role),
      OR
    - because they have been allowed the capability as a user,
      OR

  - because they are both in a group that has been allowed the capability and no rules deny them the capability as a user or member of another group.
- There is no conflicting permissions settings at another content level that takes precedence.

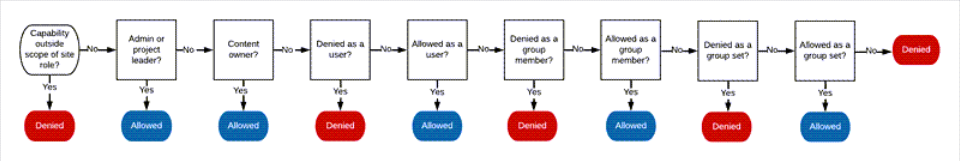
Any other situation denies the user the capability.

Hovering over a capability brings up a tooltip that explains the effective permission. Here are some common examples of why effective permissions—what the user can or can't do in actuality—might appear different than what a given permission rule states:

- A user might have a capability they are denied in a permission rule because their site role includes it (administrators).
- A user might have a capability they are denied in a permission rule because their user scenario allows it (because they own the content or are a project owner or leader).
- A user might lack a capability they are allowed in a permission rule because their site role doesn't allow it.
- A user might lack a capability they are allowed in a permission rule because a conflicting group or user rule denied it.
- A user might lack a capability they are allowed in a permission rule at one level of content (such as a workbook) because another level of content denied it (such as a view).

Evaluate permission rules

Permissions in Tableau are restrictive. Unless a capability is granted to a user, they are denied permission. The following logic evaluates if a capability is allowed or denied for an individual:



1. **Site role:** If a site role doesn't permit a capability, the user is denied. If the user's site role does permit the capability, then specific user scenarios are evaluated.
   - For example, a Viewer site role can't web edit. See General capabilities allowed with each site role for more information on what each site role can do.

2. **Specific user scenarios:**
    - If the user is an admin they have all capabilities on all content.
    - If the user is a project owner or project leader, they have all capabilities on all content in their projects.
    - If the user is the content owner, they have all capabilities* on their content.
    - If these scenarios do not apply to the user, then user rules are evaluated.

    *Exception: Content owners won't have the **Set Permissions** capability in projects where permissions are locked. Only administrators, project owners, and project leaders can set permission rules in locked projects.

3. **User rules:** If the user is denied a capability, it is denied. If they are allowed a capability, it is allowed. If a capability is unspecified, then group rules are evaluated.
4. **Group rules:** If the user is in *any* group that is denied a capability, it is denied. If the user is in a group that is allowed a capability (and not in any groups that are denied that capability), it is allowed.
    - That is to say, if a user is a member in two groups, and one is allowed a capability and one is denied the same capability, the denial takes precedence for that user and they are denied.
5. **Group set rules:** If a user is a member of a group in a group set, any group in the group set that is denied a capability, is then denied.
6. If none of the above conditions apply, the user is denied that capability. In effect, this means that capabilities left as unspecified will result in denied.

A final effective permission of **Allowed** therefore occurs in three circumstances:

- Allowed by site role (Server Administrator, Site Administrator Creator, Site Administrator Explorer)
- Allowed because the user is the content owner, project owner, or project leader
- Allowed by a group, group set, or user rule (and not denied by a rule of higher precedence)

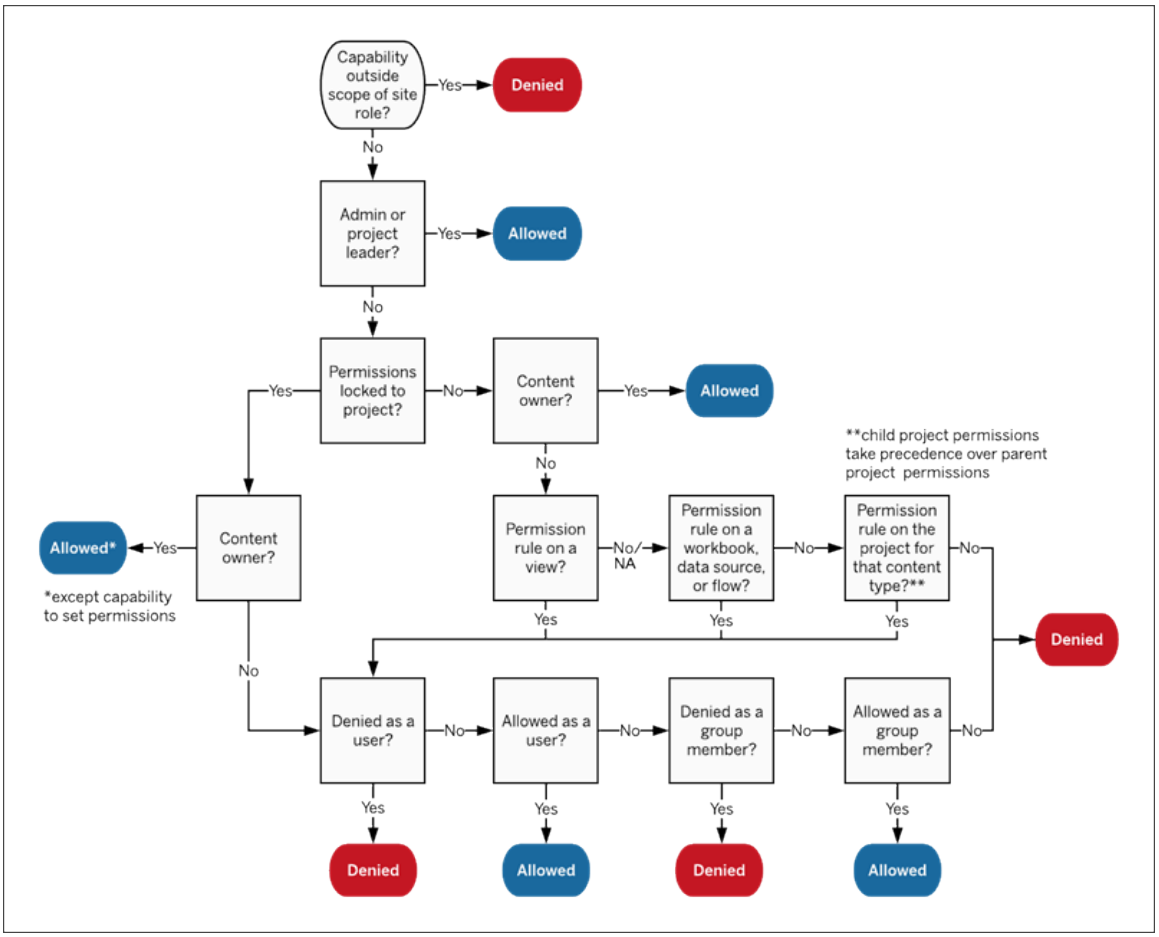
Denied occurs in three circumstances:

- Denied by site role
- Denied by a rule (and not allowed by a rule of higher precedence)
- Not granted by any rule

Evaluate permissions set at multiple levels

If **Asset permissions** are set to **Customizable**, it's possible to configure permission rules in multiple places. There are specific rules that determine what permissions are applied on the content.

- If there are nested projects, permissions set at the child level take precedence over permissions set at the parent level.
- Changes to permissions at the project level are not enforced for existing content.
- If there are permissions set on content (workbook, data source, or flow) during or after publication, these take precedence over rules set at the project level.
- If a workbook doesn't show navigational sheet tabs, any changes to the workbook-level permissions *won't* be inherited by the views and any changes to permissions must be done on the view.
- Configuring the workbook to show navigational sheet tabs will override existing view-level permissions and sync them with the workbook-level permissions. See Show or Hide Sheet Tabs.



This image shows how capabilities are evaluated through multiple levels of content.

Permissions on views

In a workbook that is not in a locked project and does not show sheets as tabs for navigation, views (sheets, dashboards, stories) inherit the workbook permissions at publication, but any changes to permission rules must be made on individual views. View capabilities are the same as those for workbooks, except for **Overwrite**, **Download Workbook/Save a Copy**, and **Move** which are only available at the workbook level.

We recommend showing navigational sheet tabs whenever possible so views continue to inherit their permissions from the workbook. For more information, see Show or Hide Sheet Tabs.

Permissions, Site Roles, and Licenses

Adding a user to Tableau Cloud requires an available license. (Users can also be added as unlicensed and configured so they will consume a license only when they first sign in. For more information, see Grant License on Sign In.) For each site the user belongs to they have exactly one site role, restricted by their license. A user has permissions for content on the site, restricted by what their site role allows.

Licenses and site roles apply to users. Permission capabilities apply to content.

Licenses are assigned to a user when they are created (or sign in for the first time) on the Tableau Server or Tableau Cloud site. Users are licensed as a **Creator**, **Explorer**, or **Viewer**.

- License levels are consumed based on the maximum *site role* a user can have on that server.
    - Server Administrator, Site Administrator Creator, and Creator site roles use a Creator license.
    - Site Administrator Explorer, Explorer (can publish), and Explorer site roles use at least an Explorer license.
    - Viewer site role uses at least a Viewer license.
    - An unlicensed user can exist on the site, but they cannot sign in unless they were added with grant site role on sign in.
- For Tableau Server, a user consumes only one license per server, even if they are a member of multiple sites. If a user is a member of multiple sites, their required license level is determined by their highest site role. (For example, if a user has a Creator site role in one site and a Viewer site role in two others, they consume a Creator license.)

Site roles are assigned to a user for each site they are a member of.

- Site roles determine the maximum capabilities a user can have in that site. (For example, a user with a site role of Viewer will never be able to download a data source even if that capability is explicitly granted to them on a specific data source.)
- Site roles do not inherently grant any capabilities in and of themselves—with the exception of the administrator site roles. Administrators always have all capabilities applicable to their license level.

Permissions consist of *capabilities*, like the ability to save to a project, web edit a workbook, connect to a data source, etc. They apply to group or user on a specific piece of content (project, data source, workbook, view, or flow).

- Permission capabilities are not given to a group or user in a vacuum but rather in the context of content. A user can have different capabilities for different content assets.
- Permissions are evaluated based on the interplay of a user's site role and the permission rules for that user or any groups they are members of.
- Some actions such as web authoring might require combinations of capabilities. For more information, see Permission settings for specific scenarios.

Site roles and their maximum capabilities

These tables indicate what capabilities are available for a site role. There may be other ways for a user with a site role to perform a similar action. For example, although Viewers can't be given the **Share Customized** capability to make their custom views visible to others on the workbook, they can share custom views by copying the view URL. See General capabilities allowed with each site role for more information on what each site role can do.

Projects

Capability	Creator	Explorer (can publish)	Explorer	Viewer
View	✓	✓	✓	✓
Publish	✓	✓	✗	✗

Workbooks

Capability	Creator	Explorer (can publish)	Explorer	Viewer
View	✓	✓	✓	✓
Filter	✓	✓	✓	✓

View Comments	✓	✓	✓	✓
Add Comments	✓	✓	✓	✓
Download Image/PDF	✓	✓	✓	✓
Download Summary Data	✓	✓	✓	✓
Run Explain Data †	✓	✓	✓	✓
Share Customized	✓	✓	✓	✗
Download Full Data	✓	✓	✓	✗
Web Edit	✓	✓	✓	✗
Download Workbook/Save a Copy	✓	✓	✓	✗
Overwrite	✓	✓	✗	✗
Create/Refresh Metrics ‡	✓	✓	✗	✗
Move	✓	✓	*	✗

Delete	✓	✓	✗	✗
Set Permissions	✓	✓	✗	✗

† Prior to Tableau 2021.3, the availability of Explain Data was controlled at the server level only using the tsm configuration set option ExplainDataEnabled. In 2021.3 and later, availability of Explain Data can be controlled in site settings and in a workbook using the Run Explain Data capability. The availability of Explain Data in viewing mode is controlled in a workbook in the Explain Data Settings dialog box.

‡ Prior to Tableau 2021.3, the Create/Refresh Metrics capability was controlled by the Download Full Data capability.

Data Sources

Capability	Creator	Explorer (can publish)	Explorer	Viewer
View	✓	✓	✓	✓
Connect	✓	✓	✓	✓
Download Data Source	✓	✓	✓	✗
Overwrite	✓	✓	✗	✗
Delete	✓	✓	✗	✗
Set Permissions	✓	✓	✗	✗

Data Roles

Capability	Creator	Explorer (can publish)	Explorer	Viewer
View	✓	✓	✓	✓
Overwrite	✓	✓	✗	✗
Move	✓	✓	*	✗
Delete	✓	✓	✗	✗
Set Permissions	✓	✓	✗	✗

Flows

To run flows on a schedule, you must have a Data Management license. For information about configuring flow settings, see Create and Interact with Flows on the Web. Explorer license users can run flows on Tableau Cloud.

Capability	Creator	Explorer (can publish)	Explorer	Viewer
View	✓	✓	✓	✓
Download Flow	✓	✓	✓	✗
Web Edit	✓	✗	✗	✗
Run Flow	✓	✓	✗	✗
Overwrite	✓	✗	✗	✗

Move	✓	✓	*	✗
Delete	✓	✓	✗	✗
Set Permissions	✓	✓	✗	✗

Ask Data Lenses

Capability	Creator	Explorer (can publish)	Explorer	Viewer
View	✓	✓	✓	✓
Overwrite	✓	✓	✗	✗
Move	✓	✓	*	✗
Delete	✓	✓	✗	✗
Set Permissions	✓	✓	✗	✗

Metrics

Retirement of the legacy metrics feature

Tableau's legacy metrics feature was retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2. In October 2023, Tableau retired the ability to embed legacy metrics in Tableau Cloud and in Tableau Server version 2023.3. With Tableau Pulse, we've developed an improved experience to track metrics and ask questions of your data. For more information, see Create Metrics with Tableau Pulse to learn about the new experience and Create and Troubleshoot Metrics (Retired) for the retired feature.

Capability	Creator	Explorer (can publish)	Explorer	Viewer
View	✓	✓	✓	✓
Overwrite	✓	✓	✗	✗
Move	✓	✓	*	✗
Delete	✓	✓	✗	✗
Set Permissions	✓	✓	✗	✗

Collections

Capability	Creator	Explorer (can publish)	Explorer	Viewer
View	✓	✓	✓	✓

Virtual Connections

Virtual connections require a Data Management license. See About Data Management for details.

Capability	Creator	Explorer (can publish)	Explorer	Viewer
View	✓	✓	✓	✓
Connect	✓	**	**	**
Overwrite	✓	✗	✗	✗

Move	✓	✓	*	✗
Delete	✓	✓	✗	✗
Set Permissions	✓	✓	✗	✗

* Although the Explorer role can be given the **Move** capability, they can't have the **Publish** capability on a project and therefore there is no place for them to move content to. The **Move** capability should therefore be considered not possible for Explorer site roles.

** Although the Explorer (can publish) role can be given the **Connect** capability for Virtual Connections, the ability to create a new data source of any kind, including Virtual Connections, is only available for users with a Creator site role. Similarly, Explorer and Viewer role users can't access the UI to connect to new or existing data sources. The **Connect** capability should be considered not possible for any role but Creator.

Quick Start: Permissions

A permission rule is a set of capabilities that defines what access a group or user has to a piece of content, such as a workbook, project, or data source.

To efficiently manage permissions:

- Remove permissions from the **All Users** group before creating more groups
- Configure template permissions on the **Default** project before creating more projects
- Manage permissions for groups, not users
- Manage permissions for projects, not content

Create group permission rules for projects

For details on the following steps, see the main article on permissions. This Quick Start guide is an overview and doesn't capture many important details about permissions and permission management.

1. Add users to groups

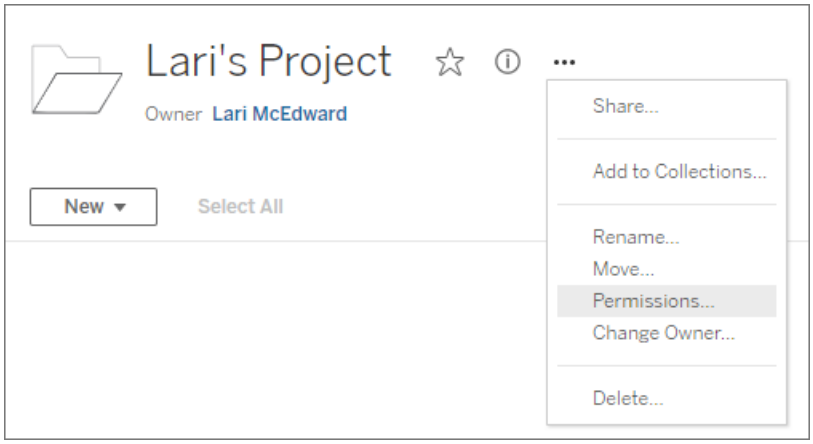
A common way to manage permissions is to use groups for users who should have the same permissions.

1. If necessary, add users to the site.
2. Within a site, select **Groups**.
3. If necessary, create a group using the **Add Group** option.
4. Click a group name to open it, then use the **Add Users** button to add existing users to the group.

2. Access project-level permissions settings

The **Explore** page displays the content on the site. Use the dropdown to display Top-Level Projects or All Projects (to see nested projects as well).

Navigate to the project you want to update, open the **Actions** (**…**) menu, then select **Permissions**.

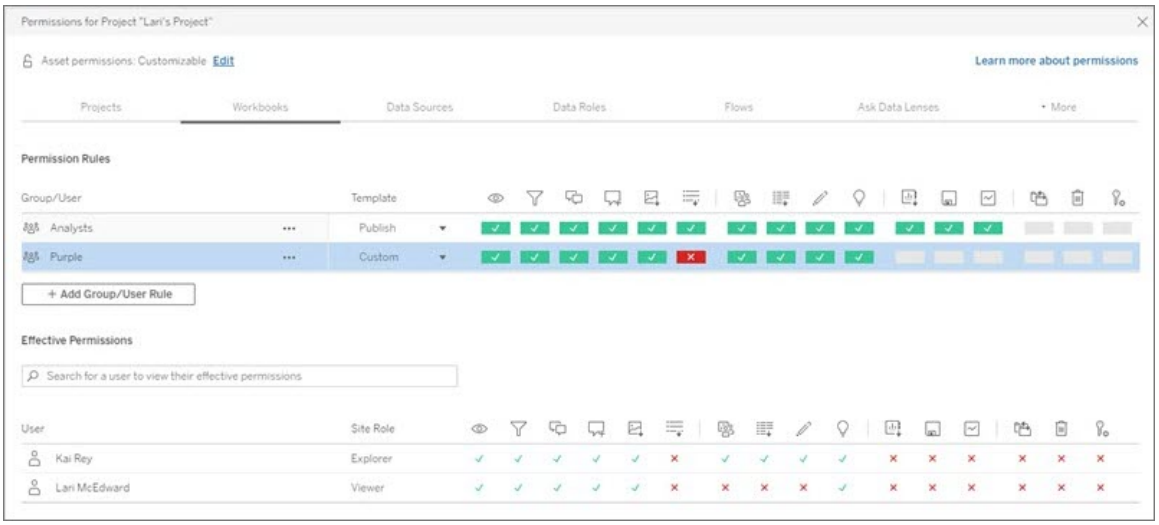


3. Create a permissions rule

Select **+ Add Group/User Rule** to create a new permission rule.

The template drop-down offers a shortcut to apply an initial set of capabilities for the group.

If desired, customize the permission rule by clicking a capability to set it to **Allowed** or **Denied**, or leave it **Unspecified**.



Whether a user can set permissions is based on their site role, content ownership, and how their **Set Permissions** capability is set.

4. View a user's effective permissions

After you save the permissions rule for the group, you can view the effective permissions for each user. Click a group name to see the group's users and their permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

Site roles

A user's site role determines the maximum permissions allowed for that user.

- Server and site administrators can access all site content and take actions on it.
- Owners always get full access to the content they've published. When the parent project permissions aren't locked, owners can change permissions for their published content.

For more information, see Set Users' Site Roles and Use Projects to Manage Content Access.

Permission logic

- **Denied** takes precedence over **Allowed**.
- **Unspecified** results in **Denied** if no other permissions are specified.
- Specific user permissions on content take precedence over group permissions on content. In other words, user permissions trump group permissions.

For more information, see Effective permissions.

Manage Content Ownership

When you publish a data source or workbook on Tableau Server or when you create a project, you become its *owner*. A content owner, a project leader with an appropriate site role, or an administrator can change ownership of a content asset. After ownership is reassigned, the original owner has no special connection to the content item, and their ability to access it is determined by their permissions on the project or that specific item.

Who can change or be given ownership, by content type

Whether you can change or be given ownership depends on your permissions and your relationship to the content asset, as described in the following table.

Note: Full project leader access is available only with some site roles. For information, see Project-level administration.

Content asset type	Who can change ownership	Who can be given ownership
Top-level projects	Server administrator[1] Site administrator	Server administrator Site administrator (Creator and Explorer) Creator Explorer (can publish)
Child (nested) projects	Server administrator Site administrator Project owner	Any administrator or owner, excluding Explorer and Viewer.
Workbooks and data	Server administrator	Server administrator

sources	Site administrator Workbook or data source owner Project leader or owner of the project that contains the item	Site administrator Creator Explorer Viewer
Metrics (The legacy Metrics feature was retired in February 2024 for Tableau Cloud and in Tableau Server version 2024.2. For more information, see Create and Troubleshoot Metrics (Retired).)	Server administrator Site administrator Metric owner Project leader or owner of the project that contains the item	Any administrator or user of the site, excluding Explorer and Viewer.
Ask Data lenses	Server administrator Site administrator Lens owner Project leader or owner of the project that contains the item	Any administrator or user of the site, excluding Explorer and Viewer.
Flows	Server administrator Site administrator	Starting in version 2021.2, server and site administrators can only change the owner to themselves.

Data Roles	Server administrator Site administrator Data role owner Project leader or owner of the project that contains the item	Any administrator or user of the site, excluding Explorer and Viewer.
Collections	Server administrator Site administrator Collection owner	Server administrator Site administrator Creator Explorer Viewer
Virtual Connections[2]	Server administrator Site administrator Virtual connection owner	Server administrator Site administrator Creator

[1] The Server Administrator site role applies to Tableau Server only; not Tableau Cloud.

[2] Virtual connections require Data Management. See About Data Management for details. Note that to edit a virtual connection, you must have the database credentials.

Considerations for changing content ownership

- Before you remove a user from Tableau Server, make sure they do not own any content assets.

  If the user does own content, you must first reassign ownership of those assets before you can delete the user. Otherwise, their site role is set to **Unlicensed**, but they are not

deleted, and only an administrator can take certain actions on that content. Reassign ownership of workbooks or data sources with embedded credentials before you set the user's site role to Unlicensed or delete the user.

- If you change the ownership of a workbook or data source that includes embedded credentials to connect to underlying data, the embedded credentials will be deleted. For flows, embedded credentials are preserved when changing ownership. Connections to published data sources are authenticated using the flow owner and authorized based on their permissions.

  You can update the embedded credentials by editing the connection information on Tableau Server. For more information, see Edit Connections. Alternatively, the new owner can download the flow, workbook, or data source and open the item in Tableau Desktop to update the embedded credentials and then re-publish the content.

- If you do not lock permissions to projects, make sure users you give content ownership to know your permissions guidelines, or you account for permissions as you change ownership. In unlocked projects, by default, content owners can set permissions on their content. For more information, see Permissions.
- While it is possible to change the owner of a metric to a user with a site role of Viewer or Explorer, it is not recommended, because doing so will cause the metric refresh to be suspended. A site role of Creator or Explorer (can publish) is required to refresh, overwrite, or delete a metric.

Change the owner of a content resource

1. Sign in to the Tableau Server web environment, and from the navigation menu, select **Explore**.
2. Navigate to the content you want to assign to someone else.

   - If you want to reassign multiple of the same type of content, for example, multiple workbooks, select the content type from the drop-down menu.

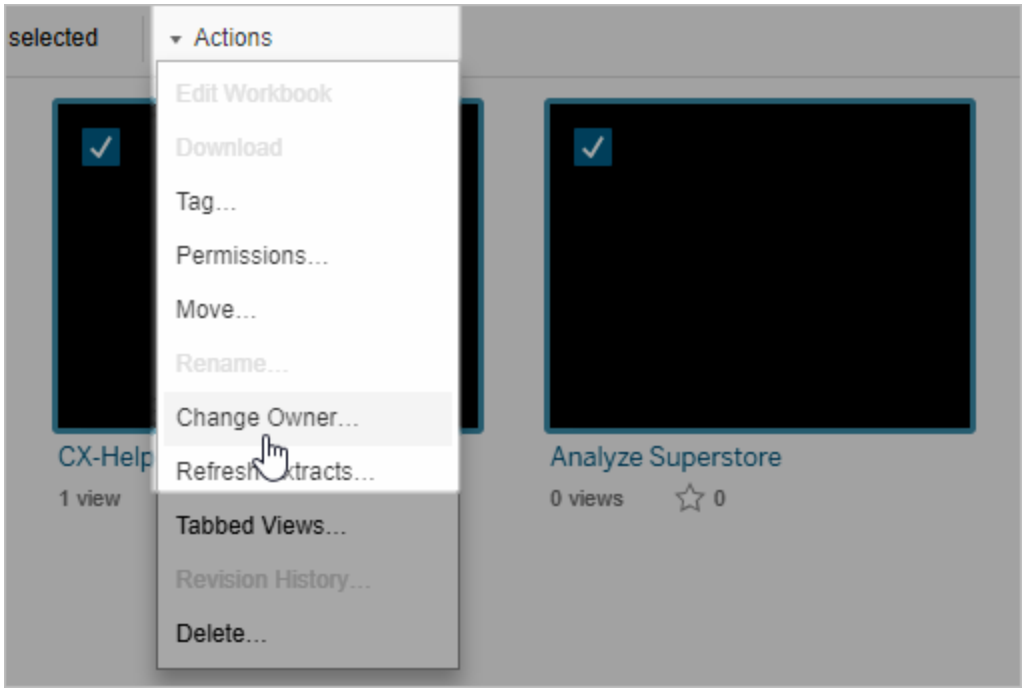
- If you want to reassign multiple items within the same project, navigate to the project.

  If you're not sure where to find a child project, display filters, and select **Show all projects**.

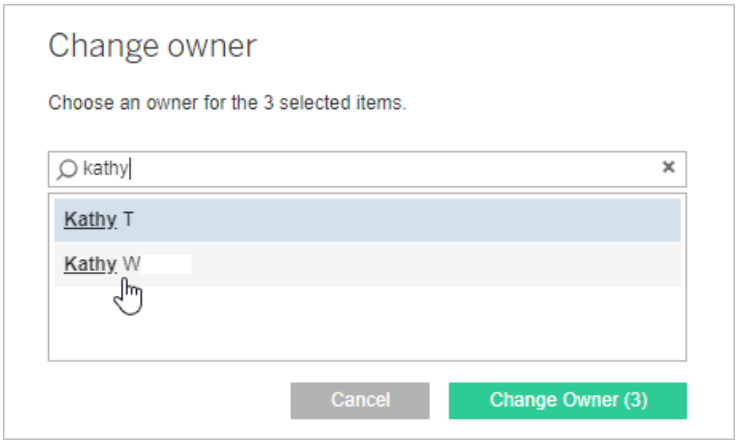
- If you want to reassign multiple content items with the same owner, find the user on the **Users** page.

3. Select the items you want to reassign, and then select **Actions** > **Change Owner**.

   The other menu commands you see will depend on the content type.



4. Type the name of a user or select a user from the list.



5. Click **Change Owner**.

Manage Permissions for External Assets

Tableau Cloud and Tableau Server provide a space for accessing and managing published content. When Tableau Cloud or Tableau Server is licensed with Data Management, you have access to Tableau Catalog. Tableau Catalog adds a complementary space and a set of features across your site to track and manage metadata and lineage of external assets used by the content published to your site.

Tableau Catalog indexes content and assets

Catalog discovers, tracks, and stores metadata from the content that you publish to Tableau Cloud or Tableau Server.

Catalog indexes metadata for the following:

- **Tableau content:** workbooks, data sources, flows, projects, metrics, virtual connections, virtual connection tables, users, and sites. (The legacy Metrics feature was retired in February 2024 for Tableau Cloud and in Tableau Server version 2024.2. For more information, see Create and Troubleshoot Metrics (Retired).)
- **External assets:** databases and tables associated with Tableau content

Catalog classifies the metadata of any data that comes from outside the Tableau environment as external assets. The data that comes from outside the Tableau environment is stored in many different formats, such as a database server or a local .json file.

Catalog tracks only the metadata of the external data and does not track the underlying data in any form (raw or aggregated).

Catalog metadata includes the following:

- **Lineage information** or the relationship between items. For example, the Sales table has a relationship with both the Superstore data source and the Superstore Sample workbook.
- **Schema information**. Some examples include:
  - Table names, column names, and column types. For example, Table A contains Columns A, B, and C, which are types INT, VARCHAR, and VARCHAR.
  - Database name and server location. For example, Database_1 is a SQL Server database at http://example.net.
  - Data source name, and the names and types of the fields the data source contains. For example, Superstore data source has fields AA, BB, and CC. Field CC is a calculated field that refers back to both field AA and field BB.
- **User curated, added, or managed information**. For example, item descriptions, certifications, user contacts, data quality warnings, and more.

How does Tableau Catalog work?

Tableau Catalog indexes all content published to Tableau Cloud or Tableau Server to track lineage and schema metadata. For example, the metadata comes from workbooks, packaged workbooks, data sources, and the Tableau Server or Tableau Cloud repository.

As part of the indexing process, lineage and schema metadata about external assets (databases, tables, and other objects) used by the published content are also indexed.

Note: In addition to accessing Catalog from Tableau Cloud or Tableau Server, indexed metadata can also be accessed from the Tableau Metadata API and Tableau Server REST API. For more information about the Tableau Metadata API or metadata methods in the

REST API, see Tableau Metadata API and Metadata Methods in the Tableau Server REST API, respectively.

Permissions on metadata

Permissions control who is allowed to see and manage external assets and what metadata is shown through lineage.

Note: If Tableau Cloud or Tableau Server is not licensed with Data Management, then by default, only admins can see database and table metadata through the Tableau Metadata API. This default can be changed to use "derived permissions," as described below.

Access metadata

The permissions used to access metadata through Catalog (or Metadata API) work similarly to permissions for accessing content through Tableau Cloud or Tableau Server, with some additional considerations for sensitive data that can be exposed through lineage and the capabilities granted on external assets.

Permissions on Tableau content

Catalog follows the view and manage capabilities that are already in place on existing Tableau content to control the metadata that you can see and manage on Tableau content. For more general information on these capabilities, see Permissions.

Permissions on external assets using derived permissions

When Tableau Cloud or Tableau Server is licensed with Data Management, by default Catalog uses *derived permissions* to automatically grant you capabilities to external assets in the following scenarios:

For **View** capability:

- If you are the owner of a workbook, data source, or flow, you can see the database and table metadata used *directly* by that workbook, data source, or flow. See Additional

notes about lineage.

- If you are a project owner or project leader, you can see all the database and table metadata used by the content published to your project.
- Embedded files use the permissions of the source content (such as the workbook, data source, or flow), rather than the derived permissions of the external asset (the database or table). For example, if you can see a workbook with an embedded file, you can see the embedded file and its metadata used by that workbook.

For both **Overwrite** and **Set Permissions** capabilities:

- If you are the owner of a flow, you can edit and manage permissions for the database and table metadata used by the flow output.

Note: In the case of flows, the capabilities mentioned above apply only after the flow has been run successfully at least once under the current owner of the flow.

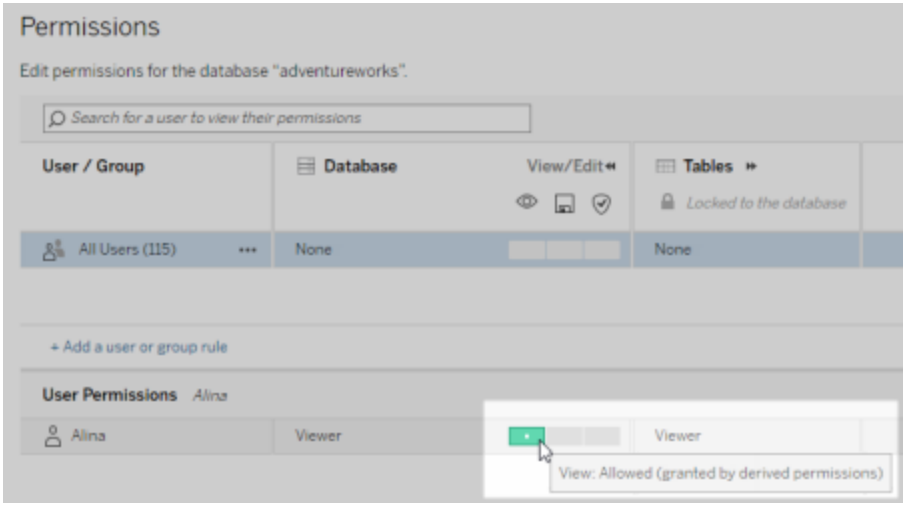
Check permissions

As an admin or someone who has been given the capability to set permissions for an asset, you can validate who has derived permissions by following the steps below.

1. Sign in to Tableau Cloud or Tableau Server.
2. From the left navigation pane, click **External Assets**.
3. From the drop-down menu, select **Databases and Files** or **Tables and Objects.**
   Note: Local files, like .json or .csv files are grouped as external assets under **Databases**.
4. Select the check box next to the database or table whose permissions you want to modify, and then select **Actions** > **Permissions**.

5. In the Permissions dialog box, click **+ Add Group/User Rule** and start typing to search for a group or user.

6. Validate the permissions by clicking a group name or user name in the permission rules to see the effective permissions below.



Order of precedence for derived permissions on external assets

When derived permissions are configured for your Tableau Cloud site or Tableau Server, each user's level of access to external assets depends on the associated Tableau content and the order of precedence of rules Tableau uses for its content.

Tableau follows the rules below, continuing on to the next rule, only if the current rule evaluates to "denied." If any rule evaluates to "allowed," the capability is allowed and Tableau stops evaluating. This rules list is based on the Permissions.

For **View** capability:

1. Admin role
2. License
3. Project leader (Tableau content)
4. Project owner (Tableau content)
5. Content owner (Tableau content)

6. *Derived permissions* (applies only to external assets and the View capability)
    a. Admin role
    b. License
    c. Project leader (external assets)
    d. Project owner (external assets)
    e. Content owner (external assets)
7. Explicit permissions

For **Overwrite** and **Set Permissions** capabilities:

1. Admin role
2. License
3. Project leader (Tableau content)
4. Project owner (Tableau content)
5. Content owner (Tableau content)
6. Explicit permissions (Tableau content)
7. *Derived permissions* (applies only to external assets and the Overwrite and Set Permissions capabilities for flow outputs)
    a. Admin role
    b. License
    c. Project leader (external assets)
    d. Project owner (external assets)
    e. Content owner (external assets)

Turn off derived permissions

As an admin, you can turn off the derived permissions default setting for a site in favor of manually granting explicit permissions to databases and tables.

1. Sign in to Tableau Cloud or Tableau Server as an admin.
2. From the left navigation pane, click **Settings**.
3. On the **General** tab, under **Automatic Access to Metadata about Databases and Tables**, clear the **Automatically grant authorized users access to metadata about databases and tables** check box.

Note: Data quality warning messages on databases and tables that are visible to users though derived permissions remain visible to those users even when the check box is not selected.

Set permissions on individual external assets

In order to grant additional users permissions to view, edit (overwrite), and manage external assets, an admin can grant those capabilities explicitly on individual databases or tables for users or groups.

Starting with Tableau Server 2022.3 and Tableau Cloud September 2022, you can organize external assets in projects. Permissions inheritance for external assets works the same way as it does for Tableau content, as described in the Permissions topic, and can simplify permissions management.

Summary of permissions capabilities

The following table shows the capabilities you can set for external assets:

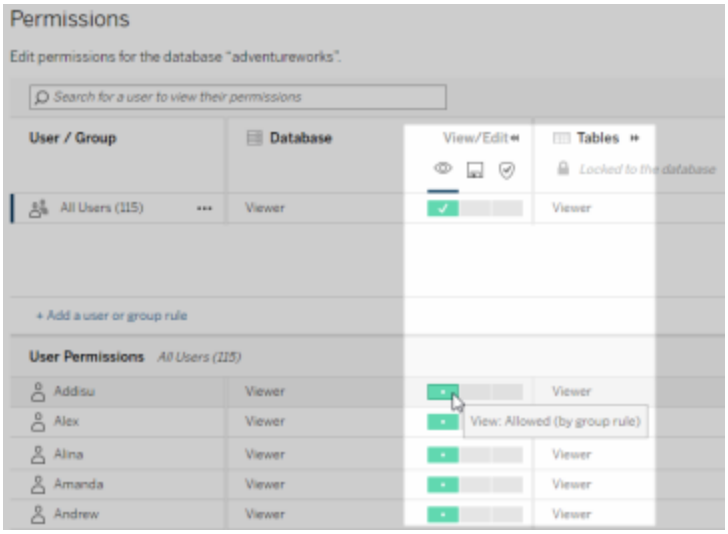
Capability	Description	Template
 View	See the database or table asset.	View
 Overwrite	Add or edit data quality warnings and descriptions of the database or table asset. Prior to version 2020.1, the Overwrite capability was called Save.	Publish
 Move	Move the database or table asset.	Administer
 Set Permissions	Grant or deny permissions for the database or table asset.	Administer

Set permissions on a database or table

To set permissions on databases or tables, use the following procedure.

1. Sign in to Tableau Cloud or Tableau Server as an admin or someone who has been granted the "Set Permissions" capability.
2. Find the database or table. You can do this through **Explore** (starting with Tableau Server 2022.3 and Tableau Cloud September 2022) if you know the current location of the database or table, or through **External Assets** to see a list of all databases, tables, and files.
    - **Explore** - From the left navigation pane, click **Explore** and locate the project the database or table is in.
    - **External Assets** - From the left navigation pane, click **External Assets**. From the drop-down menu, select **Databases and Files** or **Tables and Objects** (Note: Local files, like .json or .csv files are grouped as external assets under Databases.)
3. Select the check box next to the database or table whose permissions you want to modify, and then select **Actions** > **Permissions**.
4. In the Permissions dialog box, click **+ Add Group/User Rule** and start typing to search for a group or user.
5. Select a permission role template to apply an initial set of capability for the group or user, and then click **Save**. Available templates are: View, Publish, Administer, None, and Denied.
6. To further customize the rule, click a capability in the rule to set it to Allowed or Denied, or leave it unspecified. Click save when you are done.
7. Configure any additional rules you want for other groups or users.
8. Validate the permissions clicking a group name or user name in the permission rules to

see the effective permissions below.



External assets that are not in projects

There are some scenarios in which an external asset is not in a project:

- External assets that Catalog discovered before the **External Assets Default Project** existed (Tableau Cloud December 2022 / Server 2023.1) will not be in a project unless they've been moved into one since then.
- External assets that had their project deleted before the **External Assets Default Project** existed (Tableau Cloud December 2022 / Server 2023.1) will not be in a project unless they've been moved into one since then.
- In Tableau Server 2022.1 and earlier, external assets cannot be moved to projects at all.

If an external asset is not in a project, permissions for external assets work as they did in Tableau Server 2022.1 and Tableau Cloud June 2022 and earlier. That is, database and table permissions are controlled independently of content in projects, and table permissions can be managed through database permissions. When permissions are set at the database level in this way, those permissions can serve as a template for any newly discovered and indexed

child tables of that database. Furthermore, database permissions can also be locked so that the child tables will always use the permissions set at the database level.

> Note: You cannot lock (or unlock) permissions to a database if the database is in a project.

To lock (or unlock) permissions to the database, use the following procedure:

1. Sign in to Tableau Cloud or Tableau Server as an admin or someone who has been granted the "Set Permissions" capability.
2. From the left navigation pane, click **External Assets**. By default, the External Assets page shows a list of databases and files.
3. Select the check box next to the database whose permissions you want to lock, select **Actions** > **Permissions**, and then click the Table Permissions **Edit** link .
4. In the Table Permissions in Database dialog box, select **Locked** and click **Save**.
5. To unlock permissions, click **Edit** again, and select **Customized**.

Access lineage information

Catalog (and the Metadata API) can expose relationship and dependencies metadata, also referred to as *lineage*, among the Tableau content and external assets on Tableau Cloud or Tableau Server. Lineage shows three primary things:

- How items relate to each other, either directly or indirectly
- How many of those items relate to each other
- With the appropriate permissions, shows sensitive data about items in the lineage

Sensitive lineage data

In some cases, lineage can contain sensitive data, such as data quality warning messages, content or asset names, or related items and metadata.

By default, complete lineage information displays for all users while its sensitive data is blocked from specific users who don’t have the appropriate View capabilities. The concept of blocking sensitive data is called obfuscation.

Obfuscation allows all metadata in the lineage to be visible while keeping its sensitive data blocked from specific users who don’t have the appropriate View capabilities. This default enables workflows that rely on a complete impact analysis.

If obfuscating sensitive data in the lineage is not enough for your organization, certain parts of the lineage, including its sensitive data, can be filtered.

Filtering omits certain parts of the lineage (and lineage-related areas like data details) for specific users who don't have the appropriate **View** capabilities for its sensitive data. Because filtering omits parts of lineage, it prevents workflows that rely on a complete impact analysis.

To change how sensitive data is handled, do the following:

1. Sign in to Tableau Cloud or Tableau Server as an admin.
2. From the left navigation pane, click **Settings**.
3. On the General tab, under **Sensitive Lineage Information**, select the radio button that best handles lineage information for all users on your Tableau Cloud site or Tableau Server.

Additional notes about lineage

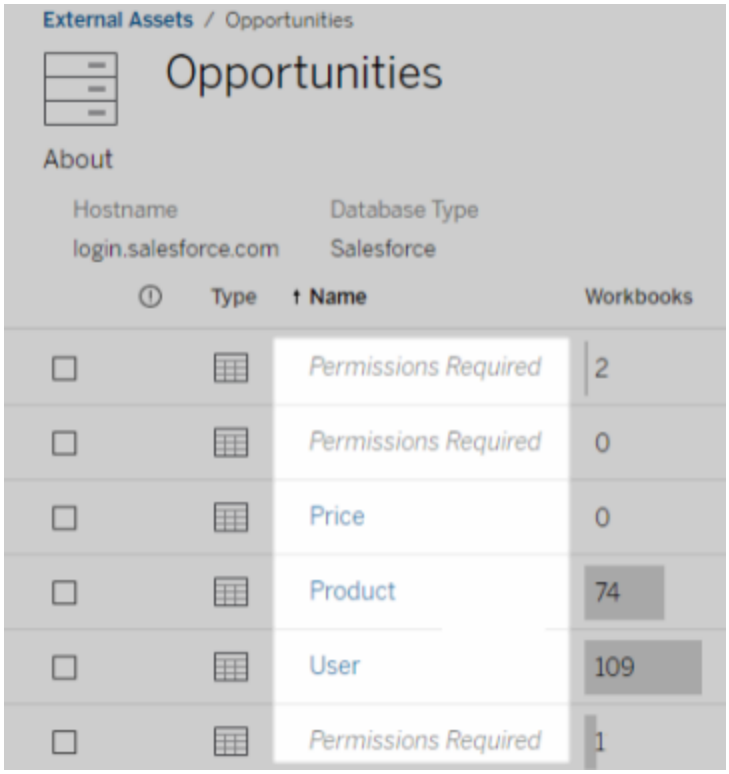
- **If you have the View capability on related assets**, you can see when and what assets and content are related to each other, and their sensitive metadata.

  For example, you can see 1) the names, data quality warnings, and total number of related upstream databases and tables and 2) the combined number of sheets (visible and hidden) in the lineage of the downstream workbook of the asset you are evaluating.

- **If you *don't* have the View capability on related assets**, you can always see when assets relate to each other.

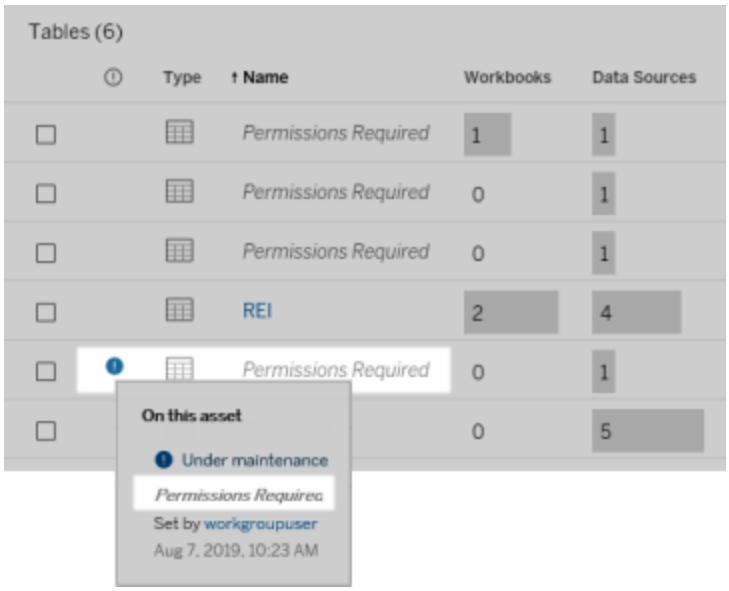
For example, you can see 1) whether related upstream databases and tables exist in the lineage and 2) the total number of databases or total number of tables that are related to the asset you are evaluating.

However, you can't see the metadata associated with those assets when you don't have the view capability for them. When metadata is blocked because of limited permissions, or the asset is in a Personal Space, you see **Permissions Required**.



- **If you *don't* have the View capability on related assets**, you can always see whether the assets are certified.

  However, if you don't have View capability, you can't see sensitive information related to the certification, like the names of the related databases and tables. When metadata is blocked because of limited permissions, or the asset is in a Personal Space, you see **Permissions Required**.



For more information about lineage see Use Lineage for Impact Analysis.

Additional notes about tags discoverable through lineage data

In addition to Tableau content, external assets can also be tagged. Although tags are always visible, tagged items that you see through lineage data can either be obfuscated (default) or filtered as described earlier in this topic.

When tagged items are obfuscated:

- **If you have the View capability for tagged items**, you can see the tagged items and related tagged items, and all metadata.
- **If you don’t have the View capability for tagged items**:
    - You can see the type of tagged and related tagged items but you can't see sensitive metadata about the items. For example, suppose you use a tag filter to see items with the tag “Noteworthy.” Although you can see that there are database items tagged with "Noteworthy," you can’t see the names of the tagged databases.

- You can see how many related tagged items there are. For example, suppose you do a tag query on “Noteworthy.” Your query returns five tagged databases.

When tagged items are filtered, the tagged and related tagged items you see are limited to only the items that you have the View capability for.

For more information about tags, see Tagged Items in the Tableau User Help.

Potential mismatch between asset results and content results

When Catalog shows lineage information, it provides information about Tableau content and external assets. Catalog lineage always shows the true count or result of associated items. However, in other areas of the site, you might see fewer items. This could be because of your **View** capabilities. Outside of Catalog, you see only the content that your permissions allow.

For example, suppose you're looking at the Superstore data source. The lineage for the Superstore data source shows how many upstream underlying tables the data source connects to and how many downstream workbooks rely on the data source. However, because you might not have permissions to see all of those downstream workbooks, the number of related workbooks in the Catalog lineage (actual total) might be greater than the number of workbooks in the **Connected Workbooks** tab (what you have permission to see).

There might be other reasons, unrelated to permissions, why you might see a mismatch between asset counts and content counts. For more information, see Use Lineage for Impact Analysis.

Who can do this

The following information summarizes the types of users who can do the tasks described in this topic.

Tableau Cloud site or Tableau Server admin

Data Management	Capability	Requirements
Licensed	See assets and their metadata	None

Data Management	Capability	Requirements
	Edit assets and their metadata	None
	Change permission on assets and their metadata	None
	Grant users ability to see assets and their metadata	**Default:** When “derived permissions” is on, your users can see metadata on external assets for the content that they own, or for the content that is published to a project that they are a project leader or project owner of. **Ad-hoc:** You can configure explicit **View** permissions on a specified external asset.
	Grant users ability to edit assets and their metadata	You can configure explicit "write" or **Overwrite** permissions on a specified external asset (if not automatically granted because the user is a flow owner) .
	Grant users ability to change permissions on assets and their metadata	You can configure explicit "edit" or **Set Permissions** on a specified external asset (if not automatically granted because the user is a flow owner) .

Data Management	Capability	Requirements
Not licensed	See all assets and their metadata	**Applies to Metadata API only**
	Edit assets and their metadata	Requires Data Management
	Change permission on assets and their metadata	Requires Data Management
	Grant users ability to see assets and their metadata	**Applies to Metadata API only:** You can turn on derived permissions as described above. If “derived permissions” is on, your users can see metadata on external assets for the content that they own, or for the content that is published to a project that they are a project leader or project owner of.
	Grant users ability to edit assets and their metadata	Requires Data Management
	Grant users ability to change permissions on assets and their metadata	Requires Data Management

User with Creator or Explorer license

Data Management	Capability	Requirements
Licensed	See assets and their metadata	**Default:** When "derived permissions" is enabled by your Tableau Cloud site admin or Tableau Server admin, you can see metadata on external assets for the content that you own, or for the content that is published to a project that you are a project leader or project owner of. **Ad-hoc:** You can see metadata on external assets that you have been granted explicit **View** permissions to.
	Edit assets and their metadata	You can edit metadata on an external asset that you have been granted explicit "write" or **Overwrite** permissions to(if not automatically granted because the user is a flow owner).
	Change permissions on assets and their metadata	You can change permissions on an external asset that you have been granted explicit "edit" or **Set Permissions** to ((if not automatically granted

Data Management	Capability	Requirements
		because the user is a flow owner).
	Grant other users permissions to see assets and their metadata	You can change permissions on an external asset that you have been granted explicit "edit" or **Set Permissions** to ((if not automatically granted because the user is a flow owner).
Not licensed	See assets and their metadata	**Applies to Metadata API only:** If “derived permissions” is enabled by your Tableau Cloud site admin or Tableau Server admin, you can see metadata on external assets for the content that you own, or for the content that is published to a project that you area project leader or project owner of.
	Edit assets and their metadata	Requires Data Management
	Change permissions on assets and their metadata	
	Grant other users permissions to see assets and their metadata	

Manage Data

You can connect to and manage the data you that you use in Tableau.

Tableau Server Data Sources

When your Tableau users want to share data connections they've defined, they can publish data sources to Tableau Server. When a data source is published to the server, other users can connect to it from their own workbooks, as they do other types of data. When the data in the Tableau data source is updated, all workbooks that connect to it pick up the changes.

> Looking for Tableau Server on Windows? See Tableau Server Data Sources.

A Tableau Server data source consists of metadata that describes the following:

- **The connection information**: Defines whether the data is in a live database or an extract, and which of that data to bring in to Tableau.
- **Customization and cleanup**: Includes information that facilitates efficient use of the data. For example, calculations, sets, groups, bins, parameters, custom field formatting, and so on.
- **Data access and refresh instructions**: Includes the location of the underlying database server (whether on-premises or in the cloud), network paths for file-based data, security information such as credentials or access tokens, and related information.

In addition to helping your users create data consistency and reliability, using Tableau data sources offers advantages to you as the administrator. Because multiple workbooks can connect to one data source, you can minimize data source proliferation and save on storage space and processing time. When someone downloads a workbook that connects to a Tableau data source that in turn has an extract connection, the extract stays on the server, reducing network traffic. Finally, if a connection requires a database driver, you need to install

and maintain the driver only on the server, instead of on each user's computer. If you use Tableau Cloud, all supported drivers are available to data sources published to your site.

Managing data sources

You can perform some or all management tasks on a data source if you have one of the following levels of access:

- Site or server administrator
- Project leader or owner of the project the data source is published to

  Full project leader access is available only with some site roles. For information, see Project-level administration.

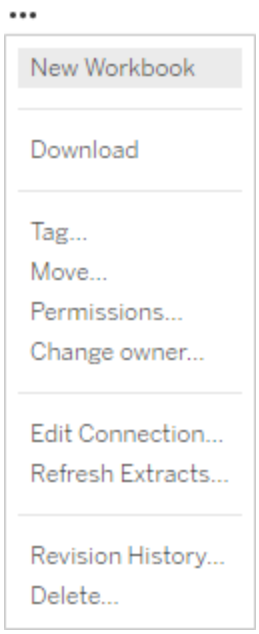
- Data source owner

Unless you're a site or server administrator, you might not have access to all management tasks. For example, if the Permissions menu is not available, chances are that the project permissions are locked, which denies setting permissions on individual workbooks and data sources.

> **Tip:** A best practice is to designate a person or team to manage all data sources published to a project or to the whole site, so that all data sources can be maintained under the same set of guidelines.

To perform the management tasks that you have access to, do the following:

1. Sign in to the site, and on the **Content** tab, select **Explore** > **Data sources**.
2. On a data source, select the **Actions** (**…**) menu.



- **New workbook or Download:** Create a new workbook in the browser environment that connects to this data source. Or download the data source to use locally.
- **Tag:** Add or remove keyword tags. Tags can contain a single word or multiple words, delimited by a comma.
- **Move:** Move a data source from one project to another. This requires specific settings on each project. For information, see Requirements for moving assets.
- **Permissions:** View or update permissions that specify which users or groups can connect to, modify, or download the data source. As mentioned at the beginning of this section, if this action is not available, the project permissions might be locked, and only the project leader or administrator can change permissions.
- **Change owner:** Making someone an owner gives them complete access to it.
- **Refresh extracts:**If a data source includes an extract, you can assign the extract to a refresh schedule.

  For information, see Refresh Data on a Schedule.

- **View the data source's revision history**
- **Delete:** Deleting a data source affects workbooks that connect to the data source. Before you delete a data source, ensure that there are no workbooks that connect to the data source or edit the workbooks to use another data source.

In addition, for data sources that are proxy connections, administrators can stay aware of how users authenticate to the database, and whether the appropriate drivers are installed. For information, see Database Drivers and Data Security.

Restrictions

Published data sources often function as curated and trustworthy data sources. As such, there are restrictions on how they can be modified and used.

Aliases and calculations can't be edited.

- New aliases also can't be created. Fields can be duplicated and the copy can be aliased.
- New calculations can be created. Existing calculation can also be copied and the copy can be edited.

Relationships and joins can't be edited.

Published data sources cannot be used in joins or relationships.

- Use blends if you need to combine published data sources.

Extract Upgrade to .hyper Format

In Tableau version 2018, we introduced the `.hyper` format for Tableau extracts to replace the old `.tde` format. The `.hyper` format has been the standard format used by Tableau to create extracts since 2018, and the large majority of extracts are now `.hyper` files. Beginning in early 2023, Tableau discontinued support for the `.tde` format on Tableau Cloud and Tableau Public. For more details about this deprecation, see this Tableau Community post.

Discontinuation of support for .tde files

Beginning in 2023, the `.tde` format for Tableau extracts was deprecated. This format was replaced by the `.hyper` format in 2018 but continued to be valid for uploaded files until March 2023.

- This change took place for Tableau Cloud and Tableau Public as of March 2023.
- Beginning with version 2023.1.0 of Tableau Server, the uploading of `.tde` format files is disabled.
- 2024.2 is the last version of Tableau Desktop that supports any .tde-based workbooks, data sources, or bookmarks. Versions 2024.3 and beyond only support .hyper format.

Manually upgrade your .tde extract using Tableau Desktop

If you manage extracts locally, you can manually upgrade your `.tde` extract to a `.hyper` extract using Tableau Desktop.

> **Note**: This functionality is only available in Tableau Desktop versions 2024.2 and older.

1. In Tableau Desktop, open a workbook that uses a `.tde` extract.
2. Select the extract data source from the **Data** menu and then select **Extract** > **Upgrade**.
3. Select **File** > **Save**, which saves the workbook and also completes the extract upgrade.

Manually upgrade your .tde with a live connection

If your .tde file uses a live connection (as opposed to an extract), you need to upgrade the file by following the instructions in this Tableau Community post about updating to .hyper files. It is not possible to update .tde files with live connections using Tableau Desktop.

Set the Site Time Zone for Extracts

The default time zone for extract-based data sources in a site is Coordinated Universal Time (UTC). Server administrators can set a different time zone.

To set the site time zone for extracts:

1. Sign in to Tableau as an administrator.
2. On the site you want to configure, click `Settings`.
3. In the `Site Time Zone for Extracts` section, select a time zone and then click `Save`.

In calculated fields, functions such as NOW() or TODAY() look at the time zone. For more information about Extracts, see Extract Your Data in the Tableau Desktop help.

The timezone setting, in addition to being used for extract-based data sources, also affects internal extracts. For example, when Tableau connects to file-based data sources like text files, an extract is automatically created internally. Similarly, this happens where Tableau uses an internal extract to integrate data from different sources.

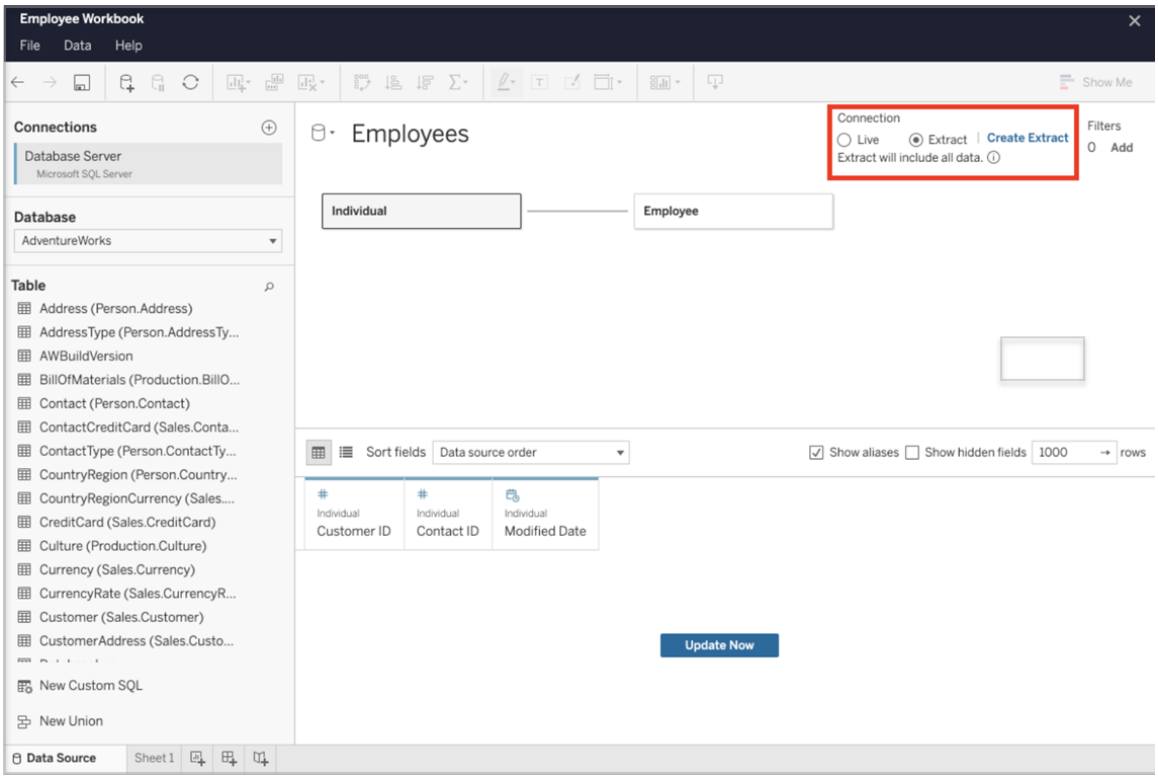
Create Extracts on the Web

You can extract your data sources in the web (without using Tableau Desktop) to improve data source performance and support additional analytical functions. When you extract your data source, Tableau will copy the data from your remote data store to Tableau Server or Tableau Cloud. To learn more about the benefits of extracting your data, see Extract Your Data. In the web, you can extract while in Web Authoring or while in Content Server.

Create extracts in Web Authoring

You can create extracts directly in web authoring with default extract settings.

Extract an Embedded Data Source in Web Authoring



Complete the following steps to create an extract in web authoring.

Tip: It's recommended to finalize your data model before you create the extract. Extract creation may take a long time and any changes to your data model, such as adding new logical tables, will invalidate the extract.

1. Select the **Data Source** tab in the bottom left corner of the web authoring pane. For new workbooks, you will start in the **Data Source** tab.
2. In the top-right corner, change the connection type from **Live** to **Extract**.
3. Select **Create Extract**. You will see the **Creating Extract** dialog box.

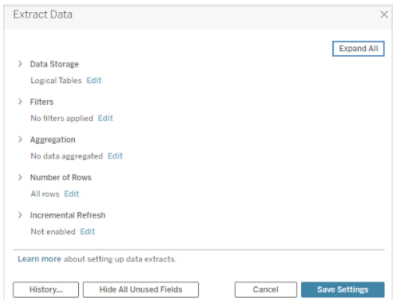
Extract creation might take a long time and you can close your authoring session while the extract is being created. To ensure your extract creation isn't lost, in the dialog box, choose **Notify Me When Complete** to specify a location for the extracted workbook to be saved. If

your extract succeeds, your workbook will be saved to the specified location and you will be notified that you can continue your web authoring session.

If your extract creation fails, you will be notified that the extract couldn't be created and you can restore your unsaved changes by reopening the original workbook in web authoring.

Define your Extract Settings

You can configure one or more of the following options to tell Tableau how to store, define filters for, and limit the amount of data in your extract.



Decide how the extract data should be stored

You can choose to have Tableau store the data in your extract using one of two structures (schemas): logical tables (denormalized schema) or physical tables (normalized schema). For more information about logical and physical tables, see The Tableau Data Model. The option you choose depends on what you need.

Logical Tables

Logical Tables store data using one extract table for each logical table in the data source. Physical tables that define a logical table are merged and stored with that logical table. For example, if a data source was made of a single logical table, the data would be stored in a single table.

If a data source was made of three logical tables (each containing multiple physical tables), the extract data would be stored in three tables—one for each logical table.

Select **Logical Tables** when you want to limit the amount of data in your extract with additional extract properties like extract filters, aggregation, Top N, or other features that require denormalized data. Also use when your data uses pass-through functions (RAWSQL). This is the default structure Tableau uses to store extract data.

If you use this option when your extract contains joins, the joins are applied when the extract is created.

Physical Tables

Physical Tables stores data using one extract table for each physical table in the data source.

Select **Physical Tables** if your extract is comprised of tables combined with one or more equality joins and meets the conditions for using the Physical Tables option listed below. If you use this option, joins are performed at query time.

This option can potentially improve performance and help reduce the size of the extract file. For more information about how Tableau recommends you use the Physical Tables option, see Tips for using the Physical Tables option in the Tableau Desktop help.

In some cases, you can also use this option as a workaround for row-level security. For more information about row-level security using Tableau, see Restrict Access at the Data Row Level in the Tableau Desktop help.

Conditions for using the Physical Tables option

To store your extract using the Physical Tables option, the data in your extract must meet all of the conditions listed below.

- All joins between physical tables are equality (=) joins.
- Data types of the columns used for relationships or joins are identical.
- No pass-through functions (RAWSQL) used.
- No incremental refresh configured.
- No extract filters configured.
- No Top N or sampling configured.
- When the extract is stored as physical tables, you can't append data to it.

- For logical tables, you can't append data to extracts that have more than one logical table.

Note: Both the Logical Tables and Physical Tables options only affect how the data in your extract is stored. The options don't affect how tables in your extract are displayed on the Data Source page.

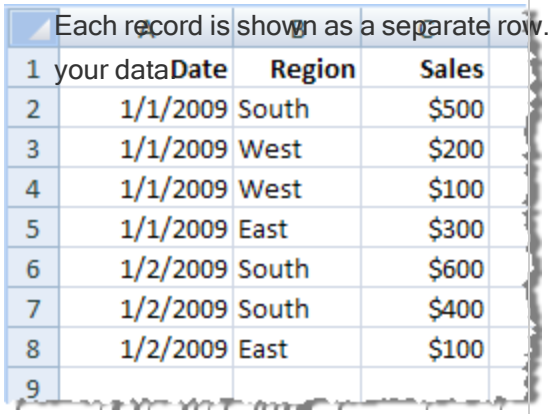
Determine how much data to extract

Select **Add** to define one or more filters to limit how much data gets extracted based on fields and their values.

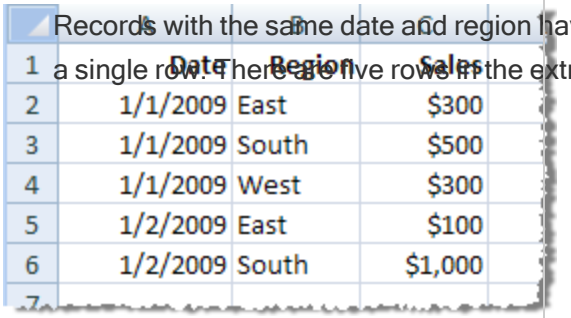
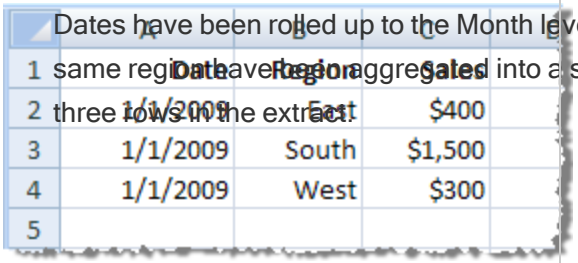
Aggregate the data in the extract

Select **Aggregate data for visible dimensions** to aggregate the measures using their default aggregation. Aggregating the data consolidates rows, can minimize the size of the extract file, and increase performance.

When you choose to aggregate the data, you can also select **Roll up dates** to a specified date level such as Year, Month, etc. The following examples show how the data will be extracted for each aggregation option you can choose.

Original data	Each record is shown as a separate row. There are seven rows in your data 	

	Date	Region	Sales
1			
2	1/1/2009	South	$500
3	1/1/2009	West	$200
4	1/1/2009	West	$100
5	1/1/2009	East	$300
6	1/2/2009	South	$600
7	1/2/2009	South	$400
8	1/2/2009	East	$100
9			

Aggregate data for visible dimensions *(no roll up)*	 Records with the same date and region have been aggregated into a single row. There are five rows in the extract.	
Aggregate data for visible dimensions *(roll up dates to Month)*	 Dates have been rolled up to the Month level and records with the same region have been aggregated into a single row. There are three rows in the extract.	

Choose the rows to extract

Select the number of rows you want to extract.

You can extract All rows or the Top N rows. Tableau first applies any filters and aggregation and then extracts the number of rows from the filtered and aggregated results. The number of rows options depend on the type of data source you are extracting from.

Notes:

- Not all data sources support sampling. So, you might not see the Sampling option in the Extract Data dialog box.
- Any fields that you hide first in the Data Source page or on the sheet tab will be excluded from the extract.

Configure Incremental Refresh Settings

Most data sources support an incremental refresh. Rather than refreshing the entire extract, you can configure a refresh to add only the rows that are new since the previous time you extracted the data.

For example, you may have a data source that is updated daily with new sales transactions. Rather than rebuild the entire extract each day, you can just add the new transactions that occurred that day. To have incremental as an option when you schedule a refresh you must first define the settings.

Periodically you might want to do a full refresh to ensure you have the most up to date data.

> **Note:** If the data structure of the source data changes (for example, a new column is added), you will need to do a full extract refresh before you can start doing incremental refreshes again.

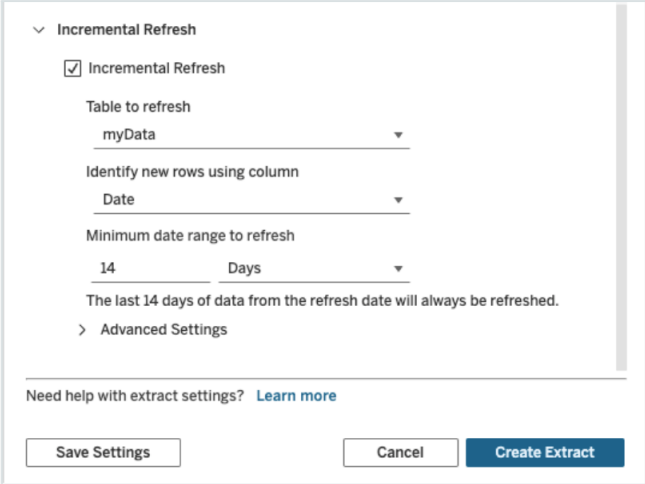
Set up Incremental Refresh

Follow the steps below to set up an extract to be refreshed incrementally but first ensure you've selected All rows as the number of Rows to extract.

Incremental refresh can only be defined when you are extracting all rows in the database. You can't increment a sample extract.

1. Select **Incremental refresh** box.
2. Choose a table to refresh.
3. Select a column to use for identifying new rows.
4. Enter a subdate range in **Minimum date range to refresh**. You have the option to choose a specific time range in days, hours, minutes, or seconds within this field.

For example, when selecting a Date or Datetime column in Tableau, users can refresh extract data within a specified timeframe, such as 14 days from the refresh date. This feature is beneficial for data sources that allow inserts and retroactive modifications within a defined time period. By using incremental extract refresh, users can capture these changes along with any new data efficiently.



Use Advanced Settings

You can expand **Advanced Settings** to establish how new rows are retrieved.

> **Note:** If you have set a Minimum date range for refreshing, the Advanced Settings feature won't be accessible.

Advanced Settings allow you to either replace the last rows added by refreshing values equal to or greater than the last recorded value, or retain the last rows added by only refreshing the extract with values greater than the last recorded value.

In the first approach, Tableau allows users to incrementally refresh extracts with a non-unique key column such as date, datetime, or ID.

This method adds a new step when performing an incremental refresh. Tableau will first delete rows in the extract that are equal to the previous highest value seen. Tableau then queries for all rows that are higher than or equal to the previous highest value which will pick up all the deleted rows and any new ones.

Conversely, you can still opt to not replace the last rows added and only add rows with values greater than the last recorded value.

To finish, select **Create Extract**.

Considerations when doing a Incremental refresh

Editing an extract:

If you're editing an existing extract, the last refresh is shown so you can be sure you are updating the extract with the correct data.

Full Refresh:

A Full Refresh replaces all of the rows with the data in the original data source every time you refresh the extract. A Full Refresh can take longer and be expensive on the database.

Data Engine:

The data engine, which is the underlying mechanism that Tableau uses to create extracts, stores time values with a precision of up to 3 decimal places.

If you specify a datetime or timestamp column for Identify new rows using column, and your database uses a higher precision than Tableau, you can end up with duplicate rows after an incremental refresh.

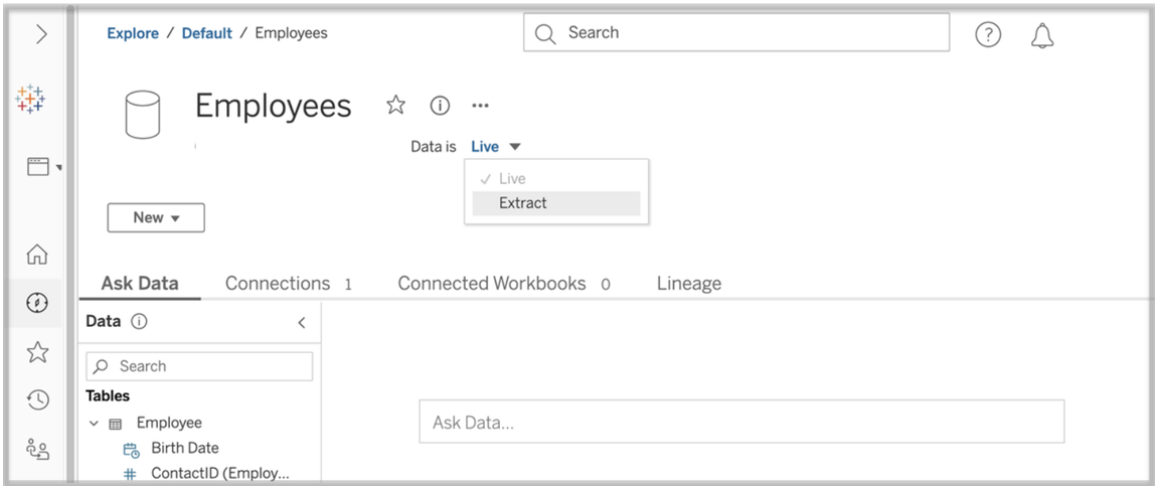
For example, if the database has two rows, one with a datetime value of 2015-03-13 17:30:56.502352 and one with a datetime value of 2015-03-13 17:30:56.502852, Tableau will store both rows using a datetime value of 2015-03-13 17:30:56.502 thereby creating duplicate rows.

Limitations

- You can't create extracts for embedded data sources that reference published data sources. As a workaround, create the extract directly on the published data source. For more information, see Extract a Published Data Source on Content Server.
- You can't create extracts for file-based data sources. File-based data sources already have special performance features and adding extraction will have no performance benefit.
- This feature doesn't apply to bridge-based data sources in Tableau Cloud.
- Custom SQL Limitation: Custom SQL queries aren't supported with Advanced Settings. Users relying on custom SQL will need to adjust their approach if they wish to use the Advanced Settings for incremental refresh.

Create extracts in Content Server

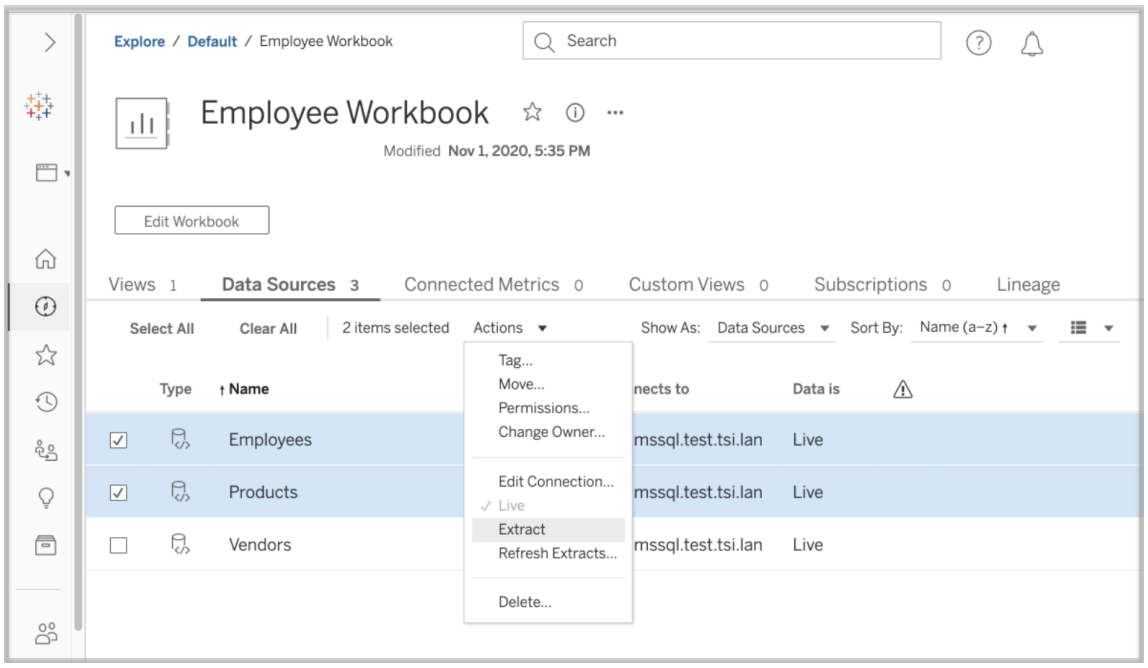
Extract a Published Data Source on Content Server



Complete the following steps to extract a published data source.

1. Sign in as an administrator or as the owner of the data source.
2. On the Content tab, select **Explore,** and then select **Data sources**.
3. Select a data source by clicking on the Data Source name.
4. At the top of the screen, under the Data Source name, select the drop-down menu that says **Live**.
5. Change the connection type from **Live** to **Extract**. If the extract encryption at rest feature is enabled on the site, select either **Encrypted** or **Unencrypted**.
6. If you see an error message about embedded credentials, embed your credentials in the data source. To do this, choose **Edit Connection**. Select "Embedded password in connection" and then choose **Save**.

Extract an Embedded Data Source on Content Server



Complete the following steps to extract one or more data sources that are embedded in a published workbook.

1. Sign in as an administrator or as the owner of the data source.
2. Navigate to the published workbook.
3. Navigate to the Data Sources tab
4. Select one or more of the data sources.
5. Choose the **Action** button.
6. Select **Extract**. If the extract encryption at rest feature is enabled on the site, select either **Encrypted** or **Unencrypted**.

Limitations

- Your connection credentials must be embedded in the data source.
- You can't create extracts for embedded data sources that reference published data sources. As a workaround, create the extract directly on the published data source.

- You can't create extracts for file-based data sources. File-based data sources already have special performance features and adding extraction will have no performance benefit.
- This feature doesn't apply to bridge-based data sources in Tableau Cloud.

Keep Extracted Data Fresh

After data is extracted, you can optionally set up an extract refresh schedule to keep the data fresh. For more information, see Refresh Data on a Schedule.

Monitor and Manage Extracts

Server administrators can monitor extract creation on the **Background Tasks for Extracts** admin view. For more information, see Background Tasks for Extracts.

Server administrators can manage extracts on the Jobs page. For more information, see Managing Background Jobs in Tableau Server.

Extract creation jobs, like extract refresh jobs, have a maximum query limit before they timeout. This is to prevent jobs from running forever and using an unbounded amount of server resources. The extract query limit timeout can be configured by server admins using the TSM. command line interface configuration setting `backgrounder.querylimit`For more information, see tsm configuration set Options.

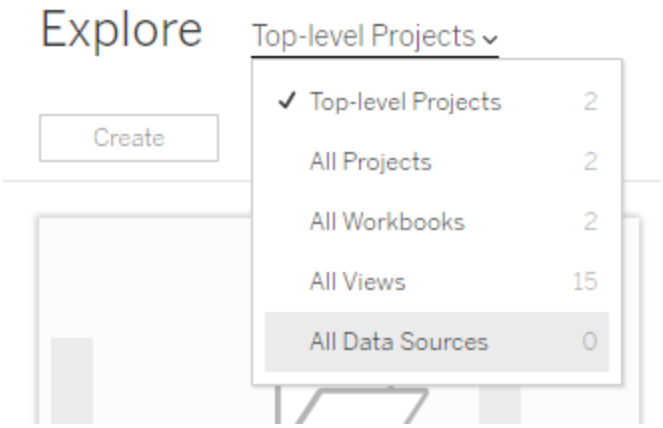
Server administrators can manage web authoring. For more information, see Set a Site's Web Authoring Access and Functions.

View Data Source Attributes

In the **Content** area of the Tableau Server web authoring environment, you can filter the view to show only data sources or connections and their attributes.

View data sources by name

To filter by data source name, under **Explore**, select **Data sources**.

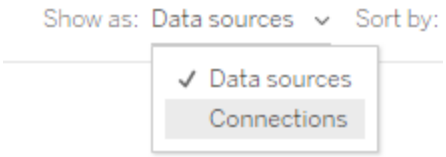


Attributes in the data source name list include the following:

- **Type**—The icon next to the data source name indicates whether the data source is published as a live connection ( ), an extract connection ( ), or is embedded in a workbook ( ).
    - Select the name of a published data source to open its data source page, with tabs for viewing connections and workbooks that connect to it.
    - Select the name of an embedded data source to open the workbook associated with it, as well as tabs for viewing other data it connects to.
- **Connects To**—Indicates the name of the database server or data file. This could be a database outside of Tableau Server, an extract, or a published data source.
- **Live or Last Extract**—This column tells you whether the connection to the data is live, or, if it is a connection to an extract, when the extract was last updated.

View a list of connections

To filter by connection type, in the **View** list, select **Connections**.



Connection attributes include:

- **Connects to**—Indicates the name of the database server or data file. This could be a database outside of Tableau Server, an extract, or a published data source.
- **Connection type**—Shows the type of data. **Tableau Server** indicates that the connection is to a data source published on the site. **Tableau Data Engine** means the data source has an extract stored on the Tableau data server.

Keep Data Fresh

The topics in this section describe how to manually refresh data, as well as schedule data refreshes.

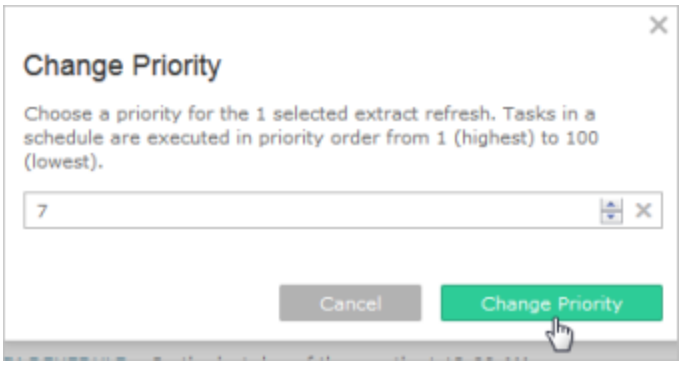
Manage Refresh Tasks

Administrators can change the priority of scheduled extract refreshes relative to other server tasks, manually refresh extracts, or delete their schedules.

1. Sign in to the site that has the schedules you want to manage, and then click **Tasks**.
2. Select one or more scheduled extract refreshes.
3. From the **Actions** menu, do any of the following:
    - Select **Change Schedule**, and choose a new schedule from the list.
    - Select **Run Now** to refresh manually.

Note: If an extract does not have a scheduled refresh, you can refresh it on demand from the Data Connections page.

- Select **Change Priority**, and enter a number between 1 and 100 to move the extract up or down in the priority list.



- Select **Delete** to completely remove the schedule for the selected data sources.

See also

Enable Extract Refresh Scheduling and Failure Notification

Refresh Data on a Schedule

You can schedule refresh tasks for published extract data sources and published workbooks that connect to extracts. New schedules can be created by Tableau Server Administrators on the **Schedules** page. For more information, see Create or Modify a Schedule.

For information on how to refresh flow outputs, see Schedule Flow Tasks

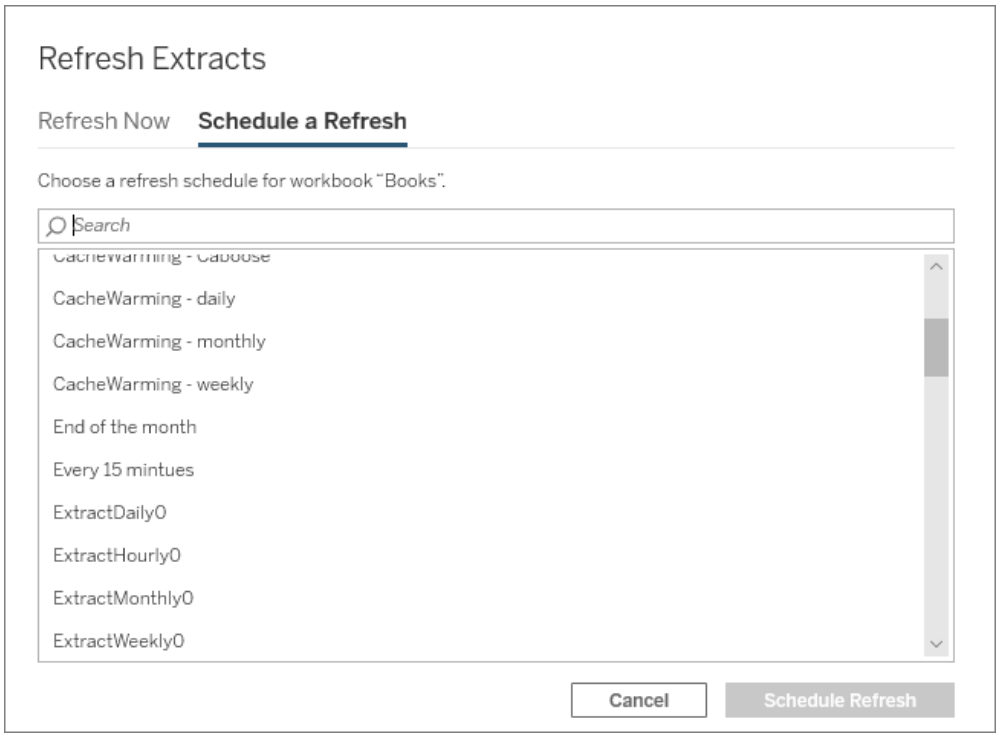
1. When you're signed in to Tableau Server, select **Explore** from the left navigation pane, and then, depending on the type of content you want to refresh, select **All Workbooks** or **All Data Sources** from the drop-down menu.
2. Select the check box for the workbook or data source you want to refresh, and then select **Actions** > **Refresh Extracts**.

3. In the Refresh Extracts dialog, select **Schedule a Refresh**, and complete the following steps:

    - Select the schedule you want.
    - If available, specify whether you want a full or incremental refresh.

      A full refresh is performed by default. Incremental refresh is available only if you configured for it in Tableau Desktop or web authoring in Tableau Server before publishing the extract. For more information, see Refreshing Extracts in the Tableau Help.

    - Click **Schedule Refresh**.



Quick Start: Refresh Extracts on a Schedule

For published workbooks that connect to data extracts, you can set up the server to refresh the extracts on a recurring schedule, so all workbooks connected to them always show the most up-to-date data.

To schedule refreshes you need to have administrator or data owner permissions.

> **Note:** This topic applies to extracts published to Tableau Server. For Tableau Cloud, how you refresh extracts depends on the underlying data they connect to. For more information, see Keep Data Fresh.

1 Set up a schedule on the server

Sign in to the server, go to the **Schedules** page, and click **New Schedule**.

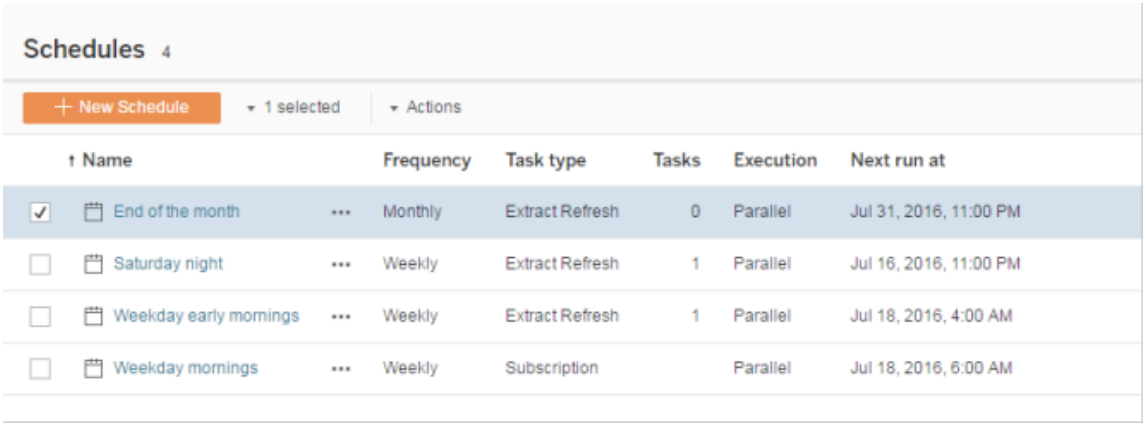


Tableau provides a few refresh schedules. You create additional schedules you need.

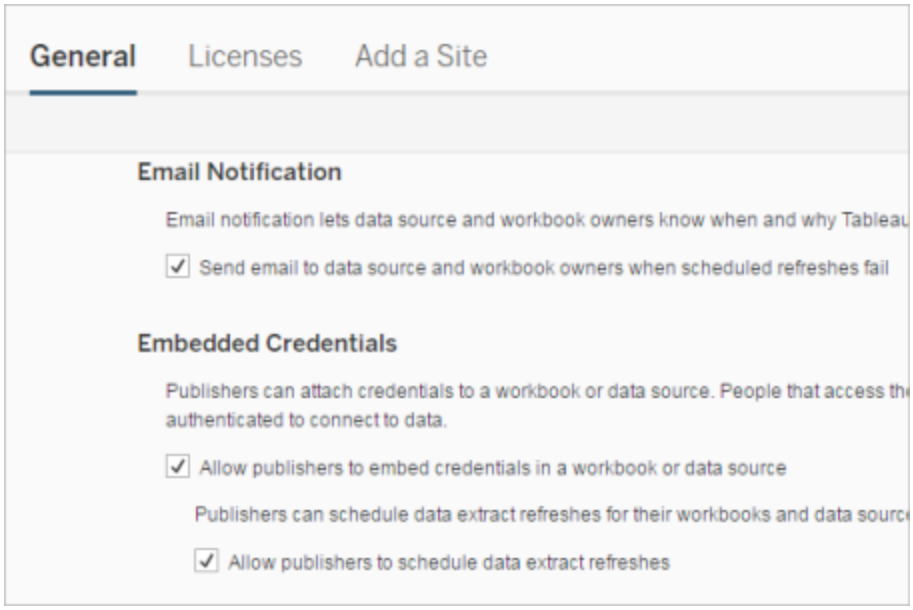
2 Enable scheduled extract refreshes and failure emails

As a server or site administrator, you can enable schedules, as well as email notification when extract refreshes fail.

Select **Settings**, and then go to the **General** page.

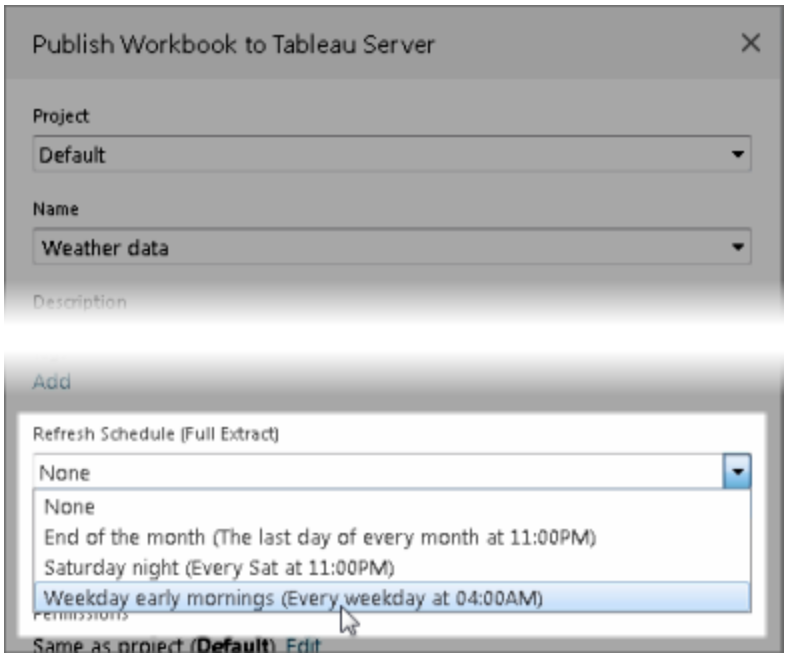
- Under Email Notification, select **Send email to data source and workbook owners when scheduled refreshes fail**.
- Under **Embedded Credentials**, select both check boxes to allow publishers to embed

credentials and schedule extract refreshes.



3 Publish a workbook with an extract

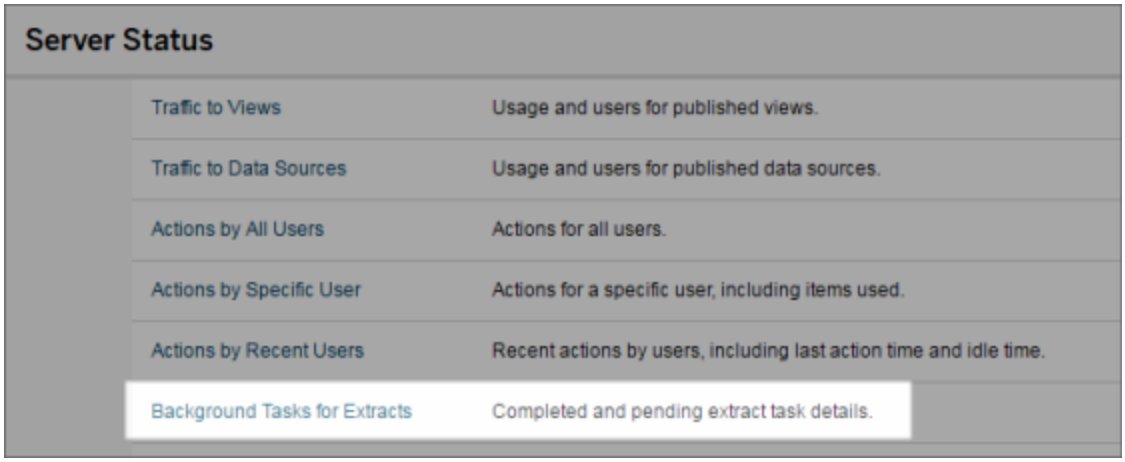
In Tableau Desktop, select **Server** > **Publish Workbook**. Sign in to the server if you're not already. In the **Publish Workbook to Tableau Server** dialog box , click **Schedules & Authentication**. Under **Extract Schedule**, select the schedule from the list.



If the original data requires authentication, you will also need to select how you want people to access it.

4 Monitor refresh performance

You can monitor scheduled tasks by viewing **Background Tasks for Extracts** on the **Status** page.



Automate Refresh Tasks

You can associate extract refresh tasks with schedules in Tableau Server to automate refreshing extracts. You can also automate extract refreshes using tabcmd, a command line utility that you can download for use with Tableau Server. In particular, you can use the `refreshextracts` command in combination with other commands in your own script. For example:

```
tabcmd login - http://mytabserver -u jsmith -p P@ssw0rd! refreshextracts --datasource salesq4
```

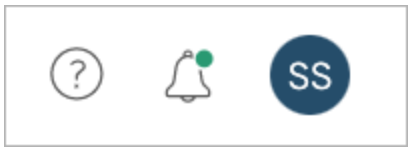
Note: When an extract refresh is performed on extracts created in Tableau 10.4 and earlier (that is, a .tde extract), the extract is upgraded to .hyper extract automatically. While there are many benefits of upgrading to a .hyper extract, your users won't be able open the extract with earlier versions of Tableau Desktop. For more information, see Extract Upgrade to .hyper Format.

For information about downloading the tabcmd utility, see tabcmd.

Handle Extract Refresh Alerts

When Tableau Server cannot complete a scheduled refresh, an alert appears to indicate that the refresh has failed. If a scheduled refresh fails five consecutive times, Tableau Server suspends the refresh. When a refresh is suspended, Tableau Server does not try to run it again until someone takes an action that attempts to correct the cause of the failure.

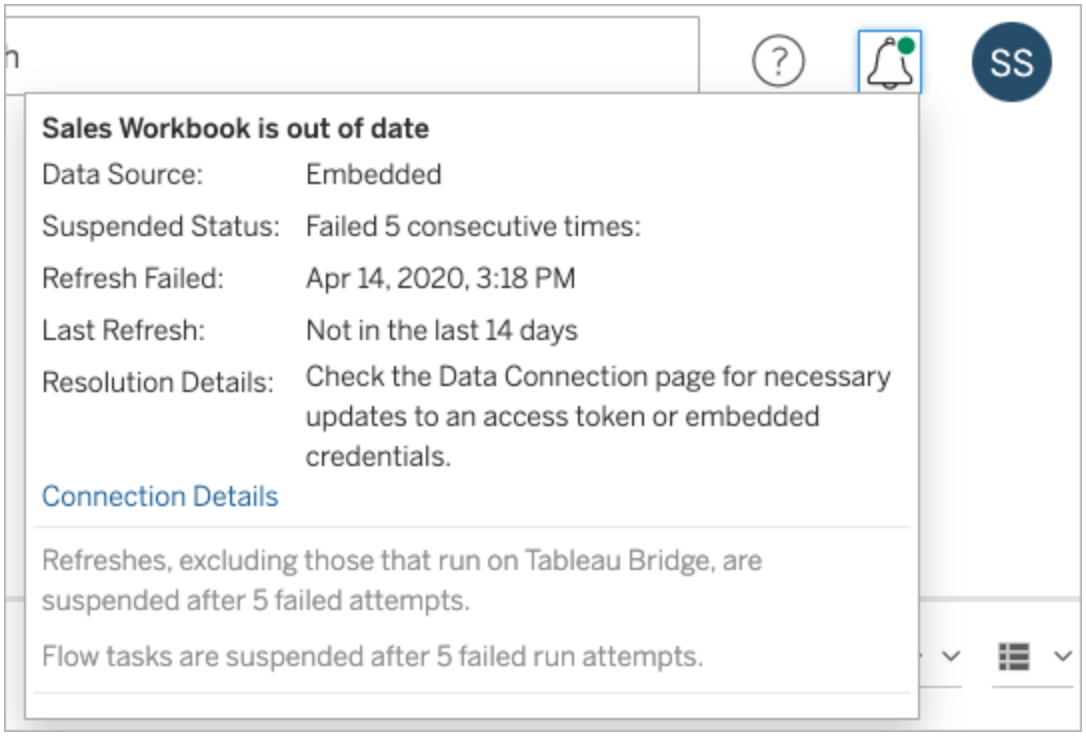
Note: The number of consecutive failures for a refresh is set to five by default, but can be changed by a Tableau Server administrator, using the `backgrounder.failure_threshold_for_run_prevention` opti0n. For more information, see tsm configuration set Options.



You will see the Alerts menu only if an extract refresh failed and you are:

- A system or site administrator
- The author of the workbook or data source that couldn't be refreshed
- The author of a workbook that connects to a data source that couldn't be refreshed

When you open the Alerts menu you can see more information about the refresh failure(s):



When a **Data source** is listed as **Embedded** it means that the data source definition (which includes things like the data source credentials or the database name) is embedded, or resides, within the workbook itself, originally created in Tableau Desktop.

When a data source name or workbook name is listed as the **Data source** (for example, **Data source**: **sales_data**), it means that the data source is a Tableau Server data source. The data source definition resides on Tableau Server.

In the Data pane on Tableau Desktop, you can determine whether the data source is on Tableau Server or is local. If the data source is on the server, a Tableau icon is displayed next to the data source name instead of a database icon :

Resolving Extract Refresh Problems

To resolve refresh issues, you can take any of these actions, based on the cause indicated in the alert:

- **Errors related to access token validation or user credentials**

  You can resolve some extract refresh problems by clicking the **Connection Details** in the alert. Select the check box next to the problematic data source, click **Actions** > **Edit Connection**, and then enter the missing information. Click **Save** when you're done. After you update the connection information, Tableau Server restarts the refresh schedule.

  If you originally embedded the credentials or other data connection information when you published the workbook or data source from Tableau Desktop, you can also republish the workbook or data source. As part of the publishing process, you can choose to set a new refresh schedule. If you don't choose a new schedule, Tableau Server restarts the existing schedule.

- **Errors that indicate the database was unreachable**

Confirm that the database is online and that you can sign in to access the data. You can use the **Try again** link in the alert to restart the refresh schedule.

- **Errors when using user filters or impersonation**

  See the Tableau Knowledge Base.

If the problem cannot be corrected by editing the data connection, you will need to resolve it in Tableau Desktop and republish the workbook.

Tip: Administrators can edit data connections at any time on the **Data Connections** page, accessible from each site by clicking the **Content** tab and Data Connections

Automatically Suspend Extract Refreshes for Inactive Workbooks and Data Sources

To save resources you can automatically suspend extract refresh tasks for inactive workbooks and published data sources. This feature applies to full extract refreshes that occur more frequently than once a week. Incremental refreshes and those that occur less frequently than weekly are not impacted.

Note: Support for automatic suspension of extract refreshes for data sources is available beginning in Tableau Server version 2023.3.

Note: Support for automatic suspension of extract refreshes for data sources is available beginning in the Tableau Cloud July 2023 release.

For a workbook, if any of the following events occur, the workbook's inactivity countdown timer is reset:

- Viewing the workbook sheets
- Having any data-driven alert or subscription set-up on the workbook
- Downloading the workbook
- Moving the workbook's location or changing the owner

For a published data source, any event which fetches the data from the data source will cause its inactivity countdown timer to be reset. These include:

- Loading a workbook view that is connected to the data source
- Visiting the data source's Ask Data page
- Tableau Desktop connecting to the data source

Configure the feature

1. Sign in to Tableau Server as a server administrator.
2. Go to the General tab of the Settings page for the site:
   - If you have a single site, at the top the browser window, click **Settings** and **General**.
   - If you have multiple sites, select the site you want to configure and click **Settings** and **General**.
3. On the **General** page, do the following:
   - Under **Automatically Suspend Extract Refresh Tasks**, select the **Automatically suspend extract refresh tasks for inactive workbooks and data-sources** check box.
   - Specify the number of days, from 7 through 100, that a workbook should be inactive before extract refresh tasks are suspended. The default is 32 days.
   - Click **Save**.

Notifications

An email notification is sent three days before the extract refresh schedule is suspended.

Another email notification is sent when the extract refresh schedule is suspended.

Resume suspended extract refreshes

Suspended extract refreshes won't automatically resume if someone uses the workbook. It must be done manually by a server or site administrator.

To view and resume extract refreshes that were suspended:

1. Sign in to the site as an administrator and click **Tasks**.
2. Click the **Extract Refreshes** tab.

3. Select one or more items.
4. From the **Actions** menu, select **Resume**.

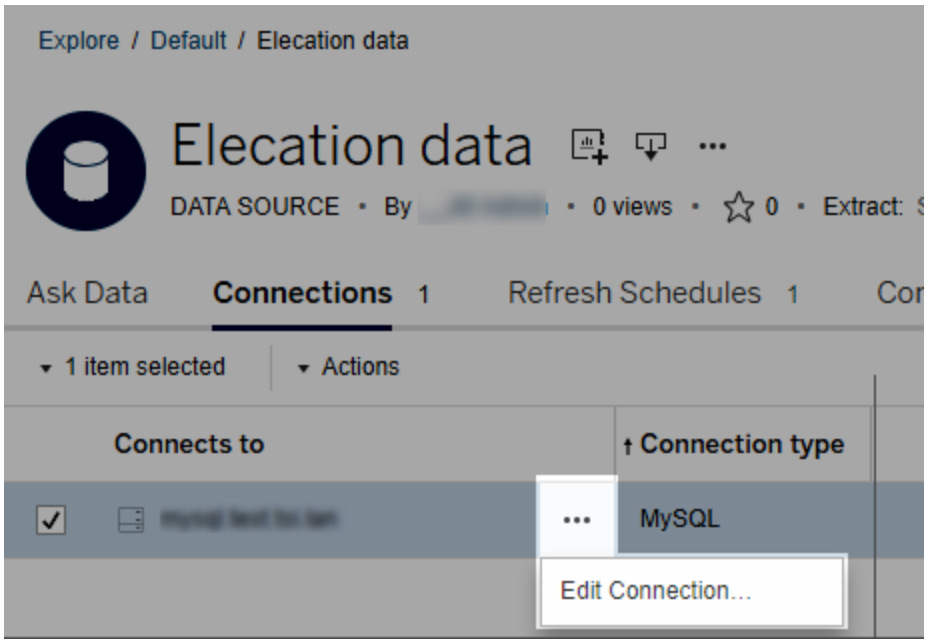
Edit Connections on Tableau Server

Administrators and data source owners can manage the information that describes how a published data source connects to the underlying data. This information includes the database server name or address, the server port, the database user name, and whether or not the password is embedded in the connection.

Note: Ability to edit connections is determined primarily by your site role, rather than by your permissions on the data source. To edit connections, your site role must be **Server Administrator**, **Site Administrator**, or **Creator**. If your site role is **Creator**, you also must be the data source owner.

1. Sign in to the site that has the data sources you want to modify, and on the **Content** tab, select **Explore** > **Data sources**.
2. Select the name of the data source with the connection you want to update.

   Display filters to search for the data source or narrow the scope of the data source list. The values you type into the **Server name** and **Database username** fields are treated as regular expressions.
3. In the **Connections** view, select the **Actions** (**…**) menu for the data source, and then select **Edit Connection**.

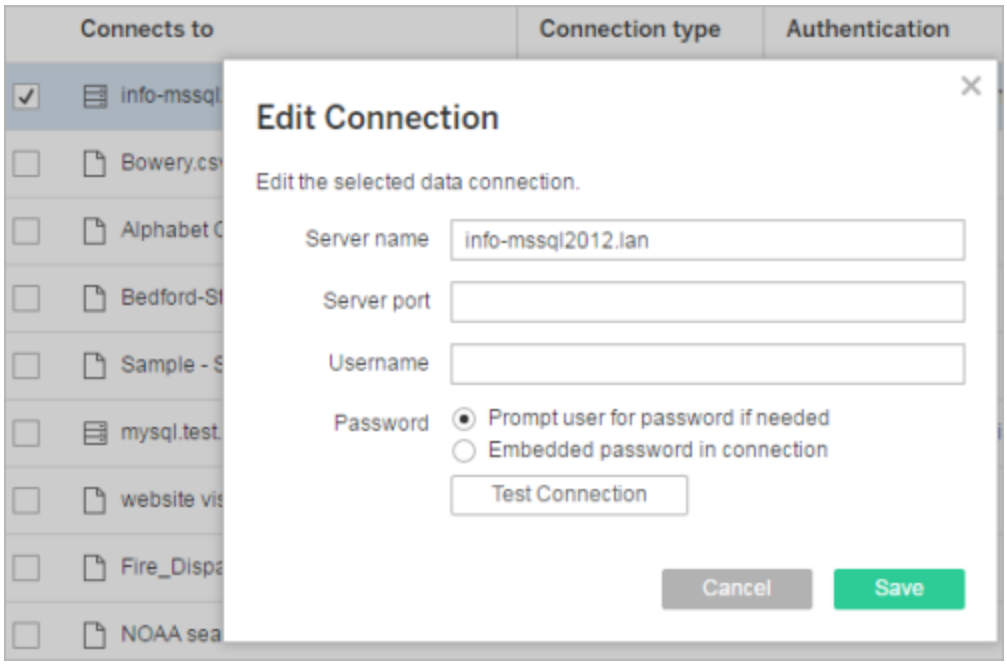


To edit multiple data sources, select the data sources you want to edit, then click the **Actions** menu and **Edit Connection**.

4. Update the connection information.

   For **Server name**, if you want to use an IP address, make sure the database or its driver supports that type of connection. If it doesn't, enter the database server name.

   See also Authentication types for Google, Salesforce, and WDC data later in this topic.



Note: The **Test Connection** button is disabled in versions 2021.4 and later when **Prompt user for password if needed** is selected.

5. Click **Save**.

6. Refresh the browser page for your changes to take effect.

Authentication types for Google, Salesforce, and WDC data

Google BigQuery, Google Analytics, Salesforce.com, and many web data connector (WDC) connections use the OAuth authentication standard, which uses secure access tokens instead of “raw” user name and password credentials. With OAuth, database credentials do not need to be stored in Tableau, and all users connect through this access token, including Tableau Desktop users who want to create or edit workbooks that connect to the data source.

The following sections describe Google and Salesforce connection options. Web data connector options vary, but all involve signing in through the provider’s web-based sign-in form to establish the access token.

Google authentication options

When you edit Google BigQuery or Google Analytics connections, select either of the following options in the **Edit Connection** dialog box:

- Select **Embed Google BigQuery (or Google Analytics) credentials in the connection** to authenticate through a designated account, and then select an existing account from the list or select **authenticate account now...** to add a new one.

  When you add a new account, the Google sign-in page appears. After you provide your database credentials, Google prompts you to confirm Tableau access to the data. When you click **Accept**, Google returns an access token to use for connecting to the data.

  > **Note**: If you create extracts of your Google data source, select this first option, so that you can schedule refresh tasks.

- Select **Prompt user for Google BigQuery/Analytics credentials** to require users to connect through their own individual access tokens or sign in each time they connect.

Salesforce.com authentication options

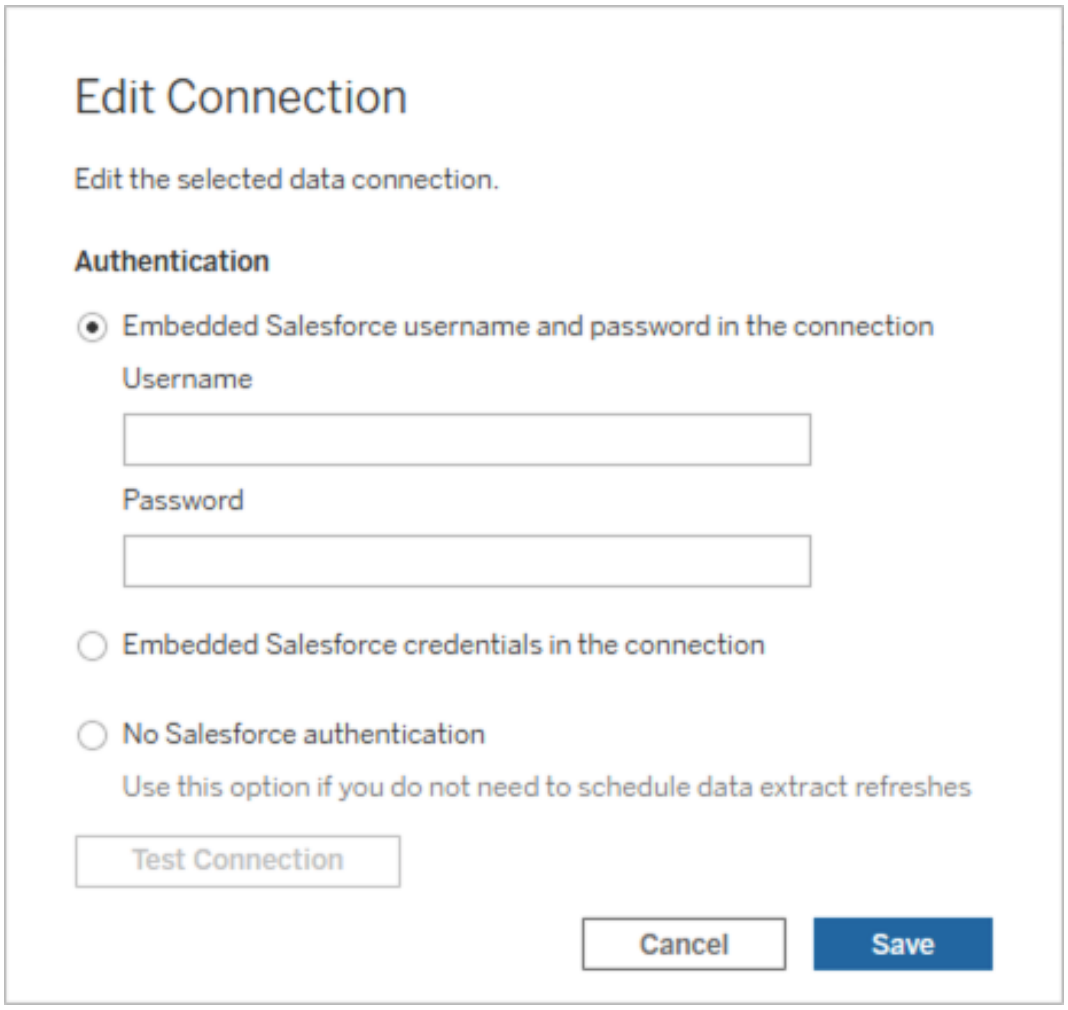
> **Note:** This applies only if Tableau Server is configured to use saved credentials for Salesforce with OAuth. If the server is not configured for this, use the standard process above for modifying connections. For more information about configuring Tableau Server to use saved credentials with OAuth, see Change Salesforce.com OAuth to Saved Credentials.

When you edit Salesforce.com connections, you can select any of the following options in the Edit Connection dialog box:

- Select **Embedded Salesforce username and password in the connection** to use a traditional authentication method.

- Select **Embedded Salesforce credentials in the connection** to use an OAuth connection and schedule refresh tasks, and then select an existing account from the list or click **Add a Salesforce Account** to add a new one.

  When you add a new account, the Salesforce.com sign-in page appears. After you provide your database credentials, Salesforce.com prompts you to confirm Tableau access to the data. When you allow Tableau access, Salesforce.com creates an access token through which it connects to the data.



- Select **No Salesforce authentication** to require users to sign in to Salesforce.com each time they connect. (This option does not allow scheduled extract refreshes.)

Monitor progress

When you save your changes in the **Edit Connection** dialog box, the dialog displays the progress. If you close the dialog box, the modifications continue to run in the background until completed. Tableau Server will make as many changes as possible. Any failures will be skipped, but they will not impede other changes. For example, if you try to change the server name and add a password to several connections, the server names will be changed, and the passwords on workbooks will be changed. However, because you cannot add a password to a data source, the passwords for the data sources will not be changed.

For information about checking the progress of these tasks, see Background Tasks for Extracts.

Cube Data Sources

Cube (multidimensional) data sources have certain characteristics that make them unique in Tableau.

Cube data sources do not support pass-through connections. This means that when a cube data source is published, you cannot make a connection from Tableau Server using the data source. It also means you cannot create a workbook using the data source in Tableau Server.

Publishing a cube data source to Tableau Server gives you the ability to store the data source on the server. However, to use the data source, you must download the data source to Tableau Desktop and use it locally. To download a published data source you need:

- The **Download/Save As** permission for the data source. For more information, see Permissions.
- Correct drivers installed and ports opened on computer running Tableau Desktop.

For information about using cube data sources with Tableau Desktop, see Cube Data Sources.

Web Data Connectors in Tableau Server

Web data connectors (WDCs) are web pages that provide a data connection that is accessible over HTTP for data sources that don't already have a connector in Tableau. WDCs allow users to connect to almost any data that is accessible over the web and to create extracts for their workbooks. Data sources for a WDC can include internal web services, JSON data, REST APIs, and other sources that are available over HTTP or HTTPS. Users can create their own WDC or use connectors that were created by others.

For information about how to use a WDC in Tableau Desktop, see Web Data Connector in the Tableau Desktop documentation.

For information about how to create a WDC, see the Web Data Connector documentation on Github.

Before you run connectors on Tableau Server

As a security measure, Tableau Server won't run WDCs unless you approve the connector, as explained in this topic.

> **Note**: You must be a server administrator to approve WDCs for use on Tableau Server.

WDCs require your approval because they contain executable code and typically make requests to third-party websites. Before a user can use a WDC with Tableau Server, you must add the domain and port used by the connector to a safe list and also include the domains that a connector can send requests to and receive requests from on a secondary safe list. Before you do this, we recommend that you vet and test the connector so that you know what the connector does and what sites it connects to. For more information, see Testing and Vetting Web Data Connectors.

When you add a connector to the safe lists, you configure Tableau Server to allow connections to a particular URL where the connector is hosted and from a URL which the connector can query. This is the only way to allow Tableau Server to run WDCs. The connectors

can then be hosted on a server inside your organization's firewall or on an external domain. Importing WDCs is not supported for Tableau Server.

Manage connectors in a safe list

To add a WDC to the safe list, use the `tsm data-access web-data-connectors add` command. This command and the related commands described below let you perform the following tasks:

- Add WDCs to the safe list and secondary safe list.
- Allow or disallow all WDCs, or WDC refreshes.
- Remove one or more connectors from the safe list.
- List all WDCs on the safe list and secondary safe list.

Updating WDC safe lists requires a server restart

After running any commands that make changes to WDCs, you need to apply your pending changes using the `tsm pending-changes apply` command.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Add connectors to the safe list and secondary safe list

To add a connector to the safe list and secondary safe list, use the tsm data-access web-data-connectors add command, providing the name, safe list URL, and secondary safe list URLs. You can also manage WDCs using the web-data-connector-settings Entity. After running this command, you need to apply your pending changes using the `tsm pending-changes apply` command. A prompt warns you that the command will restart Tableau Server. If

Tableau Server is currently running it is restarted, and if it is stopped it is left in a stopped state with no restart.

```
tsm data-access web-data-connectors add --name "USGS Earthquakes" -
-url https://t-
ableau-
.github.io:443/webdataconnector/Examples/html/earthquakeUSGS.html -
-secondary https://t-
ableau-
.git-
hub.io/.*,ht-
tps://earth-
quake.usgs.gov-
/.*,ht-
tps://max-
cdn.-
boot-
strap-
cdn.-
com/.*,ht-
tps://ajax.googleapis.com/.*,https://connectors.tableau.com/.*
```

Notes on formatting:

- Be sure to use straight quotes (" and '), not curly or "smart" quotes, around the name of the WDC. Use double quotes (") if the name contains a space.
- For many WDCs, the port specified for the `--url` option is 443 or 80, but you can check the value for your connector by looking at the data source details on Tableau Server. You must specify the port number as part of the URL if the WDC is using SSL (HTTPS). For example, to use the default port for HTTPS, the URL might look like the following: `https://example.com:443/WDC/`.

- A URL or comma-delimited list of URLs must be specified for the `--secondary` option, which indicates the domains that provide the libraries and sources that the WDC needs to access. This option cannot be omitted or left empty unless the WDC does not use any secondary domains. If you don't know whether the WDC uses secondary domains, or what the secondary domains are, you might need to contact the developer of the WDC. You can also choose to use `http://.*` and `https://.*` wildcard URLs to allow all domains. However, we strongly recommend that you use more specific URLs to increase security.
- To add an entire domain to the secondary safe list, end the domain URL with a wildcard expression. Use `.*` as the wildcard to indicate the entire domain, as in the following example: `https://example.com/.*`.

Allow or disallow WDCs or WDC extract refreshes

To allow or disallow WDCs, or WDC extract refreshes, use the `tsm data-access web-data-connectors allow` command with the `-t` or `-r` options. By default, WDCs and WDC extract refreshes are allowed. If you change this setting, you need to run the `tsm pending-changes apply` command. For more information about the command and the command options, see `tsm data-access web-data-connectors allow`.

Remove one or more WDCs from the safe list

To remove one or more WDCs from the safe list, use the `tsm data-access web-data-connectors delete` command. If you change this setting, you need to run the `tsm pending-changes apply` command. For more information, see `tsm data-access web-data-connectors delete`.

List all WDCs on the safe list

To list all WDCs on the safe list, use the `tsm data-access web-data-connectors list` command. For more information, see: `tsm data-access web-data-connectors list`.

Refresh the extract for a connector

When a user creates a workbook that uses a WDC, Tableau Server creates an extract from the data returned by the connector. If the user then publishes the workbook, the publish process sends the workbook and the data extract to the server.

Tableau can refresh an extract that was created by a WDC, the same as it can refresh any extract. If the connector requires credentials to sign in to the web-based data source, you need to ensure that the credentials are embedded with the data source, and that the WDC is on the safe list for the server. Tableau Server cannot refresh the extract if the connector requires credentials and they are not embedded with the data source. This is because the refresh can occur on a schedule or in some other background context, and the server cannot prompt for credentials.

Currently, there is no way to re-authenticate a data source from Tableau Server directly. If the data source has credentials that expire, or was published without embedding the credentials, the workbook and data extract need to be published again with the new embedded credentials.

If the background process that performs the refresh operation fails, it creates an alert and a log entry that indicates this issue. Users will be able to see that the timestamp on the extract does not change.

To disable refresh for all WDCs, use the `tsm data-access web-data-connectors allow -r false` command.

Troubleshooting

If the server experiences problems with adding connectors to the safe list, you can examine the log files. Be sure to check the log files on both the initial server node and on the other nodes that are running the gateway process. For more information about log files, Tableau Server Logs and Log File Locations.

If the issue is that Tableau Server will not refresh an extract that was created by a WDC, make sure that the `webdataconnector.refresh.enabled` configuration setting has been set

to `true`. If it is set to `false`, run the following command to allow extract refreshes for all WDCs on the server:

```
tsm data-access web-data-connectors allow -r true
```

Note: The safe list is the only way of allowing Tableau Server to run web data connectors. Importing web data connectors was deprecated starting with version 10.5.

Testing and Vetting Web Data Connectors

Web Data Connectors (WDCs) contain JavaScript that typically connects to data on another site. Because of this, you should test and vet web data connectors before users use them as data sources for a workbook, and before you use them with Tableau Server.

This topic includes some suggestions for testing and vetting web data connectors.

Examine the source

The code in a web data connector is in JavaScript, so you can open the file (and any external files that the connector uses) and examine the source code.

Many connectors reference external JavaScript libraries, such as the jQuery library or API libraries for third parties. Validate that the URL for external libraries points to a trusted location for the library. For example, if the connector references the jQuery library, make sure that the library is on a site that is considered standard and safe. If it is practical for you to change the source code of the connector, use HTTPS protocol (`https://`) to reference external libraries (if the source site supports HTTPS) to help verify the site's authenticity.

To the extent possible, make sure you understand what the code is doing. In particular, try to understand how the code is constructing requests to external sites, and what information is being sent in the request.

Note: Experienced JavaScript programmers often compress (minify) their code to reduce the size of the code for download. Dense blocks of code that use cryptic function and variable names are not uncommon. While this can make it more difficult to examine the code, it is not a sign that the code was written to be deliberately difficult to understand.

Test the web data connector in an isolated environment

If possible, test the web data connector in an environment that is isolated from your production environment and from user computers. For example, add a web data connector to a safe list on a test computer or virtual machine that's running a version of Tableau Server that is not used for production.

Monitor the traffic created by the web data connector

When you test a web data connector, use a tool like Fiddler, Charles HTTP proxy, or Wireshark to examine the requests and responses that the connector makes. Make sure that you understand what sites the connector makes requests to and what content the connector is requesting. Similarly, examine the responses and their content to be sure that the connector is not reading data or code that is not directly related to the connector's purpose.

Test the performance and resource usage of the web data connector

When you test a web data connector, use tools to monitor its CPU and memory usage. Remember that the web data connector will run on Tableau Server, which is an environment in which many processes are already running. You want to make sure that when the connector fetches data, the connector does not have an undue impact on server performance.

Check whether the connector writes to disk. If it does, check how much disk space it occupies, and examine the output to make sure you understand what it's writing and why.

Enable Tableau Catalog

Tableau Catalog discovers and indexes all of the content on your Tableau Cloud site or Tableau Server, including workbooks, data sources, sheets, metrics, and flows. (The legacy Metrics feature was retired in February 2024 for Tableau Cloud and in Tableau Server version 2024.2. For more information, see Create and Troubleshoot Metrics (Retired).) Indexing is used to gather information about the content, or metadata, about the schema and lineage of the content. Then from the metadata, Catalog identifies all of the databases, files, and tables used by the content on your Tableau Cloud site or Tableau Server.

Catalog is available with the Data Management license. For more information, see About Data Management.

In addition to Catalog, metadata about your content can also be accessed from both the Tableau Metadata API and the Tableau REST API using Metadata Methods.

Before enabling Catalog

As a Tableau Server admin, there are a few things that you need to consider before and while enabling Catalog to ensure optimal performance of Catalog in your Tableau Server environment.

Required versions

Before enabling Catalog, make sure you're running *one* of the following versions of Tableau Server:

- At least Tableau Server 2019.3.4 or later
- At least Tableau Server 2019.4.2 or later
- At least Tableau Server 2020.1.0 or later
- At least Tableau Server 2020.2.15 or later
- Tableau Server 2020.3 and later

For more information about why these versions are required, see the Tableau Knowledge Base.

What to expect when enabling Catalog

When Catalog is enabled, the content that already exists on your Tableau Server is immediately indexed.

Initial ingestion

The indexing process is comprised of two primary components, one of which is called initial ingestion. Initial ingestion can be broken down into two additional components:

- Content backfill
- Lineage backfill

The status of content backfill and lineage backfill are important to note later on when monitoring progress and validating that Catalog has been successfully turned on and is running in your Tableau Server environment.

Initial ingestion speed

The time it takes Catalog to index the content for the first time depends on a couple of factors:

- **Amount of content on Tableau Server:** The amount of content is measured by the total number of workbooks, metrics, published data sources, and flows published to Tableau Server. For more information, see Disk space to store metadata.
- **Number of non-interactive microservice containers:** Catalog uses the non-interactive microservice containers to index all the content on Tableau Server. For more information, see Memory for non-interactive microservice containers.

Understanding the factors that impact initial ingestion can help you gauge how long it might take to enable and run Catalog in your environment.

Disk space to store metadata

During initial ingestion, metadata is generated and stored in the Tableau Server repository (“relationship” PostgreSQL database). The amount of disk space needed to store the

metadata is roughly up to half of the disk space currently used by the repository ("workgroup" PostgreSQL database).

For example, suppose the repository uses 50 GB of disk space prior to enabling Catalog, the repository can use up to 75 GB of disk space after enabling Catalog.

Memory for non-interactive microservice containers

Initial ingestion runs inside of the non-interactive microservice container. The non-interactive microservice container is one of two Tableau Server microservice container processes. By default, one instance of the non-interactive microservice container process is added to every node that has a backgrounder process installed.

By default, initial ingestion on a single instance of the non-interactive microservice container can use up to 4 GB of memory on the backgrounder node. If the amount of content on Tableau Server exceeds 10,000, a non-interactive microservice container process may require up to 16 GB of memory on the backgrounder node. Therefore, when enabling Catalog, ensure that every backgrounder node has the available capacity to support each non-interactive microservice container during the initial ingestion process. If capacity needs to be increased, you must update the JVM heap size for non-interactive microservice containers to allocate up to 16 GB of memory on the backgrounder nodes. For more information, see non-interactive.vmopts.

If you are planning to add more non-interactive microservice containers to decrease the time of initial ingestion, first determine how many total containers are needed (using Step 2: Estimate how long initial ingestion will take, below) and then verify if your Tableau Server environment is configured with enough capacity to support *all* non-interactive microservice containers. Depending on how your Tableau Server environment is already configured, you might not be able to add all the additional non-interactive microservice containers that you need to decrease initial ingestion time.

Best practices for enabling Catalog

Because the speed of initial ingestion and requirements are unique to each Tableau Server environment, Tableau recommends that when you enable Catalog you do one or more of the following:

- Make sure there is enough disk space that the Tableau Server repository can use to support the additional metadata that initial ingestion will generate and store. As a general rule, the repository will need an additional 50% of disk space currently used by the repository. For more information about Tableau Server disk usage, see Server Disk Space.
- Depending on the amount of content on Tableau Server, make sure each backgrounder node has at least 4-16 GB of available memory for each instance of a non-interactive microservice container during initial ingestion.
- Perform the process over the weekend to allow initial ingestion to complete before your users start using Catalog capabilities.
- Perform the process in a test environment with production content first. This is because the type of content that needs to be ingested can play a significant role on ingestion speed.

Summary of steps to enable Catalog

The following steps summarize the process to turn on and run Catalog on Tableau Server. The steps must be performed sequentially.

1. Determine the amount of content on Tableau Server
2. Estimate how long initial ingestion will take
3. Decrease the time of initial ingestion
4. Activate the Data Management license
5. Turn off Catalog capabilities
6. Run the tsm maintenance metadata-services command
7. Monitor initial ingestion progress and validate its status

8. Configure SMTP
9. Turn on Catalog capabilities

Note: Because indexing metadata about Tableau content on Tableau Sever is powered by the Metadata API, enabling the Metadata API is required to run and use Catalog.

Enable Catalog

Step 1: Determine the amount of content on Tableau Server

To determine the amount of content on Tableau Server, do the following:

1. Sign in to Tableau Server using your admin credentials.
2. Go to the **Explore** page.
3. Click the Top-Level Project drop-down menu and add the numbers next to **All Workbooks**, **All Metrics**, **All Data Sources**, and **All Flows** together. This is the total amount of content on Tableau Server.

Step 2: Estimate how long initial ingestion will take

To estimate the time it will take Catalog to ingest content on Tableau Server for the first time (initial ingestion), compare *your* Tableau Server setup to a *baseline* Tableau Server setup.

For a Tableau Server with the following setup, initial ingestion could take about 6 hours to complete.

Components	**Baseline values**
Content	17,000 workbooks, metrics, published data sources, and flows
Non-interactive microservice containers	10
Ingestion	**~6 hours**

If you have roughly *half* the content in your Tableau Server environment, initial ingestion might take half the time to complete.

For example: 8,500 (workbooks, metrics, published data sources, and flows) + 10 non-interactive microservice containers = ~3 hours (initial ingestion)

If you have roughly *double* the content in your Tableau Server environment, initial ingestion might take double the time to complete.

For example: 34,000 (workbooks, metrics, published data sources, and flows) + 10 non-interactive microservice containers = ~12 hours (initial ingestion)

Step 3: Decrease the time of initial ingestion

As a general rule, the time it takes for Catalog to perform initial ingestion is correlated to the number of non-interactive microservice containers. To help decrease the time of initial ingestion, you can increase the number of non-interactive microservice containers.

Increase the number of non-interactive microservice containers

By default, one non-interactive microservice container is added to every node that has a backgrounder. To help decrease the time of initial ingestion, Tableau recommends that you increase the number of non-interactive microservice containers using the `tsm topology set-process` command.

1. Open a command prompt as an admin on the initial node (where TSM is installed) in the cluster.
2. Run the command: `tsm topology set-process --count <process_count> --node <node_ID> --process <process_name>`

   For example, to increase the non-interactive microservice containers on the initial node to 4 containers, run the following command:

   ```
   tsm topology set-process --count 4 --node node1 --process non-interactive
   ```

For more information about running the command and its global options, see tsm topology.

Important: Before increasing the number of non-interactive microservice containers, review the following:

- The recommendation for increasing non-interactive microservice containers is for the total number of non-interactive microservice containers, not total non-interactive microservice containers per node. For example, suppose you have 4 nodes but you want to increase the number of non-interactive microservice containers to 8. The `--count` value you use in the tsm command is 2.
- For each non-interactive microservice container added, 4 GB of additional memory will be used on the node and load will be added to the Tableau Server repository (PostreSQL database).
    - Tableau recommends that you incrementally increase non-interactive microservice containers by no more than 2 at a time while closely monitoring your Tableau Server environment to avoid issues with CPU utilization of the Tableau Server repository (PostgreSQL database).
    - Be aware that when too many non-interactive microservice containers are added, CPU utilization of the PostgreSQL database might spike and failover. Symptoms to watch for include SQLException errors in the vizportal logs. For more information, see Repository Failover topic.

Step 4: Activate the Data Management license

(Requires Data Management)

If not already done, you can activate Data Management. For more information, see License Data Management.

Step 5 (optional): Turn off Catalog capabilities for each site

(Requires Data Management)

As part of Data Management activation, Catalog capabilities are turned on by default. Because of the indexing process and the estimated time it takes to complete, consider temporarily turning off Catalog capabilities for each site so that Tableau Server users can't access Catalog capabilities until Catalog is ready and able to provide complete and accurate results.

1. Sign in to Tableau Server using your admin credentials.
2. From the left navigation pane, click **Settings**.
3. On the General tab, under Tableau Catalog, clear the **Turn on Tableau Catalog** check box.
4. Repeat steps 2-3 for each site on your Tableau Server.

Step 6: Run the tsm maintenance metadata-services command

Run the `tsm maintenance metadata-services` command to enable the Tableau Metadata API. Running the command begins initial ingestion. If your Tableau Server is licensed with Data Management, running the command also turns on Catalog capabilities (if it wasn't turned off earlier).

1. Open a command prompt as an admin on the initial node (where TSM is installed) in the cluster.
2. Run the command: `tsm maintenance metadata-services enable`

   For more information about running the tsm command, see tsm maintenance.

Notes: When running this command, keep the following points in mind:

- This command stops and starts some services used by Tableau Server, which causes certain functionality, such as the Recommendations capability, to be temporarily unavailable.

- A new index of metadata is created at this time. Running this command any subsequent times will create and replace the previous index.

Step 7: Monitor initial ingestion progress and validate its status

Running the tsm command above starts the initial ingestion process. To ensure that the initial ingestion process is going smoothly, you can monitor its progress using the Backfill API. For more information, see Get Initial Ingestion Status.

Step 8: Configure SMTP Setup

If not already set up for Tableau Server, configure SMTP Setup. SMTP supports sending email to owners who need to be contacted about changes to data. For more information about configuring SMTP, see Configure SMTP Setup.

Step 9 (optional): Turn on Catalog capabilities for each site

(Requires Data Management)

If you turned off Catalog capabilities before enabling Catalog in one of the procedures above, you must turn on Catalog to make its capabilities accessible to your users.

1. Sign in to Tableau Server using your admin credentials.
2. From the left navigation pane, click **Settings**.
3. On the General tab, under Tableau Catalog, select the **Turn on Tableau Catalog** check box.
4. Repeat steps 2-3 for each site on your Tableau Server.

Troubleshoot Catalog

You or your users might encounter one of the following issues when using Catalog.

Timeout limit and node limit exceeded messages

To ensure that Catalog tasks that have to return a large number of results don't take up all Tableau Server system resources, Catalog implements both timeout and node limits.

- **Timeout limit**

  When tasks in Catalog reach the timeout limit, you and your users see the following message:

  "*Showing partial results, Request time limit exceeded. Try again later.*" or TIME_LIMIT_EXCEEDED

  To resolve this issue, as a Tableau Server admin, you can increase the timeout limit using the `tsm configuration set -k metadata.query.limits.time` command. For more information, see the tsm configuration and tsm configuration set Options topics.

  Important: Increasing the timeout limit can utilize more CPU for longer, which can affect the performance of other processes on Tableau Server.

- **Node limit**

  When tasks in Catalog reach the node limit, you and your users see the following message:

  NODE_LIMIT_EXCEEDED

  To resolve this issue, as a Tableau Server admin, you can increase the node limit using the `tsm configuration set -k metadata.query.limits.count` command. For more information, see the tsm configuration and tsm configuration set Options topics.

  Important: Increasing the timeout limit can affect system memory.

Missing content

- If you suspect, after initial ingestion, content is missing from Catalog, you can use the Eventing API to help troubleshoot. Eventing handles indexing content on Tableau Server after initial ingestion. For more information, see Get Eventing Status.

- When the connection between an embedded external asset and its downstream Tableau content is removed, it remains in Catalog (or the Tableau Metadata API) until it's automatically deleted by a backgrounder process that runs everyday at 22:00:00 UTC (coordinated universal time). For example, suppose a workbook, initially published with an embedded text file A, is republished with an embedded text file B. File A remains visible (or query-able) as an external asset until the backgrounder processes is able to delete it.

  You can disable this backgrounder process from running if you do not want to remove these types of external assets or if you believe that it takes up system resources that you don't want to dedicate to this process. Alternatively, you can adjust the number of external embedded assets that are deleted. For more information, see features.DeleteOrphanedEmbeddedDatabaseAsset and databaseservice.max_database_deletes_per_run.

  You can monitor this process in one of two ways:

  - Filter on the **One-time job re-canonicalize existing database/table assets after a canonicalization logic change** task type in the Background Tasks for Non Extracts admin view.
  - Refer to the **Finished removal of orphaned embedded databases** or **database_service_canonicalization_change** events in the Tableau Server log files.

Performance after initial ingestion

In some Tableau Server environments where specific content that is updated very frequently (for example, through high-frequency schedules or command line or API requests), the indexing process might get over saturated. In these cases, as the server admin, you might consider enabling event throttling to better preserve Catalog performance. For more information, see metadata.ingestor.pipeline.throttleEventsEnable.

Note: When event throttling is enabled, users might notice an intended delay in content changes in Catalog.

Out of memory errors

In some cases, Tableau Server out of memory errors can occur as a result of problems with ingesting complex content. If you suspect ingestion is the cause of the out of memory errors on your Tableau Server, contact and work with Tableau Support to metadata.ingestor.blocklist from being ingested to help resolve the issue.

Disable Catalog

You can disable Catalog in one of two ways.

Turn off Catalog capabilities for each site

(Requires Data Management)

You can turn off Catalog capabilities at any time. When Catalog capabilities are turned off, the features of Catalog, such as adding data quality warnings or the ability to explicitly manage permissions to database and table assets, are not accessible. However, Catalog continues to index published content and the metadata is accessible from the Tableau Metadata API and metadata methods in the Tableau REST API.

1. Sign in to Tableau Server using your admin credentials.
2. From the left navigation pane, click **Settings**.
3. On the General tab, under Tableau Catalog, clear the **Turn on Tableau Catalog** check box.

Stop indexing metadata

To stop indexing the published content on Tableau Server, you can disable the Tableau Metadata API. To disable the Metadata API, run the `tsm maintenance metadata-services disable` command. For more information, see tsm maintenance.

Get Initial Ingestion Status

After enabling the Tableau Metadata API using the `tsm maintenance metadata-services` command, you can use the Backfill API to monitor progress of initial ingestion and get status information for content and lineage backfills.

The steps described below must be performed by a server admin and recommended to be used in conjunction with Step 7: Monitor initial ingestion progress and validate its status in the Enable Tableau Catalog topic.

Step 1: Authenticate using the REST API

To access the Backfill API, you must first authenticate against Tableau Server and get a token. You can do this using the Tableau REST API. For more information, see Signing In and Signing Out (Authentication) in the REST API Help.

Alternatively, you can sign in to Tableau Server using your admin credentials.

Step 2: Make a GET request

Make the following GET request or paste the URL into your browser:

```
http://my_tableau_server/api/metadata/v1/control/backfill/status
```

The request initially returns information about content backfill. When content backfill is complete, lineage backfill information is returned.

- For content backfill, the request returns a status summary and additional status information for each content type depending on what content is available on Tableau Server. Indexing for each content type happens concurrently.
- For lineage backfill, the request returns a status summary.

Status values from the response

The following values are returned by the Backfill API.

- **`contentBackfillTotalDurationSeconds`** and **`lineageBackfillTotalDurationSeconds`** is how much time has progressed, in seconds, on the respective backfill type. When `backfillComplete` is `true`, `contentBackfillTotalDurationSeconds` and `lineageBackfillTotalDurationSeconds` is the total time spent to complete the respective backfill type.

For content backfill:

- **contentType** can show the following content types: `PublishedDatasource`, `Database`, `DatabaseTable`, `Metric`, `Workbook`, and `Flow`.
- **contentId** is the identifier of the last indexed item.
- **successfullyIngestedCount** is the number of items successfully indexed.
- **failedIngestedCount** is the number of items that could not be indexed.
- **durationSeconds** is the time spent, in seconds, to index items for the content type.
- **totalCount** is the total number of items to index.
- **checkpointCreatedTime** is the last recorded time, in UTC, an item was indexed. The Backfill API checks for the last indexed item every five minutes.
- **backfillComplete** is `true` when indexing is complete for all items of the content type.

For lineage backfill:

- **totalCount** is the total number of lineage relationships to index.
- **processedCount** is the number of indexed lineage relationships.
- **lastLineageConnection** is the last indexed lineage relationship.
- **backfillComplete** is `true` when indexing is complete for all lineage relationships.

Example response

The request returns JSON text. To view the JSON in a more readable format, you can use a JSON viewer or browser add-on.


```
{
  "contentBackfillTotalDurationSeconds": 362,
  "lineageBackfillTotalDurationSeconds": 14,
  "contentBackfillStatuses": [
    {
        "contentType": "PublishedDatasource",
        "contentId": "sites/1/datasources/-631379806-1912815680",
        "successfullyIngestedCount": 20,
```

```
            "failedToIngestCount": 0,
            "durationSeconds": 312,
            "totalCount": 20,
            "checkpointCreatedTime": "2020-07-29T23:50:25.763Z",
            "backfillComplete": true
        },
        {
            "contentType": "Database",
            "contentId": "sites/1/databases/e1331f9d-4d73-ee04-9edf-
    96fd1c37cb8e",
            "successfullyIngestedCount": 35,
            "failedToIngestCount": 0,
            "durationSeconds": 26,
            "totalCount": 35,
            "checkpointCreatedTime": "2020-04-29T23:50:25.769Z",
            "backfillComplete": true
        },
        {
            "contentType": "DatabaseTable",
            "contentId": "sites/1/tables/d946d084-53a8-09b6-2ad2-93301e6b4b15",
```

```
    "successfullyIngestedCount": 64,
    "failedToIngestCount": 0,
    "durationSeconds": 49,
    "totalCount": 64,
    "checkpointCreatedTime": "2020-04-29T23:50:25.774Z",
    "backfillComplete": true
},
{
    "contentType": "Metric",
    "contentId": "sites/1/metrics/metric1",
    "successfullyIngestedCount": 2,
    "failedToIngestCount": 0,
    "durationSeconds": 254,
    "totalCount": 2,
    "checkpointCreatedTime": "2020-04-29T23:50:25.779Z",
    "backfillComplete": true
},
{
    "contentType": "Workbook",
    "contentId": "sites/1/workbooks/6749399-1501801290",
```

```
            "successfullyIngestedCount": 10,
            "failedToIngestCount": 0,
            "durationSeconds": 267,
            "totalCount": 10,
            "checkpointCreatedTime": "2020-04-29T23:50:25.784Z",
            "backfillComplete": true
        },
        {
            "contentType": "Flow",
            "contentId": "sites/1/flows/4",
            "successfullyIngestedCount": 4,
            "failedToIngestCount": 0,
            "durationSeconds": 195,
            "totalCount": 4,
            "checkpointCreatedTime": "2020-04-29T23:50:25.788Z",
            "backfillComplete": true
        }
      ],
     "lineageBackfillStatus": {
            "totalCount": 45,
```

```
        "processedCount": 18,
        "lastLineageConnection": "CloudFile downstreamWorkbooks Workbook",
        "backfillComplete": false
    }
}
```

Get Eventing Status

After you have enabled Tableau Catalog (or the Tableau Metadata API) in your Tableau Server environment, you can use the Eventing API to gauge indexing performance.

The steps described below must be performed by a server admin.

Step 1: Authenticate using the REST API

To access the Eventing API, you must first authenticate against Tableau Server and get a token. You can do this using the Tableau REST API. For more information, see Signing In and Signing Out (Authentication) in the REST API Help.

Alternatively, you can sign in to Tableau Server using your admin credentials.

Step 2: Make a GET request

Make the following GET request or paste the URL into your browser:

```
http://my_tableau_server/api/metadata/v1/control/eventing/status
```

Status values from the response

The following values are returned by the Eventing API.

- **`contentType`** is the content type that was most recently indexed.
- **`queueSize`** is the number of items in the indexing queue. The larger the queue size, the longer it can take for items to show in Catalog or Metadata API. If the queue size increases over time, you might need to adjust backgrounder capacity to support the

non-interactive containers and indexing process. For more information, see Memory for non-interactive microservice containers.

- **`checkpointCreatedTime`** is the last recorded time, in UTC, an item was indexed. The Eventing API checks for the last indexed item every five minutes.

Example response

The request returns JSON text. To view the JSON in a more readable format, you can use a JSON viewer or browser add-on.


```
{
   "contentType": "PublishedDatasource",
   "queueSize": 312,
   "checkpointCreatedTime": "2020-07-29T23:50:25.763Z"
}
```


Use Lineage for Impact Analysis

Knowing where your data comes from is key to trusting the data, and knowing who else uses it means you can analyze the impact of changes to data in your environment. The lineage feature in Tableau Catalog helps you do both these things.

When you have a Data Management license and Tableau Catalog enabled, you have access to lineage information for your content. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Cloud Help.

Navigate lineage

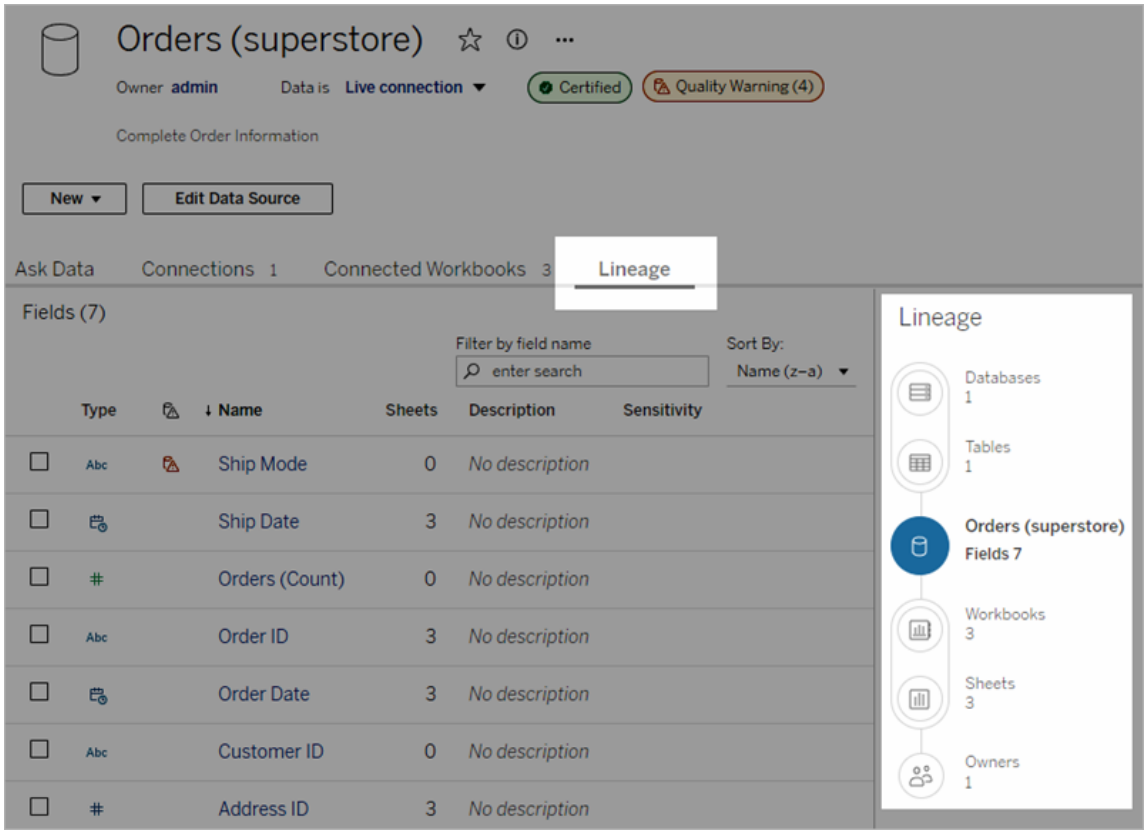
To see the lineage for an asset, first navigate to the asset's page. Your options for this step include:

- Search for the asset and select it.
- Navigate to it from **Explore**.

- If it's an external asset (such as a database or table) that's not in a project, navigate to it through **External Assets**.(This option also works for external assets that *are* in projects.)

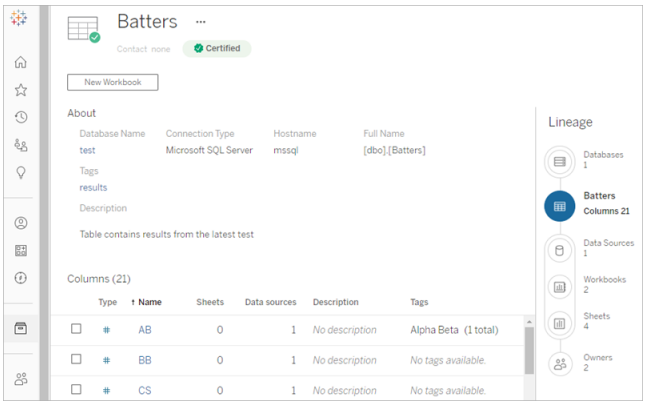
Then select the **Lineage** tab.

> **Note**: Lineage data for flows won't show if the flow includes parameter values. For more information about using parameters in flows see Create and Use Parameters in Flows in the Tableau Prep help.



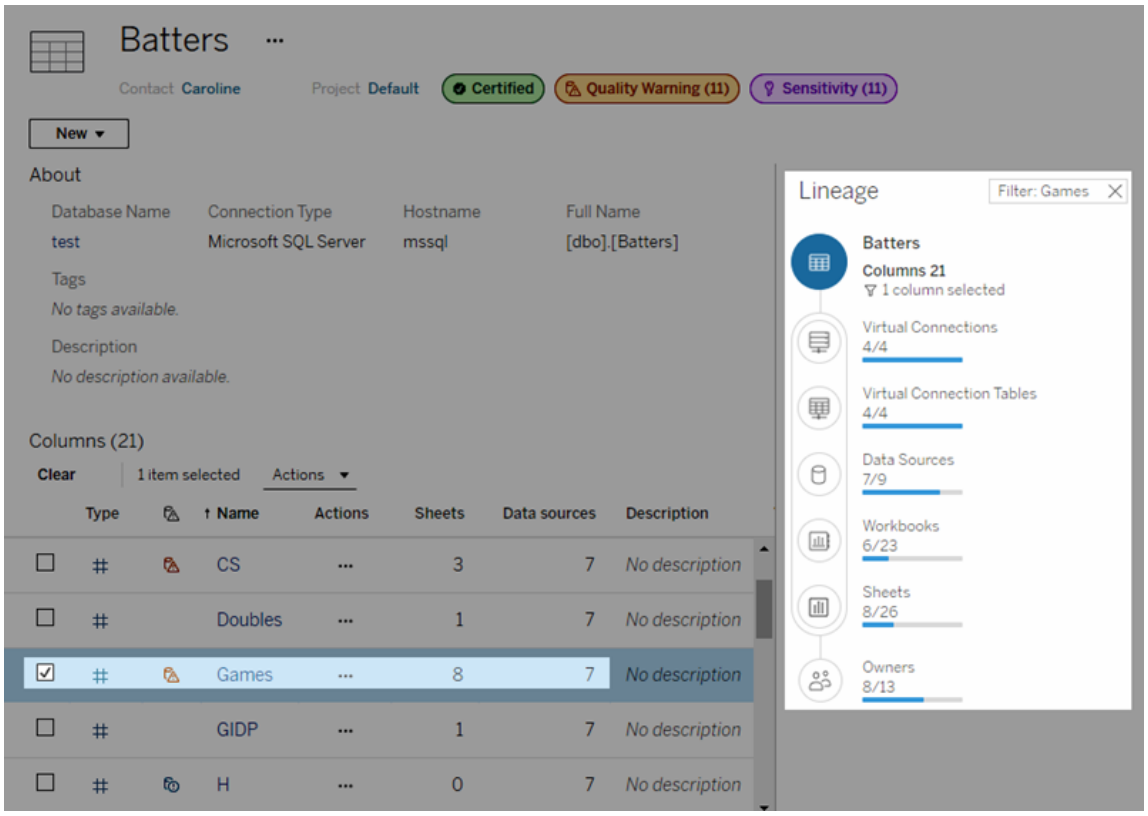
Lineage shows dependencies in relationship to the lineage anchor, which is the asset selected. A lineage anchor can be a database, table, workbook, published data source, virtual connection, virtual connection table, or flow. (In the image above, the anchor is the "Orders (superstore)"data source, and in the image below, the anchor is the "Batters" table.) All the

assets below the anchor depend, either directly or indirectly, on the anchor and are called outputs or downstream assets. The assets above the anchor are the assets the anchor is either directly or indirectly dependent on and are called inputs or upstream assets.



Starting in Tableau Cloud June 2023 and Tableau Server 2023.3, lineage pages for data sources include search and filtering (in the top-right of the fields list) that allow you to quickly find fields of interest or relevance.

When you select a field in a data source or a column in a table, the lineage is filtered to show only downstream assets that depend on the field (or column) or upstream inputs to the field (or column), as in this 'Batters' table example that shows the lineage filtered for the 'Games' column:



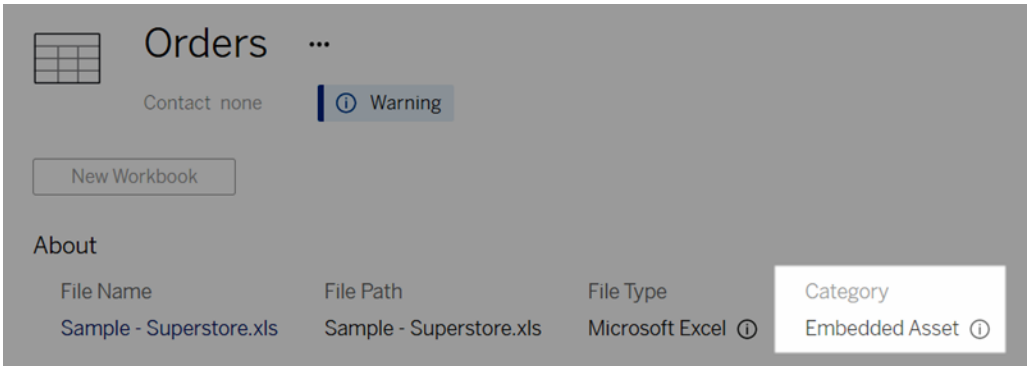
You can select an upstream or downstream asset in the Lineage pane to see its details. For example, when you select Data Sources, the list of data sources that depend on this table appear to the left of the Lineage pane.

From the Lineage pane, you can navigate to any asset related to your initial choice, in this case the table, by following the links that interest you.

Embedded asset appears in External Assets

Tableau Catalog treats an external asset as 'embedded' if the **Include external files** check box is selected when a data source or workbook is published. When an external asset (database, table, or file) is embedded in published Tableau content (workbooks, data sources, and flows), the external asset is used by the content, but is not shareable with other users. That embedded external asset appears in the lineage upstream from its Tableau content and is listed in External Assets.

To see if an external asset is embedded, go to the external asset's detail page and see if "Embedded Asset" is listed under **Category**.



For information about embedded data, see Publishing data separately or embedded in workbooks in Tableau Desktop and Web Authoring Help.

Lineage and custom SQL connections

When you view the lineage of a connection that uses custom SQL, keep in mind the following:

- Lineage might not be complete.
- Catalog doesn't support showing column information for tables that it only knows about through custom SQL.
- Field details cards might not contain links to connected columns, or might not show any connected columns at all.
- Column details cards might not contain links to fields that use the column, or might not show any fields at all.

For more information, see Tableau Catalog support for custom SQL in the Tableau Desktop and Web Authoring Help.

Catalog doesn't support cubes

Cube data sources (also known as multidimensional or OLAP data sources) are not supported by Tableau Catalog. Tableau content (such as a data source, view, or workbook) that relies on cube data does not display any cube metadata or cube lineage in Catalog.

Mismatch between lineage count and tab count

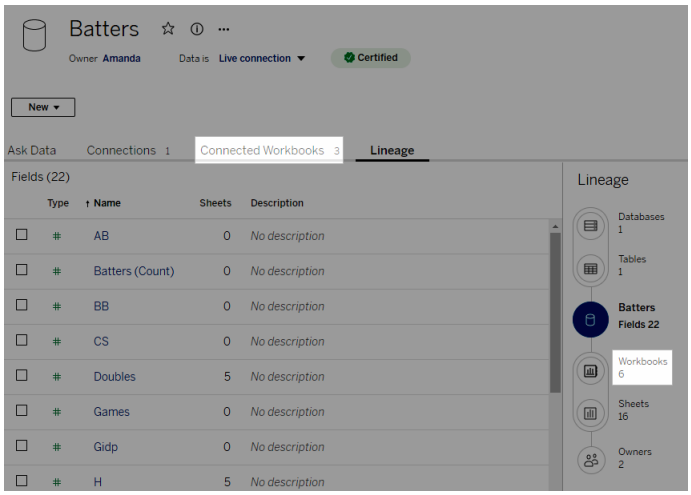
You might notice a mismatch in the count of assets between the Tableau Catalog Lineage tool and the tabs in Tableau Server or Tableau Cloud.

The count mismatch is explained by the fact that each—lineage count vs. tab count—counts assets a different way. For example, at any given point in time, Catalog can count only assets that are indexed, whereas Tableau Server or Tableau Cloud counts any assets that are published. Other reasons for count differences include whether:

- You have "View" permissions for the asset.
- An asset is hidden.
- Any fields are used in a workbook.
- An asset is directly or indirectly connected to.
- An asset is in a Personal Space.

Workbook count mismatch example

As an example, here's how the tab count vs. the lineage count is determined for workbooks.



Connected Workbooks tab counts workbooks that meet both these criteria:

- Connects to the data source (whether or not any fields are actually used in the workbook).
- The user has permissions to view (whether it's a worksheet, dashboard, or story).

Tableau Catalog Lineage counts workbooks that meet all these criteria:

- Has been indexed by Tableau Catalog.
- Connects to the data source and uses at least one field in the data source.
- Contains worksheets, including dashboards or stories that contain a worksheet, that use at least one field in the data source.

When metadata is blocked because of limited permissions, or the asset is in a Personal Space, Catalog still counts the workbook. But instead of seeing some of the sensitive metadata, you see **Permissions required**. For more information, see Access lineage information.

Use email to contact owners

At the end of the lineage is Owners. The list of owners includes anyone assigned as the owner or contact for any content downstream from the lineage anchor.

You can email owners to let them know about changes to the data. (To email owners, you must have the 'Overwrite' (Save) capability on the lineage anchor content.)

1. Select **Owners** to see the list of people who are impacted by the data in this lineage.
2. Select the owners you want to send a message to.
3. Click **Send Email** to open the email message box.
4. Enter the Subject and your message in the text box, and click **Send**.

Data Labels

Data labels are metadata that you can attach to data assets. Data labels help classify data and pass information to users. For example:

- One published data source is more authoritative than other, similarly named ones. The certification data label can help you inform users which data source is recommended.
- A column in a database contains outdated information. A warning data label can help you tell workbook authors and viewers that the data isn't up to date.
- A table of employee income contains sensitive information that shouldn't be shared. A sensitivity data label can inform users that they must take care when using data from the table.

- Some published data sources can be grouped based on the department that published them. A custom label category with custom labels can identify the departments responsible for the data sources.

Note: Data labels are a more recent and extensible way of thinking about ways to classify metadata. Certifications and data quality warnings, which were part of the Data Management license long before the term "data labels" existed, are now considered categories in the broader data label concept, along with the Sensitivity Labels released in Tableau Cloud June 2023 and Tableau Server 2023.3.

A Data Management license is required for all data label operations except for ones related to the certification of published data sources.

Assets you can label

You can add labels to the following Tableau content and external assets:

- Databases
- Tables
- Columns (except for certification) *(column labels introduced in Tableau Cloud October 2022 / Server 2022.3)*
- Data sources
- Flows
- Virtual connections
- Virtual connection tables

Label names and categories

Each label has a name and category. The names and categories built-into Tableau are:

Name	Category
Certified	Certification
Deprecated	Data Quality Warning

Stale data	Data Quality Warning
Under maintenance	Data Quality Warning
Warning	Data Quality Warning
Extract refresh failed	Data Quality Warning
Flow run failed	Data Quality Warning
Sensitive data[1]	Sensitivity

[1]*In Tableau Cloud March 2023 / Server 2023.1 and earlier, the Sensitive data label uses the Data Quality Warning category.*

Starting with Tableau Cloud October 2023 and Tableau Server 2023.3, using the label manager on the Data Labels page or the REST API, an administrator can customize the built-in labels or create new label names and categories. (Using the REST API, Tableau Cloud Administrators have been able to modify some built-in labels and add others in certain categories since June 2023.) For more information, see Manage Data Labels.

Label categories

A label's category affects where and how the label appears, whether it appears on downstream assets, and which parts are customizable, among other things.

Certification

In a self-service environment with multiple publishers and numerous assets, it can be difficult to find recommended content. Using certification, you can mark assets as trusted, and the assets display badges in various places across Tableau. For complete information, see Use Certification to Help Users Find Trusted Data.

Data quality warnings

Identifying problematic data is important for building trust with users. Data quality warnings allow you to mark data assets that have known issues. When you attach a data quality warning

to an asset, a warning shows on it and any downstream assets that use it, making data consumers aware of problems with the source data. For example, if you mark a database table as deprecated, users viewing workbooks based on that table may see a warning.

Furthermore, data quality warnings can be set automatically when an extract refresh or flow run fails, and removed again when it succeeds. And using the Data Labels page or the REST API, administrators can create new, customized data quality warning labels, adding nuance and specificity to the warnings that users can choose from. For complete information, see Set a Data Quality Warning

Sensitivity labels

Some data needs to be handled differently. Using Sensitivity labels, you can relay data sensitivity information to consumers of that data. When you mark an asset as sensitive, users browsing Tableau Cloud see badges on it and any downstream assets that use it. For example, if you mark a table column as sensitive, a user authoring a new workbook based on that table may see a warning. Furthermore, using the **Data Labels** page or the REST API, administrators can create customized sensitivity labels, adding nuance and specificity to the range of classifications that users can choose from when using sensitivity labels.

> Note: Sensitivity labels were introduced in Tableau Cloud June 2023 and Tableau Server 2023.3. Earlier versions of Tableau Cloud and Tableau Server relay data sensitivity through the "Sensitive data" data quality warning instead of using a dedicated sensitivity category.

For complete information, see Sensitivity Labels.

Custom label categories

Sometimes you need to classify data in a way that isn't covered by certification, data quality warnings, or sensitivity labels. Using custom categories that administrators define, you can use labels to categorize assets in any way that your organization sees fit. For example, an administrator in your organization might create a category called "Department" with labels in it

for sales, marketing, and other departments, ready to be applied to assets on your site. For complete information, see Labels with Custom Categories.

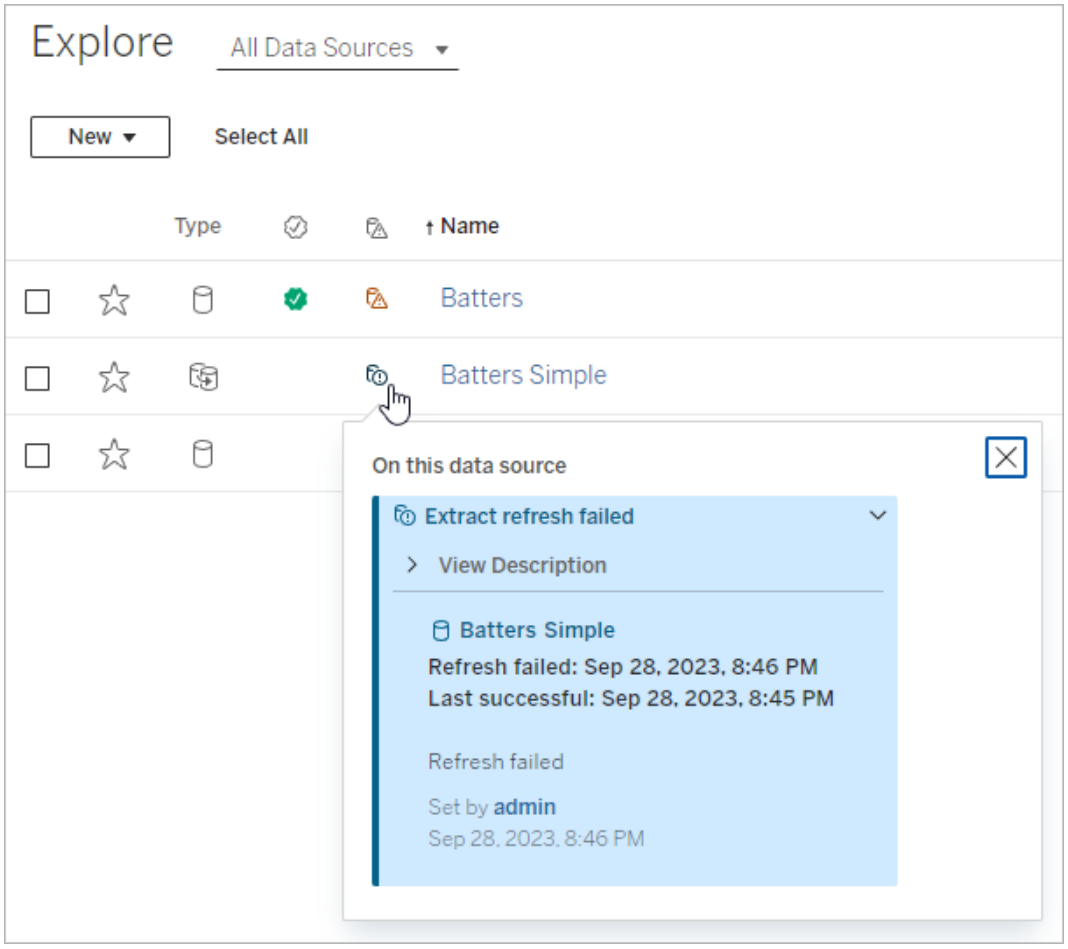
Note: The ability for administrators to create label names and categories through the label manager was released with Tableau Cloud October 2023 and Tableau Server 2023.3. Tableau Cloud administrators could use the REST API in a more limited way to create custom label names with built-in categories in June 2023.

Where data labels appear

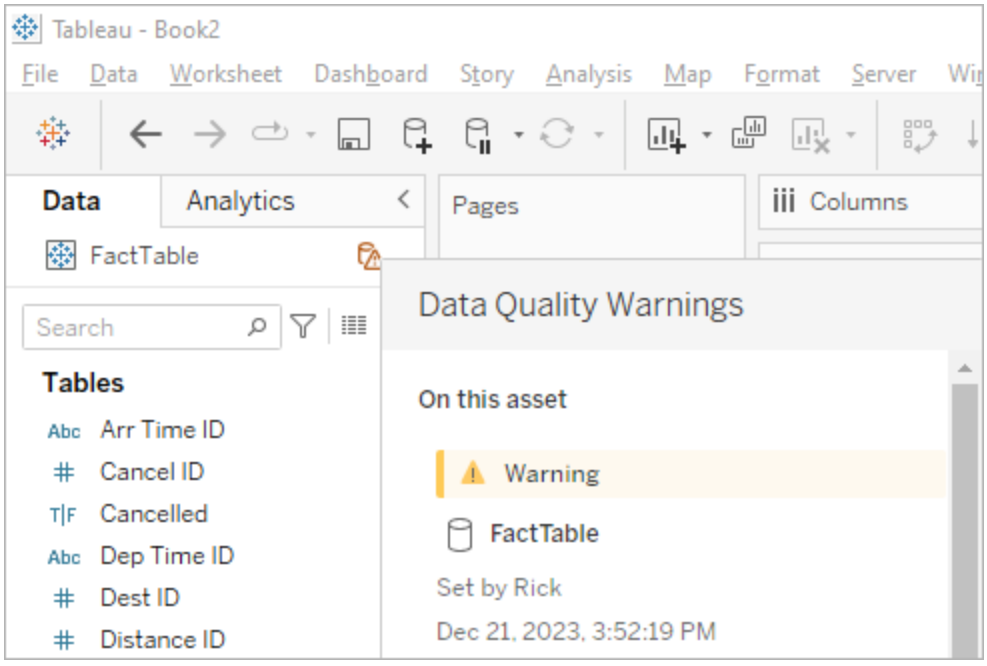
Data labels appear in various places, such as

- the tops of asset pages (workbooks, data sources, tables, and so forth)

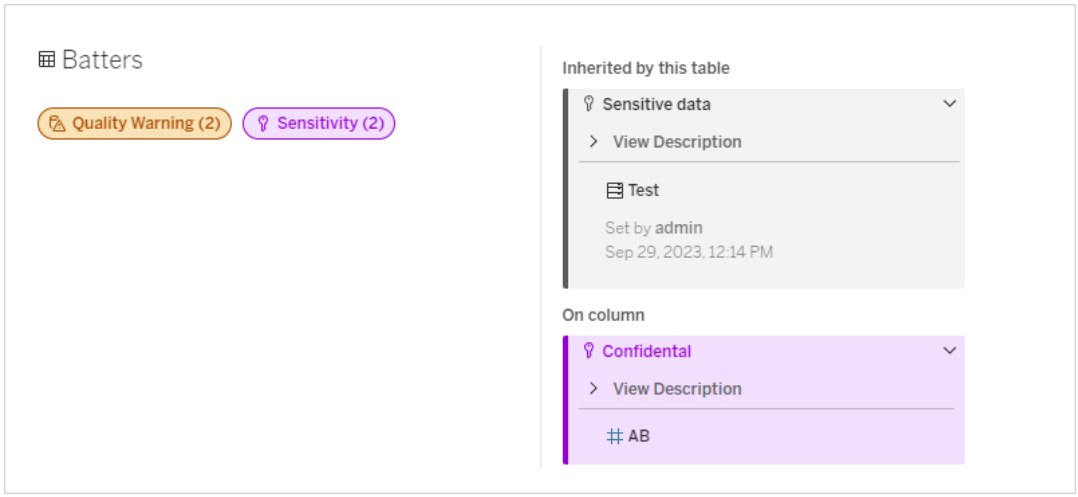
- lists of assets (**Explore** pages, **External Assets** page, and so forth)



- the Desktop **Data** pane



- the web authoring **Catalog Details** window



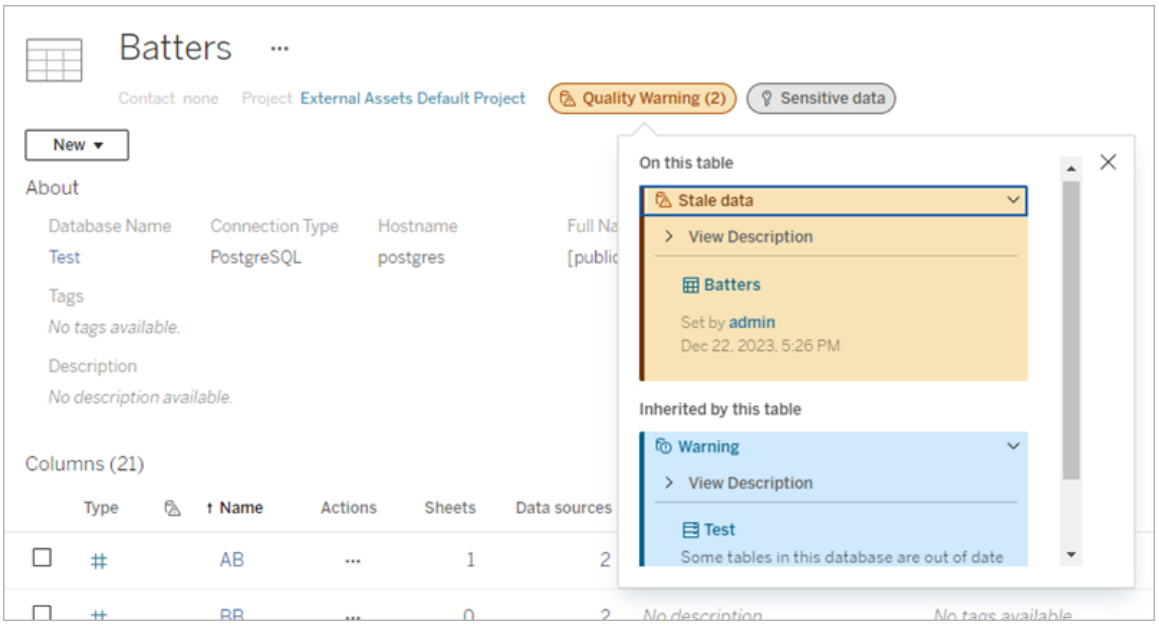
- the **Data Details** pane

Color indicates the category and visibility level of a label:

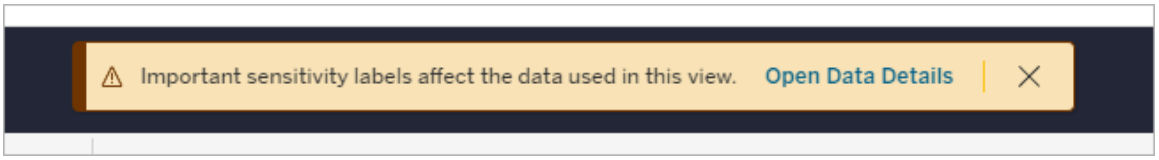
- **Green** indicates the asset is certified
- **Blue** indicates a standard visibility quality warning
- **Yellow** indicates a high visibility quality warning
- **Gray** indicates a standard visibility sensitivity label or label with a custom category
- **Purple** indicates a high visibility sensitivity label

Select a label badge to see details for labels of that category on the asset. In the case of quality warning labels and sensitivity labels, the details will include labels inherited from upstream assets. If you're examining quality warning labels or sensitivity labels on a table, the details will include labels on downstream columns. To go to the related asset's page, select it. Certification labels and custom category labels are not inherited from upstream assets.

There's a single indicator for each label category, no matter how many labels of that category are on the asset or inherited by it. The indicator is colored for high visibility if one or more of the labels it represents are high visibility labels. For example, suppose that a table has a standard visibility quality warning on it, and the table's upstream database has a high visibility quality warning on it. You'll see a yellow **Quality Warning (2)** indicator because the indicator represents two quality warnings, one of which is high visibility.



High visibility quality warnings and high visibility sensitivity labels that affect views and web authoring sessions cause alerts to be shown.

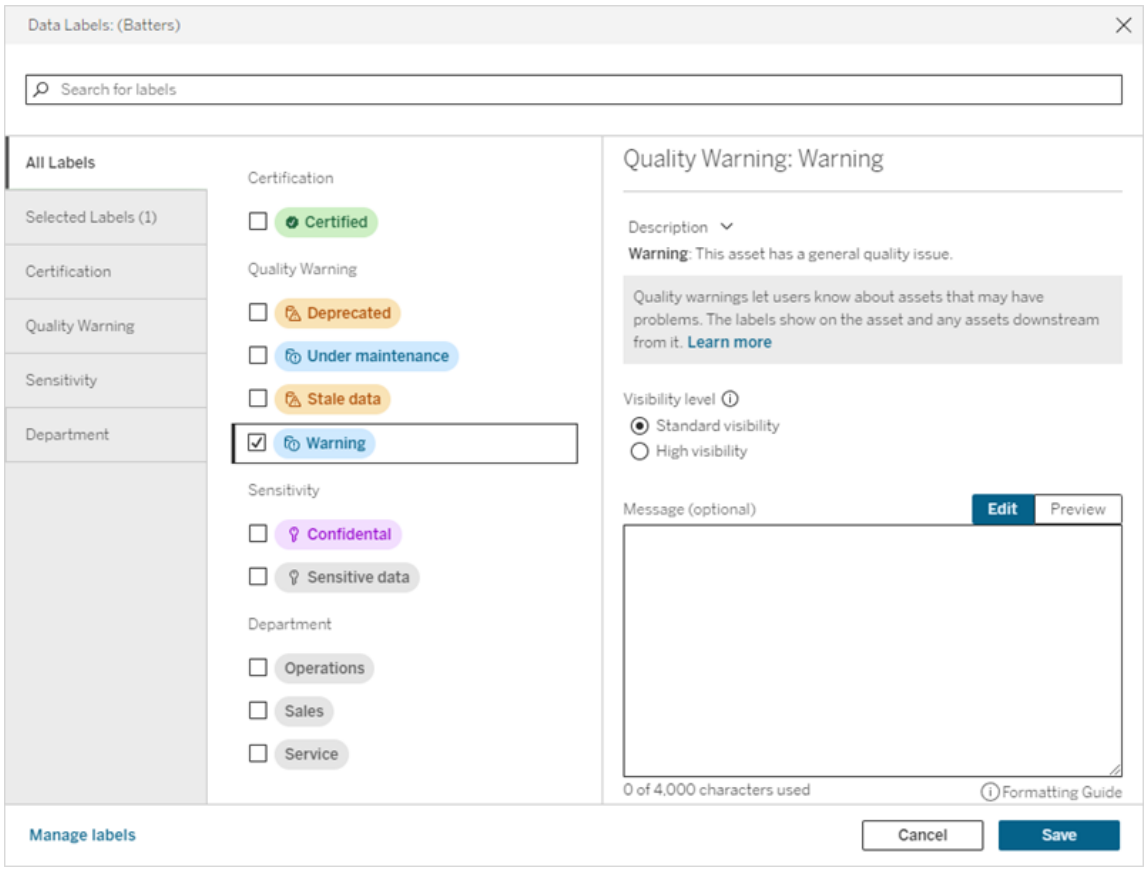


These alerts inform users that the data they're seeing needs to be treated with care. Selecting the Open Data Details link shows more information on the labels affecting the data.

The Data Labels dialog

Starting in Tableau Cloud February 2024 and Tableau Server 2024.2, you add, remove, and modify data labels on assets using the consolidated **Data Labels** dialog. (Extract refresh monitoring and flow refresh monitoring are still controlled through separate dialogs.)

To open the **Data Labels** dialog for an asset, select the actions (...) menu next to the asset, then select **Data Labels**. Choose **Certification**, **Data Quality Warning**, **Sensitivity Label**, or **All Labels** to open the **Data Labels** dialog with the appropriate tab showing.



The vertical tabs on the dialog's left side correspond to data label categories, except for the **All labels** and **Selected labels** tabs at the top of the list.

- The **All labels** tab lists all the site's data labels across all categories. Each label selected for the asset is checked. This includes data labels that were selected for the asset when the dialog was opened, in addition to any labels that have been selected since then.
- The **Selected labels** tab lists all the data labels that have been selected for the asset. This includes data labels that were selected for the asset when the dialog was opened, in addition to any labels that have been selected since then.
- The other tabs correspond to data label categories. Those tabs list all data labels associated with the category. Each label selected for the asset is checked. This includes data labels that were selected for the asset when the dialog was opened, in addition to any labels that have been selected since then.

The search bar at the top of the dialog returns data labels that match the search term you provide. You can select or deselect any labels you want from the results.

To change the data labels on an asset:

1. Navigate to the label by using the search bar or vertical tabs to find the label.
2. To add a data label to the asset, check the box next to it.
3. Select a visibility level (if applicable) and a message if desired.
4. To remove a data label from an asset, clear the box next to it.
5. Add, remove, or modify more labels for the asset by repeating these steps
6. To commit your changes to the data labels on the asset and close the dialog, click **Save**. Or, if you want to abandon all your changes since the dialog opened, click **Cancel** and confirm you want to abandon the changes.

Note: Selecting the data label instead of the check box next to it will show details about the label without changing the status of the check box. This action is useful for seeing the label description or making changes to the message without changing the status of the label on the asset.

The consolidated Data Labels dialog isn't available in Tableau Server.

For detailed information on the labels in specific categories, see the appropriate topic:

- Use Certification to Help Users Find Trusted Data
- Set a Data Quality Warning
- Sensitivity Labels
- Labels with Custom Categories

Permissions required to interact with data labels on assets

Permissions required to view, add, update, and delete labels on assets are as follows:

- To view a data label, you must have **read** permissions on the associated asset.
- To add, update, or delete a data label other than a certification label, you must have **write** permission on the associated asset.
- To add, update, or delete a certification label, you must be an administrator, or else you must be a project leader or product owner for the project the asset is in.

- To add, update, or delete a certification label for an external asset *not* in a project, you must have the **change permissions** permission on the associated asset.

Comparison of data labels and tags

Tableau Cloud and Tableau Server also feature another solution for asset classification: Tags. Data labels and tags differ in significant ways:

Area	Data Labels	Tags
Structure and control	Administrators control the range of data labels	No administrative control over the range of tags users add
Permissions	Ability to add/update/remove data labels is controlled through asset permissions	Explorers and Creators can tag any assets that they can view
Appearance	Data label iconography is easily seen and color-coded by category and visibility level	Tags appear in fewer places than data labels and have no iconography
Inheritance	Some data labels (like warnings and sensitivity labels) show on downstream assets	No inheritance
Searching/Filtering	Certification and quality warnings can be used as filters in some asset lists	Search results return assets with matching tags, and tags can be used as filters in some asset lists
API accessible	Access via REST API and Metadata API is possible	Access via REST API and Metadata API is possible
License requirements	Requires a Data Management license (except for the certification of published data sources)	No licensing requirements
Use	Structured categorization	Open-ended method to cat-

focused on providing important information that can influence users' use of data	egorize assets

For more information on tags, see Use Tags in the Tableau Desktop and Web Authoring Help.

Use Certification to Help Users Find Trusted Data

In a self-service environment with multiple publishers, it's common for a project on Tableau Server to contain a variety of content that is named similarly, or is based on the same or similar underlying data, or is published without any descriptive information about it. When this is the case, analysts might lack confidence about the data they should use.

To help your users find the data that's trusted and recommended for their type of analysis, you can *certify* the data that complies with your organization's data standards.

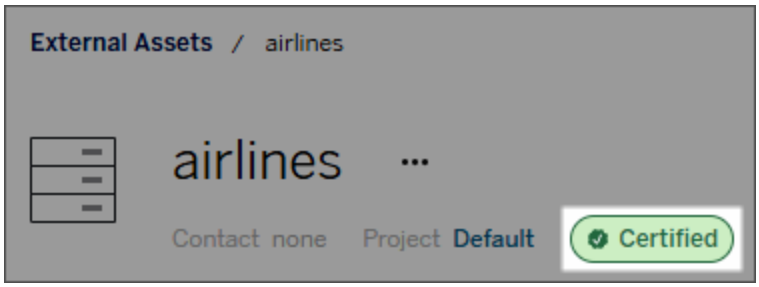
Certification complements the Recommendations Training Schedule feature by offering a way to promote data through curation.

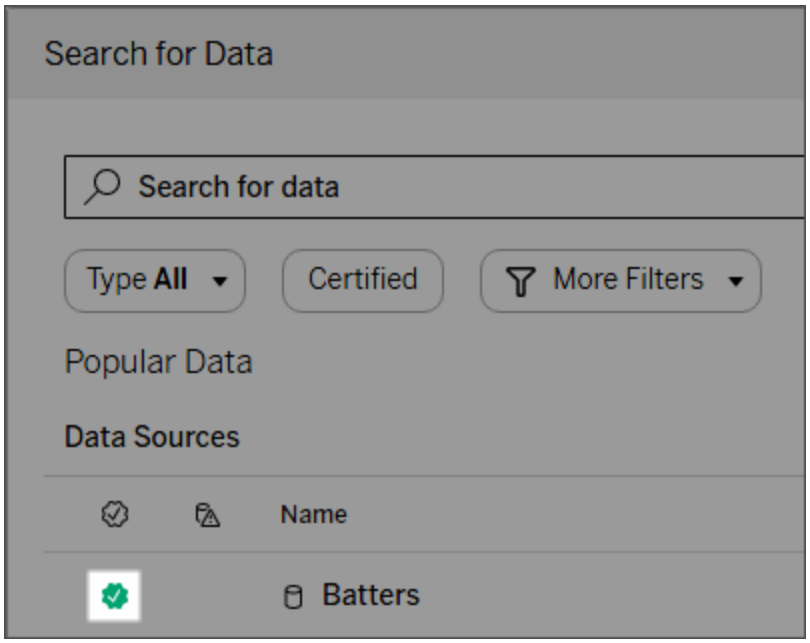
In addition to certifying published data sources, if you have a Data Management license for Tableau Server or Tableau Cloud:

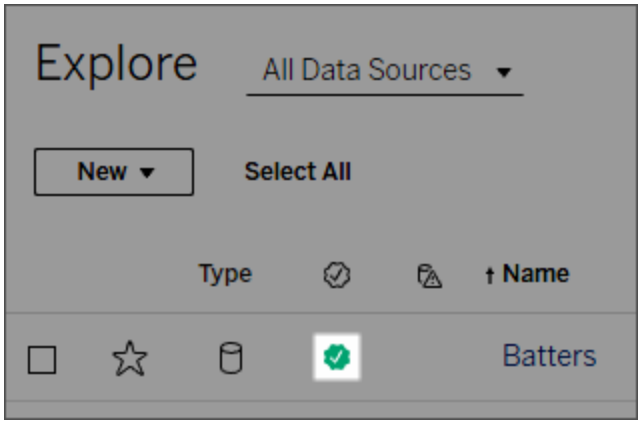
- If Tableau Catalog is enabled, you can certify databases and tables that are associated with your Tableau content. (For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Cloud Help.)
- Starting in Tableau 2022.1, you can certify virtual connections and virtual connection tables.

How certification helps users find trusted data

When you certify an asset, users see a green badge or green check mark, depending on where the asset is being viewed.

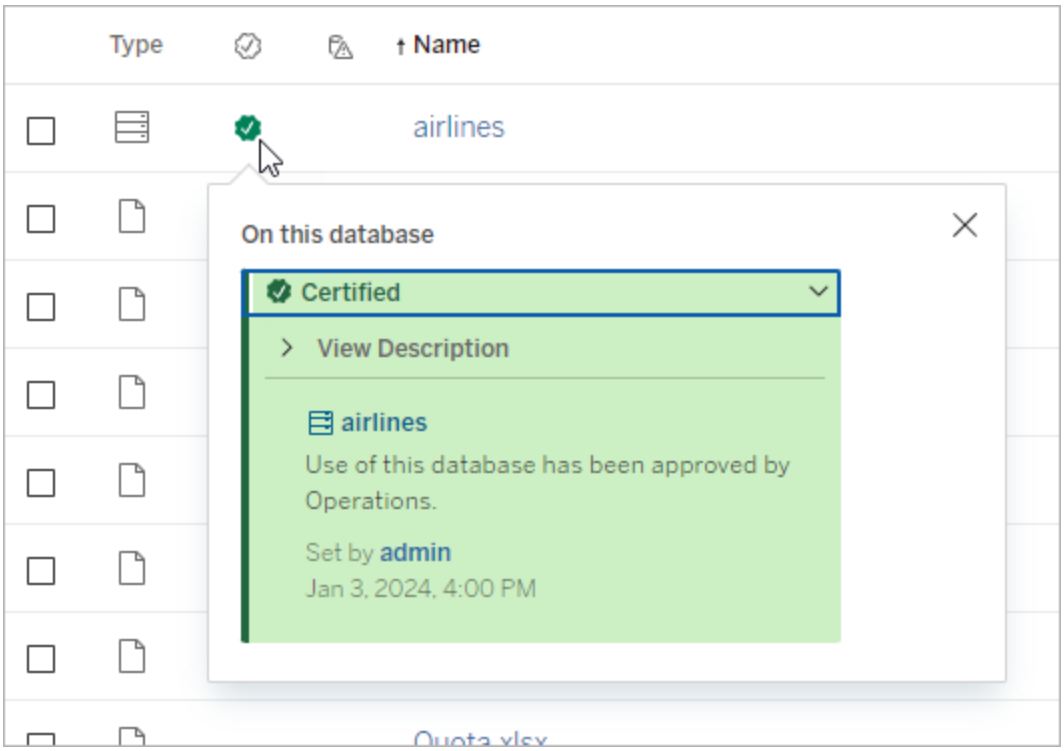






Certified data sources rank higher in search results and are added to recommended data sources.

In addition, you can provide notes about the certification status, which appear when users click the badge, or in a tooltip when they hover over the data source icon in web authoring or Tableau Desktop. The information also shows who certified the data source.



Create guidelines for selecting data to certify

As with most Tableau functionality, certification is flexible. You can define for your organization the criteria you use to determine when to certify an asset. As you do so, document and share your guidelines. The guidelines can help you, other administrators, and project leaders to be consistent with your certification choices. They can also help users understand what certification means.

Whether you use the same certification criteria across all projects, or define unique criteria for each project, the important thing is to be clear about what certification means in your environment.

Who can certify data

To certify a data source, you must

- be a Server or Site Administrator, *or*
- have a site role of **Explorer (Can Publish)** or **Creator** *and* be the project owner or have the **Project Leader** capability for the project containing the data you want to certify.

To certify virtual connections and virtual connection tables, you must have a Data Management license in your environment, and you must

- be a Server or Site Administrator, *or*
- have a site role of **Explorer (Can Publish)** or **Creator** *and* be the project owner or have the **Project Leader** capability for the project containing the data you want to certify.

To certify databases or tables, you must have Tableau Catalog enabled in your environment, and you must

- be a Server or Site Administrator, *or*
- have the **Set permissions** capability on the database to certify that database or any tables within that database.

How to certify data

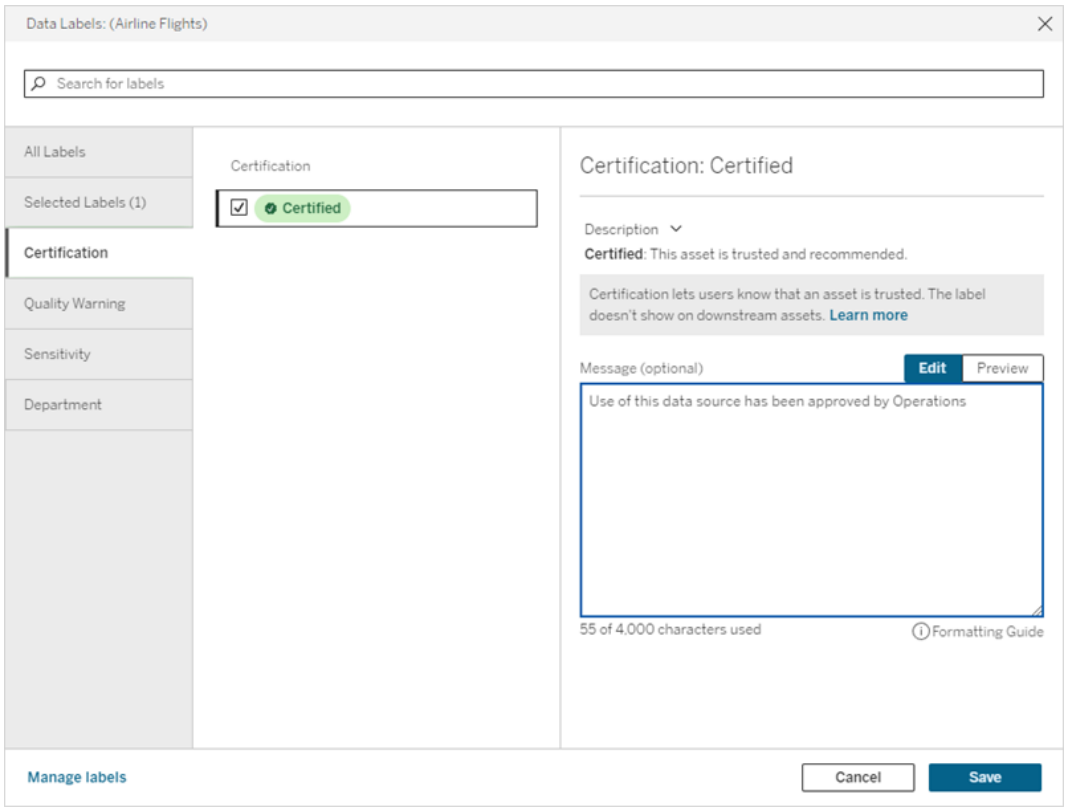
The data you can certify depends on the permissions you have, and whether you have a Data Management license and Tableau Catalog enabled in your environment.

- All users with permissions can certify data sources.
- If you have a Data Management license, users with permissions can also certify virtual connections and virtual connection tables.
- If you have a Data Management license and Tableau Catalog is enabled, users with permissions can also certify databases, tables, and files.

To certify an asset:

Note: Starting in Tableau Cloud February 2024 and Tableau Server 2024.2, you add and remove certifications using the consolidated Data Labels dialog instead of separate dialogs for each type of label. For information on the Data Labels dialog, see The Data Labels dialog.

1. Search for or navigate to the asset. The steps to navigate depend on the type of asset you want to certify:
    - Data source or virtual connection - on the **Explore** page, select **All Data Sources** or **All Virtual Connections**.
    - Virtual connection table - on the **Explore** page, select **All Virtual Connections**, and select the virtual connection that contains the virtual connection table you want to certify. Then select the virtual connection table.
    - Database or table - on the **Explore** page, navigate to the database or table. Or on the **External Assets** page, select **Databases and Files** or **Tables and Objects**.
2. On the page, select the More actions menu (...) next to the asset name you want to certify.
3. Select **Data Labels** > **Certification** (or **Edit Certification** in Tableau Server 2023.1 and earlier)
4. Select the **Certified** checkbox. (In earlier versions of Tableau Server, use the switch.)
5. Add a message if desired. The message gives users context for the certification status, intended use for the data, or other helpful information. Information you add to the **Message** section appears in the certification badge or tooltip, mentioned earlier in How certification helps users find trusted data. You can format the text in a message with bold, underline, and italics, and include a link or an image. To see text formatting tips, click the information (i) icon above the **Save** button. (Starting in Tableau Cloud February 2024, the message is optional. Earlier versions of Tableau Cloud and Tableau Server required it.)



6. Select **Save**.

Customize certification

Beginning with Tableau Cloud June 2023 and Tableau Server 2023.3, using the label manager on the Data Labels page or the REST API, an administrator can change the certification description that users see in the certification dialog. For more information, see Manage Data Labels.

Set a Data Quality Warning

Data quality warnings are a feature of Tableau Catalog, part of the Data Management offering for Tableau Server and Tableau Cloud. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Cloud Help.

When Tableau Catalog is enabled in your environment, you can set data quality warnings on data assets so that users of that data are aware of issues. For example, you might want to let

users know that a data source has been deprecated, or that a refresh has failed for an extract data source.

You can set data quality warnings on data sources, databases, tables, flows, virtual connections, virtual connection tables, and columns.

Data quality warnings for data sources, databases, tables, and flows were introduced in version 2019.3 for Tableau Cloud and Tableau Server. Data quality warnings for virtual connections and virtual connection tables were added in Tableau Cloud March 2022 and Tableau Server 2022.1, and for columns in Tableau Cloud October 2022 and Tableau Server 2022.3.

About data quality warnings

There are two kinds of data quality warnings: Quality warnings that you set, and quality warnings that Tableau sets when an extract refresh or flow run fails, also known as monitoring quality warnings.

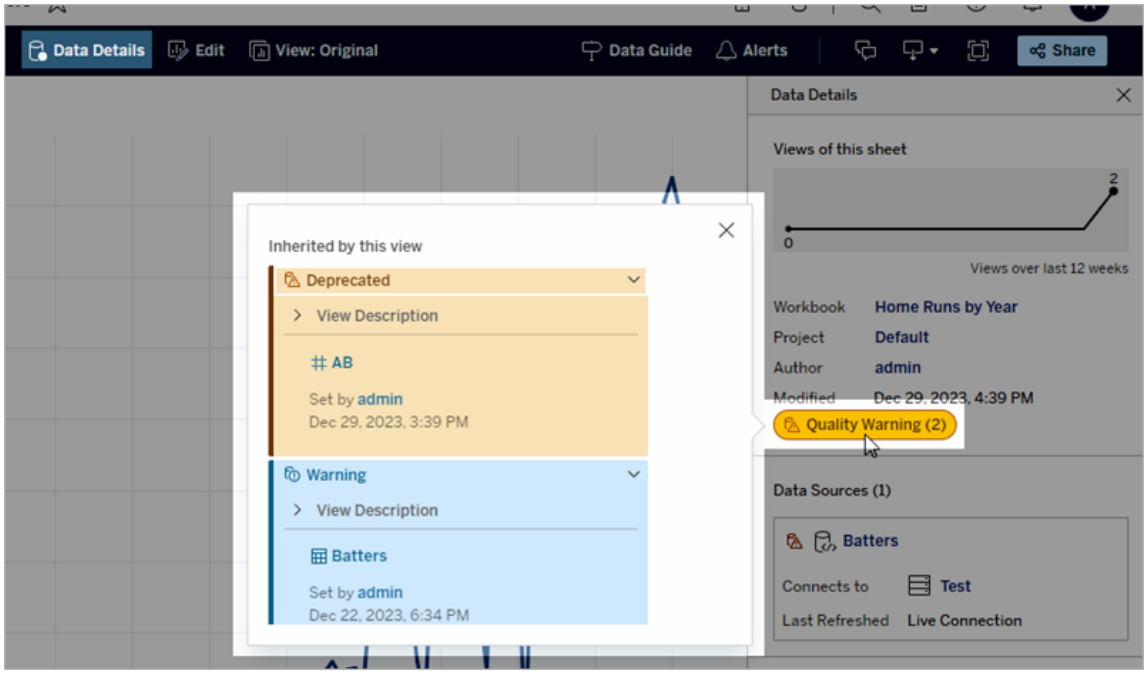
- **Quality warnings that you set**: A quality warning that you set is visible to users until you or another user removes it.
- **Monitoring quality warning**: If you enable a monitoring quality warning for an extract or flow asset, Tableau automatically adds a quality warning to the asset when an extract refresh or flow run fails. Later, if the extract refresh or flow run succeeds, Tableau automatically removes the quality warning.

Starting with Tableau Cloud October 2023 and Tableau Server 2023.3, in addition to setting monitoring warnings at the asset level, you can also turn extract refresh and flow run monitoring on or off for the entire site at once. For information about site-wide monitoring, see Site-wide monitoring for extract refresh and flow run failures.

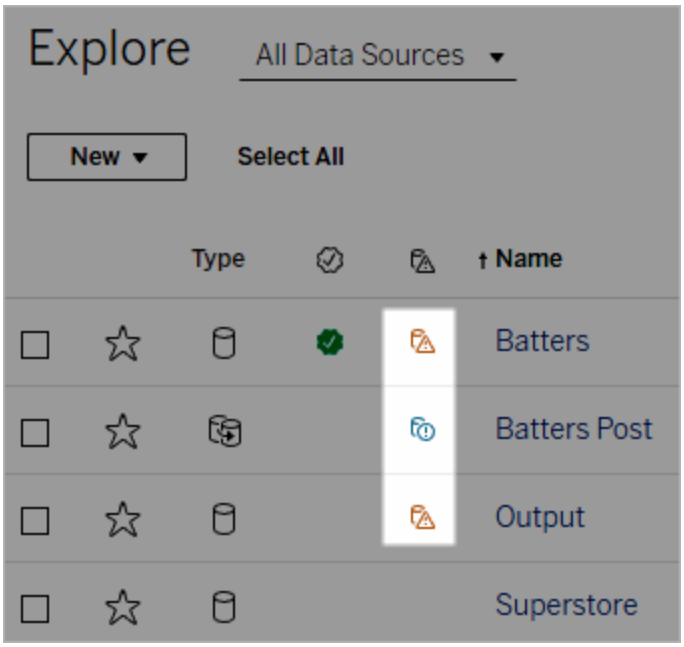
Note: You can enable and modify both kinds of data quality warnings using the REST API. For more information, see the Label and Data Quality Warning Metadata Methods in the Tableau REST API Help.

Where data quality warnings appear

In Tableau Cloud and Tableau Server, when you set a warning on a data source, flow, database, table, column, virtual connection, or virtual connection table, the warning is visible to users of the asset and any assets downstream from it. For example, a warning set on a table is visible to users looking at a dashboard that depends on that table. The users see a warning icon on the dashboard's Data Details tab and can open the pane to see more information.



Data quality warnings appear when exploring some types of content in a list view:



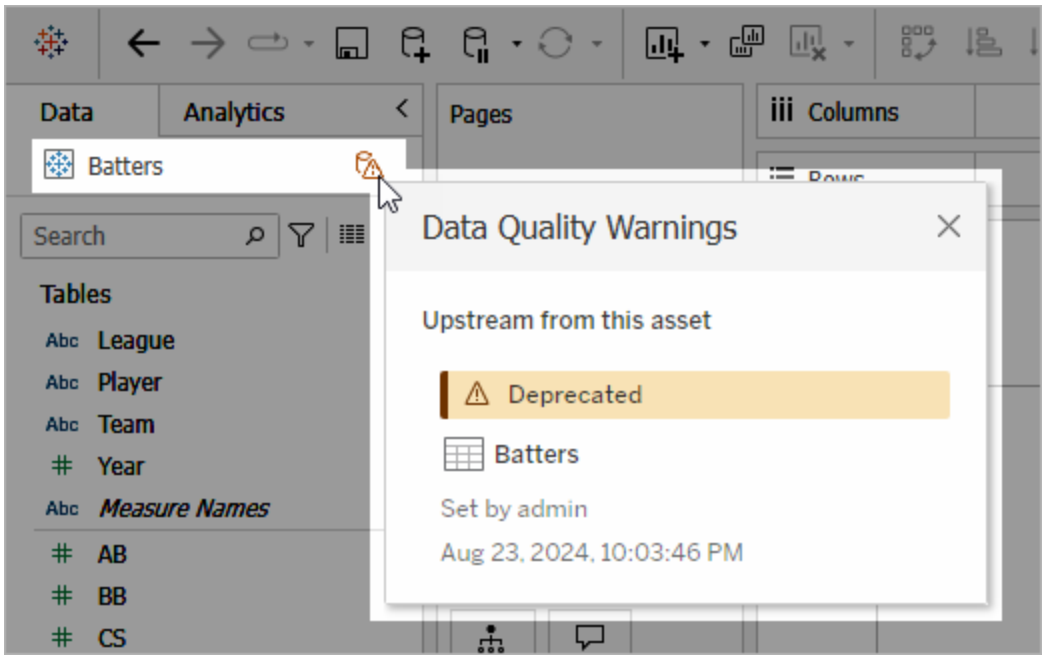
Data quality warnings also appear at the top of asset pages:

In Tableau Desktop, users see an icon next to the data source in the **Data** pane when

- there's a warning on a data source used in the workbook, or
- there's a warning upstream from the data source used in the workbook

Note: Data quality warnings for columns and virtual connections don't appear in Tableau Desktop.

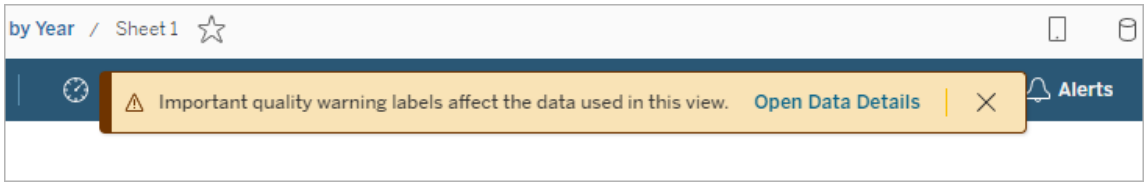
To see the details of the warning, select the warning icon.



In Tableau Cloud web authoring, you can look at all label properties for an object in the data pane (data connection or field) by selecting **Catalog Details** on the data connection or field.

Visibility

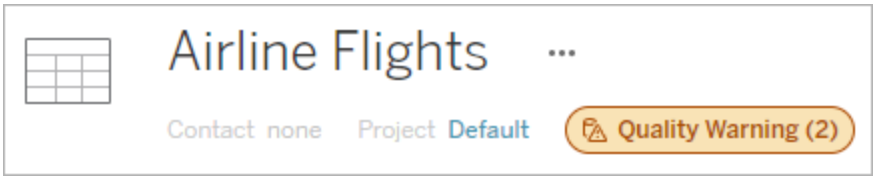
Data quality warnings can be classified as high visibility. High visibility quality warnings appear more urgent and appear in more places. For example, a high visibility warning on a data source generates a notification when anyone opens a view that depends on it.



The icon and badge for high visibility quality warnings is yellow or orange, while standard visibility quality warnings are blue.

If more than one quality warning applies to an asset (because the asset has more than one quality warning or is inheriting upstream ones), the badge includes a number, and the color is determined by the highest visibility label. For example, if two quality warnings apply to an

asset, one of which is standard visibility and one of which is high visibility, the badge is yellow or orange.



Data quality warnings in subscriptions

Administrators can turn on data quality warnings in email subscriptions. If this feature is turned on, emails the users receive include high visibility data quality warnings for that view, with links to:

- Relevant views or workbooks with their **Data Details** pane open.
- Relevant upstream assets, such as data sources, tables, or databases.

Administrators can turn on data quality warnings in email subscriptions by selecting the **High-Visibility Data Labels in View and Workbook Subscriptions** option (previously the **Data Quality Warnings in Subscriptions** option) on the Tableau Server or Tableau Cloud site settings page. For more information, see High-Visibility Data Labels in View and Workbook Subscriptions in the Site Settings Reference.

How to set a quality warning

You can set several different data quality warnings on an asset. Starting with Tableau Cloud June 2023 and Tableau Server 2023.3, an administrator can add to the list of available data quality warnings by customizing data labels.

> Starting with Tableau Cloud June 2023 and Tableau Server 2023.3, "Sensitive data" is no longer a data quality warning, but is a sensitivity label instead. For more information, see Sensitivity Labels. In Tableau Server 2023.1 and earlier, "Sensitive data" remains a data quality warning.

The following data quality warnings are built in:

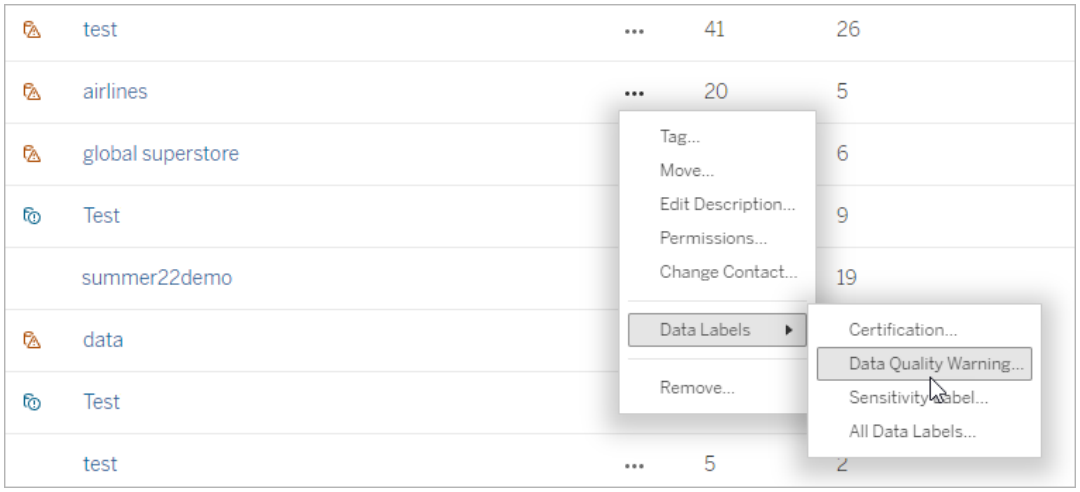
- Warning
- Deprecated
- Stale data
- Under maintenance

To attach a data quality warning to an asset:

> Note: Starting with Tableau Cloud February 2024 and Tableau Server 2024.2, you add and remove quality warnings using the consolidated Data Labels dialog instead of separate dialogs for each type of label. For information on the Data Labels dialog, see The Data Labels dialog.

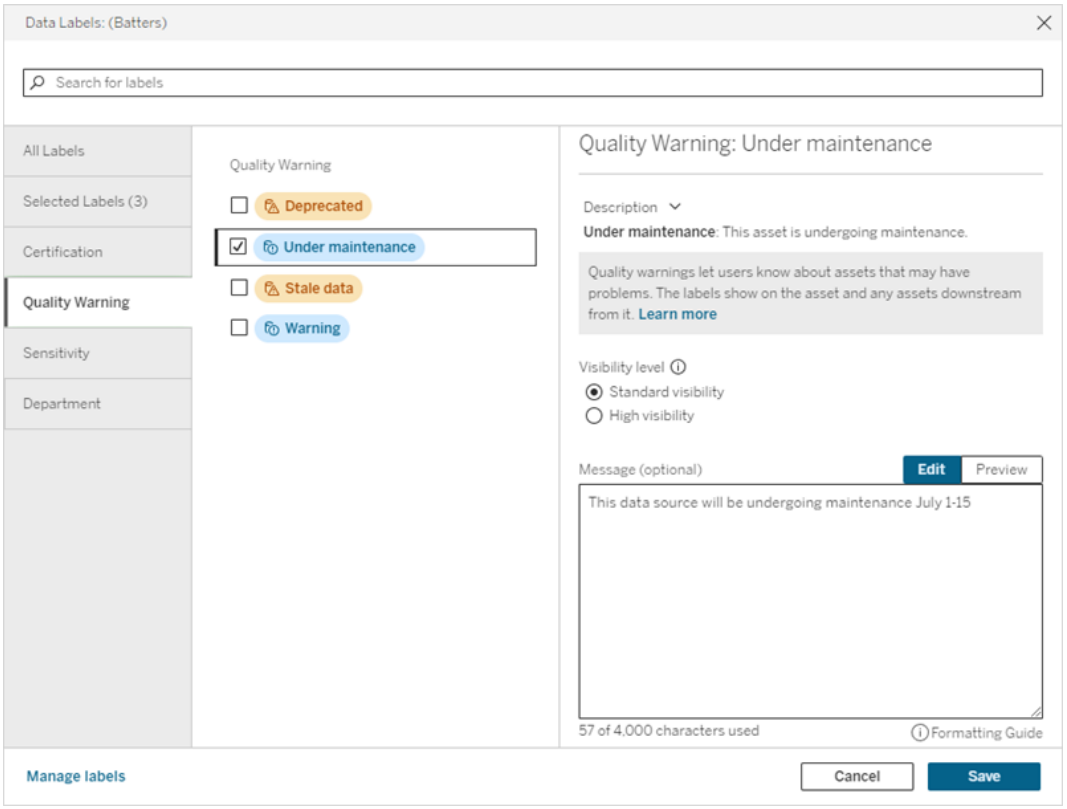
1. Search for or navigate to the asset. The steps to navigate depend on the type of asset you want to add the quality warning to:
    - Data source or virtual connection - on the **Explore** page, select **All Data Sources** or **All Virtual Connections**.
    - Virtual connection table - on the **Explore** page, select **All Virtual Connections**, and select the virtual connection that contains the virtual connection table you want to certify. Then select the virtual connection table.
    - Database or table - on the **Explore** page, navigate to the database or table. Or on the **External Assets** page, select **Databases and Files** or **Tables and Objects**.
    - Column - on the **Explore** page, navigate to the table. Or on the **External Assets** page, select **Tables and Objects** and navigate to the table. Then find the column in the list.
2. Select the actions menu (. . .) next to the asset, and then select **Data Labels** > **Data Quality Warning**. (For columns in Tableau Server 2022.3 and earlier, instead select

the column, and then click the actions dropdown and select **Quality Warning**.)



3. Select the checkbox beside quality warnings you want attached to the asset. Optionally, if you know the name of a quality warning, you can search for it at the top of the dialog, and then select the checkbox beside it. (In Tableau Server 2023.3 and earlier, you can only attach one quality warning to each asset. Use the **Show warning** switch or **Enable warning** checkbox to turn on a quality warning for that asset, then select the desired warning from the dropdown list.)
4. Set the visibility level.
5. If desired, enter a message to display to users. (In Tableau Server 2023.3 and earlier, a message is required.)You can format the text in a message with bold, underline, and italics, and include a link or an image. To see text formatting tips, click the information (i)

icon above the **Save** button.



6. Select **Save**.

Remove a data quality warning

When a warning no longer applies, you can remove it by navigating to the data asset with the warning.

> Note: Starting with Tableau Cloud February 2024 and Tableau Server 2024.2, you add and remove quality warnings using the consolidated Data Labels dialog instead of separate dialogs for each type of label. For information on the Data Labels dialog, see The Data Labels dialog.

1. Select the actions menu (. . .) next to the asset, and then select **Quality Warning**. (For columns in Tableau Server 2022.3 and earlier, instead select the column, and then click the actions dropdown and select **Quality Warning**.)

2. Uncheck the boxes beside quality warnings you want to remove from the asset. (In Tableau Server 2023.3 and earlier, use the **Show warning** switch or **Enable warning** checkbox to turn off a quality warning for that asset.)
3. Turn off the warning.
4. Select **Save**.

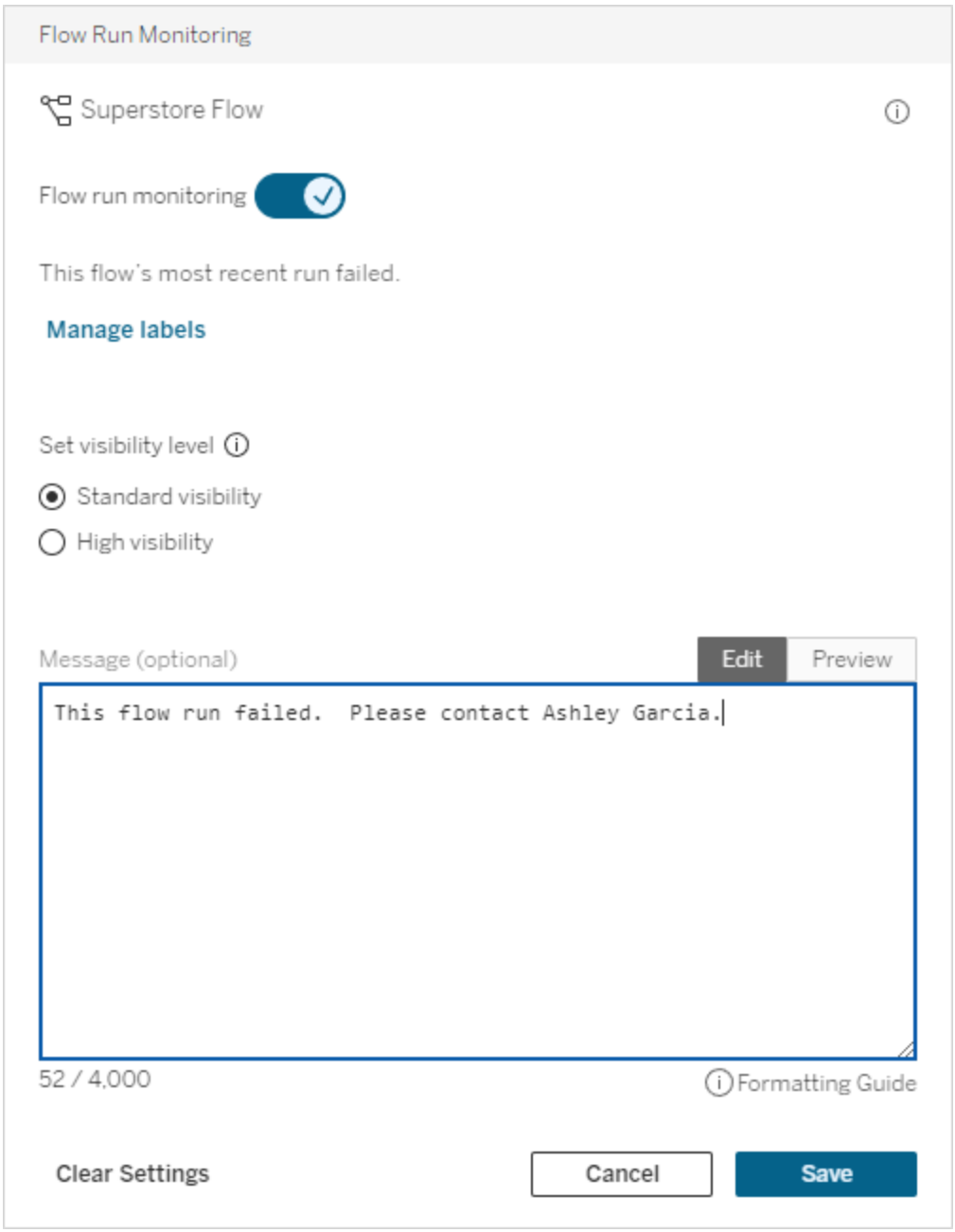
How to turn on a monitoring quality warning

You can set Tableau to monitor for two events: extract data source refresh failure and flow run failure. When the event occurs, Tableau generates a quality warning that appears in the same places that a manual quality warning appears.

You can turn on monitoring explicitly on the extract or flow, or, starting with Tableau Cloud October 2023 and Tableau Server 2023.3, you can enable site-wide monitoring for all extract refresh and flow run failures. For information on site-wide monitoring, see Site-wide monitoring for extract refresh and flow run failures.

To explicitly monitor for either an extract refresh or flow run failure:

1. Select the actions menu (. . .) next to the extract data source or flow you want to create a warning for, and then select the appropriate option:
    - In Tableau Cloud and Tableau Server 2023.3 and later:
        - **Data Labels** > **Extract Refresh Monitoring**
        - **Data labels** > **Flow Run Monitoring**
    - In Tableau Server 2023.1 and earlier:
        - **Quality Warning** > **Extract Refresh Monitoring**
        - **Quality Warning** > **Flow Run Monitoring**
2. Enable the warning.
3. Set the visibility level. (Older versions of the dialogs have a checkbox for high visibility.)
4. If desired, enter a message for users to see in the warning details if the extract refresh or flow run fails. You can format the text in a message with bold, underline, and italics, and include a link or an image. To see text formatting tips, click the information (i) icon above the **Save** button.
5. Click **Save**.



How to turn off a monitoring quality warning

To turn off monitoring for either an extract refresh or flow run failure:

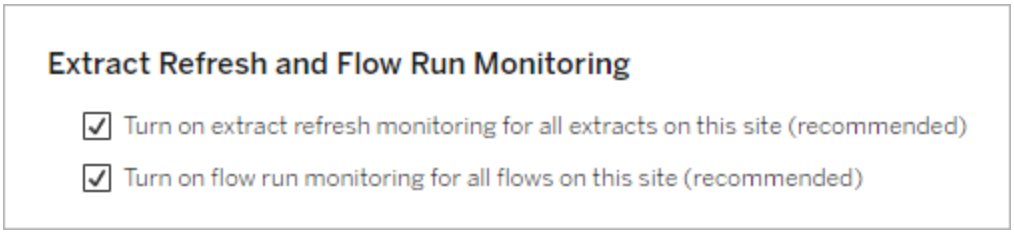
1. Select the actions menu (. . .) next to the extract data source or flow you want to create a warning for, and then select the appropriate option:

- In Tableau Cloud and Tableau Server 2023.3 and later:
  - **Data Labels** > **Extract Refresh Monitoring**
  - **Data labels** > **Flow Run Monitoring**
- In Tableau Server 2023.1 and earlier:
  - **Quality Warning** > **Extract Refresh Monitoring**
  - **Quality Warning** > **Flow Run Monitoring**

2. Turn off the warning.
3. Click **Save**.

Site-wide monitoring for extract refresh and flow run failures

Starting with Tableau Cloud October 2023 and Tableau Server 2023.3, an administrator can turn on site-wide monitoring to add or remove data quality warnings for extract refresh failures and flow run failures. You can control this feature through the Settings page, under the Extract Refresh and Flow Run Monitoring section:



These settings are turned on by default for all new sites. Sites that existed before the change will have the settings turned off, but an administrator can turn them on.

Note: Data quality warning notifications aren't displayed for extract refreshes that use Tableau Bridge.

Interaction of site-wide monitoring and explicit monitoring

The interaction of explicit monitoring on assets and site-wide monitoring of all assets is as follows:

- If monitoring is explicitly turned on for an asset *and* site-wide monitoring is turned on, explicit settings on the asset take precedence over the site-wide settings. Settings include properties like visibility level and message.

- When you turn off site-wide monitoring:
  - Assets *with* monitoring explicitly turned on aren't changed.
  - Assets *without* monitoring explicitly turned on stop monitoring for extract refresh or flow run failures, and warnings that previously arose from extract refresh or flow run failures on those assets are removed.
  - Catalog ingestion performance might be temporarily reduced as Catalog re-ingests assets that may no longer have warning labels.

Site-wide monitoring was released in Tableau Cloud October 2023 and Tableau Server 2023.3. There's no interaction of explicit monitoring and site-wide monitoring in earlier versions.

Who can set quality warnings

To set a data quality warning, you must either

- be a server or site administrator, or
- have the **Overwrite** capability for the asset.

Customize data quality warnings

Starting with Tableau Cloud June 2023 and Tableau Server 2023.3, using the label manager on the Data Labels page or the REST API, an administrator can change the data quality warnings that users see in the data quality warning dialog, or create new ones. For more information, see Manage Data Labels.

Sensitivity Labels

Some data needs to be handled more carefully. To ensure trust and security, it's important that users know which data that is. Starting in Tableau Cloud June 2023 and Tableau Server 2023.3, if you have a Data Management license, Tableau offers a new category of data label: *Sensitivity labels*. Users can use sensitivity labels to indicate the level of care that should be taken when creating views or sharing information. Furthermore, sensitivity labels can co-exist on the same asset as other labels, such as certification and data quality warnings. And, using the label manager on the **Data Labels** page or the REST API, an administrator can create sensitivity labels to suit the needs of their organization.

> Note: In Tableau Cloud March 2023 and Tableau Server 2023.1 and earlier, data sensitivity was expressed using the "sensitive data" data quality warning. With the upgrade to Tableau Cloud June 2023 and Tableau Server 2023.3, "sensitive data" data quality warnings were migrated to sensitivity labels.

Sensitivity labels can be attached to the same types of assets that other data labels can.

Attach a sensitivity label to an asset

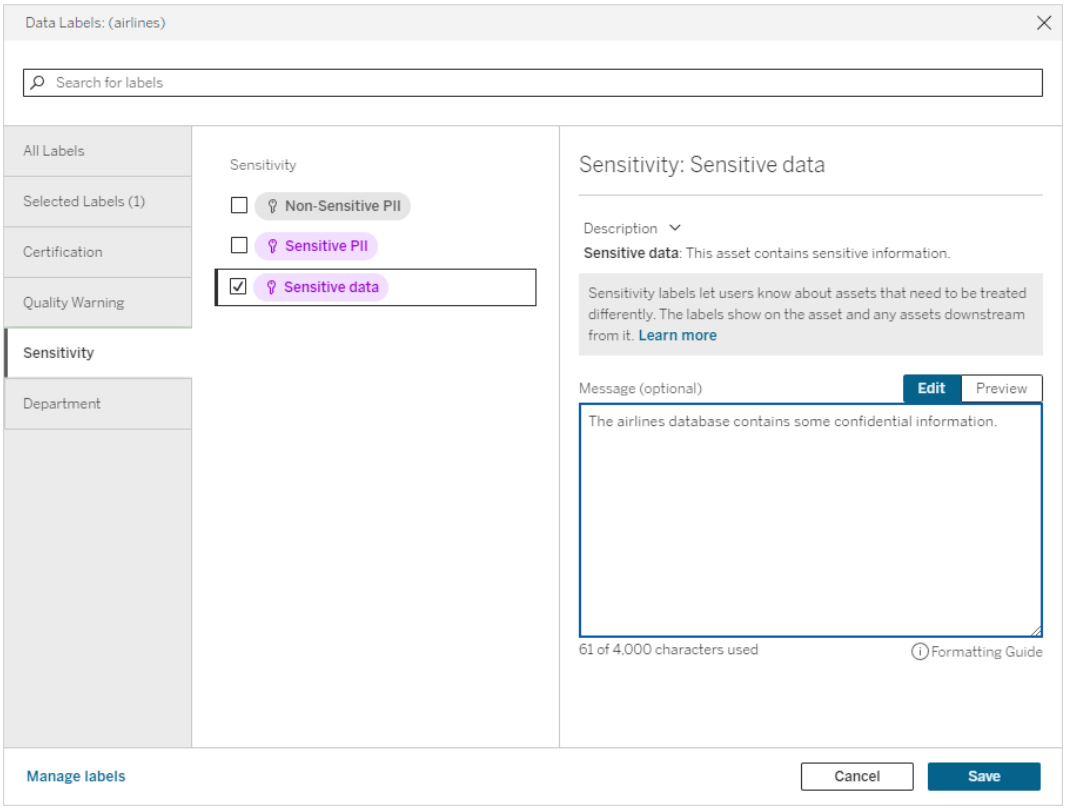
To attach a sensitivity label to an asset:

> Note: Starting with Tableau Cloud February 2024 and Tableau Server 2024.2, you add and remove sensitivity labels using the consolidated Data Labels dialog instead of separate dialogs for each type of label. For information on the Data Labels dialog, see The Data Labels dialog.

1. Search for or navigate to the asset. The steps to navigate depend on the type of asset you want to add the sensitivity label to:
   - Data source or virtual connection - on the **Explore** page, select **All Data Sources** or **All Virtual Connections**.
   - Virtual connection table - on the **Explore** page, select **All Virtual Connections**, and select the virtual connection that contains the virtual connection table you want to certify. Then select the virtual connection table.
   - Database or table - on the **Explore** page, navigate to the database or table. Or on the **External Assets** page, select **Databases and Files** or **Tables and Objects**.
   - Column - on the **Explore** page, navigate to the table. Or on the **External Assets** page, select **Tables and Objects** and navigate to the table. Then find the column in the list.
2. Select the actions menu (. . .) next to the asset, and then select **Data Labels** > **Sensitivity Label**.
3. Select the checkbox beside sensitivity labels you want attached to the asset. Optionally, if you know the name of a sensitivity label, you can search for it at the top of the dialog, and then select the checkbox beside it. (In Tableau Server 2023.3 and earlier, you can

only attach one sensitivity label to each asset. Use the **Show label** switch to turn on a sensitivity label for that asset, then select the desired sensitivity label from the drop-down list.)

4. If desired, enter a message to display to users. You can format the text in a message with bold, underline, and italics, and include a link or an image. To see text formatting tips, hover over the information (i) icon above the **Save** button.
5. Select **Save**.



Remove a sensitivity label from an asset

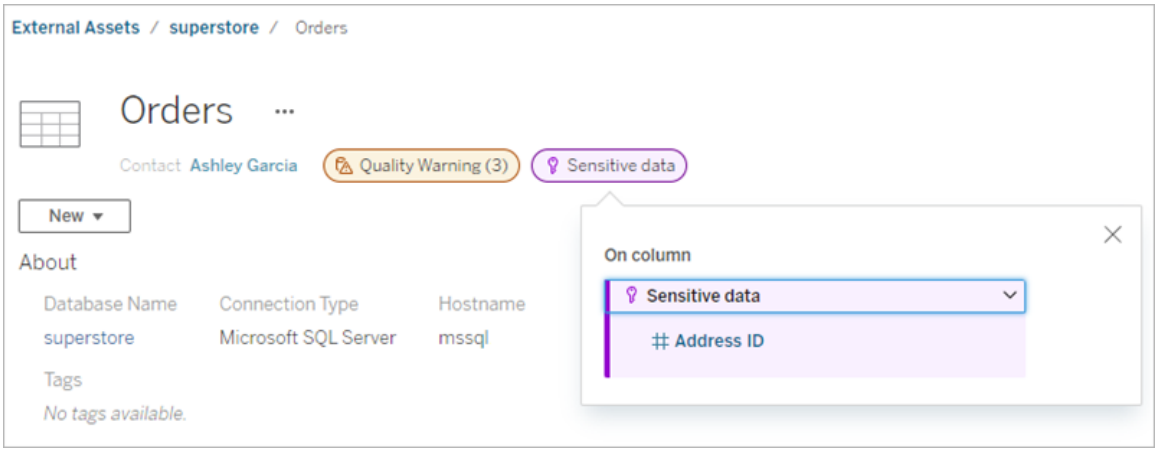
To remove a sensitivity label from an asset:

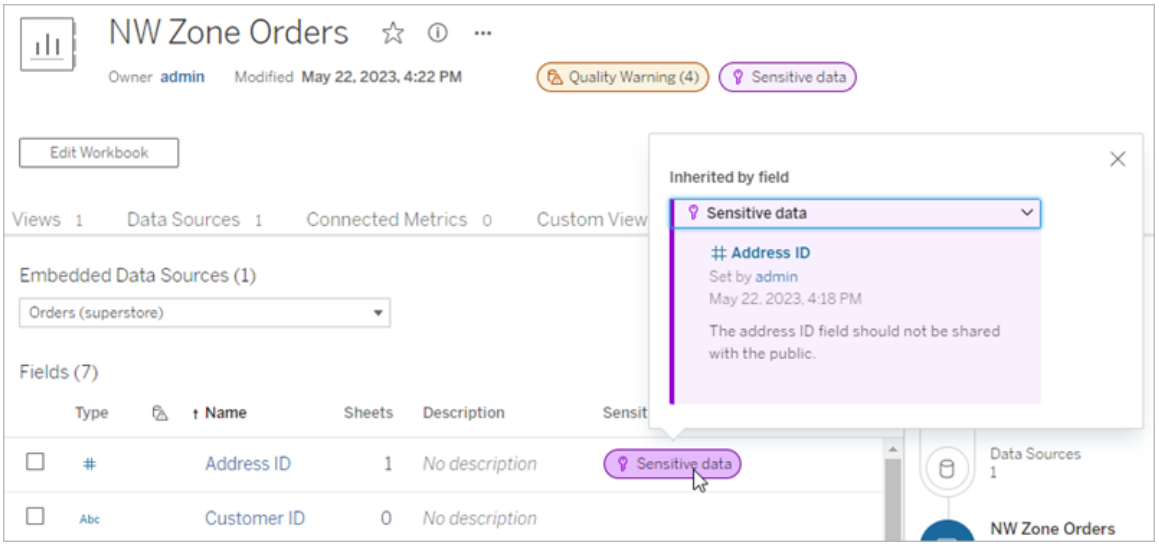
Note: Starting with Tableau Cloud February 2024 and Tableau Server 2024.2, you add and remove sensitivity labels using the consolidated Data Labels dialog instead of separate dialogs for each type of label. For information on the Data Labels dialog, see The Data Labels dialog

1. Select the actions menu (. . .) next to the asset, and then select **Data Labels** > **Sensitivity Label**.
2. Uncheck the boxes beside sensitivity labels you want to remove from the asset. (In Tableau Server 2023.3 and earlier, turn off the label with the **Show label** switch.)
3. Select **Save**.

Where sensitivity labels appear

Sensitivity labels appear on assets when navigating Tableau Server. Like data quality warnings, sensitivity labels appear downstream from the assets on which they're attached. For example, a sensitivity label on a column appears in the columns row of the table page, again at the top of the table page, and on the database page.





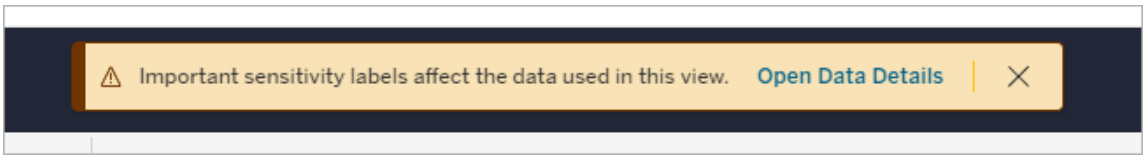
In web authoring, users see an icon next to the data source in the **Data** pane when

- there's a sensitivity label on a data source used in the workbook, or
- there's a warning upstream from the data source used in the workbook

To see the details, hover over the icon with the cursor. Or, in Tableau Cloud web authoring, you can select a data source or a column and then select **Catalog Details** to see all its labels.

Visibility

High visibility sensitivity labels appear more urgent and appear in more places. For example, a high visibility sensitivity label on a table generates a notification when anyone authors a view or opens a published view that depends on it.



The icon and badge for high visibility sensitivity labels is purple, while standard visibility ones are gray. By default, the built-in sensitivity label called "Sensitive data" is high visibility.

If more than one sensitivity label applies to an asset (because the asset has more than one sensitivity label or is inheriting upstream ones), the badge includes a number, and the color is determined by the highest visibility label. For example, if two sensitivity labels apply to an asset, one of which is standard visibility and one of which is high visibility, the badge is purple.

Sensitivity labels in email subscriptions

Administrators can turn on sensitivity labels in email subscriptions so that when users subscribe to a view, the email they get includes high visibility sensitivity labels associated with that view. Emails with high visibility sensitivity labels contain:

- Links to relevant views or workbooks with their **Data Details** pane open.
- Links to relevant upstream assets, such as data sources, tables, or databases.

Administrators can turn on sensitivity labels in email subscriptions by selecting the **High-Visibility Data Labels in View and Workbook Subscriptions** option (previously the **Data Quality Warnings in Subscriptions** option) on the Tableau Server or Tableau Cloud site settings page. For more information, see High-Visibility Data Labels in View and Workbook Subscriptions in the Site Settings Reference.

Who can set sensitivity labels

To set a sensitivity label, you must either

- be a server or site administrator, or
- have the **Overwrite** capability for the asset.

Customize sensitivity labels

There's only one built-in sensitivity label: *Sensitive data*. Starting with Tableau Cloud June 2023 and Tableau Server 2023, using the labels manager on the **Data Labels** page or the REST API, an administrator can create sensitivity labels or change the name and description of an existing ones. Typical additions (name and description) might be:

- **Public**: Available to the public to view.
- **Internal**: Restricted to company employees and contractors. This data must not be shared publicly, but it can be shared with customers, partners, and others under a non-disclosure agreement (NDA).
- **Confidential**: Available to an approved group of employees and contractors. This data isn’t restricted by law, regulation, or a company master service agreement (MSA). It can be shared with customers, partners, and others under an NDA.
- **Restricted**: Available only to an approved group of employees and contractors. This data is likely restricted by law, regulation, an NDA, or a company MSA.
- **MissionCritical**: Available only to a small group of approved employees and contractors. Third parties who are given access could be subject to heightened contractual requirements. This data is almost always restricted by law, regulation, an NDA, or a company MSA.

For more information, see Manage Data Labels.

Labels with Custom Categories

Tableau offers several data labels – certification, quality warnings, and sensitivity labels – that cover a wide variety of ways to classify data. Still, there may be times that users need other labels and categories that match other use cases. Starting in Tableau Cloud October 2023 and Tableau Server 2023.3, users can classify assets using labels with custom categories that an administrator has defined. For example, an administrator could create a category called "Department" with labels for the sales, service, and operations departments ready to be applied to assets.

Labels with custom categories require a Data Management license with Tableau Catalog enabled, and can be attached to the same kinds of assets that other data labels can. However, labels with custom categories don't show on downstream assets the way that data quality warnings and sensitivity labels do.

> Note: If you're an administrator who wants to create custom categories and labels, see Manage Data Labels.

Attach labels with custom categories to an asset

> Note: Starting in Tableau Cloud February 2024 and Tableau Server 2024.2, you add and remove labels with custom categories using the consolidated Data Labels dialog instead of separate dialogs for each type of label. For information on the Data Labels dialog, see The Data Labels dialog.

To attach a label with a custom category to an asset:

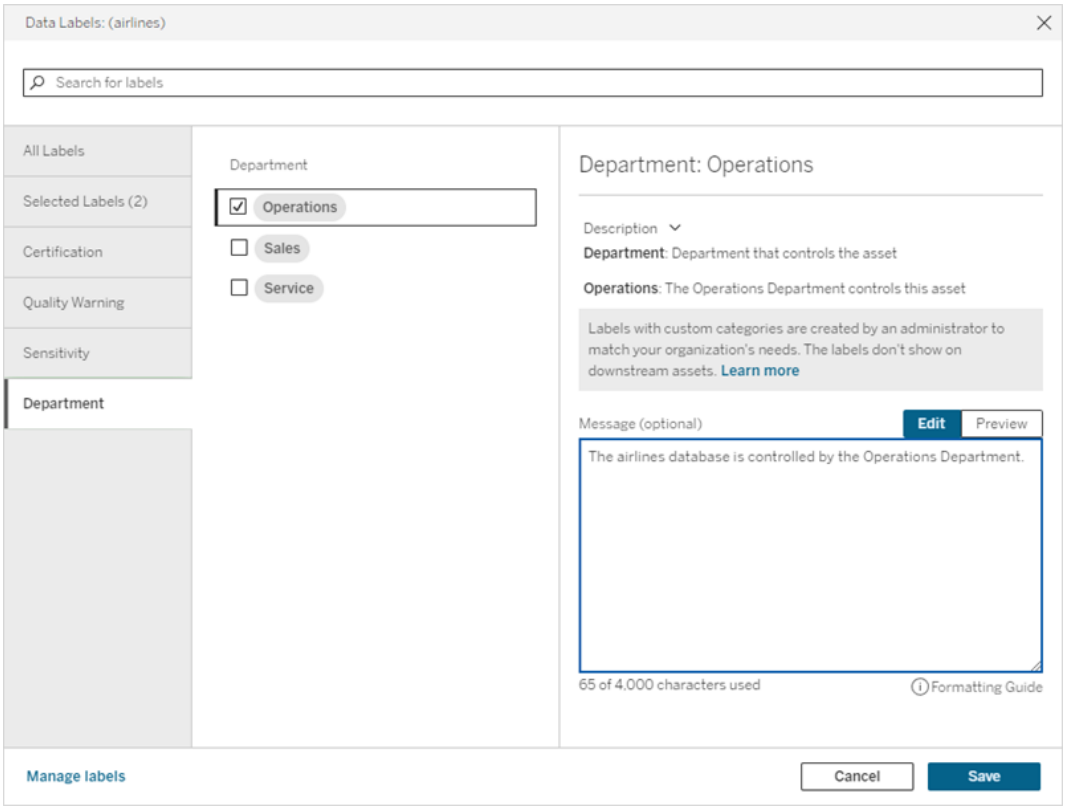
In Tableau Cloud and Tableau Server 2024.2 and later

1. Search for or navigate to the asset. The steps to navigate depend on the type of asset you want to add the label to:
    - Data source or virtual connection - on the **Explore** page, select **All Data Sources** or **All Virtual Connections**.

- Virtual connection table - on the **Explore** page, select **All Virtual Connections**, and select the virtual connection that contains the virtual connection table you want to certify. Then select the virtual connection table.
- Database or table - on the **Explore** page, navigate to the database or table. Or on the **External Assets** page, select **Databases and Files** or **Tables and Objects**.
- Column - on the **Explore** page, navigate to the table. Or on the **External Assets** page, select **Tables and Objects** and navigate to the table. Then find the column in the list.

2. Select the actions menu (...) next to the asset, and then select **Data Labels** > **All Data Labels**.
3. Select the vertical tab on the left side of the dialog that corresponds to the custom label category. Optionally, if you know the name of a label, you can search for it at the top of the dialog.
4. Select the checkbox beside labels you want attached to the asset.
5. If desired, enter a message to display to users. You can format the text in a message with bold, underline, and italics, and include a link or an image. To see text formatting tips, hover over the information (i) icon above the **Save** button.
6. Repeat steps 3 through 5 for each label you want to add.

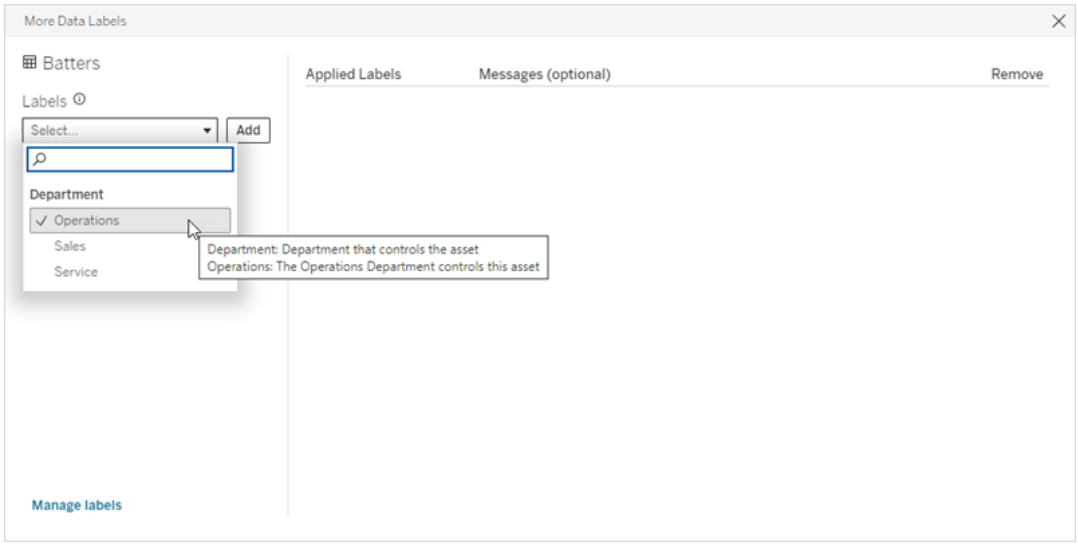
7. Select **Save**.



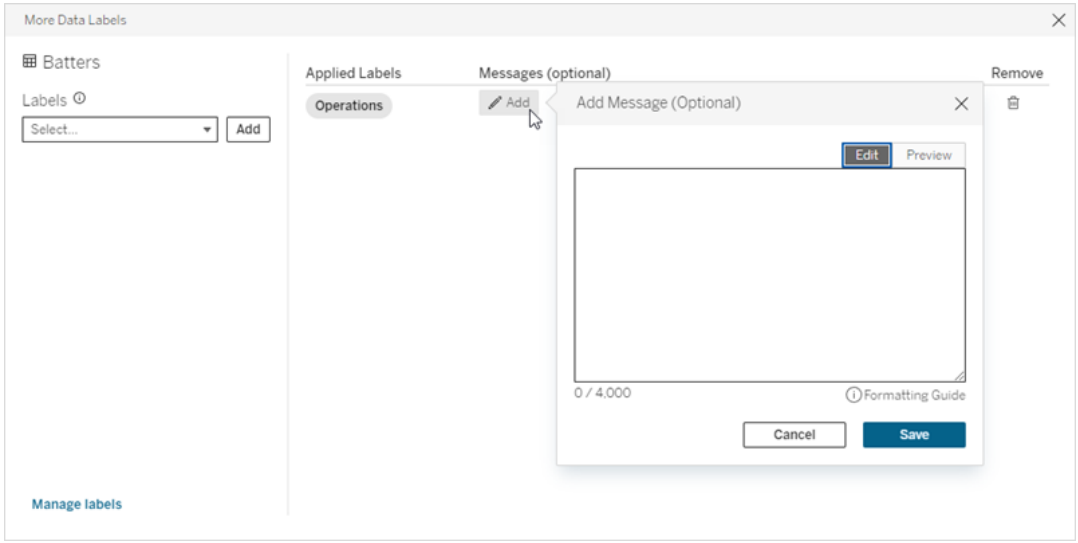
In Tableau Server 2023.3

1. Select the actions menu (...) next to the asset, and then select **Data Labels** > **More Data Labels**.

2. Select the **Labels** dropdown. The label names are arranged by label category, and you can scroll through them.



3. Select the label to attach, then select **Add** next to the **Labels** dropdown. The label appears in the **Applied Labels** list on the right side of the dialog.
4. To add an optional label message on this specific asset, select the pencil in the **Messages (Optional)** column, then select **Save**.



5. Repeat steps 2 through 5 for each label you want to add.
6. When you're finished adding labels, close the dialog. (Select the X in the dialog box's upper right corner or select something outside of the dialog box to close it.)

Remove labels with custom categories from an asset

> Note: Starting in Tableau Cloud February 2024 and Tableau Server 2024.2, you add and remove labels with custom categories using the consolidated Data Labels dialog instead of separate dialogs for each type of label. For information on the Data Labels dialog, see The Data Labels dialog.

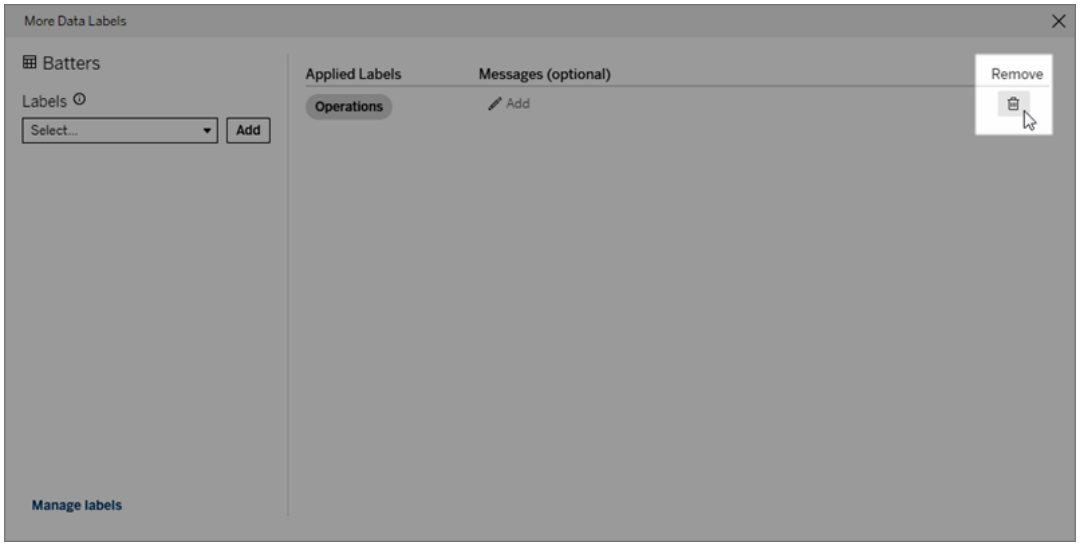
To remove a label with a custom category from an asset:

In Tableau Cloud and Tableau Server 2024.2 and later.

1. Search for or navigate to the asset. The steps to navigate depend on the type of asset you want to add the label to:
    - Data source or virtual connection - on the **Explore** page, select **All Data Sources** or **All Virtual Connections**.
    - Virtual connection table - on the **Explore** page, select **All Virtual Connections**, and select the virtual connection that contains the virtual connection table you want to certify. Then select the virtual connection table.
    - Database or table - on the **Explore** page, navigate to the database or table. Or on the **External Assets** page, select **Databases and Files** or **Tables and Objects**.
    - Column - on the **Explore** page, navigate to the table. Or on the **External Assets** page, select **Tables and Objects** and navigate to the table. Then find the column in the list.
2. Select the actions menu (...) next to the asset, and then select **Data Labels** > **All Data Labels**.
3. Select the vertical tab on the left side of the dialog that corresponds to the custom label category. Optionally, use the **Selected Labels** vertical tab to see all the labels attached to the asset. Or, if you know the name of a label, you can search for it at the top of the dialog.
4. Deselect the checkbox beside labels you want removed from the asset.
5. Repeat steps 3 and 4 for each label you want to remove.
6. Select **Save**.

In Tableau Server 2023.3

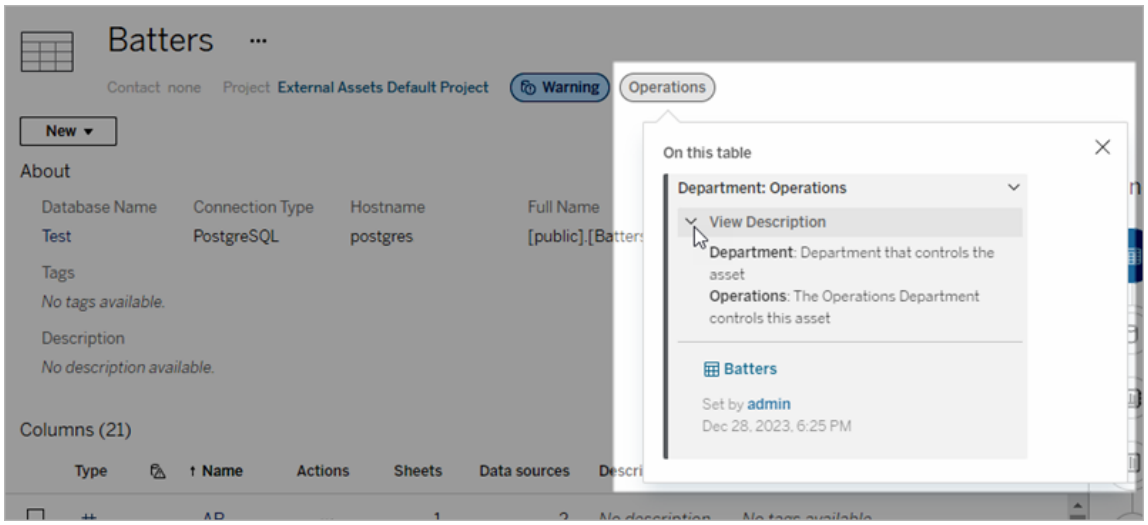
1. Select the actions menu (...) next to the asset, and then select **Data Labels** > **More Data Labels**.
2. In the **Applied Labels** section of the dialog, select the trash icon in the row for the label to remove.



3. Repeat step 2 for each label you want to remove.
4. When you're finished removing labels, close the dialog. (Select the X in the dialog box's upper right corner or select something outside of the dialog box to close it.)

Where labels with custom categories appear

Custom labels appear on assets when navigating Tableau Cloud and Tableau Server.



In web authoring, you can select a data source or a column and then select **Catalog Details** to see all its labels.

Unlike quality warnings and sensitivity labels, labels with custom categories don't appear downstream from assets they're attached to. For example, suppose your organization has a custom label category named "Department" to which a custom label named "Sales" belongs. If you attach the "Sales" label to a table called "Orders", the label only appears on the "Orders" table and not on workbooks downstream from it.

Who can add custom category labels

To add a label with a custom category to an asset (or to remove one from an asset), you must either

- be a server or site administrator, or
- have the Overwrite capability for the asset.

Customize a label with a custom category

For information on how administrators can create or edit custom categories and labels that appear in the **More Data Labels** dialog, see Manage Data Labels.

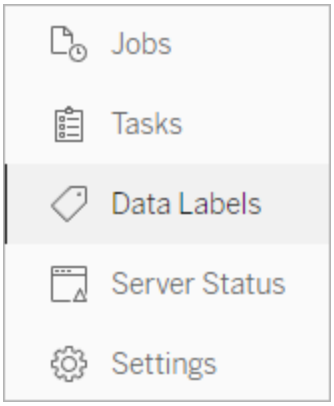
Manage Data Labels

Starting in Tableau Cloud October 2023 and Tableau Server 2023.3, if you have a Data Management license and are an administrator, you can use the label manager to create or edit label names and label categories. These customizations affect the way that labels appear throughout Tableau when users interact with labels.

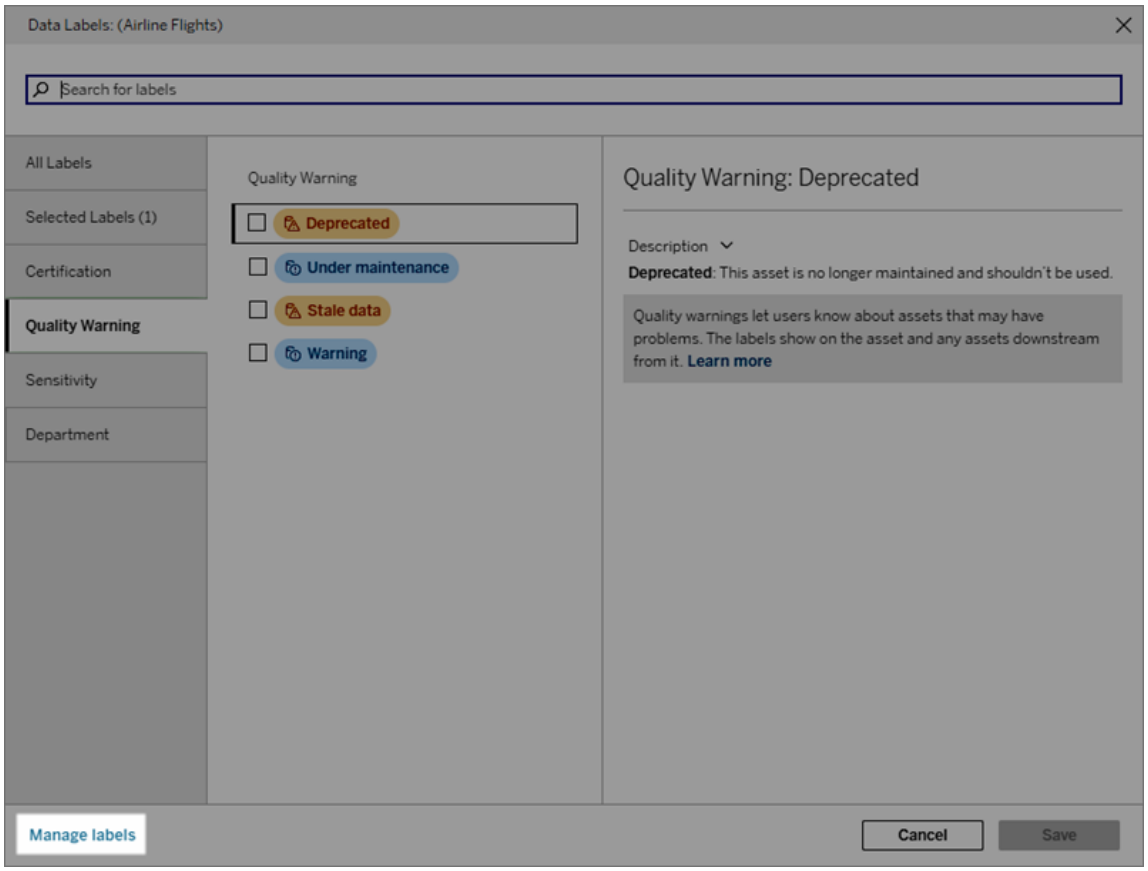
Note: You can also use the REST API's labelValues methods and labelCategories methods to create and edit labels and label categories. (Tableau Cloud administrators have been able to create and modify label names and descriptions using the labelValues methods since Tableau Cloud June 2023.) For more information, see the Metadata Methods in the REST API Reference.

Label Manager

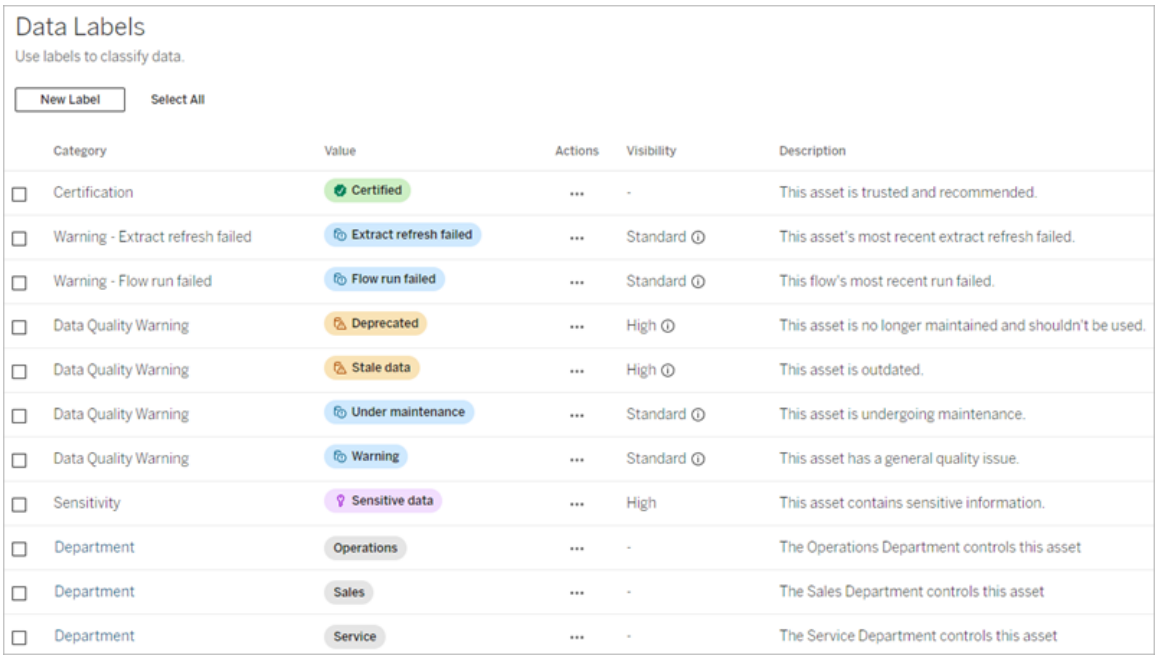
To use the label manager, log in as an administrator and select **Data Labels** from the left navigation pane.



Alternatively, if you're logged in as an administrator and you open the **Data Labels** dialog to label an asset, a **Manage labels** link shows in the lower-left corner. The **Manage labels** link leads to the **Data Labels** page. (In Tableau Server 2023.3 and earlier, the **Manage labels** link shows in the label selection dropdowns of the individual certification, data quality warning, sensitivity label, and custom label dialogs instead.)



The label manager page shows a row for each label, sorted by label category. Each row includes the label category, name (here known as the value), an **Actions** menu (...) to perform actions on that label, visibility, and description.



Use the label manager to:

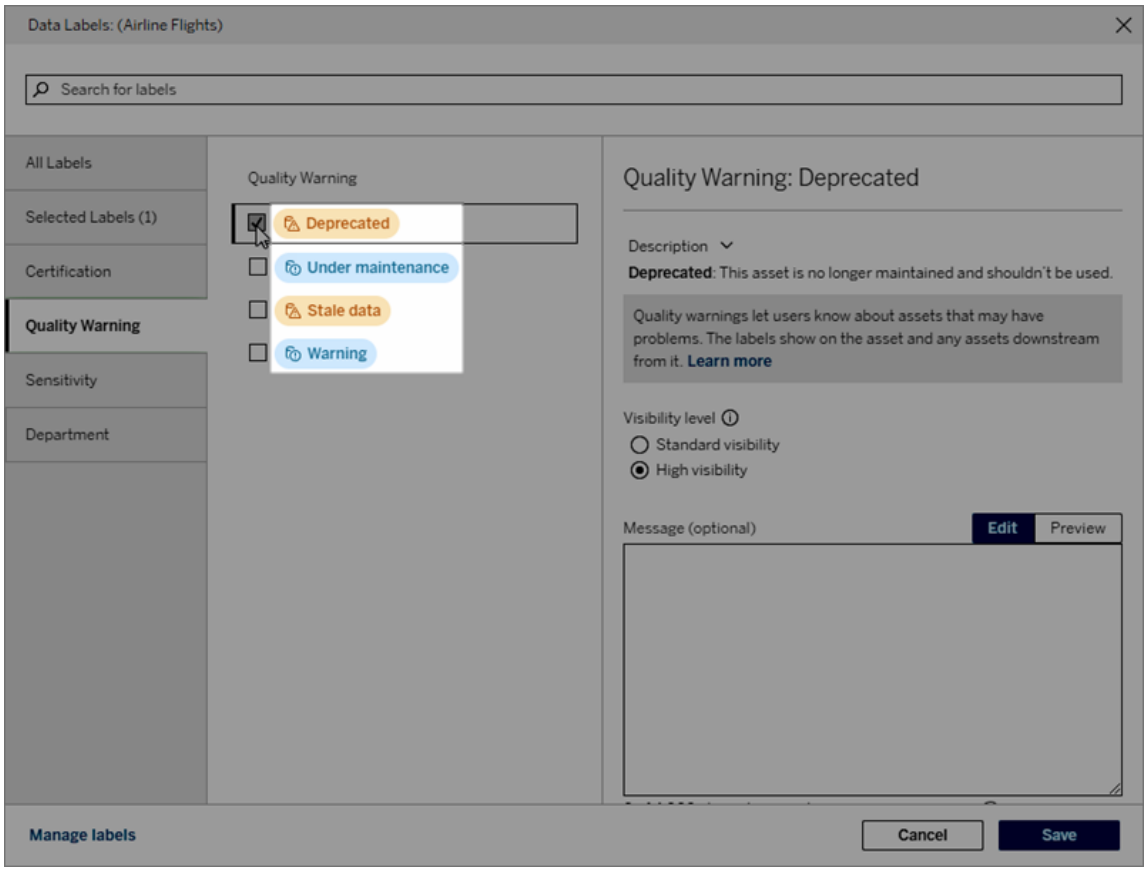
- **Edit built-in labels**
  - Example: You change the name of the built-in "Under maintenance" quality warning to "Maintenance mode".
  - Example: You change the visibility of extract refresh monitoring labels from standard visibility to high visibility so that they show in views.
- **Create new labels for the existing, built-in categories**
  - Example: You add a new sensitivity label called "Confidential".
- **Revert a built-in label to its default name, description, and visibility**
  - Example: You previously changed the "Stale data" quality warning name to "Outdated", and you want to revert it to its default name.
- **Create custom categories**
  - Example: You create a new label category called "Department" with the intention of adding labels for different business units.
- **Create new labels in custom categories**
  - Example: You create new "Sales", "Service", and "Operations" labels for your newly created "Department" category.

Properties of Data Labels

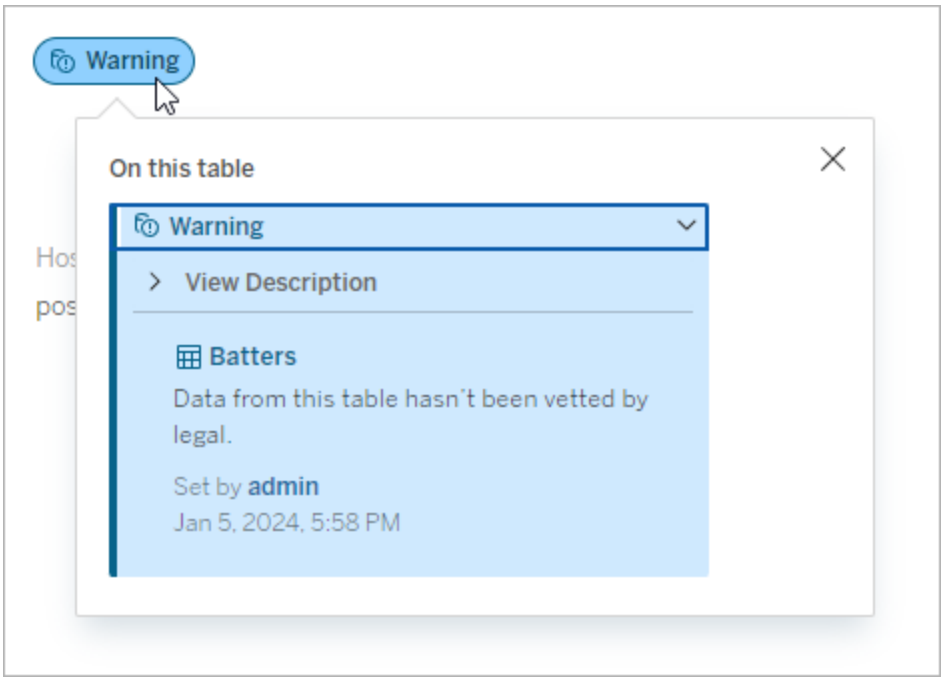
A label has a name, a category, and a description. Labels with a Quality Warning or Sensitivity category also have a visibility level.

Name

The label name is the common name for the label as it appears in various places. For example, here the label name "Deprecated" is selected in the **Quality Warning** tab of the **Data Labels** dialog.



Here the label name "Warning" shows at the top of the "Batters" table page, and again in the label details.

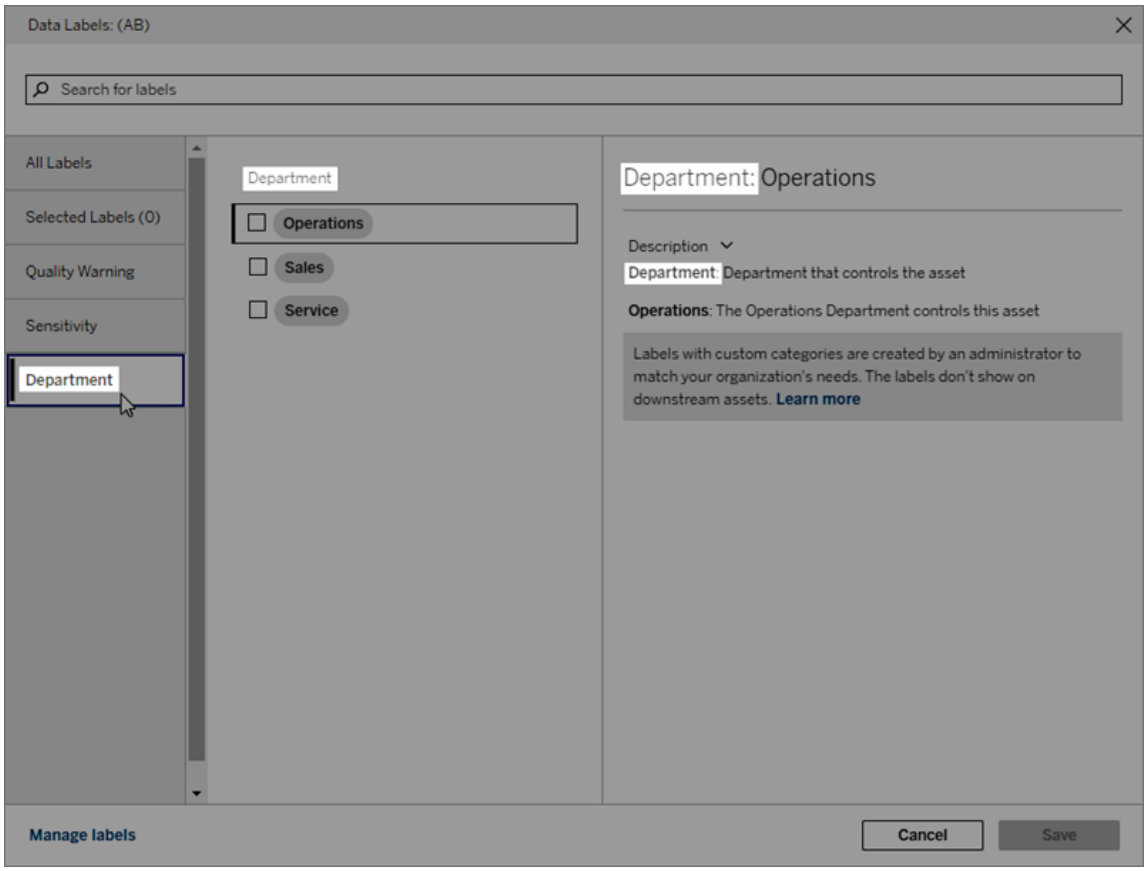


Category

The label category affects where and how the label appears, whether it appears on assets that are downstream from the one it's attached to, and which parts are customizable, among other things. For example, quality warnings and sensitivity labels appear on downstream assets, but other labels with other categories don't. Another example: You can change the description of a certification label, but not the name.

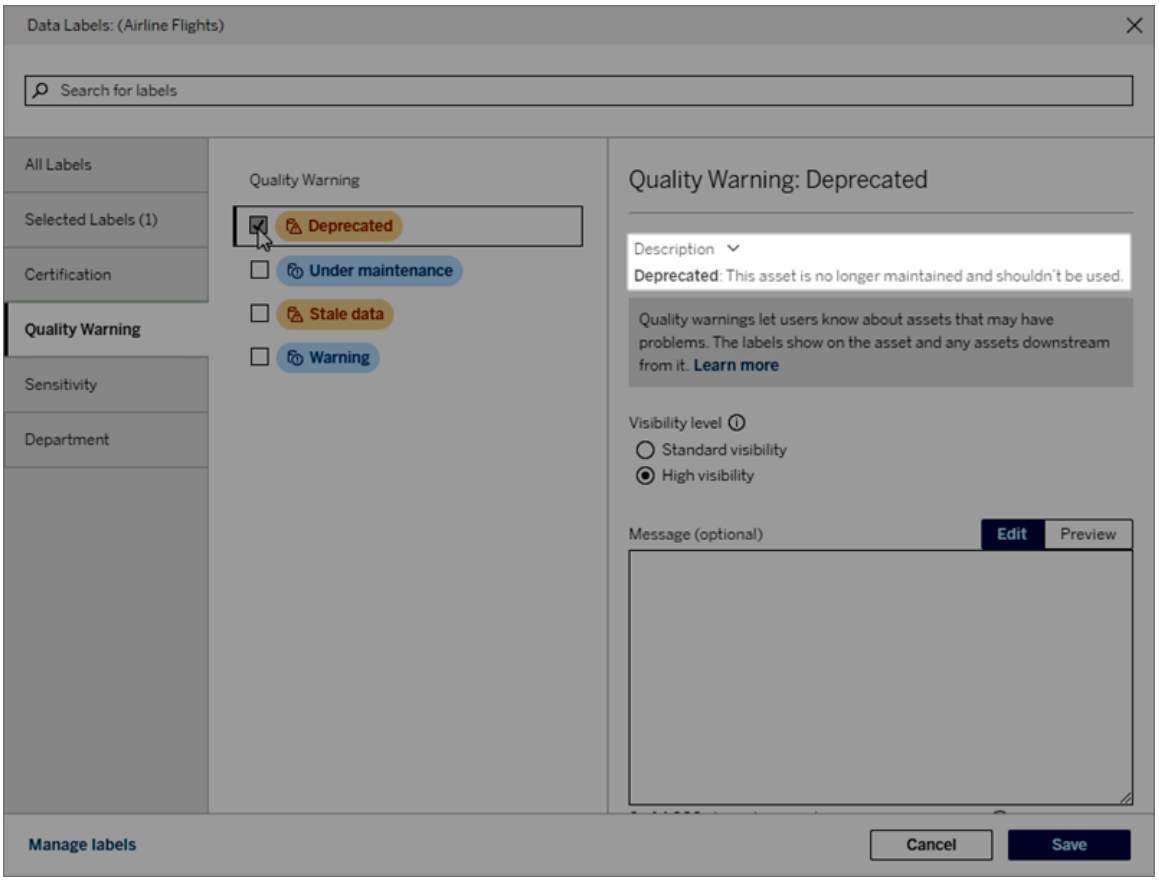
The built-in categories are **certification**, **quality warning**, and **sensitivity**.

For custom categories, users see the category name in the vertical category tabs of the **Data Labels** dialog, among other places. For example, here the category name "Department" appears in the vertical category tabs, the top of the label name list, and other places.



Description

The label description appears in various places, including in the **Data Labels** dialog, and helps the user understand what the label is used for. For example, the label description for this quality warning says "This asset is no longer maintained and shouldn't be used."



Visibility

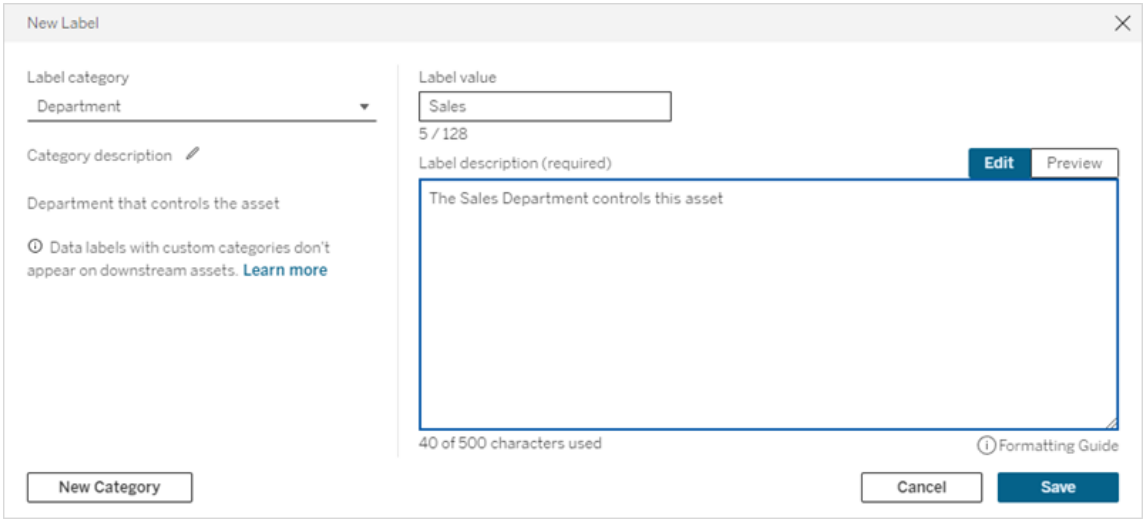
The visibility of a label determines its appearance. High-visibility labels appear in more places and may appear more urgent to the user. You can only set the visibility level on Quality Warning or Sensitivity labels. Furthermore, if a label has a category of Quality Warning, users with permissions can override the default visibility on each asset that they attach a Quality Warning label to. For more information, see the Set a Data Quality Warning and Sensitivity Labels topics.

Create a data label

To create a label:

1. From the **Data Labels** page, select **New Label**.
2. Select a category from the **Label** category dropdown.
3. Enter the label name in the **Label value** field.

4. Enter the label description in the **Label description** field. You can format the text with bold, underline, and italics, and include a link or an image. To see text formatting tips, hover over the information (i) icon above the **Save** button.
5. If the label has a category of Quality Warning or Sensitivity, set the visibility level. For more information, see Visibility.
6. Select **Save**.



Limitations for creating labels

- You can't create a label in the certification category. The certification category allows only the single, built-in label.
- You can't create new monitoring warnings. However, the extract refresh failure warning and flow run failure warning can be edited in limited ways, as described in the "Edit a label" section.
- The maximum length for a label name is 128 characters in Tableau Cloud and Tableau Server 2024.2 and later. The maximum length for a label name is 24 characters in Tableau Server 2023.3 and earlier.
- The maximum length for a label description is 500 characters.

Edit a data label

To edit an existing label:

1. From the **Data Labels** page, select the **Actions** (...) menu in the label's row. Or select the row using its checkbox on the left and click the **Actions** dropdown at the top of the

label list.
2. Select **Edit**.
3. (Optional) Change the label name using the **Label value** field.
4. (Optional) Change the label description using the **Label description** field.
5. (Optional) If the label has a category of Quality Warning or Sensitivity, set the visibility level. For more information, see Visibility.
6. Select **Save**.

Limitations for editing labels

- You can't change the category on an existing label.
- The maximum length for a label name is 128 characters in Tableau Cloud and Tableau Server 2024.2 and later. The maximum length for a label name is 24 characters in Tableau Server 2023.3 and earlier.
- The maximum length for a label description is 500 characters.

The different label categories allow different degrees of label editing. The following table lists the editable properties of labels with the given categories:

Label category	Can edit label category	Can edit label names	Can edit label descriptions	Can edit label visibility
Certification	No	No	Yes	N/A
Quality Warning	No	Yes[1]	Yes	Yes[2]
Sensitivity	No	Yes	Yes	Yes
Custom	No	Yes	Yes	N/A

[1] *You can't edit the label name (label value) of the extract refresh or flow run monitoring warnings.*

[2] *The visibility level you set for quality warnings is the default visibility. Users with permission can override the default visibility when they attach a quality warning to an asset. For more information, see Visibility.*

Delete a data label

To delete an existing label:

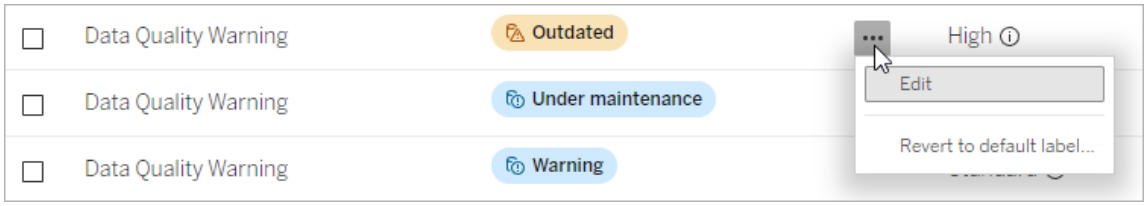
1. From the **Data Labels** page, select the **Actions** (...) menu in the label's row. Or select the row using its checkbox on the left and click the **Actions** dropdown at the top of the label list.
2. Select **Delete**.

Limitations for deleting labels

- You can't delete a built-in label. Built-in labels are the default labels in Tableau Server.

Revert a built-in data label to its defaults

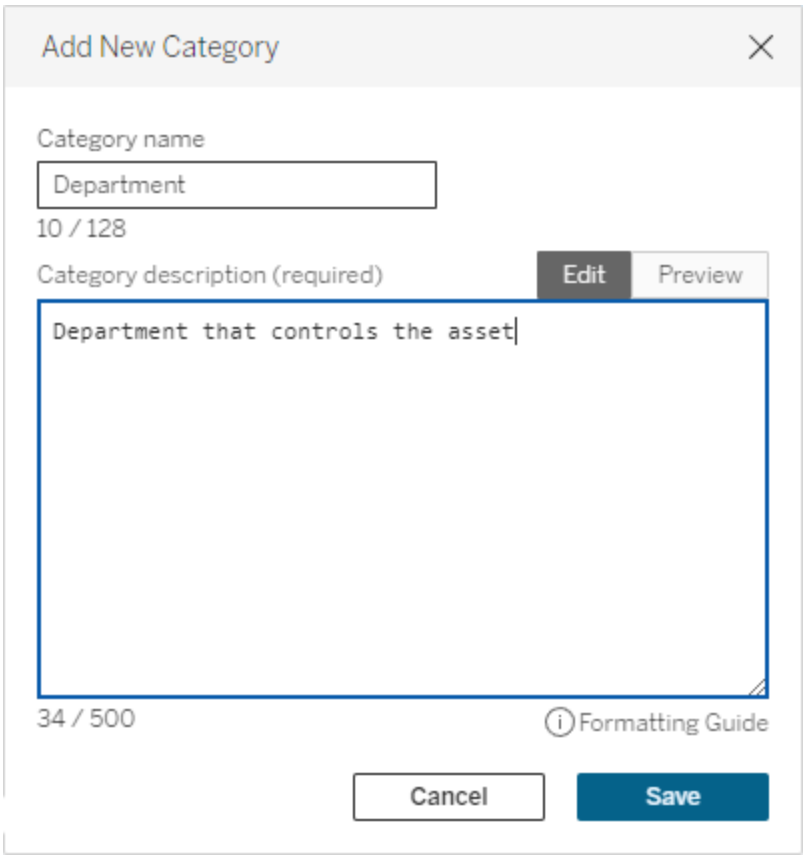
If an administrator previously edited a built-in label, the **Actions** menu for that label contains **Revert to default label**. Reverting a label to its defaults returns the label's name (value), description, and visibility to its built-in defaults.



Create a data label category

To create a label category:

1. From the **Data Labels** page, select **New Label**.
2. In the **New Label** dialog, select **New Category**.
3. Enter the category name in the **Category name** field.
4. Enter the category description in the **Category description** field. You can format the text with bold, underline, and italics, and include a link or an image. To see text formatting tips, hover over the information (i) icon above the **Save** button.
5. Select **Save**.



Limitations for creating label categories

- The maximum length for a category name is 128 characters in Tableau Cloud and Tableau Server 2024.2 and later. The maximum length for a category name is 24 characters in Tableau Server 2023.3 and earlier.
- The maximum length for a category description is 500 characters.

Edit a data label category

To edit a label category:

1. From the **Data Labels** page, select the label category, then select the pencil icon. Or from the **New Label** or **Edit Label** dialogs, select the category in the **Label category** dropdown and then select the pencil icon next to **Category description**.
2. (Optional) Change the category name using the **Category name** field.
3. (Optional) Change the category description using the **Category description** field.
4. Select **Save**.

Limitations for editing label categories

- You can't edit a built-in category.
- The maximum length for a category name is 128 characters in Tableau Cloud and Tableau Server 2024.2 and later. The maximum length for a category name is 24 characters in Tableau Server 2023.3 and earlier.
- The maximum length for a category description is 500 characters.

Delete a data label category

Currently, there isn't a method to delete a label category through the regular Tableau Server interface. To delete a category using the REST API, see the Delete Label Category method in the REST API Reference.

Scenarios for customization

Scenario: Customize a built-in data label

Suppose you decide that the data quality warning called "Warning" could be more specific. As an administrator, you change the label name from the default ("Warning") to something you think is more useful to your organization: "Not approved". The label name "Not approved" now appears in label dialogs when users are selecting labels.

Alternatively, you could change the label description so that the user learns more about the warning in the label dialog. For example: "This asset doesn't meet quality standards required by the marketing department."

Scenario: Create a custom data label

Suppose you want users to have more granular control over classifying the sensitivity of assets. You create two sensitivity labels with the names "Public" and "PII". The custom label names "Public" and "PII" now appear in the label dialog dropdowns and descriptions, alongside the built-in sensitivity label.

Scenario: Create a new data label category and associated data labels

Suppose you need a way to identify the business units that are responsible for assets. You create a label category called "Department". Then you create three labels – "Sales",

"Service", and "Operations" – with "Department" as their category. The category "Department" and the three associated labels now appear in the **More Data Labels** dialog for users to attach to assets.

Manage Dashboard and Viz Extensions in Tableau Server

Dashboard extensions are web applications that run in custom dashboard zones and can interact with the rest of the dashboard using the Tableau Extensions API. Dashboard extensions give users the ability to interact with data from other applications directly in Tableau. Like dashboard extensions, viz extensions are web applications that use the Tableau Extensions API and allow developers to create new viz types. Tableau users can access viz extensions through the worksheet Marks card.

> **Note:** You must be a server administrator to enable dashboard and viz extensions on the server, or to block specific extensions from running. You must be a server administrator to add extensions to the safe list and to control the type of data the extensions can access. The server administrator can also configure whether users on the site see prompts when they add or view extensions. For information about extension security and recommended deployment options, see Extension Security - Best Practices for Deployment

For information about using dashboard extensions in Tableau, see Use Dashboard Extensions.

For information about using viz extensions, see Add Viz Extensions to Your Worksheet.

> Looking for Tableau Cloud? See Manage Dashboard Extensions in Tableau Cloud.

Before you run extensions on Tableau Server

Tableau supports two ways of hosting extensions:

- Network-enabled extensions, which are hosted on web servers located inside or outside of your local network. Network-enabled extensions have full access to the web.
- Sandboxed extensions, which run in a protected environment without access to any other resource or service on the web.

Note: Beginning with version 2021.1.0 Tableau supports integration with Einstein Discovery through the Einstein Discovery Dashboard extension. This is a special extension that has access to data in Salesforce.com and is allowed by default. It is not considered a Network-enabled extension or a Sandboxed extension. For more information on Einstein Discovery integration, see Tableau Server Release Notes.

Sandboxed extensions are hosted by Tableau and employ W3C standards, such as Content Security Policy (CSP), to ensure the extension can't make network calls outside of the hosting Tableau Server. A Sandboxed extension can query data in the dashboard, but it can't send that data anywhere outside of the sandbox. Sandboxed Extensions are supported in Tableau 2019.4 and later. By default, Sandboxed extensions are allowed to run if extensions are enabled for the site.

Network-enabled extensions are web applications and could be running on any computer set up as a web server. This includes local computers, computers in your domain, and third-party web sites. Because Network-enabled extensions could be hosted on third-party sites and could have access to the data in the workbook, you want to only allow the extensions you trust. See Test Network-enabled extensions for security.

For security, you can use the settings for extensions on Tableau Server to control and limit the extensions that are allowed to run.

- By default, Sandboxed extensions are allowed to run if extensions are enabled for the site.
- By default, no Network-enabled extensions are allowed unless they've been explicitly added to the safe list.

- By default, only extensions that use the HTTPS protocol are allowed, which guarantees an encrypted channel for sending and receiving data (the only exception is for `http://-localhost`).
- If the Network-enabled extension requires full data (access to the underlying data) the extension can't run on Tableau Server unless you explicitly add the extension to the safe list and grant the extension access to full data.

Control extensions and access to data

Server administrators can control a global setting to allow extensions for all sites on the server. Server administrators can also put extensions, including Sandboxed extensions, on a global block list to prevent them from running (see Block specific extensions). By default, all Sandboxed extensions are enabled on the server, but site administrators can choose to override the default and prohibit Sandboxed extensions for the site.

Change the global setting enabling extensions on the server

1. To change this setting for the server, go to **Manage All Sites > Settings > Extensions**. If the server just has a single site, the global controls appear on the settings page for the site.
2. Under Dashboard and Viz Extensions, select or clear the **Let users run extensions on this server** checkbox. If this option is not selected, extensions are not allowed to run. This global setting overrides the **Let users run extensions on this site** settings for each site.

Change the default settings for a site

Server administrators can control whether to enable extensions for the site and whether to allow Sandboxed extensions on the site. That is, if extensions are enabled on the server, the default site settings allow Sandboxed extensions to run on the site, provided the extension is not specifically blocked on the server. The default site settings allow Network-enabled extensions to run that appear on the safe list for the site. Individual Sandboxed extensions can also be added to the safe list, if Sandboxed extensions are not allowed by default.

1. To change these settings for the site, go to **Settings > Extensions**.

2. Under Dashboard and Viz Extensions, configure these options:

    - **Let users run extensions on this site**
    - **Let Sandboxed extensions run unless they are specifically blocked by a server administrator**

Server administrators can add or remove Network-enabled and Sandboxed extensions from the safe list for a site. When you add an extension to the safe list, you can control whether to allow the extension to have access to full data. See Add extensions to the safe list and configure user prompts.

Identifying the URL of an extension

As a web application, an extension is associated with a URL. You can use this URL to test and verify the extension. You also use the URL to add the extension to the safe list to allow full data access, or to the block list to prohibit any access.

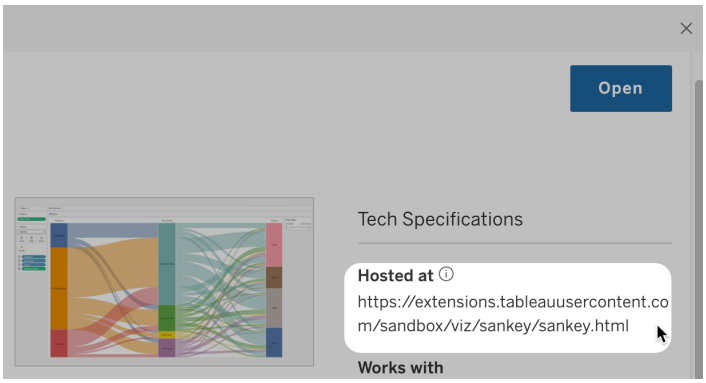
From the manifest file

If you have the extension manifest file (`.trex`), an XML file that defines properties for the extension, you can find the URL from the `<source-location>` element.

```
<source-location>
    <url>https://www.example.com/myExtension.html</url>
</source-location>
```

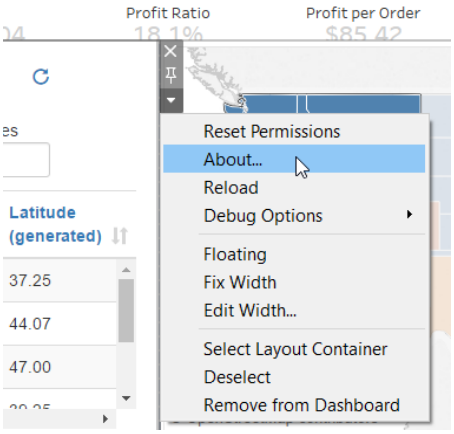
From Tableau Exchange

If you added or downloaded an extension from the Tableau Exchange, you can find the URL for the extension on the Exchange. Open the tile for the extension, under Tech Specifications, look for the URL under the heading, **Hosted at**.

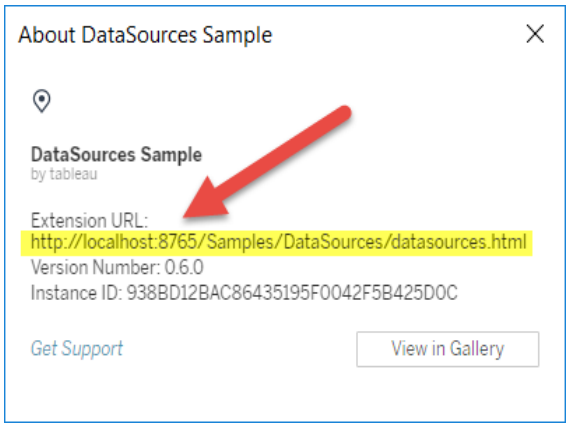


Identifying a dashboard extension using the About dialog box

If you have added the extension to the dashboard, you can find the URL from the extension properties. From the **More Options** menu, click **About**.



The About dialog box lists the name of the extension, the author, the web site of the author, and the URL of the extension.



Add extensions to the safe list and configure user prompts

To ensure that users can use Network-enabled extensions that are trusted, you can add them to the safe list for the site. You can also add Sandboxed extensions to the safe list, if Sandboxed extensions aren't enabled by default on the site.

On the safe list, you can control whether to grant the extension full data access. By default, when you add an extension to the safe list, the extension only has access to the summary (or aggregated) data. You can also control whether users see a prompt asking them to allow the extension access to data. You might want to add an extension to the safe list (for example, a Sandboxed extension) so that you can configure whether users see the prompts. When you hide the prompt from users, the extension can run immediately.

1. Go to **Settings > Extensions**.
2. Under **Enable Specific Extensions**, add the URL of the extension. See Identifying the URL of an extension.

   Tip: You can use a period and asterisk (`.*`) as a wildcard in the URL to allow all extensions in a certain domain or location. For example, to allow all extensions in the domain under example.com that use port 8080, you would add the URL: `https://example.com:8080/.*` . For more information, see Using regular expressions in the safe list URL.
3. Choose to **Allow** or **Deny** the extension **Full Data Access**.

Full data access is access to the underlying data in the view, not just the summary or aggregated data. Full data access also includes information about the data sources, such as the names of the connection, fields, and tables. Usually, if you are adding an extension you want to use to the safe list, you also want to allow the extension to have access to full data, if the extension requires it. Before adding extensions to the safe list, be sure to Test Network-enabled extensions for security.

4. Choose to **Show** or **Hide** the **User Prompts**.

   Users see the prompts by default when they are adding a dashboard extension to a dashboard, or a viz extension to a worksheet, or when they are interacting with a view that has an extension. The prompt tells users details about the extension and whether the extension has access to full data. The prompt gives users the ability to allow or deny the extension from running. You can hide this prompt from users, allowing the extension to run immediately.

Block specific extensions

The default global policy allows all Sandboxed extensions and those Network-enabled extensions that appear on the safe list for a site. Server administrators can keep specific extensions from running by adding them to the block list for the server. If an extension is on the global block list it overrides any settings for the extension on the safe list for a site.

1. To add an extension to the blocked list for the server, go to **Manage All Sites > Settings > Extensions.** On single-site installations, the block list is on the site **Extensions** settings page.

2. Under **Block Specific Extensions**, add the URL of the extension. See Identifying the URL of an extension.

Using regular expressions in the safe list URL

In general, when you add an extension to the safe list, you should use the specific URL of the extension. However, there are times when you might want to allow multiple extensions that are

hosted from the same domain and location. In this case it is convenient to use a wildcard in the URL. The extension settings support the use of regular expressions.

Regular expression	Description
.	A period (.) is a wildcard you can use to match any character. If you need to specify a period (.) in the URL instead of a wildcard, you can escape the character with a backslash (\ .).
*	An asterisk (*) is a quantifier that specifies one or more instances of the previous character.

Use care if you use wildcards so that you don't make the safe list too permissive, and inadvertently allow access to extensions that should not have access.

The following table shows some examples of using regular expressions in the URL. Note that these examples do not show the protocol and the full URL of the extension. Only extensions that use the HTTPS protocol are allowed (with the exception of http://localhost).

To specify...	Example	Specifies
Range of domains	.*\.example.com	All subdomains under example.com.
All ports	example.com:.*	Extensions are allowed access from all ports on example.com.
All extensions under domain, port, and path	example.com:8080/xyz/.*	All extensions under the domain example.com that use port 8080 and are located in zyz, are allowed access.
All ports for a range of domains	.*\.example.com:.*	Allows access to extensions on all ports on all subdomains under example.com.

All extensions under a domain and path that match the pattern	example.com/t.c/.*	Allows access to extensions running on example.com under folders that match the pattern t.c. For example, tic, tac, toc.

Test Network-enabled extensions for security

Dashboard and viz extensions are web applications that interact with data in Tableau using the Extensions API. Network-enabled extensions could be hosted on web servers inside or outside of your domain, and can make network calls and have access to resources on the Internet. Because of the potential vulnerabilities, such as cross-site scripting, you should test and vet Network-enabled extensions before they are used in Tableau Desktop, and before you allow the extensions on Tableau Server.

Examine the source files

Dashboard and viz extensions are web applications and include various HTML, CSS, and JavaScript files, and an XML manifest file (*.trex) that defines the properties in the extension. In many cases, the code for a extension is publicly available on GitHub and can be examined there or downloaded. In the manifest file (`*.trex`), you can find the source location, or URL indicated where the extension is hosted, the name of the author, and the web site of the author or company to contact for support. The `<source-location>` element specifies in the URL, the `<author>` element, specifies the name of the organization and the web site to contact for support (`website="SUPPORT_URL"`). The web site is the **Get Support** link user see in the **About** dialog box for the extension.

Many extensions reference external JavaScript libraries, such as the jQuery library or API libraries for third parties. Validate that the URL for external libraries points to a trusted location for the library. For example, if the connector references the jQuery library, make sure that the library is on a site that is considered standard and safe.

All extensions are required to use the HTTPS protocol (`https://`) for hosting their extensions. You should examine the source files for the extension to ensure that any reference to

external libraries is also using HTTPS or is hosted on the same web site as the extension. The one exception to the requirement of HTTPS is if the extension is hosted on the same computer as Tableau (`http://localhost`).

To the extent possible, make sure you understand what the code is doing. In particular, try to understand how the code is constructing requests to external sites, and what information is being sent in the request. In particular, check if any user-supplied data is validated to prevent cross-site scripting.

Understand data access

The Tableau Extensions API provides methods that can access the names of the active tables and fields in the data source, the summary descriptions of the data source connections, and the underlying data in a worksheet. If an extension uses any of these methods in a view, the extension developer must declare that the extension requires full data permission in the manifest file (`.trex`). The declaration looks like the following.

```
<permissions>
   <permission>full data</permission>
</permissions>
```

Tableau uses this declaration to provide a prompt to users at run time that gives them the option of allowing this access. If the extension uses any one of these methods, without declaring full-data permission in the manifest file, the extension loads but the method calls fail.

For information about how an extension accesses data from the dashboard, and the JavaScript methods used, see Accessing Underlying Data in the Tableau Extensions API. To get a better understanding about what the extension can find out about the data, you can use the DataSources sample dashboard extension (available from the Tableau Extensions API GitHub repository) to see what data is exposed when the `getDataSourcesAsync()` method is called.

Test the extension in an isolated environment

If possible, test the extension in an environment that is isolated from your production environment and from user computers. For example, add a dashboard or viz extension to a safe

list on a test computer or virtual machine that's running a version of Tableau Server that is not used for production.

Monitor traffic created by the dashboard extension

When you test a Network-enabled extension, use a tool like Fiddler, Charles HTTP proxy, or Wireshark to examine the requests and responses that the extension makes. Make sure that you understand what content the extension is requesting. Examine the traffic to be sure that the extension is not reading data or code that is not directly related to the purpose of the extension.

Configure Connections with Analytics Extensions

Tableau supports a set of functions that your users can use to pass expressions to analytics extensions for integration with R, Python, and Einstein Discovery.

> **Note**: You can use R and Python scripts to perform complex cleaning operations in your Tableau Prep flows, but configuration and functionality supported can be different. For information see Use R and Python Scripts in your Flow in the Tableau Prep help.

This topic describes how to configure sites on Tableau Server with analytics extensions.

Because Tableau Server provides an authentication mechanism, it can be more secure to expose analytics extensions functionality to users through Tableau Server than in Tableau Desktop.

For more information about user scenarios and configuring Tableau Desktop, see Pass Expressions Analytics Extensions, in the *Tableau Desktop and Web Authoring Help*.

The configuration steps in this article are specific to Workbooks. For information about how you can use R and Python scripts to incorporate predictive modeling data into your flow, see Use R and Python scripts in your flow in the *Tableau Prep Help*.

Feature change history:

- 2021.2 — You can configure multiple analytics extension connections for each site. (You are limited to a single Einstein Discovery connection per site.)

  For information about how to determine analytics extension usage in workbooks, see Determining analytics extensions usage.
- 2021.1 — Einstein Discover is included as an analytics extension option. Einstein Discovery in Tableau is powered by salesforce.com. Consult your agreement with salesforce.com for applicable terms.
- 2020.2 — You can configure a different analytics extension connection for each site on your server. Prior to this change a single analytics extension configuration applied globally to all sites on the server.
- 2020.1 — This functionality is now called *analytics extensions*. Previously the feature was called "external services."

Server SSL

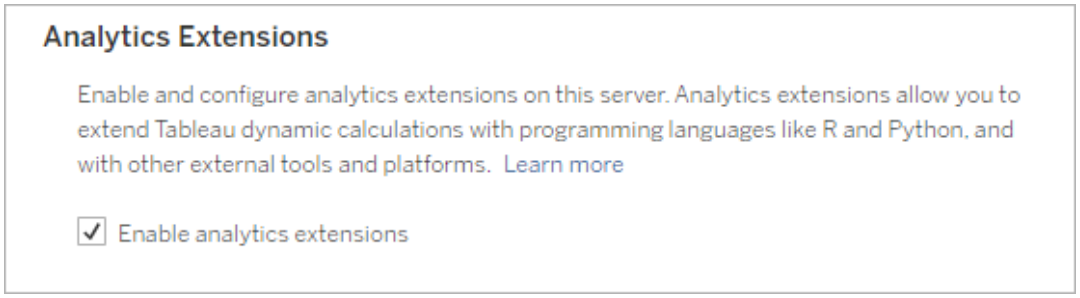
To configure SSL for analytics extensions, you must install a valid certificate on the computer running Tableau Server. The certificate must be trusted by the computer running Tableau Server. The certificate Subject field or one of the SAN entries on must exactly match the URI of the analytics extensions service configuration.

Enable analytics extensions

Before you configure extensions, you must enable analytics extensions server-wide.

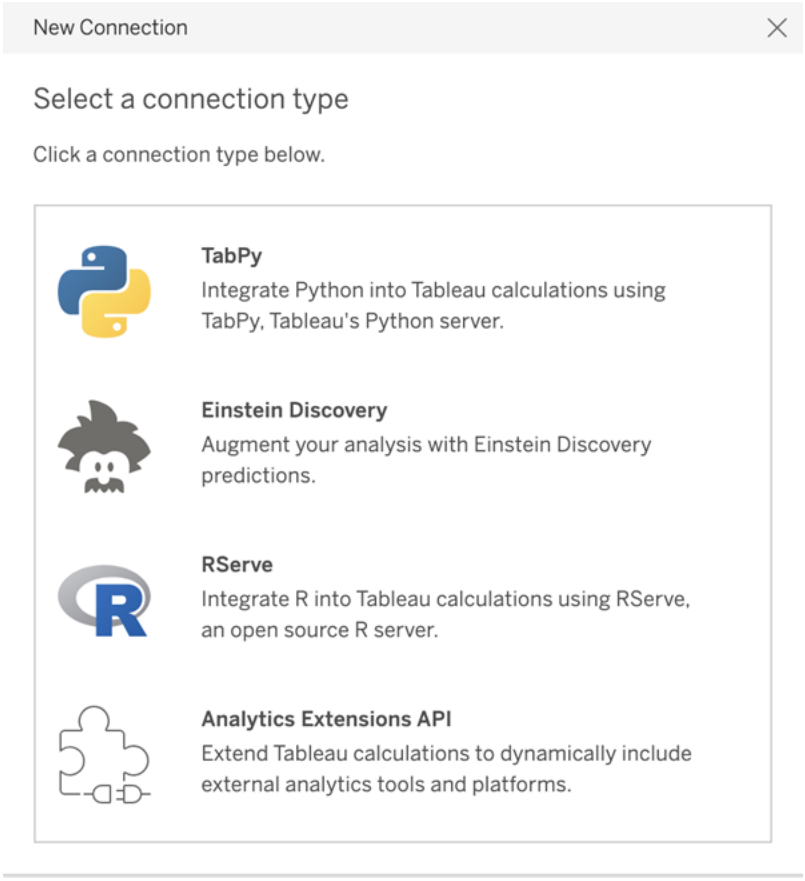
1. Sign in to the Tableau Server Admin Area.
   - If you only have a single site (default) on your server, click **Settings**, and then go to Step 2.
   - If you have multiple sites on your server:
     a. Under **All Sites**, click **Manage all sites**.
     b. Click the **Extensions** tab.
2. Scroll to **Analytics Extensions**, select **Enable analytics extensions**, and then click

Save.



Configure analytics extensions settings

1. Sign in to the Tableau Server Admin Area.
2. On the Settings page, click the **Extensions** tab and then scroll to **Analytics Extensions**. (On multi-site deployments of Tableau Server, navigate to the site where you want to configure analytics extensions, and then click **Settings**>**Extensions**.)
3. **Multi-site deployments only**: you must enable Analytics Extensions on each site. Under Analytics Extensions, select **Enable analytics extensions for site**.
4. Under Analytics Extensions, click **Create new connection**.
5. In the **New Connection** dialog, click the connection type you want to add, then enter the configuration settings for your analytics service:



6. The options you need to configure depend on the connection type you choose:

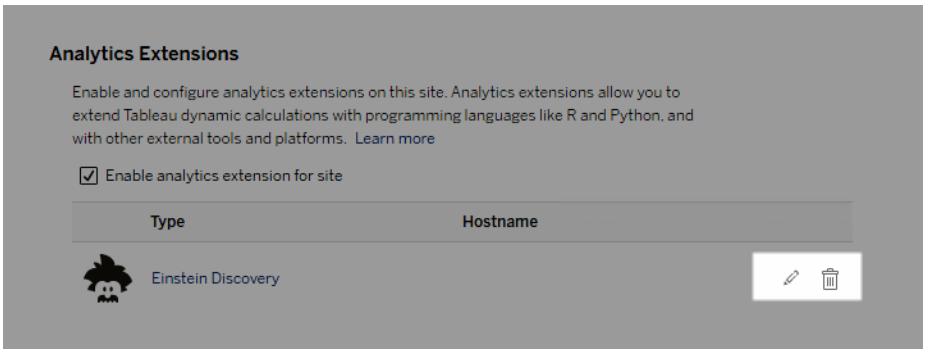
    - For an Einstein Discovery connection, click **Enable**.
    - For TabPy, RServer, and Analytics Extensions API connections, enter the following information:
        - **Connection Name** (Required): Specify the server type you are connecting to. RSERVE supports connections to R using the RServe package. TABPY supports connections to Python using TabPy, or to other analytics extensions.
        - **Require SSL** (Recommended): Select this option to encrypt the connection to the analytics service. If you specify a `HTTPS` URL in the **Hostname** field, then you must select this option.
        - **Hostname** (Required): Specify the computer name or URL where the analytics service is running. This field is case sensitive.
        - **Port** (Required): Specify the port for the service.

- **Sign in with a username and password** (Recommended): Select this option to specify user name and password that is used to authenticate to the analytics service.

7. Click **Create**.

Edit or delete an analytics extension connection

To edit or delete a configuration, navigate to **Analytics Extensions** on the **Extensions** tab of your site.



Click the **Edit** or **Delete** icon and follow the prompts to change the configuration.

Client requirement: Intermediate certificate chain for Rserve external service

As of Tableau Server version 2020.1, you must install a full certificate chain on Tableau Desktop computers (Windows and Mac) that are connecting to a Rserve external connection through Tableau Server. This requirement is due to how Rserve manages the handshake on secure connections.

Importing a root certificate on the Tableau Desktop is not sufficient, the entire certificate chain must be imported onto the client computer.

Script errors

Tableau cannot verify that workbooks that use an analytics extension will render properly on Tableau Server. There might be scenarios where a required statistical library is available on a

user's computer but not on the analytics extension instance that Tableau Server is using.

A warning will be displayed when you publish a workbook if it contains views that use an analytics extension.

> This worksheet contains external service scripts, which cannot be viewed on the target platform until the administrator configures an external service connection.

Determining analytics extensions usage

Beginning with version 2021.2, analytics extensions configurations are mapped at the workbook level. This allows administrators to use custom views to query the Tableau Repository and determine which workbooks are using which extensions, and how often they are used.

To do this you need to join the workbook connections table to tables showing workbook usage. For details about creating and using custom administrative views, see Collect Data with the Tableau Server Repository and Create Custom Administrative Views.

Table Extensions

Table Extensions allow you to create new data tables with an analytics extensions script. You can write a custom TabPy or Rserve script and optionally add one or more input tables. Table extensions are supported by Tableau Cloud, Tableau Server, and Tableau Desktop. This document focuses on Tableau Server.

> **Note:** The data refreshes every time you open up a workbook or refresh a data source.

Benefits

Table Extensions have the following benefits for both new and experienced users.

- Faster data processing
- Low code editor
- Integrates with Ask Data and Explain Data

- Integrates with TabPy and Rserve
- Results can be used to construct dashboards or visualizations.

Prerequisites

Before you can use table extensions, you must complete the following list.

- Configure an analytics extension
  - For steps to configure analytics extension connections, see Configure Connections with Analytics Extensions..
- Publish your workbook.

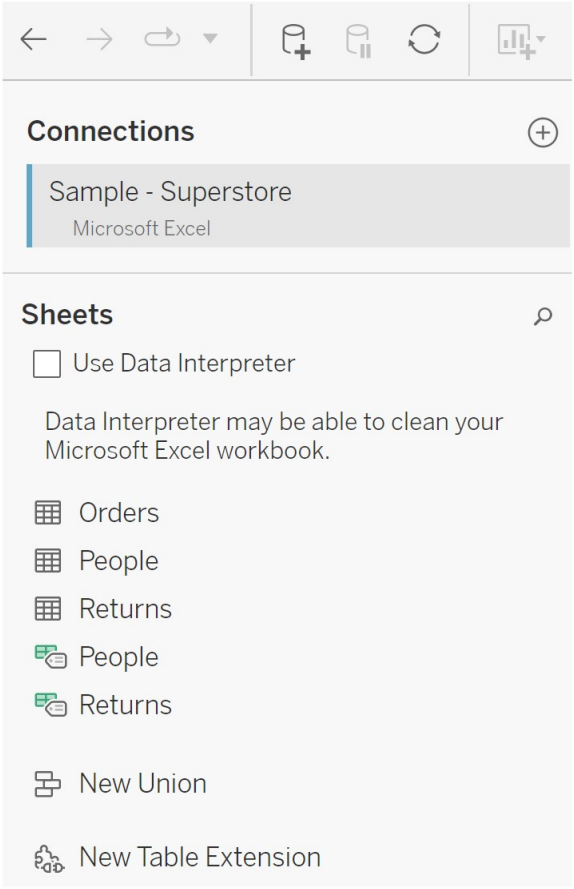
Create a Table Extension

To create a new table extension, complete the steps below.

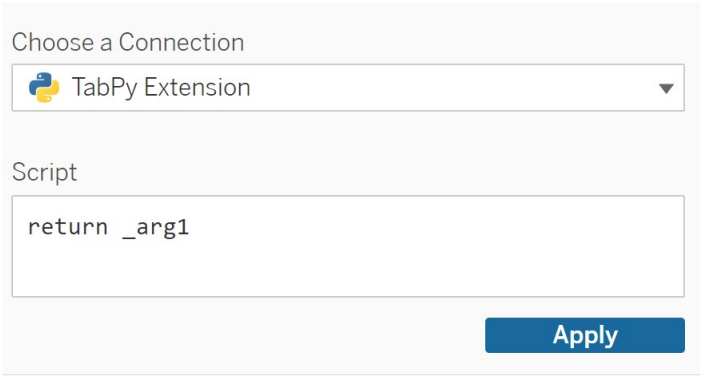
1. Open a published workbook.

   Note: The workbook must be published before you can add a table extension.

2. Under **Sheets**, choose **New Table Extension**.

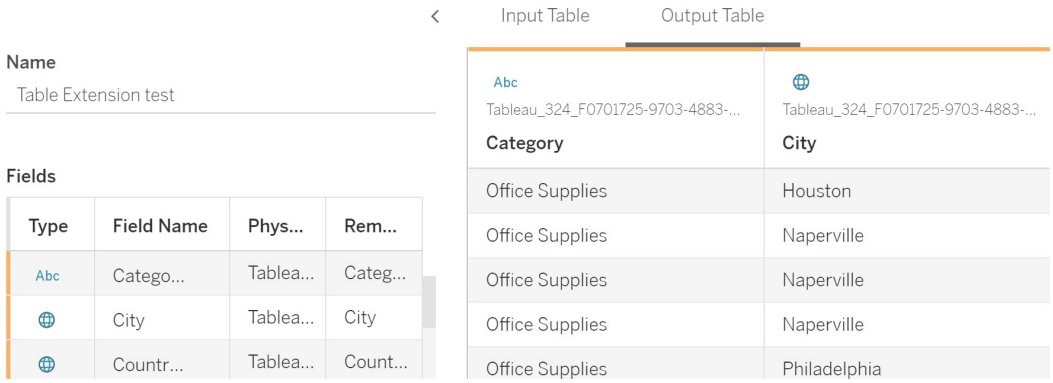


3. (Optional) Drag sheets into the table extension pane.
4. Under **Choose a Connection**, select an analytics extension.



5. In **Script**, enter your script.

6. Select **Apply**.
7. Choose **Update Now** and the results will appear in the **Output Table** tab.



8. In the **Name** field, enter a unique name for your new table extension.
9. Go to the sheet tab and publish the workbook to save it.

> **Note:** If you edit the input table you have to press **Apply** again before you can see or use the updated output table data.

Troubleshooting tip: If your table extension is hitting an error, try using the circular **Refresh Data Source** button, located next to the **Save** button.

Table Extensions vs Analytics Extensions

Tableau has a few different features with "extension" in the name. While some of these products aren't related, table extensions and analytics extensions are. The table extensions feature relies on a connection with a analytics extensions to work. Let's break down each feature.

Table Extensions

The table extensions feature lets you create workbook calculations that send data and a processing script to your analytics extension. The returned results are displayed as a table on the Data Source tab and as measures and dimensions in the workbook.

Analytics Extensions

The analytics extensions feature allows you to extend Tableau dynamic calculations with programming languages like Python, external tools, and external platforms. After you create a connection to an analytics extension, you can communicate with your external server through calculated fields. For more information, see Configure Connections with Analytics Extensions.

Configure Einstein Discovery Integration

Beginning with version 2021.1.0, Tableau Server supports integration with Einstein Discovery, making Einstein Discovery predictions available to authors and viewers of workbooks and dashboards. Starting in version 2021.2.0, Einstein Discovery predictions is also now available when authoring flows on the web.

Einstein Discovery in Tableau is powered by salesforce.com. Consult your agreement with salesforce.com for applicable terms.

To integrate Einstein Discovery with Tableau Server, there are several necessary configuration steps, including some in Tableau Server, and some in the Salesforce org running Einstein Discovery. This overview outlines these steps for Dashboard extensions, Analytics extensions, and Tableau Prep extensions, and provides links to specific topics with steps for completing the server configuration.

For details on how to use Einstein Discovery predictions in Tableau, including licensing and permission requirements, see Integrate Einstein Discovery Predictions in Tableau in the Tableau Desktop and Web Authoring Help. For information about adding predictions in flows, see Add Einstein Discovery Predictions to your flow.

Einstein Discovery dashboard extensions

The Einstein Discovery dashboard extension allow workbook authors to surface real-time predictions in Tableau. The dashboard extension delivers predictions interactively, on-demand,

using source data in a Tableau workbook and an Einstein Discovery-powered model deployed in Salesforce.

To configure Tableau Server for the Einstein Discovery dashboard extension you need to do the following:

1. In Tableau Server:
    a. Enable saved OAuth tokens for data connections and extensions in Tableau Server. Allow Saved Access Tokens
    b. Enable Dashboard extensions for the server. See: Manage Dashboard and Viz Extensions in Tableau Server
2. In Salesforce, in the organization running Einstein Discovery:
    a. Configure CORS in Salesforce.com for Einstein Discover Integration in Tableau Server.
    b. In Salesforce, in the organization running Tableau CRM, create a connected app. See Step 1: Create a Salesforce connected app.
3. In Tableau Server, configure server for saved SF OAuth credentials using information from the connected app. Step 2: Configure Tableau Server for Salesforce.com OAuth

Einstein Discovery analytics extensions

The Einstein Discovery analytics extension gives your users the ability to embed predictions directly in Tableau calculated fields. A table calc script requests predictions from a model deployed in Salesforce by passing its associated prediction ID and input data that the model requires. Use Model Manager in Salesforce to auto-generate a Tableau table calculation script, and then paste that script into a calculated field for use in a Tableau workbook.

To configure Tableau Server for either the Einstein Discovery analytics extension you need to do the following:

1. In Tableau Server:
    a. Enable saved OAuth tokens for data connections and extensions in Tableau Server. Allow Saved Access Tokens
    b. Enable analytics extensions for the server and configure a connection type. See: Configure Connections with Analytics Extensions

2. In Salesforce, in the organization running Einstein Discovery, create a connected app. See Step 1: Create a Salesforce connected app.
3. In Tableau Server, configure server for saved SF OAuth credentials using information from the connected app. Step 2: Configure Tableau Server for Salesforce.com OAuth

Einstein Discovery Tableau Prep extensions

Supported in Tableau Server and Tableau Cloud starting in version 2021.2.0

The Einstein Discovery Tableau Prep extension enables users to embed Einstein predictions directly in their flows when authoring flows on the web.

To configure Tableau Server or Tableau Cloud for the Einstein Discovery Tableau Prep extension you need to do the following:

1. In Tableau Server:
    a. Enable saved OAuth tokens for data connections and extensions in Tableau Server. See Allow Saved Access Tokens
    b. Enable Tableau Prep Extensions for the server. See Enable Tableau Prep Extensions.
2. In Salesforce, in the organization running Einstein Discovery, create a connected app. See Step 1: Create a Salesforce connected app.
3. In Tableau Server, configure server for saved SF OAuth credentials using information from the connected app. Step 2: Configure Tableau Server for Salesforce.com OAuth

Configure External Actions Workflow Integration

Note: External Actions in Tableau rely on functionality provided by Salesforce Flow. The feature sends your selected data to Salesforce Flow, which runs on separate Salesforce infrastructure. Use of Salesforce Flow and other Salesforce products and services is subject to your agreement with Salesforce.

For more information on how to use External Actions, see Integrate External Actions.

Editions, site roles, and permissions requirements

To configure and use External Actions workflows, you and anyone who will be using workflows must have certain site roles and permissions in editions of Salesforce and Tableau that support External Actions.

Product	Editions	Site Roles and Permissions
Tableau	Tableau Cloud, Tableau Desktop, or Tableau Server versions 2022.3 or later	**To create or edit a workflow:** Creator or Explorer (can publish) site role, and permissions to edit and save workbooks (Linux | Windows) **To send data**: Download Summary Data permission capability **To use a workflow:** Any site role
Salesforce	Essentials, Professional, Enterprise, Performance, Unlimited, or Developer edition (see Salesforce Editions)	**To create or edit a flow:** Manage Flows permission **To use a flow:** Run Flows permission or Flow User setting on the user detail page or *Override default behavior and restrict access to enabled profiles or permission sets* setting in Flow node (See How Does Flow Security Work?)

Deployment requirements for External Actions

To use the External Actions workflow extension, the Salesforce administrator must create a connected app in Salesforce for Tableau Server.

Also, the domain for the Tableau Server or Tableau Cloud site where the extension will be deployed must be added to the Salesforce Cross-Origin Resource Sharing (CORS) allowlist.

Turn External Actions On or Off

In Tableau 2022.3 and later, the Tableau External Actions feature is turned on by default. To turn off the feature, you can use the Tableau Services Manager (TSM) command line interface (CLI) or configure site-level settings.

Use the TSM CLI

Use the TSM CLI to set the value for the `vizqlserver.workflow_objects_enabled` configuration key to `tsm configuration set -k vizqlserver.workflow_objects_enabled -v false`.

After changing the configuration key value, be sure to apply the change using the `tsm pending-changes apply` command. If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there's no restart. You can suppress the prompt using the `--ignore-prompt` option, but this option doesn't change the restart behavior. If the changes don't require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Modify Site-Level Settings

For more information on how to customize site-level settings in Tableau Server, see Site Settings Reference.

Integrate Tableau with a Slack Workspace

Beginning with version 2021.3, Tableau Server and Tableau Cloud support integration with the Tableau App for Slack, so your team can collaborate-share Tableau snapshots, search for Tableau content, and receive notifications about Tableau data–right where they’re working in a Slack workspace.

The Tableau App for Slack lets you connect your Tableau site with a Slack workspace. After it’s enabled, Tableau users can:

- See notifications in Slack when teammates share content with them, when they're mentioned in a comment, or when data meets a specified threshold in a data-driven alert. If a site administrator in Tableau Cloud or a server administrator in Tableau Server enables notifications on a site, users can control which notifications they receive in Slack by configuring their Account Settings.
- See a preview of a viz when a Tableau URL is pasted into Slack, allowing users to share data-related content with context directly in Slack.
- Search for Tableau views or workbooks in Slack DMs and channels.
- Access Recents and Favorites from the Tableau App for Slack.

For more information, see Receive Notifications, Search, and Share Using the Tableau App for Slack.

> **Note:** Some notifications preferences might not be available if the features are turned off for your site. For example, if the User Visibility setting is set to Limited, notifications are turned off. For more information, see Site Settings Reference and Manage Site User Visibility.

To integrate Slack with your Tableau site, there are a few necessary configuration steps, including some in your Tableau site, and some in the Slack workspace you want to connect. This overview outlines these steps for both Tableau site administrators on Tableau Cloud or a Tableau Server Administrator on Tableau Server, and Slack workspace administrators.

Requirements

Enabling Tableau in Slack requires both a Slack workspace administrator and either a Tableau site administrator in Tableau Cloud, or a Tableau server administrator in Tableau Server.

Connect a Tableau Server site to a Slack workspace

A Tableau Server administrator can connect a Slack workspace to one Tableau Server site. Connecting your Tableau site to a Slack workspace consists of three tasks:

- **Slack workspace administrator**: Create a private Slack application on the Slack API platform.

- **Tableau server administrator**: Use the app information to add an OAuth client.
- **Tableau server administrator**: Connect your Tableau site to Slack.

Each Tableau site can connect to one Slack workspace per Tableau site. For information about creating Slack apps, see Slack's Best practices and guidelines for Slack platform.

Note: If using a proxy server, make sure you've followed proxy settings guidance in Configuring Proxies and Load Balancers for Tableau Server before you begin.

Additionally, for **Windows**:

- Make sure that the Windows environment variables already have `http_proxy` and `https_proxy` specified. For more information and instructions on specifying environment variables, see Configuring Proxies and Load Balancers for Tableau Server.
- Add the Slack domains in this list to the allowlist. For more information, see Communicating with the Internet.

Step 1: Create a Tableau App for Slack

Slack workspace administrator:

1. Go to Slack's API documentation and select **Create New App**
2. Select **From Scratch**, then add an app name and the workspace for the app to exist within. Select **Create New App**.
3. Give your app a name and select a Slack workspace.
4. You'll be taken to the app's basic information settings. From here, you can modify the app's privileges, description, and more.
5. Select **Bots** from the Add features and functionality section, then select **Review Scopes to Add**.
6. Under Bot Token Scopes, add these scopes:
    - `chat:write`
    - `files:write`
    - `users:read`
    - `users:read.email`

7. Select **OAuth & Permissions** from the navigation menu.

   Important: Opting in to token rotation for your Tableau App for Slack will cause notifications to stop working in Slack. Token rotation can't be removed after it's added.

8. Select **Add New Redirect URL**.
9. Add a fully qualified URL `https://<Tableau Server URL>/auth/add_oauth_token`
10. Select **Basic Information** from the navigation menu.
11. Give the Client ID, Client Secret, and Redirect URL to the Tableau server administrator.
12. Add the Tableau App for Slack into the Slack workspace by selecting Basic Information from the Settings menu, then **Install**.

Note: Administrators can share their app with other Slack workspaces by activating a distribution option. For more information, see Manage Distribution in Slack's documentation.

Step 2: Add an OAuth client to the Tableau site

Tableau server administrator:

The Tableau server administrator will add the OAuth client from the Slack workspace administrator, then connect the Tableau site to Slack.

1. Sign in to the site you'd like to connect to Slack. On the Settings page of your site, select the **Integrations** tab.

2. Under Slack Connectivity, select **Add OAuth Client**.
3. Add the **Client ID**, **Client Secret**, and **Redirect URL** from the Slack workspace administrator, then select **Add OAuth Client** in the dialog.
4. The connection type and Client ID will appear in the table.

Step 3: Finalize the connection

Tableau server administrator:

When the OAuth client is added under Slack Connectivity:

1. Select **Connect to Slack**.
2. Follow the prompt to sign in to your Slack workspace.
3. Select **Allow** to give your Tableau site access to the Slack workspace.

The Tableau site and Slack workspace are now connected. In the Slack workspace, licensed Tableau users can receive Slack notifications when someone shares Tableau content, when a data-driven alert is triggered, or when someone is @mentioned in a comment on a view or workbook.

Disconnect a Tableau site from Slack

As a site admin, you can disconnect a Tableau site from a Slack workspace by selecting **Disconnect from Slack** in the **Integrations** tab of site settings. Users continue to receive notifications for some time. The OAuth client information you added in Step 2 is retained and can be used to connect to a new workspace, if needed.

As a Slack user, you can disconnect from Slack by selecting **Disconnect from Tableau** from the **Home** tab in the Tableau App for Slack.

Update your Tableau App for Slack

When a new version of the Tableau App for Slack is available, Tableau recommends updating the application to maintain app performance and use new features.

To update the Tableau App for Slack:

1. From the **Settings** page of your site, select the **Integrations** tab.

2. Under **Slack Connectivity**, select **Connect to Slack**.
3. Select **Update**.

Note: App updates applied by any Tableau admin affect all Tableau sites connected to the same workspace.

Troubleshoot the Tableau App for Slack

It's a best practice to Manage app approvals for your Slack workspace. However, if your Slack workspace allows users who aren't admins to remove apps, it's possible that a user can remove the Tableau App for Slack for the site. In this scenario, other users will see the Tableau App for Slack, but its features won't work as expected. To resolve this issue, have your Slack admin uninstall and reinstall the Tableau App for Slack. Then, have your admin follow the steps to connect Slack with your Tableau site.

Creators: Connect to Data on the Web

Before you can create a workbook and build a view on the web to analyze your data on the web, you must connect to your data. Tableau supports connecting to data sources on the web published through Tableau Desktop, or, connecting to data directly through Tableau Cloud, Tableau Server, or Tableau Public.

Starting in 2019.3, Tableau Catalog is available as part of the Data Management offering for Tableau Server and Tableau Cloud. When Tableau Catalog is enabled in your environment, in addition to navigating and connecting to data from Explore, you can navigate and connect to more kinds of data, like databases and tables, from Tableau Catalog For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Cloud Help. Starting in 2021.4, Data Management includes virtual connections, a central access point to data. For more information, see "About Virtual Connections and Data Policies" in the Tableau Server or Tableau Cloud help.

Open the Connect to Data page

On the web, you use the Connect to Data page to access data to connect to. After you sign in to Tableau Server or Tableau Cloud, you can open this page two ways:

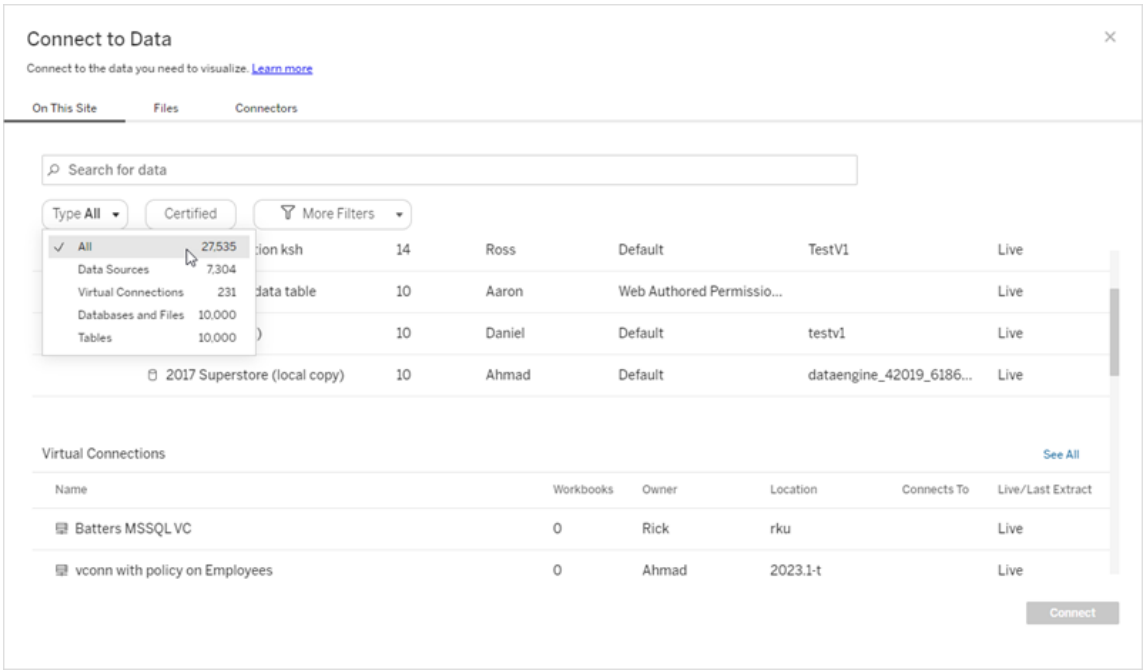
- **Home** > **New** > **Workbook**
- **Explore** > **New** > **Workbook**

If you're on Tableau Public, you can open this page from your author profile:

- **My Profile** > **Create a Viz**

The **Connect to Data** dialog displays a scrollable list of mixed content that's popular. If you have a Data Management license, you can connect to data with a virtual connection, and if you have Data Management with Tableau Catalog enabled, you can also connect to external assets, like databases, files, and tables.

The responsive search field shows a list of suggestions that updates as you enter text. Filter search results by type of data, certification status, or other filters that depend on the type of data selected. For example, some types of data may allow you to filter based on tags, connection type, data quality warnings, or other criteria. Older versions of the dialog look and function slightly differently, but the overall function is similar.



On the Connect to Data page, the tabs you see depend on the product you have.

Tableau Server

On Tableau Server, select from the following tabs to connect to data: On this site, Files, and Connectors.

Connect to data On this site

1. Select **On this site** to browse to or search for published data sources.
2. Select the data source under **Name** and click the **Connect** button.

> **Note:** In addition to connecting to data sources, when you have Data Management, you can use **On this site** to connect to data using a virtual connection. When Tableau Catalog is enabled you can also connect to databases, files, and tables.

Connect to files

Tableau supports uploading Excel, text-based data sources (.xlsx, .csv, .tsv), and spatial file formats that only require one file (.kml, .geojson, .topojson, .json, and Esri shapefiles and Esri File Geodatabases packaged in a .zip) directly in your browser. In the **Files** tab of the **Connect to Data** pane, connect to a file by dragging and dropping it into the field or clicking **Upload from Computer**. The maximum file size you can upload is 1 GB.

Use connectors

From the **Connectors** tab, you can connect to data housed in a cloud database or on a server in your enterprise. You must supply connection information for each data connection that you make. For example, for most data connections, you must supply a server name and your sign-in information.

Supported Connectors has information on how to connect Tableau to each of these connector types to set up your data source. If the connector you need doesn't appear in the Connectors tab, you can connect to data through Tableau Desktop and publish your data source to Tableau Cloud or Tableau Server for web authoring. Learn more about how to Publish a Data Source in Tableau Desktop.

When Tableau successfully connects to your data, the Data Source page opens so that you can prepare the data for analysis and begin building your view. To learn more, see Creators: Prepare Data on the Web.

Tableau Server connectors

Actian Matrix*

Alibaba AnalyticDB for MySQL‡

Alibaba Data Lake Analytics‡

Alibaba MaxCompute‡

Amazon Athena‡

Amazon Aurora for MySQL‡

Amazon EMR Hadoop Hive‡

Amazon Redshift‡

Apache Drill‡

Aster Database*

Azure Data Lake Storage Gen2‡

Box‡

Cloudera Hadoop‡

Databricks‡

Datorama by Salesforce‡

Denodo‡

Google BigQuery**‡

Google BigQuery JDBC**‡

Google Cloud SQL‡

Google Drive‡

Impala‡

Kognito*

Kyvos‡

Hortonworks Hadoop Hive

IBM BigInsights

IBM DB2‡

IBM PDA (Netezza)*

Kyvos‡

MariaDB‡

MarkLogic*

Microsoft Azure SQL Database‡

Microsoft Azure Synapse Analytics‡

Microsoft SQL Server‡

MonetDB*

OData‡

OneDrive‡

Oracle‡

Pivotal Greenplum Database‡

PostgreSQL‡

Progress OpenEdge*

Presto‡

Qubole Presto‡

SAP HANA (for virtual connections only)‡

SAP Sybase ASE*

SAP Sybase IQ*

Salesforce‡

SharePoint Lists‡

SingleStore (formerly MemSQL)‡

Snowflake‡

Spark SQL‡

Teradata***‡

Vertica‡

Dremio by Dremio‡	MongoDB BI Connector‡
Dropbox‡	MySQL‡
Esri Connector‡	
Exasol‡	

*Not available on Linux servers.

**Google BigQuery needs OAuth when creating data sources from the web. Learn more about how server administrators can Set up OAuth for Google.

***Teradata web authoring currently doesn't support query banding functionality. See Teradata for details.

‡Supports virtual connections if you have Data Management. See About Virtual Connections and Data Policies in the Tableau Server help for details.

Tableau Catalog Supported Connectors

Tableau Catalog supports making a connection with a subset of the data connectors that Tableau Server supports. If a data source, database, file, or table is grayed out, you can't connect from Tableau Server. You can, however, connect from the Tableau Desktop **Connect** pane, if you have the correct permissions.

Tableau Cloud

On Tableau Cloud, select from the following tabs to connect to data: On this site, Files, Connectors, and Dashboard Starters.

Connect to data On this site

1. Select **On this site** to browse to or search for published data sources.
2. Select the data source under **Name** and click the **Connect** button

Note: In addition to connecting to data sources, when you have Data Management, you can use **On this site** to connect to data using a virtual connection. When Tableau Catalog is enabled you can also connect to databases, files, and tables.

Connect to files

Tableau supports uploading Excel or text-based data sources (.xlsx, .csv, .tsv) directly in your browser. In the **Files** tab of the Connect to Data pane, connect to an Excel or text file by dragging and dropping it into the field or clicking **Upload from Computer**. The maximum file size you can upload is 1 GB.

Use connectors

From the **Connectors** tab, you can connect to data housed in a cloud database or on a server in your enterprise. You must supply connection information for each data connection that you make. For example, for most data connections, you must supply a server name and your sign-in information.

Supported Connectors has information on how to connect Tableau to your data using connectors. If the connector you need doesn't appear in the Connectors tab, you can connect to data through Tableau Desktop and publish your data source to Tableau Cloud or Tableau Server for web authoring. Learn more about how to Publish a Data Source in Tableau Desktop.

Note: If you're unable to connect to your data from Tableau Cloud, check to see if the database is publicly accessible. Tableau Cloud can only connect to data that's accessible from the public internet. If your data is behind a private network, you can connect using Tableau Bridge. To learn more, see Publishers: Use Tableau Bridge to Keep Tableau Cloud Data Fresh.

Tableau Cloud Connectors

- Alibaba AnalyticsDB for MySQL‡
- Alibaba Data Lake Analytics‡
- Amazon Athena‡
- Amazon Aurora for MySQL‡
- Amazon EMR Hadoop Hive‡
- Amazon Redshift‡
- Apache Drill‡
- Azure Data Lake Storage Gen2‡
- Azure Synapse Analytics (SQL Server compatible)
- Box‡
- Cloudera Hadoop‡
- Databricks‡
- Datorama by Salesforce‡
- Denodo‡
- Dremio by Dremio‡
- Dropbox*‡
- Esri Connector‡
- Exasol‡
- Google BigQuery*‡
- Google Cloud SQL (MySQL compatible)‡§
- Google Drive‡
- Hortonworks Hadoop Hive
- Impala‡
- Kyvos‡
- MariaDB‡
- Microsoft Azure SQL Database‡
- Microsoft Azure Synapse Analytics‡
- Microsoft SQL Server‡
- MongoDB BI Connector‡
- MySQL‡
- OData‡
- OneDrive*‡
- Oracle‡
- Pivotal Greenplum Database‡
- PostgreSQL‡
- Presto‡
- Qubole Presto‡
- Salesforce‡
- SAP HANA (for virtual connections only)‡
- SharePoint Lists‡
- SingleStore (formerly MemSQL)‡
- Snowflake‡
- Spark SQL‡
- Teradata**‡
- Vertica‡

*For more information about using OAuth 2.0 standard for Google BigQuery, OneDrive, and Dropbox connections in Tableau Cloud, see OAuth Connections.

**Teradata web authoring currently doesn’t support query banding functionality. See Teradata for details.

‡Supports virtual connections if you have Data Management. See About Virtual Connections and Data Policies in the Tableau Cloud help for details.

§Tableau Cloud doesn't support SSL using Google Cloud SQL.

Tableau Catalog Supported Connectors

Tableau Catalog supports making a connection with a subset of data connectors that Tableau Cloud supports. If a data source, database, file, or table is grayed out, you can't connect from Tableau Cloud. You can, however, connect from the Tableau Desktop **Connect** pane, if you have the correct permissions.

Use Dashboard Starters

On Tableau Cloud, you can author and analyze data from LinkedIn Sales Navigator, Oracle Eloqua, Salesforce, ServiceNow ITSM, and QuickBooks Online using Dashboard Starters. On the **Dashboard Starter** tab, from the list of pre-built designs, select an option and click **Use Dashboard**. See Dashboard Starters for Cloud-based Data Sources for details.

Tableau Public

On Tableau Public, you can connect to data by uploading a supported file.

Connect to files

Tableau supports uploading Excel or text-based data sources (.xlsx, .csv, .tsv) directly in your browser. In the **Files** tab of the Connect to Data pane, connect to an Excel or text file by dragging and dropping it into the field or clicking **Upload from Computer**. The maximum file size you can upload is 1 GB.

If you don’t have a data set, check out the free sample data sets on the Tableau Public website.

Use connectors

From the **Connectors** tab, you can connect to data housed in a cloud database. You must supply connection information for each data connection that you make. For example, for most data connections, you must supply your sign-in information.

Supported Connectors has information on how to connect Tableau to your data using connectors. If the connector you need doesn't appear in the Connectors tab, you can connect to data through Tableau Desktop and create an data extract.

> **Note**: If you're unable to connect to your data from Tableau Public, check to see if the database is publicly accessible. Tableau Public can only connect to data that's accessible from the public internet.

Tableau Public Connectors

Google Drive

OData

After you connect

When Tableau connects to your data, the Data Source page opens so that you can prepare the data for analysis and begin building your view. To learn more, see Creators: Prepare Data on the Web.

Keep data fresh in web authoring

Update uploaded files in Tableau Cloud or Tableau Server: If you manually upload a file (Excel or text) for web authoring, Tableau can't refresh the file automatically. To update your data, select "Edit Connection" to upload a new version of the file.

In Tableau Public, go to your viz and click **Request Update**. You can also keep your data fresh automatically by selecting "Keep this data in sync" in Tableau Desktop Public Edition.

Update file-based published data sources in Tableau Cloud: If you have a published data source in Tableau Cloud (published through Tableau Desktop) that uses file-based data, you can keep it fresh using Tableau Bridge. For more information, see Expand Data Freshness Options by Using Tableau Bridge.

Run Initial SQL

Note: Tableau Prep Builder version 2019.2.2 and later supports using Initial SQL, but doesn't yet support all of the same options supported by Tableau Desktop For information about using Initial SQL with Tableau Prep Builder, see Use Initial SQL to query your connections in the Tableau Prep Builder Salesforce Help.

When connecting to some databases, you can specify an initial SQL command that will run or use a cached value when a connection is made to the database, for example, when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server.

Note: Initial SQL is different than a custom SQL connection. A custom SQL connection defines a relation (or table) to issue queries against. For more information, see Connect to a Custom SQL Query.

You can use this command to:

- Set up temporary tables to use during the session.
- Set up a custom data environment.

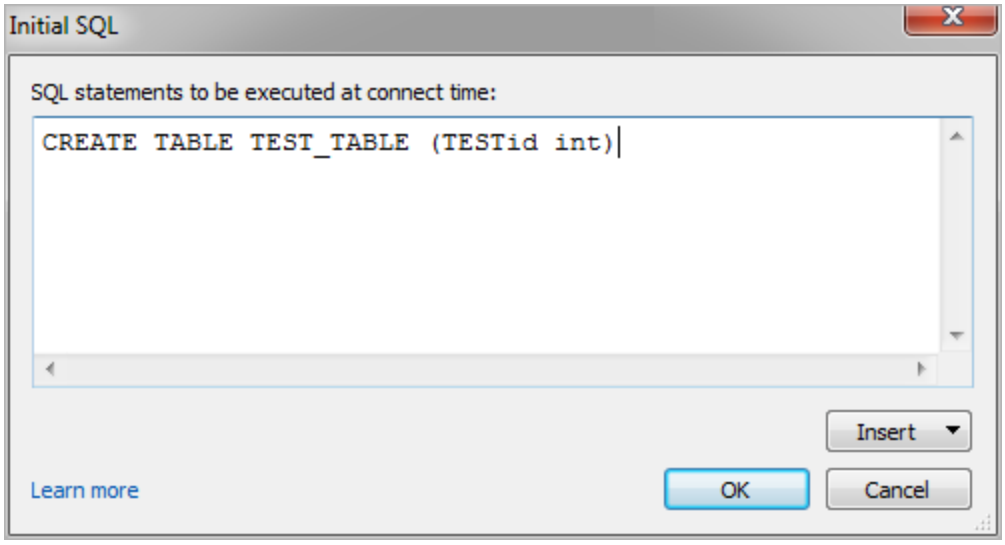
You have the option to add an initial SQL command in the Server Connection dialog box or on the Data Source page.

Note: If your data source supports running an initial SQL statement, an **Initial SQL** link appears in the lower-left corner of the Server Connection dialog box. For information about your data source, see Supported Connectors.

To use initial SQL

1. In the Server Connection dialog box, click **Initial SQL**. Or, on the Data Source page, select **Data** > **Initial SQL** or **Data** > **Query Banding and Initial SQL** depending on the database you connect to.

2. Enter the SQL command into the Initial SQL dialog box. You can use the **Insert** drop-down menu to pass parameters to your data source.



Note: Tableau doesn't examine the statement for errors. This SQL statement is sent to the database when you connect.

Your software license may restrict you from using initial SQL with your connection. If you publish to Tableau Server, the server must be configured to allow Initial SQL statements. By

default, the server software is configured to allow these statements to run when the workbook is loaded in a web browser.

Administrators can configure the server to ignore initial SQL statements by using the `tsm configuration set` command:

```
tsm configuration set -k vizqlserver.initialsql.disabled -v true
```

If the server doesn't allow initial SQL statements, the workbook opens, but the initial SQL commands aren't sent.

For more information about the `tsm configuration set` command, see the Tableau Server Help.

Parameters in an initial SQL statement

You can pass parameters to your data source in an initial SQL statement. The following list has several benefits of using parameters in a initial SQL statement.

- You can configure impersonation using the **TableauServerUser** or **TableauServerUserFull** parameters.
- If your data source supports it, you can set up row-level security (for example, for Oracle VPD or SAP Sybase ASE) to make sure that users see only the data that they're authorized to see.
- You can provide more details in logging, for example, the Tableau version or the workbook name.

The following parameters are supported in an initial SQL statement:

Parameter	Description	Example of returned value
TableauServerUser	The username of the current server user. Use when setting up impersonation on the server. Returns an	jsmith

	empty string if the user isn't signed in to Tableau Server.	
TableauServerUserFull	The username and domain of the current server user. Use when setting up impersonation on the server. Returns an empty string if the user isn't signed in to Tableau Server.	domain.lan\jsmith
TableauApp	The name of the Tableau application.	Tableau Desktop Professional Tableau Server
TableauVersion	The version of the Tableau application.	9.3
WorkbookName	The name of the Tableau workbook. Use only in workbooks with an embedded data source.	Financial-Analysis

Warning: Tableau Desktop doesn't include domain. You can include it if you aren't using delegation and you set tsm configuration set -k DelegationUseFullDomainName=-v true--force-keys

The following examples show different ways you can use parameters in an initial SQL statement.

- This example sets the security context on Microsoft SQL Server:

  ```
  EXECUTE AS USER = [TableauServerUser] WITH NO REVERT;
  ```

- This example shows how, on a DataStax data source, you can use parameters to add detail to logging or to set up a session variable to track the data:

  ```
  SET TABLEAUVERSION [TableauVersion];
  ```

- This example can be used to help set up row-level security for Oracle VPD:

```
begin

 DBMS_SESSION.SET_IDENTIFIER([TableauServerUser]);

end;
```

> **Note:** Oracle PL/SQL blocks require a trailing semicolon to terminate the block. Consult Oracle documentation for the proper syntax.

Defer execution to the server

You can defer an initial SQL statement so that it's executed only on the server. One reason to defer execution to the server is if you don't have permission to execute the commands that set up impersonation. Use <ServerOnly></ServerOnly> tags to enclose the commands to be executed only on the server.

Example:

```
CREATE TEMP TABLE TempTable(x varchar(25));
INSERT INTO TempTable VALUES (1);
<ServerOnly>INSERT INTO TempTable Values(2);</ServerOnly>
```

Security and impersonation

When you use the **TableauServerUser** or **TableauServerUserFull** parameter in an initial SQL statement, you'll create a dedicated connection that can't be shared with other users. This can also restrict cache sharing, which can enhance security, but may also slow performance.

Troubleshoot 'create table' for MySQL and Oracle connections

For MySQL connections, tables aren't listed after using initial SQL to create a table

After you connect to MySQL and run an initial SQL statement, the tables might not show because of the way Tableau constructs the query.

```
CREATE TABLE TestV1.testtable77(testID int);
```

To resolve this issue, add `IF NOT EXISTS` to the SQL statement:

```
CREATE TABLE IF NOT EXISTS TestV1.TestTable(testID int);
```

For Oracle connections, using initial SQL to create a table causes Tableau to stall

After you connect to Oracle and run an initial SQL statement, Tableau is stalled with a spinning wheel because of the way Tableau constructs the query.

```
CREATE TABLE TEST_TABLE (TESTid int)
```

To resolve this issue, use the following SQL statement:

```
BEGIN
EXECUTE IMMEDIATE 'create table test_table(testID int)';
EXCEPTION
WHEN OTHERS THEN NULL;
END;
```

Create and Interact with Flows on the Web

Starting in version 2020.4, you can create and interact with flows on Tableau Server to clean and prepare your data. Connect to your data, build a new flow, or edit an existing flow and your work is automatically saved every few seconds as you go. Create draft flows that are only available to you or publish your flow to make it available for others. Run your individual flows right from the web or run your flows automatically on a schedule using Tableau Prep Conductor if Data Management is licensed. For more information, see Tableau Prep on the Web.

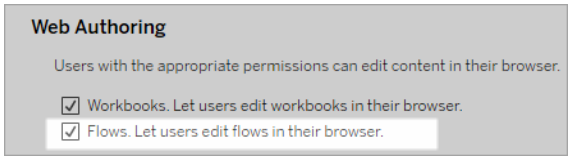
For more information about configuring the Tableau Prep Flow Authoring process on Tableau Server, see Tableau Prep Flow Authoring .

Complete the tasks described in this article to enable web authoring for flows and other flow features on the web.

Turn flow web authoring on or off for a site

Enabled by default, this option controls whether users can create and edit flows on Tableau Server or Tableau Cloud

1. In a web browser, sign in to the server as an administrator, go to the site and click **Settings**.
2. In the **Web Authoring** section, clear or select **Flows. Let users edit flows in their browser.** to turn the functionality off or on.



3. If you want the change to take effect immediately, restart the server. Otherwise, the change takes effect after server session caching expires or the next time users sign in after signing out.

Enable linked tasks

Supported in Tableau Cloud and Tableau Server version 2021.3 and later.

Use the **Linked Tasks** option to schedule up to 20 flows to run sequentially, one after the other. Linked tasks can only be run on schedules with the **Linked Tasks** option selected. For more information about setting up linked tasks, see Schedule Linked Tasks.

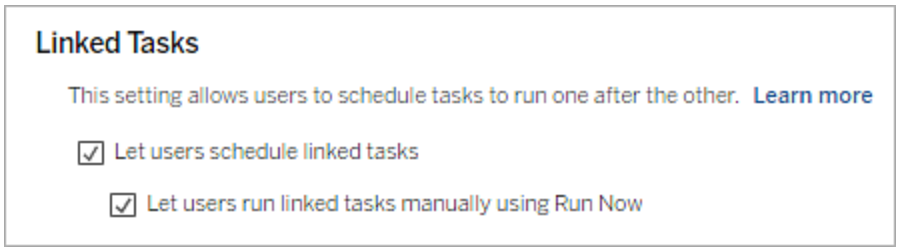
Starting in version 2022.1, **Linked Tasks** is enabled by default on the **Server Settings** and for any new flow schedules that you create. In prior versions, administrators must first enable the option.

If you have multiple sites, you can turn off **Linked Tasks** for individual sites by clearing the check boxes described below.

If the setting is turned off after linked tasks are scheduled, any tasks that are running will complete and the scheduled linked tasks are hidden and no longer show on the **Scheduled Tasks** tab.

Enable Linked Tasks (version 2021.4 and earlier)

1. In a web browser, sign in to the server as an administrator and go to the site in which you want to enable **Linked Tasks** for flows. In that site, click **Settings**.
2. In the **Linked Tasks** section, select **Let users schedule linked tasks** to enable administrators to configure schedules to run linked tasks.
3. Select **Let users run linked tasks manually using Run Now** to enable users to run linked flow tasks using **Run Now**.



Enable flow parameters

Enable users to schedule and run flows that include parameters. Parameters enable users to scale their flows by building them once, then changing the parameter values to accommodate different data scenarios.

Parameters can be entered in an input step for file name and path, table name, or when using custom SQL queries, in an output step for file name and path and table name, and in any step type for filters or calculated values.

Starting in Tableau Prep Builder and Tableau Cloud version 2023.2, you can add system parameters to flow output names to automatically include the flow run start date and time.

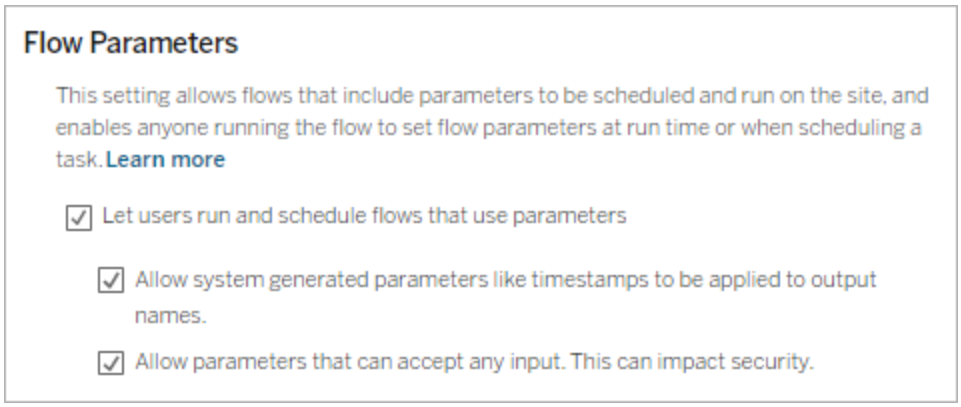
Flow parameter settings can be applied at the server level to include all sites on Tableau Server. The settings can be disabled at the site level to include only specific sites.

For more information about using parameters in flows, see Create and Use Parameters in Flows in the Tableau Prep help.

1. In a web browser, sign in to the server as an administrator and go to the site in which you want to enable **Flow Parameters**. In that site, click **Settings**.
2. In the **Flow Parameters** section, select **Let users run and schedule flows that use parameters** to enable the functionality.
3. (version 2023.2 and later) Select **Allow system generated parameters like timestamps to be applied to output names** to enable users to add a date or time stamp to the flow output name at runtime for file and published data source output types.
4. Select **Allow parameters that can accept any input** to enable anyone running the flow to enter any parameter value in the flow at run time.

   Important: Setting this option enables any flow user to enter any value in a parameter, potentially exposing data that the user should not have access to.

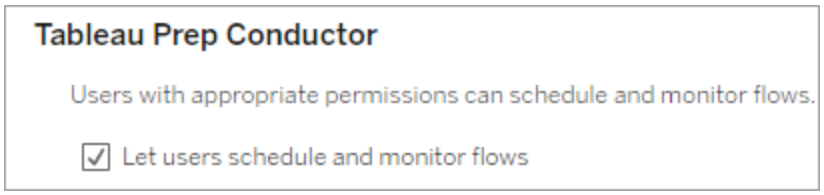
   If this option is not selected, users can only select from predefined list of parameter values and any flows that include parameters that accept any value cannot be run or scheduled to run.



Enable Tableau Prep Conductor

If Data Management is licensed, enable this option to let users schedule and track flows in Tableau Server and Tableau Cloud. For information about the additional configuration requirements for Tableau Prep Conductor, see Tableau Prep Conductor.

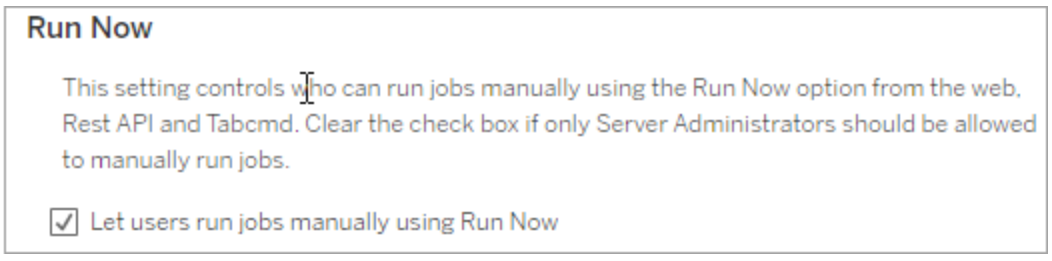
1. In a web browser, sign in to the server as an administrator and go to the site in which you want to enable Tableau Prep Conductor. In that site, click **Settings**.
2. In the **Tableau Prep Conductor** section, select **Let users schedule and monitor flows** to enable the functionality.



Enable Run Now

Control whether users or only administrators can run flows manually using the **Run Now** option. Data Management is not required to run flows manually.

1. In a web browser, sign in to the server as an administrator and go to the site in which you want to enable Run Now for flows. In that site, click **Settings**.
2. In the **Run Now** section, select **Let users run jobs manually using Run Now** to enable the functionality.

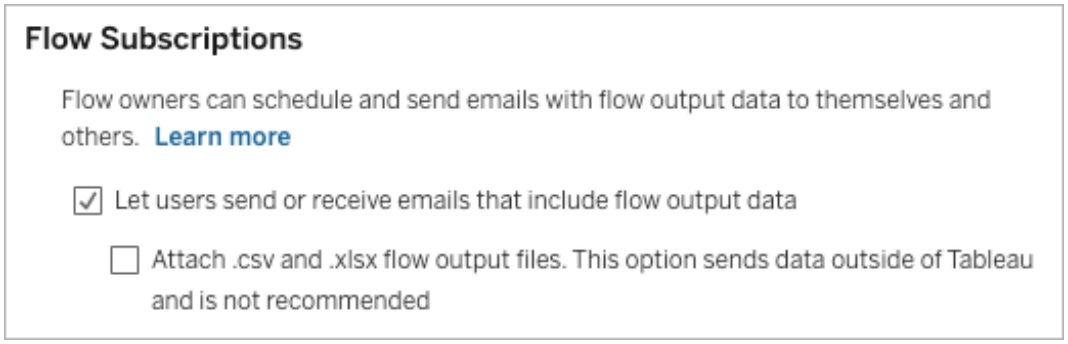


Clear the check box if only Server Administrators can run flows manually.

Flow Subscriptions

Control whether users can receive flows notifications about scheduled tasks for successful flow runs. Data Management is required to enable notifications.

1. In a web browser, sign in to the server as an administrator and go to the site in which you want to enable flow subscriptions. In that site, click **Settings**.
2. In the **Flow Subscriptions** section, select **Let users send or receive emails that include flow output data** to enable the functionality.



Note: The option to attach either a .csv or .xlsx file type to the email is only available for on-premise environments.

Enable Tableau Prep Extensions

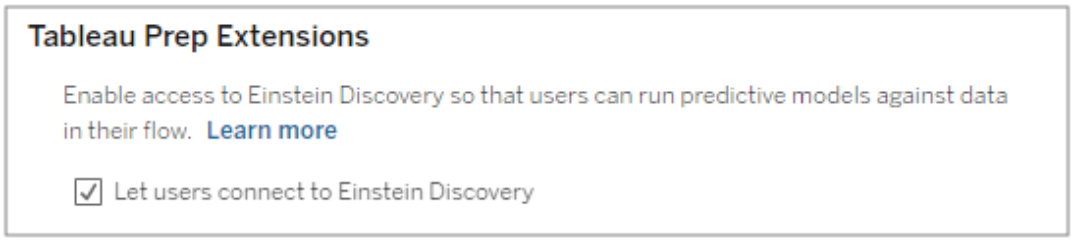
Supported in Tableau Server and Tableau Cloud starting in version 2021.2.0

Set this option to control whether users can connect to Einstein Discovery to run predictive models against data in their flow.

You can use Einstein Discovery-powered models to bulk score predictions for the data in your flow when authoring flows on the web. Predictions can help you make better informed decisions and take actions to improve your business outcomes.

You'll need to configure additional settings to include predictions in our flow. For more information, see Add Einstein Discovery Predictions to your flow and Configure Einstein Discovery Integration.

1. In a web browser, sign in to the server as an administrator and go to the site in which you want to enable Tableau Prep Extensions. In that site, click **Settings** > **Extensions**.
2. In the **Tableau Prep Extensions** section, select **Let users connect to Einstein Discovery** to enable the functionality.



Turn autosave off or on

Enabled by default, this feature automatically saves a user's flow work every few seconds.

While not recommended, administrators can disable autosave on a site using the Tableau Server REST API method "Update Site" and `flowAutoSaveEnabled` setting. For more information, see Tableau Server REST API Site Methods: Update Site. For more information about autosave on the web, see Turn autosave off or on.

For more information about configuring site settings, see Site Settings Reference in the Tableau Server help.

Tableau Prep on the Web

Internet Explorer 11 on Windows and compatibility mode for Internet Explorer is not supported.

Starting in version 2020.4 ,Tableau Prep supports web authoring for flows. Now you can create flows to clean and prepare your data using Tableau Prep Builder, Tableau Server, or Tableau Cloud. You can also manually run flows on the web and the Data Management is not required.

While most of the same Tableau Prep Builder functionality is also supported on the web, there are a few differences when creating and working with your flows.

Important: To create and edit flows on the web you must have a Creator license. Data Management is only required if you want to run your flows on a schedule using Tableau Prep Conductor. For more information about configuring and using Tableau Prep Conductor, see Tableau Prep Conductor in the Tableau Server or Tableau Cloud help.

Installation and Deployment

To enable users to create and edit flows on the web, you'll need to configure several settings on your server. For more information about each of these settings, see Create and Interact with Flows on the Web.

- **Web Authoring**: Enabled by default, this option controls whether users can create and edit flows on Tableau Server or Tableau Cloud.
- **Run Now**: Controls whether users or only administrators can run flows manually using the **Run Now** option. The Data Management isn't required to run flows manually on the web.
- **Tableau Prep Conductor**: If Data Management is licensed, enable this option to let users schedule and track flows.
- **Tableau Prep Extensions** (version 2021.2.0 and later): Controls whether users can connect to Einstein Discovery to apply and run predictive models against data in their flow.
- **Autosave**: Enabled by default, this feature automatically saves a user's flow work every few seconds.

On Tableau Server, administrators can fine-tune the configuration of the Tableau Prep Flow Authoring processes. For more information, see Tableau Prep Flow Authoring.

Sample data and processing limits

To maintain performance while working with flows on the web, limits are applied to the amount of data you can include in a flow.

The following limits apply:

- When connecting to files, the maximum file size is 1GB.
- The data sampling option to include all data is not available. The default sample data limit is 1 million rows.
- The maximum number of rows that a user can select when using large data sets is configured by the administrator. As a user, you can select the number of rows up to that limit. For more information, see tsm configuration set Options.

For more information about setting your data sample, see Set your data sample size in the Tableau Prep help.

Available features on the web

When you create and edit flows on the web you may notice a few differences in navigation and the availability of certain features. While most features are available across all platforms, some features are limited or not yet supported in Tableau Server or Tableau Cloud. The following table identifies features where differences might apply.

Feature area	Exceptions	Tableau Prep Builder	Tableau Server	Tableau Cloud
Connect to Data	Some connectors may not be supported on the web. Open the **Connect** pane on your server to see supported connectors.	✓	✓	✓
Build and Organize your Flow		✓	✓	✓
Set your data sample size	In Tableau Server and Tableau Cloud, the data sample size is subject to limits set by your administrator	✓	✓	✓
Union files and database tables in the	Input unions can't be edited or created in Tableau Server or Tableau Cloud. Only in	✓	✓	✓

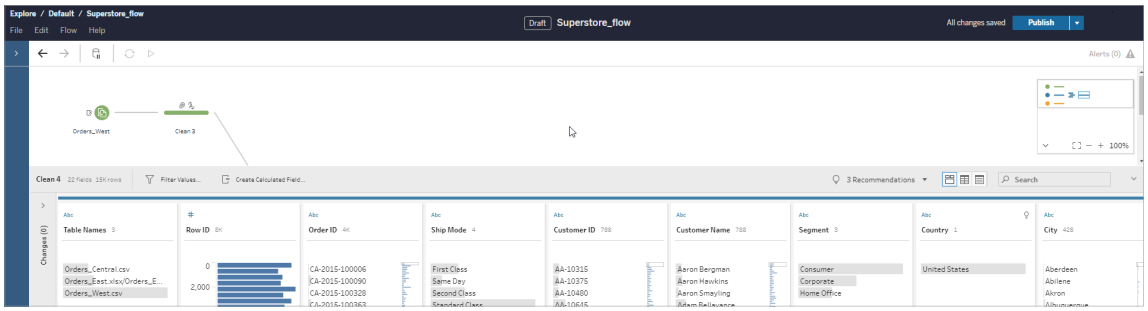
input step	Tableau Prep Builder.			
Clean and Shape Data		✓	✓	✓
Copy data grid values	Available in Tableau Prep Builder and Tableau Server starting in version 2022.3 and Tableau Cloud starting in 2022.2 (August)	✓	✓	✓
Aggregate, Join, or Union Data		✓	✓	✓
Use R and Python Scripts in your Flow	Script steps can't be added when creating or editing a flow in Tableau Cloud. This is currently supported only in Tableau Prep builder and Tableau Server.	✓	✓	⊘
Create reusable flow steps		✓	⊘	⊘
Automatically save your flows on the web		Not Applicable	✓	✓
Automatic file recovery		✓	Not Applicable	Not Applicable
View flow output in Tableau Desktop		✓	⊘	⊘
Create an extract to a file		✓	⊘	⊘

Create an extract to a Microsoft Excel worksheet		✓	⊘	⊘
Connect to a Custom SQL Query		✓	✓	✓
Create a published data source		✓	✓	✓
Save flow output to external databases		✓	✓	✓
Add Einstein Discovery Predictions to your Flow		✓	✓	✓

Autosave and working with drafts

When you create or edit flows on the server, your work is automatically saved as a draft every few seconds so that in the event of a crash, or when closing a tab by accident, you don't lose your work.

Drafts are saved to the server and project you are signed into. You can't save or publish a draft to another server, but you can save the flow to another project on that server using the **File** > **Publish As** menu option.



Draft content can only be seen by you until you publish it. If you publish changes and need to revert them, you can use the **Revision History** dialog to view and revert to a previously published version. For more information about saving flows on the web, see Automatically save your flows on the web.

Publishing flows on the web

Whether you create a flow from scratch on the web or edit an existing flow, before you can run the flow you'll need to publish it.

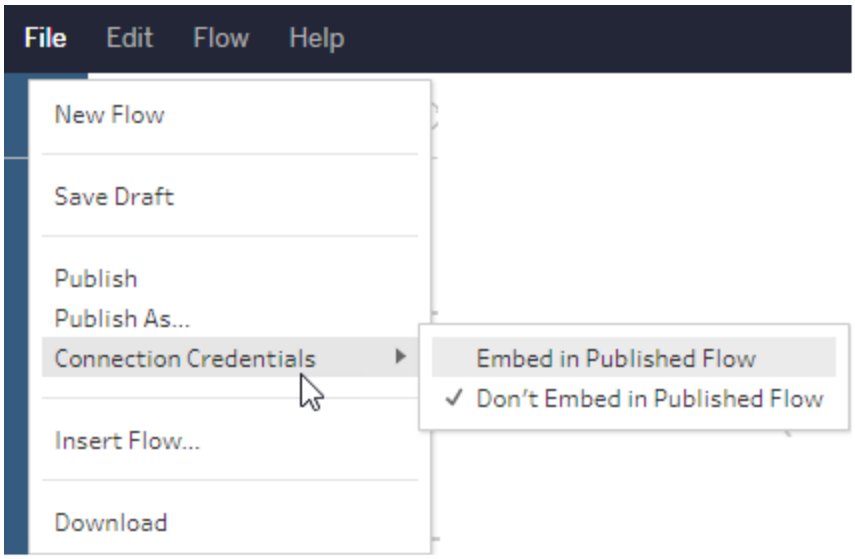
- You can only publish draft flows to the same server you are signed into.
- You can publish a draft to a different project using the **File** menu and selecting **Publish As**.
- You can embed credentials for your flow's database connections to enable the flow to run without having to manually enter the credentials when the flow runs. If you open the flow to edit it, you'll need to re-enter your credentials.

Embed credentials

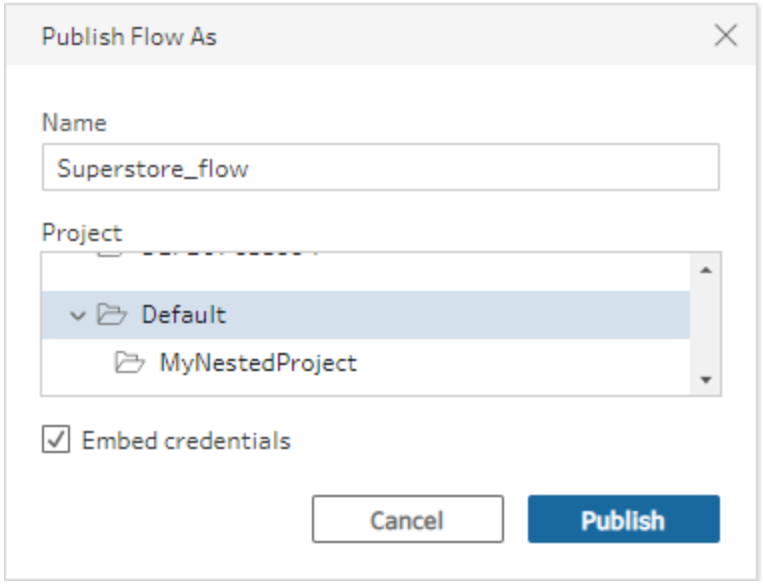
Embedding credentials only applies to running flows on your server. Currently, you will manually need to enter your credentials when editing a flow connected to a database. Embedding credentials can only be set at the flow level and not at the server or site level.

Do one of the following:

- From the top menu, select **File** > **Connection Credentials** > **Embed in Published Flow**.



- When publishing a flow, select the **Embed credentials** check box. This option shows when you select **Publish As** to publish the flow to a new project for the first time or when you are editing a flow that was last published by someone else.



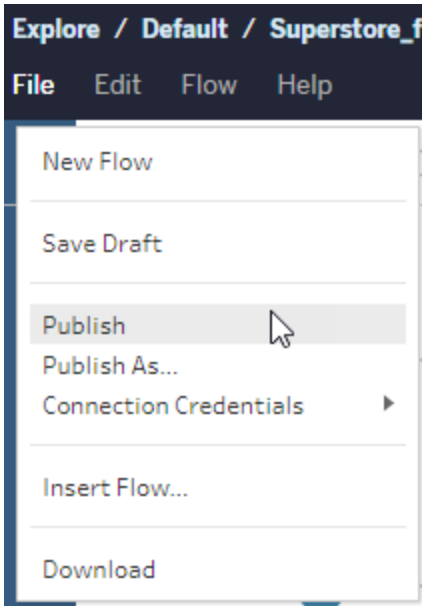
Publish a flow

When you publish your flow, it becomes the current version of the flow and can be run and seen by others who have access to your project. Flows that are never published or flow

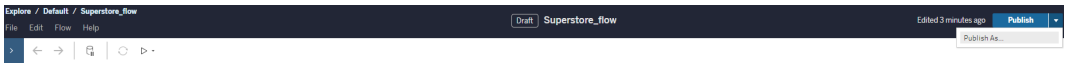
changes that you make to a draft can only be seen by you until you publish the flow. For more information about flow statuses, see Automatically save your flows on the web.

To publish your flow, do one of the following:

- From the top menu, select **File** > **Publish** or **File** > **Publish As**



- From the top bar, click the **Publish** button or click the drop arrow to select **Publish As**.



Who can do this

- Server Administrator, Site Administrator Creator, and Creator allow full connecting and publishing access.
- Creator can perform web authoring tasks.

Create Views and Explore Data on the Web

You can create and interact with views on Tableau Server. For more information, see the following topics in the Tableau Help for users.

Using Tableau on the Web

Tour your Tableau Site

Edit Views on the Web

Join your Data

Create a Dashboard

Create a Story

Embed Views and Dashboards in Web Pages

Make Workbooks Compatible Between Versions

Alerts and subscriptions

Troubleshoot Subscriptions

Send Data-Driven Alerts from Tableau Cloud or Tableau Server

Manage Saved Credentials for Data Connections

Saved credentials enable you to connect to a data source without being prompted for your credentials. The credentials saved for your connection can be OAuth access tokens, or other credentials, such as user name and password. You can manage saved credentials on your account settings page.

On Tableau Server, if your server administrator has allowed you to save credentials, you can find and manage them in the **Saved Credentials** section on your **My Account Settings** page. If you don't see the **Saved Credentials** section, consult with your admin about allowing saving credentials. For more information, see Allow Saved Access Tokens.

Note: When editing Tableau Prep flows on the web, you may still be prompted to re-authenticate.

Test connections using saved credentials

If the connector supports test functionality, you can test the connection using saved credentials.

1. While you're signed in to Tableau Server or Tableau Cloud, display your Account Settings page.
2. In the Saved Credentials section, select the **Test** link next to the stored connection that you want to test.

This test confirms that Tableau Cloud or Tableau Server can access your account using this corresponding saved credential. If the test succeeds, but you can't access your data through this managed connection, confirm that the credentials you provided for this connection can access your data.

For example, if you accidentally created the connection using your personal Gmail account, but you use a different account to access a Google Analytics database, you'll need to delete the saved credentials and sign in to the data using the appropriate Gmail account.

Update saved credentials

To help ensure uninterrupted data access from existing Tableau content after a custom OAuth client has been configured for your site, we encourage you to update your saved credentials. To update saved credentials, you can delete the previous saved credentials for a particular connector and then add it again.

When you add saved credentials again, both new and existing Tableau content will access the data using the custom OAuth client configured by your server administrator. For more information about custom OAuth clients, see Configure a custom OAuth for a site.

1. Sign in to Tableau Server and navigate to your **My Account Settings** page.
2. Under **Saved Credentials for Data Sources**, do the following:
    1. Click **Delete** next to the saved credentials for a connector.
    2. Next to the same connector, click **Add** and follow the prompts to 1) connect to the custom OAuth client that your site admin notified you about and 2) save the latest credentials.

Clear all saved credentials

When you select **Clear All Saved Credentials**, the following items are removed from your user account:

- All saved credentials for connections that are stored in your account.

  Caution: If any of these saved credentials are stored with published workbooks or data sources, deleting them also removes access to the data source from those locations. Effectively, this is like "changing the locks" anywhere the affected saved credentials are used.

- Passwords you've used to access published data extracts or workbooks that connect to them.

Remove saved credentials

To remove Tableau access to data, delete the associated saved credentials for that data from your account. After you delete the credentials, you'll need to sign in to the data the next time you access it. This creates new saved credentials.

Your administrator might choose for all users to use the same shared credentials for connecting to a data source. If this is the case, the saved credential is associated with the data connection for all users, and it doesn't appear under Saved Credentials on your Account Settings page.

Note: If you're a Tableau Server user and can't delete saved credentials, ask your administrator if they've cleared the Allow users to save data source access tokens option in the server settings.

Create and Edit Private Content in Personal Space

Personal Space is a private location for all Explorers and Creators to save content to when working in a Tableau Site. Content saved in Personal Space can’t be shared with other users but can be moved to a project when you’re ready for others to see it. Within Personal Space, you can create a new workbook or save a workbook to Personal Space as a separate copy. You can also move existing content you own into Personal Space for editing, then move it back to a project later. Explorers can download workbooks in Personal Space, including all data included in the workbook.

Privacy in Personal Space

Content saved to your Personal Space is only visible to you and site administrators. Site administrators can’t directly access any user’s Person Space or edit content in someone else's Personal Space, but they can view and manage Personal Space workbooks. Personal Space workbooks appear in administrator search results and as a workbook location on the Explore page. In addition, the permissions menu is unavailable when a workbook is in Personal Space because the workbook is private.

Tableau Catalog and Personal Space

Starting in 2019.3, Tableau Catalog is available with Data Management in Tableau Server and Tableau Cloud. For more information, see "About Tableau Catalog" in the Tableau Server or Tableau Cloud Help.

When Tableau Catalog is enabled in your work environment, the information about workbooks you save in your Personal Space is indexed by Catalog. These workbooks are included

in lineage counts, however, only you can see the workbooks. Furthermore, users who browse through the lineage tool see Permissions required instead of information about workbooks in your Personal Space.

Collaboration tools

When a workbook is in Personal Space, some functionality is disabled, including share, metrics, comments, alerts, and subscriptions. Existing alerts and subscriptions to you will continue running, but alerts and subscriptions to others will fail, since the content is now private. Metrics can't be created in Personal Space but will continue to work if a connected workbook is moved there. (The legacy Metrics feature was retired in February 2024 for Tableau Cloud and in Tableau Server version 2024.2. For more information, see Create and Troubleshoot Metrics (Retired).)

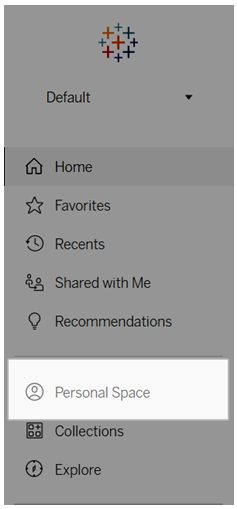
These limitations are removed when the workbook is moved or saved to another location. For example, if a workbook contains comments and moves to Personal Space, existing comments are hidden. Comments restore when the workbook is moved to another location.

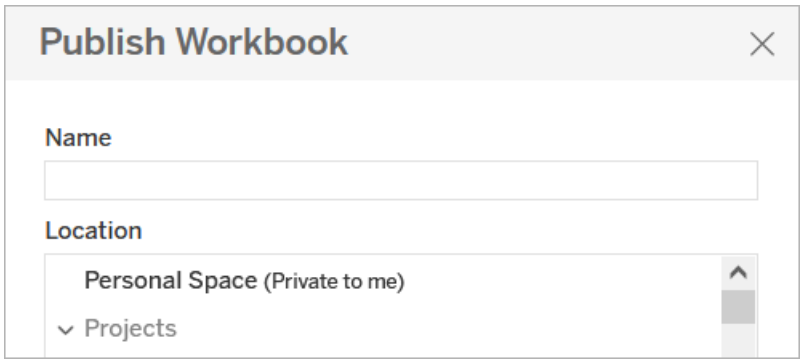
Extract refreshes in Personal Space

To limit resource consumption, existing extract refreshes continue to run if they've been scheduled, but new extract refreshes can't be scheduled while a workbook is in Personal Space.

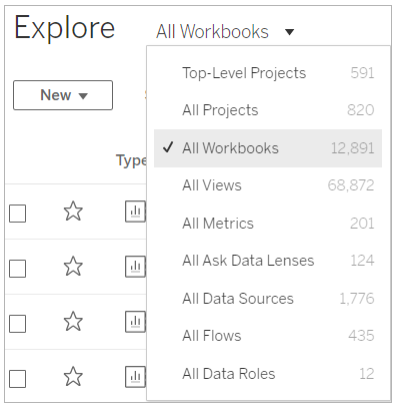
Find content in Personal Space

You can access Personal Space from the left navigation menu to see all your Personal Space content or create a new workbook, and you can save to Personal Space when creating or editing a workbook anywhere on the site.





You can also see workbooks in Personal Space from the Explore page when All Workbooks is selected, and you can filter down to Personal Space content.

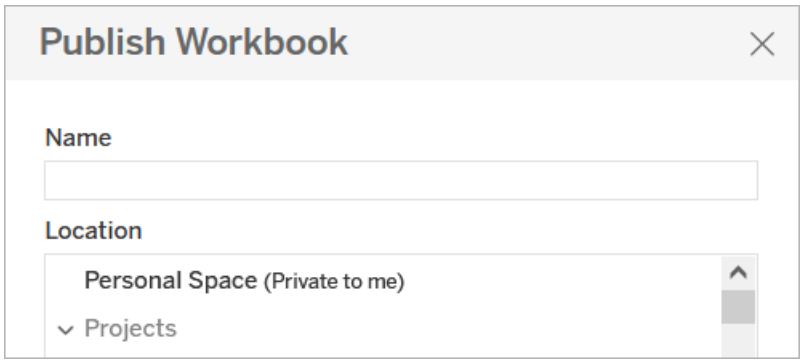


Publish a workbook to Personal Space

Personal Space works much like a private project for you to publish a new or existing workbook to from Tableau Cloud, Tableau Server, or Tableau Desktop.

Publish a workbook to Personal Space on Tableau Server or Tableau Cloud

1. With the workbook open, select **File** > **Publish As**.
2. Under Location, select **Personal Space**.



Note: Explorers can only save workbooks to Personal Space and may not see a location selection dialog.

Publish a workbook to Personal Space from Tableau Desktop

Starting in 2023.1, you can publish a workbook to Personal Space from Tableau Desktop.

1. With the workbook you want to publish open in Tableau Desktop, select **Server** > **Publish Workbook**.
2. Under Project, select **Personal Space**.
3. Under Data Sources, select **Edit**.

4. In the Manage Data Sources popup under Publish type, select **Embedded in workbook** for all data sources. You must embed data sources when publishing from Tableau Desktop, because you can't publish data sources separately to Personal Space.

5. Fill out the remainder of the publishing options as usual. For more information, see Comprehensive Steps to Publish a Workbook.

Move workbooks to Personal Space

You can move an existing workbook to Personal Space if you are the owner of the workbook and there is room in your Personal Space. Personal Space storage limits are set by administrators.

To move a workbook to Personal Space:

- Select a workbook, then click the **Actions** drop-down menu.
- Select **Move**.
- Under Location, Select **Personal Space**.

> **Note:** Explorers can only save workbooks to Personal Space and may not see the move action or location selection dialog.

For more information, see Perform actions in the Manage Web Content help topic.

When you move an existing workbook or data source to Personal Space, tools like share, alerts, and subscriptions become hidden. Existing extract refreshes continue to run if they've been scheduled, but users can't schedule new extract refreshes within their Personal Space.

Existing subscriptions and alerts also continue but can't be edited from Personal Space and will fail if other users are recipients. Existing connected metrics will continue to refresh, but the connected view will not be visible to other users.

Move workbooks from Personal Space

When you move a workbook out of Personal Space, collaboration tools like share, alerts, and subscriptions become visible, and any existing comments reappear.

Use Relationships for Multi-table Data Analysis

Tables that you drag into this canvas use relationships. Relationships are a flexible way to combine data for multi-table analysis in Tableau.

Think of a relationship as a contract between two tables. When you are building a viz with fields from these tables, Tableau brings in data from these tables using that contract to build a query with the appropriate joins.

We recommend using relationships as your first approach to combining your data because it makes data preparation and analysis easier and more intuitive. Use joins only when you absolutely need to. Learn more about the basics of creating relationships in this 5-minute video.

> **Note**: The interface for editing relationships shown in this video differs slightly from the current release but has the same functionality.

Learn more about how relationships work in these Tableau blog posts:

- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

Also see video podcasts on relationships from Action Analytics, such as Why did Tableau Invent Relationships? Click "Video Podcast" in the Library to see more.

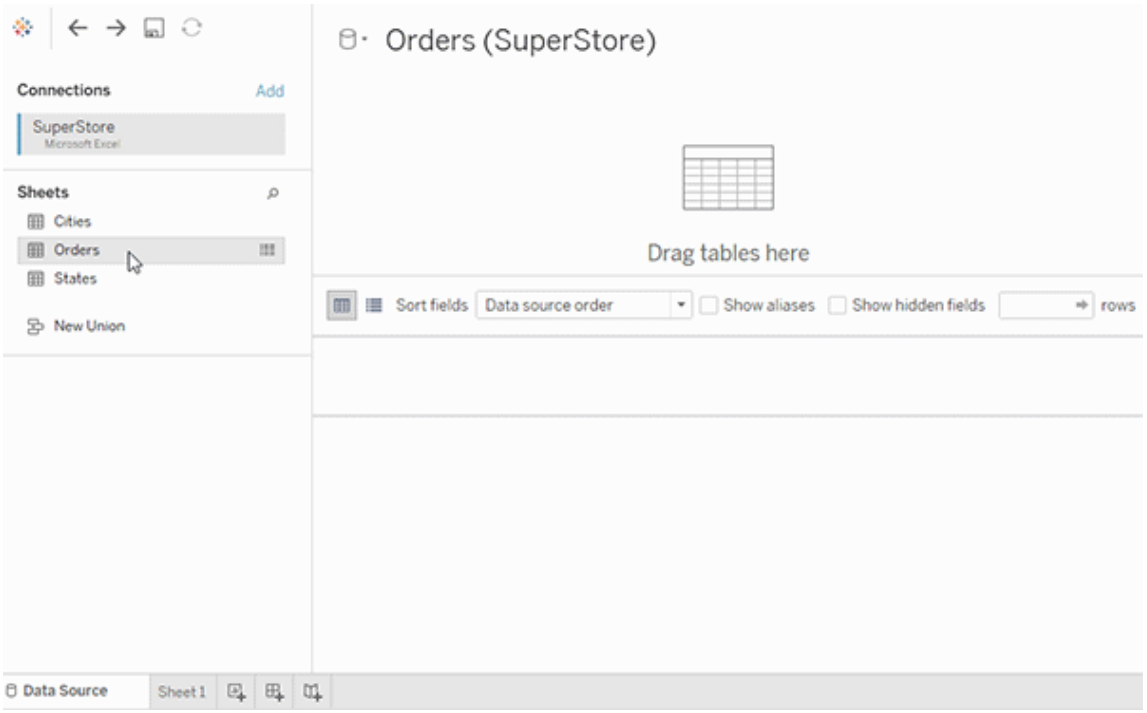
In Tableau version 2024.2 and later, the Tableau data model supports multi-fact analysis and shared dimensions through multi-fact relationships. For more information, see About Multi-fact

Relationship Data Models, When to Use a Multi-fact Relationship Model, and Build a Multi-fact Relationship Data Model.

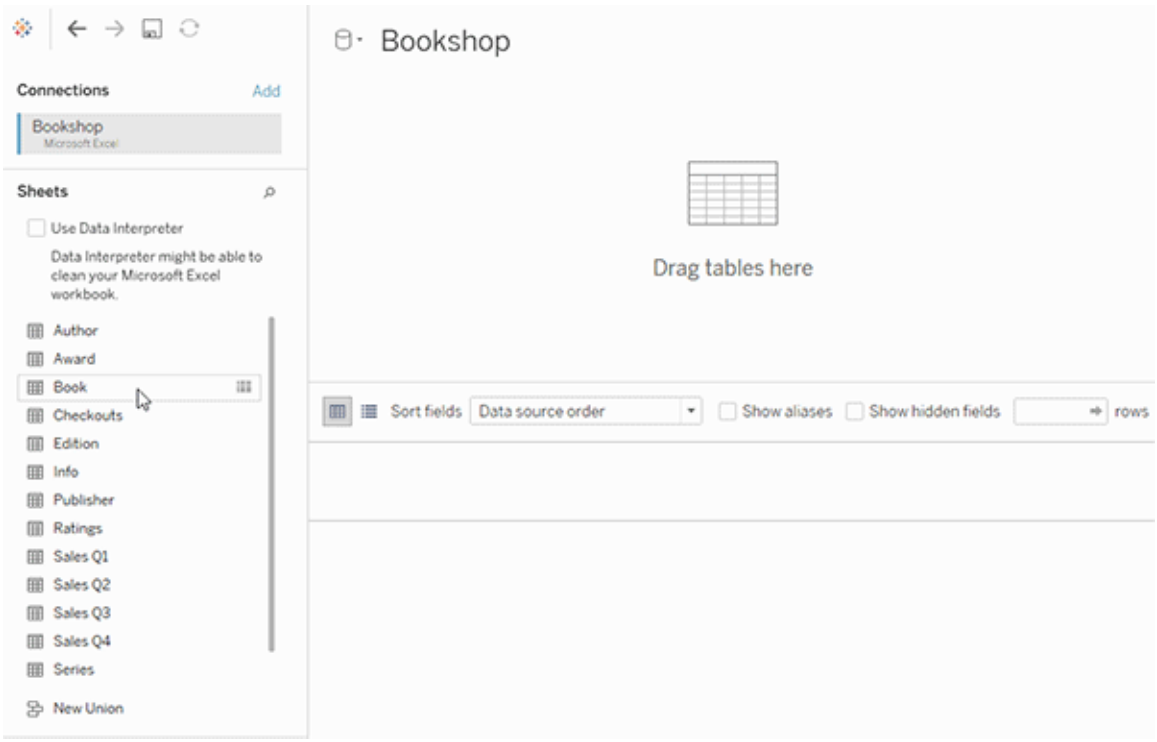
Are you building a new data source and workbook?

Drag a table to the Data Source page canvas to start building your data source.

A data source can be made of a single table that contains all of the dimension and measure fields you need for analysis...



Or, you can create a multi-table data source by dragging out more tables and defining their relationships...



Watch this 1-minute video about getting started with using relationships.

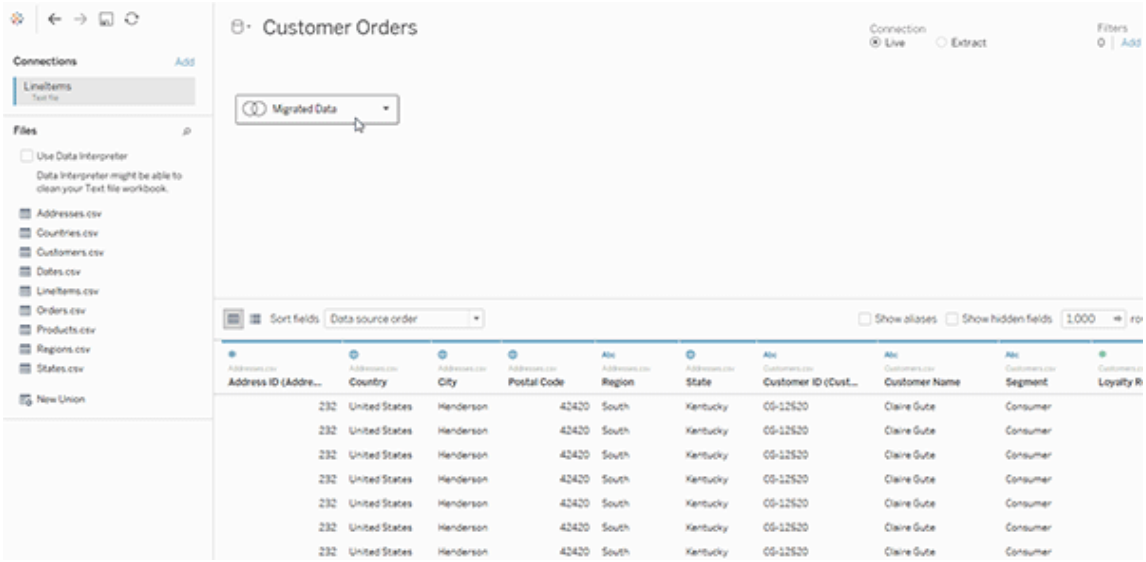
Note: The interface for editing relationships shown in this video differs slightly from the current release but has the same functionality.

For more information about using relationships, see Relate Your Data, How Relationships Differ from Joins, The Tableau Data Model, and Create and define relationships.

For more information on changes to data sources and analysis in Tableau 2020.2 and later, see What's Changed with Data Sources and Analysis and Questions about Relationships, the Data Model, and Data Sources.

Are you opening an older workbook or data source?

When you open a pre-2020.2 workbook or data source in 2020.2, your data source will appear as a single logical table in the canvas, with the name "Migrated Data" or the original table name. Your data is preserved and you can continue to use the workbook as you did before.



To see the physical tables that make up the single logical table, double-click that logical table to open it in the physical layer. You will see its underlying physical tables, including joins and unions.

For more information on changes to data sources and analysis in Tableau 2020.2 and later, see What's Changed with Data Sources and Analysis and Questions about Relationships, the Data Model, and Data Sources.

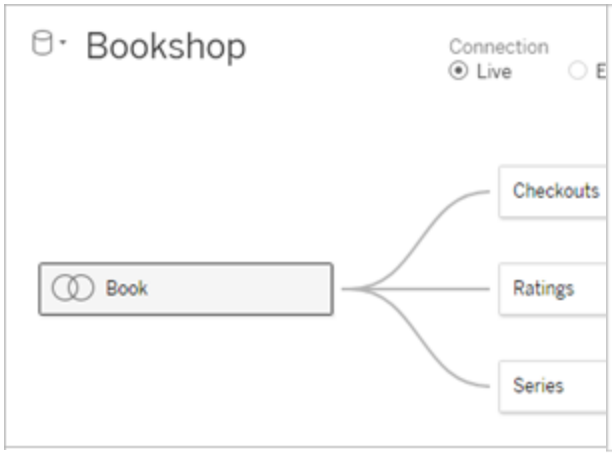
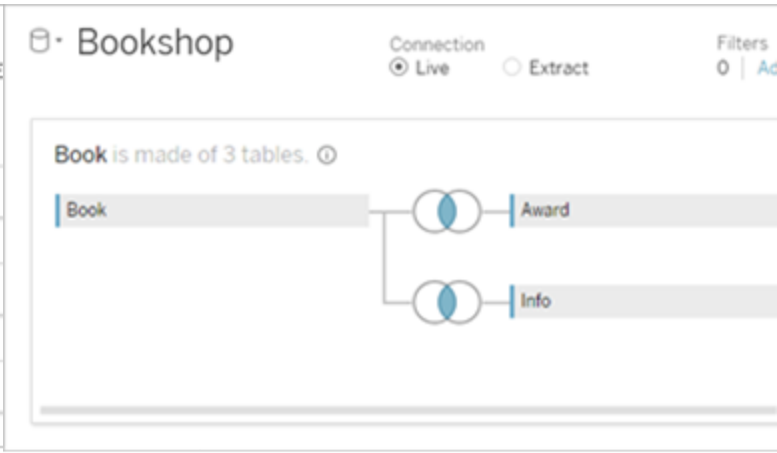
The Tableau Data Model

Every data source that you create in Tableau has a data model. You can think of a data model as a diagram that tells Tableau how it should query data in the connected database tables.

The tables that you add to the canvas in the **Data Source** page create the structure of the data model. A data model can be simple, such as a single table. Or it can be more complex, with multiple tables that use different combinations of relationships, joins, and unions.

The data model has two layers:

- The default view that you first see in the Data Source page canvas is the *logical layer* of the data source. You combine data in the logical layer using relationships (or noodles). Think of this layer as the Relationships canvas in the Data Source page. For more information, see Use Relationships for Multi-table Data Analysis.
- The next layer is the *physical layer*. You combine data between tables at the physical layer using joins and unions. Each logical table contains at least one physical table in this layer. Think of the physical layer as the Join/Union canvas in the Data Source page. Double-click a logical table to view or add joins and unions.

Logical Layer	**Physical Layer**
Noodles = Relationships	Venn diagram = Joins
	
The top-level view of a data source with multiple, related tables. This is the logical layer. Logical tables can be combined using relationships (noodles). They don't use join types. They act like containers for physical tables.	Double-click a logical table to open it and see its physical tables. Physical tables can be combined using joins or unions. In this example, the Book logical table is made of three, joined physical tables (Book, Award, Info).

Logical Layer	Physical Layer
Relationships canvas in the Data Source page	Join/Union canvas in the Data Source page
Tables that you drag here are called logical tables	Tables that you drag here are called physical tables
Logical tables can be related to other logical tables	Physical tables can be joined or unioned to other physical tables
Logical tables are like containers for physical tables	Double-click a logical table to see its physical tables
Level of detail is at the row level of the logical table	Level of detail is at the row level of merged physical tables
Logical tables remain distinct (normalized), not merged in the data source	Physical tables are merged into a single, flat table that defines the logical table

Layers of the data model

The top-level view that you see of a data source is the logical layer of the data model. You can also think of it as the Relationships canvas, because you combine tables here using relationships instead of joins.

When you combine data from multiple tables, each table that you drag to the canvas in the logical layer must have a relationship to another table. You do not need to specify join types for relationships; during analysis Tableau automatically selects the appropriate join types based on the fields and context of analysis in the worksheet.

The physical layer of the data model is where you can combine data using joins and unions. You can only use pivots in this canvas. You can think of it as the Join/Union canvas. In previous versions of Tableau, the physical layer was the only layer in the data model. Each logical table can contain one or more physical tables.

Important: You can still create single-table data sources in Tableau that use joins and unions. The behavior of single-table analysis in Tableau has not changed. Your upgraded workbooks will work the same as they did before 2020.2.

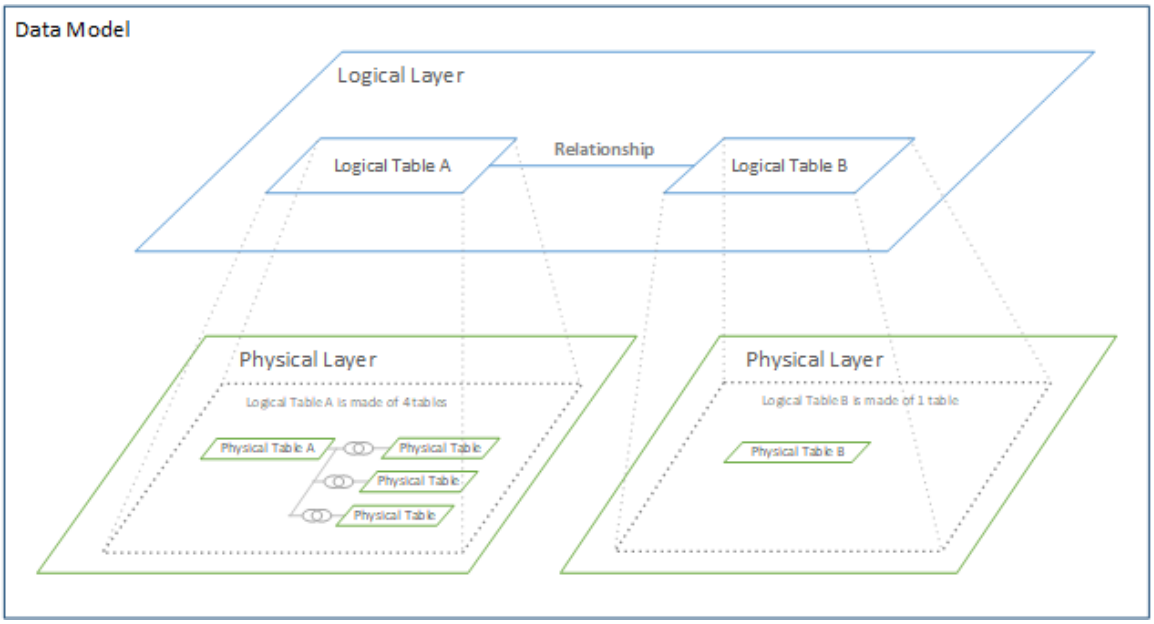
Learn more: For related information on combining data using relationships, also see these topics and blog posts:

- How Relationships Differ from Joins
- Use Relationships for Multi-table Data Analysis
- Relate Your Data
- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

Also see video podcasts on relationships from Action Analytics, such as Why did Tableau Invent Relationships? Click "Video Podcast" in the Library to see more.

Understanding the data model

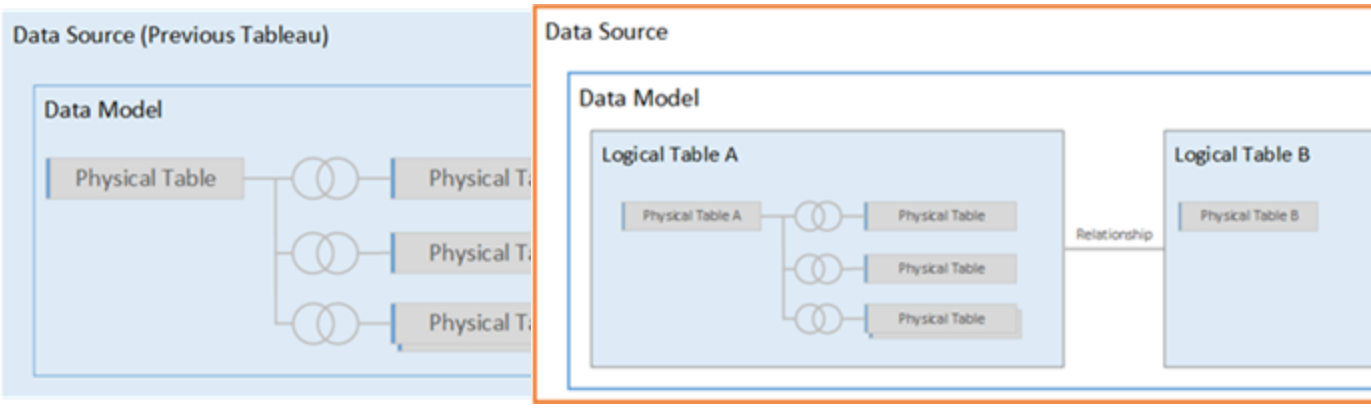
In previous versions of Tableau (pre-2020.2), the data model had a physical layer only. In Tableau 2020.2 and later, the data model has the logical (semantic) layer and a physical layer. This gives you more options for combining data using schemas to fit your analysis.



In Tableau 2020.2 and later, a logical layer has been added in the data source. Each logical table contains physical tables in a physical layer.

In earlier versions of Tableau (pre-2020.2), the data model in your data source consisted of a single, physical layer where you could specify joins and unions. Tables added to the physical layer (joined or unioned) create a single, flattened table (denormalized) for analysis.

Previous versions **2020.2 and later**



In versions of Tableau before 2020.2, the data model has only the physical layer

In 2020.2 and later, the data model has two layers: the logical layer and the physical layer

In Tableau 2020.2 and later, the data model in your data source includes a new semantic layer above the physical layer—called the logical layer—where you can add multiple tables and relate them to each other. Tables at the logical layer are not merged in the data source, they remain distinct (normalized), and maintain their native level of detail.

Logical tables act like containers for merged physical tables. A logical table can contain a single, physical table. Or it can contain multiple physical tables merged together through joins or unions.

Build a new model

When you add one or more tables to the logical layer, you are essentially building the data model for your data source. A data source can be made of a single, logical table, or you can

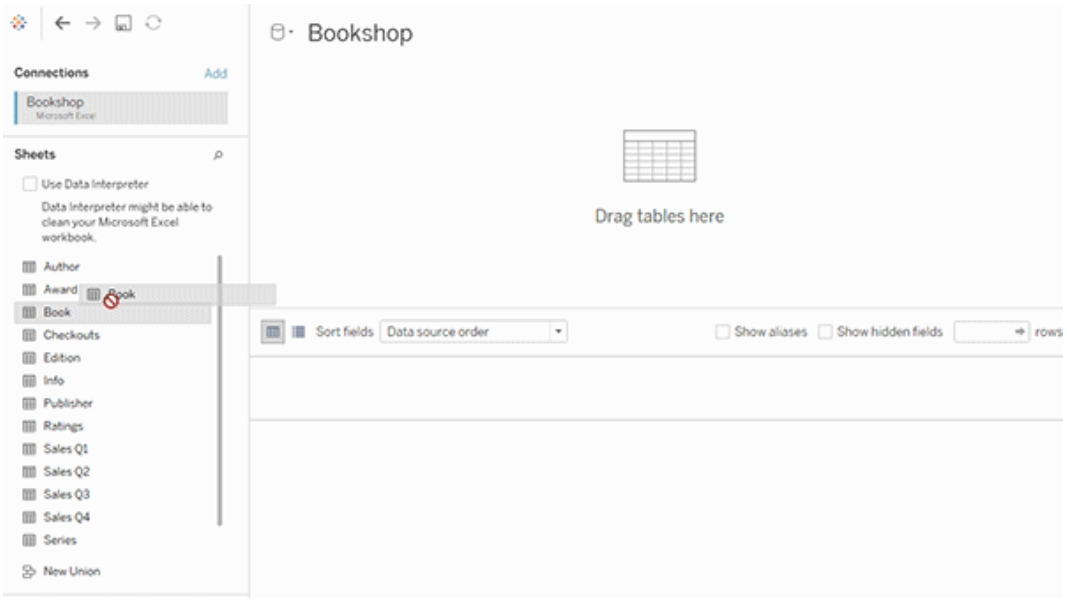
drag multiple tables to the canvas to create a more complex model.

- The first table that you drag to the canvas becomes the root table for the data model in your data source.
- After you drag out the root table, you can drag out additional tables in any order. You will need to consider which tables should be related to each other, and the matching field pairs that you define for each relationship.
- If you are creating a star schema, it can be helpful to drag the fact table out first, and then relate dimension tables to that table.
- Deleting a table in the canvas automatically deletes its related descendants as well. If you delete the root table, all other tables in the model are also removed.
- Each relationship must be made of at least one matched pair of fields. Add multiple field pairs to create a compound relationship. Matched pairs must have the same data type. Changing the data type in the Data Source page does not change this requirement. Tableau will still use the data type in the underlying database for queries.
- Relationships can be based on calculated fields.
- You can specify how fields used in the relationships should be compared by using operators when you define the relationship.

For more information about relationships, see Create and define relationships in Relate Your Data.

Multi-table model

- To create a multi-table model, drag tables to the logical layer of the Data Source page canvas.



Tables that you drag to the logical layer of the Data Source page canvas must be related to each other. When you drag additional tables to the logical layer canvas, Tableau automatically attempts to create the relationship based on existing key constraints and matching fields to define the relationship. If it can't determine the matching fields, you will need to select them.

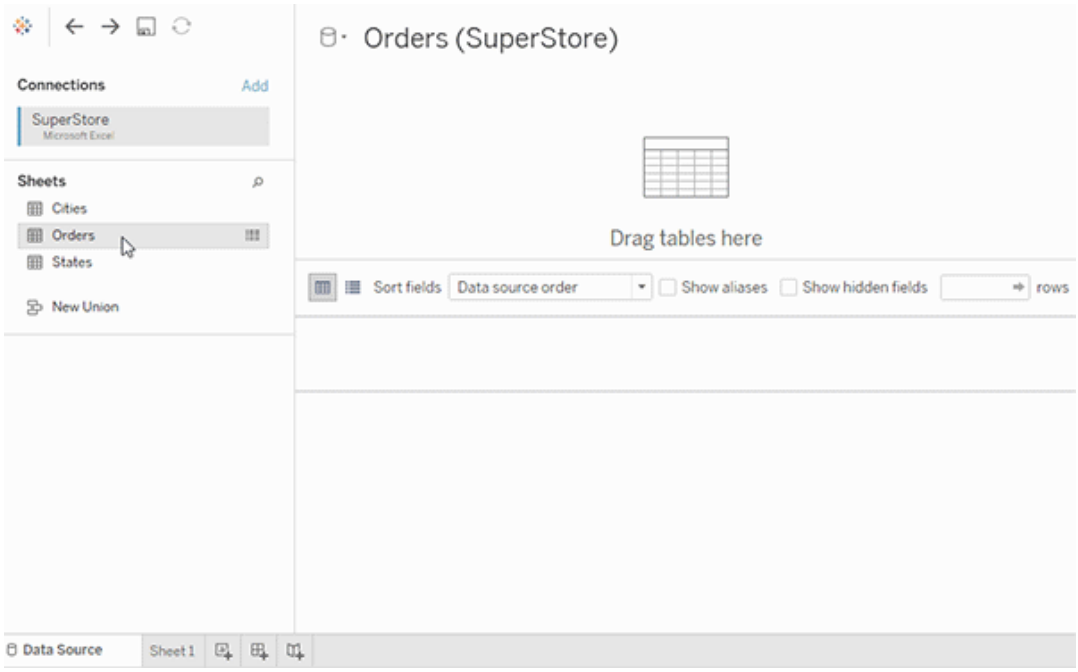
If no constraints are detected, a **Many-to-many** relationship is created and referential integrity is set to **Some records match**. These default settings are a safe choice and provide the most a lot of flexibility for your data source. The default settings support full outer joins and optimize queries by aggregating table data before forming joins during analysis. All column and row data from each table becomes available for analysis.

You can add more data inside any logical table by double-clicking the table. This opens the physical layer of the Data Source page canvas. If you need to use joins or unions, you can drag the tables you want to join or union into the physical layer canvas. The physical tables are merged in their logical table.

Follow the steps in Create and define relationships to combine multiple tables.

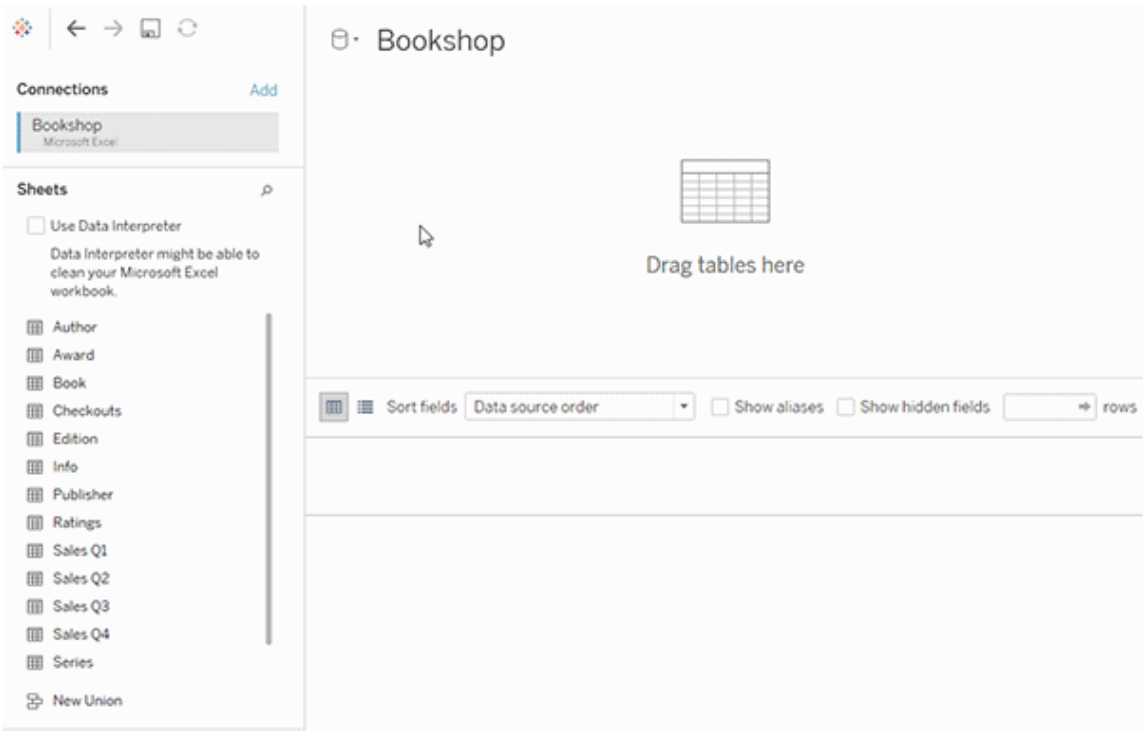
Single-table model

- To create a single-table model, drag a table into the logical layer canvas of the Data Source page. You can then use the fields from that table in the Data pane for analysis.



Single-table model that contains other tables

You can add more data inside the single, logical table by double-clicking the table. This opens the physical layer of the Data Source page canvas. If you need to use joins or unions, you can drag the tables you want to join or union into the physical layer canvas. The physical tables are merged in their logical table.



This example shows the Book table in the Relationships canvas (logical layer) of the data source. Double-clicking the Book logical table opens the Join/Union canvas (physical layer).

In this example, the joins merge the Award and Info tables with the Book table. In this case, the join between Book and Award will be one-to-many, at the level of detail of awards. This would duplicate measure values for Book and Info. To avoid duplication, you could relate Award and Info to Book instead of joining them inside of the Book logical table.

Supported data model schemas

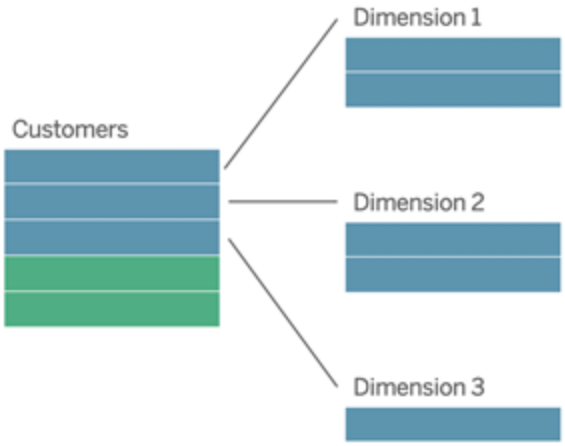
The data modeling capabilities in Tableau (version 2020.2 and later) are designed to make analysis over common multi-table data scenarios—including star and snowflake data models—easy. The following types of models are supported in Tableau data sources.

Single-table

Analysis over a single logical table that contains a mixture of dimensions and measures works just as in Tableau pre-2020.2. You can build a logical table using a combination of joins, unions, custom SQL, and so on.

Star and snowflake

In enterprise data warehouses, it is common to have data structured in star or snowflake schemas where measures are contained in a central fact table and dimensions are stored separately in independent dimension tables. This organization of data supports many common analysis flows including rollup and drill down.



These models can be directly represented with relationships in the data modeling capabilities available in Tableau 2020.2 and later.

Drag the fact table into the model first and then relate the dimension tables to the fact table (in a star schema) or to other dimension tables (in a snowflake).

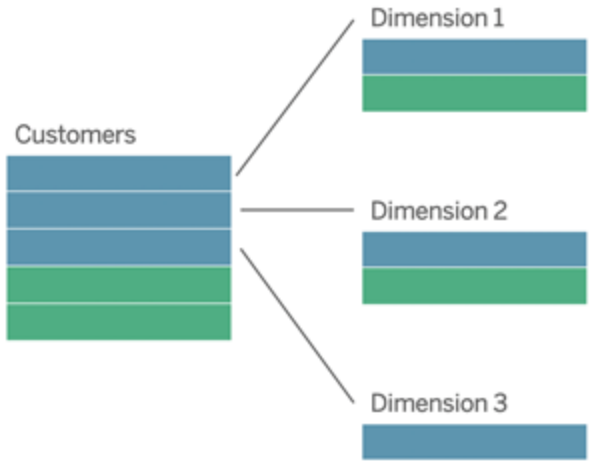
Typically, in a well-modeled star or snowflake schema, the relationships between the fact table and the dimension tables will be many-to-one. If this information is encoded in your data warehouse, Tableau will automatically use this to set the relationship's Performance Options. If not,

you can set this information yourself. For more information, see Optimize Relationship Queries Using Performance Options.

In a well-modeled star or snowflake schema, every row in the fact table will have a matching entry in each of the dimension tables. If this is true and captured in your data warehouse integrity constraints, Tableau will automatically use this information to set the referential integrity setting in Performance Options. If some fact table rows do not have a matching row in a dimension table (sometimes called “late-arriving dimensions” or “early-arriving facts”), Tableau will default to retaining all rows when computing measures, but may drop values when showing dimension headers. For more information, see Optimize Relationship Queries Using Performance Options.

Star and snowflake with measures in more than one table

In some star or snowflake schemas, all the measures for your analysis are contained in the fact table. However, it is often true that additional measures of interest may be related to the dimension tables in your analysis. Even if the dimension tables do not contain measures, it is common in analysis to want to count or otherwise aggregate dimension values. In these cases, the distinction between fact tables and dimension tables is less clear. To create clarity when viewing your data model, we recommended adding the finest grain table to the data source canvas first, and then relating all other tables to that first table.



If you were to join these tables together into a single logical table, the measures in the dimension tables would be replicated, resulting in distorted aggregates unless you took precautions to deduplicate the values using LOD calculations or COUNT DISTINCT. However, if you instead create relationships between these tables, Tableau will aggregate measures before performing joins, avoiding the problem of unnecessary duplication. This relieves you of the need to carefully track the level of detail of your measures.

Multi-fact analysis

In version 2024.2 and later, Tableau's data modeling capabilities support multi-fact analysis through the use of multi-fact relationships. For in-depth information on how to create multi-fact relationships data models, see:

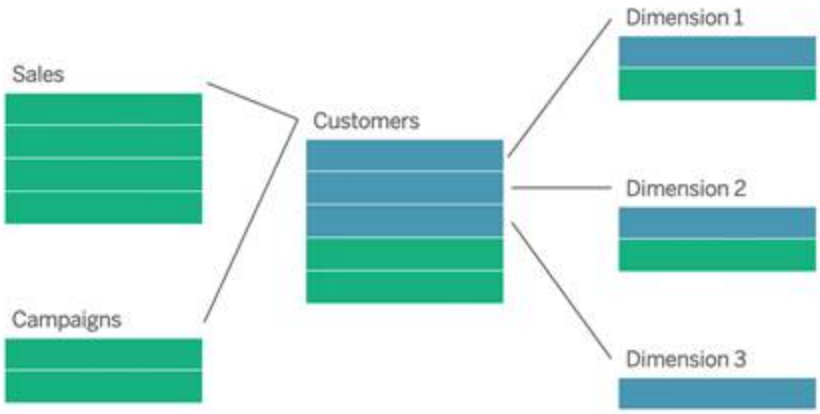
- About Multi-fact Relationship Data Models
- When to Use a Multi-fact Relationship Model
- Build a Multi-fact Relationship Data Model

A multi-fact relationship model—a data model with multiple base tables—permits unrelated tables in the model when shared tables also exist in the model. During analysis, fields from a shared table "stitch" together otherwise unrelated tables of data based on the shared dimensions they have in common (such as happening in the same place or at the same time). All the benefits of relationships are maintained, including the retention of each table's grain, or native level of detail.

Similar to a single base table data model, Tableau determines the best join type to use behind the scenes based on the structure of the viz. But in a multi-fact relationship model, the join options are expanded to include outer and cross joins to handle different levels of relatedness. For more information, see About Multi-fact Relationship Data Models.

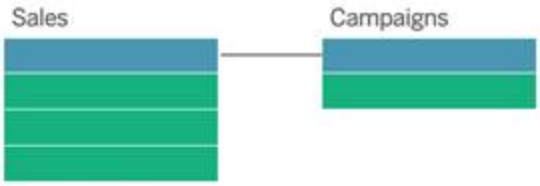
Note: In versions 2020.2 through 2024.1, you can add fact tables (containing measures) to star and snowflake models only if they are related to a single dimension table.

For example, you can bring two or more fact tables together to analyze a shared dimension, such as in Customer 360-like analyses. These fact tables can be at a different level of detail than the dimension table, or from each other. They can also have a many-to-many relationship with the dimension table. In these scenarios, Tableau will ensure that values are not replicated before aggregation.



If you don’t have a shared dimension table that relates your fact tables, you can sometimes dynamically build one using custom SQL or by using joins or unions of other dimension tables.

Two fact tables can be related directly to each other on a common dimension. This type of analysis works best when one of the fact tables contains a superset of the common dimension.



There are various scenarios that may indicate you should build a multi-fact relationship model with multiple base tables rather than a single base table data source:

- **Circular relationships**. Circular relationships are not supported. If you're trying to build a data source with a cycle, use multi-fact relationships and make the downstream

table another base table instead.

- **Conformed dimensions** and **Contextual OR relationships**. If you have a series of tables that are related on the same sets of relationship clauses (such as date and location), those dimensions should be pulled out and made into shared tables instead.
    - This is especially useful because multiple relationship clauses must all be true (logically, an AND) for the tables to be related for those records.
    - If, instead, you want to analyze records where one may be true at a time (a contextual OR), this flexibility is provided by setting up a data model with shared dimension tables instead.
- **Equivalent blends**. If you're using a blend but want to have an equivalent blend without primary and secondary data sources, build a data model that combines the data sources from the blend with their linking fields in a shared table or tables.

Requirements for relationships in a data model

- When relating tables, the fields that define the relationships must have the same data type. Changing the data type in the Data Source page does not change this requirement. Tableau will still use the data type in the underlying database for queries.
- You can't define relationships based on geographic fields.
- Circular relationships aren't supported in the data model.
- You can't define relationships between published data sources.

Factors that limit the benefits of using related tables

- Dirty data in tables (i.e. tables that weren't created with a well-structured model in mind and contain a mix of measures and dimensions in multiple tables) can make multi-table analysis more complex.
- Using data source filters will limit Tableau's ability to do join culling in the data. Join culling is a term for how Tableau simplifies queries by removing unnecessary joins.
- Tables with a lot of unmatched values across relationships.
- In versions 2020.2 through 2024.1: Interrelating multiple fact tables with multiple dimension tables (attempting to model shared or conformed dimensions). In version 2024.2 and later, you can use multi-fact relationships to address these cases.

How Relationships Differ from Joins

Relationships are a dynamic, flexible way to combine data from multiple tables for analysis. You don’t define join types for relationships, so you won’t see a Venn diagram when you create

them.

Think of a relationship as a contract between two tables. When you are building a viz with fields from these tables, Tableau brings in data from these tables using that contract to build a query with the appropriate joins.

- **No up-front join type**. You only need to select matching fields to define a relationship (no join types). Tableau first attempts to create the relationship based on existing key constraints and matching field names. You can then check to ensure they are the fields you want to use, or add more field pairs to better define how the tables should be related.
- **Automatic and context-aware**. Relationships defer joins to the time and context of analysis. Tableau automatically selects join types based on the fields being used in the visualization. During analysis, Tableau adjusts join types intelligently and preserves the native level of detail in your data. You can see aggregations at the level of detail of the fields in your viz rather than having to think about the underlying joins. You don't need to use LOD expressions such as FIXED to deduplicate data in related tables.
- **Flexible**. Relationships can be many-to-many and support full outer joins. When you combine tables using relationships, it's like creating a custom, flexible data source for every viz, all in a single data source for the workbook. Because Tableau queries only tables that are needed based on fields and filters in a viz, you can build a data source that can be used for a variety of analytic flows.

For more information, see Relate Your Data and Don't Be Scared of Relationships.

Joins are still available as an option for combining your data. Double-click a logical table to go to the join canvas. For more information, see Where did joins go?

Watch a video: For an introduction to using relationships in Tableau, see this 5-minute video.

Note: The interface for editing relationships shown in this video might differ slightly from the current release but has the same functionality.

Also see video podcasts on relationships from Action Analytics, such as Why did Tableau Invent Relationships? Click "Video Podcast" in the Library to see more.

For related information about how relationship queries work, see these Tableau blog posts:

- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

Characteristics of relationships and joins

Relationships are a dynamic, flexible way to combine data from multiple tables for analysis. We recommend using relationships as your first approach to combining your data because it makes data preparation and analysis easier and more intuitive. Use joins only when you absolutely need to.

Here are some advantages to using relationships to combine tables:

- Make your data source easier to define, change, and reuse.
- Make it easier to analyze data across multiple tables at the correct level of detail (LOD).
- Do not require the use of LOD expressions or LOD calculations for analysis at different levels of detail.
- Only query data from tables with fields used in the current viz.

Relationships

- Are displayed as flexible noodles between logical tables
- Require you to select matching fields between two logical tables
- Do not require you to select join types
- Make all row and column data from related tables potentially available in the data source
- Maintain each table's level of detail in the data source and during analysis
- Create independent domains at multiple levels of detail. Tables aren't merged together in the data source.
- During analysis, create the appropriate joins automatically, based on the fields in use.
- Do not duplicate aggregate values (when Performance Options are set to Many-to-Many)
- Keep unmatched measure values (when Performance Options are set to Some Records Match)

Joins

Joins are a more static way to combine data. Joins must be defined between physical tables up front, before analysis, and can't be changed without impacting all sheets using that data source. Joined tables are always merged into a single table. As a result, sometimes joined data is missing unmatched values, or duplicates aggregated values.

- Are displayed with Venn diagram icons between physical tables
- Require you to select join types and join clauses
- Joined physical tables are merged into a single logical table with a fixed combination of data
- May drop unmatched measure values
- May duplicate aggregate values when fields are at different levels of detail
- Support scenarios that require a single table of data, such as extract filters and aggregation

Requirements for using relationships

- When relating tables, the fields that define the relationships must have the same data type. Changing the data type in the Data Source page does not change this requirement. Tableau will still use the data type in the underlying database for queries.
- You can't define relationships based on geographic fields.
- Circular relationships aren't supported in the data model.
- You can't define relationships between published data sources.

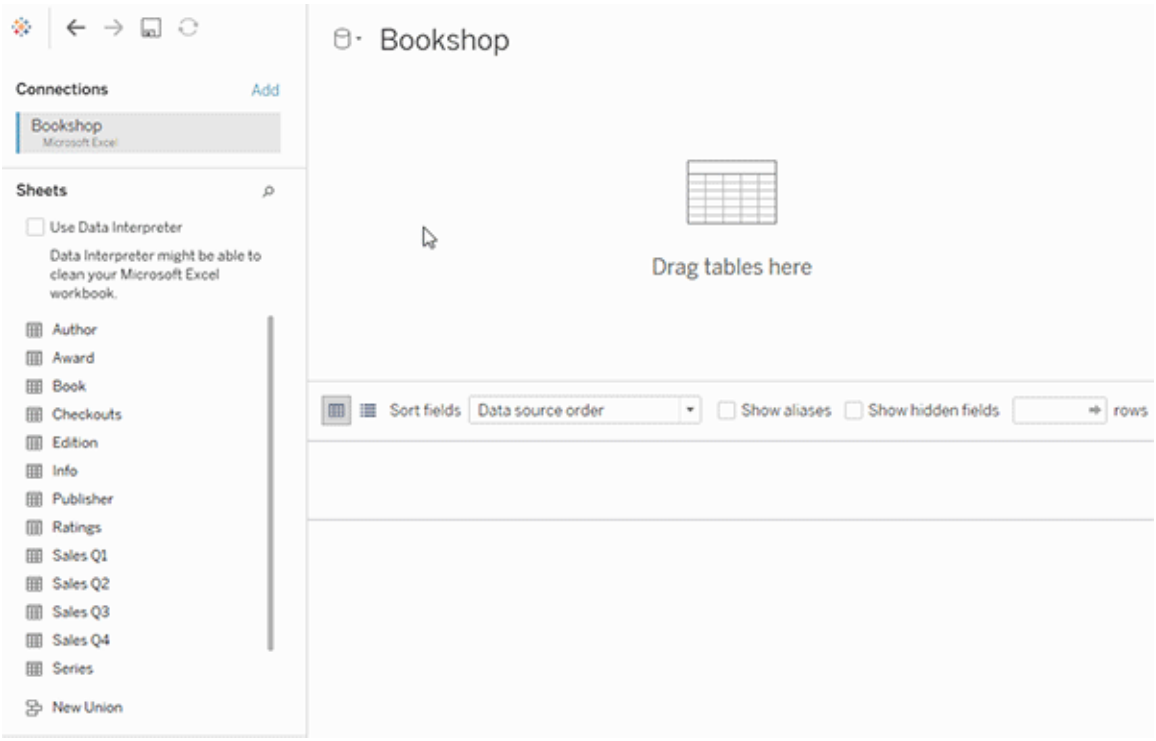
Factors that limit the benefits of using related tables

- Dirty data in tables (i.e. tables that weren't created with a well-structured model in mind and contain a mix of measures and dimensions in multiple tables) can make multi-table analysis more complex.
- Using data source filters will limit Tableau's ability to do join culling in the data. Join culling is a term for how Tableau simplifies queries by removing unnecessary joins.
- Tables with a lot of unmatched values across relationships.
- In versions 2020.2 through 2024.1: Interrelating multiple fact tables with multiple dimension tables (attempting to model shared or conformed dimensions). In version 2024.2 and later, you can use multi-fact relationships to address these cases. For more information, see Multi-fact analysis with relationships and About Multi-fact Relationship Data Models.

Where did joins go?

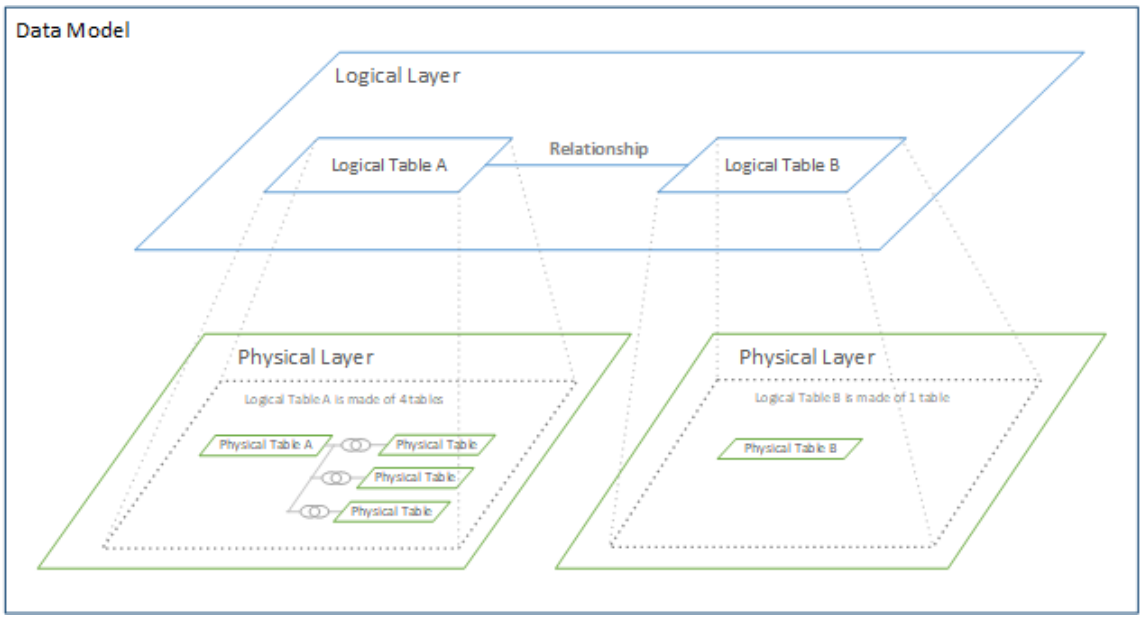
You can still specify joins between tables in the physical layer of a data source. Double-click a logical table to go to the Join/Union canvas in the physical layer and add joins or unions.

Every top-level, logical table contains at least one physical table. Open a logical table to view, edit, or create joins between its physical tables. Right-click a logical table, and then click **Open**. Or, just double-click the table to open it.



When you create a data source, it has two layers. The top-level layer is the logical layer of the data source. You combine data between tables in the logical layer using relationships.

The next layer is the physical layer of the data source. You combine data between tables at the physical layer using joins. For more information, see Logical and physical tables in the data model.



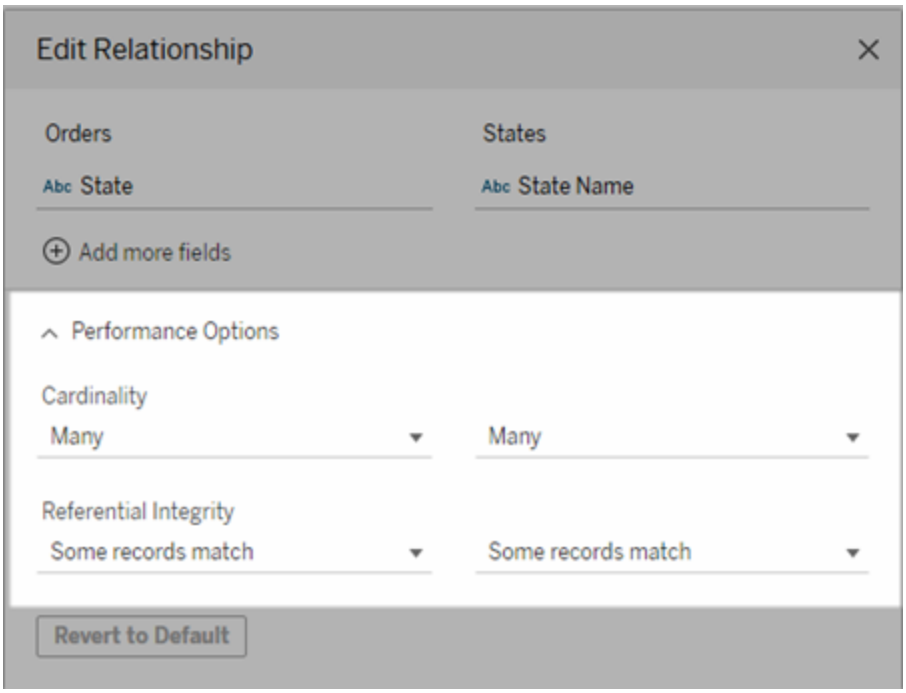
Optimize Relationship Queries Using Performance Options

Performance Options are optional settings that define the cardinality (uniqueness) and referential integrity (matching records) between the two tables in a relationship. These settings help Tableau optimize queries during analysis.

- **If you aren't sure what to choose**, use the recommended default settings indicated by Tableau. Using the defaults is safe and will automatically generate correct aggregations and joins during analysis. If you don't know the cardinality or referential integrity, you don't need to change these settings.
- **If you know the shape of your data**, you can optionally change these settings to represent uniqueness and matching between the records in the two tables.

In many analytical scenarios, using the default settings for a relationship will give you all of the data you need for analysis. In some scenarios, you might want to adjust the Performance Options settings to describe your data more accurately. For more details about using relationships to combine and analyze data, see Relate Your Data and this Tableau blog post: Relationships, part 1: Introducing new data modeling in Tableau.

What the Cardinality and Referential Integrity settings mean



Cardinality options

Cardinality settings determine if Tableau aggregates table data before or after automatically joining the data during analysis.

- Select **Many** if the field values aren't unique, or you don't know. Tableau will aggregate the relevant data before forming joins during analysis.
- Select **One** if field values are unique. During analysis, the relevant data will be joined before aggregation. Setting this option correctly optimizes queries in the workbook when the field values in the relationship are unique. However, selecting **One** when field values aren’t unique can result in duplicate aggregate values being shown in the view.

Note: Selecting **One** treats records as if each key value is unique and there is at most only one row with a null value.

Referential Integrity options

Referential Integrity settings determine the type of join used to get the dimension values for a measure during analysis.

- Select **Some Records Match** if some values in the field don't have a match in the other table, or you don't know. During analysis, Tableau uses outer joins to get dimensions values for a measure. All measure values will be shown in the view, even unmatched measures.
- Select **All Records Match** if values in the field are guaranteed to have a match in the other table. This setting generates fewer and simpler joins during analysis, and optimizes queries. You might see inconsistent results during analysis (unmatched values removed or missing in view) if there are unmatched values in this table.

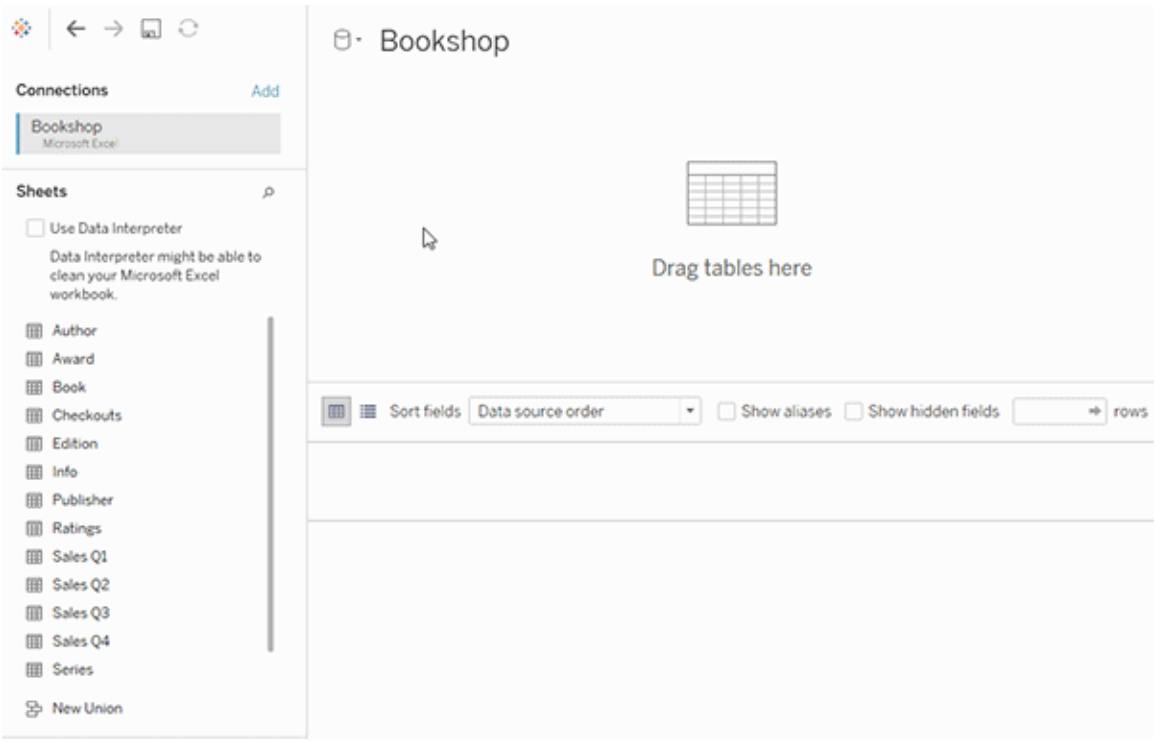
Notes: Selecting **All Records Match** treats records as if no null values exist in the fields used for the relationship. During analysis, Tableau will use inner joins to get dimension values for a measure. By default, Tableau will never join null keys.

For more information about Cardinality and Referential integrity as concepts, see Cardinality and Referential Integrity.

Where did joins go?

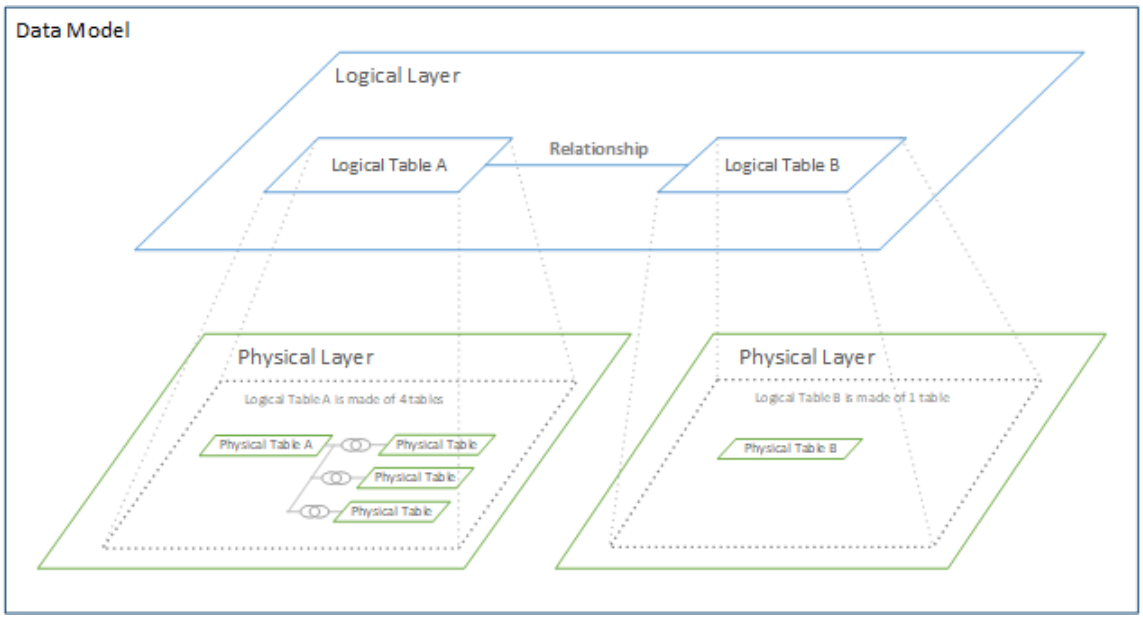
You can still specify joins between tables in the physical layer of a data source. Double-click a logical table to go to the join canvas.

Every top-level, logical table contains at least one physical table. Open a logical table to view, edit, or create joins between its physical tables. Right-click a logical table, and then click **Open**. Or, just double-click the table to open it.



When you create a data source, it has two layers. The top-level layer is the logical layer of the data source. You combine data between tables in the logical layer using relationships.

The next layer is the physical layer of the data source. You combine data between tables at the physical layer using joins. For more information, see Logical and physical tables in the data model.



Tips on using Performance Options

If you know the shape of your data, you can use the optional settings in Performance Options to establish the cardinality of the tables to each other (one-to-one, one-to-many, many-to-many) and indicate referential integrity (values from one table will always have match in the other table).

Instead of thinking of the settings in Performance Options as “yes” and “no”, think of them as “yes” and “I don’t know”. If you are sure that a table's values are unique, select **One**. If you are sure that each record in one table matches one or more records in the other table, select **All Records Match**. Otherwise, leave the default settings as they are.

If you aren't sure about the shape of your data, use the default settings. When Tableau can't detect these settings in your data, the default settings are:

- Cardinality: Many-to-Many
- Referential integrity: Some Records Match

If Tableau detects key relationships or referential integrity in your data, those settings will be used and indicated as "detected".

To reapply the default settings, click **Revert to Default**.

Terms defined

Cardinality refers to the uniqueness of data contained in a field (column) or combination of fields. When the tables you want to analyze contain many rows of data, queries can be slow (and performance of the overall data source is affected) so we recommend choosing a method for combining data based on the cardinality of the related columns between tables.

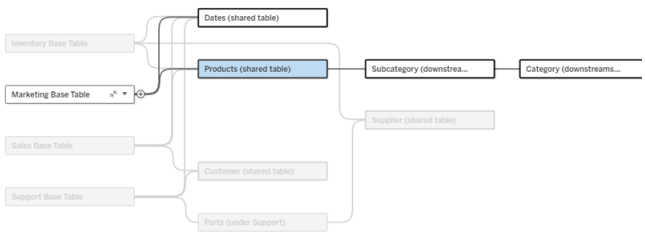
- Low cardinality: When related columns have a lot of repeated data. For example, a table called Products might contain a Category column that contains three values: Furniture, Office Supplies, and Technology.
- High cardinality: When related columns have highly unique data. For example, a table called Orders might contain an Order ID column that contains a unique value for every order of product.

Referential integrity means that one table will always have a matching row in the other table. For example, a Sales table will always have a matching row in the Product Catalog table.

About Multi-fact Relationship Data Models

Multi-fact relationships let you build data sources with more than one *base table*. Using multiple base tables in your data model allows you to perform multi-fact analysis in Tableau.

By establishing *trees* of tables, rooted in a base table, you can model data structures with different conceptual domains and use their shared characteristics to connect them. This type of analysis is often referred to as multi-fact analysis, conformed dimensions, or shared dimensions. In Tableau, we call this a multi-fact relationship data model because you use relationships to build it. A multi-fact relationship data model always contains multiple base tables. Base tables are the left-most tables in the data model. For guidance on how to determine which tables to use as base tables, see When to Use a Multi-fact Relationship Model.



A multiple base table data model with one base table's tree highlighted.

Levels of relatedness

Data models with multiple base tables have a lot of flexibility to how pieces of data can relate—or not relate—to each other.

Note: Relatedness at any level is only relevant in data models with multiple base tables. Prior to multi-fact relationship data models, either everything was related (within a single data source) or nothing was (blending across multiple data sources).

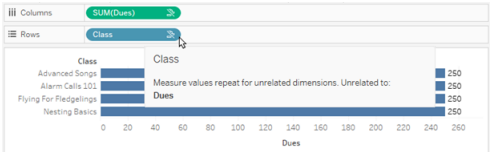
Relatedness in the data model

Tables are related, unrelated, or shared based on the structure of the data model. In a data source, the relatedness of tables is a constant. As a brief overview:

- *Related tables* are in the same tree.
    - Prior to 2024.2, all data sources were single base table data sources consisting of a single tree, and in a single base table data source all tables are related.
- *Unrelated tables* are in different trees. Base tables are always unrelated to each other. Tables that are downstream of exactly one base table are also unrelated to tables in other trees.
- *Shared tables* have multiple incoming relationships and belong to more than one tree.
    - Tables downstream from a table with multiple incoming relationships are also considered shared.

Relatedness during analysis

Fields can be related, unrelated, not yet related, ambiguously related, or they can act as stitching fields. The relatedness among a group of fields is determined on a sheet-by-sheet basis based on the structure of the data model, what fields are actively in use (that is, on the shelves as pills), and if those fields are dimensions or measures.



To make a visualization with fields from multiple tables, Tableau has to perform joins behind the scenes to compute the values. The type of join used depends on the relatedness of the fields. As a brief overview:

- When *related fields* are used in a viz, dimensions are inner joined and measure values are broken down by the dimensions.
    - It's a little more complicated than that—additional joins might be needed behind the scenes to ensure that no measure values are dropped. But in a dimension-only viz, related dimensions are inner joined and that's the main concept here.
    - This is the same behavior as single-base table models.
- When *unrelated fields* are used in a viz, dimensions are cross joined. Measure values are table scoped (that is, aggregated locally to a single value for their entire table) and repeated.
    - It's also possible for fields to be *not yet related* or *ambiguously* related, which means that for the combination of active fields, there is more than one way for the relationships between their tables to be resolved. If Tableau encounters uncertainty, it treats the fields as unrelated.
- When fields are *stitched* based on a shared field, dimensions are outer joined. Measure values are aggregated at the level of whatever dimension they can be broken down by and might be repeated.
    - *Stitching dimensions* are similar to linking fields in data blending. Results are calculated for each pair of related fields, then the unrelated values are stitched together across the shared values of the dimension shared between them.

All of these concepts and definitions are discussed in more detail later in this topic.

An aside on dimensions and measures

In Tableau, *measures* are *aggregations*—they're aggregated up to the *granularity* set by the *dimensions* in the view. The value of a measure therefore depends on the context of the

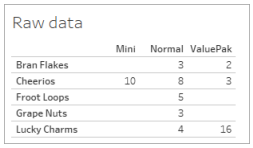
dimensions. For example, "number of cereal boxes" depends on if we mean the total inventory or the number of boxes per brand.

Dimensions are usually categorical fields, such as country or brand. In Tableau, dimensions set the granularity, or the *level of detail*, of the view. We typically want to group our data into marks by some combination of categories. What dimensions we use to build the view determines how many marks we have.

When a measure is used without dimensions, it's said to be *table scoped*. This means its value is the fully aggregated value for the entire table. As soon as we use a dimension such as brand in the viz, the measure is broken down more granularly. The total number of cereal boxes is now per brand.

Aggregation refers to how the data is combined. Tableau's default aggregation is SUM. You can change the aggregation to other options, including: average, median, count distinct, minimum, and so on. *Granularity* refers to how detailed or broken down the measure is—which is controlled by the dimensions. Unless the granularity of the measure is row level (aka disaggregated), its value must be aggregated.

Example



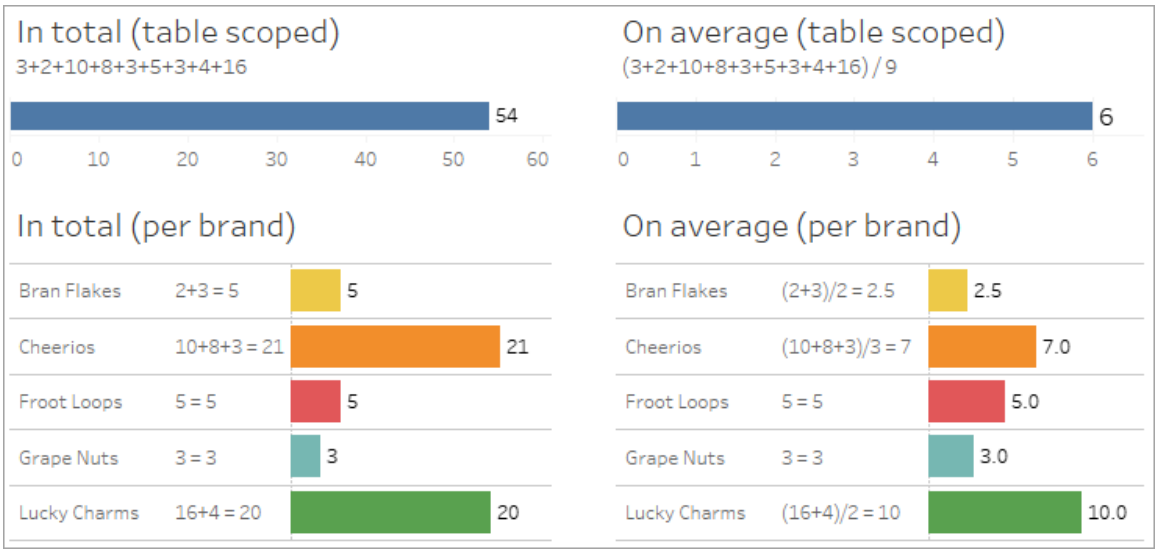
Raw data

	Mini	Normal	ValuePak
Bran Flakes		3	2
Cheerios	10	8	3
Froot Loops		5	
Grape Nuts		3	
Lucky Charms		4	16

What's the value of "number of boxes of cereal?"

Well, it depends on the aggregation type and the granularity as set by the dimensions.

- Aggregations:
  - Sum (or total)
  - Average
- Granularity:
  - Table scoped / fully aggregated (the blue bars in the example)
  - Broken down by the **Brand** dimension (the colored bars in the example)

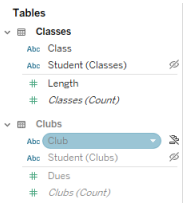


Field-level relatedness indicators

There are several visual clues that can help you understand the degree of relatedness for fields you're using in an analysis.

Relatedness indicators on a worksheet

- **Unrelated icon**: Tableau uses an unrelated icon to indicate not everything in the view is related. If you see an unrelated icon on a pill in the view or in the Data pane, you can hover over the icon to get more information.
  - The related icon indicates that field is stitching together unrelated fields.
- **Light gray field names**: Field names are displayed in light gray text in the Data pane when they're not related to *any* fields in use on shelves. You can still use these fields for analysis in that viz, but unrelated fields are evaluated differently in analysis than fields that are related. On hover, these fields also display an unrelated icon.

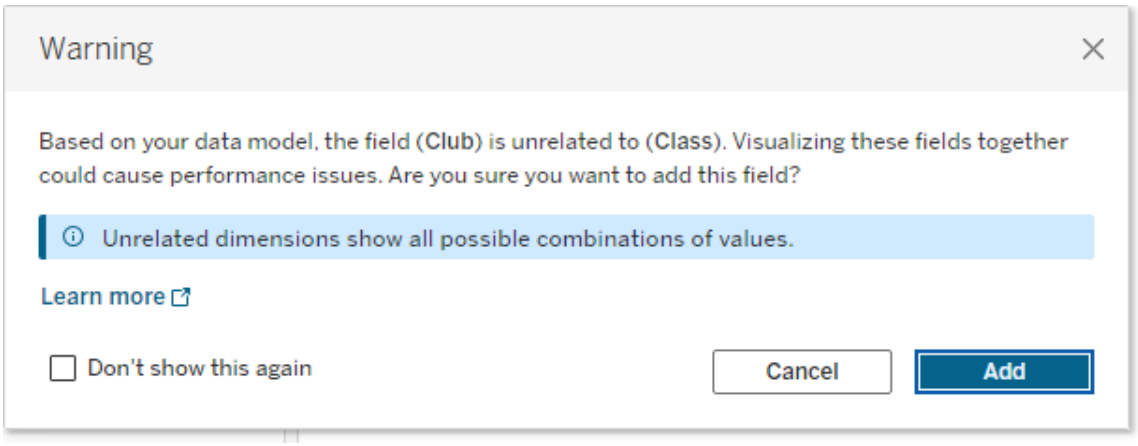


Note: In previous versions of Tableau, light gray field names indicated that the fields were hidden and **Show hidden fields** was selected. Hidden fields, when shown, are now indicated with a clickable eye icon .

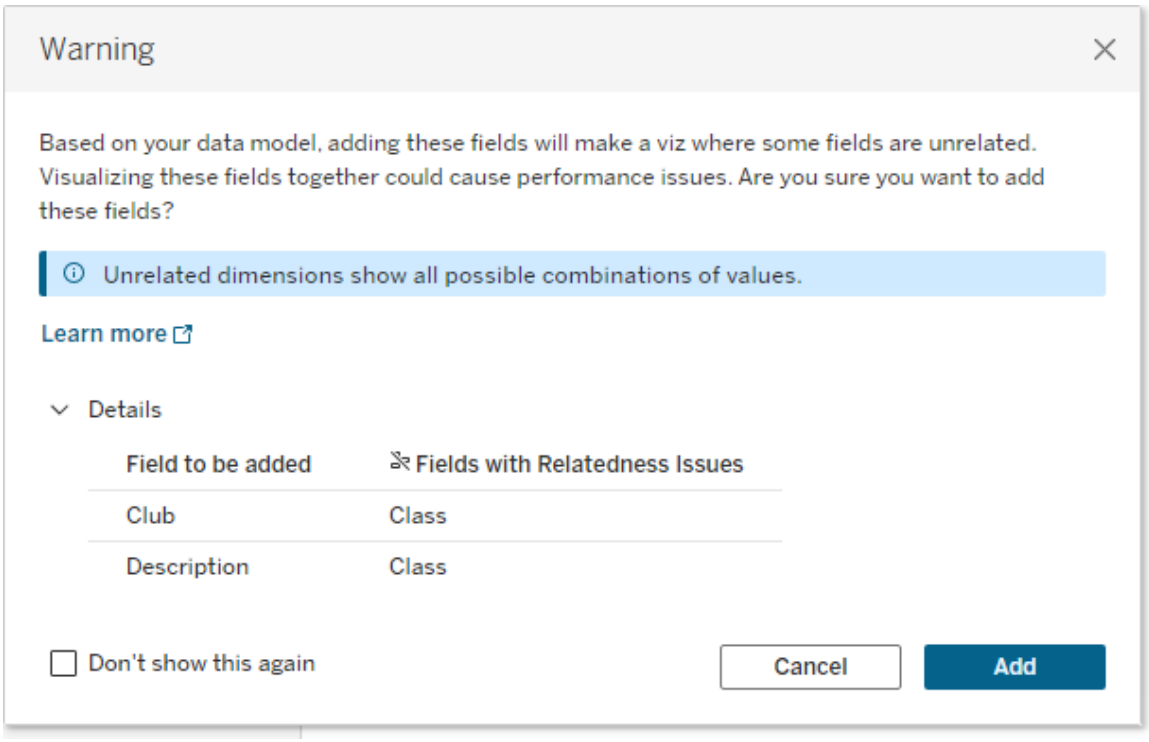
Relatedness warning dialog

When unrelated fields are used together in a viz, Tableau shows a warning dialog to let you know that the fields aren't related. This warning appears each time you add an unrelated field to prevent accidental cross joins that might impact performance.

- If you want to use unrelated fields without stitching, click **Add** to continue adding the field to the viz.
- If you want to stitch unrelated fields, a best practice is to bring out the stitching field before an otherwise unrelated field. The dialog won't show if the stitching field is already in use. See How joins are used for each level of relatedness for more information about how stitching prevents cross joins.



When multiple fields are being added or are already present in the view, the **Details** area appears in the dialog. Expand it to see more information about the relatedness of all the fields in use and identify where the unrelatedness issue is coming from.



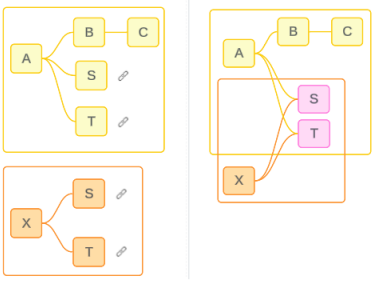
To stop the warning message from appearing at all, select the option **Don't show this again**. You can always re-enable these warning messages again by turning them back on:

- In Tableau Desktop, open the **Help** menu > **Settings and Performance** > **Reset Ignored Messages**.
- In a browser, clear your cached data. For example in Chrome, open the **3 dots menu** > **Delete Browsing Data...** > Choose "**Cached images and files**" > **Delete data**.

Table-level relatedness in the data model

In a data model with multiple base tables, each base table defines a set of tables that are related and form a conceptual *tree*. These trees must be connected by at least one shared table to ensure the overall data source is a single entity.

What might previously have been two data sources that could be blended using linking fields can now be a single data source with two trees, connected by the shared tables that contain those common fields.



Tip:How tables are related in the data model impacts how their fields can be related in the analysis. It can be useful to refer back to the Data Source tab during analysis to see how a table fits into the overall data model.

Let's walk through what tables are related, unrelated, or shared using this example data source. There are two trees, one established by base Table A and one by base Table B.

Unrelated tables

Base tables are fundamentally unrelated. Similarly, any tables that exist solely in a single tree are unrelated to tables in other trees.

Table A and Table X are unrelated

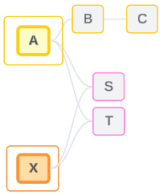
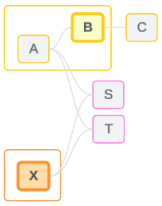


Table B and Table X are unrelated

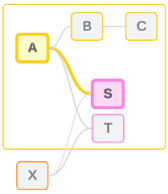


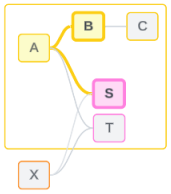
Related tables

Tables in the same tree are considered related.

Table A and Table S are related

Table B and Table S are related (through Table A)

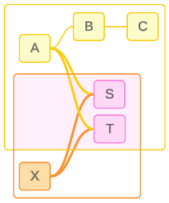




Shared tables

Shared tables have multiple incoming relationships. These tables belong to multiple trees and are shared across them.

Table S and Table T are shared.



Field-level relatedness in the analysis

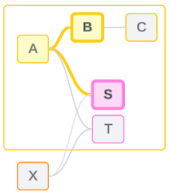
Relatedness between fields is determined on a sheet-by-sheet basis based on the structure of the data model, what fields are actively in use (that is, what fields are in the viz as pills on shelves), and if those fields are dimensions or measures. How field relatedness impact the results of a viz is covered in the next section.

Let's walk through some scenarios using the same example data source. Each field's name indicates which table it is from, such as FieldB from Table B. Fields can be dimensions or measures unless otherwise noted.

Related fields

At a high level, fields are related when Tableau can clearly determine how to evaluate them together based on a relationship path within a single tree.

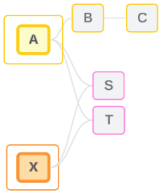
For example, FieldB (from Table B) and FieldS (from Table S) are related.



Unrelated fields

At a high level, fields are unrelated in any case when they're not related. This could be because the fields are from unrelated tables, such as using fields from two base tables. In this case, fields from different base tables are fundamentally unrelated.

For example, FieldA and FieldX are unrelated.



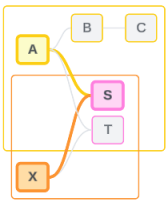
Or fields can be treated as unrelated for a point in time—such as in ambiguous or not yet related cases. For the most part, you can rely on relatedness indicators to alert you when fields are unrelated in the context of a viz.

Stitching dimension

Stitching is how Tableau evaluates fields from unrelated tables in a multi-fact data model during analysis. In a viz, using a dimension from a shared table stitches together otherwise unrelated fields and allows them to be evaluated simultaneously in the same viz. Think of this as juxtaposing results from two trees together based on a dimension they share.

For example, if a viz is built with FieldA and FieldX, these two fields are unrelated. Adding DimensionS introduces a stitching field.

- FieldA and DimensionS are evaluated together.
- FieldX and DimensionS are evaluated together.
- Those intermediate results are brought together based on the values of DimensionS.
- FieldA and FieldX are now stitched.



Tip: A best practice is to use a stitching field in the viz before bringing out an unrelated field. For example, drag out DimensionS first, or FieldA then DimensionS then FieldX, instead of FieldA then FieldX then DimensionS. Adding the stitching field first ensures that Tableau is always aware of how to evaluate the relationships and avoids potential performance issues from evaluating unrelated dimensions together with cross joins.

Stitching requires a dimension from a shared table to be active in the viz. Fields placed on the Filters shelf or on the Tooltip property of the Marks card aren't considered active for the purposes of stitching.

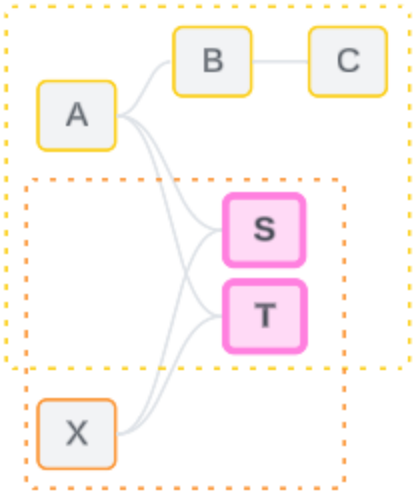
Not yet related fields

Fields can also have multiple ways they could be related but aren't yet. This happens when there's more than one possible relationship between two shared tables (or downstream shared tables).

Consider FieldS and FieldT. Their tables are related to each other both through the tree defined by base Table A and through the tree defined by base Table X.

In a viz with just FieldS and FieldT, there's no information about which tree should be used to relate them. Without additional information Tableau can't evaluate whether to relate these fields through Base Table A's tree or Base Table B's tree.

FieldS and FieldT are treated as unrelated although there are multiple potential relationships.



These could-be-but-aren't-yet-related fields are evaluated as unrelated because Tableau can't clearly determine their relationship path. Unlike truly unrelated fields which can only be stitched, not yet related fields can be resolved and the fields can be directly related.

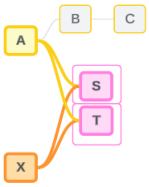
Ambiguously related fields

Fields can also be ambiguously related. This happens when there is more than one active possible relationship between shared tables (or downstream shared tables). Unlike not yet related fields, which can be thought of as hypo-related or under related, ambiguously related fields are hyper-related or over related.

Consider FieldS and FieldT. Their tables are related to each other both through the tree defined by base Table A and through the tree defined by base Table X.

In a viz with FieldA, Field X, FieldS, and FieldT, there's too much information to decide which tree should be used to relate them. Without trimming the information, Tableau can't evaluate whether to relate these fields through Base Table A's tree or Base Table B's tree.

FieldS and FieldT are treated as unrelated although there are multiple active relationships.

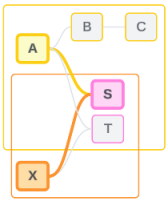


These ambiguously related fields are evaluated as unrelated because Tableau can't clearly determine their relationship path. Unlike truly unrelated fields which can only be stitched, ambiguously related fields can be resolved and the fields can be directly related.

Measure from a shared table

When a dimension is used from a shared table, it stitches together fields from its unrelated upstream tables. Measure can't stitch, however, and the value of a measure depends on its related dimensions.

In a viz with DimensionA and DimensionX, these two dimensions are unrelated. If MeasureS is brought out from Table S, it is unrelated to the combination DimensionA and DimensionX together. Although it could be related to either one independently, it can't be simultaneously related to both of them in the same viz.



A shared measure can be considered a type of ambiguity or over relatedness and is resolved the same way.

Resolve unclear relationships between fields

Whenever there is uncertainty about how to relate fields, Tableau won't make an arbitrary decision and instead treats them as unrelated. It's often better to relate these fields by clarifying the uncertainty around which tree to use.

Resolving not yet related fields is done by adding a field to establish which tree to use. Resolving ambiguously related fields is done by removing fields to establish which tree to use.

Example:

Resolving not yet related: add a field

- In a viz of FieldS and FieldT, adding a field from Table A, B, or C to the viz makes Base Table A's tree active and resolves the desired path between FieldS and FieldT.
- Alternatively, using a field from Table X resolves the desired path between FieldS and FieldT to Base Table X's tree.

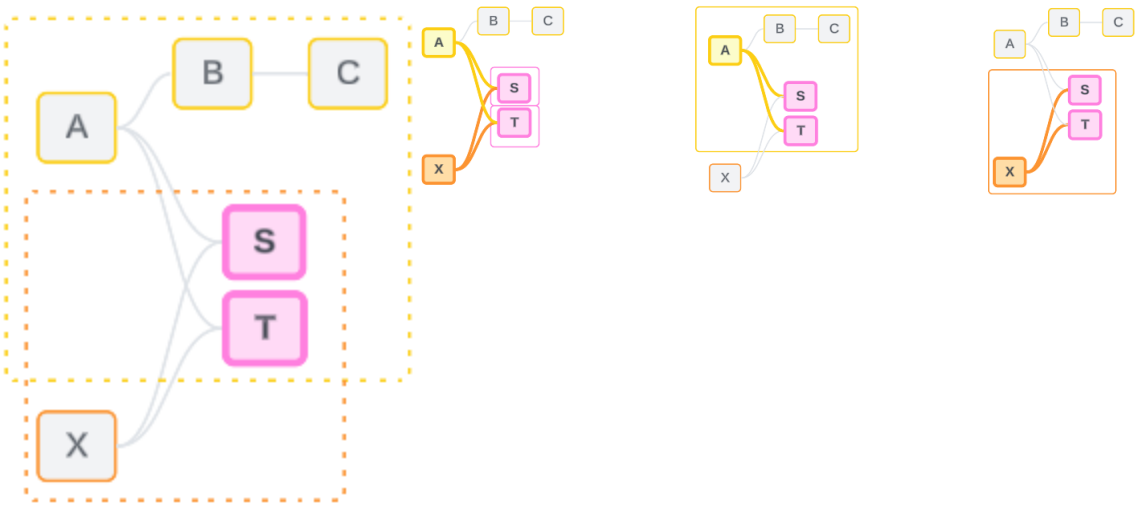
Resolving ambiguously related: remove a field or fields

- In a viz of FieldA, FieldX, FieldS, and FieldT, removing FieldX makes only Base Table A's tree active and resolves the desired path between FieldS and FieldT.
- Alternatively, removing FieldA resolves the desired path between FieldS and FieldT through Base Table X's tree.

Resolving a shared measure: remove a field or fields

- In a viz of DimensionA, DimensionX, and MeasureS, removing DimensionX makes only Base Table A's tree active and resolves the desired path between DimensionA and MeasureS.
- Alternatively, removing DimensionA resolves the desired path between DimensionX and MeasureS through Base Table X's tree.

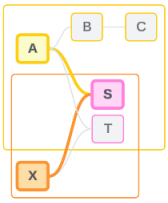
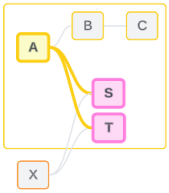
Not yet related	**Ambiguously related**	**Relatedness resolved to a single tree**	
		Related through base Table A	Related through base Table X



Resolving uncertainty is similar to using a FIXED Level of Detail (LOD) expression. In a FIXED LOD expression, you tell Tableau what level of detail to aggregate to by defining the dimension declaration. Uncertainty is resolved by changing the structure of the viz to make only one tree active, thus telling Tableau what relationship paths it can consider to perform the analysis.

Stitching vs resolving uncertainty

Both stitching and resolving uncertainty are ways of dealing with unrelatedness, but they have different outcomes:

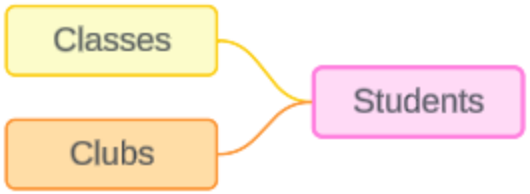
Stitching	Resolving uncertainty
	
Unrelated FieldA and FieldX stitched by DimensionS	FieldS and FieldT evaluated through the tree defined by base Table A
Juxtaposes unrelated fields based on shared attributes	Narrows down which relationship path to use when there are multiple options (ambiguity or a

	shared measure), or establishes a relationship path when there wasn't one (not yet related).
Uses multiple base table logic to calculate results	Uses single base table logic to calculate results
Analysis involves unrelated tables	Analysis involves shared tables

How joins are used for each level of relatedness

After the field-level relatedness has been determined, Tableau has to evaluate the results to create the actual visualization. The queries used to calculate the values shown in a viz rely on joins. Whether fields are related, unrelated, or stitched has a different impact on what joins are performed. Remember, ambiguously related and not yet related fields are treated as unrelated in this context.

To explain relatedness and joins, this section covers tables and their fields, plus the values in those fields. Consider the following data model with two base tables, Classes and Clubs, and a shared table, Students.



Classes	Clubs	Students

Classes
7 rows 3 fields

Abc Classes Class	Abc Class Stuc
Nesting Basics	Robi
Advanced Songs	Spar
Flying For Fledgelings	Robi
Nesting Basics	Spar
Advanced Songs	Finc
Nesting Basics	Finc
Alarm Calls 101	

Clubs
7 rows 3 fields

Abc Clubs Club	Abc Clubs Student ...
Photography	Finch
Travel	Cardinal
Juggling	Sparrow
Art	Finch
Art	Cardinal
Art	Sparrow
First Aid	Robin

Students
5 rows 3 fields

Abc Students Bus Rider	Abc Students Student	# Students Age
yes	Finch	3
yes	Cardinal	4
no	Sparrow	6
yes	Robin	3
no	Jay	8
		10
		0

Fields:

- **Class**, a dimension with values of Nesting Basics, Advanced Songs, Flying for Fledglings, and Alarm Calls 101
- **Length**, a measure
- **Student**, a dimension used to relate to the Student table

Fields:

- **Club**, a dimension with values of Photography, Travel, Juggling, Art, and First Aid
- **Dues**, a measure
- **Student**, a dimension used to relate to the Student table

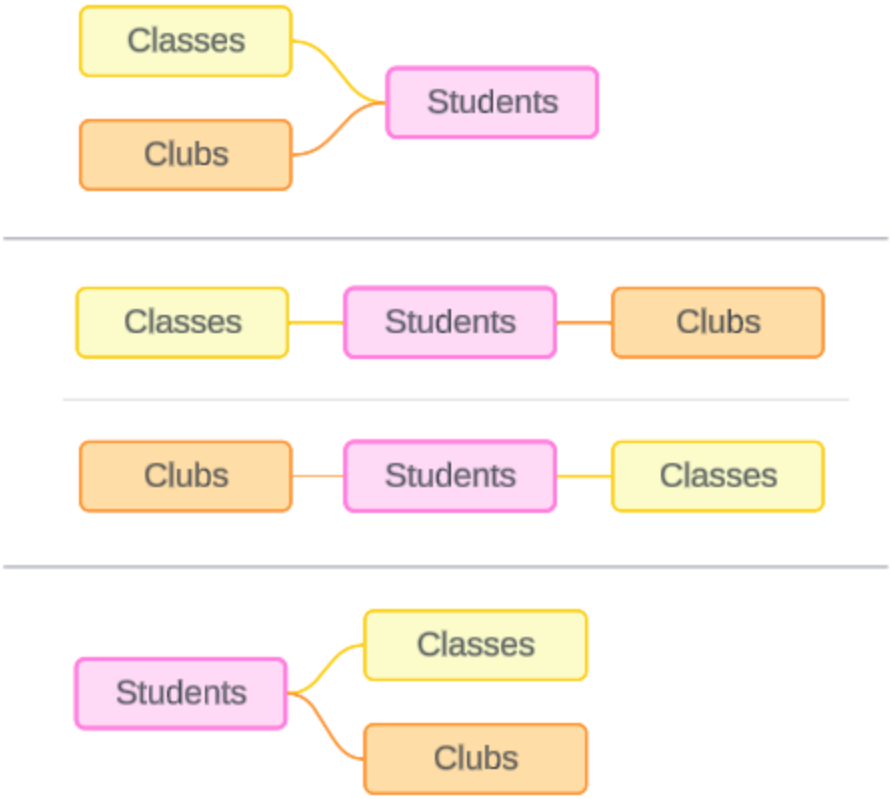
Fields:

- **Bus Rider**, a dimension with values of yes or no
- **Student**, a dimension with values of Finch, Cardinal, Sparrow, Robin, and Jay. Used to relate to the other two tables
- **Age**, a measure

This very simple model illustrates how the high-level join logic is calculated for multi-fact relationship data models. For more information about the basics of joins used in single base table data models built on relationships, see How Analysis Works for Multi-table Data Sources that Use Relationships.

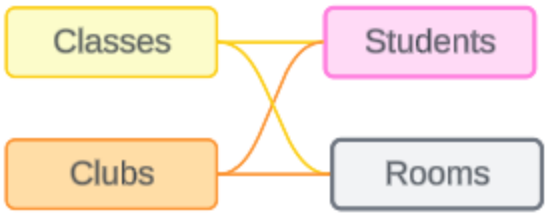
Should this example be a data model with multiple base tables?

For this three table data model, it might be tempting to set it up as a single base table model, as Classes-Students-Clubs or Clubs-Students-Classes, or with Students as a base table. As a rule, multi-fact relationship data models are intended for specific kinds of data schemas or analysis scenarios. If your data model has characteristics that are best suited to a multi-fact relationship data model, set it up that way to keep your base tables conceptually unrelated. However, if your data doesn't require this type of structure, a single base table model can be simpler to use.



Models that could be built for these three tables: (1) Classes and Clubs as base tables with Students as a shared table, (2) linearly, starting with either Classes or Clubs, and (3) Students as a single base table with Classes and Clubs as downstream tables.

In this particular instance, there's nothing about these tables, the data, or the model that truly requires multiple base tables. We're using this model as an example to keep it simple so the focus can be on the join logic. Or you could imagine that there's another related table, Rooms, that we're simply ignoring to avoid over complicating the discussion.



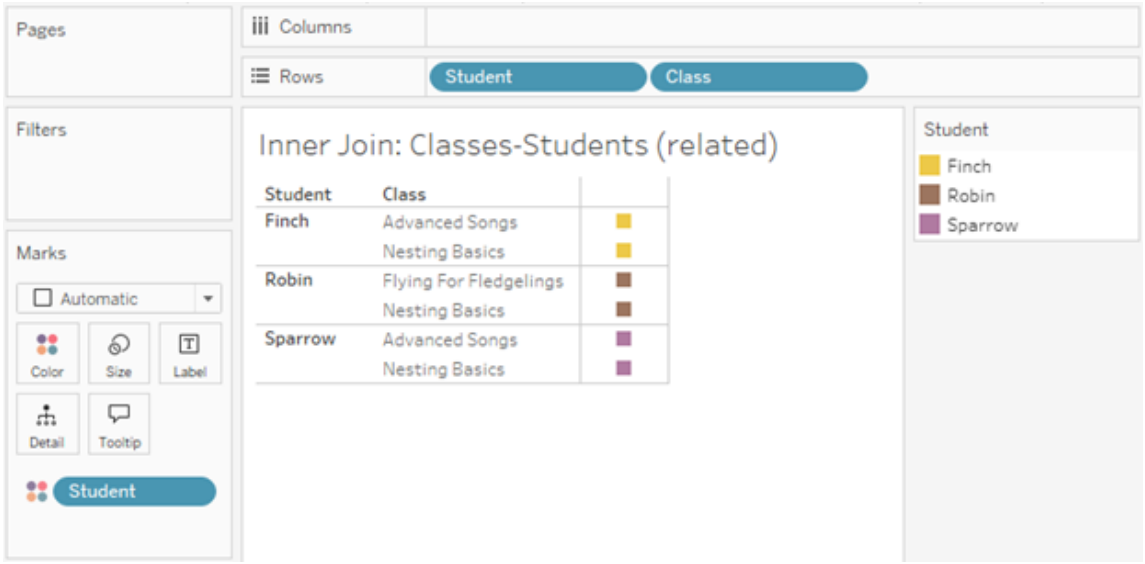
As a best practice, however, only use a multi-fact relationship model when your data requires it.

Related dimensions use inner joins

Related dimensions are inner joined. Inner joins drop any dimension values that aren't shared across both tables.

- Tableau uses additional logic to ensure measure values aren't lost. This section uses only dimensions to demonstrate the basics of how Tableau applies inner joins to related dimensions.

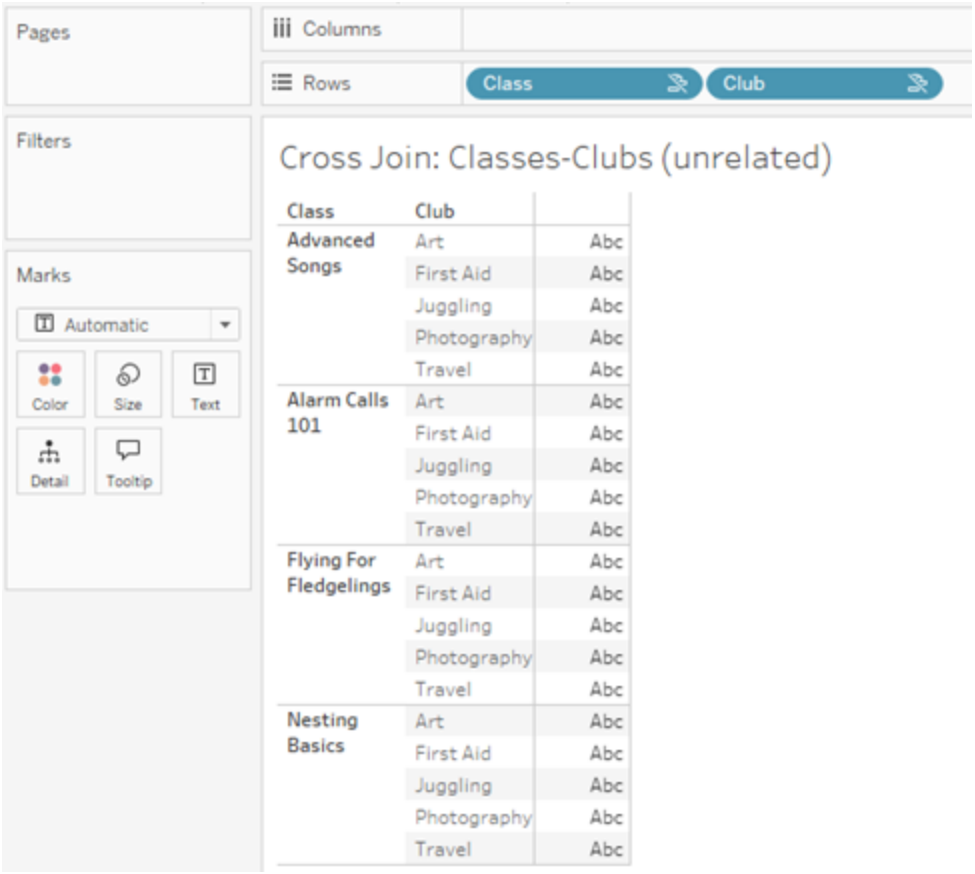
The following example shows how related dimensions only return rows that are present in the data. No students are in the Alarm Calls 101 class, so it's not present in the results. Cardinal and Jay aren't in any classes, so they're not present in the results.



Unrelated dimensions use cross joins

Unrelated dimensions—on their own, without a stitching dimension—are cross joined.

In a cross join, every value from one dimension is combined with every value from the other dimension, even if a resulting combination doesn't actually exist in the data. In this example, the cross join adds a row for each possible combination of Class and Club.



It's important to recognize when a cross join is occurring in your analysis. Although there's a row for Advanced Songs + First Aid in the results table for the cross join, no students are actually in this combination of activities (we'll see proof of this in the stitching example in the next section).

Why is it important to recognize that not all cross join results are based in the data? Imagine you were trying to build a schedule for classes and clubs so there were no conflicts for any students. There aren't any students in Advanced Songs and First Aid, so you could ignore this result and schedule that class and club simultaneously. The cross join doesn't represent combinations of values that actually exist in the data.

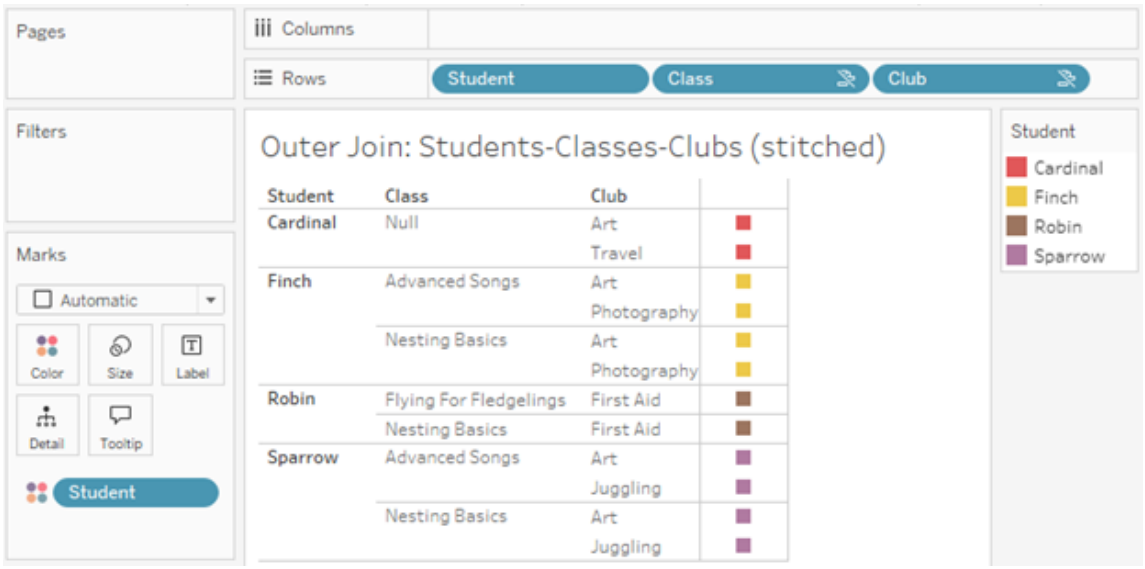
Additionally, cross joins when there is high cardinality (a large number of unique values) can impact performance. Imagine cross joining every phone number with every email address in

your contacts. That would be a huge explosion of combinations and a potentially costly operation.

Stitched dimensions use outer joins

Unrelated dimensions—in the presence of a stitching dimension—are outer joined.

In this example, both the Classes table and Clubs table are related to the shared Students table but not to each other, so the fields Class and Club are unrelated. Adding the Student dimension lets Tableau know which values from Class and which values from Club should be juxtaposed in the analysis. We call this outer join behavior *stitching*.



Stitching is similar to data blending in that there are intermediate results that are brought back together for the overall results. Unlike blending, however, stitching is an outer join, not a left join, and doesn't drop values from either side. There's no concept of primary or secondary data sources when it's all one data source, so both of the unrelated fields are given equal precedence.

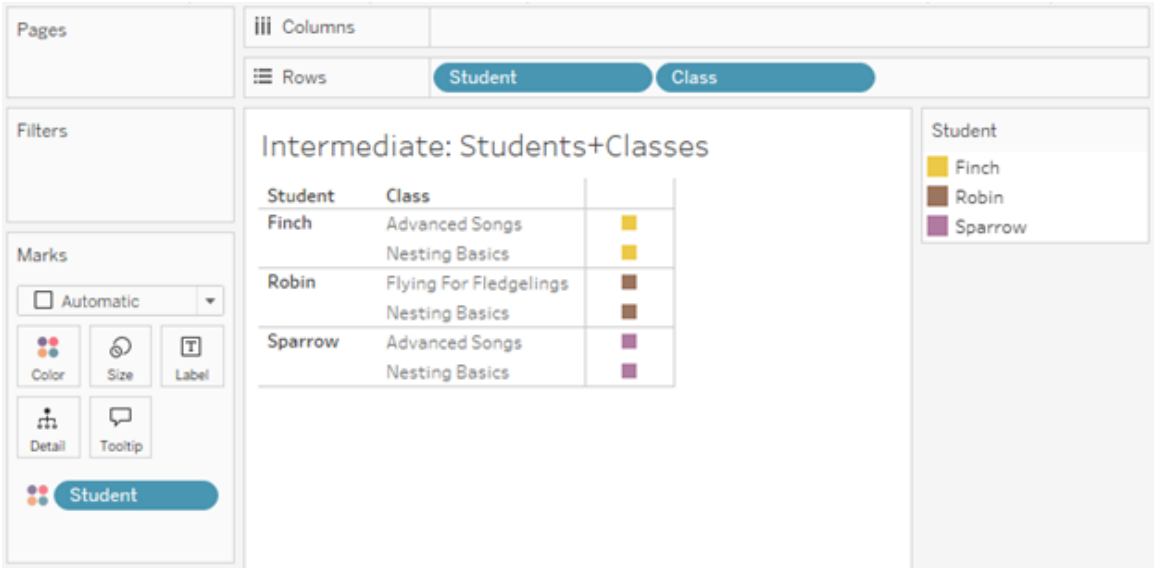
Intermediate results are outer joined

What goes into the outer join for stitched fields? An immediate inner join is computed for each of the unrelated fields and the stitching field in turn, then those intermediate results are outer

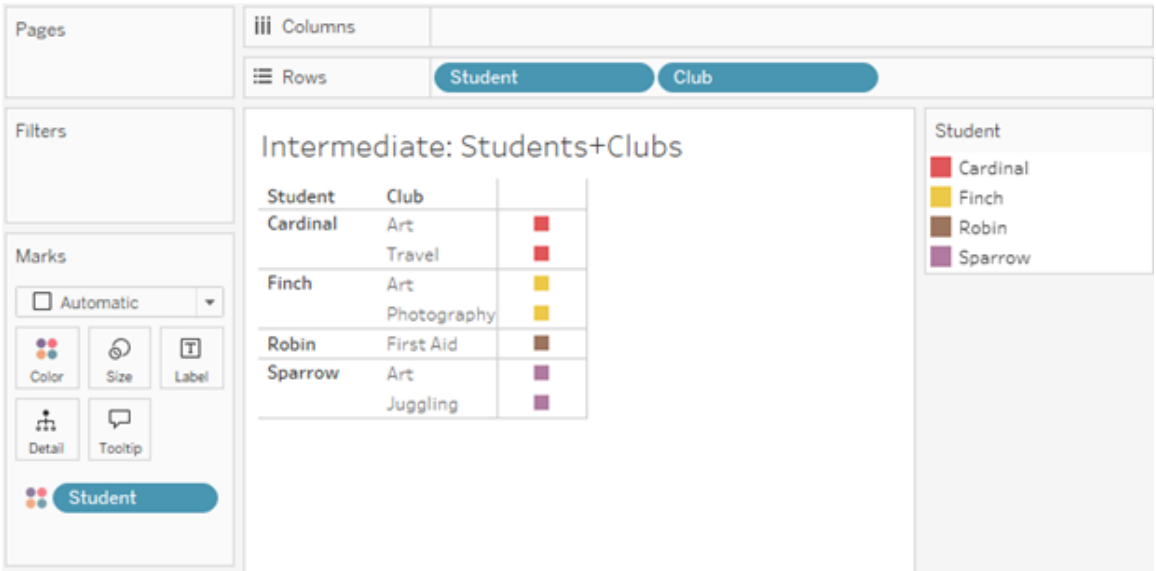
joined based on the values of the stitching dimension.

Example

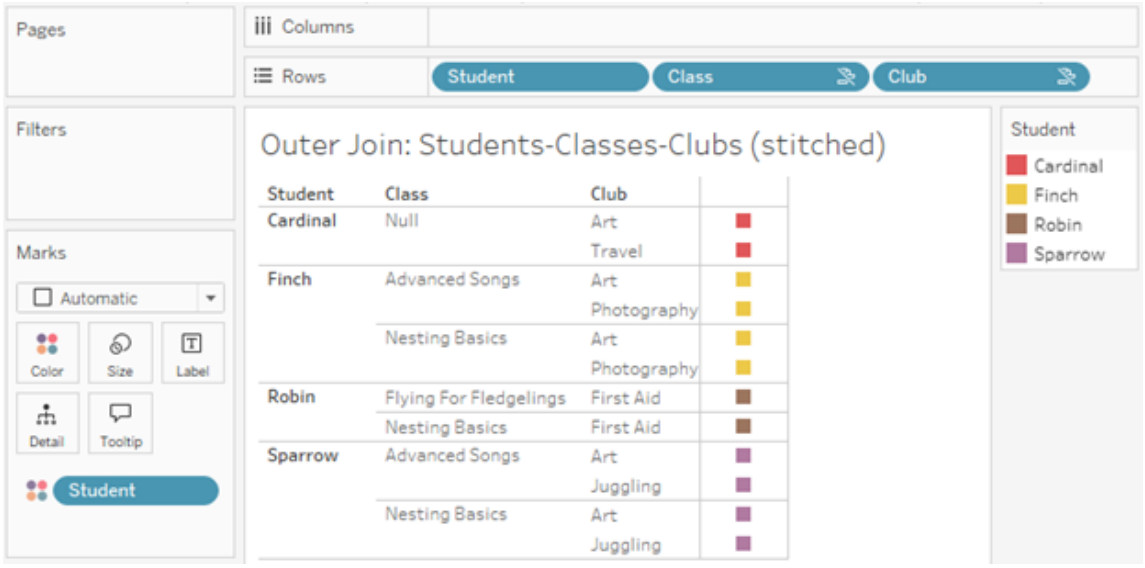
An inner join for Student and Class...



...and an inner join for Student and Club...



...are then outer joined on Student.



Additional joins to retain measures

In addition to the join logic for dimensions, measures can introduce additional joins. When relationships were first introduced in Tableau, one of the core principles was that measure values aren't lost. This is also maintained in multi-fact relationship data models.

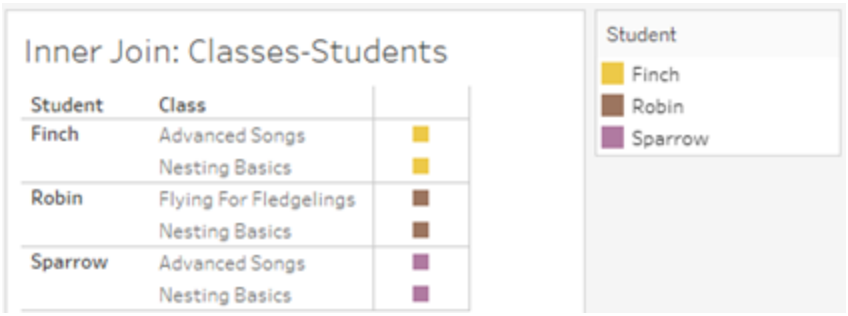
The essential details are:

- Measure values are broken down only by related dimensions.
- Measure values repeat for unrelated dimensions.
- Dimension values that would be dropped in dimension-only vizzes may be returned if there are relevant measure values associated with them.

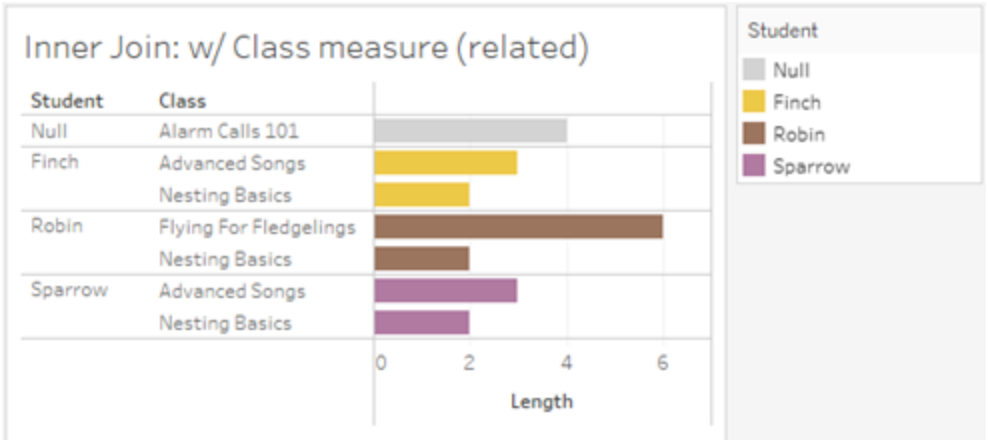
Note: Remember that measures are aggregations—they're computed at the level of detail (the granularity) set by the combination of dimensions in the viz. This is referred to as a measure being *broken down* by a dimension. When a measure is used without any dimensions, it is said to be *table scoped*. This means the measure's value is the fully aggregated value. As soon as we use a dimension in the viz, the measure is broken down more granularly based on the dimension values. The value of a measure in an analysis therefore depends on the context of the dimensions.

Related measures

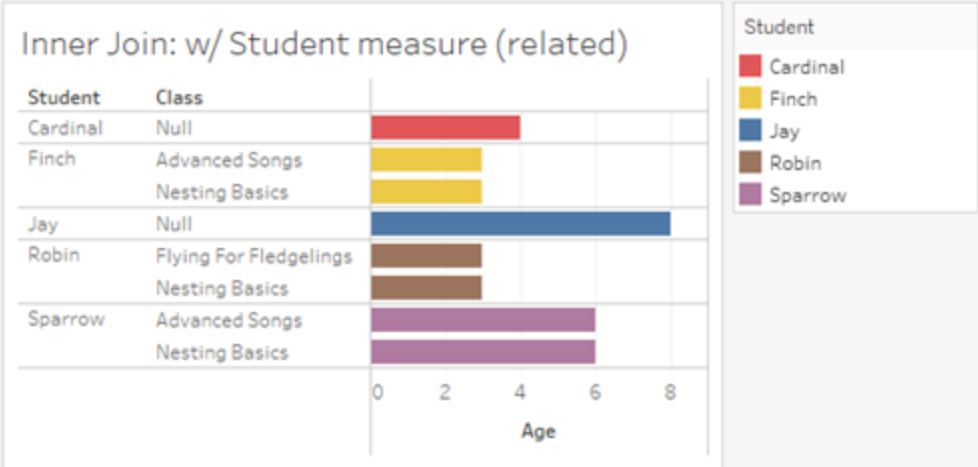
Consider the subset of dimension values that are returned for an inner join on the related dimensions **Student** and **Class.** There are three student values, Finch, Robin, and Sparrow; and three class values, Advanced Songs, Nesting Basics, and Flying for Fledgelings.



If we add the **Length** measure from the Class table, we see that all four classes are shown and there's a null for Student. Every class **Length** is displayed, at the level of **Class**.



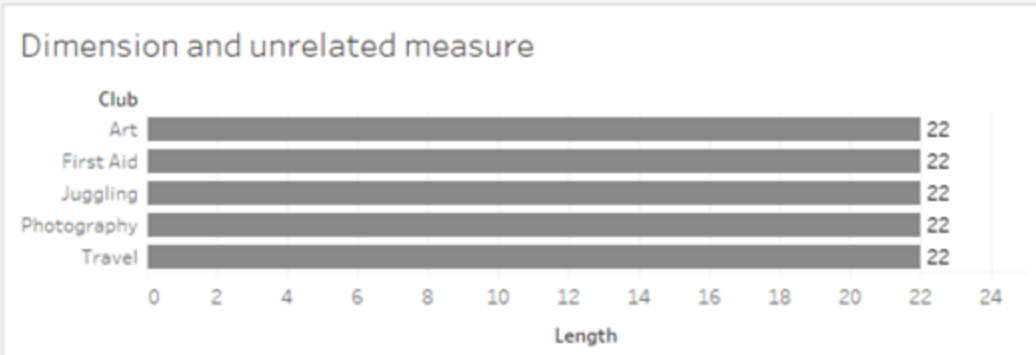
If we instead add the **Age** measure from the Student table, we see that all five students are shown and there are two nulls for Class. The results preserve every student, even if they're not in a class. Every student **Age** is displayed, at the level of **Student**.



Unrelated measures

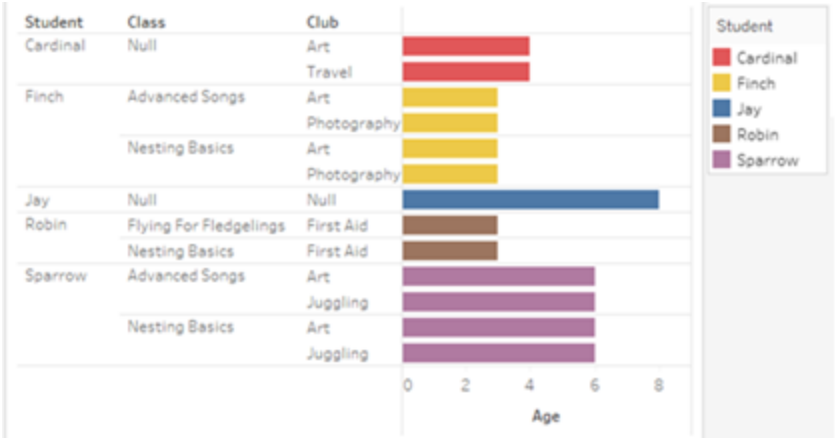
Measure values are repeated for unrelated dimension values.

If we look at the **Length** measure from the Classes table and the unrelated **Club** dimension, the measure is table scoped and repeated across all the dimension values of **Club**.



In the presence of a stitching dimension, measures can be both broken down and repeated.

Here, the measure **Age** is from the Students table and is broken down to the level of student. Each time a student is repeated based on the dimensions for **Class** and **Club**, the **Age** value is repeated.



Troubleshooting

Considerations when working with multi-fact relationship data models

Per table extract filters

All extract filters for a multi-fact relationship data model extract are per-table (not pervasive). Because of this, filtering results may be different between live and extract connection.

Row level calculations

Row level calculations can only refer to fields which share the same upstream base table. That is, row level calculations can't be performed across trees.

Combined Fields

All fields in a combined field must share an upstream table. That is, you can't create a combined field using fields that are in different trees.

Sets

Sets can only be created with a definition that involves fields that share the same upstream base table. However, in a viz, the option to Add to Set may be available from a mark when that mark is defined by fields unrelated to the fields used to define the set. If you choose Add to Set, Tableau will add only the related fields to the set definition. This is different from the behavior

for Add to Set in single base table data sources, when Add to Set adds everything that defines the mark.

Validate INCLUDE level of detail expressions

INCLUDE LOD expressions can't be evaluated across unrelated fields. Because relatedness between fields is evaluated on a sheet-by-sheet basis, it's possible to have a valid LOD expression in the Data pane or calculation editor that becomes invalid in the context of a specific viz (in the presence of an unrelated dimension). When this happens, the LOD pill will turn red. You can update the LOD expression to remove unrelated field conflicts, change the structure of the viz, or remove the LOD expression from the viz.

Updating Published Data Sources

As a best practice, create a copy of an existing published data source if you plan to modify it to become a multi-fact relationship data model when not all of its connected workbooks need the new data model. Don't update the existing version of the data source unless all its workbooks need the new tables. Publish the modified data source as a new data source and create new workbooks from it. This will prevent the existing workbooks from being converted to use VDS instead of data server when they don't need the functionality, preventing the potential for a performance hit.

Resolved issues

Resolved Issue	Fixed as of
Extracts *Local data source (in a workbook):* Attempting to extract a multi-fact relationship data source will give a "No such table" error. *Published data source:* Extracting a published multi-fact relationship data source appears to succeed, but field values can be swapped.	• Tableau Cloud: Resolved as of mid July updates. This also applies to public.tableau.com. • Tableau Desktop: Resolved as of maintenance release 2024.2.1 released July 24th, 2024 • Tableau Server: Resolved as of maintenance release

EXCLUDE Level of Detail expressions Only INCLUDE LODs should be validated in the presence of unrelated fields. However, EXCLUDE LODs may also be incorrectly marked as not valid in the same conditions.	2024.2.1 released July 24th, 2024 If you still see these issues in Tableau Desktop or Tableau Server, upgrade to a version from July 24th, 2024 or later.
Nested user calculations Nested user calculations are not available in published data sources with a multi-fact relationship data model.	

Known issues in 2024.2

Relatedness indicators with multiple Marks cards

When a viz is built with multiple measures on the Rows shelf or on the Columns shelf, each measure gets its own Marks card. The logic used to determine relatedness indicators (the unrelated icon, the text in tooltips, and the relatedness warning dialog) may not give expected results depending on which Marks card is open. The viz itself, however, is correctly computed based on the relatedness of each pair of fields. There is a planned fix for this behavior.

BatchQueryProcessor

BatchQueryProcessor must be enabled to support multi-fact relationship data models. This is expected behavior with no currently planned fix.

Tableau Pulse

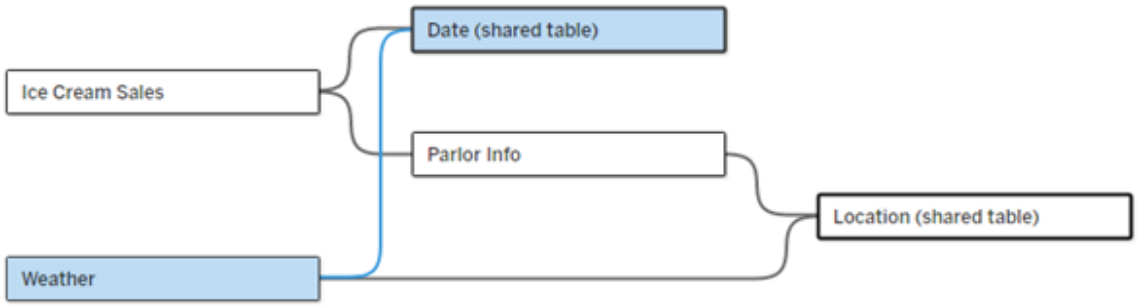
Pulse may not work with multi-fact relationship data models. You may be prevented from creating a metric definition, or any metrics that are created may be blank. This is not expected behavior but there is no currently planned fix yet.

When to Use a Multi-fact Relationship Model

A multi-fact relationship model is a data model that lets you add unrelated tables in a single data source and then uses related fields during visual analysis to essentially stitch the tables together based on the context. Unlike blending, the data exists within a single data source—the concepts of primary and secondary data sources don't apply and no data is dropped from left joins. Unlike a single table data model, multiple base tables maintain their own context regarding tables shared between them. A multi-fact relationship data model gives you more options for performing multi-fact analysis in Tableau.

Imagine you want to analyze how weather and ice cream sales trend together. Weather and ice cream sales both happen at specific times and specific places, but there's no direct connection between ice cream sales and weather. These are unrelated pieces of data that both relate to the shared concepts of date and location.

This question lends itself to creating a multi-fact relationship model. Ice Cream Sales and Weather each can be added as a base table and related on Date and Location, which are shared tables.



A multiple base table data model, with two unrelated tables (Ice Cream Sales and Weather) and two shared tables (Date and Location). There is an intermediate table, Parlor Info, between Ice Cream Sales and Location.

Why did we build the capability to model unrelated tables?

Analysis often involves bringing together tables of data that don't even have a direct relationship to each other but that both relate to the same, common information (such as date or

location). A multi-fact relationship model supports loose semantic coupling by introducing the concept of degrees of relatedness and the ability to build a data model with multiple, unrelated base tables.

- Semantic coupling is a term used to describe how tightly combined data is. A *join* or a *union* is a tight semantic coupling; they bring multiple tables together into a new physical table that then acts as a single table. A *relationship* is a looser coupling between tables that ties tables together logically, maintaining their distinct status as separate tables. Even further along the semantic coupling spectrum is *data blending*, where results from separate data sources are visually combined based on elements shared between both them. A *multi-fact relationship model* is closer to the blending end of the spectrum, but within a single data source instead of across data sources.

A multi-fact relationship model—a data model with multiple base tables—permits unrelated tables in the model as long as shared tables exist in the model, too. During analysis, fields from a shared table "stitch" together otherwise unrelated tables of data based on the shared dimensions they have in common (such as happening in the same place or at the same time). All the benefits of relationships are maintained, including the retention of each table's grain, or native level of detail.

Similar to a single base table data model, Tableau determines the best join type to use behind the scenes based on the structure of the viz. But in a multi-fact relationship model, the join options are expanded to include outer and cross joins to handle different levels of relatedness. For more information, see About Multi-fact Relationship Data Models.

Where did the name come from?

Multi-fact relationships get their name from multi-fact analysis. In a data warehouse model, data is stored in a central fact table surrounded by dimension tables. In this context, *fact* refers to measurements or metrics, which are numeric fields of data that capture facts about the data—Tableau's measures. Dimension tables contain attributes about these facts.

Schemas based on fact tables are often structured as a star or snowflake, depending on how the dimension tables are organized. When analysis needs to be performed across fact tables,

this is called multi-fact analysis. Analysis is done in the context of the common dimension tables, known as shared dimensions or conformed dimensions. In Tableau you build these data models using relationships, so we've named this suite of capabilities multi-fact relationships.

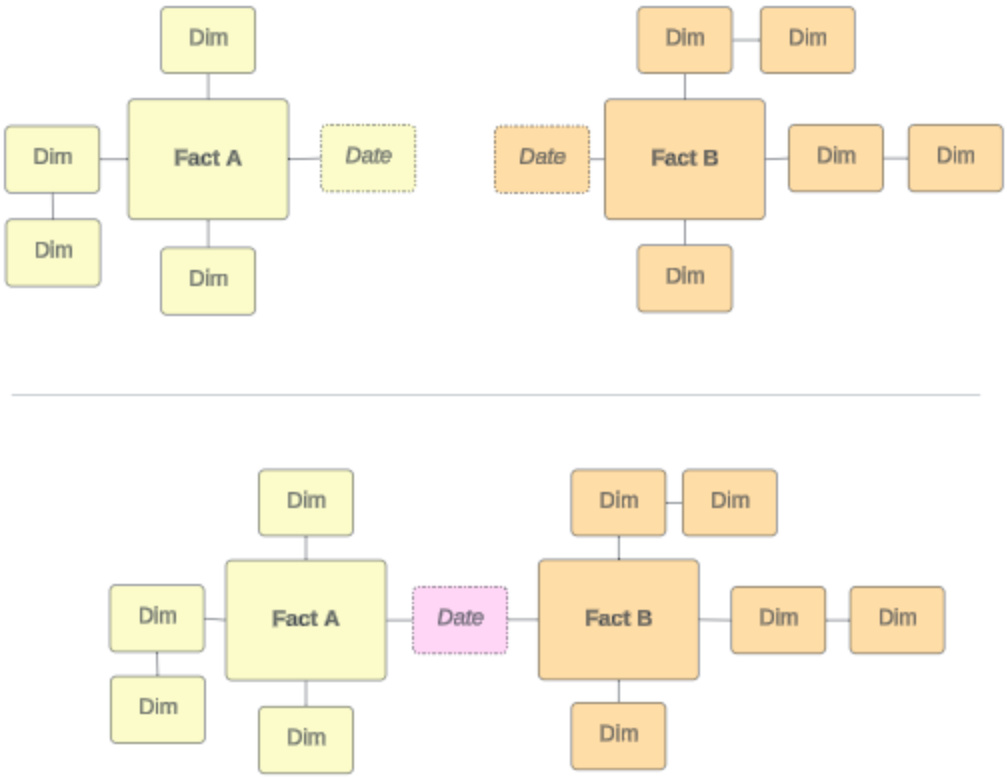
When to use multi-fact relationship data models

If your data consists of tables that are all related to each other, you can stick with single base table data sources built with relationships. A multi-fact relationship model is called for when your data spans different concepts, either in the form of multiple fact tables, or different unrelated contexts.

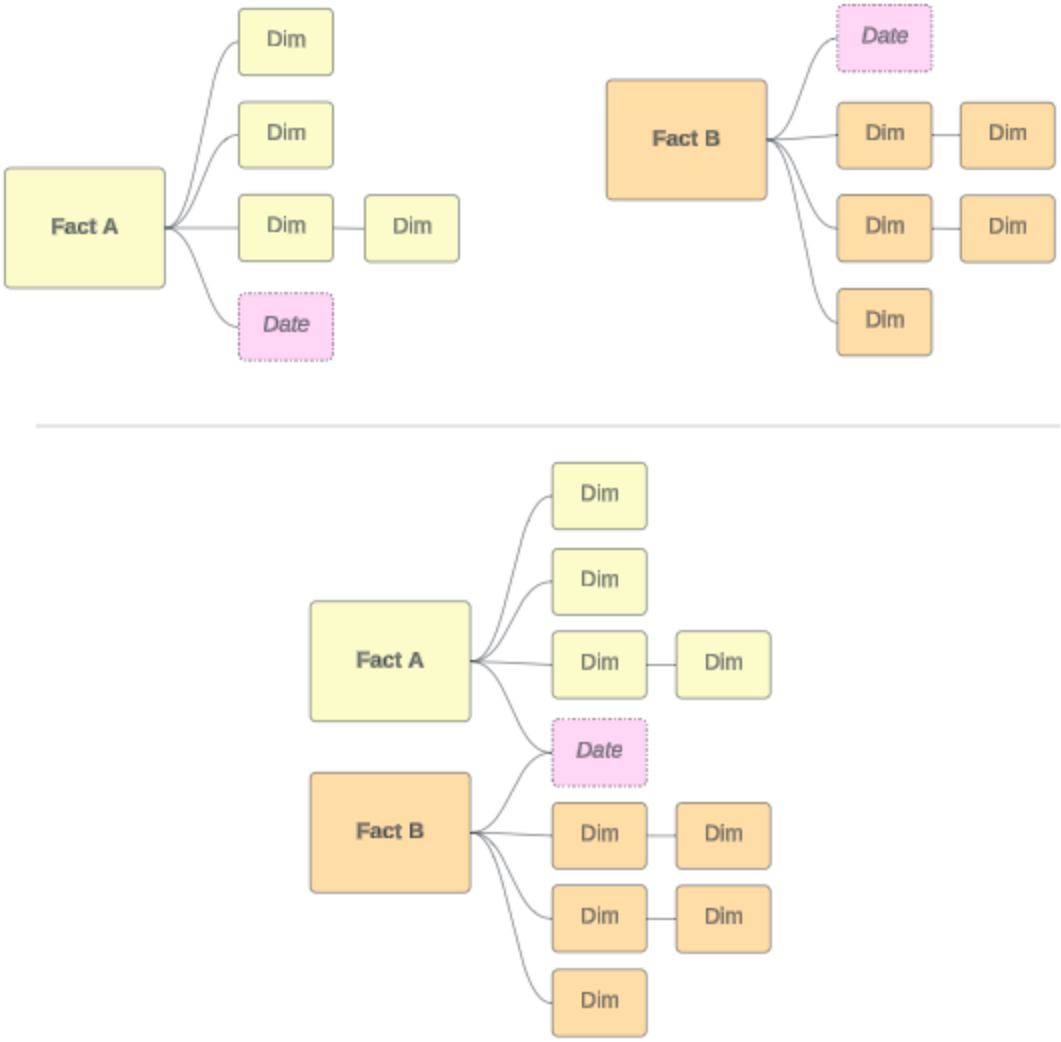
Whenever possible, build your data sources with a single base table. In a single base table data model every table is related and there is no need to consider degrees of relatedness. Only use multi-fact relationships when that data model structure is called for.

Multi-fact analysis

Multi-fact analysis is a core use case for multi-fact relationships in Tableau. In this example, Fact A and Fact B share a table Date.



To model this in Tableau, the fact tables become base tables and multiple incoming relationships are established for their shared dimension table.



Other scenarios

Multi-fact relationship data models aren't just for multi-fact analysis, however. Tableau doesn't require a strict definition of fact or dimension tables. Any table can be a base table (although it should suit the characteristics of base tables). Some scenarios that indicate a multiple base table data source might be helpful include:

- **Moving through stages**, such as base tables for applications, transcripts, and alumni events for a shared student table.

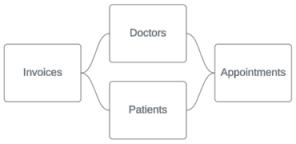
- **Different contexts for the same events**, such as base tables for the events of medical appointments and billing invoices, with shared tables to set the context to doctors or patients.
- **Different domains that may correlate,** such as scenarios that would previously be best handled with data blending, like ice cream sales and weather correlated through the shared tables of date and location.

Learn more about when multi-fact relationships are useful in this Tableau blog post: When and How to Use Multi-fact Relationships in Tableau.

Identify the base tables

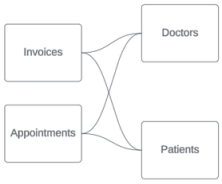
In a multi-fact relationship model, directionality matters. That is to say, which tables are the base tables along the left side of the model and which tables are shared downstream impacts how the relationships are evaluated to return the analytical results.

Consider a conceptual bow tie of invoices, appointments, doctors, and patients:

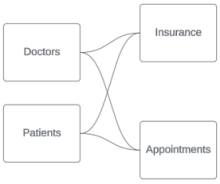


The correct way to build the data model in Tableau is with Invoices and Appointments as the base tables, and with Doctors and Patients as the shared tables (not with Doctors and Patients as the base tables).

Correct: Invoices and Appointments as the base tables



Incorrect: Doctors and Patients as the base tables



Conceptually, a patient (or doctor) is the entity that stitches together the event of an appointment and the event of an invoice.

If your data model is backwards (such as with Doctors and Patients as the base tables instead of Appointments and Invoices), the outer join stitching behavior won't be as useful. Your analysis might show a lot of table scoped measures and ambiguity. If you find yourself with ambiguously related fields that you weren't expecting , reevaluate the tables you are using as base tables and see if your data model needs to be reversed.

Characteristics of base tables and shared tables

If you're performing multi-fact analysis, the fact tables become the base tables and any shared dimension tables are shared tables. Tableau doesn't require a strict adherence to fact and dimension table characteristics. However, there are certain attributes that can help you identify which tables should be base tables and which should be shared tables.

Base table	Shared table
Fact tables in a data warehouse schema	Shared or conformed dimension tables in a data warehouse schema
Specific to the context or analysis	Consistent concept across various contexts
(flight information, energy usage)	(date, location)
Measure heavy	Primarily dimensions
More frequently updated/transactional	More stable/durable
(medical appointments, prescriptions, vitals)	(doctor, patient)
Has foreign key fields	Has primary key fields
Event based	Entity based

(class schedule, grade on an assignment)	(student, classroom)

Note that if there are intermediate tables between a base table and a shared table, you can swap which one is the base table without fundamentally altering the data model. (Such as Parlor Info and Ice Cream Sales in the first example.) What matters is which tables are upstream of the shared tables and which are shared.

Try an additional base table instead

There are various scenarios that may indicate you should build a multi-fact relationship model with multiple base tables rather than a single base table data source:

- If you're trying to build a data source with a cycle, the downstream table should be another base table instead.
- If you have a series of tables that are related on the same sets of relationship clauses (such as date and location), those dimensions should be pulled out and made into shared tables instead.
    - This is especially useful because multiple relationship clauses must all be true (logically, an AND) for the tables to be related for those records.
    - If, instead, you want to analyze records where one may be true at a time (a contextual OR), this flexibility is provided by setting up a data model with shared dimension tables instead.
- If you're using a blend but want to have an equivalent blend without primary and secondary data sources, build a data model which combines the data sources from the blend with their linking fields in a shared table or tables.

Understand Tooltips for Multi-fact Relationship Data Models

Note: For single table data sources or single-base table data sources, all the tables are related. Everything on this page refers to multiple base table data sources.

Field-level relatedness

Data models with multiple base tables have a lot of flexibility in how the tables can relate—or not relate—to each other. The relatedness of the *tables* is a constant based on the data model. However, the relatedness of *fields* in a viz depends on what fields are active (that is, what fields are in use on the worksheet shelves as pills). At the level of a single viz, Tableau evaluates active fields in pairs to determine how they relate to each other.

An unrelated icon on a field means it's unrelated to at least one other field in the viz. This icon can appear in a pill on a shelf or in the Data pane. (In some instances, there may be a related icon .) Hovering over the icon opens a tooltip with more information. There are different messages for different types of field relatedness:

- Unrelated dimension-dimension pair
- Unrelated dimension-measure pair
- Stitching dimension
- Measure from a shared table
- Unrelated filter pair

The messages also varies slightly depending on whether the field is in use on a shelf or in the Data pane.

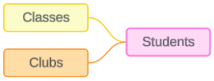
- **On a shelf**: The tooltip for pills on shelves provides information about how the fields in the viz are related to each other and what impact that has on how Tableau computed the results.
- **In the Data pane**: The tooltip for fields in the data pane provides information about what *would* happen if that field were added to the viz. Fields in the Data pane can also be de-emphasized with light gray text if they don't related to any fields in the viz.

Tip: In Tableau, aggregated dimensions—such as ATTR(dimension) or MIN(dimension)—act like measures.

Know your data model

Whenever you're working with a data model that contains multi-fact relationships, it's useful to refer to the model regularly on the Data Source page. The examples in this topic are based on

a data model with two base tables, Classes and Clubs, and two shared tables, Students and Rooms.



Classes

Classes
7 rows 3 fields

Class	Student (Classes)	Length
Nesting Basics	Robin	2
Advanced Songs	Sparrow	3
Flying For Fledgelings	Robin	6
Nesting Basics	Sparrow	2
Advanced Songs	Finch	3
Nesting Basics	Finch	2
Alarm Calls 101	*Null*	4

Fields:

- **Class**, a dimension with values of Nesting Basics, Advanced Songs, Flying for Fledglings, and Alarm Calls 101
- **Length**, a measure
- **Student**, a dimension used to relate to the Students table

Clubs

Clubs
7 rows 3 fields

Club	Student (Clubs)	Dues
Photography	Finch	50
Travel	Cardinal	90
Juggling	Sparrow	80
Art	Finch	10
Art	Cardinal	10
Art	Sparrow	10
First Aid	Robin	0

Fields:

- **Club**, a dimension with values of Photography, Travel, Juggling, Art, and First Aid
- **Dues**, a measure
- **Student**, a dimension used to relate to the Students table

Students

Students
5 rows 3 fields

Bus Rider	Student	Age
yes	Finch	3
yes	Cardinal	4
no	Sparrow	6
yes	Robin	3
no	Jay	8

Fields:

- **Bus Rider**, a dimension with values of yes or no
- **Student**, a dimension with values of Finch, Cardinal, Sparrow, Robin, and Jay. Used to relate to the base tables
- **Age**, a measure

Unrelated dimension-dimension pair

Unrelated dimensions are cross joined, which can result in combinations of dimension members across the headers that do not reflect actual combinations of data in the underlying tables.

The message for an unrelated dimension-dimension pair is:

- **On a shelf**: `Unrelated dimensions show all possible combinations of values. Unrelated to: <list of dimensions>`
- **In the Data pane**: `If used, this dimension will show all possible combinations of values with unrelated dimensions: <list of dimensions>`
- **Grayed out in the Data pane**: `This dimension isn't related to any dimensions in the viz. If used, it will show all possible combinations of values with other unrelated dimensions.`

Cross joins can be expensive operations that negatively impact performance. Because of this, Tableau also displays a Relatedness warning dialog if you add an unrelated dimension to the viz.

Stitching dimensions

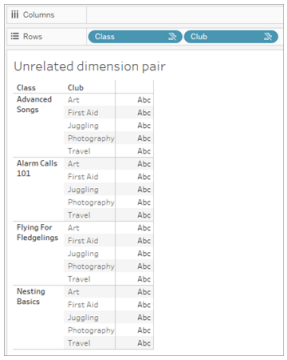
Although there are analytically relevant reasons to visualize unrelated dimensions alone, a common "happy path" for analysis with multiple base tables is to use a stitching dimension in addition. In the presence of a stitching dimension, the unrelated dimensions are no longer cross joined but are outer joined instead. Outer joins may still introduce nulls, but the dimension member headers are trimmed down from every possible combination to combinations that are relevant to at least one side of the outer join. They also don't have the same potential for performance impacts as cross joins. If there are no other relatedness issues that would call for an unrelated icon, a stitching dimension shows a related icon instead.

The message for a stitching dimension is:

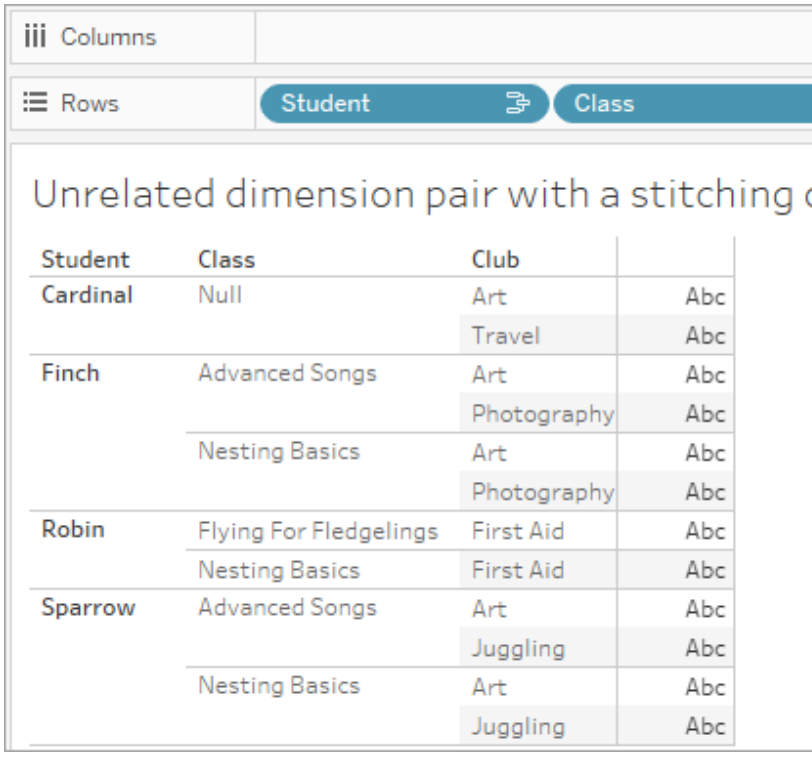
- **On a shelf**: `This dimension stitches together the following fields: <list of fields>`
- **In the Data pane**: `If used, this dimension will stitch together the following fields: <list of fields>`
- **Grayed out in the Data pane**: doesn't apply, stitching only occurs in a viz

Comparing unrelated dimensions with stitched dimensions

Unrelated: Cross join	Stitched: Outer join of intermediate inner joins



A viz showing a cross join of Class and Club with rows for every combination of Advanced Songs/Alarm Calls 101/Flying for Fledglings/Nesting Basics with Art/First Aid/Juggling/Photography.



A viz showing the results of an outer join of the Student-Class inner join and the Student-Club inner join. Not all combinations of classes and clubs are represented, and there are rows for students and clubs without a class.

An aside on how measure values are computed

When a measure isn't related to a dimension, it can't be broken down by that dimension's members (that is, you can't break down the average class length per club when clubs don't have a class length). Instead, the measure will be aggregated at a different level than the dimension member's headers in the view.

Terminology for dimensions and measures

In Tableau, *measures* are *aggregations*—they're aggregated up to the *granularity* set by the *dimensions* in the view. The value of a measure therefore depends on the context of the dimensions. For example, "number of cereal boxes" depends on if we mean the total inventory or the number of boxes per brand.

Dimensions are usually categorical fields, such as country or brand. In Tableau, dimensions set the granularity, or the *level of detail*, of the view. We typically want to group our data into marks by some combination of categories. What dimensions we use to build the view determines how many marks we have.

Aggregation refers to how the data is combined. Tableau's default aggregation is SUM. You can change the aggregation to other options, including: average, median, count distinct, minimum, and so on. *Granularity* refers to how detailed or broken down the measure is—which is controlled by the related dimensions. Unless the granularity of the measure is row level (aka disaggregated), its value must be aggregated.

Example

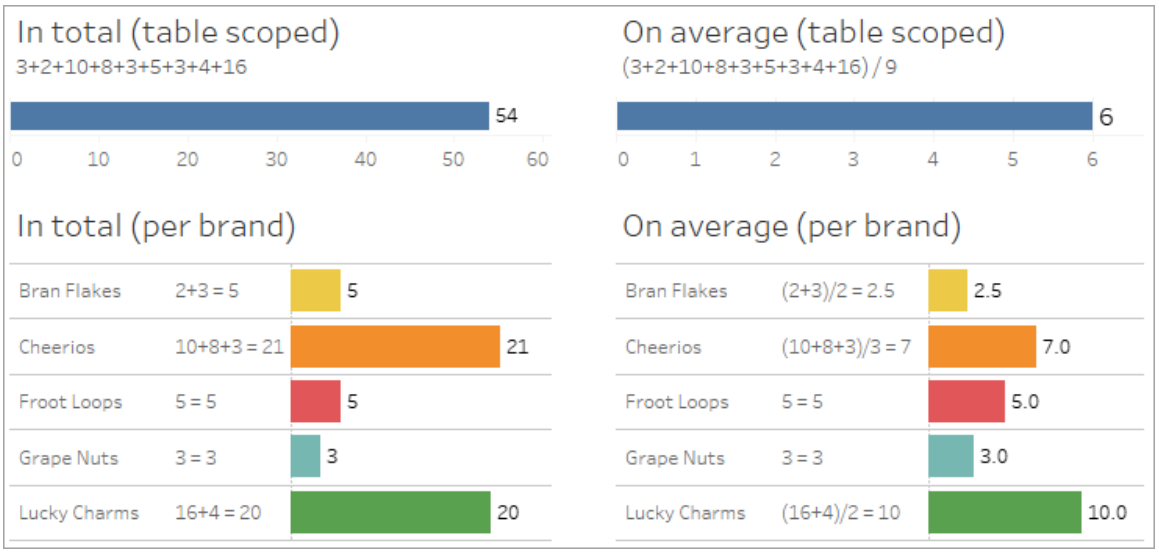
Raw data

	Mini	Normal	ValuePak
Bran Flakes		3	2
Cheerios	10	8	3
Froot Loops		5	
Grape Nuts		3	
Lucky Charms		4	16

What's the value of "number of boxes of cereal"?

Well, it depends on the aggregation type and the granularity as set by the dimensions.

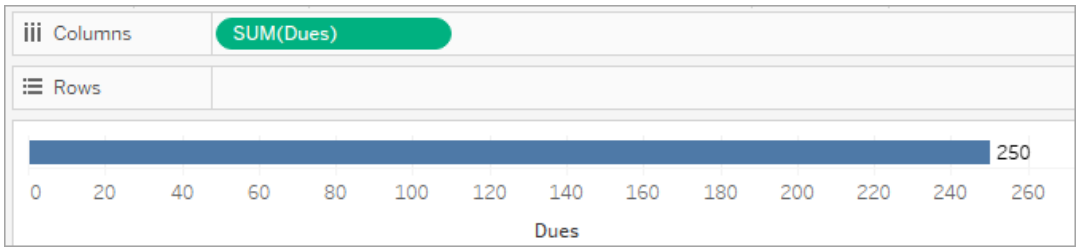
- Aggregations:
  - Sum (or total)
  - Average
- Granularity:
  - Table scoped / fully aggregated (the blue bars in the example)
  - Broken down by the **Brand** dimension (the colored bars in the example)



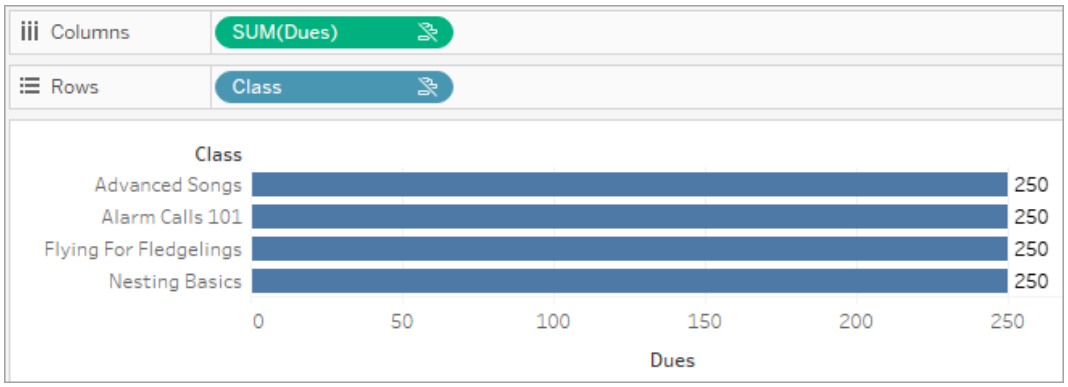
The value of a measure trails the dimension members

A measure's value is determined by the dimensions it is related to. A measure without a related dimension is table scoped. A measure with a related dimension is broken down by the related dimension's members (that is, the value of the measure is computed for each dimension member). If a related dimension's members are repeated due to the presence of an unrelated dimension, the measure's values are repeated based on its dimension members.

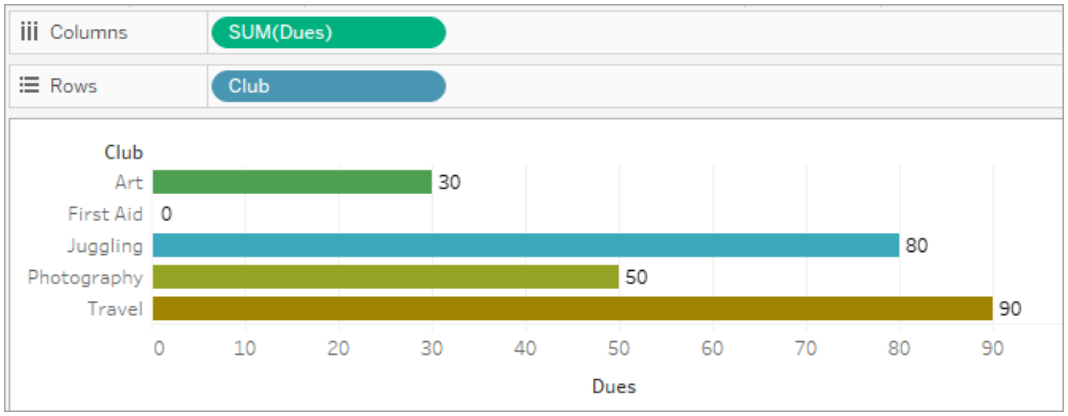
1. A measure without a dimension is table scoped to its overall value.



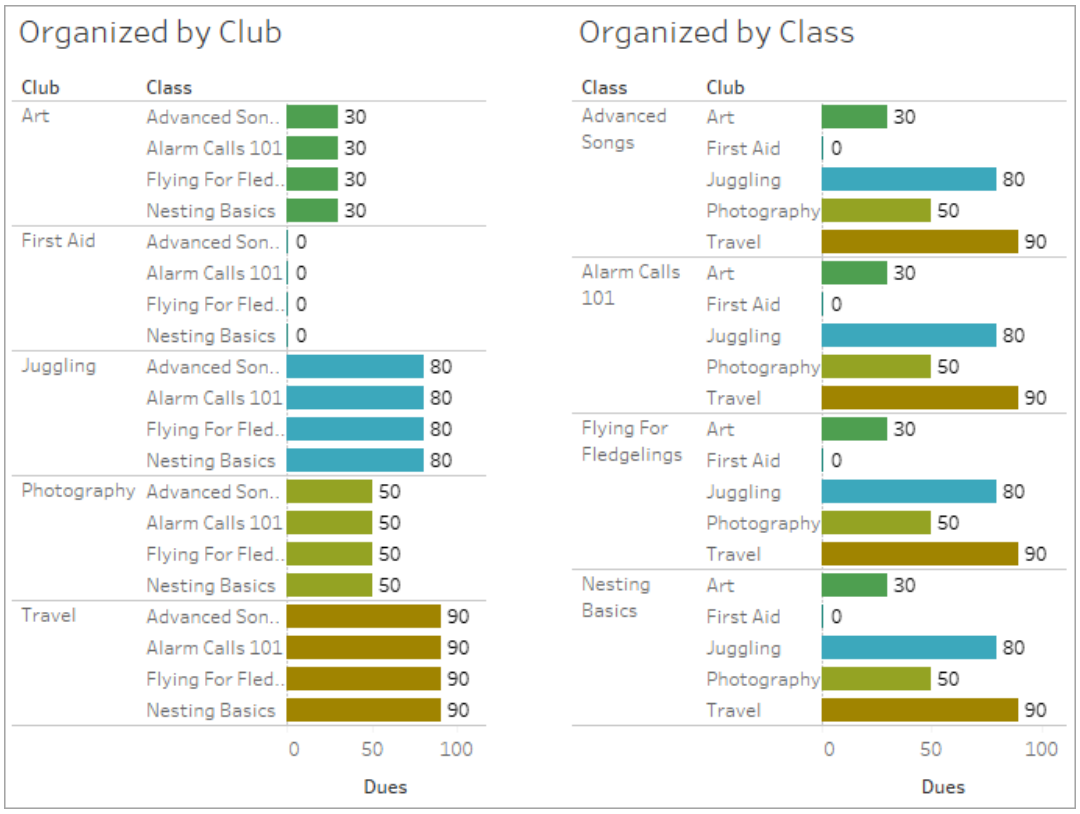
2. A measure, in the presence of an unrelated dimension alone, is table scoped and repeated for the unrelated dimension's members.



3. A measure in the presence of related dimension is broken down more granularly and its value is computed per member of the related dimension.



4. A measure, in the presence of an unrelated dimension and a related dimension, is broken down by the dimension it's related to. Wherever those related dimension members are repeated for unrelated dimensions, the measure value trails along with its related dimension member.



Because dues are per club, the value of dues for each club is repeated every time that club is repeated.

Unrelated dimension-measure pair

The message for the measure is:

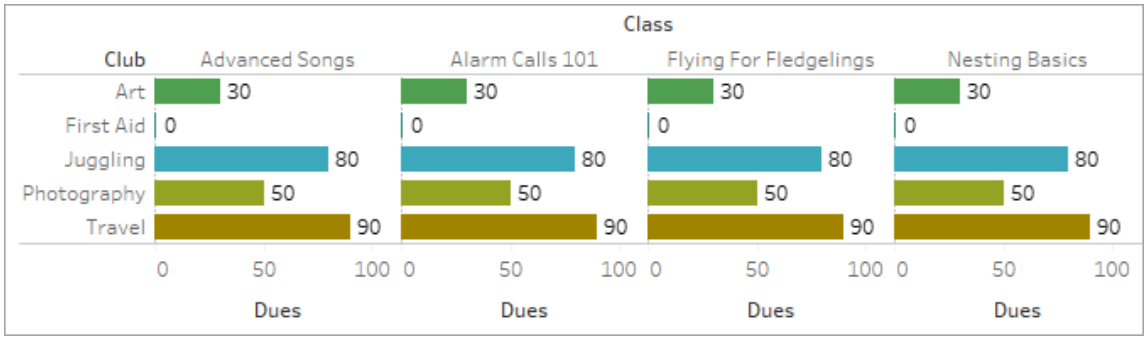
- **On a shelf**: `This measure can't be broken down by unrelated dimensions: <list of dimensions>`.
- **In the Data pane**: `If used, this measure won't be broken down by unrelated dimensions: <list of dimensions>`
- **Grayed out in the Data pane**: `This measure isn't related to any dimensions in the viz. If used, it won't be broken down.`

The message for the dimension is:

- **On a shelf**: `This dimension can't break down unrelated measures: <list of measures>`

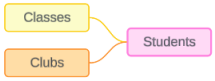
- **In the Data pane**: `If used, this dimension won't break down unrelated measures: <list of measures>`
- **Grayed out in the Data pane**: `This dimension isn't related to any measures in the viz. If used, it won't break down measure values.`

The result in a viz is a repeated value for the measure across the unrelated dimension's values. This behavior is similar to when an LOD expression is used to set the level of aggregation for a measure at a different level of detail from the native granularity of the viz. An unrelated dimension is essentially EXCLUDED from the computation of the measure's aggregated value.



Measure from a shared table

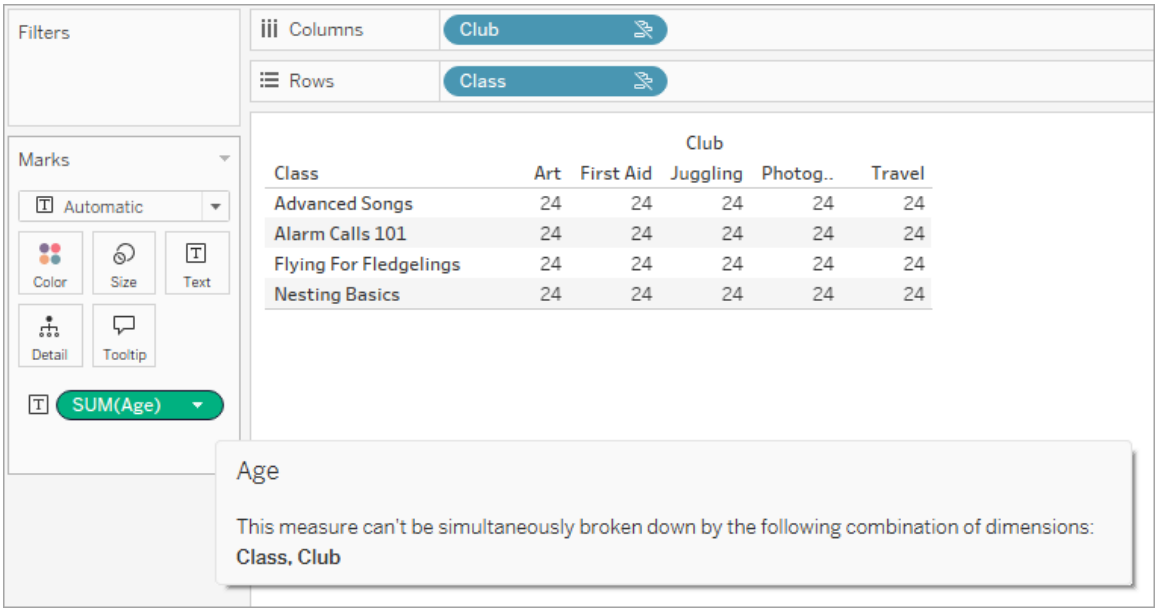
When a dimension from a shared table (like Students) is used, it stitches together the dimensions from otherwise unrelated tables (such as Classes and Clubs). But what if instead of a dimension, you use a measure from the Students table instead?



Measures can't stitch. Additionally, their value is determined by their related dimensions. In a case where there are unrelated dimensions visualized together, the measure can't be broken down by those dimensions simultaneously. In this case, we treat the measure as unrelated to the combination of dimensions even though it would be related to either dimension individually.

The message for a measure shared across unrelated dimensions is:

- **On a shelf**: `This measure can't be simultaneously broken down by the following combination of dimensions: <list of dimensions>`
- **In the Data pane**: `If used, this measure won't be broken down by the following combination of dimensions in the viz: <list of dimensions>`



To resolve this and prevent the measure from being table scoped, the unrelated dimensions could be stitched or one or more dimensions could be removed until there is a clear relationship path for aggregating the measure.

Related measure

When a measure is related to some dimensions in the viz but not others, the measure may have an additional message in the tooltip that explains what dimensions it is related to. This can help explain how the measure is aggregated. This message only appears when the measure is also unrelated to a dimension in the viz. Otherwise, it's standard behavior that the measure is aggregated to the level of detail of its related measures.

- **On a shelf**: `This measure is aggregated to the level of detail of related dimensions: <dimensions in the viz this measure is related to>`

- **In the Data pane**: `If used, this measure will be aggregated to the level of detail of related dimensions: <dimensions in the viz this measure is related to>`

This message is intended to help identify which dimension or dimensions are considered when the measure value is computed. In the example of clubs and dues and classes, the tooltip for the measure clarifies the value is aggregated at the level of detail of Club and repeated for the dimension Classes.

Filters

Relatedness is also evaluated for fields on the filter shelf compared to fields otherwise active in the viz.

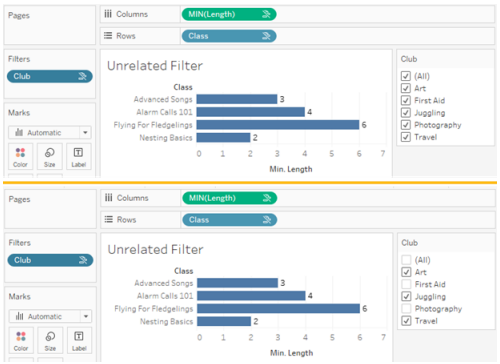
An icon and tooltip appears when a filter is unrelated to at least one field in the viz. Both the filter field and field in the viz have a tooltip.

The message for an unrelated filter is:

- **On the filter shelf**: `This filter doesn't apply to unrelated fields: <fields>`
- **In the viz**: `This field isn't filtered by unrelated filters: <fields>`
- **In the Data pane**: `If used, this field will be ignored by unrelated filters: <fields>`

The behavior of filters also depends on their relatedness to other fields in the viz. A filter doesn't impact the values of fields it isn't related to. Unless the filter is set to no values (exclud-

ing everything or including nothing), the viz will remain unchanged for any fields that aren't related to the filter. However, deselecting every option in the filter will return a blank viz.



Two screenshots of viz with an unrelated filter, showing that deselecting options in the interactive filter control doesn't impact the viz

Related fields are filtered as expected. In a more complex viz with a combination of related and unrelated fields (such as in a stitching context), the filter will only impact values that are related to the filter field.

Build a Multi-fact Relationship Data Model

Analysis often involves bringing together tables of data that don't have a direct relationship to each other yet both relate to the same, common information such as date or location. This type of analysis is sometimes referred to as multi-fact analysis with shared dimensions.

To perform this kind of analysis in Tableau, you need to create a data source that uses multiple base tables connected by shared tables.

- *Base tables* are the left-most tables in the data model on the Data Source tab. For guidance on how to determine which tables to use as base tables, see When to Use a Multi-fact Relationship Model.
- *Shared tables* are downstream tables with multiple incoming relationships. These tables contain fields that can be used to stitch together unrelated fields during analysis in a viz. Date and Location are examples of commonly shared tables.

Build the model

Building a multi-fact relationship data model is essentially the same as creating any other data source that uses relationships, but with two additional pieces: additional base tables and multiple incoming relationships to shared tables.

1. Connect to your data. You can use multiple data connections if your tables aren't all in the same database.
2. Drag a table onto the canvas to create the first base table.
3. Drag another table from the left pane to the **New Base Table** drop area.

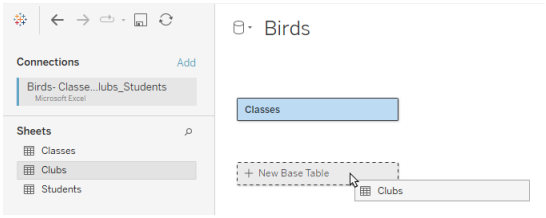
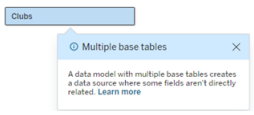
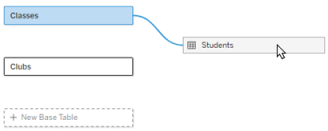


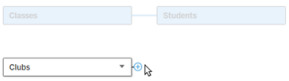
Tableau shows a warning that you're building a data model with multiple base tables. We recommend that you only set up a multiple base table model if your data needs it. Otherwise, use a single base table model to avoid the complexities that come with a multiple base table model.



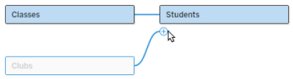
4. Drag another field to the canvas and relate it to one of the base tables. Configure each relationship if necessary.



5. Hover over the not-yet-related base table to display a plus sign (affectionately known as a "meatball").



6. Drag the plus sign icon to the shared table to create a new incoming relationship (also known as a "noodle").



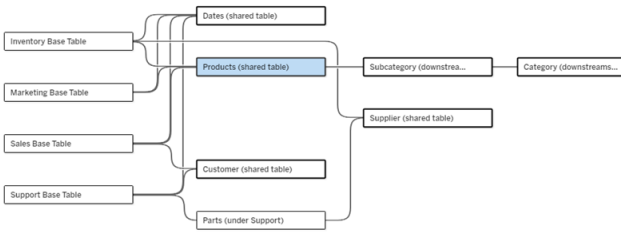
7. *Optional:* Continue adding downstream tables and base tables. Click and drag the plus sign icon to build shared tables as needed.

> **Note**: Rather than starting with the base tables, you can also build a single base table model first and add additional base tables later.

Explore the model

A data model with multiple base tables is built with relationships, but not all tables are related to each other to the same degree. Each base table defines a *tree*, which contains every table related to that base table, either directly related or downstream from a related table. Shared tables exist in multiple trees. For more information about degrees of relatedness, see About Multi-fact Relationship Data Models.

When you view a model with multiple base tables, there are various options for exploring and managing the data model. These options are especially useful when a data model is complex.



An example of a complex data model with four base tables, multiple shared tables, and down-stream tables that are both shared and unshared between the base tables.

2024.2 introduced some new layout details for the data model. In a multi-fact relationship data model, relationships bundle together to help track how many incoming relationships a table

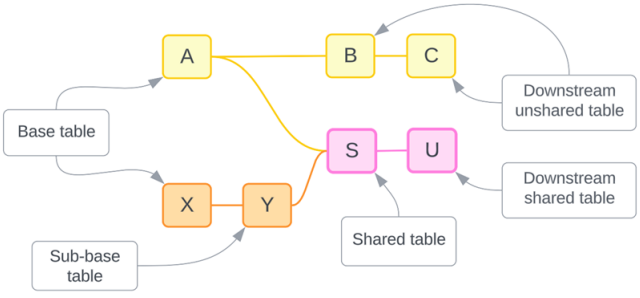
has, and shared tables (and downstream shared tables) have a bolder outline than tables that aren't shared.

Terminology

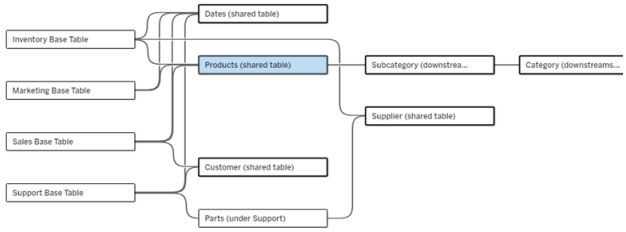
Tables in a multiple base table data model have specific roles. Base tables and shared tables are the only tables that must exist in a multi-fact relationship data model. (Without more than one base table, it's not a multi-fact relationship data model. Without a shared table connecting the base tables, it's not a valid data model.)

Because of the often complex nature of these models, it's useful to have a shared terminology for discussing other types of tables by how they fit into the data model.

- *Base tables* are on the far left and have no incoming relationships.
    - In the example, Inventory, Marketing, Sales, and Support are base tables.
- *Sub-base tables* are between a base table and a shared table.
    - In the example, Parts is a sub-base table.
- *Shared tables* have more than one incoming relationship.
    - In the example, Products, Dates, Customer, and Supplier are shared tables.
- *Downstream shared tables* have exactly one incoming relationship and have a shared table somewhere upstream of them.
    - In the example, Subcategory and Category are downstream shared tables.
- *Downstream unshared tables* have exactly one incoming relationship and have no shared tables upstream of them.
    - In the example, there are no downstream unshared tables.



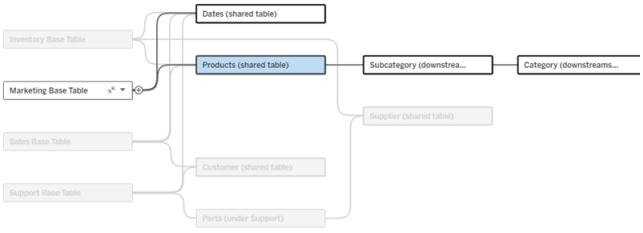
Test your understanding: identify the types of tables in the example data source



- **Base tables**: Inventory, Marketing, Sales, and Support
- **Sub-base table:** Parts
- **Shared tables:** Products, Dates, Customer, and Supplier
- **Downstream shared tables:** Subcategory and Category
- **Downstream unshared tables:** none

Identify a relationship tree

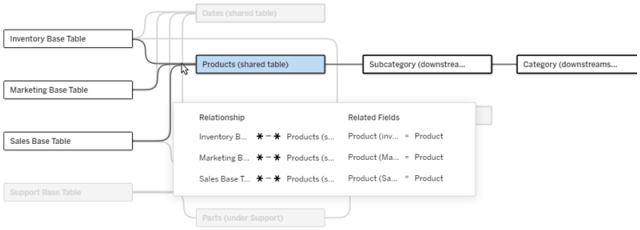
Hover over a table to highlight the tables it's related to. Tableau emphasizes the tree for that table and deemphasizes unrelated tables.



The Marketing base table tree consists of two shared tables, Dates and Products, and downstream shared tables Subcategory and Category.

View relationship details

Hover over a relationship or bundle of relationships to see the details in a tooltip.



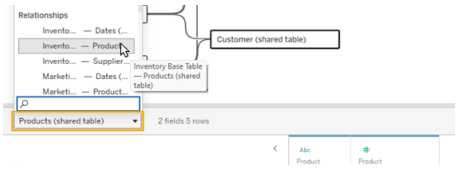
The incoming relationships to the Products table include three individual relationships to the Inventory, Marketing, and Sales base tables.

Select a relationship

When a relationship is selected, it is highlighted in blue in the data model canvas and its details appear in the Table Details pane below the data model canvas. The Table Details pane is where you can inspect or modify the relationship clause.

There are multiple ways to select a relationship:

- Click a relationship line (noodle) in the canvas. Every relationship has a clickable zone that selects just that noodle.
- Right-click or control-click a table in the canvas to open its menu. Select the **Select Relationship** option and choose which table's relationship you want.
- Click a bundle of relationships in the canvas to bring up a persistent tooltip (hovering over a bundle brings up the tooltip, you have to click to make it persist). Then select a row in the tooltip details to highlight that relationship in the model.
- Open the menu in the toolbar of the Table Details pane and select the desired relationship. You can also use this menu to select a specific table to see its preview in the pane.



Swap with base table

Intermediate tables in a relationship between a base table and shared tables give you the option to swap the downstream table with the base table. This is purely a visual change to aid with conceptual understanding and doesn't change the structure of the data model.

Right-click or control-click a downstream table and select **Swap with base table (table name)**. The swap option is also only present on the downstream table and not the base table.

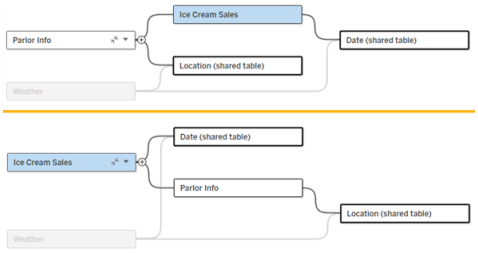
The swap option is not available for tables which would alter the data model if swapped, such as shared tables or downstream shared tables. Only downstream unshared tables or sub-base tables can be swapped with base tables.

Example

In this example, the **Parlor Info** and the **Ice Cream Sales** tables can be swapped without changing the data model's fundamental structure. No other tables can be swapped.

- Ice Cream Sales is related to both Parlor Info and the shared Date table.
- Parlor Info is related to both Ice Cream Sales and the shared Location table.
- Weather is related to both the shared tables of Date and Location.

These two models are conceptually equivalent:



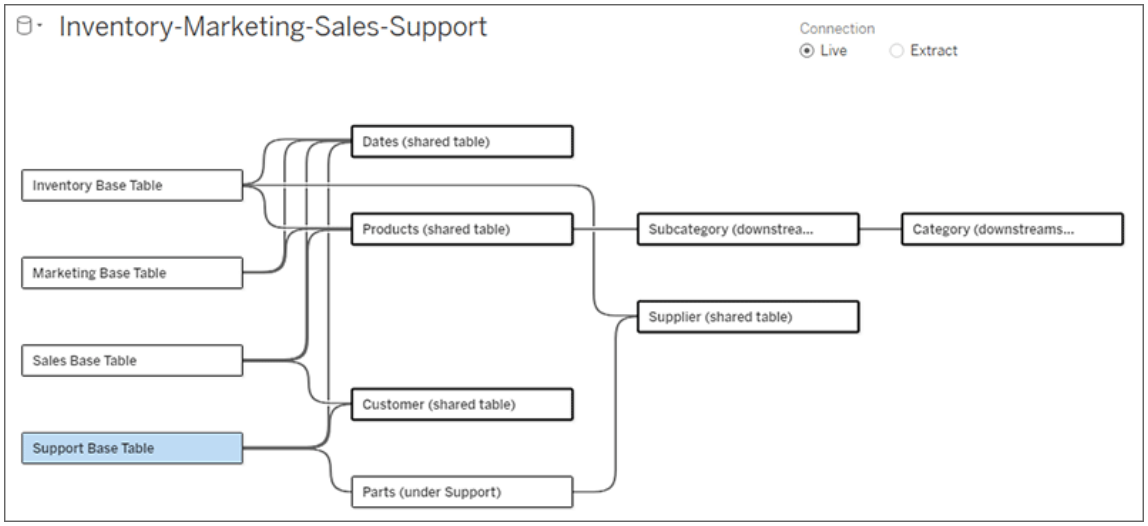
Collapse a base table

You can also collapse a relationship path, or tree, to just its base table to temporarily simplify the view of the data model.

Click the Collapse or Expand buttons on a base table to collapse or expand its entire tree. Alternatively, right-click or ctrl-click on a downstream table and select **Collapse this path** or **Collapse other paths**. This option is not available on shared tables or tables downstream of shared tables.

Collapsing a tree to its base table is purely visual and won't trigger the Unrelated Tables alert. A collapsed path is indicated by a base table with a stacked table and an Expand button. Collapsing affects all tables and relationships that are relevant only to that tree, so tables that are shared with an uncollapsed path are not hidden.

Use the Expand button to re-open the base table and everything in the tree.



Troubleshooting

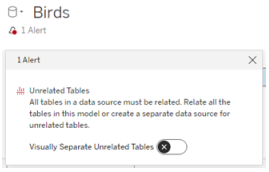
Create a single data source

All tables must be related to the entire data source in some way. If there are any tables that aren't related to the overall data model, an alert appears. The alert remains until no tables or trees are fully separate from the rest of the data model. When the alert is active, the data source can't be published and you can't use the data source in an analysis.

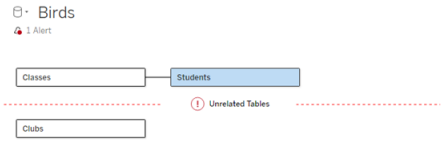
You can open the alert and set Visually Separate Unrelated Tables to identify which tables triggered the alert. This option is useful when you have a complex model and need to identify which table or tables aren't yet related to the rest of the data model.

Example

In the steps under Build a model, an alert displays in Step 4 before a relationship is added to connect the second base table.

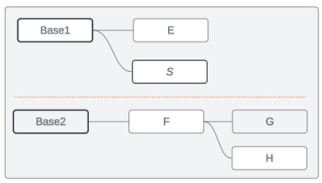


Setting **Visually Separate Unrelated Tables** to On moves the table **Clubs** underneath the **Unrelated Tables** line. Relating **Clubs** to **Students** resolves the alert.

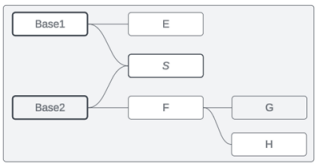


Resolve a cycle

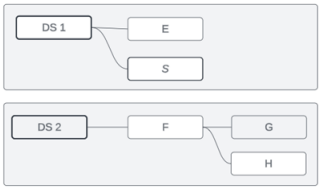
Even if some tables aren't directly related to each other, the entire data model must be a cohesive whole. In this example, each base table defines a tree but there is no shared table connecting them. This isn't a valid model for analysis.



The two groups of related tables need to be combined via a shared table...



...or the data model needs to be created as two separate data sources.

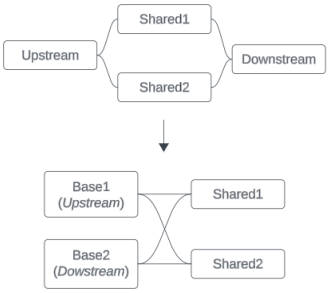


Data model restrictions

Some relationship paths between tables are not supported in a multi-base table model. If you're unable to drop the meatball when you attempt to create a second incoming relationship on a table, make sure the structure you're trying to create is supported in Tableau. Examples of unsupported models include:

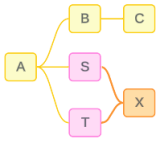
Cycles

Cycles—where there's more than one relationship path from an upstream table to the same downstream table—are not supported. This unsupported structure is sometimes called a bowtie. To model this kind of relationship between tables in Tableau, use multiple base tables instead of a bowtie by converting the downstream table to another base table.

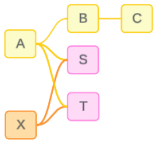


The data model must be a *directed acyclical graph*. This means every incoming relationship to a table must be traceable upstream to a different base table.

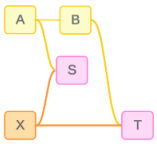
Not supported. Table X has two incoming relationships that are both from tables downstream from Base Table A



Supported. Tables S and T both have multiple incoming relationships, but each one is from a different base table.

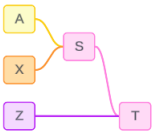


Supported. Although it may visually look like there's a cycle, the incoming relationships for both Table S and Table T can be traced back to different base tables.



Nested shared tables

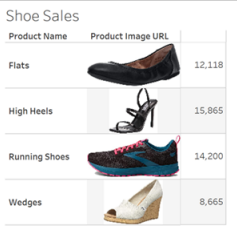
Nested shared tables are not supported. Any tables downstream from a shared table (a table with multiple incoming relationships) can only have one incoming relationship.



Not supported. Table T is downstream from a shared table and can't receive an additional incoming relationship.

Add Web Images Dynamically to Worksheets

At Tableau, we know that images are a powerful tool when analyzing data. Imagine that you're looking at a viz of monthly shoe sales. The data is telling you that you sold more high heels than wedges, but you can't picture the difference in the two types of shoe. That's where Image Role comes in. You can dynamically add web images to your worksheets and use them in your headers to add visual detail.



Prepare your data source

Image Role can be assigned to discrete dimension fields that contain URLs that point to web images. To prepare your data, be sure that your image fields meet the requirements set by Tableau to be assigned an image role:

- Make sure your URLs navigate to .png, .jpeg, .jpg, .svg, .webp, .jfif, .ico, or .gif image files.
- Verify that each URL begins with http or https. If a transport protocol isn't included, Tableau assumes https.
- Optimize the number of images used in your data set. Usually, you can load up to 500 images per field.
- Ensure that each image file is smaller than 200 kb.

Note: If you're using Tableau 23.1 or earlier, your URLs must navigate to image files with .jpg, .jpeg, or .png file extensions.

In Tableau 23.2 and later, .gif files are supported, but .gif *animations* will only show on Tableau Cloud and Tableau Server with a client-side render. In Tableau Desktop and Tableau Server with a server-side render, the .gif file will show as a static image.

Depending on the complexity of your viz, Tableau may default to a server-side render, which limits the number of images to 100 per field. To learn more about complexity settings and server-side rendering, see Configure Client-Side Rendering.

Example data set:

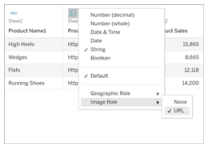
Product Name	Product Image URL	Product Sales
Flats	https://img.example.com/flats.png	12,118
High Heels	https://img.example.com/highheels.png	15,865
Running Shoes	https://img.example.com/runningshoes.png	14,200
Wedges	https://img.example.com/wedges.png	8,665

Assign an image role to your URLs

After you've connected to your data source, you can assign an Image Role to your URLs from either the Data Source page or on the Data pane in a worksheet.

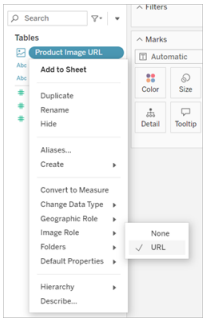
From the Data Source page:

1. Locate the column that has image URLs.
2. Right-click (control click on Mac) the icon on the top left of the column and select **Image Role** > **URL**.
   The icon changes to an image icon, and your images are ready to use.



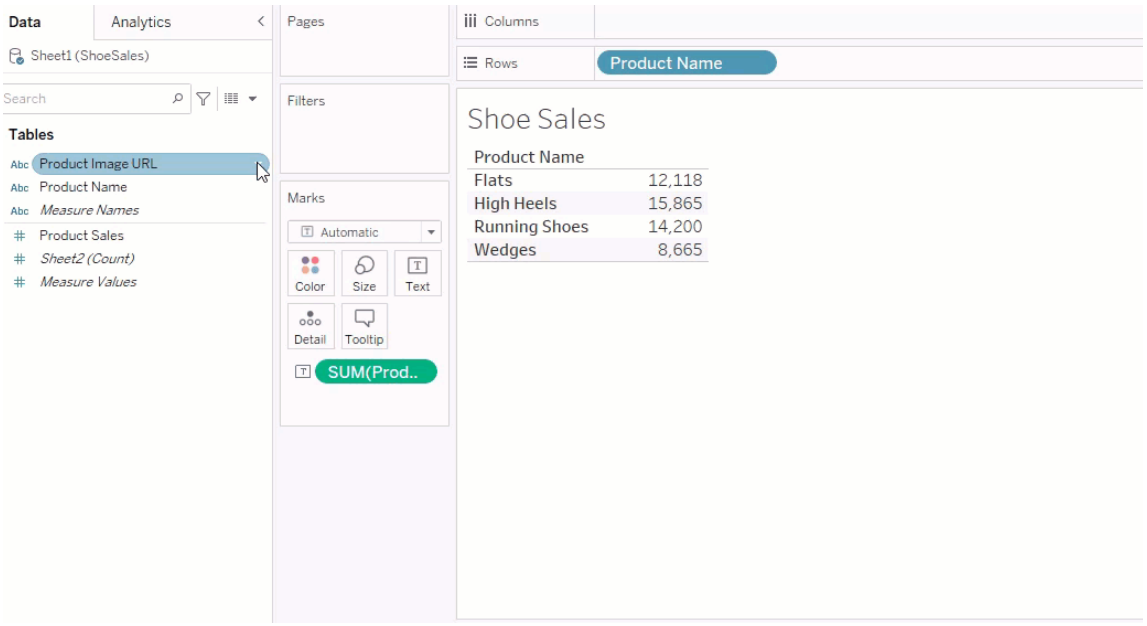
From a worksheet:

1. Open a new worksheet.
2. Locate a discrete dimension field that has image URLs.
3. Right-click (control click on Mac) the dimension field and select **Image Role** > **URL**.
   The icon changes to an image icon, and your images are ready to use.



Add images to your visualizations

From your worksheet, drag the Image Role field onto the Rows or Columns shelf. You can now see images along with their associated data on your viz.



Share your visualizations

You can export your workbooks with images and share them. Be sure that the images used in your vizzes are available to be viewed by everyone you share your vizzes with. For example, if you're using images that are hosted on an internal server behind a firewall, be sure that everyone you share the viz with has sufficient permissions to access the images. In this

example, users who view the viz while connected to the same server shouldn’t have any issues seeing the images. But users who export the viz to a pdf on a Tableau Cloud server may not be able to see the images.

If you export your workbook and your viewer attempts to open it in Tableau 2022.3 or earlier, your viewer won’t be able to see the images.

Note: It’s a best practice to always align an Image Role field with a text description to make the content accessible for screen readers and other accessibility software.

Troubleshoot image connections

Sometimes images won't display if you exceed the number of images allowed per field, have large image files on a complex viz, or are viewing the viz on a mobile device. This section helps you troubleshoot those errors.

None of the images are displaying in my viz

There are too many images in the viz

Depending on the complexity of your viz, you can typically load 500 images per field. If you have a complex viz, it may default to server-side rendering. With server-side rendering, you can load 100 images per field.

If you get an error message that there are too many images in the viz, filter out images and try again.

Tableau couldn't access the images

If you receive broken image icons instead of your images, first confirm that you have sufficient permissions to view the images (or if, for example, they're behind a firewall). Tableau must be

able to access the images, and the images can't require a separate authentication to view them.

If you're sure that you have permission to view the images and they're within the size requirements, check that web images are enabled on your settings page.

In Tableau Desktop:

1. From your workbook, click **Help** in the toolbar.
2. Select **Settings and Performance** > **Set Dashboard Web View Security**.
3. Make sure that **Enable Web Page Objects and Web Images** is checked.

In Tableau Cloud:

1. From the home page, click **Settings**.
2. Under general, scroll down and locate **Web Page Objects and Web Images**.
3. Make sure that **Enable Web Page Objects and Web Images** is checked.

Some of the images aren't displaying in my viz

The image file is too large

Each image file must be smaller than 200 kb to render. Check your image file size and try again.

You're using an earlier version of Tableau

If you’re using Tableau 23.1 or earlier, only .png, .jpeg, and .jpg image files are supported. Upgrade your version of Tableau or use a supported file type for the version of Tableau you're using.

The image URL must begin with http or https

Each image URL must begin with either http or https. Tableau doesn't currently support FTP/SMTP calls. Verify your URL format and try again.

The image file must be a URL

An Image Role can be assigned to only URLs that navigate to .png, .jpeg, or .jpg, .svg, .webp, .jfif, .ico, .bmp, or .gif image files. Verify your URL format and try again.

The image file type isn’t supported

An Image Role can be assigned to only URLs that navigate to .png, .jpeg, or .jpg, .svg, .webp, .jfif, .ico, .bmp, or .gif image files. Verify your URL format and try again.

If you’re using Tableau 23.1 or earlier, only .png, .jpeg, and .jpg image files are supported. Upgrade your version of Tableau or use a supported file type for the version of Tableau you're using.
In Tableau 23.2 and later, .gif files are supported, but .gif *animations* will only show on Tableau Cloud and Tableau Server with a client-side render. In Tableau Desktop and Tableau Server with a server-side render, the .gif file will show as a static image.

The image file contains bad characters

An Image Role can't be assigned to URLs that have the following characters:

```
< > & \ ^ '
```

or the following character sequences:

```
.. \\. \r \n \t
```

Verify that your URL doesn't contain any of these characters or character sequences and try again.

The images aren't displaying outside of my worksheet

The images aren't displaying in Viz in Tooltip

Viz in Tooltip is processed with server-side rendering, which allows you to load up to 100 images per field. Verify that you have fewer than 100 images per field and try again.

For more about server and client-side rendering, see Configure Client-Side Rendering.

For more about using Viz in Tooltip, see Create Views in Tooltips (Viz in Tooltip).

The images aren't displaying in View in Thumbnail

View Thumbnail is processed with server-side rendering, which allows you to load up to 100 images per field. Verify that you have fewer than 100 images per field and try again.

For more about server and client-side rendering, see Configure Client-Side Rendering.

The images aren't displaying when I export my workbook

All exports and export-related features are processed with server-side rendering, which allows you to load up to 100 images per field. Verify that you have fewer than 100 images per field and try again.

The images aren't displaying on a mobile device

Mobile devices have a lower complexity threshold than computers, so the processing is completed through server-side rendering, which allows you to load up to 100 images per field. Verify that you have fewer than 100 images per field and try again.

You can change the complexity settings on your mobile device. For more info on complexity thresholds, see Configure the complexity threshold for computers and mobile devices.

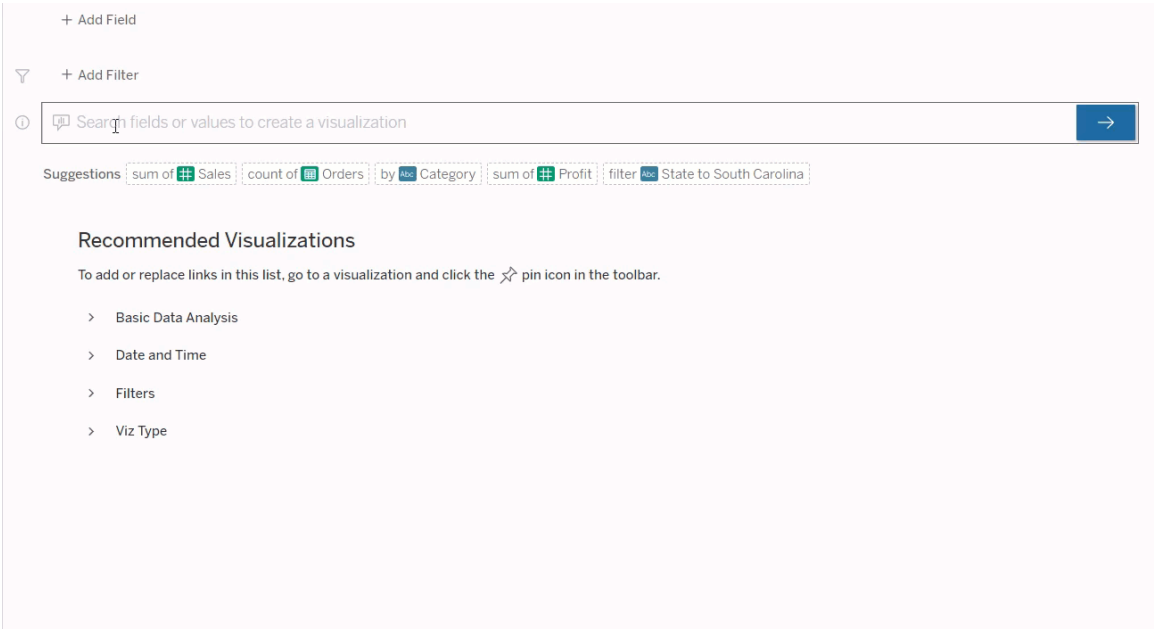
Automatically Build Views with Ask Data

Important changes for Ask Data and Metrics

Tableau's Ask Data and Metrics features were retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2. With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see *How Tableau AI and Tableau Pulse are reimagining the data experience.*

Ask Data lets you type a question in common language and instantly get a response right in Tableau. Answers come in the form of automatic data visualizations, with no need to manually drag-and-drop fields or understand the nuances of your data's structure.

Ask Data lets you ask sophisticated questions naturally, with support for key analytical concepts such as time series and spatial analysis, and an understanding of conversational phrases such as "last year" and "most popular."

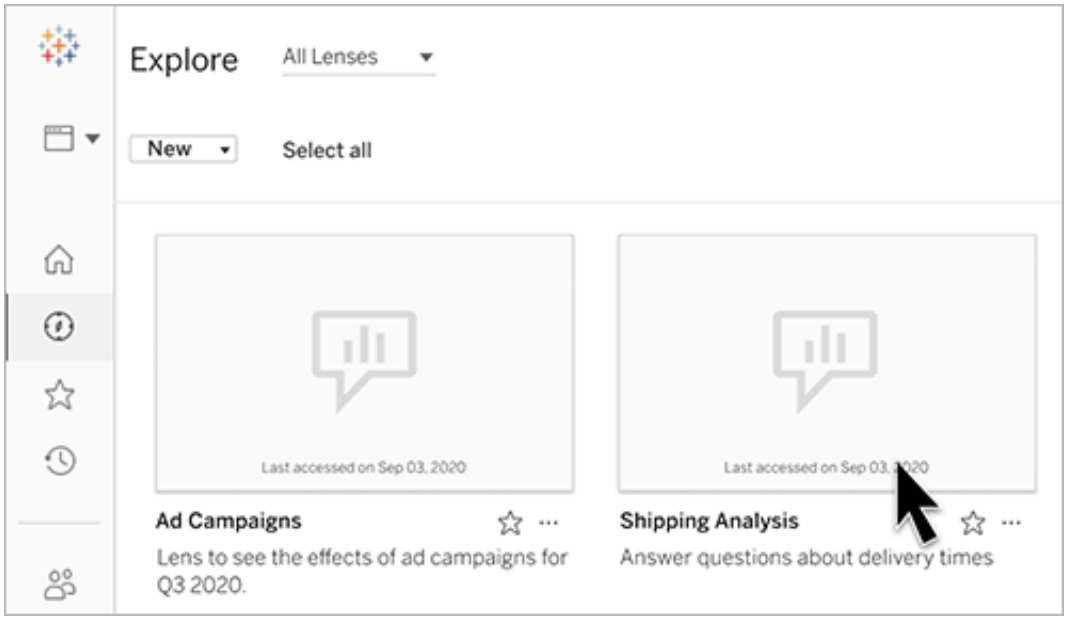


Navigating to Ask Data lenses

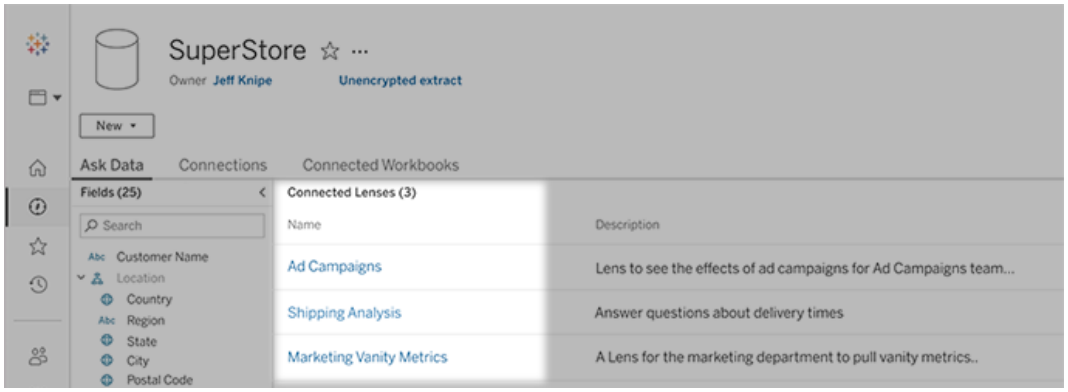
Before you can query a data source with Ask Data, a Tableau author must first create a lens that specifies the subset of data fields the lens uses.

In Tableau, here are all the places where you can access an Ask Data lens:

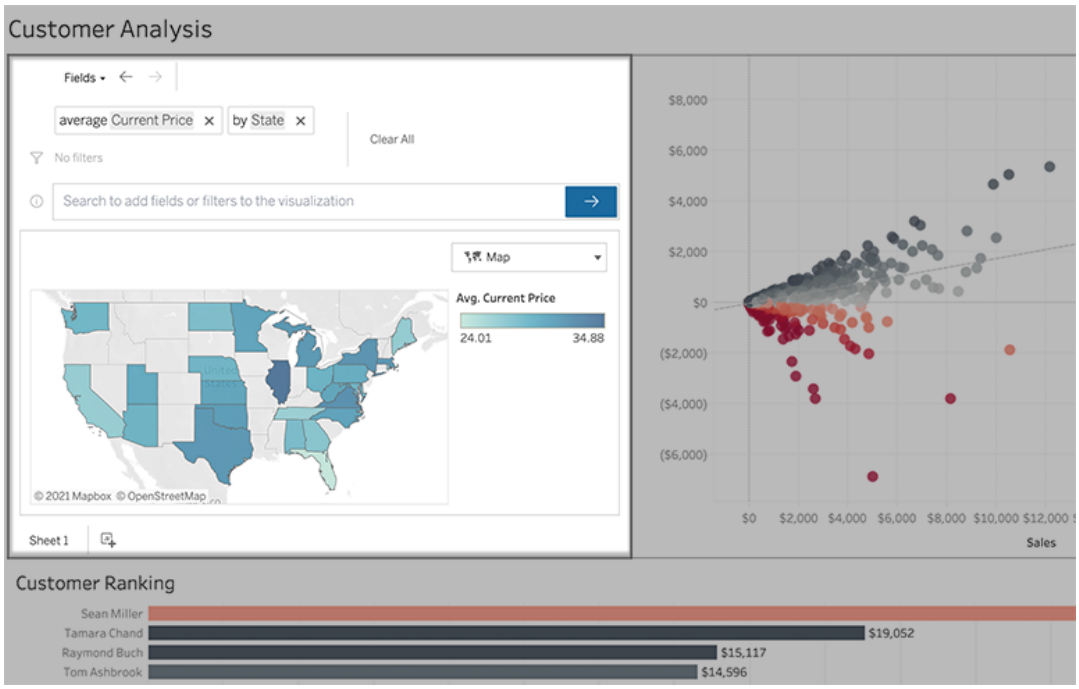
- On the All Lenses page at the top level of your Tableau Cloud or Tableau Server site.



- On the Ask Data tab for a data source for which lenses have been created.



- In an Ask Data object on a dashboard.



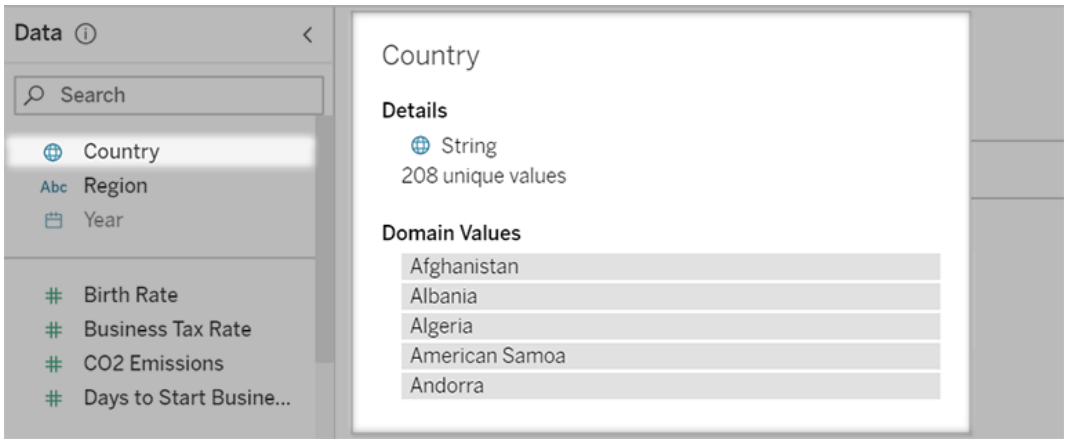
Ask Data from a lens page or dashboard object

Navigate to a lens and learn more about its data

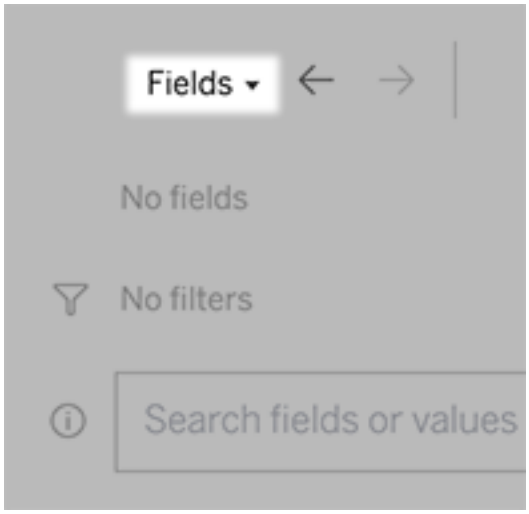
1. Navigate to a lens via the All Lenses page for your Tableau site, the Ask Data tab for a data source, or an Ask Data object on a dashboard.
2. (Optional) Under **Recommended Visualizations**, click an entry to quickly see visualizations the lens author has created for your organization.

   If the recommendations don't address your current data analysis needs, build a query to create your own question.

3. In the Data pane at left, briefly hover over each field to learn more about the data it contains.

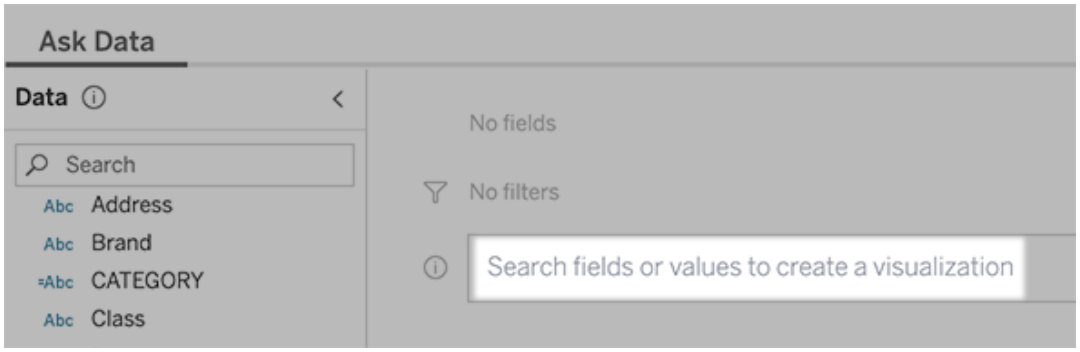


In a narrower dashboard object, the Data pane may be hidden, but you can see the same information by clicking the **Fields** drop-down menu.

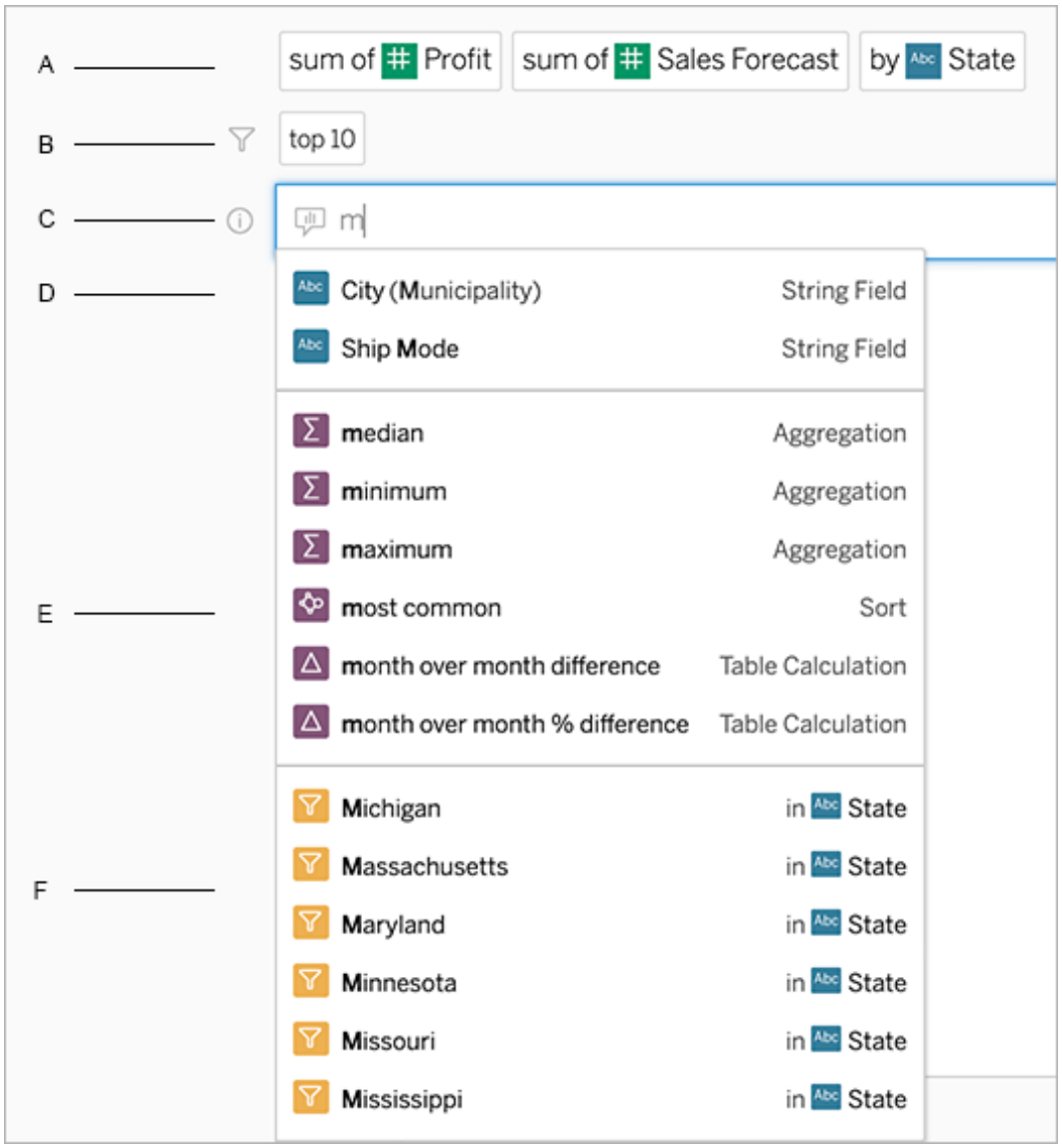


Build queries by entering text

1. Type in the box reading **Search fields or values to create a visualization**.



2. As you type, Ask Data searches data fields, functions, and string values and displays results in a drop-down list. Click items in the list to add them to your current entry, shown above the search box. To automatically create a viz using the current entry, press **Enter** at any time.



Modifying a query by searching for fields and analytical functions

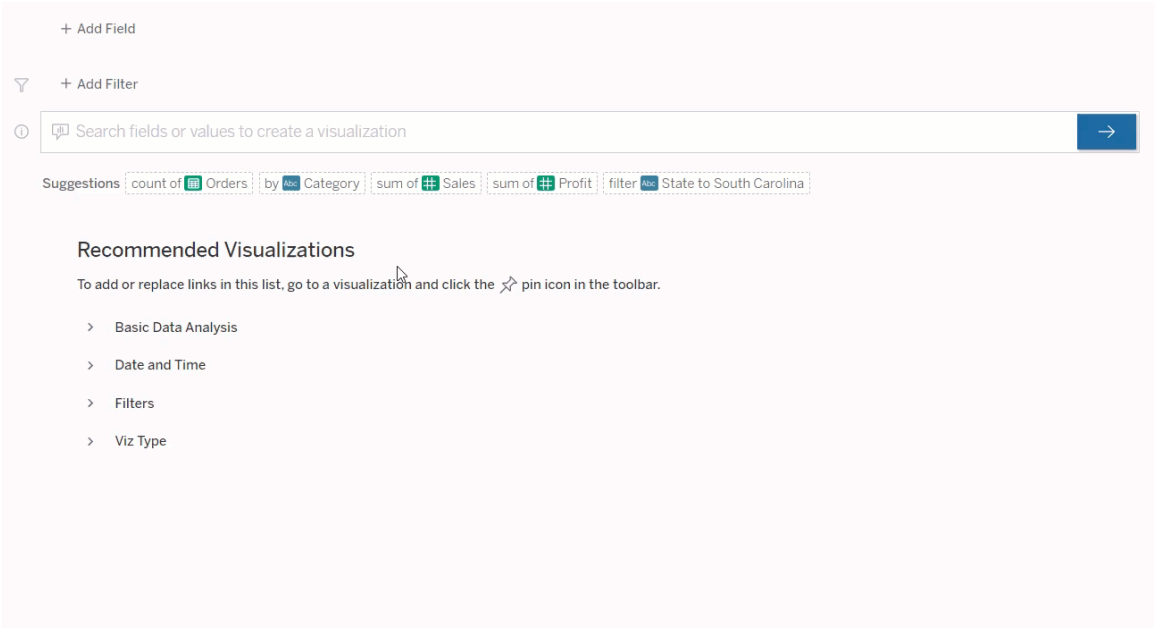
A. Current entry B. Current filters C. Search box D. Returned fields E. Returned analytical functions F. Returned field values

Build queries by adding suggested phrases

Tableau will suggest phrases based on the most common queries asked in your lens and by others in your organization. When you open your lens, you'll notice suggestions that will help

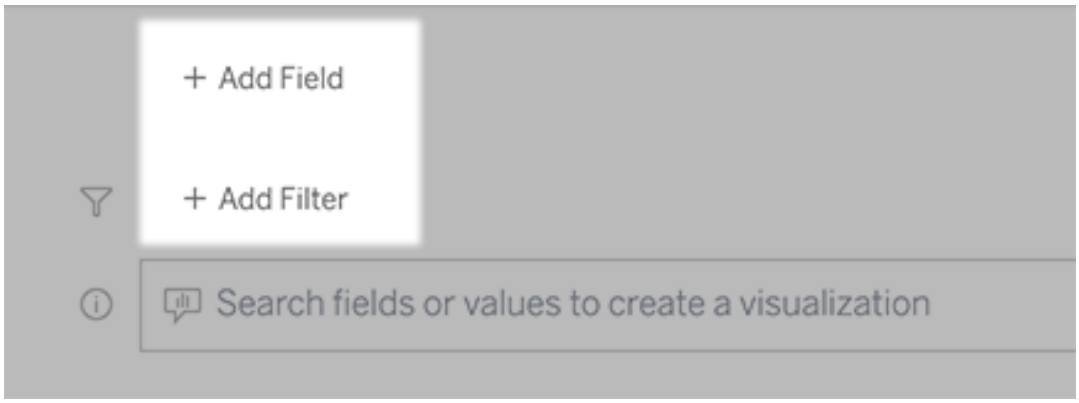
you get quick answers to common questions.

You can add these suggestions to your query by clicking them. As you add phrases to your query, the suggestions dynamically update with more relevant phrases. The view automatically builds with each selection.

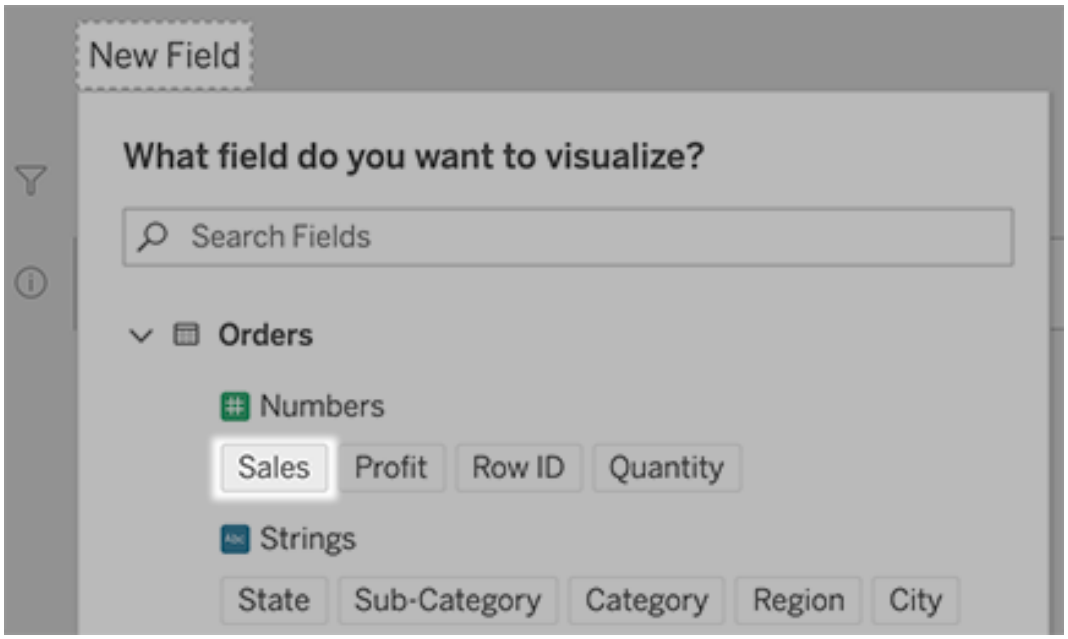


Build queries by adding fields and filters

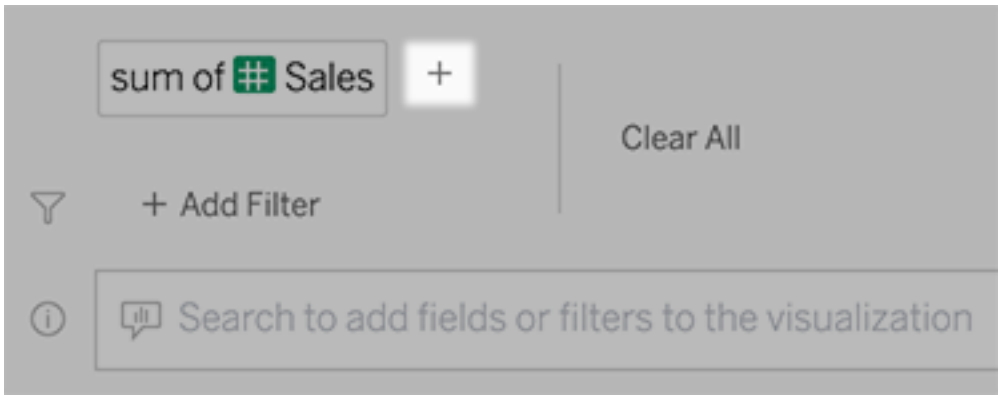
1. Click **Add Field** or **Add Filter**.



2. Click the desired field. (To narrow down a long list, first type in the **Search Fields** box.)

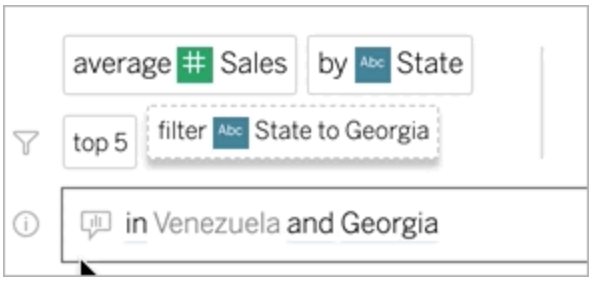


3. Set any sub-options, such as aggregation type for a numeric field, or grouping for string and date fields.
4. To add more fields or filters, click the plus sign.



See how elements of your query are applied

To see how elements of your query are applied, hover over them in the text box or the interpretation above it. Words that aren't used are grayed out, helping you rephrase your query in a way that’s clearer to Ask Data.



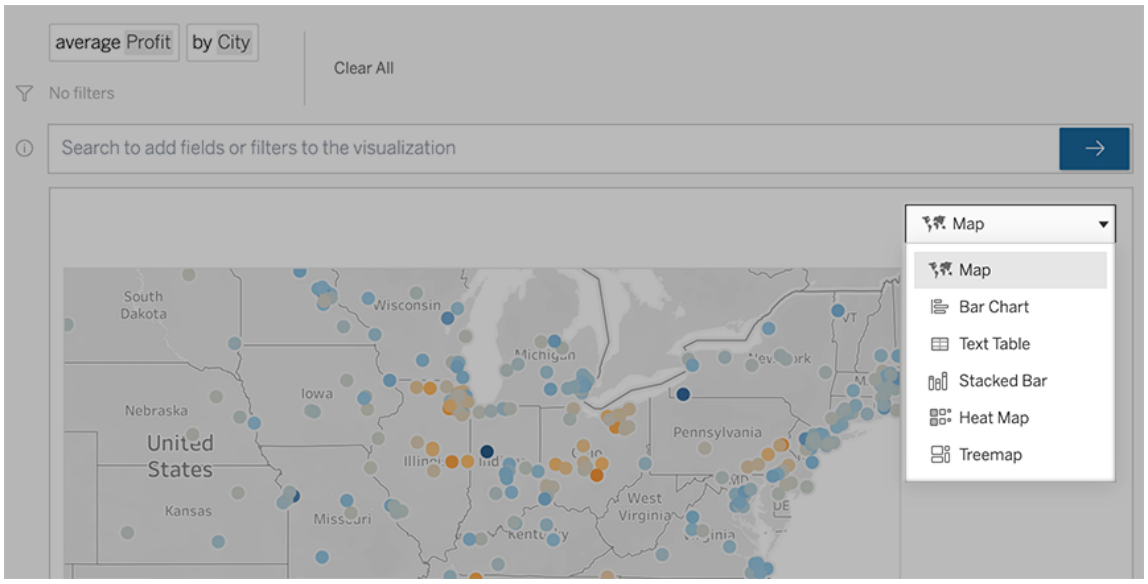
Rephrase your question

You can rephrase questions by clicking options, data fields, and filters in the user interface.

Change the viz type

If the default viz doesn't fully reveal your data, click the menu at upper right, and choose from these supported viz types:

- Bar Chart
- Gantt Bar
- Heat Map
- Histogram
- Line Chart
- Map
- Pie Chart
- Scatter
- Stacked Bar Chart
- Text Table
- Treemap

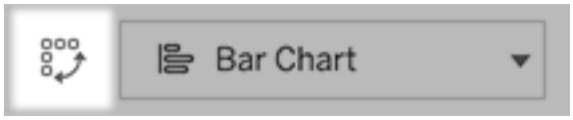


Note: To automatically create certain viz types, Ask Data sometimes adds fields such as "Number of Records" to your entries.

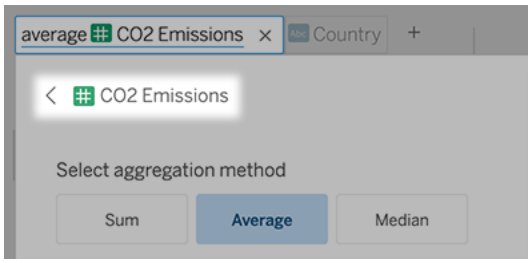
Change fields, filters, and displayed data

Ask Data gives you several ways to fine-tune how field values are displayed.

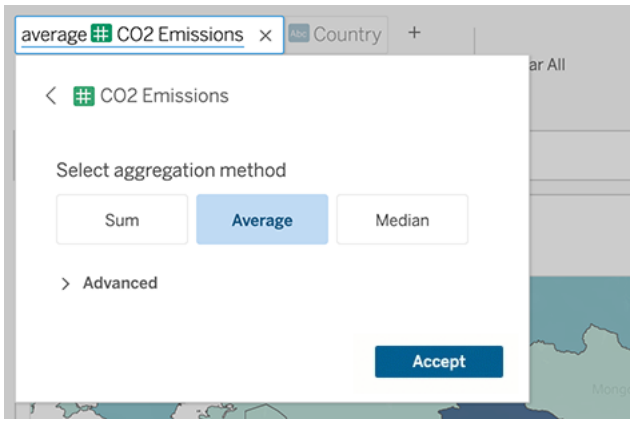
- To switch the fields used for the vertical and horizontal axes, click the Swap Axes button to the left of the viz selection menu:



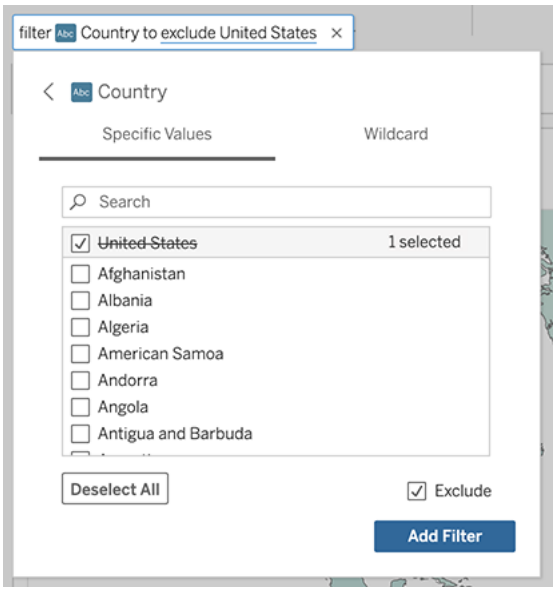
- To change a field, first click it in your query entry, and then click the field name below. (To change fields used in difference calculations, see Compare differences over time.)



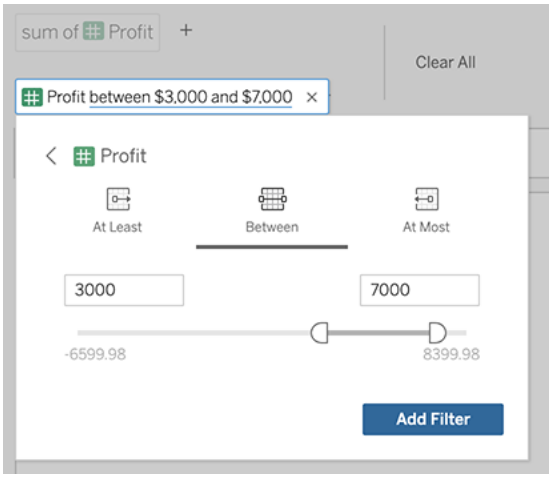
- To change a field's aggregation or grouping type (for example, from average to sum), click the field name in the text box, and then choose a different aggregation or grouping.



- For categorical filters, click values (for example, "exclude United States" in the example below) to change specific values or enter wildcard parameters.



- To adjust a numeric range, click words such as "high" or "cheap."



- To delete a field or filter, hover over it and click the **X**.

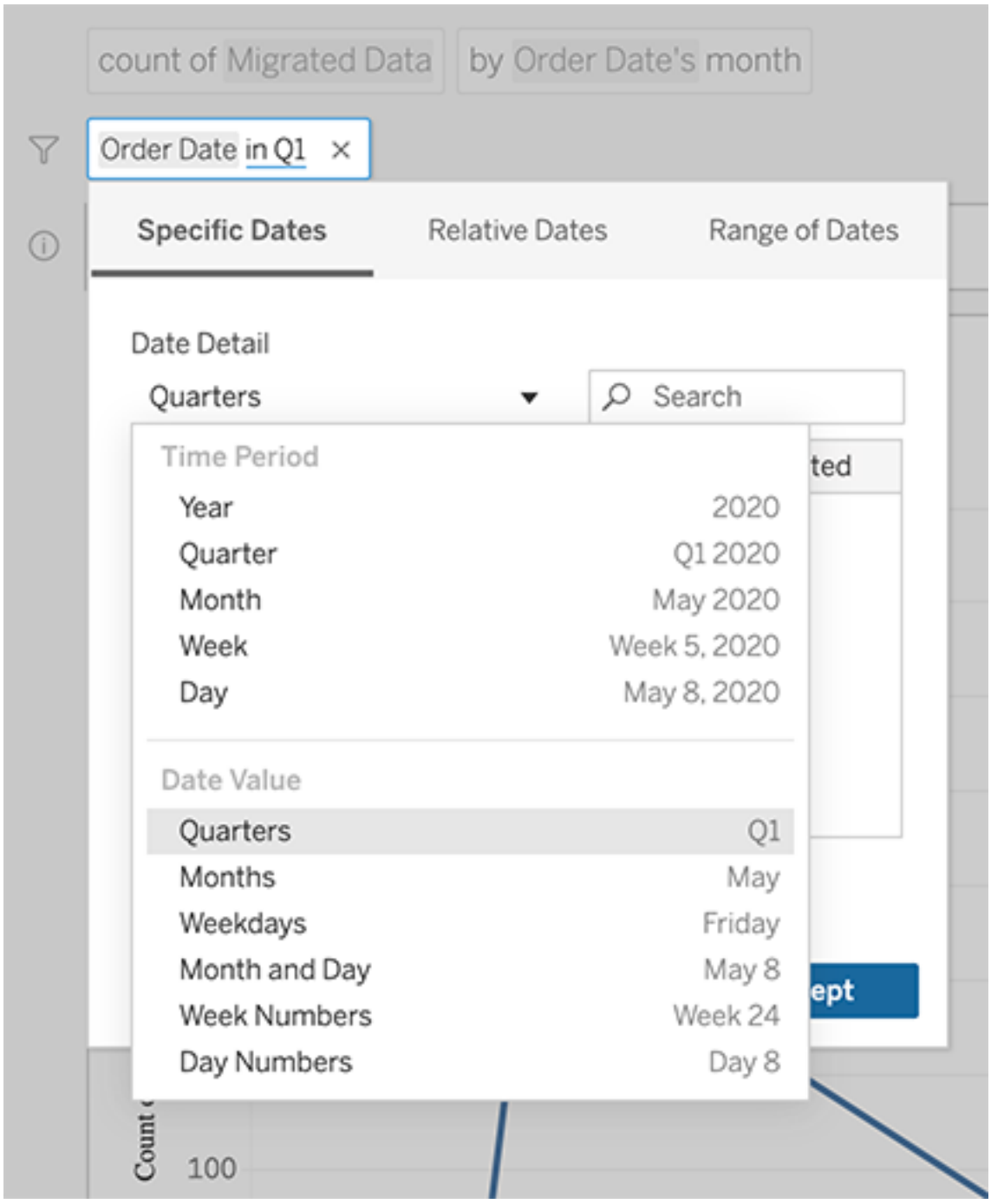
Adjust date filters

To adjust a date filter, click words such as "last" or "previous." Then click one of the following:

- **Specific Dates** to enter a specific time period or date value
- **Relative Dates** to show a date range relative to the present day
- **Range of Dates** to enter specific start and end points

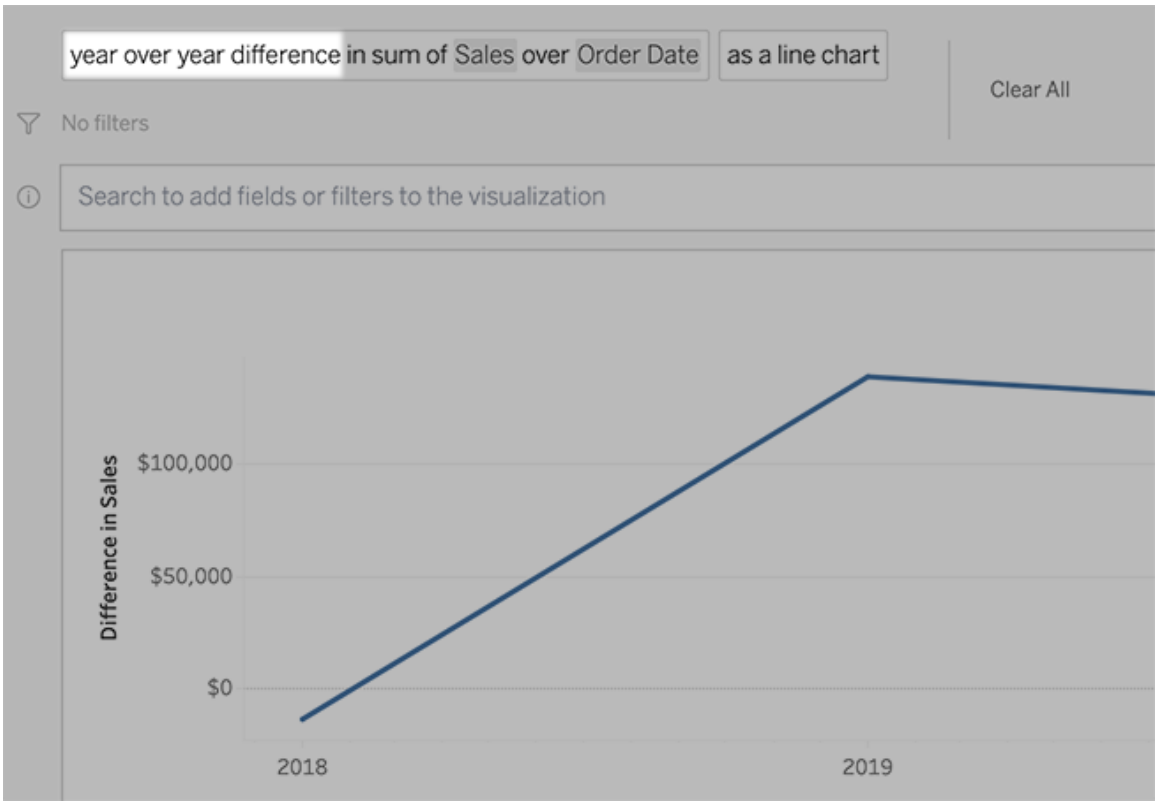
Specific Dates offers some unique options in the **Date Detail** menu:

- **Time Period** options show a single continuous date range
- **Date Value** options show ranges that can repeat in multiple time periods. For example, to see combined sales performance for Q1 across multiple years, under Date Value, you would choose Quarters.

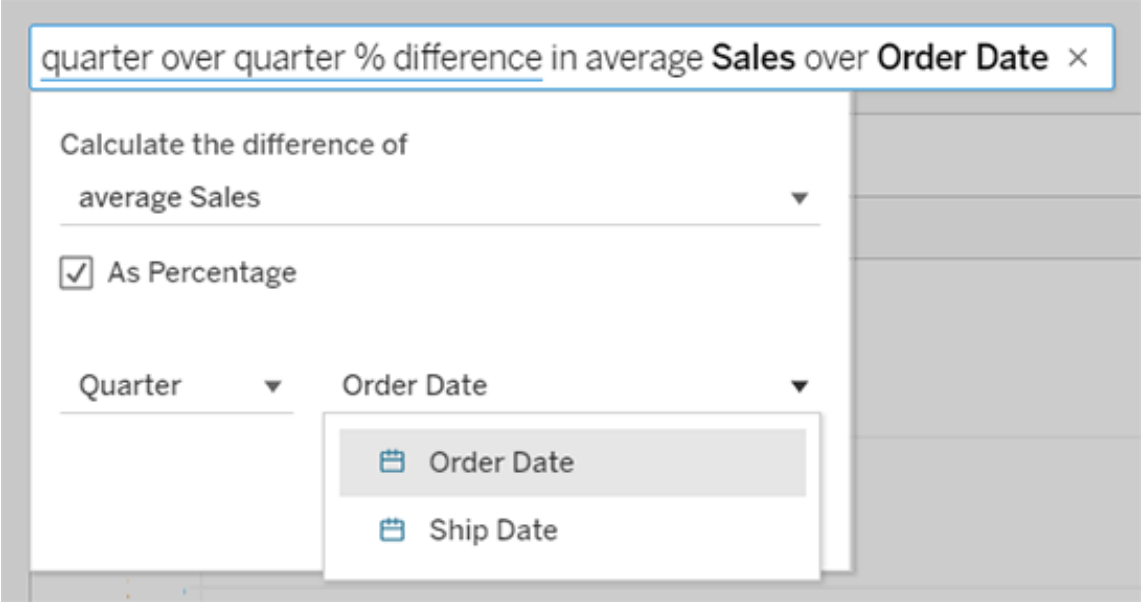

count of Migrated Data
by Order Date's month
Order Date in Q1
Specific Dates
Relative Dates
Range of Dates
Date Detail
Quarters
Search
Time Period
Year 2020
Quarter Q1 2020
Month May 2020
Week Week 5, 2020
Day May 8, 2020
Date Value
Quarters Q1
Months May
Weekdays Friday
Month and Day May 8
Week Numbers Week 24
Day Numbers Day 8
100

Compare differences over time

Ask Data lets you compare time periods with phrases such as "year over year" or "quarter over quarter." The results appear as difference or percent difference table calculations in workbooks you save from Ask Data.



In the text box, click a difference calculation to choose other fields, aggregation methods, and time periods.



Apply simple calculations

Ask Data supports simple calculations between two measures, which you can apply using these symbols:

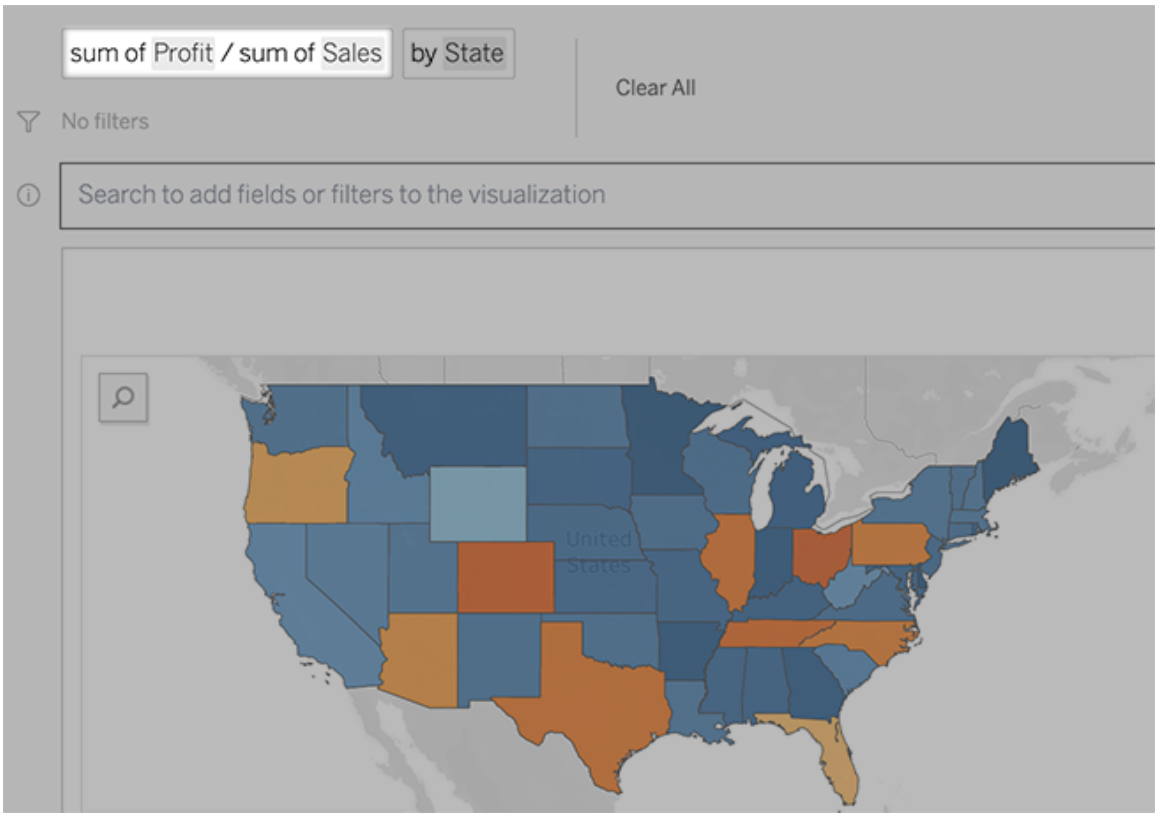
+ sums the measures

- produces the difference between them

***** multiplies

/ divides

In workbooks you save from Ask Data, these calculations don't become calculated fields but instead ad hoc calculations on the Columns, Rows, or Marks shelves.

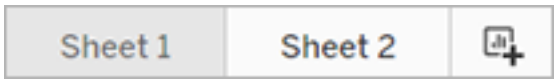


Add sheets with other vizzes

To quickly create multiple different vizzes from a lens, add sheets in Ask Data.

At the bottom of the web page, do any of the following:

- Click the **Add Sheet** icon to the right of named sheets.



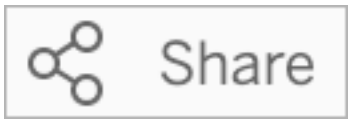
- Right-click a sheet name, and choose either **Duplicate** or **Delete**.

  (To rename sheets from Ask Data, you need to save them in a new workbook.)

Share Ask Data vizzes via email, Slack, or a link

You can quickly share Ask Data vizzes with anyone who has access to a lens.

1. In the upper right corner of the browser, click the Share icon.

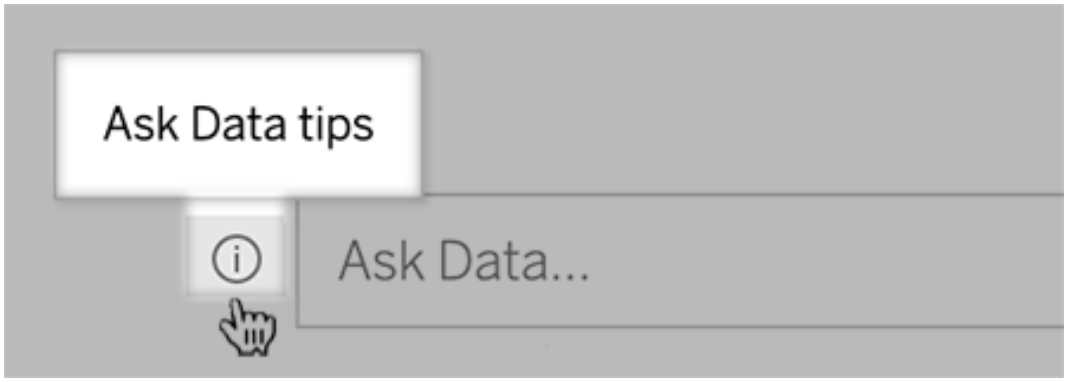


2. Do either of the following:
   - To share the viz via email or Slack, enter specific user names in the text box. (Email and Slack integration must previously be configured by your Tableau administrator.)
   - To copy a URL you can paste into custom emails and other messages, click **Copy Link**.

Send feedback to the lens owner

If you have questions about the structure of a lens or how best to use it with Ask Data, you can send feedback directly to the author. (This option is enabled by default, but lens authors may disable it.)

1. To the left of the query box for Ask Data, click the **Ask Data tips** icon.



2. At the bottom of the tips dialog, click **Contact the Lens Author**.

Tips for successful queries

As you structure questions for Ask Data, apply these tips to get better results.

- **Use keywords** — For example, instead of "I want to see all the countries that these airports are in, try "by airport and country."
- **Use exact wording for field names and values** — For example, if your lens includes Airport Code, Airport Name, and Airport Region fields, specify those by name.
- **See a ranked list** — Ask Data maps terms such as "best" and "worst" to Top 1 and Bottom 1, respectively. If you want to see broader rankings, use "high" and "low" instead. For example, enter "houses with low sale prices."
- **Query table calculations** — In query expressions for table calculation fields, note that you can't filter, limit, or include "year over year difference."
- **Surround unusually long values with quotation marks** — To analyze long field values that contain line returns, tabs, or more than ten words, surround them with quotation marks. To improve performance, Ask Data doesn't index fields of that length, or anything beyond the first 200,000 unique field values.

Create Lenses that Focus Ask Data for Specific Audiences

Important changes for Ask Data and Metrics

Tableau's Ask Data and Metrics features were retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2. With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see *How Tableau AI and Tableau Pulse are reimagining the data experience.*

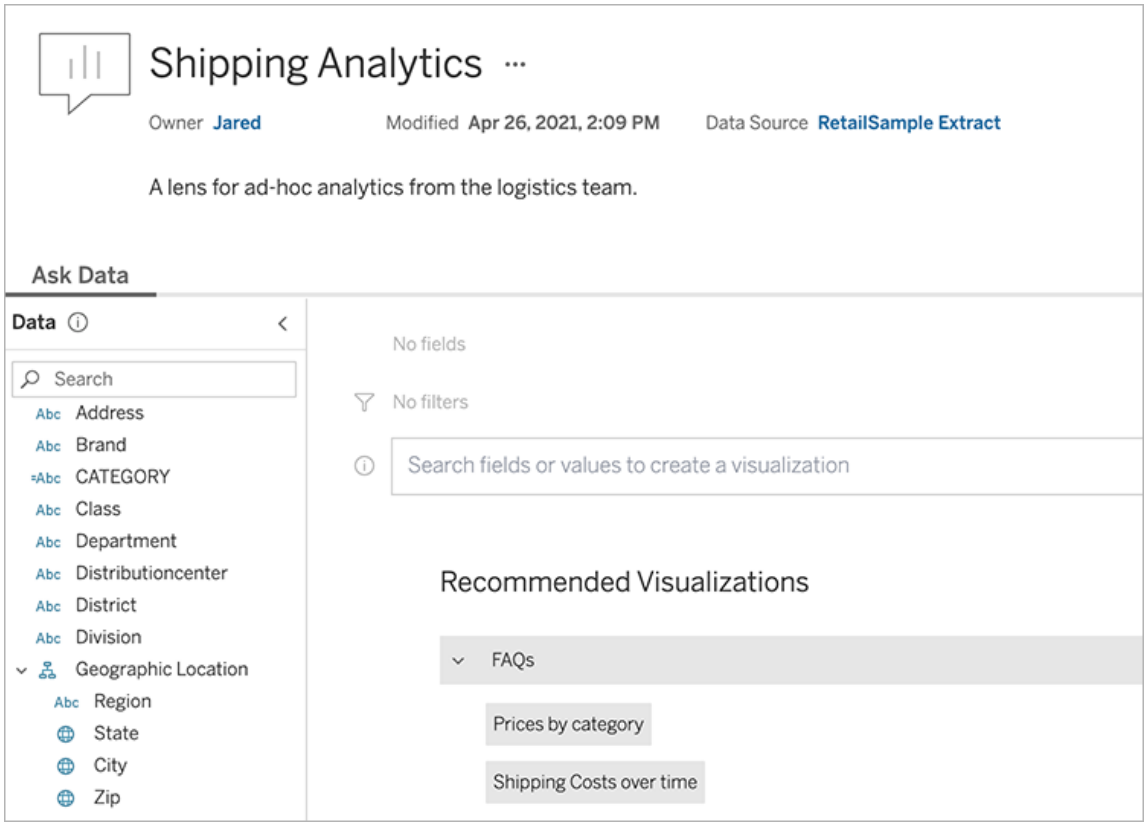
Most people don’t need information from an entire data source but instead want data visualizations relevant to their job function, like sales, marketing, or support. To optimize Ask Data for different audiences like these, Tableau authors create separate Ask Data *lenses*, which query a selected subset of fields. For the selected fields, authors can specify synonyms for field names and values, reflecting terms the lens audience uses in common language (for example, "SF" for "San Francisco"). Lens authors then customize the recommended visu-

alizations that appear below the Ask Data query box, which provide answers to users with a single click.

Note: Ask Data lenses can be created only for data sources published separately to a Tableau site. Lenses can't be created for data sources embedded in workbooks or those with a virtual connection.

Create or configure a lens page on your Tableau site

On your Tableau site, each lens has a separate page where users can query Ask Data and authors can configure lens fields, synonyms, and suggested questions.

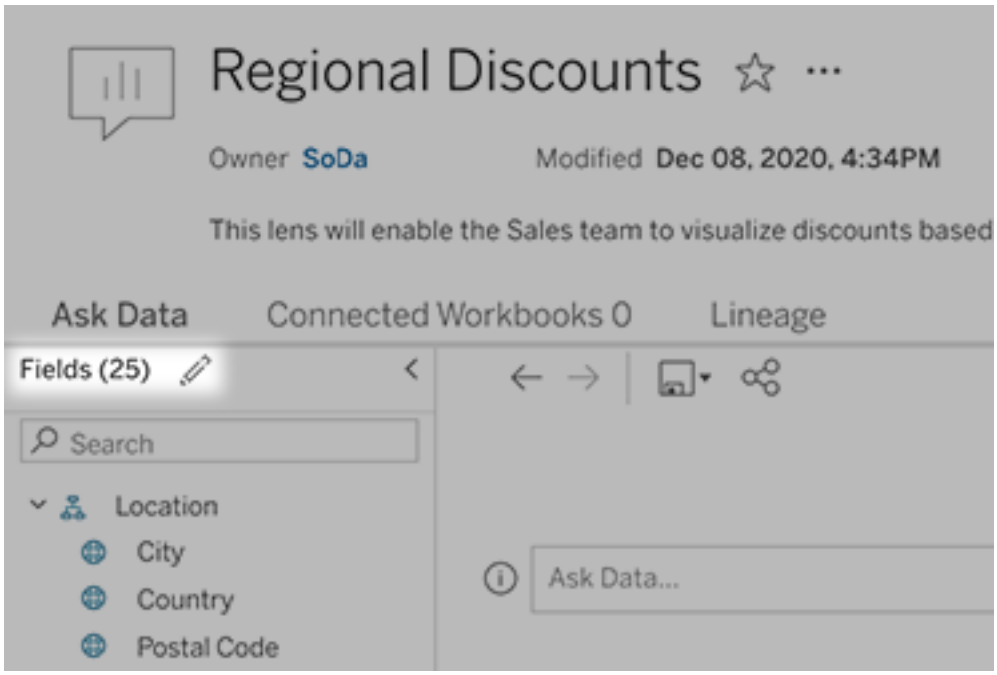


A lens page on a Tableau site

1. To create a lens page on your Tableau site, go to a data source page, and choose **New** > **Ask Data Lens**.

   To configure an existing lens, go to the lens page on your site. (From an Ask Data object in a dashboard, you can click the pop-up menu in the upper corner and choose **Go to Lens Page**.)

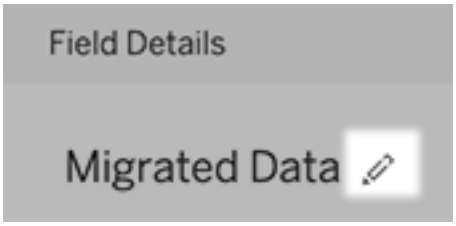
2. If you're creating a new lens, enter a name, description, and project location, and then click **Publish Lens**.
3. At the top of the Fields pane at left, click the pencil icon. Then select the relevant fields for lens users, and click **Save**.



4. At left, hover over individual tables or fields, and click the pencil icon:

Then do any of the following:

- Provide a more representative name by clicking the pencil icon to the right.



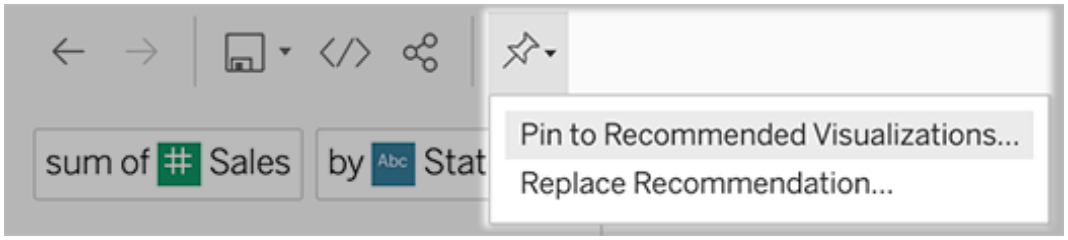
- Add common synonyms for field names and values that lens users may enter in their queries.
- Edit descriptions that appear when users hover over fields.

Change the list of recommended visualizations

To address common queries from lens users, you can customize the recommended visualizations that appear below the query box.

Add or replace a recommended visualization

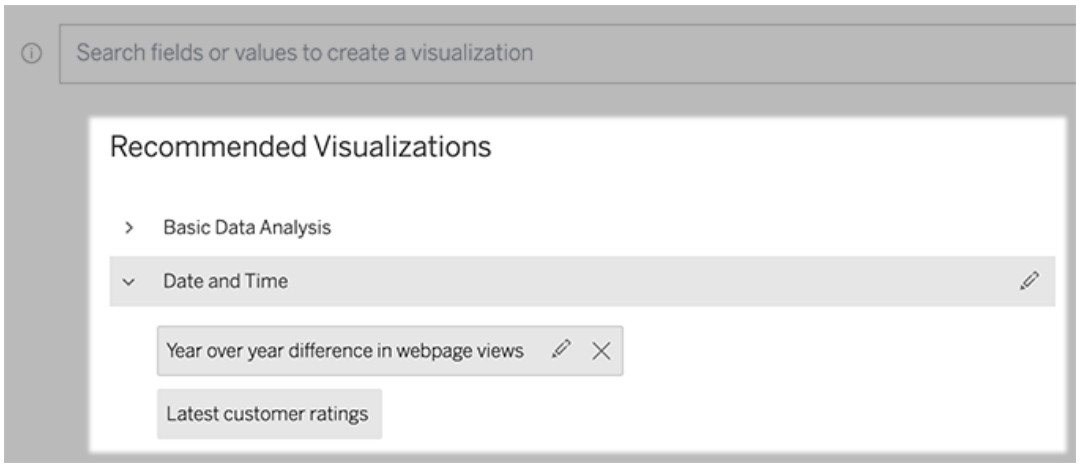
1. Enter a query into the text box, and press Enter or Return.
2. After the visualization appears, from the pin icon in the toolbar, choose either **Pin to Recommended Visualizations** or **Replace Recommendation**.



3. For a new recommendation, enter a name, and choose the section in which you want it to appear. For a replacement recommendation, choose the existing one you want to overwrite.

Edit section titles and recommendation names, or delete recommendations

- To edit a section title, click the pencil icon to the right of the title.
- To change the name of a recommendation, hover over it and click the pencil icon. To delete a recommendation, click the X.

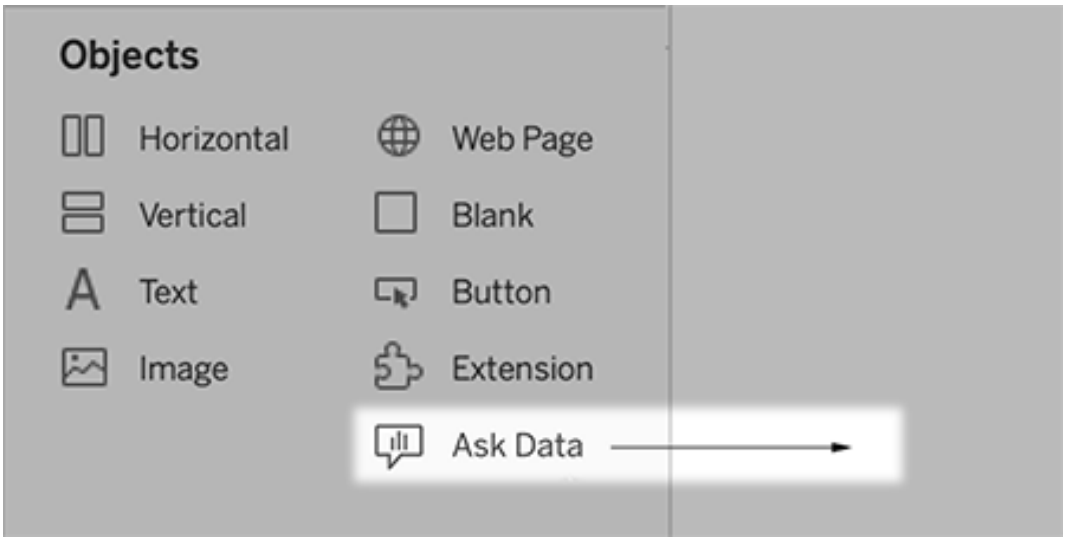


Add an Ask Data lens to a dashboard

On a dashboard, you can add an Ask Data object that lets users query a published data source via a lens on your Tableau site.

1. While editing a dashboard in Tableau Cloud or Tableau Server, drag the Ask Data object to the canvas.

> **Note:** In Tableau Desktop, you can also drag an Ask Data object to the canvas for placement purposes. But to select a lens, you will need publish to Tableau Cloud or Tableau Server and edit the object there.

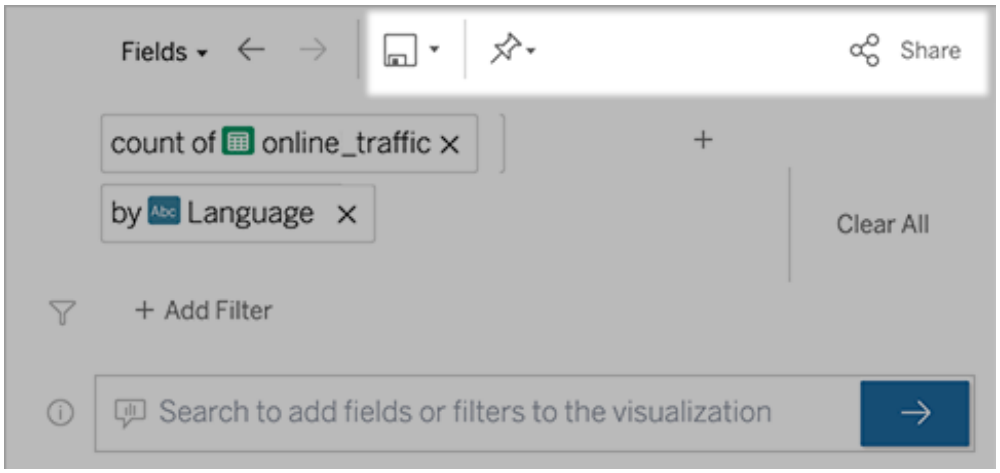
2. 

3. Select a published data source previously connected to the workbook.
4. To use an existing lens, select it, and click **Use Lens**.

   Or, to create a new lens, do one of the following:

   - If there are no lenses for the data source, click **Go to Data Source Page**.
   - If lenses already exist, click the data source name at the bottom of the dialog.
5. (New lenses only) Complete the steps in Create or configure a lens page on your Tableau site.
6. Under **Toolbar Options for Lens Users**, select the buttons you want available to users.
   - **Add Visualization to Pins** lets users add to the Recommended Visualizations list, which appears just below the query box.
   - **Publish as Workbook** lets users save visualizations as workbook sheets to their Tableau site.
   - **Share Visualization** lets users share via email, Slack, or a link.

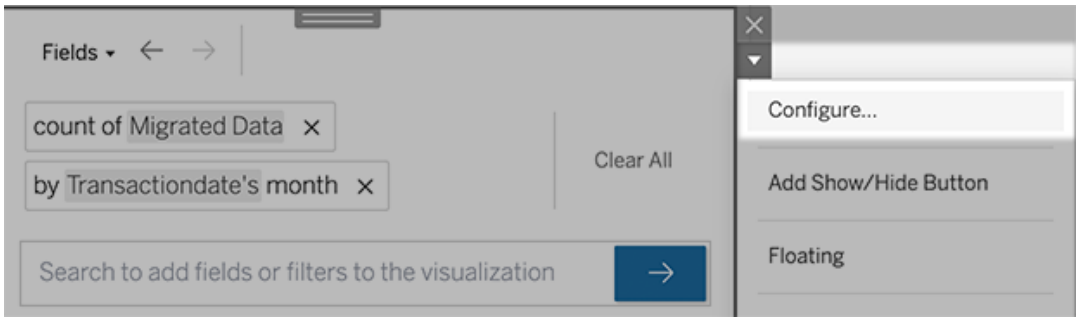
     In the lens, the publish (save icon), pin, and share options appear in the upper right corner:



7. (New lenses only) After you finish creating the lens, return to the Lens object in your dashboard, and click **Refresh**. Then select the new lens, and click **Use Lens**.

Apply a different lens to an Ask Data dashboard object

1. From the pop-up menu at the top of the object, choose **Configure**.



2. Go to Add an Ask Data lens to a dashboard, and repeat steps 2 onward.

Change a lens name, description, or project location

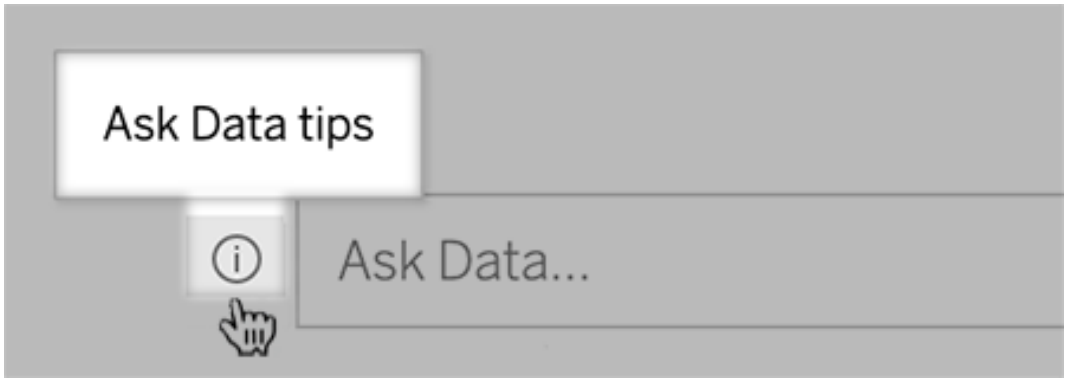
1. Navigate to the lens page on Tableau Cloud or Tableau Server.
2. To the right of the lens name at the top of the page, click the three dots (...), and choose **Edit Workbook**.
3. Click **Edit Lens Details**.

See how people use Ask Data with a lens

For data source owners and lens authors, Ask Data provides a dashboard that reveals the most popular queries and fields, the number of visualization results that users clicked, and other helpful information. Filters let you narrow data down to specific users and time ranges. These stats help you further optimize a lens to increase the success of your users.

Note: If you use Tableau Server, you can access this data in the Tableau Server Repository to create custom dashboards.

1. In Tableau Server or Tableau Cloud, navigate to a lens page.
2. To the left of the Ask Data text box, click the "Ask Data tips" icon.



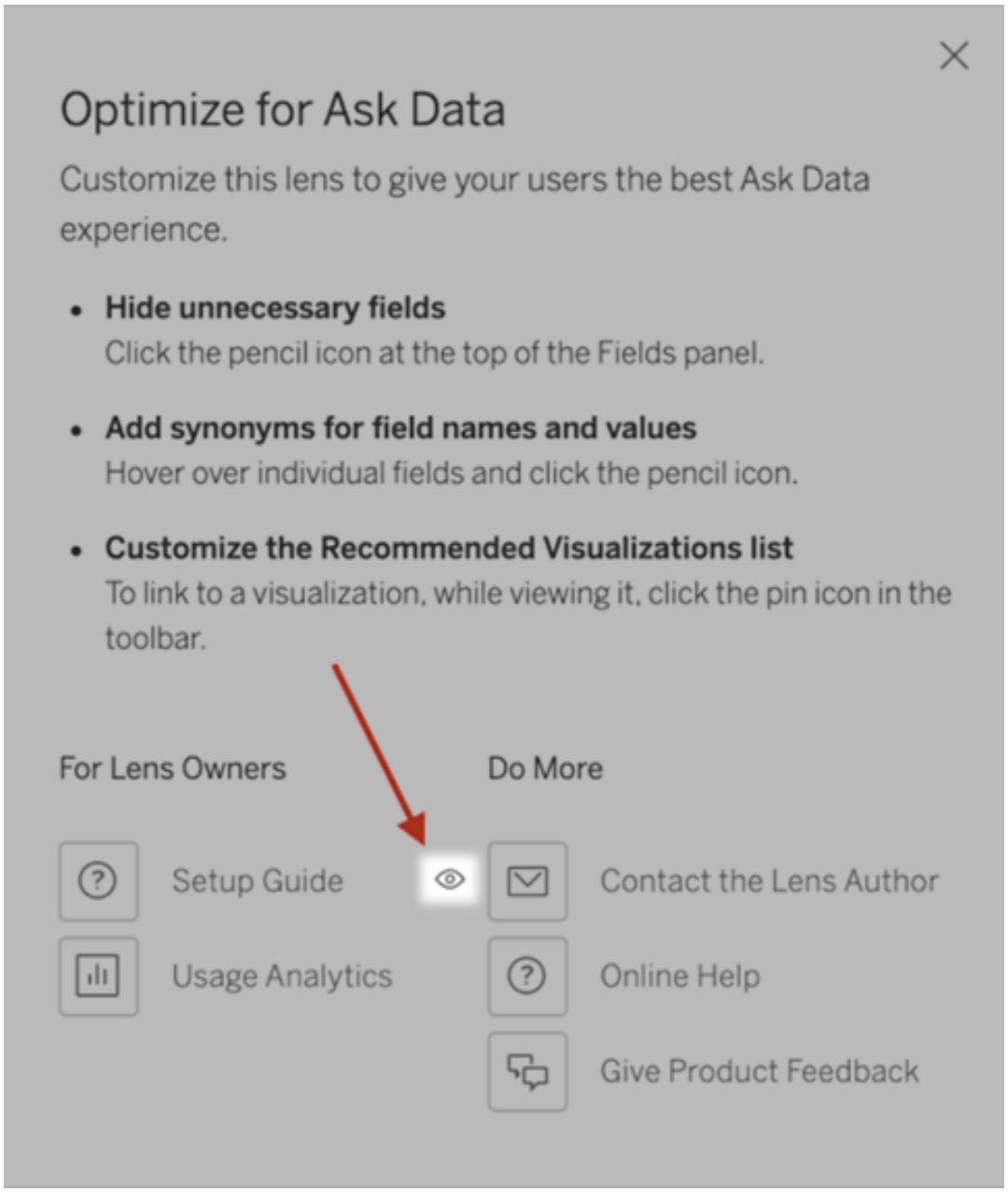
3. In the lower-left corner of the tips dialog, click **Usage Analytics**.

Let users email you questions about a lens

As a lens owner, you can allow users to email you with questions about data structure, expected results, and more. This option is on by default, but you can turn it off using the steps below.

1. In Tableau Server or Tableau Cloud, navigate to a lens page.
2. To the left of the Ask Data text box, click the "i" shown above in See how people use Ask Data with a lens.
3. At the bottom of the tips dialog, click the eye icon next to "Contact the Lens Author" to

enable or disable feedback.



Permissions for publishing and viewing lenses

For Ask Data objects in dashboards, no change to permissions should be required: by default, existing workbook authors can create lenses, and existing dashboard audiences can view

them. But for reference, here's a detailed outline of required lens permissions for both dashboards and direct access via a data source page.

To create and publish a lens, a user needs:

- The Creator or Explorer user role
- Lens Creation permission for the data source (inherited by default from the Connect permission)
- Write permission for the parent project to which the lens is published

To access and interact with a published lens, a user needs:

- The Viewer role or above
- Connect permission for the data source
- View permission for the lens

Note: By default, lens permissions like View reflect a project's permissions for workbooks. If Tableau administrators want to change default lens permissions, they can do so either individually for each project, or in bulk using the permissions API.

Disable or Enable Ask Data for a Site

Important changes for Ask Data and Metrics

Tableau's Ask Data and Metrics features were retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2. With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see *How Tableau AI and Tableau Pulse are reimagining the data experience.*

Ask Data is enabled for sites by default, but Tableau administrators may disable it.

1. Go to the **General** site settings.
2. (Tableau Server only) In the **Web Authoring** section, select **Let users edit workbooks in their browser**.
3. In the **Availability of Ask Data** section, choose from these options:

- **Enabled** enables creation of Ask Data lenses for all published data sources.
- **Disabled** hides Ask Data throughout the site, while preserving information about previously created lenses so they can be restored if Ask Data is re-enabled.

Note: Beginning with version 2019.4.5, Tableau Server administrators can configure whether or not Ask Data is enabled by default.

Optimize Data for Ask Data

Important changes for Ask Data and Metrics

Tableau's Ask Data and Metrics features were retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2. With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see *How Tableau AI and Tableau Pulse are reimagining the data experience.*

If you manage and publish data sources, here are some tips to help make users of Ask Data more successful. By spending a little extra time on this process, you'll open up data analysis to a wider range of people at your organization, helping them independently answer questions and gain deeper insights.

Optimize data in Ask Data

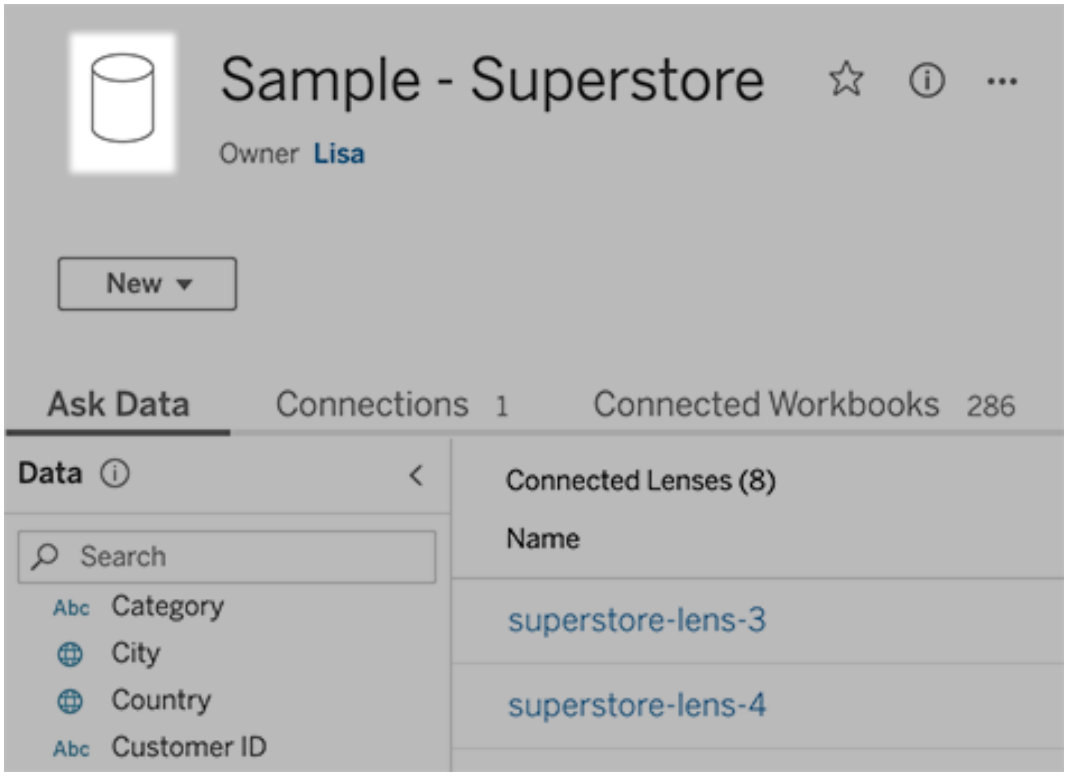
In the Data pane on the left of the Ask Data interface, data source owners can add synonyms for fields and exclude irrelevant values.

Changing settings at the data source or lens level

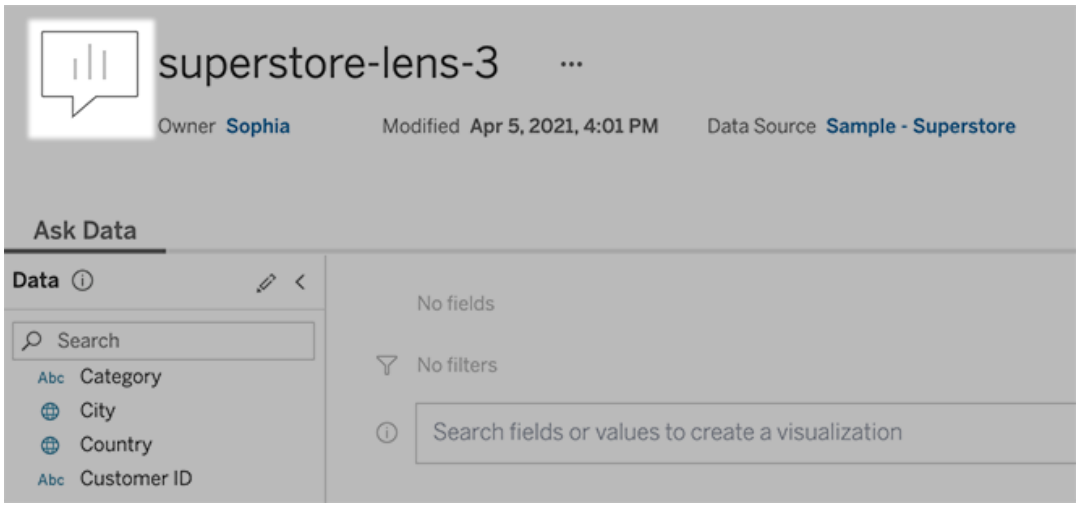
When changing settings in the Data pane for Ask Data, pay close attention to whether you're at the data source or lens level. (For more information, see Create Lenses that Focus Ask Data for Specific Audiences.)

- At the data source level, you'll see the cylindrical data source icon in the upper left corner. Here, changes you make in the Data pane will apply by default to all subsequently created lenses.

  Note: For extracts, two cylinders will appear.



- For an individual lens, you'll see the quotation icon in the upper left corner. Here, changes you make in the Data pane will apply to this lens alone.



Add synonyms for field names and values

People may not use the same terminology found in your data source, so data source owners and Tableau administrators can add synonyms for specific data field names and values. Synonyms you enter are available throughout your organization, making data analysis quicker and easier for everyone.

Exclude values of specific fields from search results

To improve the usability of search results in Ask Data, you can exclude the values of specific fields from indexing. Though Ask Data doesn't add non-indexed values to search results, the values still appear in visualization results when relevant. For example, if you don't index values from a "Product" field because they add unnecessary detail to search results, Ask Data can still display values such as "iPhone 12" in resulting data visualizations. And users can manually add non-indexed values to queries by surrounding them with quotation marks (for example, "Sales for Product containing "iPhone 12"").

Note: This field-level setting is ignored if the value indexing setting for the data source is set to Disabled. Field names and related synonyms are always indexed.

1. Go to the Ask Data tab for a data source or individual lens.
2. Hover over a data source field at left, and click the **Edit Field Details** icon (the pencil).

3. Deselect **Index field values.**

   Either click the text box that appears to reindex the data source now, or let it reindex based on its regular indexing schedule.

Optimize data sources

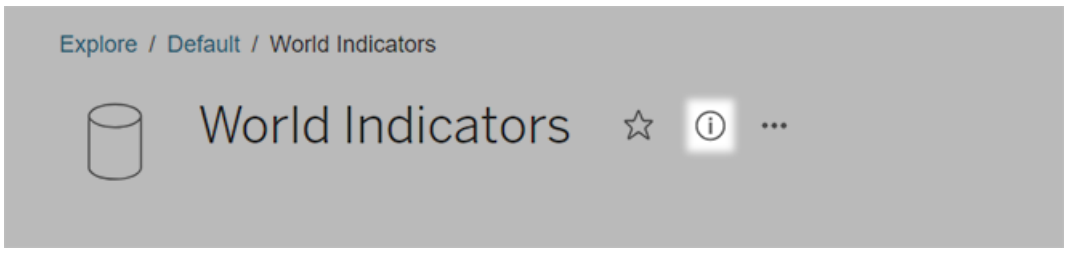
To create the best experience for Ask Data users, optimize the original data source.

> **Note:** Ask Data doesn't support multidimensional cube data sources, or non-relational data sources like Google Analytics, or data sources with a virtual connection.

Optimize indexing for Ask Data

Data source owners can change how often field values are indexed for Ask Data, optimizing system performance.

1. At the top of a data source page, click the Details icon:



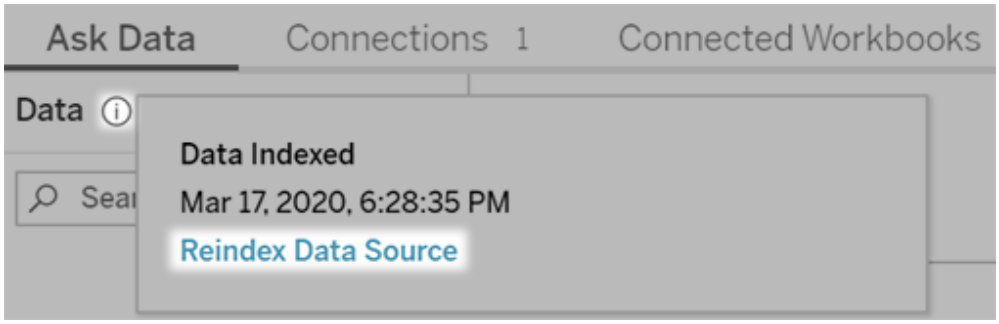
2. In the Ask Data section, click **Edit**.

3. Choose an indexing option for field values:

   - **Automatic** checks for changes every 24 hours and analyzes the data source if it is live, has had an extract refreshed, or has been republished. Choose this option for a data source frequently used with Ask Data, so it will be ready before users query it.

   - **Manual** analyzes the data source only when Tableau creators manually trigger indexing on the data source page. Choose this option if the data source changes frequently but users query it with Ask Data only occasionally.

     To trigger manual indexing, go to the data source page, click the circled “i” in the Data pane at left, and then click **Reindex Data Source**.

   - **Disabled** analyzes only field names, not values.

4. Click **Save**

Use data extracts for faster performance

For improved performance and support for large data sets, use Ask Data with published extracts rather than live data sources. For more information, see Create an extract.

Ensure that users can access the data source

To use Ask Data, users must have permission to connect to the individual data source. If a data source has row-level permissions, those permissions also apply to Ask Data, which won’t recognize secure values or make related statistical recommendations.

Be aware of unsupported data source features

Ask Data supports all Tableau data source features except the following. If your data source contains these, Ask Data users won't be able to query related fields.

- Sets
- Combined fields
- Parameters

Anticipate user questions

Anticipate the kinds of questions your users will ask, and then optimize your data source for those questions using these techniques:

- Clean and shape data in Tableau Prep or a similar tool.
- Join data to include all fields users may have questions about in one table, improving performance.
- Add calculated fields that answer common user questions.
- Create bins with appropriate sizes for quantitative variables that users are likely to want to see as a histogram or another binned form.

Simplify the data

To make data easier to understand by both users and Ask Data, simplify the data source as much as possible during the data prep process.

1. Remove any unnecessary fields to improve performance.
2. Give each field a unique and meaningful name.
    - For example, if there are five field names that start with “Sales …”, better distinguish them so Ask Data can properly interpret the term “sales”.
    - Rename “Number of records” to something more meaningful. For example, use “Number of earthquakes” in a data source where each record is an earthquake.
    - Avoid field names that are numbers, dates, or boolean (“true” or “false”) values.
    - Avoid names which resemble analytical expressions such as “Sales in 2015” or “Average Products Sold”.
3. Create meaningful aliases for field values, reflecting terms people would use in conversation.

Set appropriate field defaults

To help Ask Data analyze data correctly, ensure that default field settings reflect the content of each field.

- Set data types for text, time, date, geographic, and other values.
- Assign the proper data role: dimension or measure, continuous or discrete.
- For each measure, assign appropriate default settings in Tableau Desktop, such as color, sort order, number format (percentage, currency, etc.), and aggregation function. For example, SUM may be appropriate for “Sales”, but AVERAGE might be a better default for “Test Score”.

> **Tip:** It's particularly helpful to set a default comment for each field, because these comments appear as informative descriptions when users hover over fields in Ask Data.

Create hierarchies for geographic and categorical fields

For time data, Tableau automatically creates hierarchies, which let users quickly drill up and down in vizzes (for example, from day to week to month). For geographic and categorical data, however, we recommend that you create custom hierarchies to help Ask Data produce visualizations that reflect the relationships between fields. Be aware that Ask Data won't show the hierarchies in the data pane.

Ask Data doesn't index hierarchy names, only names of fields within hierarchies. For example, if a geographic hierarchy named “Location” contains “Country” and “City” fields, users should enter “Country” and “City” in their questions for Ask Data.

Create a Tableau Data Story (English Only)

Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of

changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

If you've ever written an executive summary of your Tableau dashboard, then you know it can be time-consuming. It takes time to choose which insights to share, and you have to rewrite your summaries each time the data is updated. Tableau Data Stories automatically generates narrative insights within your dashboard, saving time and surfacing relevant insights. As you explore the vizzes in your dashboard, the stories written by Data Stories adjust, allowing you to dive deeper into data and identify key insights faster.

From where you're already working in Tableau, you can quickly add the **Data Story** object to your dashboard. And you can customize the terms and metrics used in your story, so Data Stories speaks the language used by your business.

Today, you can write and view a Tableau Data Story anywhere you use Tableau. After you create your story, you can also view your Data Story in Tableau Mobile. However, Data Stories aren't included if you export your dashboard, for example to a PDF.

Understand how Data Stories handles data

To write Data Stories, Tableau uses a service hosted in your Tableau Cloud or Tableau Server environment. When you Add a Tableau Data Story to a Dashboard or view a Data Story from a dashboard, Tableau sends associated worksheet data to the environment that you're logged in to (i.e., your Tableau Cloud site or your Tableau Server instance), using the security standards outlined in Security in the Cloud and Security in Tableau Server. Data Stories can be written and viewed from anywhere you use Tableau.

Learn about how Data Stories are written

Tableau Data Stories is powered by rules-based templated natural language generation (NLG). Data Stories performs automated analytics to determine relevant and accurate facts about the underlying data–from basic calculations to more advanced statistics. To write a story, Data Stories uses a library of predefined language templates to synthesize these facts into natural language insights. Data Stories processes these templates at run-time, using the most up-to-date summary data from the Tableau worksheet it is connected to. You can

leverage the custom language feature to generate your own language templates, add functions, and define business rules, helping you build a more relevant and contextual Data Story.

> **Note:** Data Stories doesn't use generative AI, large language models (LLMs), or machine learning to write insights and stories.

Manage Data Stories for your site

Tableau administrators can choose whether Tableau Data Stories are available for their site. Data Stories are turned on by default.

1. Sign in to your Tableau site.
2. From the left pane, choose **Settings**.
3. From the **General** tab, scroll to the **Availability of Data Stories** section.
4. Choose whether you want to **Turn on** or **Turn off** Data Stories.

> **Note:** If Data Stories are turned off, then turning the feature back on restores Data Stories that were already in dashboards.

Add a Tableau Data Story to a Dashboard

Important changes for Tableau Data Stories

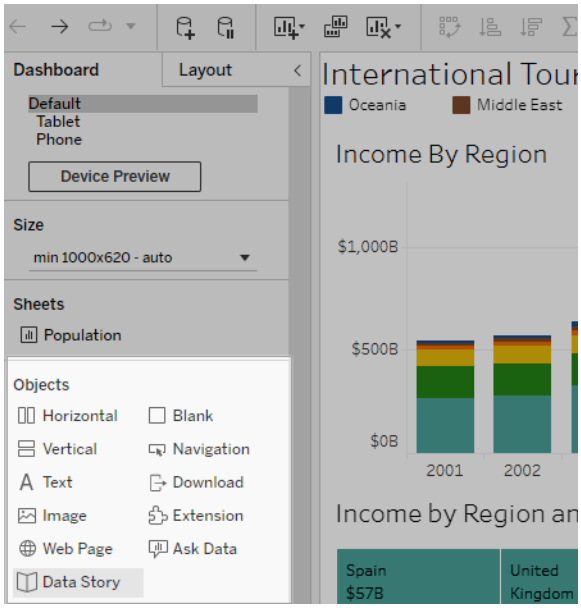
Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

After you Create a Dashboard, you can add the **Data Story** object to your dashboard to display insights about your viz that are written in natural language. Today, Tableau Data Stories are written in English only and are available in Tableau Cloud, Tableau Server (version 2023.1 and later), and Tableau Desktop. There is no data size limit when creating **Data Stories**.

However, story generation times out after 45 seconds if it's trying to analyze a lot of data. We recommend using **Data Stories** with visualizations that have 1,000 or fewer data points.

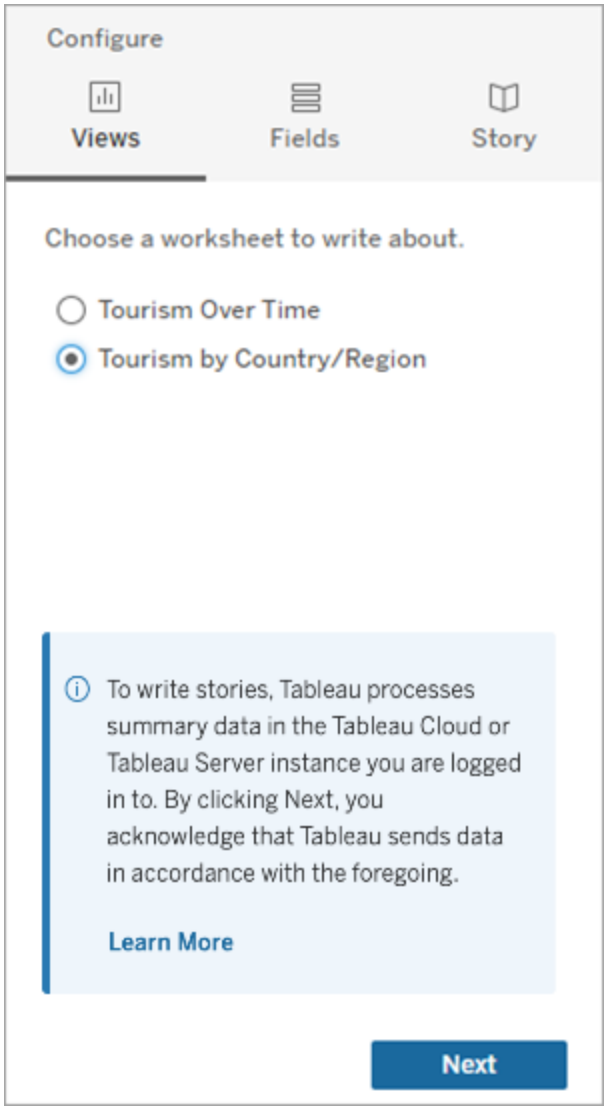
> **Note:** Tableau Data Stories opens in a pop-up window, so be sure to allow pop-ups. If you're using full screen mode, then Data Stories can open in a new tab.

1. Drag the **Data Story** object into your Tableau dashboard. If you haven't already, add a sheet to your dashboard to use Tableau Data Stories.

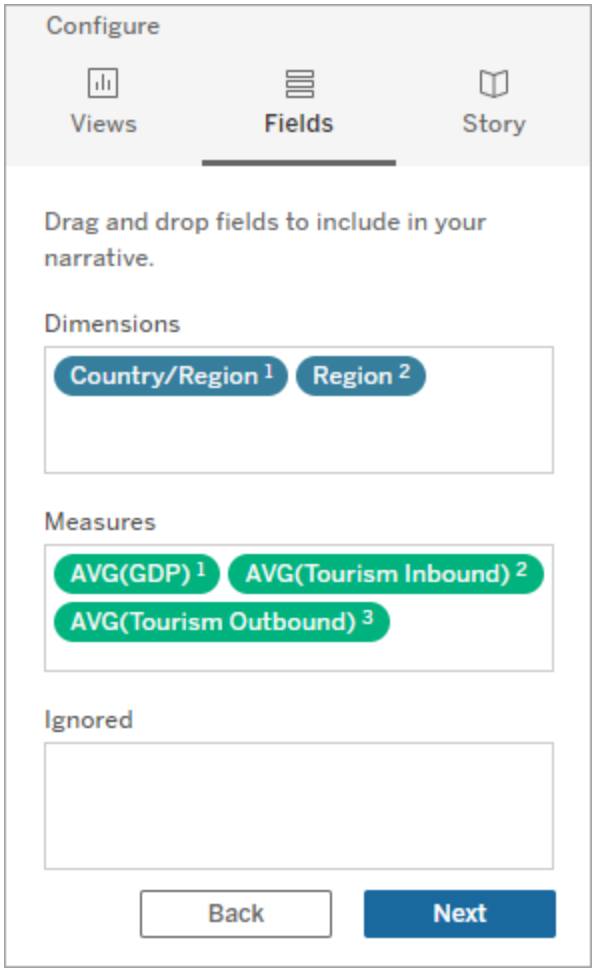


2. In the Data Story dialog box, configure your story by first choosing the worksheet to write about. When you click **Next**, Tableau sends all associated workbook data to the

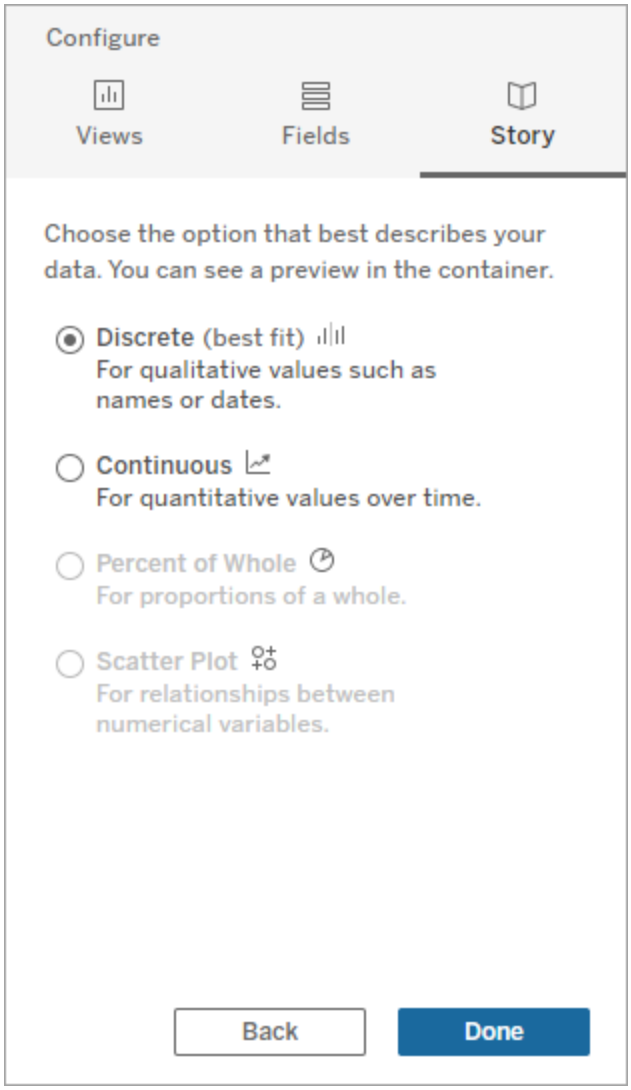
Tableau Cloud or Tableau Server instance that you are logged in to.



3. Choose the dimensions and measures to include in your story.



4. Choose the type of story that best describes your data:



Discrete is best for qualitative values such as names or dates, such as in bar or column charts.
Continuous is best for quantitative values over time, such as a plotted in a line chart.
Percent of Whole is best for proportions of a whole, such as a pie chart.
Scatter Plot is best for relationships between numerical values, such as a scatter plot chart.

5. Click **Done**.

To filter your Data Story by clicking different sections on your visualization, open the menu on your visualization and click **Use as filter**.

After your story is generated, click **Settings** at the top of your **Data Story** object for a guided experience that helps you personalize and contextualize your story. For more information, see Configure Settings for a Tableau Data Story.

Note: If you experience a discrepancy in your Tableau Data Story (for example, if the numbers in your story are different than in your visualization), it may be caused by way your visualization is set up. Try creating a new visualization on a different sheet, and then add a new Data Story with the Use a hidden sheet technique to uncover the underlying issue.

Choose the Right Story Type for Your Tableau Data Story

Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

When you Add a Tableau Data Story to a Dashboard, it’s important to choose the right type of story for your data. Do you want your story about trends over time? Or do you want your story about two values that you’re comparing? To help you tell the right story, this topic describes the different types of stories, including an example of each story type.

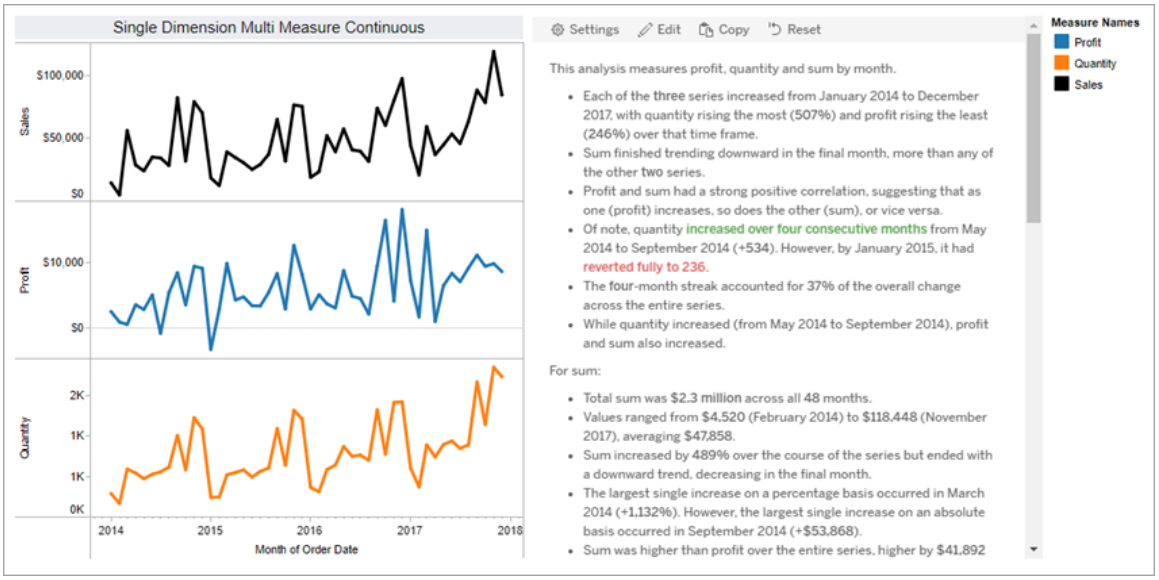
Continuous

Continuous stories are best for analyzing trends or progress over time.

When you create a continuous story, it includes content for performance, segments, volatility, and trend lines. The story also includes contribution analysis and correlation for stories that use more than one dimension. To use a continuous story, your worksheet must have:

- 1 dimension that has between 1-10 measures
- 2 dimensions and up to 3 measures

The following example is a continuous story for a line chart that has a single dimension and multiple measures:



Discrete

Discrete stories are best for comparing values and understanding the distribution of data in each value. When you create a discrete story, the story includes content about the distribution and groupings or clusters across the data. And the story includes contribution analysis for worksheets that use multiple dimensions.

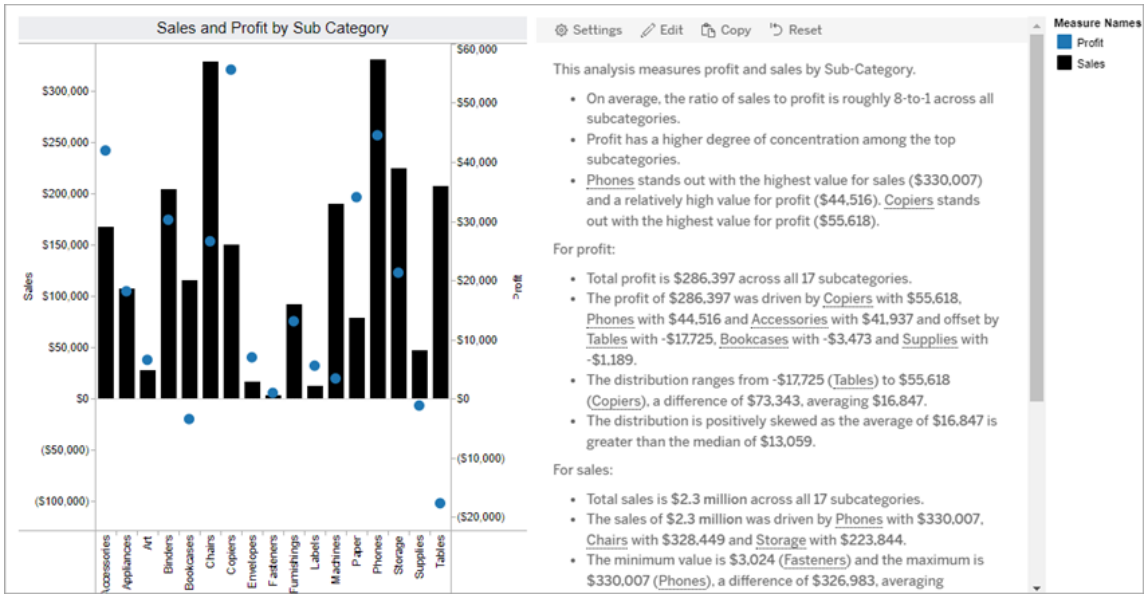
Consider using a discrete story when you want to:

- Understand drivers of your key performance indicators (KPIs) in sales reports.
- Identify and understand outliers quickly during data discovery.
- Identify trends that aren't easily observable in the visual when performing an audit.
- Uncover complex utilization insights instantly for geographic analysis.
- Identify and call out key relationships, for example, between sales and profit.

To use a discrete story, your worksheet must have:

- 1 dimension that has between 1-10 measures
- 2 dimensions and up to 3 measures

The following example is a discrete story for a bar chart that has a single dimension and two measures:

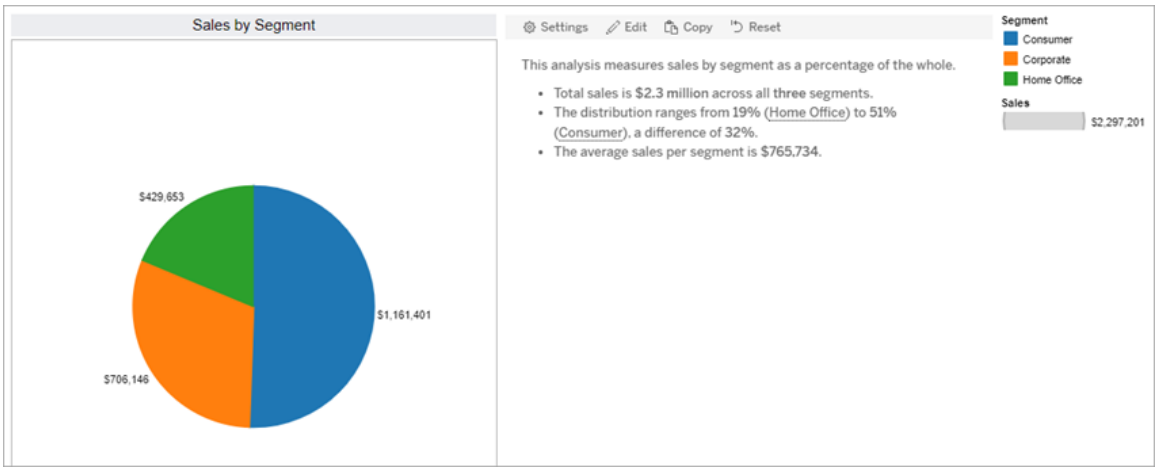


Percent of whole

Percent of whole stories are best for pie charts. To use a percent of whole story, your worksheet must have:

- 1 dimension
- 1 measure

The following example is a percent of whole story that uses a pie chart with a single dimension and a single measure:



Scatter plot

Scatter plot stories are best for understanding the relationship between two measures. When you create a scatter plot story, the story includes content about the relationship (regression) between two measures. And the story includes content about groups (clusters) within the data, when they exist.

Consider using a scatter plot story when you want to:

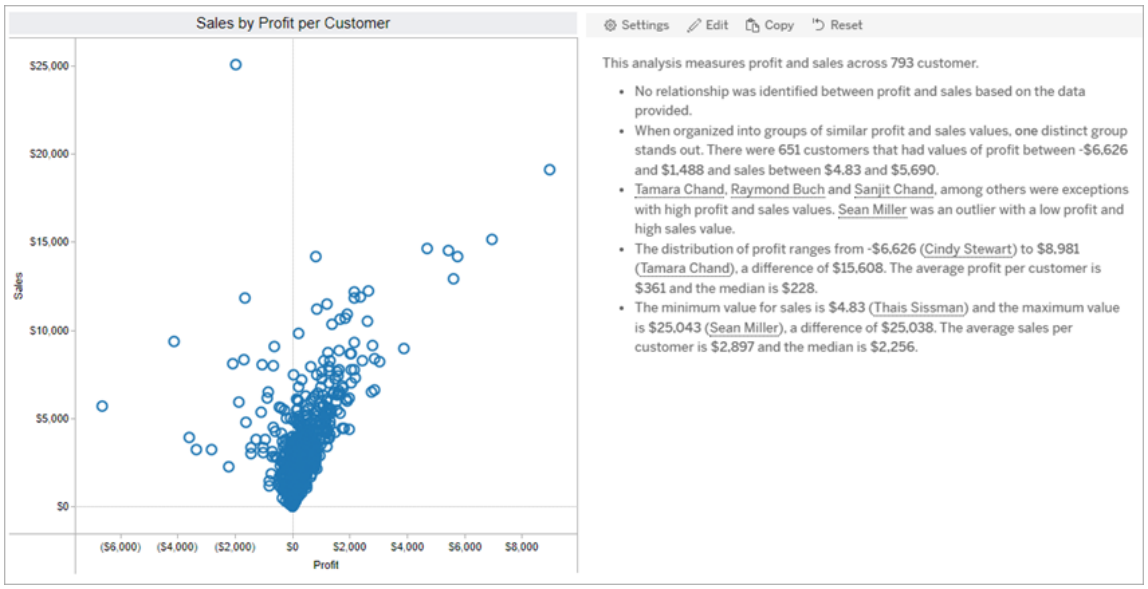
- Call out relationships between two measures to identify impact (regression analysis).
- Identify and understand outliers that are above or below defined thresholds.
- Analyze how your data is distributed.

To use a scatter plot story, your worksheet must have:

- 1 dimension
- 2 or 3 measures

Note: When you create your scatter plot story, the first measure you select is treated as the independent variable and the second measure is the dependent variable.

The following example is a scatter plot story that uses a scatter plot that has a single dimension and two measures:



Configure Settings for a Tableau Data Story

Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

After you Add a Tableau Data Story to a Dashboard, you can configure and edit your Tableau Data Story so it's tailored to your needs–use language specific to your data, specify which analytics are written about, and customize how your Tableau Data Story is displayed.

Configure Tableau Data Story Settings: Analytics

Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on

top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

Within your Tableau Data Story, you can choose which analytics to write about and when those analytics are written about. Different types of analytics are available depending on your story type and how many dimensions and measures your story has. However, analytics aren't currently supported for scatter plot story types. For more information, see Choose the Right Story Type for Your Tableau Data Story.

Configure analytics for your story

1. Add a Tableau Data Story to a Dashboard.
2. From your dashboard, click the **Settings** icon at the top-left corner of your **Data Story** object.
3. In the Data Story dialog box, click the **Analytics** tab.
4. Click the switches to turn on different types of analytics.
5. For **Segments** and **Trend line**, expand **Settings** to set thresholds for performing those analytics.
6. Click **Save**.

Understand different types of analytics

Correlation

Use **Correlation** to identify true statistical correlations between two series. If you have more than two series, then all combinations are analyzed for correlations. For example, you might turn on **Correlation** to identify when two products are often purchased together.

Clustering

Use **Clustering** to identify distinct groups of data points (clusters) using a single statistical analysis. For example, you might turn on **Clustering** to identify when a product is very popular in a specific geographic region.

Distribution

Use **Distribution** to rank data points relative to each other using nonstatistical observations, such as mean, median, skew, etc. For example, you might turn on **Distribution** to identify which product has the highest profit ratio.

Segments

Use **Segments** to highlight noteworthy changes to data points within a series. First, set the minimum percentage of change that you want to be written about in a segment. Changes that fall below your defined threshold aren't written about. For example, if you set your segment threshold for changes that are greater than 60%, then your story doesn't write about a trough in a time series that features a 30% decrease.

After you've set your threshold, choose whether to **Apply formatting**, and set the minimum percentage of change that you want formatted.

Trend line

Use **Trend line** to calculate a linear best fit line and identify data that falls within a defined percentage of confidence. Data that has high variability has a lower confidence level than data that's more consistent, and that confidence level affects whether trend lines are written about. You can use trend lines for stories that have one dimension and one measure, or you can use trend lines in a drilldown. For more information about drilldowns, see Configure Tableau Data Story Settings: Narrative.

Set the minimum percentage of confidence for your trend line. If you set your threshold at 95%, but a trend line could be drawn at 90% confidence, then your story doesn't write about trend lines. After you've set your threshold, choose whether to **Apply formatting**. Then set the minimum percentage of change that you want formatted.

Tableau Data Stories about trend lines communicate the absolute change over a period. The story written about your trend line varies depending on the level of verbosity you set for your story. If your story uses high verbosity, then your story writes about the R-squared value,

which is a statistical concept that quantifies how well your data fits the trend line. For more information about verbosity settings, see Configure Tableau Data Story Settings: Narrative.

Within the **Trend line** settings, you can also choose how many periods into the future for which you want your story to write predictions. When you use predictions, your story uses the slope and intercept of the trend line to calculate predicted values for future periods. The confidence of the prediction adds upper and lower bounds to the confidence threshold you set for trend lines. You can use predictions when your story has at least 30 data points that are linear.

Volatility

Use **Volatility** to analyze standard deviations over time. For example, use **Volatility** when you want your story to write about values that fall outside the average range for your data.

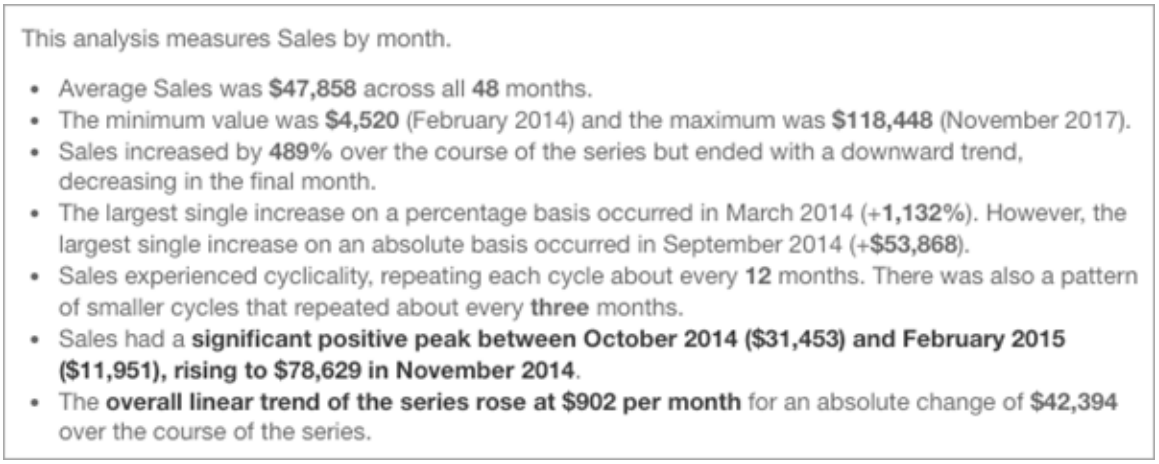
Break down how analytics are used to generate stories

At this point, you might be wondering how the analytics for different story types work. Let's take a look at an example for each story type and break down each sentence in the story.

Understand analytics for discrete stories

Because continuous stories measure trends over time, Data Stories writes about performance, progression, averages, totals, streaks, volatility, segments, and predictions.

The following example of a continuous story is about sales per month:


This analysis measures Sales by month.

- Average Sales was **$47,858** across all **48** months.
- The minimum value was **$4,520** (February 2014) and the maximum was **$118,448** (November 2017).
- Sales increased by **489%** over the course of the series but ended with a downward trend, decreasing in the final month.
- The largest single increase on a percentage basis occurred in March 2014 (+**1,132%**). However, the largest single increase on an absolute basis occurred in September 2014 (+**$53,868**).
- Sales experienced cyclicality, repeating each cycle about every **12** months. There was also a pattern of smaller cycles that repeated about every **three** months.
- Sales had a **significant positive peak between October 2014 ($31,453) and February 2015 ($11,951), rising to $78,629 in November 2014**.
- The **overall linear trend of the series rose at $902 per month** for an absolute change of **$42,394** over the course of the series.

Example story	Story breakdown
• Average Sales was **$47,858** across all **48** months. • The minimum value was **$4,520** (February 2014) and the maximum was **$118,448** (November 2017).	The first two sentences use average and range functions to write about the average, maximum, and minimum values across the period you're analyzing.
• Sales increased by **489%** over the course of the series but ended with a downward trend, decreasing in the final month.	The third sentence is about overall performance of the measure over the period. For example, a sentence can be about whether sales increased, decreased, or trended differently during a specific period.
• The largest single increase on a percentage basis occurred in March 2014 (**+1,132%**). However, the largest single increase on an absolute basis occurred in September 2014 (**+$53,868**).	The fourth sentence uses progression analysis. This sentence writes about the largest increase and decrease based on the measure during the period using both a percentage basis and absolute basis.
• Of the **three** series, the strongest relationship was between Corporate and Home Office, which had a moderate positive correlation, suggesting that as one (Corporate) increases, the other (Home Office) generally does too, or vice versa.	This sentence is a **Correlation** insight. This type of analytic insight writes about notable correlations between different series in your data.
• Sales experienced cyclicality, repeating each cycle about every **12** months. There was also a pattern of smaller cycles that repeated about every **three** months. • Sales had a **significant positive peak between October 2014 ($31,453) and February 2015 ($11,951), rising to $78,629 in**	This sentence is a **Segment** insight. This type of analytic insight writes about noteworthy increases and decreases over time.

November 2014.	
• The **overall linear trend of the series rose at $902 per month** for an absolute change of **$42,394** over the course of the series. If this trend continued for the next one month, Sales is **predicted to be about $69,958**.	This sentence is a **Trend line** insight. This type of insight writes about how well trends fit your data with a certain percentage of confidence, and trend lines allow you to make predictions based on historic trends.

Understand analytics for discrete stories

Because discrete stories allow you to compare values and understand the distribution of the data, the story writes about distribution, averages, totals, and groupings or clusters across the data.

The following example of a discrete story is about sales by product:

This analysis measures Sales by product.

- Total Sales is **$2.3 million** across all **17** products.
- The Sales of **$2.3 million** was driven by Phones with **$330,007**, Chairs with **$328,449** and Storage with **$223,844**.
- The distribution ranges from **$3,024** (Fasteners) to **$330,007** (Phones), a difference of **$326,983**, averaging **$135,129**.
- The distribution is positively skewed as the average of **$135,129** is greater than the median of **$114,880**.
- Sales is somewhat concentrated with **eight** of the **17** products (**47%**) representing **78%** of the total.
- The top **two** products represent over a quarter (**29%**) of overall Sales.
- Phones (**$330,007**) is more than **two** times bigger than the average across the **17** products.

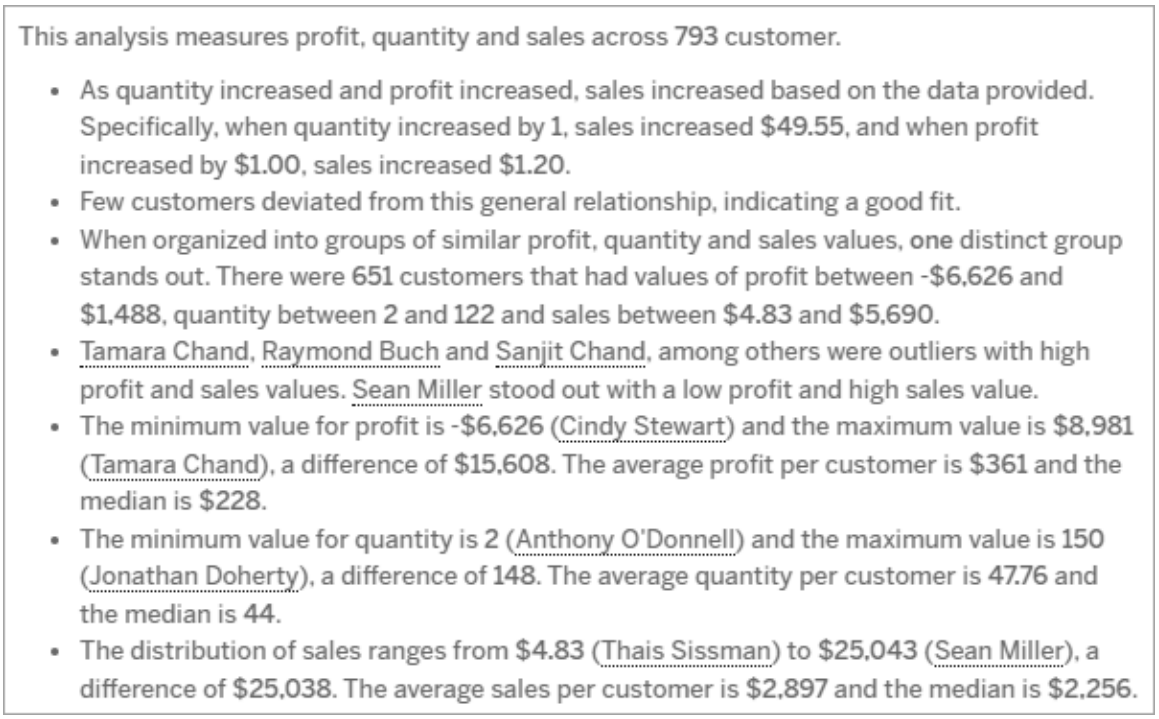
Example story	Story breakdown
• Total Sales is **$2.3** million across all **17** products.	The first sentence calculates the total value of your measure.
• The Sales of **$2.3** million was driven by Phones with **$330,007**, Chairs with **$328,449**, and Storage with **$223,844**.	The second sentence writes about the dimension drivers. In this example, the dimension drivers are the products that contributed the

	most to total sales.
• The distribution is positively skewed as the average of **$135,129** is greater than the median of **$114,880**. • Sales is relatively concentrated with **78%** of the total represented by **eight** of the **17** products (**47%**).	The third and fourth sentences analyze the distribution of the data. This analyzes the averages, medians, concentration of data (if any exist), and how the data is skewed. This helps identify how balanced these grouped variables are compared to one another.
• The top **two** products combine for over a quarter (**29%**) of overall Sales.	This sentence uses **Clustering** to write about measures that can be grouped. This helps identify whether there are distinct groups that stand out in the data.
• Phones (**$330,007**) is more than two times bigger than the average across the **17** products.	The final sentence writes about notable outliers.

Understand analytics for scatter plot stories

Scatter plot story types are best used to understand the relationship between two measures, and for that reason, scatter plot stories require 2–3 measures. The scatter plot analysis writes about the relationship (regression) between two measures, and it writes about groups (clusters) within the data, if they exist.

The following example of a scatter plot story is about profit and sales across a dimension:



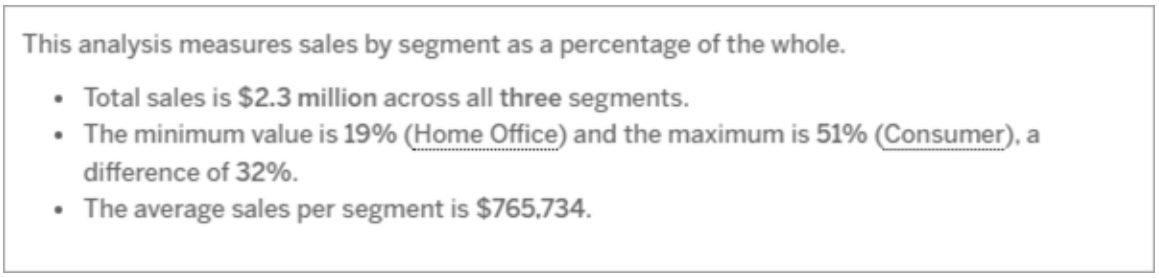
Example story	Story breakdown
• As quantity increased and profit increased, sales increased based on the data provided. Specifically, when quantity increased by **1**, sales increased **$49.55**, and when profit increased by **$1.00**, sales increased **$1.20**. • Few customers deviated from this general relationship, indicating a good fit.	The first two sentences are powered by regression analytics. Regression shows how one measure affects another. Notice that in the first sentence, the story has identified a relationship between profit and sales.
• When organized into groups of similar profit, quantity and sales values, one distinct group stands out. There were **651** customers that had values of profit between **-$6,626** and **$1,488**, quantity between **2** and **122** and sales between **$4.83** and **$5,690**.	The third sentence is derived from clustering. Clustering analytics tries to identify key groups or clusters across all the variables in the data.

• Tamara Chand, Raymond Buch, and Sanjit Chand, among others were outliers with high profit and sales values. Sean Miller stood out with a low profit and high sales value.	The fourth sentence is written about outliers–values that fall significantly above or below the average.
• The minimum value for profit is **-$6,626** (Cindy Stewart) and the maximum value is **$8,981** (Tamara Chand), a difference of **$15,608**. The average profit per customer is **$361** and the median is **$228**. • The minimum value for quantity is **2** (Anthony O'Donnell) and the maximum value is **150** (Jonathan Doherty), a difference of **148**. The average quantity per customer is **47.76** and the median is **44**. • The distribution of sales ranges from **$4.83** (Thais Sissman) to **$25,043** (Sean Miller), a difference of **$25,038**. The average sales per customer is **$2,897** and the median is **$2,256**.	The remaining sentences for scatter plot stories use range and average analysis to write insights.

Understand analytics for percent of whole stories

Percent of whole story types are best for understanding what part of a whole a dimension or measure represents.

The following example of a percent of whole story is about sales by segment:



Example story	Story breakdown
• Total SUM(Sales) is 2.3 million across all three entities.	The first sentence calculates the total value of your measure.
• The SUM(Sales) of 2.3 million was driven by Consumer with 1.2 million, Corporate with 706,146 and Home Office with 429,653.	The second sentence writes about drivers. In this example, the drivers are segments that contributed the most to total sales.
• The minimum value is 429,653 (Home Office) and the maximum is 1.2 million (Consumer), a difference of 731,748, averaging 765,734.	The final sentence analyzes the distribution of the data.

Configure Tableau Data Story Settings: Characteristics

Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

Within your Tableau Data Story, you can configure the characteristics settings to give context to your data, so you get more insightful stories. For example, in your story you can specify that in the context of sales, a higher number is good. But in the context of customer complaints, a higher number is bad.

Use dimension and measure characteristics

Your story and number formatting adjusts based on what the measure is. By default, your story writers all measure values as numbers, and your story won't perform any additional calculations or apply any special rendering rules.

1. Add a Tableau Data Story to a Dashboard.
2. From your dashboard, click the **Settings** icon at the top-left corner of your Data Story object.
3. In the Data Story dialog box, click the **Characteristics** tab.
4. Configure your formatting, such as number type, decimal places, and negative values.
5. Click **Save**.

Learn more about measure characteristics

Formatting

If values are formatted as **Percentages**, then the story writes about percentage point differences, rather than percent changes as a story would for number values. When you format a number as a **Percentage** in the **Characteristics** tab, the **Data Story** multiplies the value of the number by 100 to create the percentage that shows in your story.

If values are formatted as **Currency**, then you can specify your preferred currency. You can also specify how you want large values (numbers greater than one million) formatted, for example $1.3 million instead of $1,300,000.00.

For both **Numbers** and **Currency**, you can specify how you want large values and negative values to be written about. If you choose to have negative values written about in parentheses, you might see nested parentheses in your story when the negative value is written about in a parenthetical phrase.

When you choose **Number** formatting, you can also specify whether you want numbers less than or equal to 10 to be spelled out (rather than using a numeral) in your story.

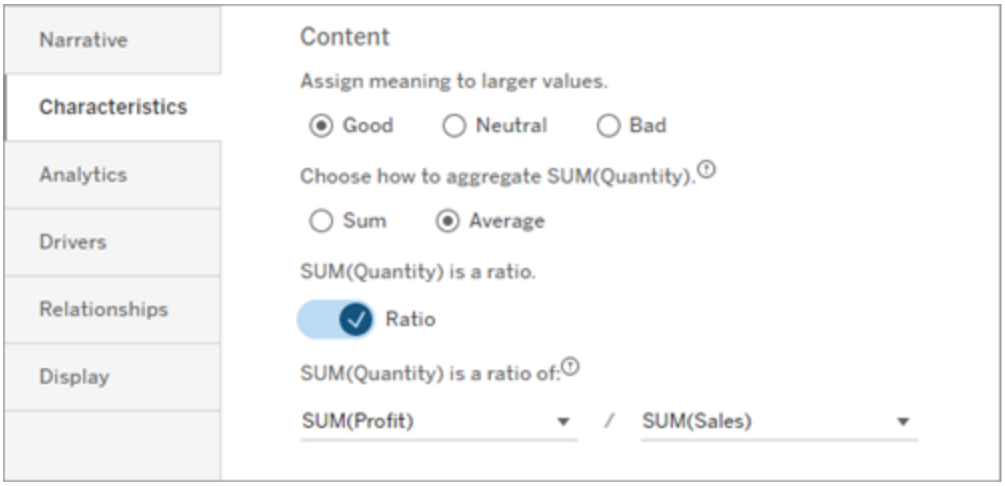
For decimal places, **Dynamic** is the default option. This means that the story rounds to different decimal places depending on how large or small the number is. If the percent value is less than 10, then the number has two decimal places. If the percent value is greater than 10,

then the number rounds to the nearest whole number. You can also specify how many decimal places you want used, which is used consistently throughout your story.

Content

You can assign meaning to larger values. For example, larger values for sales are good, but larger values for losses are bad.

In addition, you can choose how to aggregate values by sum or average. It's a best practice to choose the same aggregation method that you're using in the viz. For ratio measures, choose **Average** and then define the ratio by selecting the component measures of that ratio measure. Measures that are components of a ratio must be summable.



For cumulative measures (available for continuous stories only), choose **Sum** and then specify that the measure is already cumulative. Continuous stories write about the total of the measure across the series.

Sorting

To sort dimension values, click the arrow up/down icon to sort based on the oldest or newest time values in your dimension.

Note: Sorting dimension values is available for only continuous stories.

Configure Tableau Data Story Settings: Display

Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

You can configure how the text in your Tableau Data Story is displayed, such as font color and size. You can also choose whether your story uses bulleted lists or paragraphs.

Configure the display for your story

1. Add a Tableau Data Story to a Dashboard.
2. From your dashboard, click the **Settings** icon at the top-left corner of your Data Story object.
3. In the Data Story dialog box, click the **Display** tab.
4. Choose whether you want your story structure to be **Bullets** or **Paragraphs**.
5. Choose your **Font Size**.
6. Pick colors to represent good and bad changes (available for continuous stories).
7. Choose whether to use **Dynamic Ordering**.
8. Choose whether to use a **Condensed View**.
9. Click **Save**.

Understand when to use story display settings

To use color, your story must be continuous. When using color, you can choose colors from the palette to represent good changes and bad changes. For your story to know whether a change is good or bad, you must assign meaning to larger values in the **Characteristics** tab. For more information, see Configure Tableau Data Story Settings: Characteristics. After you configure your display, the styles and colors are applied in your story based on thresholds for trend line or segment analytics.

If you turn on **Dynamic Ordering**, then the insights for measures in a story are dynamically ordered from the best to the worst average value. If you already have an order you want to maintain, turn off **Dynamic Ordering**.

If you turn on **Condensed View**, then additional space is removed from your story. This is helpful if you don't have much extra space in your dashboard or when you have multiple Data Story objects in a dashboard.

Configure Tableau Data Story Settings: Drivers

Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

Hypothetically, let's say your month-over-month sales increased significantly. What drove that increase in sales? And what might have detracted from (offset) those increased sales? Setting up drivers in your Data Story can answer those questions.

In Data Stories, drivers contribute toward a total value. Offseters detract from a total value. You'll find insights about drivers and offsetters in discrete and continuous stories. And these insights make it easy to understand exactly what's going on in the data and why.

Set dimension drivers

1. Add a Tableau Data Story to a Dashboard.
2. From your dashboard, click the **Settings** icon at the top-left corner of your Data Story object.
3. In the Data Story dialog box, click the **Drivers** tab.
4. From the **Dimension Drivers** section, select the type of driver that has the greatest impact on your analysis:
   For **Count**, set the maximum number of contributors and offsetters.
   For **Individual %**, set thresholds for writing about individual contributors and offsetters.

For **Cumulative %**, set thresholds for writing about contributors and offsetters based on their collective value.

5. Click **Save**.

Understand dimension driver types

- **Count** specifies the number of entities (contributors and offsetters) called out in your story. For example, use **Count** to see the top three contributors and offsetters in your data.
- **Individual %** sets a threshold, and values higher than that threshold are included in your story. For example, use **Individual %** to specify that you want to write about only entities that represent more than 5% of the total value.
- **Cumulative %** sets a percentage threshold of the total value that included entities collectively account for. For example, use **Cumulative %** to specify that you want to write about the entities that contributed to at least 90% of that total value. In this example, entities are written about in order of magnitude until the cumulative value of those entities account for 90% of the total value.

Use secondary contributors

To use secondary contributors, you must have a second dimension that isn't time. When you use secondary contributors, each driver that is written about also has details about and drivers for its secondary contributor. For example, if you are analyzing store sales, a secondary contributor would be a class within a department. Secondary contributors allow for deeper analysis. But secondary contributors can also contain a lot of information to fit into a single sentence in your story.

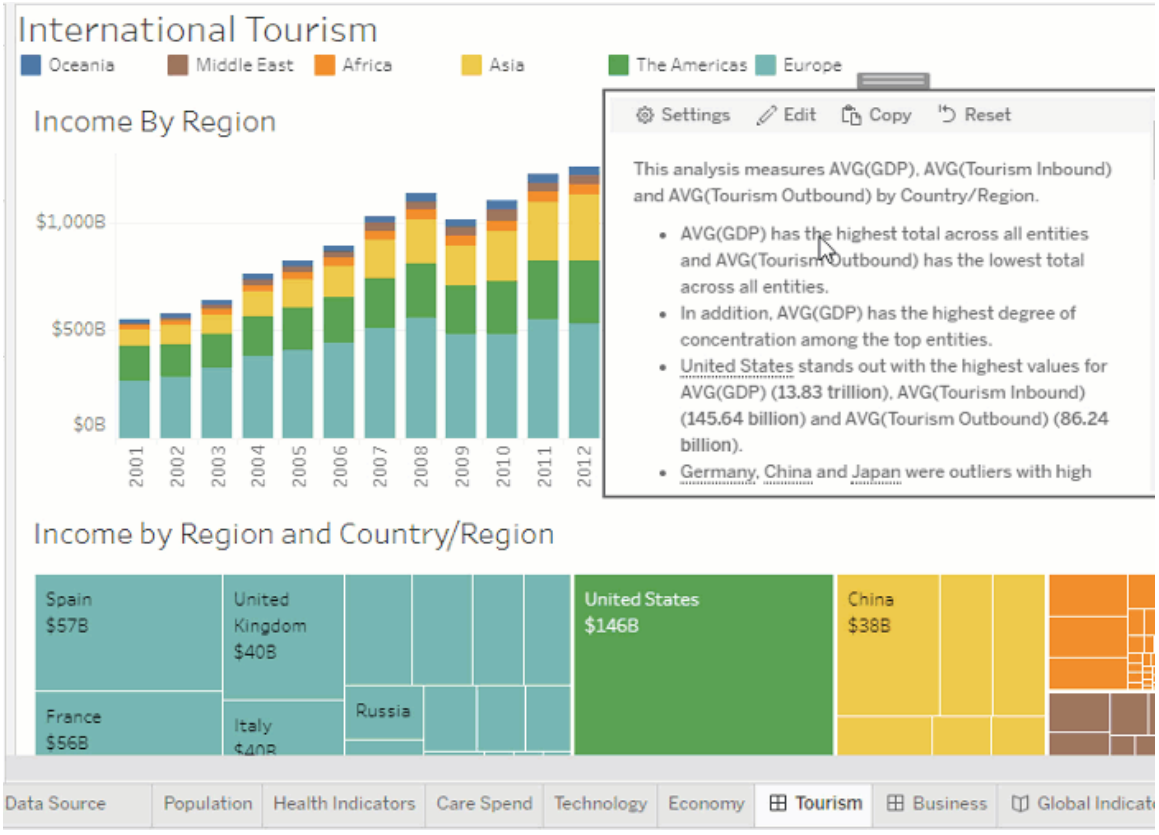
Set metric drivers

For measures that are composed of other subcategory measures, driver analysis can explain the impact that each measure had on the top-level value. For example, material costs and operating costs contribute to total cost.

To use metric drivers, you must have multiple measures for metric analysis. Then, you can specify the relationships between each measure.

1. Add a Tableau Data Story to a Dashboard.
2. From your dashboard, click the **Settings** icon at the top-left corner of your Data Story object.
3. In the Data Story dialog box, click the **Drivers** tab.
4. From the **Metric Drivers** section, first choose the measure that is a subcategory of another measure.
5. Then, choose the measure that is the primary category.
6. Click **Save**.

Tip: The verbosity setting also applies to drivers. By changing your story's verbosity setting, you can adjust the way insights are written. If you use high verbosity, then you'll see more information in parentheses. If you use low verbosity, then you'll get a more concisely written insight about your drivers. For more information, see Configure Tableau Data Story Settings: Narrative.



Configure Tableau Data Story Settings: Narrative

Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

You can customize the narrative settings of your Tableau Data Story. Specifically, you can specify the verbosity and level of analytical detail in your story, and you can add terms that are unique to your data. These settings help you write a story that uses the right language and speaks to the right level of detail for your dashboard.

Set verbosity

Verbosity specifies the length and analytical detail written in your story. If you choose high, then your story has longer insights with more analytic analysis. If you choose low, then your story is more concise with fewer details.

If you allow viewers to change verbosity, then viewers of published dashboards can change the level of verbosity in the Data Story object. This is helpful when your dashboard is used by a broader audience with viewers who want varying levels of detail from your story.

Set drilldowns

A drilldown includes two dimensions, and drilldowns describe figures associated with each dimension of your dashboard.

Let's say you have a dashboard that has monthly sales by product category. Your story is configured to write about both the **Time** and **Category** dimensions. In this case, this story includes an insight for each Category that describes its performance in and across the **Time** dimension.

By setting the maximum number of drilldowns, you can control how many insights are included in your story. Drilldown insights are also ranked based on the meaning assigned to

the measure characteristics. As the number of drilldowns is reduced, the lowest performing measures (measures that are assigned a Bad meaning) are eliminated.

Add dimension terms

By adding terms, you can define the way each of your measures and dimensions are labeled and referenced in your story.

1. Add a Tableau Data Story to a Dashboard.
2. From your dashboard, click the **Settings** icon at the top-left corner of your Data Story object.
3. In the Data Story dialog box, click the **Narrative** tab.
4. Expand the dimension to see how it will be written about in both singular and plural form.
5. Click **Add Term** to add another variation for your story to use to describe your dimension.
6. Click **Save**.

Your story uses (at random) the terms you've added when writing about a dimension.

Manage measure labels

Similarly to dimensions, you can manage labels used for measures in your story.

1. From the **Narrative** tab, expand the measure to see its label.
2. Enter the new label that you want used for your measure.
3. Click **Save**.

Configure Tableau Data Story Settings: Relationships

Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

The **Relationships** setting allows you to assign relationships between measures in a Tableau Data Story that has multiple measures. To configure Relationships, your story must have one

dimension and multiple measures. For more information, see Create Custom Measure Relationships in Your Tableau Data Story.

There are two types of relationships:

- Actual vs. Benchmark
- Current/Most Recent vs Previous Period

Use **Actual vs. Benchmark** when you want to know if you're performing above or below your performance benchmarks, for example, when performing quota reporting. This type of relationship is also helpful for identifying data points that require additional analysis because they're significantly above or below your benchmark. To use **Actual vs. Benchmark**, the measures you're comparing must have the same value type.

Use **Current/Most Recent vs. Previous Period** when you want to see if your key performance indicators (KPIs) are increasing, decreasing, or remaining consistent over time. To use **Current/Most Recent vs. Previous Period**, you must use a discrete story type. For more information, see Choose the Right Story Type for Your Tableau Data Story.

Additionally, you can use **Actual vs. Benchmark** and **Current/Most Recent vs. Previous Period** relationships simultaneously. Measures in your viz that aren't part of the configured relationships are written about in separate paragraphs.

Create Actual vs. Benchmark relationship for continuous or discrete stories

Use the **Actual vs. Benchmark** relationship when one measure is a benchmark for other measures. For example, you could compare actual sales to a sales target, so your story writes insights about whether you outperformed or underperformed your goal. When you use this type of relationship, the story removes unnecessary content and focuses on what's most important–comparing a metric to its associated benchmark.

1. Add a Tableau Data Story to a Dashboard.
2. From your dashboard, click the **Settings** icon at the top-left corner of your Data Story object.
3. In the Data Story dialog box, click the **Relationships** tab.
4. Check the box for **Actual vs. Benchmark**.
5. First, select the measure that is the benchmark.

6. Then, select the measure that you want to compare against the benchmark.
7. Click **Save**.

Create Current/Most Recent vs. Previous Period relationship

Use the **Current/Most Recent vs. Previous Period** relationship to compare the performance of two measures over a period. For example, you could compare two products to see which product generated the most revenue over the last year.

1. Add a Tableau Data Story to a Dashboard.
2. From your dashboard, click the **Settings** icon at the top-left corner of your Data Story object.
3. In the Data Story dialog box, click the **Relationships** tab.
4. Check the box for **Current/Most Recent vs. Previous Period**.
5. First select the measure for the previous period.
6. Then, select the measure for the current period.
7. Enter the label for the period that you're measuring, for example, year.
8. Choose the number of periods to measure.
9. Click **Save**.

Customize Your Tableau Data Story

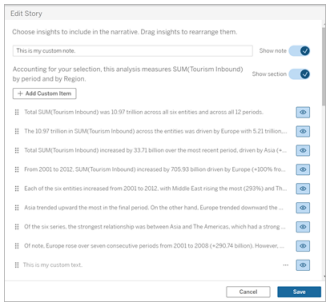
Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

You can use custom language, tailored to your audience, to supplement your Tableau Data Stories with insights specific to your business. Identify the analytics and data from the Data Story that matters most to your audience, and use your own language to create the most impactful story. As with the overall Tableau Data Story, data and variables used in custom content are dynamic, adjusting along with the dashboard.

Add your own insights

1. In your **Data Story**, click **Edit** to open the Edit dialog box.
2. Find the section you want to write about and click **Add Custom Item.**



3. Enter your custom text in the field that appears.
4. Click **Save**.

Add headers and footers

You can insert custom text at the top and bottom of your **Data Story**. With headers and footers, you can add your own qualitative analysis to stories, include additional explanations of data trends, or append legal and privacy disclaimers.

1. In your **Data Story**, click **Edit** to open the Edit dialog box.
2. Click the **Show note** switch.
3. Enter your custom note.

4. Click **Save**.

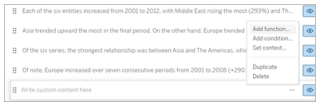
Your note now shows at the top or bottom of your **Data Story**.

Add functions

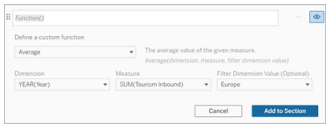
Using a function in your Data Story is a great way to customize your story and find the insights that are most important to you and your business.

For example, if you want to know the average sum of tourism revenue over a period, select Average as your function and then designate the measure as SUM(Tourism Inbound). This returns the average sum of inbound tourism.

1. In your **Data Story**, click **Edit** to open the Edit dialog box.
2. Click the menu in the right side of the box and select **Add Function**.



3. Select a Data Story Function and fill in the required fields.



4. Click **Add to Section**.
5. Click **Save**.

The custom content now shows in your Data Story.

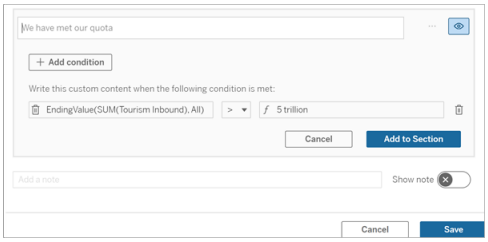
Add conditions

For each custom sentence you write, you can add a condition that determines whether the sentence renders in your **Data Story**. If the condition is met, the custom sentence appears in your story. If the condition isn't met, the custom sentence doesn't appear.

You can apply multiple conditions to each custom sentence, and the conditions can be combined using the **Any** or **All** buttons within the Add a condition dialog box.

Conditional statements are most often used with numerical comparisons, but the function also supports string matching using the equal (=) or not equal (!=) symbols.

1. In your Data Story, click **Edit** to open the Edit dialog box.
2. Enter your custom sentence.
3. Click the menu on the right side of the box and select **Add Condition**.
4. Define the custom function to be used to inform the conditional logic. In this example, the sentence "We have met our quota" shows if the Sum of Tourism Inbound is greater than 5 trillion.



5. Click **Add to Section**.
6. Click **Save**.

The custom sentence now appears in your Data Story only if the conditions are met.

Duplicate custom content

You can easily duplicate custom content added to your Data Story, making it easier to build different variations of a sentence. We recommend copying a fully built custom sentence when applying thresholds, building in language variation, and creating different logical variations.

1. In your Data Story, click **Edit** to open the Edit dialog box.
2. Create a custom sentence, complete with functions and conditions, if desired.
3. In the completed sentence box, Click the menu in the right side of the box and select **Duplicate**.

4. Click into your duplicated sentence, update as desired, and click **Add to Section**.
5. Click **Save**.

When you copy a sentence, all functions and conditional statements also copy over. The copied bullet appears directly below the original bullet in the same section.

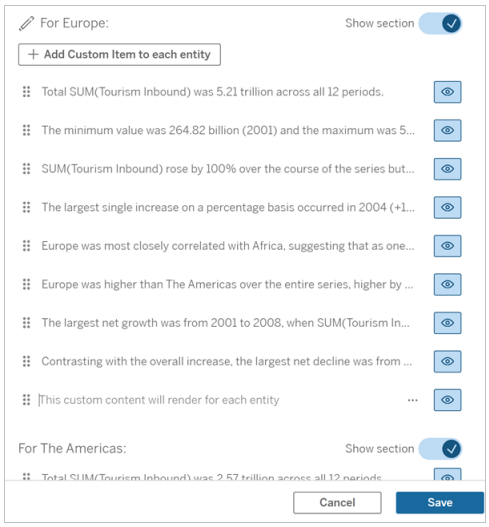
> **Note**: Copied bullets can only be added to the section that the original bullet is in.

Add custom content in drilldown sections

For stories that have two dimensions, each section after the first section is called a drilldown section. Drilldown sections focus on an individual primary dimension entity (i.e. The Americas in the following example).

Because the drilldown sections have the same content structure, custom content added in the first section (i.e., Europe) are applied to each additional section (i.e., The Americas). You can only create or edit content in the first drilldown section.

1. In your Data Story, click **Edit** to open the Edit dialog box.
2. In the first drilldown section, click **Add Custom Item** to each entity.
3. Enter your custom content.
4. Click **Save**.



Custom content in drilldown sections already has a context variable called Current Category value (dynamic). This creates a dimension value option called Current Category value (dynamic) which always represents the section that the drilldown section is about.

Tip: To create content that only appears in a single specific drilldown section, you can use conditional logic to ensure it only writes where appropriate.

Customize Your Tableau Data Story: Context Variables

Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

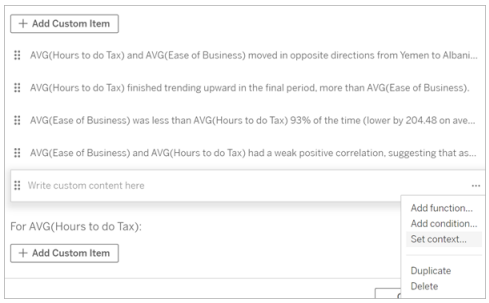
Context variables are functions that can be referenced by other functions. In other words, you can use context variables to nest functions within other functions.

After you define your context variable, it appears as a function that you can use when adding new functions to your Tableau Data Story.

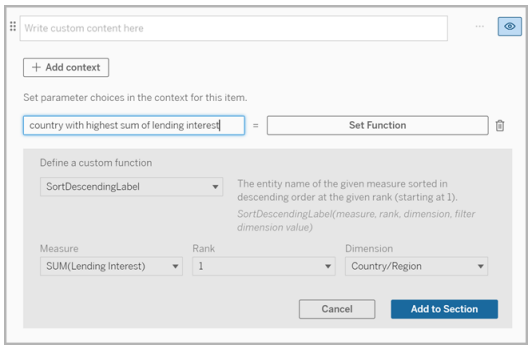
Note: You can have multiple context variables set for each custom sentence, but you must define each context variable separately for each piece of custom content.

Set a context variable

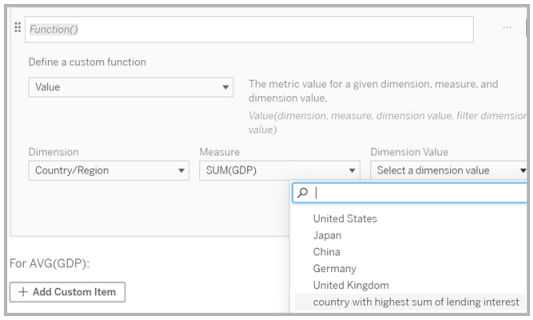
1. In your **Data Story**, click **Edit** to open the Edit dialog box.
2. Click **Add Custom Item.**
3. Click the menu in the right side of your custom content box and select **Set context**.



4. Click **Add context**.
5. Name the context variable and click **Set Function**.
6. Define your custom function and choose a dimension.



7. Click **Add to Section.**
8. Click back into the sentence where you set your context variable.
9. Follow the steps to Add functions.



Now, your context variable is listed as an option in the **Dimension Value** drop-down list when adding your function.

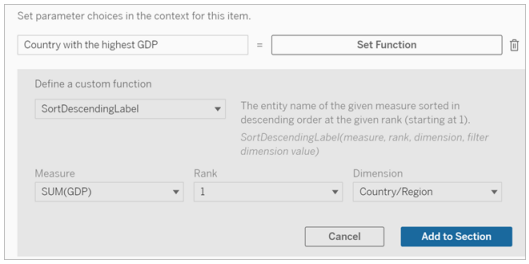
When to use a context variable: reference two or more measures

Let's say that you want to refer to two or more measures in one analytical sentence in your **Data Story**. Without a context variable, we can write a sentence for only one measure at a time. But if we use a context variable, we can reference more than one measure in one sentence.

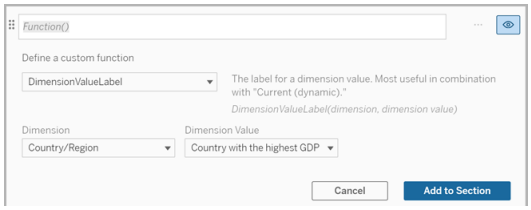
To reference two or more measures with a context variable, your **Data Story** must have:

- 1 dimension
- 2 or more measures

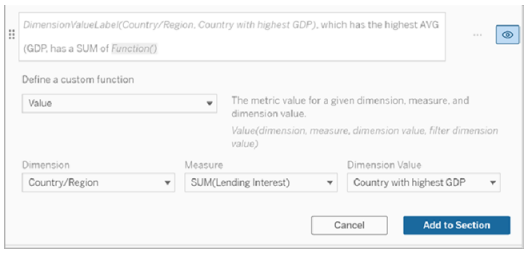
1. In your **Data Story**, click **Edit** to open the Edit dialog box.
2. Click **Add Custom Item**.
3. Click the menu in the right side of your custom content box and select **Set context**.
4. Click **Add context**.
5. Name the context variable and click **Set Function.**
6. Define your custom function and choose a dimension.
7. Click **Add to Section**.



8. Click back into the sentence where you set your context variable.
9. Add your first function and fill in the required fields. In this example, we selected **DimensionValueLabel**, and then chose Country/Region from **Dimension**, and then country with the highest GDP (our context variable) from **Dimension Value**.



10. Click **Add to Section**.
11. Add your second function and fill in the required fields. In this example, we selected **Value**, and then chose Country/Region from **Dimension**, SUM(Lending Interest) from **Measure**, and country with the highest GDP (our context variable) from **Dimension Value**.



12. Click **Add to Section.**
13. Click **Save.**

Your **Data Story** writes a sentence that gives us insight into a secondary measure (Lending Interest) for the country that we're interested in (the country with the highest GDP).

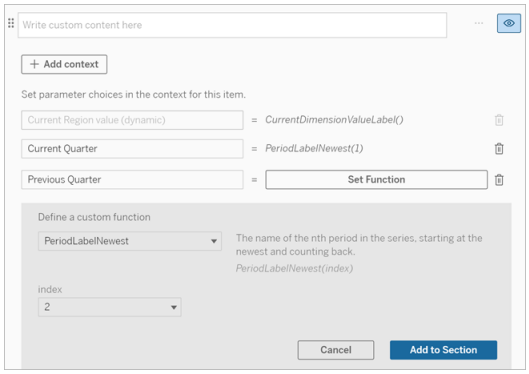
When to use a context variable: period-over-period analysis

A context variable is helpful when you want to analyze performance over two different periods in your **Data Story**. You can create a custom sentence that writes about a measure displayed in your drilldown section and compares the measure against different periods, such as year over year or month over month.

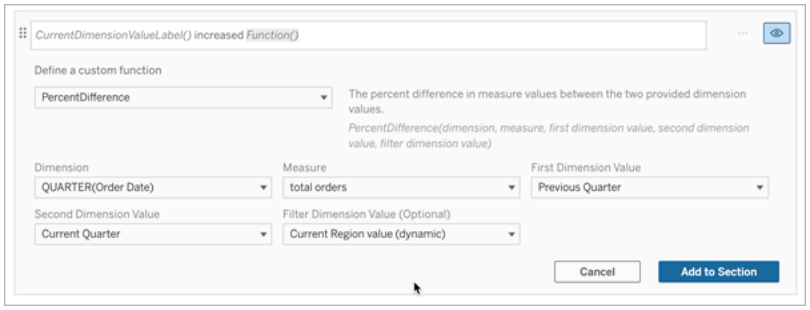
To set up a period-over-period analysis, your Data Story must have:

- 2 dimensions: 1 time period dimension (primary) and 1 non-time period dimension (secondary)
- 1–3 measures

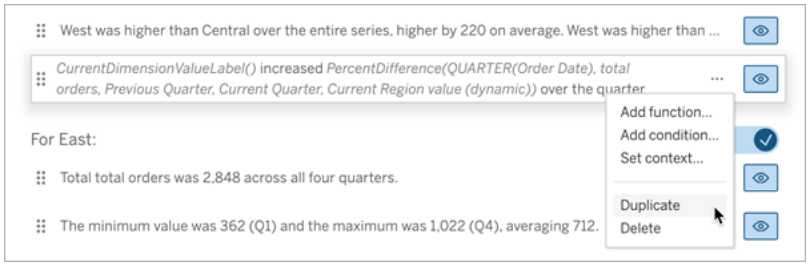
1. Create your **Data Story**.
2. In the Fields dialog box, make sure your time period dimension is ordered first and click **Next**.
3. In the Story dialog box, select **Continuous** and click **Done**.
4. Open the Edit dialog box, and select **Add Custom Item** in the first drilldown area.
5. Create two context variables that represent your time periods. For example, "Current Quarter" and "Previous Quarter."



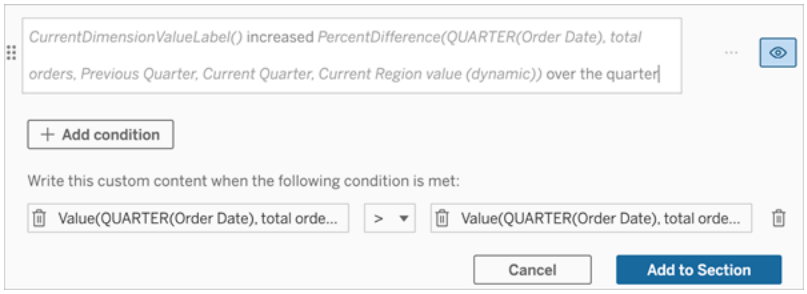
6. Create an *increased* sentence by adding custom language and functions. The content of this sentence results in "[CurrentDimensionValueLabel] increased [X%] over the quarter."
7. Type in the function followed by the word "increased" in the custom text box.
8. Add the function that returns the percent change of your measurement during your time period.



9. Click **Add to Section**.
10. Duplicate custom content and create a *decreased* version by replacing the word "increased" with "decreased." The functions stay the same.



11. Add conditions for each sentence so that only one is written, depending on the data.



12. For the *increased* sentence, set conditions in line with the following example:

    Left Argument = Value function

    Dimension = Quarter(OrderDate). Select your time period dimension

    Measure = SUM(Total Orders). Select the measure you used for the calculation

    Dimension Value = Current Quarter. One of the context variables

    Filter Dimension Value = Current Region value (dynamic). This is the preset context variable

    Middle Argument = > (greater than)

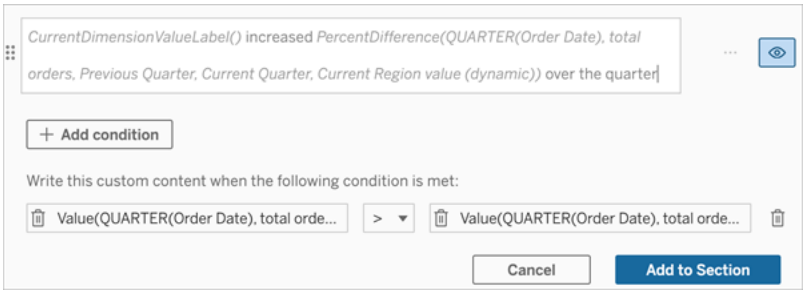
    Right Argument = Value function

Dimension= Quarter(OrderDate). Select your time period dimension

Measure = Total(Total Orders). Select the measure you used for the calculation

Dimension Value = Previous Quarter. One of the context variables

Filter Dimension Value = Current Region value (dynamic). This is the preset context variable



13. For the *decreased* sentence, set the same conditions, but replace the > (greater than) sign with the < (less than) sign. The right and left arguments remain the same.
14. Click **Save**, and your **Data Story** writes a sentence that includes the insights from analyzing the two time periods.

Customize Your Tableau Data Story: Functions

Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

If you'd like to customize your Tableau Data Story with a function, it helps to know what functions you can use, what each function does, and what dimensions and measures you'll need for each function.

Learn how to Add functions to your Data Story.

Click a letter to see functions that begin with that letter. If no functions start with that letter, the functions that start with the next letter in the alphabet are shown. You can also press Ctrl+F (Command-F on a Mac) to open a search box that you can use to search the page for a specific function.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Average

Description: The average value of the given measure.

Syntax: Average(dimension, measure, filterDimensionValue)

Count

Description: The number of dimension values within a given dimension.

Syntax: Count(dimension)

Difference

Description: The difference in measure values between the two provided dimension values.

Syntax: Difference(dimension, measure, firstDimensionValue, secondDimensionValue, filterDimensionValue)

DifferenceFromMean

Description: The difference between the mean and the measure value for the given dimension value.

Syntax: DifferenceFromMean(dimension, measure, firstDimensionValue, filterDimensionValue)

Direction

Description: Language describing the direction (e.g., increase or decrease) between measure values for the two provided dimension values.

Syntax: Direction(dimension, measure, firstDimensionValue, secondDimensionValue, filterDimensionValue, phrase)

Ending Label

Description: The name of the last period in the series.

Syntax: Label(measure)

EndingValue

Description: The value at the last period in the series for the given measure.

Syntax: EndingValue(measure, filterDimensionValue)

Label

Description: The label for the given measure.

Syntax: Label(measure)

LargestNegativeChangeDifference

Description: The value of the largest negative period-over-period difference, on an absolute basis, in the given series.

Syntax: LargestNegativeChangeDifference(measure, filterDimensionValue)

LargestNegativeChangeEndingLabel

Description: The name of the ending period for the largest negative period-over-period difference, on an absolute basis, in the given series.

Syntax: LargestNegativeChangeEndingLabel(measure, filterDimensionValue)

LargestNegativeChangeEndingValue

Description: The ending value of the largest negative period-over-period difference, on an absolute basis, in the given series.

Syntax: LargestNegativeChangeEndingValue(measure, filterDimensionValue)

LargestNegativeChangePercentDifference

Description: The percent change of the largest negative period-over-period difference, on an absolute basis, in the given series.

Syntax: LargestNegativeChangePercentDifference(measure, filterDimensionValue)

LargestNegativeChangeStartingLabel

Description: The name of the starting period for the largest negative period-over-period difference, on an absolute basis, in the given series.

Syntax: LargestNegativeChangeStartingLabel(measure, filterDimensionValue)

LargestNegativeChangeStartingValue

Description: The starting value of the largest negative period-over-period difference, on an absolute basis, in the given series.

Syntax: LargestNegativeChangeStartingValue(measure, filterDimensionValue)

LargestNegativePercentChangeDifference

Description: The value of the largest negative period-over-period difference, on a percentage basis, in the given series.

Syntax: LargestNegativePercentChangeDifference(measure, filterDimensionValue)

LargestNegativePercentChangeEndingLabel

Description: The name of the ending period for the largest negative period-over-period difference, on a percentage basis, in the given series.

Syntax: LargestNegativePercentChangeEndingLabel(measure, filterDimensionValue)

LargestNegativePercentChangeEndingValue

Description: The ending value of the largest negative period-over-period difference, on a percentage basis, in the given series.

Syntax: LargestNegativePercentChangeEndingValue(measure, filterDimensionValue)

LargestNegativePercentChangePercentDifference

Description: The percent change of the largest negative period-over-period difference, on a percentage basis, in the given series.

Syntax: LargestNegativePercentChangePercentDifference(measure, filterDimensionValue)

LargestNegativePercentChangeStartingLabel

Description: The name of the starting period for the largest negative period-over-period difference, on a percentage basis, in the given series.

Syntax: LargestNegativePercentChangeStartingLabel(measure, filterDimensionValue)

LargestNegativePercentChangeStartingValue

Description: The starting value of the largest negative period-over-period difference, on a percentage basis, in the given series.

Syntax: LargestNegativePercentChangeStartingValue(measure, filterDimensionValue)

LargestPositiveChangeDifference

Description: The value of the largest positive period-over-period difference, on an absolute basis, in the given series.

Syntax: LargestPositiveChangeDifference(measure, filterDimensionValue)

LargestPositiveChangeEndingLabel

Description: The name of the ending period for the largest positive period-over-period difference, on a absolute basis, in the given series.

Syntax: LargestPositiveChangeEndingLabel(measure, filterDimensionValue)

LargestPositiveChangeEndingValue

Description: The ending value of the largest positive period-over-period difference, on an absolute basis, in the given series.

Syntax: LargestPositiveChangeEndingValue(measure, filterDimensionValue)

LargestPositiveChangePercentDifference

Description: The percent change of the largest positive period-over-period difference, on an absolute basis, in the given series.

Syntax: LargestPositiveChangePercentDifference(measure, filterDimensionValue)

LargestPositiveChangeStartingLabel

Description: The name of the starting period for the largest positive period-over-period difference, on an absolute basis, in the given series.

Syntax: LargestPositiveChangeStartingLabel(measure, filterDimensionValue)

LargestPositiveChangeStartingValue

Description: The starting value of the largest positive period-over-period difference, on an absolute basis, in the given series.

Syntax: LargestPositiveChangeStartingValue(measure, filterDimensionValue)

LargestPositivePercentChangeDifference

Description: The value of the largest positive period-over-period difference, on a percentage basis, in the given series.

Syntax: LargestPositivePercentChangeDifference(measure, filterDimensionValue)

LargestPositivePercentChangeEndingLabel

Description: The name of the ending period for the largest positive period-over-period difference, on a percentage basis, in the given series.

Syntax: LargestPositivePercentChangeEndingLabel(measure, filterDimensionValue)

LargestPositivePercentChangeEndingValue

Description: The ending value of the largest positive period-over-period difference, on a percentage basis, in the given series.

Syntax: LargestPositivePercentChangeEndingValue(measure, filterDimensionValue)

LargestPositivePercentChangePercentDifference

Description: The percent change of the largest positive period-over-period difference, on a percentage basis, in the given series.

Syntax: LargestPositivePercentChangePercentDifference(measure, filterDimensionValue)

LargestPositivePercentChangeStartingLabel

Description: The name of the starting period for the largest positive period-over-period difference, on a percentage basis, in the given series.

Syntax: LargestPositivePercentChangeStartingLabel(measure, filterDimensionValue)

LargestPositivePercentChangeStartingValue

Description: The starting value of the largest positive period-over-period difference, on a percentage basis, in the given series.

Syntax: LargestPositivePercentChangeStartingValue(measure, filterDimensionValue)

LongestStreakDifference

Description: The difference over the longest streak of consecutive increases or decreases for the given series.

Syntax: LongestStreakDifference(measure, filterDimensionValue)

LongestStreakDirection

Description: The direction (positive or negative) of the longest streak of consecutive increases or decreases for the given series.

Syntax: LongestStreakDirection(measure, filterDimensionValue)

LongestStreakEndingLabel

Description: The name of the ending period for the longest streak of consecutive increases or decreases for the given series.

Syntax: LongestStreakEndingLabel(measure, filterDimensionValue)

LongestStreakEndingValue

Description: The ending value of the longest streak of consecutive increases or decreases for the given series.

Syntax: LongestStreakEndingValue(measure, filterDimensionValue)

LongestStreakLength

Description: The largest number of periods of consecutive increase or decrease for the given series.

Syntax: LongestStreakLength(measure, filterDimensionValue)

LongestStreakPercentDifference

Description: The percent difference over the longest streak of consecutive increases or decreases for the given series.

Syntax: LongestStreakPercentDifference(measure, filterDimensionValue)

LongestStreakStartingLabel

Description: The name of the starting period for the longest streak of consecutive increases or decreases for the given series.

Syntax: LongestStreakStartingLabel(measure, filterDimensionValue)

LongestStreakStartingValue

Description: The starting value of the longest streak of consecutive increases or decreases for the given series.

Syntax: LongestStreakStartingValue(measure, filterDimensionValue)

MaxLabel

Description: The name of the entity with the maximum value for the given measure.

Syntax: MaxLabel(dimension, measure, filterDimensionValue)

MaxValue

Description: The maximum value for the given measure.

Syntax: MaxValue(measure)

Median

Description: The median value for the given measure.

Syntax: Median(dimension, measure, filterDimensionValue)

MinLabel

Description: The name of the entity with the minimum value for the given measure. Syntax: MinLabel(dimension, measure, filterDimensionValue)

MinValue

Description: The minimum value for the given measure.

Syntax: MinValue(dimension, measure, filterDimensionValue)

PercentDifference

Description: The percent difference in measure values between the two provided dimension values.

Syntax: PercentDifference(dimension, measure, firstDimensionValue, secondDimensionValue, filterDimensionValue)

PercentOfWhole

Description: The percent in measure values for a given dimension value over the total measure values for that dimension.

Syntax: PercentOfWhole(dimension, measure, dimensionvalue, filterDimensionValue)

PeriodLabel

Description: The name of the nth period in the series, starting at 1.

Syntax: PeriodLabel(index)

PeriodLabelNewest

Description: The name of the nth period in the series, starting at the newest and counting back.

Syntax: PeriodLabelNewest(index)

PeriodValue

Description: The value of the given measure at the nth period in the series, starting at 1.

Syntax: PeriodValue(measure, index, filterDimensionValue)

PeriodValueNewest

Description: The value of the given measure at the nth period in the series, starting at the newest and counting back.

Syntax: PeriodValueNewest(measure, index)

Range

Description: The difference between the maximum and minimum values for the given measure.

Syntax: Range(dimension, measure, filterDimensionValue)

SortAscendingLabel

Description: The entity name of the given measure sorted in descending order at the given rank (starting at 1).

Syntax: SortAscendingLabel(measure, rank, dimension, filterDimensionValue)

SortAscendingValue

Description: The value of the given measure sorted in ascending order at the given rank (starting at 1).

Syntax: SortAscendingValue(measure, rank, dimension, filterDimensionValue)

SortDescendingLabel

Description: The entity name of the given measure sorted in descending order at the given rank (starting at 1).

Syntax: SortDescendingLabel(measure, rank, dimension, filterDimensionValue)

SortDescendingValue

Description: The value of the given measure sorted in descending order at the given rank (starting at 1).

Syntax: SortDescendingValue(measure, rank, dimension, filterDimensionValue)

StartingLabel

Description: The name of the first period in the series.

Syntax: StartingLabel()

StartingValue

Description: The value at the first period in the series for the given measure.

Syntax: StartingValue(measure, filterDimensionValue)

StartToFinishDifference

Description: The difference between the values for the first and last periods in the given series.

Syntax: StartToFinishDifference(measure, filterDimensionValue)

StartToFinishPercentDifference

Description: The percent difference between the values for the first and last periods in the given series.

Syntax: StartToFinishPercentDifference(measure, filterDimensionValue)

StdDev

Description: The standard deviation value for the given measure.

Syntax: StdDev(dimension, measure, filterDimensionValue)

Sum

Description: The sum of measure values for the two provided dimension values.

Syntax: Sum(dimension, measure, firstDimensionValue, secondDimensionValue, filterDimensionValue)

Total

Description: The sum total value for the given measure.

Syntax: Total(dimension, measure, filterDimensionValue)

Value

Description: The metric value for a given dimension, measure, and dimension value.

Syntax: Value(dimension, measure, dimension value, filterDimensionValue)

Z-Score

Description: The z-score for the given measure.

Syntax: Z-Score(dimension, measure, firstDimensionValue, filterDimensionValue)

Customize Your Tableau Data Story: Hide and Reorder Content

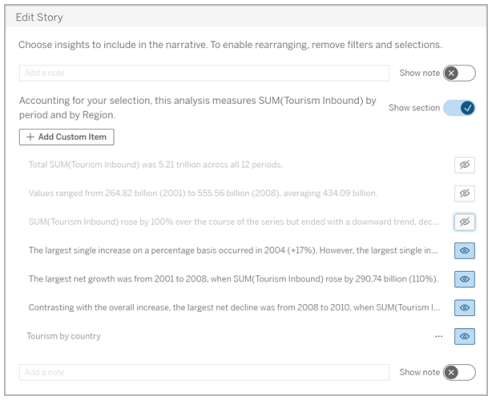
Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

Customize your Tableau Data Story with the content that is most important to you and your audience by hiding or reordering content within your story.

Hide content and sections

1. Create your **Data Story** and click **Edit** to open the Edit dialog box.
2. Set sections to show or hide by clicking the **Show section** switch to the on or off position.
3. Hover over the blue box to the right of each individual sentence, and click the box to show or hide from view.
4. Click **Save**.

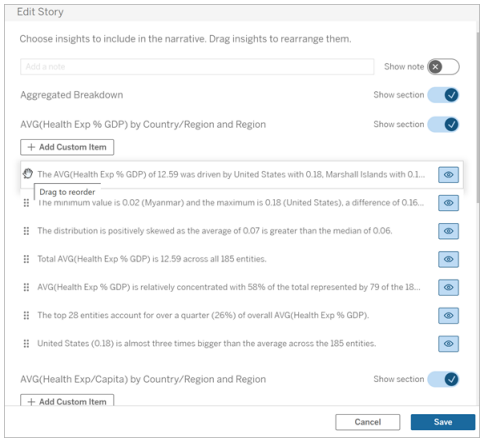


Now, only the sentences and sections that are set to **Show** appear in your Data Story.

Reorder content within a section

1. Create your **Data Story** and click **Edit** to open the Edit dialog box.
2. Hover over the left-side menu of the content you want to reorder. Your cursor turns into a hand icon.
3. Click the item with your cursor and drag it anywhere within the same section.

4. Click **Save**.



Now, the sentences appear in your **Data Story** in the order that you set them to.

> **Note:** Currently, content can only be moved within the same section. Moving entire sections isn't yet supported.

Add More Data to Your Tableau Data Story

Important changes for Tableau Data Stories

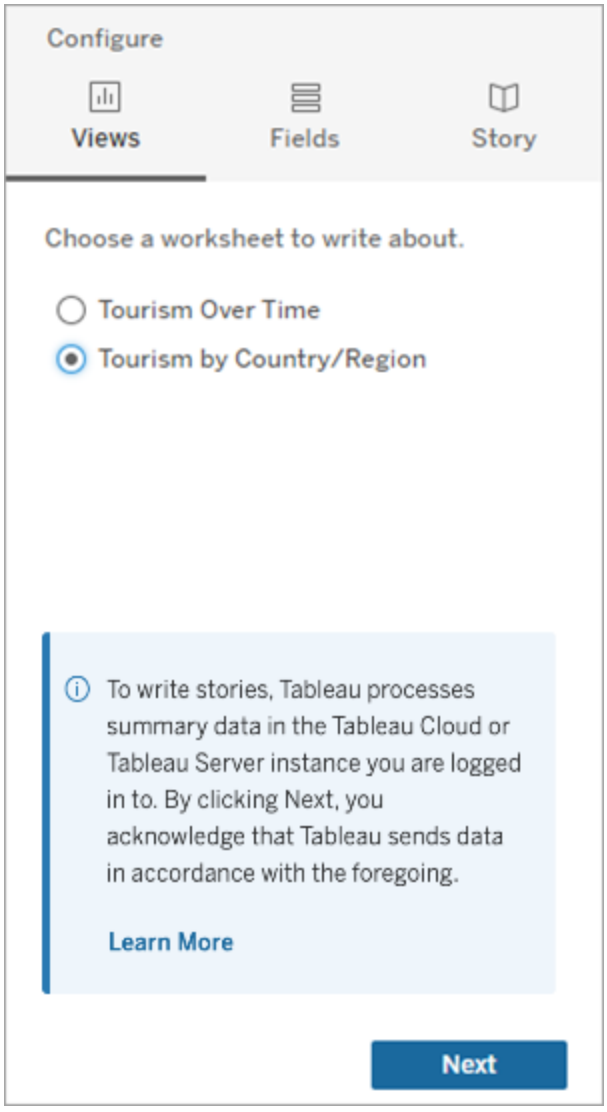
Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

Data Stories currently supports stories with two dimensions and one measure, or one dimension and up to 10 measures. If you'd like to write about data that you don't need to show on your dashboard, then use a hidden sheet to simplify your dashboard. If you'd like to add more than two dimensions to your story, then concatenate dimensions or create multiple data stories and stack them.

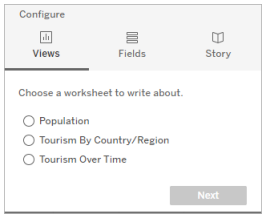
Use a hidden sheet

If you have data that you want to include in a Tableau Data Story and don't need to show all the data that drove the insight, you can use a hidden sheet to bring additional measures and dimensions into your story without cluttering the dashboard.

1. Drag the **Data Story** object to your dashboard to see which worksheets you can write about in the Data Story dialog box. In this example, there are two worksheets available to write about.

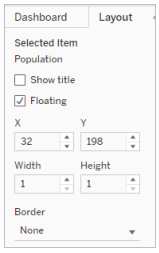


2. Bring in another data source, such as "Population," by navigating to the left-hand menu, selecting **Floating**, and dragging that sheet onto your dashboard.



The Data Story dialog box updates with the new available data source.

3. Click into **Layout** and adjust the size to 1 x 1 to hide the sheet but keep the underlying data in your story.



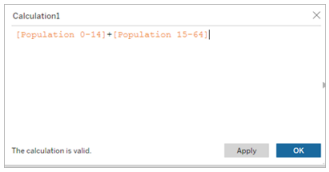
You can now configure your stories using this hidden sheet.

Note: You may need to deselect additional legend elements to keep the sheet hidden.

Concatenate dimensions

If your data contains three dimensions and one measure and is a **Discrete Story**, you can concatenate (link together) two of those dimensions by creating a calculated field.

1. From the worksheet you want to use in your story, click **Analysis** and select **Create Calculated Field**.
2. Name the calculated field and use the following formula to create your calculation, using the + sign to join the dimensions.
   [Dimension 1] + [Dimension 2]



Tip: Drag your dimensions into the **Calculated Field** box and place them in the formula.

3. Click **OK**.
4. Drag your new calculated field into the **Detail** pane to make it accessible in your data story.

Stack multiple data stories

Write about more measures and dimensions by creating multiple **Data Stories** and stacking them vertically or horizontally on your dashboard.

For example, if you wanted to create a story about actual revenue vs benchmark revenue, you could create two different stories—one with the actual revenue and the first benchmark, and another with the actual revenue and the second benchmark—and compare them.

Add a Pop-Up Tableau Data Story to Your Dashboard

Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

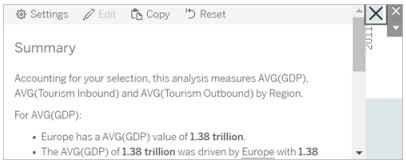
You can place a Tableau Data Story in a pop-up window that allows your users to open a story, read it, and then close it when they're done. This is a great way to save space being used by already-established dashboards or to reduce the amount of clutter and information on a dashboard.

1. Add a Tableau Data Story to a Dashboard.
2. Set the container to **Floating** by clicking the menu and selecting **Floating**.

Tip: Another way to set the container to **Floating** is by holding the shift key while dragging the container onto your dashboard.

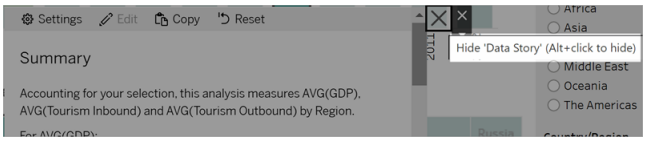
3. Navigate to the **Layout** tab in the left-hand column and set the background color to white.

4. Click the menu that is associated with your floating container to show its settings, and select **Add Show/Hide** Button. This creates an "X" icon that allows you to show or hide your story.



Note: If your story is selected, then your button is partially obscured by the sidebar options that are part of the Data Story object. Click anywhere outside of the story to reveal the button.

5. Hover over the "X" icon to show instructions for opening or closing the story. In this example, you're being prompted to press the Alt key at the same time you click the "X" icon.



6. Collapse the story by clicking the "X" icon at the same time as pressing the key indicated in your prompt.

The story collapses, but the menu remains on the dashboard so that the user can expand the story when needed. You can move your collapsible, floating story around your dashboard as desired.

Create Custom Measure Relationships in Your Tableau Data Story

Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

You can create a relationship story in your Tableau Data Story to see how data compares to another set of data. To build a relationship story, you must have at least two measures and one dimension. Data that you'd like to compare is often grouped into one column of data named something such as "Year" or "Month," with values such as "2022" or "March."

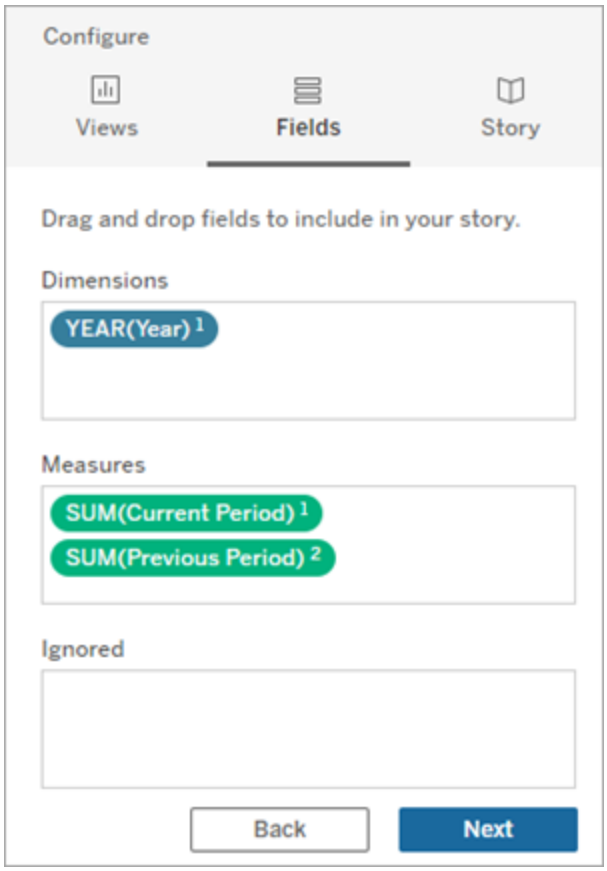
You can Create a Simple Calculated Field to separate "2022" from "2021" (or March from February), so that you can compare the two time periods in a relationship story.

1. Start in the sheet that you want to use in your **Data Story**.
2. Click **Analysis**, and select **Create a Calculated Field**.
3. Create a calculated field such as "Current Period."

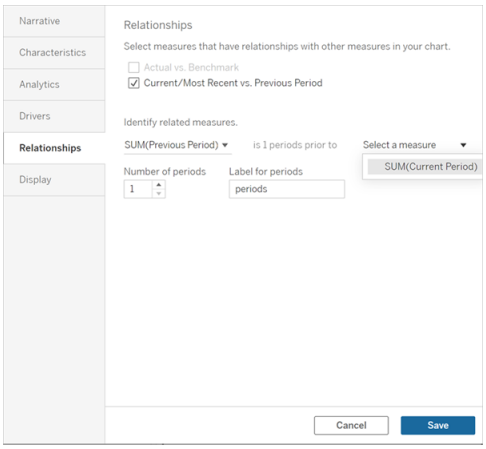
Tip: Follow the structure of the calculated field shown here, but substitute your own dimension or measure names (orange text).

4. Create a calculated field such as "Previous Period."

5. Drag the new measures onto the **Detail** mark.
6. From your dashboard, click the story and add the two new measures into your **Data Story**.



7. From your dashboard, click the **Settings** icon at the top-left corner of your Data Story object.
8. In the Data Story dialog box, click the **Relationships** tab.
9. Set up a relationship story with the two custom measures.



10. Click **Save**.

Your story now writes sentences that compare the custom measures.

Refresh Parameters in a Tableau Data Story

Important changes for Tableau Data Stories

Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

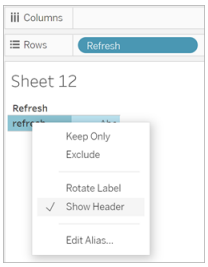
If you've added a parameter to your dashboard and are using Data Stories, you'll notice that clicking the parameter refreshes your visualization, but not the story. This happens because the parameter doesn't refresh the underlying data like a filter does.

To refresh your Tableau Data Story with the parameter data, add a "refresh" button to your dashboard that updates your story to align with your parameter.

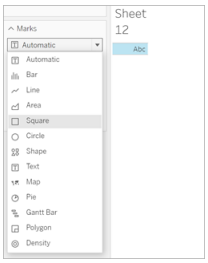
1. Create a new sheet in your workbook.
2. Create a **Calculated Field** in the new sheet with the following info:

   Name: Refresh

3. Click **OK**, and then drag the new calculated field (Refresh) onto your new sheet.
4. Right-click the field and click **Show Header** to hide the header.



5. Choose a shape for your button.



6. Return to your dashboard and drag the sheet containing the refresh button onto your dashboard next to the parameter.
7. Hover over the button, select **More Options,** and click **Title** to hide the title.
8. Hover over the button and click **Use as Filter**.

9. Adjust your parameter and then click on the new refresh button. The button updates your story to align with the parameter.

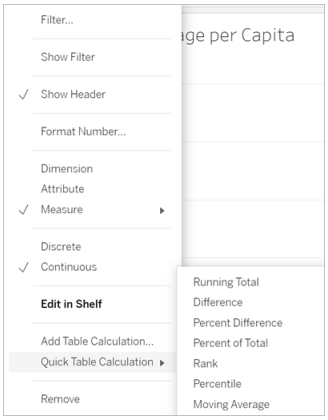
Use a Table Calculation in a Tableau Data Story

Important changes for Tableau Data Stories

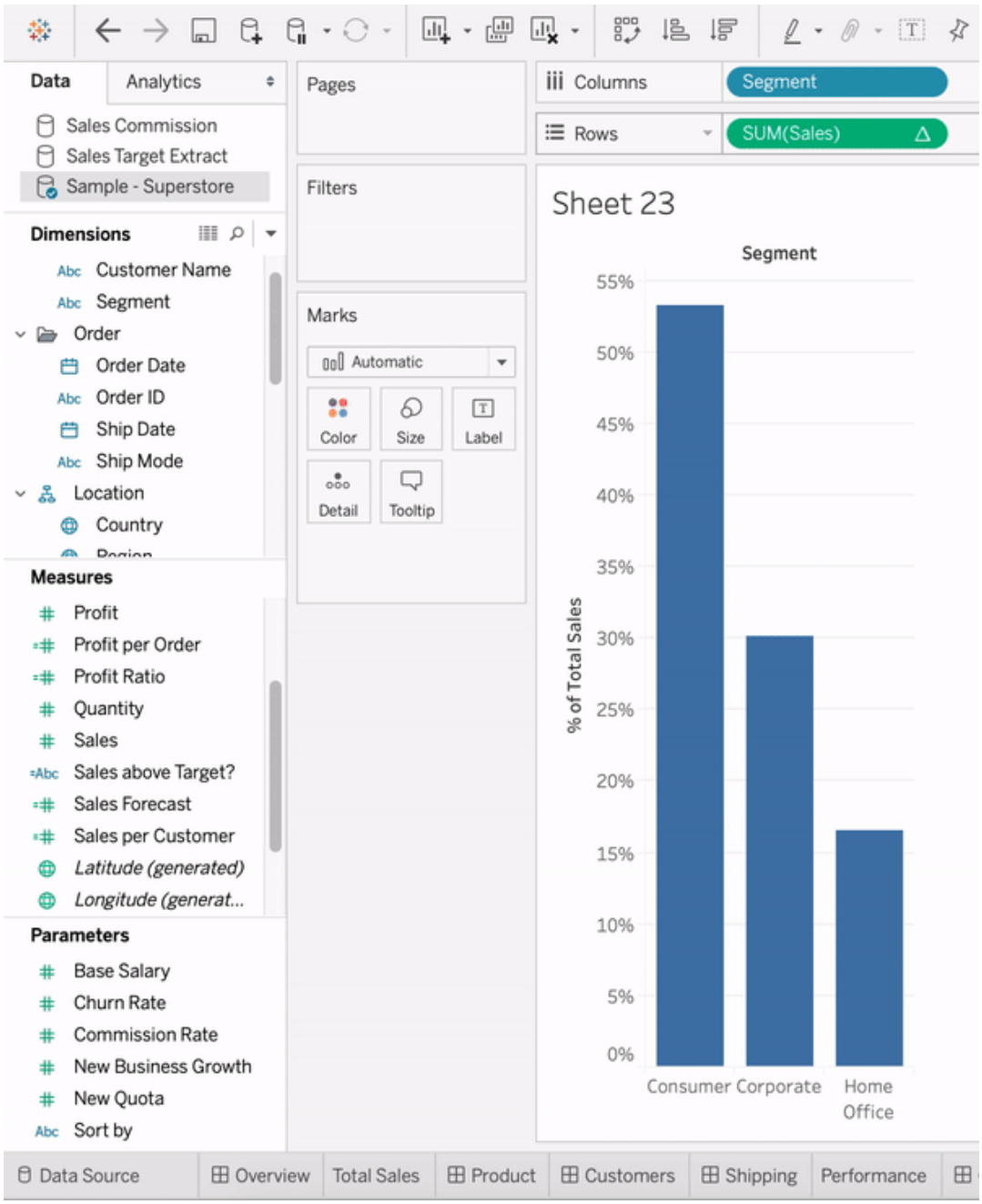
Tableau Data Stories will be retired in Tableau Desktop, Tableau Cloud, and Tableau Server in January of 2025 (2025.1). With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see How Tableau Pulse powered by Tableau AI is Reimagining the Data Experience.

You can use a **Table Calculation** as a measure in a Tableau Data Story in addition to the measure that you used to create the table calculation.

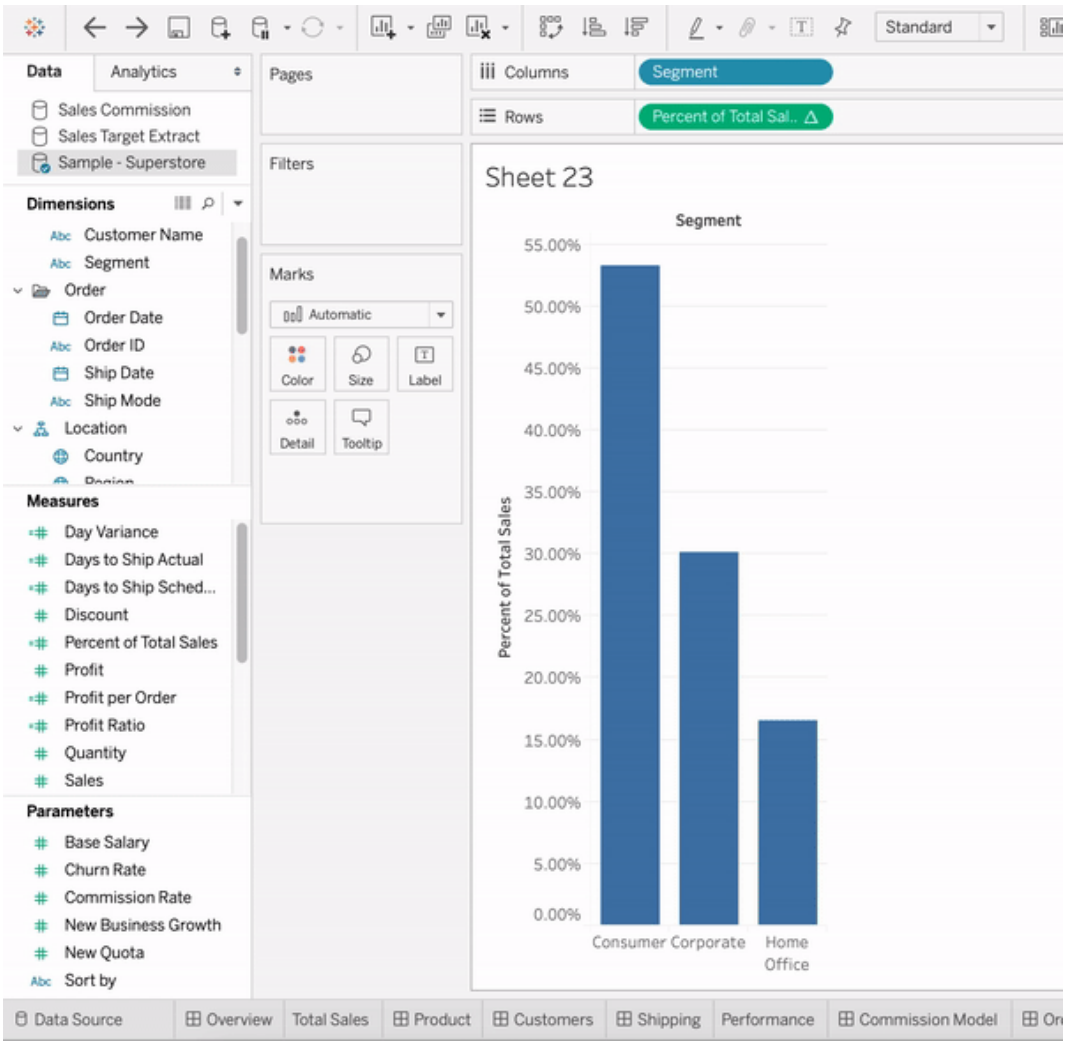
1. From the **Marks** card, click the right side of your field to open a menu, and click **Quick Table Calculation.**



2. After you create your table calculation, drag it to **Measures** in the **Data** pane and rename it.



3. Drag your original measure (e.g., Sales) back to the **Rows** shelf, and then drag your new calculation (e.g., Sum of Sales) into the **Tooltip.** Your visualization may remain the same, but you now have access to the new measure.



4. Go to your dashboard and drag the **Data Story** object onto the dashboard. Both measures appear when creating your story.

Discover Insights Faster with Explain Data

Explain Data in the Data Guide helps you to inspect, uncover, and dig deeper into the marks in a viz as you explore your data. You can use Explain Data to analyze dashboards, sheets, or selected marks for possible outliers and correlations in the underlying data. Explain Data builds statistical models and proposes possible explanations for individual marks in a viz, including potentially related data from the data source that isn't used in the current view.

For information on running Explain Data and exploring explanations, see Get Started with Explain Data.

> **Note**: This topic describes how Explain Data works in Tableau 2021.2 and later versions. If you have a previous version of Tableau, read this topic in version 2021.1 of Explain Data help.

As you build different views, use Explain Data as a jumping-off point to help you explore your data more deeply and ask better questions. For more information, see How Explain Data helps to augment your analysis. For information on what characteristics make a data source more interesting for use with Explain Data, see Requirements and Considerations for Using Explain Data.

Access to Explain Data

Explain Data is enabled by default at the site level. Server administrators (Tableau Server) and site administrators (Tableau Cloud) can control whether Explain Data is available for a site. For more information, Disable or Enable Explain Data for a Site.

Authors who can edit workbooks and have the Run Explain Data permission capability for a workbook can run Explain Data in editing mode. All users with the Run Explain Data capability can run Explain Data in viewing mode in Tableau Cloud and Tableau Server.

When allowed by site administrators, explanations can be shared in viewing mode via email or Slack with other Tableau Cloud and Tableau Server users. For more information, see Configure Tableau to allow users to share explanations via email and Slack.

Authors can use Explain Data Settings to control which explanation types are displayed in the Data Guide pane.

For information on controlling access to Explain Data, explanation types, and fields, see Control Access to Explain Data.

How Explain Data helps to augment your analysis

Explain Data is a tool that uncovers and describes relationships in your data. It can't tell you what is causing the relationships or how to interpret the data. **You are the expert on your data.** Your domain knowledge and intuition are key in helping you decide what characteristics might be interesting to explore further using different views. For related information, see How Explain Data Works and Requirements and Considerations for Using Explain Data.

For more information on how Explain Data works and how to use Explain Data to augment your analysis, see these Tableau Conference presentations:

- From Analyst to Statistician: Explain Data in Practice (1 hour)
- Leveraging Explain Data (45 minutes)

Get Started with Explain Data

Use Explain Data in your flow of analysis as you are exploring the marks in a viz. Explain Data runs automatically when the Data Guide pane is open and updates based on the current selection (dashboard, sheet, or mark).

Use Explain Data

- Run Explain Data on a dashboard, sheet, or mark
- Drill into explanations
- View analyzed fields
- Terms and concepts in explanations
- Explanation Types

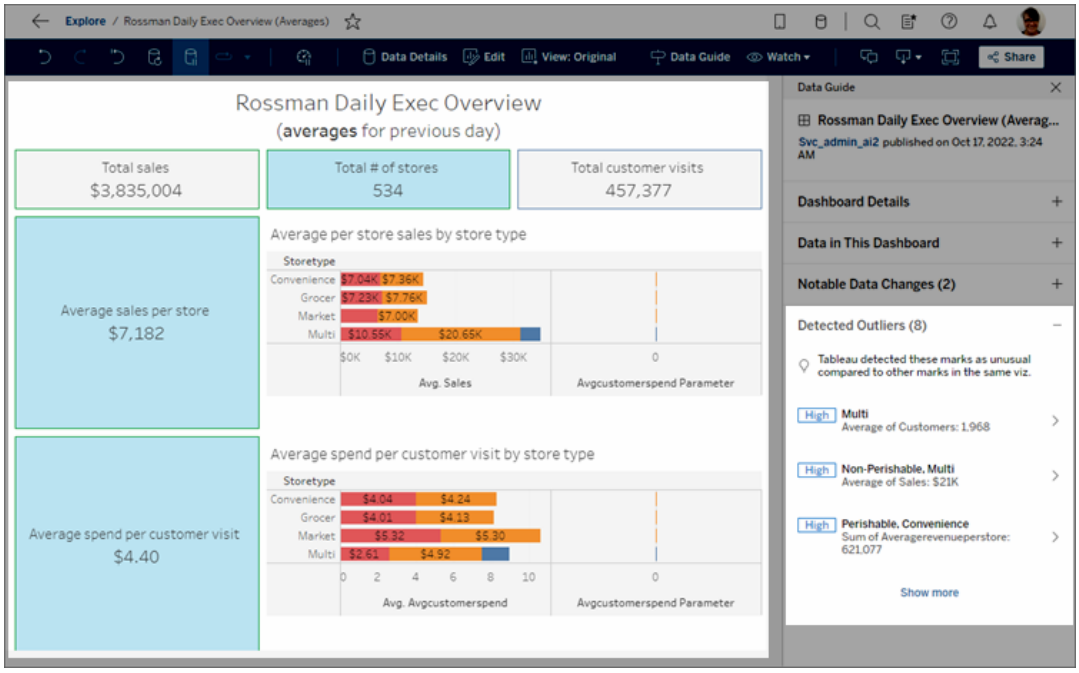
Author Workbooks and Control Access

- Requirements and Considerations for Using Explain Data
- Change Explain Data Settings (Authors-only)
- Control Access to Explain Data
- Disable or Enable Explain Data for a Site
- How Explain Data Works

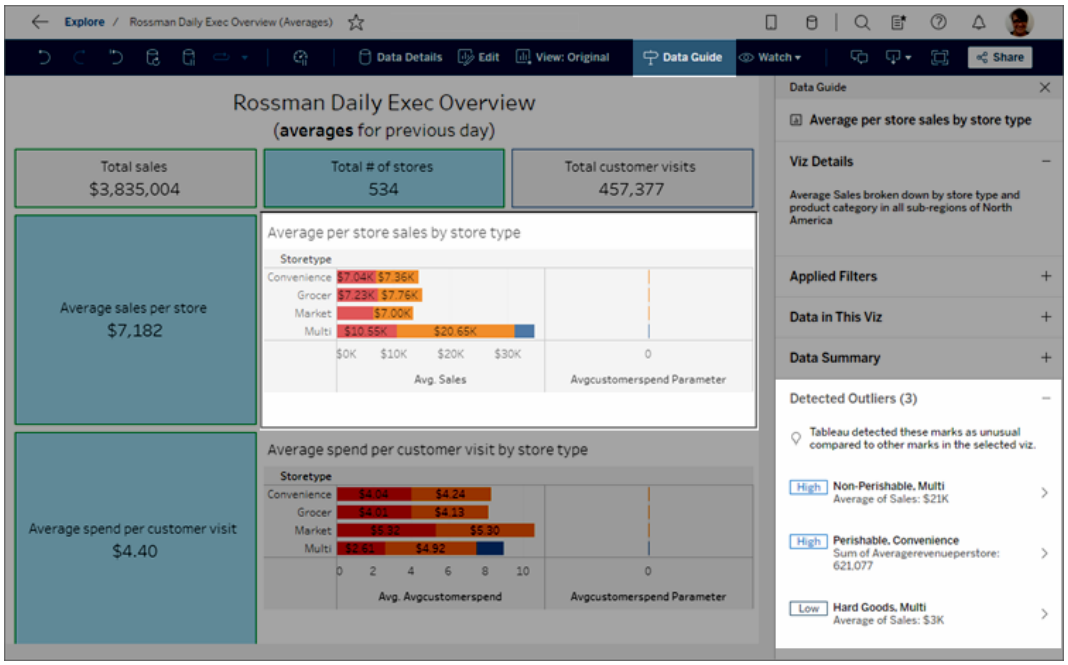
Run Explain Data on a dashboard, sheet, or mark

These are the basic steps to run Explain Data in Tableau Desktop, Tableau Cloud, and Tableau Server:

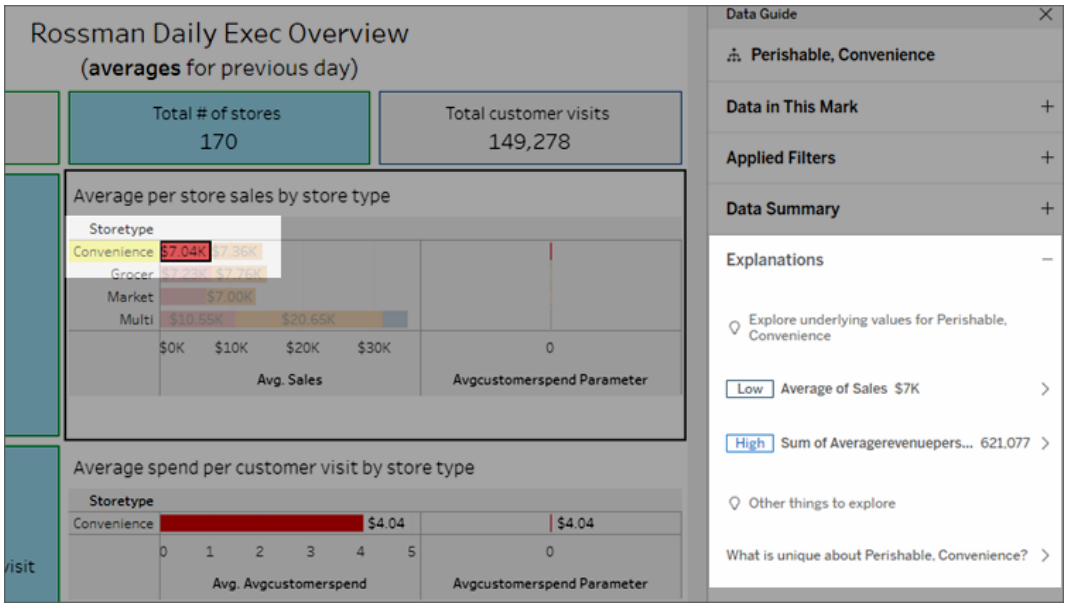
1. Open a dashboard or sheet in a workbook.
2. In the view toolbar, select **Data Guide** to open the Data Guide pane.
3. If you open a dashboard, Explain Data will analyze it for outliers.



If you select a sheet in the dashboard, Explain Data analyzes the marks in that sheet for outliers.



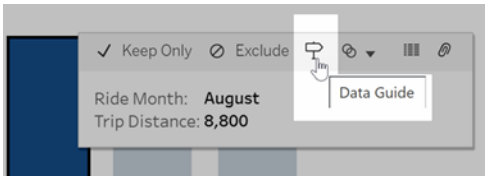
If you select a mark in the dashboard, Explain Data specifically analyzes that mark for explanations.



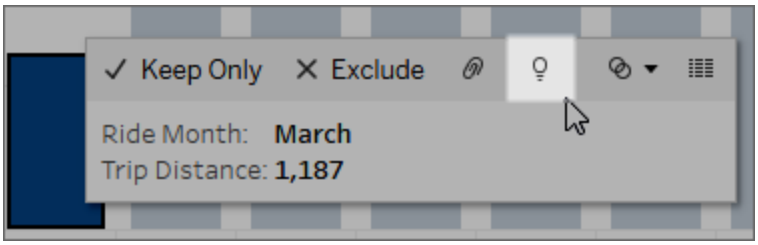
The marks that are being explained are highlighted in the viz as you select

corresponding explanations.

Optionally, you can select a mark in a viz, hover the cursor over the mark. In the tooltip menu, select **Data Guide**.



In Tableau Public, select the lightbulb in the tooltip menu to run Explain Data.



Possible explanations for the value of the analyzed mark are displayed in the Data Guide pane. Select different explanation names to expand the details and start exploring.

Explain Data permissions required for seeing explanations

If you see Detected Outliers with a note to contact the owner of the viz, it is because you need permission to see these types of explanations. Select the owner name to go to their Tableau content page with their email address. Contact the owner to ask them to give you Explain Data permissions for the workbook or view.

If you are the owner of the workbook, for more information on setting permissions, see Control who can use Explain Data and what they can see.

Tips for using Explain Data

- Multiple marks can't be selected for comparison with each other.
- The view must contain marks that are aggregated using SUM, AVG, COUNT, COUNTD, or AGG (a calculated field).
- When Explain Data cannot analyze the type of mark selected, a message is displayed to indicate why. For more information, see Situations where Explain Data is not available.
- The data you analyze must be drawn from a single, primary data source. Explain Data does not work with blended or cube data sources.
- For information on what characteristics make a data source more interesting for use with Explain Data, see Requirements and Considerations for Using Explain Data.

Drill into explanations

1. In the Data Guide pane, select an explanation name to see more details.

   Select an explanation to expand or contract its details.

2. Scroll to see more explanation details.

3. Hover over charts in the explanations to see details on different data points. Select the **Open** icon to see a larger version of the visualization.

   Creators or Explorers who open the view for editing can select the **Open** icon to open the visualization as a new worksheet and explore the data further.

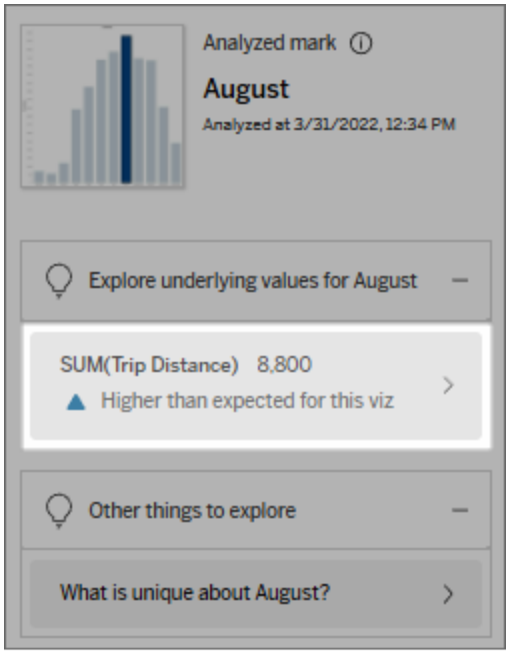
   > **Note**: Creators and Explorers who have editing permissions can also control Explain Data Settings. For more information, see Control Access to Explain Data.

4. Hover over a Help icon to see tooltip help for an explanation. Select the Help icon to keep the tooltip open. Select a **Learn More** link to open the related help topic.

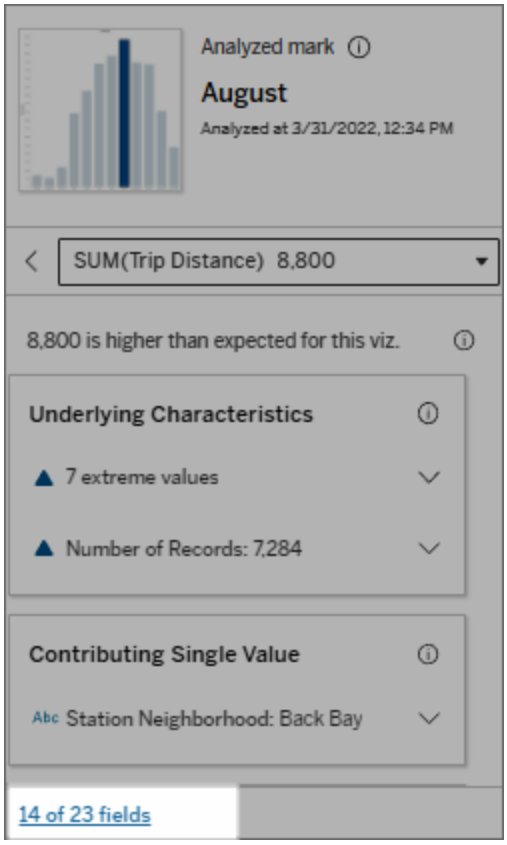
View analyzed fields

1. Run Explain Data on a dashboard, sheet, or mark.

2. In the Data Guide pane, under **Explore underlying values for**, select a target measure name.



3. Select the *number-of-fields* link at the bottom of the pane.



Authors have the option to open Explain Data Settings to control which fields are included in the analysis. For more information, see Change fields used for statistical analysis.

Terms and concepts in explanations

The following terms and concepts appear frequently in explanations. You may find it helpful to become acquainted with their meaning in the context of using Explain Data.

What is a mark?

A mark is a selectable data point that summarizes some underlying record values in your data. A mark can be made of a single record or multiple records aggregated together. Marks

in Tableau can be displayed in many different ways such as lines, shapes, bars, and cell text.

Tableau gets the records that make up the mark based on the intersection of the fields in the view.

The **analyzed mark** refers to a mark in a dashboard or sheet that was analyzed by Explain Data.

For more information on marks, see Marks.

What does expected mean?

The expected value for a mark is the median value in the expected range of values in the underlying data in your viz. The expected range is the range of values between the 15th and 85th percentile that the statistical model predicts for the analyzed mark. Tableau determines the expected range each time it runs a statistical analysis on a selected mark.

If an expected value summary says the mark is *lower than expected* or *higher than expected*, it means the aggregated mark value is outside the range of values that a statistical model is predicting for the mark. If an expected value summary says the mark is *slightly lower* or *slightly higher* than expected or *within the range of natural variation*, it means the aggregated mark value is within the range of predicted mark values, but is lower or higher than the median.

For more information, see What is an expected range?

What are dimensions and measures?

Each column name in a database is a field. For example, Product Name and Sales are each fields. In Tableau, fields like Product Name that categorize data are called dimensions; fields with quantifiable data like Sales are called measures. Tableau aggregates measures by default when you drag them into a view.

Some explanations describe how the underlying record values and the aggregations of those values may be contributing to the value of the analyzed mark. Other explanations may mention the distribution of values across a dimension for the analyzed mark.

When you run Explain Data on mark, the analysis considers dimensions and measures in the data source that aren't represented in the view. These fields are referred to as unvisualized dimensions and unvisualized measures.

For more information on dimensions and measures, see Dimensions and Measures.

What is an aggregate or aggregation?

An aggregate is a value that is a summary or total. Tableau automatically applies aggregations such as SUM or AVG whenever you drag a measure onto Rows, Columns, a Marks card option, or the view. For example, measures are displayed as SUM(Sales) or AVG(Sales) to indicate how the measure is being aggregated.

To use Explain Data, your visualization must use a measure that is aggregated with SUM, AVG, COUNT, COUNTD, or AGG.

For more information about aggregation, see Data Aggregation in Tableau.

What is a record value?

A record is a row in a database table. A row contains values that correspond to each field. In this example, Category, Product Name, and Sales are fields (or columns). Furniture, Floor Lamp, and $96 are the values.

Category	Product Name	Sales
Furniture	Floor Lamp	$96.00

What is a distribution?

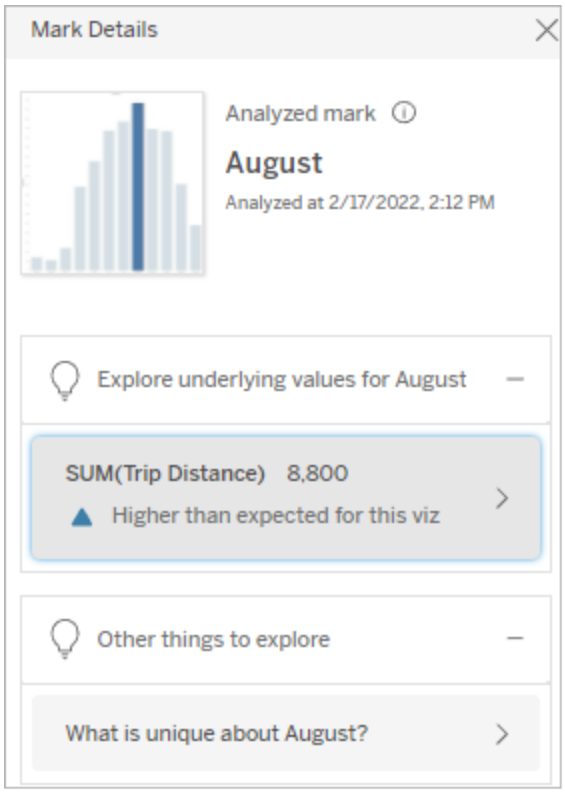
A distribution is a list of all the possible values (or intervals) of the data. It also indicates how often each value occurs (frequency of occurrence).

Explanation Types in Explain Data

Each time you select a new mark in a viz or dashboard and run Explain Data, Tableau runs a new statistical analysis considering that mark and the underlying data in the workbook. Possible explanations are displayed in expandable sections for the Data Guide pane. For information about how Explain Data analyzes and evaluates explanations, see How Explain Data Works.

Explore underlying values

This section lists explanations for each measure that can be explained (referred to as *target measures*). Each explanation listed here describes a relationship with the values of the target measure that are tested on the analyzed mark. Use your real-world, practical understanding of the data to determine if the relationships found by Explain Data are meaningful and worth exploring.



In this example, Trip Distance is the target measure

Underlying Characteristics

These explanations describe how underlying records of the marks in the view may be contributing to the aggregated value of the measure being explained. Mark attributes can include Extreme Values, Null Values, Number of Records, or the Average Value of the mark.

Note: For definitions of common terms used in explanations, see Terms and concepts in explanations.

Extreme Values

This explanation type indicates if one or more records have values that are significantly higher or lower than most records. If the explanation is supported by a model, it indicates the

extreme value is affecting the target measure of the analyzed mark.

When a mark has extreme values, it doesn't automatically mean it has outliers or that you should exclude those records from the view. That choice is up to you depending on your analysis. The explanation is simply pointing out an extreme value in the mark. For example, it could reveal a mistyped value in a record where a banana cost 10 dollars instead of 10 cents. Or, it could reveal that a particular sales person had a great quarter.

Note: This explanation must be enabled by the author to be visible in viewing mode for a published workbook. For more information, see Control Access to Explain Data.

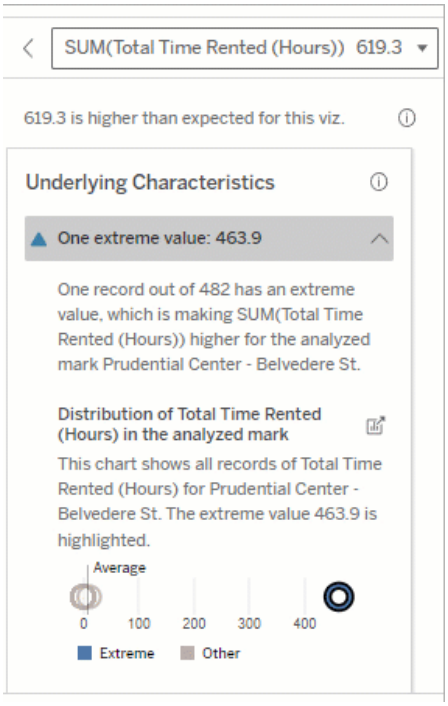
This explanation shows:

- The number of underlying records in the analyzed mark.
- The extreme value or values contributing to the value of the target measure.
- The distribution of values in the mark.
- The record details that correspond to each distribution value.

Exploration options:

- Hover over a circle in the chart to see its corresponding value.
- Select the left or right arrow below the details list to scroll through record details.
- If available, select **View Full Data**, and then select the **Full Data** tab to see all records in a table.
- Select the **Open** icon to see a larger version of the visualization.

Next steps for analysis:



In this example, a single extreme value of 463 hours rented is contributing to the higher than expected sum of Total Time Rented of 613 hours.

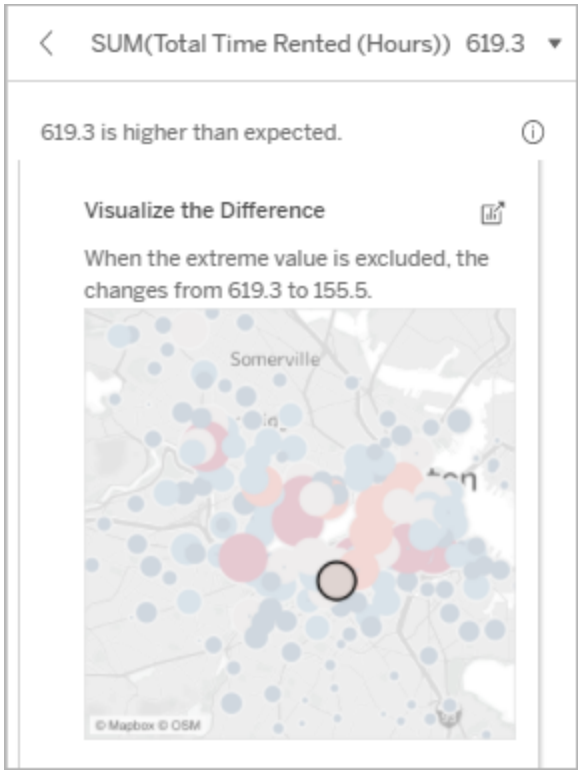
- If the number of records is low, examine these values compared to the extreme value.
- If the extreme value is significantly higher or lower than the other record values, exclude it and consider how it changes the value of the analyzed mark.
- When considering the data with and without the extreme value, use this as an opportunity to apply your practical knowledge about the data.

A likely reason for this high value could be that someone forgot to dock the bike when they returned it. In this case, the author might want to exclude this value for future analysis.

Visualize the Difference

This section shows:

- How the analyzed mark value changes when the extreme value is excluded.



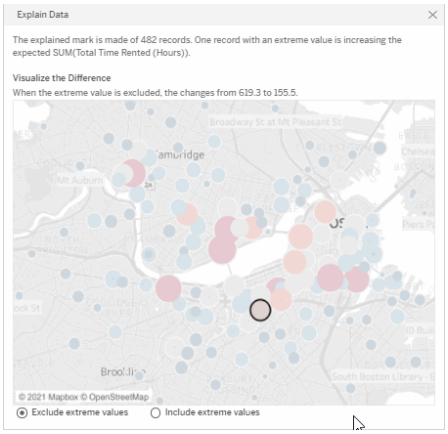
Exploration options:

- Select the **Open** icon to see a larger version of the visualization.

- Explore the difference with and without the extreme value (or values).
- Authors can open the view as a new sheet and apply a filter to exclude the extreme value.

Next steps for analysis:

- If the extreme value is significantly higher or lower than the other record values, exclude it and see how it changes the value of the analyzed mark.
- When considering the data with and without the extreme value, use this as an opportunity to apply your practical knowledge about the data.



In this example, when the extreme value of 483 is excluded, the analyzed mark is no longer high compared to other marks in the view. Other marks now stand out. The author might want to explore the other marks to consider why these other locations have higher hours for bike rentals.

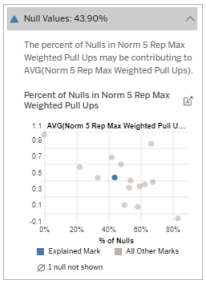
Null Values

The Null Values explanation type calls out situations where there is a higher than expected amount of missing data in a mark. It indicates the fraction of target measure values that are null and how the null values might be contributing to the aggregate value of that measure.

This explanation shows:

- The percent of values that are null in the target measure for the analyzed mark (blue circle).

Exploration options:



- Hover over each circle in the scatter plot to see its details.
- Scroll to see more of the chart.
- Select the **Open** icon to see a larger version of the visualization.

In this example, the percent of null values in the target measure is shown as a blue circle.

Next steps for analysis:

- Optionally exclude null values in the mark for further analysis.

Number of Records

This explanation type describes when the count of the underlying records is correlated to the sum. The analysis found a relationship between the number of records that are being aggregated in a mark and the mark's actual value.

While this might seem obvious, this explanation type helps you explore whether the mark's value is being affected by the magnitude of the values in its records or simply because of the number of records in the analyzed mark.

This explanation shows:

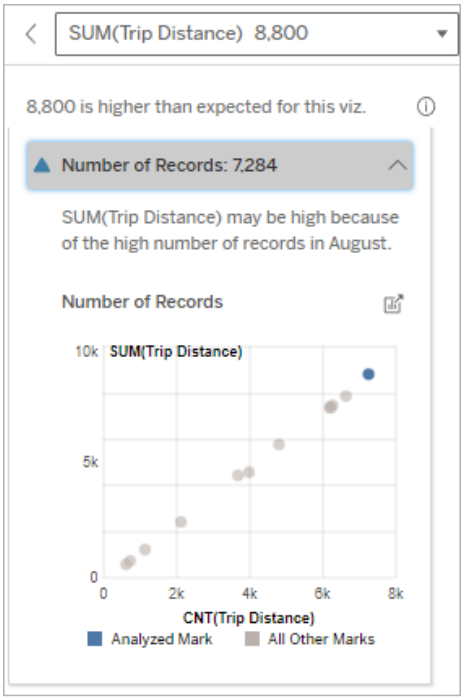
- The number of records in the target measure for the analyzed mark (dark blue bar).
- The number of records in the target measure for other marks in the source visualization (light blue bar).

Exploration options:

- Hover over each bar to see its details.
- Scroll to see more of the chart.
- Select the **Open** icon to see a larger version of the visualization.

Next steps for analysis:

- Compare whether the individual values of records are low or high, or the number of records in the analyzed mark is low or high.
- Authors, if you are surprised by a high number of records, you might need to normalize the data.



In this example, the number of records for Trip Distance is listed for each value of Ride Month, which is a dimension in the original visualization. August has the highest total trip distance value.

You might explore whether August has the highest value for trip distance because more rides occurred in August, or if it has the highest trip distance because some rides were longer.

Average Value of Mark

This explanation type describes when the average of a measure is correlated to the sum. Compare whether the average value is low or high, or the number of records is low or high.

This explanation shows:

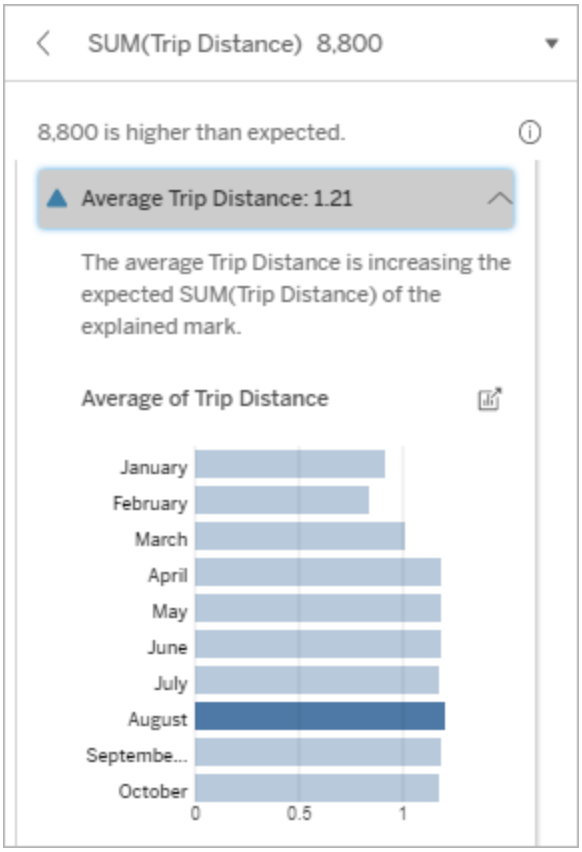
- The average of the target measure for each value of a dimension used in the source visualization.

Exploration options:

- Hover over each bar to see its details.
- Scroll to see more of the chart.
- Select the **Open** icon to see a larger version of the visualization.

Next steps for analysis:

- Compare whether the average value is low or high, or the number of records is low or high. For example, are profits high because you sold a lot of items or because you sold expensive items?
- Try to figure out why the analyzed mark has a significantly higher or lower average value.



In this example, the average trip distance for August is not significantly higher or lower than most months. This suggests that trip distance is higher for August because there were more rides in August, rather than from people taking longer rides.

Contributing Single Value

Use this explanation to understand the composition of the record values that make up the analyzed mark.

This explanation type identifies when a single value in an unvisualized dimension may be contributing to the aggregate value of the analyzed mark. An unvisualized dimension is a dimension that exists in the data source, but isn't currently being used in the view.

This explanation indicates when every underlying record of a dimension has the same value, or when a dimension value stands out because either many or few of the records have the same single value for the analyzed mark.

Note: For definitions of common terms used in explanations, see Terms and concepts in explanations.

This explanation shows:

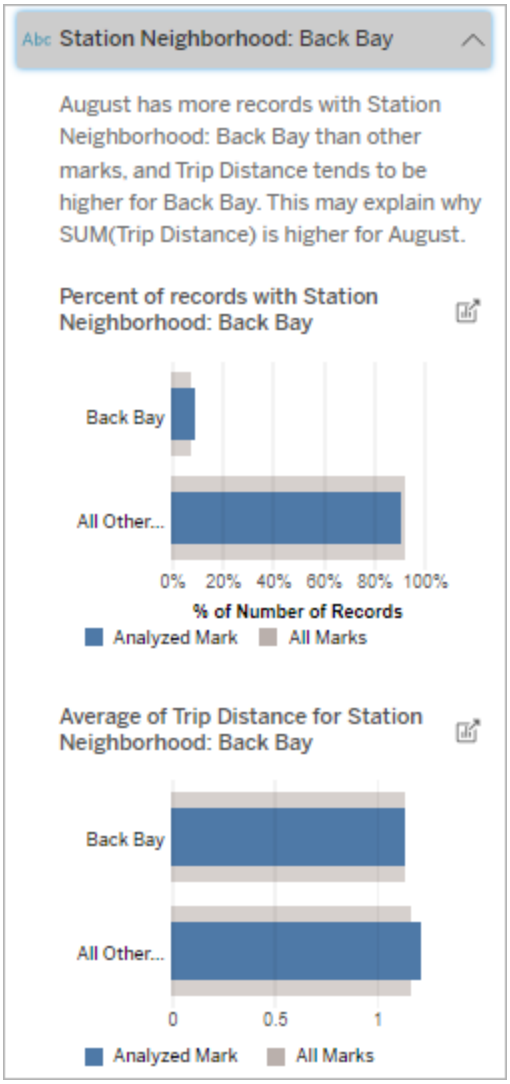
- The percent of the number of records for a single value of a dimension for the analyzed mark (blue bar) versus all marks (gray bar) in the source visualization.
- The percent of the number of records for all other values of a dimension for the analyzed mark (blue bar) versus all marks (gray bar) in the source visualization.
- The average of the target measure for the single value of a dimension in the analyzed mark (blue bar) versus all marks (gray bar).
- The average of the target measure for all other values of a dimension for the analyzed mark (blue bar) versus all marks (gray bar) in the source visualization.

Exploration options:

- Hover over each bar to see its details.
- Select the **Open** icon to see a larger version of the visualization.

Next steps for analysis:

- Use this explanation to understand the composition of the record values that make up the analyzed mark.
- Authors might want to create a new visualization to explore any unvisualized dimension surfaced in this explanation.



In this example, the statistical analysis has exposed that many of the rides come from the station neighborhood of Back Bay. Note that Station Neighborhood is an unvisualized dimension that has some relationship to Trip Distance in the underlying data for the source visualization.

Top Contributors

Use this explanation to see the values that make up the largest fraction of the analyzed mark.

For a COUNT aggregation, the top contributors show dimension values with the most records. For SUM, this explanation shows dimension values with the largest partial sum.

Contributing Dimensions

Use this explanation to understand the composition of the record values that make up the analyzed mark.

This explanation type shows that the distribution of an unvisualized dimension may be contributing to the aggregate value of the analyzed mark. This type of explanation is used for target measure sums, counts, and averages. An unvisualized dimension is a dimension that exists in the data source, but isn't currently being used in the view.

Note: For definitions of common terms used in explanations, see Terms and concepts in explanations.

This explanation shows:

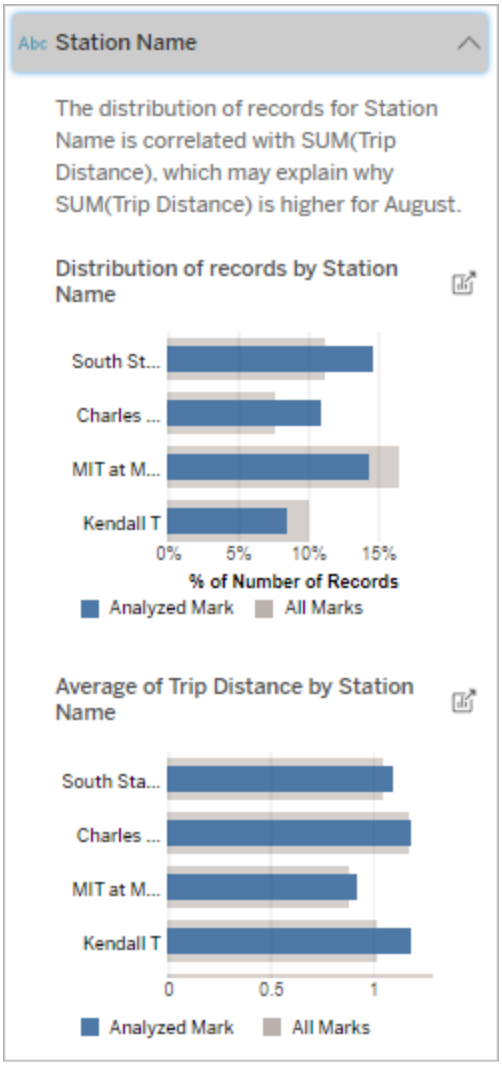
- The percent of the number of records for all values of a dimension for the analyzed mark (blue bar) versus all values of a dimension for all marks (gray bar) in the source visualization.
- The average of the target measure for all values of a dimension for the analyzed mark (blue bar) all values of a dimension for all marks (gray bar).

Exploration options:

- Hover over each bar to see its details.
- Scroll to see more of the chart.
- Select the **Open** icon to see a larger version of the visualization.

Next steps for analysis:

- Use this explanation to understand the composition of the record values that make up the analyzed mark.
- Authors might want to create a new visualization to explore any unvisualized dimensions surfaced in this explanation.



In this example, the statistical analysis has exposed that more rides were taken from South Station and MIT and fewer rides were taken from Charles Circle and Kendall, compared to rides taken for marks overall.

Note that Station Name is an unvisu-

	alized dimension that has some relationship to Trip Distance in the underlying data for the source visualization.

Contributing Measures

This explanation type shows that the average of an unvisualized measure may be contributing to the aggregate value of the analyzed mark. An unvisualized measure is a measure that exists in the data source, but isn't currently being used in the view.

This explanation can reveal a linear or quadratic relationship between the unvisualized measure and the target measure.

Note: For definitions of common terms used in explanations, see Terms and concepts in explanations.

This explanation shows:

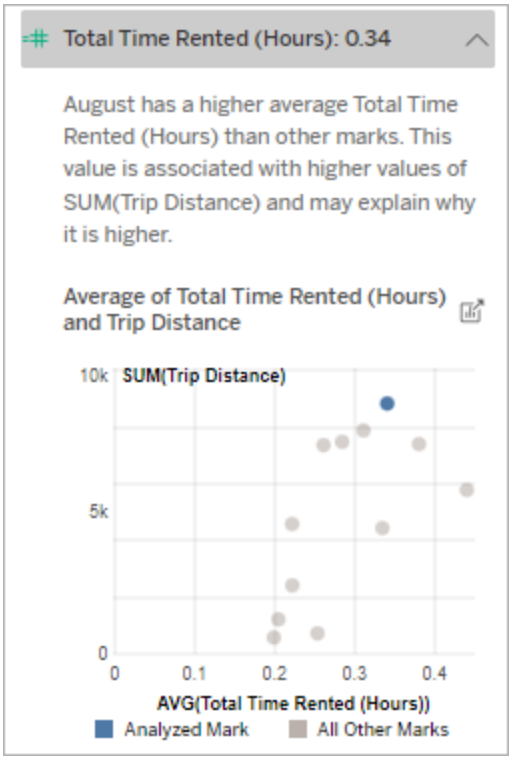
- The relationship between the sum of the target measure and the average of an unvisualized measure for the analyzed mark (blue circle) and all marks (gray circles) in the view.
- If the sum of the target measure is high or low because the average value of the unvisualized measure is high or low.

Exploration options:

- Hover over each circle to see its details.
- Select the **Open** icon to see a larger version of the visualization.

Next steps for analysis:

- Authors might want to create a new visualization to explore any unvisualized measures surfaced in this explanation.



In this example, one possible reason why trip distance is high is because the average total time rented is also high.

Other things to explore

This section provides possible reasons why the analyzed mark is unique or unusual. These explanations:

- Do not explain why the value of this mark is what it is.
- Are not related in any way to the value of the measures in the source visualization.
- Do not take any target measures into account.

Other Dimensions of Interest

Use this explanation to understand the composition of the record values that make up the analyzed mark.

The distribution of an unvisualized dimension in the analyzed mark is unusual compared to the distribution of values for all other marks in the view. An unvisualized dimension is a dimension that exists in the data source, but isn't currently being used in the view.

Note: For definitions of common terms used in explanations, see Terms and concepts in explanations.

This explanation shows:

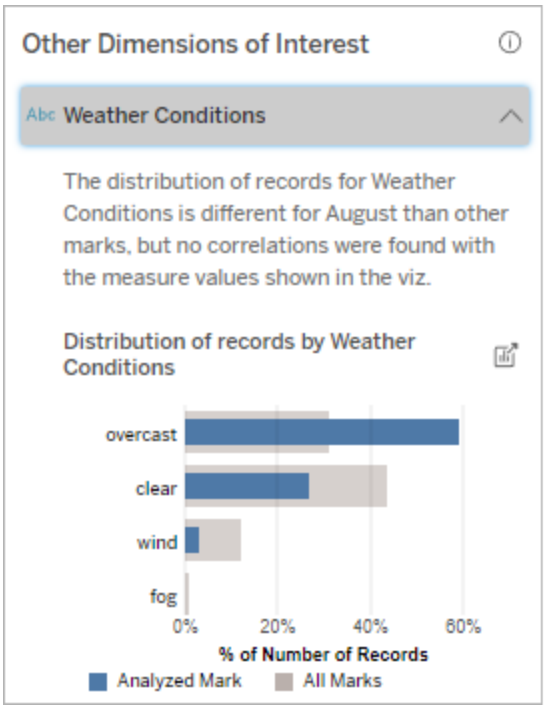
- The percent of the number of records for all values of a dimension for the analyzed mark (blue bar) versus all values of a dimension for all marks (gray bar) in the source visualization.

Exploration options:

- Hover over each bar to see its details.
- Scroll to see more of the chart.
- Select the **Open** icon to see a larger version of the visualization.

Next steps for analysis:

- Use this explanation to understand the composition of the record values that make up the analyzed mark.
- Authors might want to create a new visualization to explore any unvisualized dimensions surfaced in this explanation.



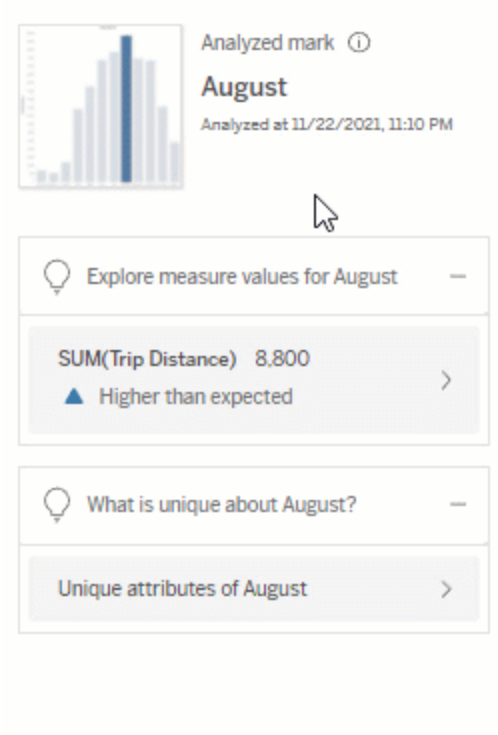
In this example, a high percentage of records are associated with overcast weather. Because the data is about bike rentals in Boston, and the analyzed mark is Trip Distance for

August, we can assume that the weather is typically warm and humid. People might have rented bikes more often on overcast days to avoid the heat. It's also possible there were more overcast days in August.

Analyzed Fields in Explain Data

Explain Data runs a statistical analysis on a dashboard or sheet to find marks that are outliers, or specifically on a mark you select. The analysis also considers possibly related data points from the data source that aren't represented in the current view.

Explain Data might not include every column from the data source in the analysis. In many cases, certain types of fields will be automatically excluded from the analysis. For more information, see Fields excluded by default.

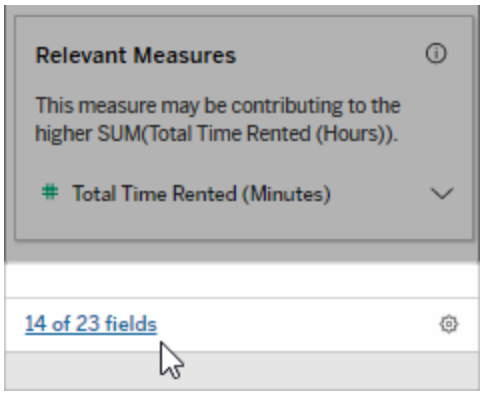


Note: Dimensions with more than 500 unique values won't be considered for analysis (unless allowed by the author in Explain Data Settings).

All users can view information on which fields are included or excluded in the current analysis. Creators and Explorers who have editing permissions can edit the fields used by Explain Data for statistical analysis.

View fields analyzed by Explain Data

When you expand an explanation for a measure that is contributing to the value of the mark, a link that indicates the number of fields considered in the analysis is displayed at the bottom of the Data Guide pane.

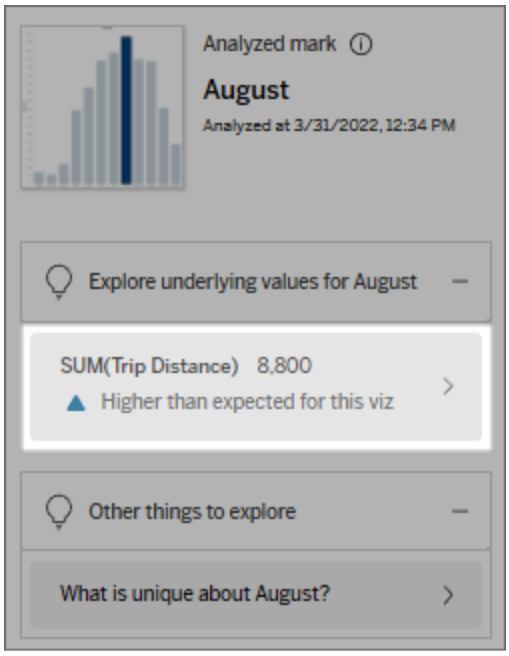


Click the link to see the list of fields included in or excluded from the current statistical analysis.

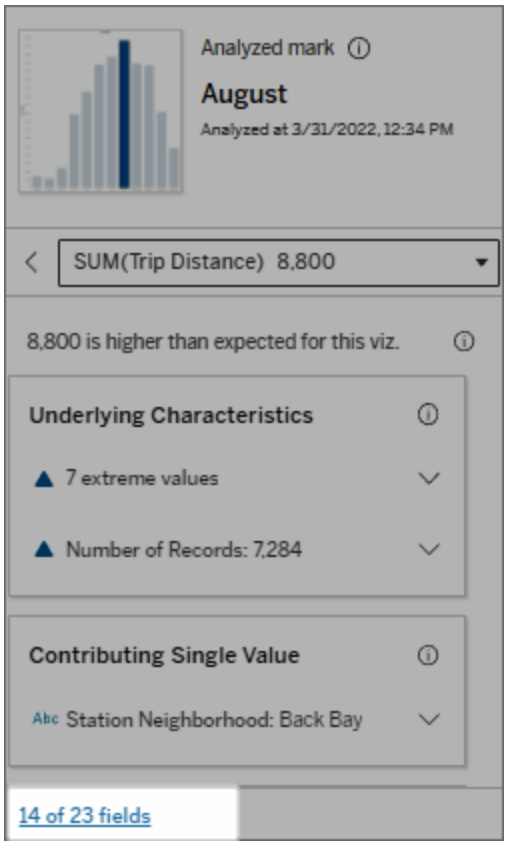
When a data source contains more than 1000 unvisualized dimensions or measures, you might see an alert asking if you want Explain Data to consider more fields. Click **Explain All** to run an analysis that includes more fields. The analysis may take longer to complete.

To view fields used by Explain Data for statistical analysis

1. Run Explain Data on dashboard, sheet, or mark.
2. In the Data Guide pane, under **Contributing to the value of**, click a measure name.

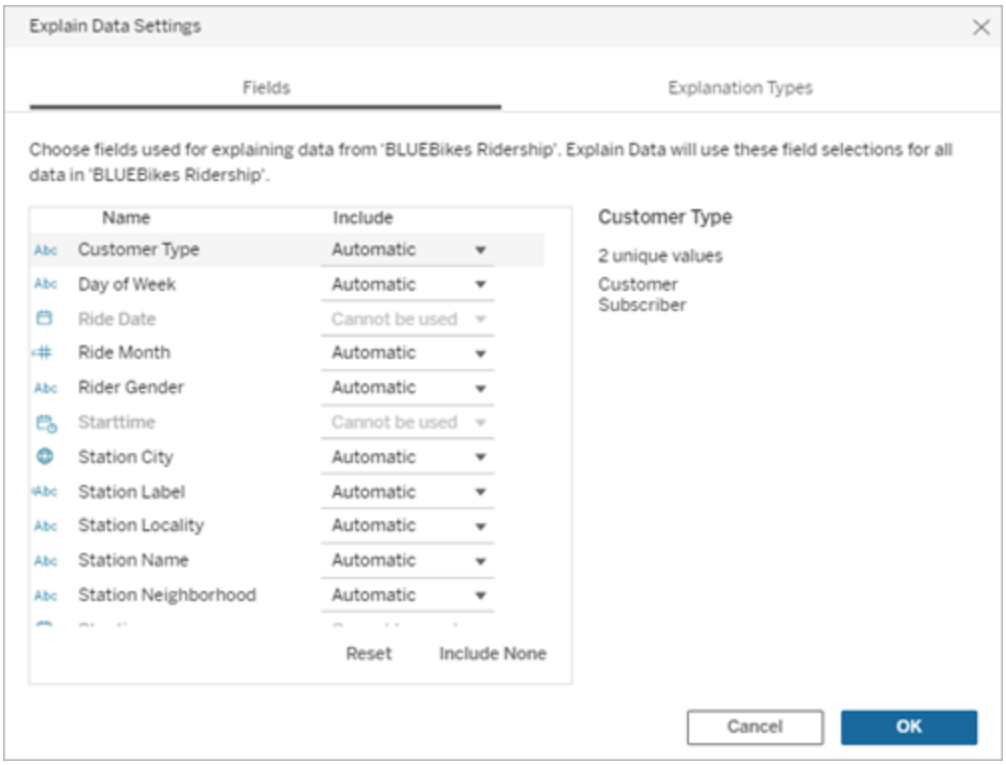


3. Click the *number-of-fields* link at the bottom of the pane.



Change fields used for statistical analysis

Creators and Explorers who have editing permissions can select fields to be included or excluded from the statistical analysis in the Fields tab of the Explain Data Settings dialog box.

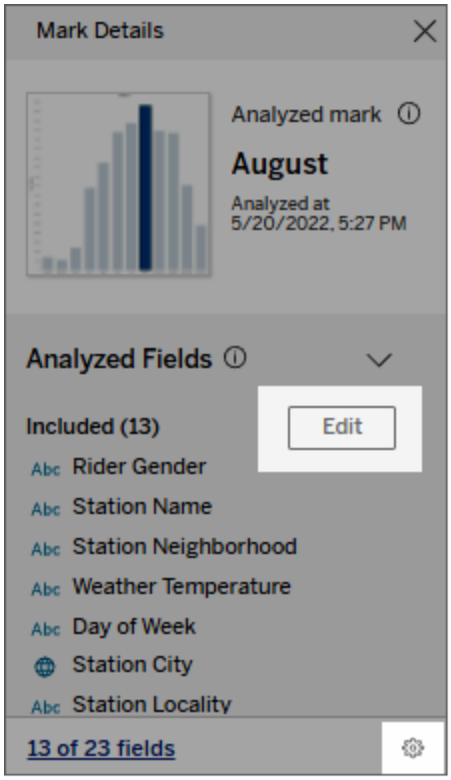


When a data source contains dimensions with a large number of unique values (up to 500), those fields won't be considered for analysis.

To edit the fields used by Explain Data for statistical analysis

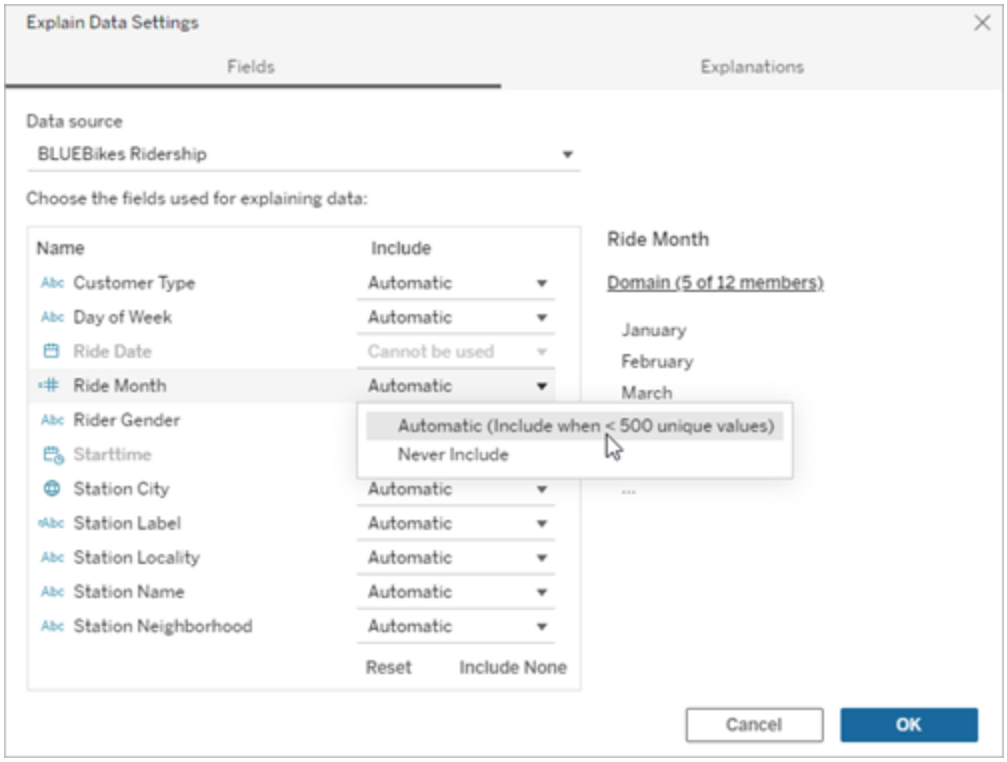
Settings for analyzed fields are applied at the data source level.

1. Run Explain Data on a mark when editing a view.
2. In the Data Guide pane, click the settings icon at the bottom of the pane. Or, click the **Edit** button in the Analyzed Fields view (how to open analyzed fields).



3. In the Explain Data Settings dialog box, click the **Fields** tab.
4. Click a drop-down arrow next to a field name, select **Automatic** or **Never Include**, and then click **OK**.

   Note that fields must have less than 500 unique values to be included in the analysis.



Fields excluded by default

Fields excluded by default	Reasons for exclusion
All unvisualized measures when there are more than 1,000 measures in the data source.	Computing explanations for more than 1000 unvisualized measures or dimensions can take longer to compute, sometimes several minutes. These fields are excluded by default for initial analysis, but you can choose to include them for further analysis.
All unvisualized dimensions when there are more than 1,000 dimensions in the data source.	In this situation, you might see an alert asking if you want Explain Data to consider more fields. Click the alert link to get more information. Click **Explain All** to run an analysis that includes more fields.

Fields excluded by default	Reasons for exclusion
Fields that use geometry, latitude, or longitude	Geometry, latitude, or longitude by themselves can never be explanations. It is highly likely that an explanation that calls out the latitude or the longitude as an explanation is due to a spurious correlation and not a probable explanation.
Dimensions with high cardinality (dimensions with > 500 members)	High cardinality dimensions take longer to compute. Dimensions with more than 500 unique values will not be considered for analysis.
Groups, bins, or sets	Not currently supported.
Table calculations	Table calculations cannot be analyzed when table calculations are at a different level of detail than the view.
Unvisualized measures that can't be averaged	Unvisualized measures that can't be averaged include measures that are calculated fields where the calculation expression includes aggregations (display as AGG() fields when added to the sheet).
Discrete measures and continuous dimensions	Not currently supported.
Hidden fields	Not available.
Calculated fields with errors	No values present to analyze.

Requirements and Considerations for Using Explain Data

Explain Data is always available to authors in Tableau Desktop.

For Tableau Cloud and Tableau Server: When Explain Data is enabled for a site, Creators and Explorers with the appropriate permissions can run Explain Data when editing a work-

book. All users with the appropriate permissions can run Explain Data in viewing mode in published workbooks. For more information, see Control Access to Explain Data.

What makes a viz a good candidate for Explain Data

Explain Data works best on visualizations that require deeper exploration and analysis, rather than infographic-style, descriptive vizzes that communicate summarized data.

- Row-level data is necessary for Explain Data to create models of your data and generate explanations. Vizzes with underlying, row-level data, where relationships might exist in unvisualized fields are good candidates for running Explain Data.
- Vizzes based on pre-aggregated data without access to row-level data are not ideal for the statistical analysis performed by Explain Data.

What data works best for Explain Data

When you are using Explain Data in a worksheet, remember that Explain Data works with:

- **Single marks only**—Explain Data analyzes single marks. Multiple mark analysis is not supported.
- **Aggregated data**—The view must contain one or more measures that are aggregated using SUM, AVG, COUNT, or COUNTD. At least one dimension must also be present in the view.
- **Single data sources only**—The data must be drawn from a single, primary data source. Explain Data does not work with blended or cube data sources.

When preparing a data source for a workbook, keep the following considerations in mind if you plan to use Explain Data during analysis.

- Use a data source with underlying data that is sufficiently wide. An ideal data set has at least 10-20 columns in addition to one (or more) aggregated measures to be explained.
- Give columns (fields) names that are easy to understand.
- Eliminate redundant columns and data prep artifacts. For more information, see Change fields used for statistical analysis.
- Don't discard unvisualized columns in the data source. Explain Data considers fields in the underlying data when is analyzes a mark.

- Low cardinality dimensions work better. The explanation of a categorical dimension is easier to interpret if its cardinality is not too high (< 20 categories). Dimensions with more than 500 unique values will not be considered for analysis.
- Don't pre-aggregate data as a general rule. But if the data source is massive, consider pre-aggregating the data to an appropriate level of detail.
- Use extracts over live data sources. Extracts run faster than live data sources. With live data sources, the process of creating explanations can create many queries (roughly one query per each candidate explanation), which can result in explanations taking longer to be generated.

Situations where Explain Data is not available

Sometimes Explain Data will not be available for a selected mark, depending on the characteristics of the data source or the view. If Explain Data cannot analyze the selected mark, the Explain Data icon and context menu command will not be available.

Explain Data can't be run in views that use:	• Map coordinate filters • Blended data sources • Data sources with parameters • Data sources that don't support COUNTD or COUNT(DISTINCT ...) syntax, such as Access. • Filters on aggregate measures • Disaggregated measures
Explain Data can't be run if you select:	• Multiple marks • Axis • Legend • Grand total • Trend line or reference line • A mark in a view that contains a very low number of marks
Explain Data can't be run when the measure to be used for an explanation:	• Isn't aggregated using SUM, AVG, COUNT, COUNTD • Is a table calculation • Is used in measure values

Explain Data can't offer explanations for a dimension when it is:	• A calculated field • A parameter • Used in Measure Names and Measure Values • A field with more than 500 unique values. Dimensions with more than 500 unique values will not be considered for analysis.

Control Access to Explain Data

Your access to Explain Data will vary depending on your site role and content permissions. Explain Data is always available to authors in Tableau Desktop. Authors with appropriate permissions can run Explain Data in editing mode in Tableau Cloud and Tableau Server.

Authors can also control whether Explain Data is available in viewing mode in published workbooks and which explanation types are displayed.

Be aware that Explain Data can surface values from dimensions and measures in the data source that aren't represented in the view. As an author, you should run Explain Data and test the resulting explanations to make sure that sensitive data isn't being exposed in your published workbooks.

Who can access Explain Data

Explain Data is enabled by default at the site level. Server administrators (Tableau Server) and site administrators (Tableau Cloud) can control whether Explain Data is available for a site. For more information, Disable or Enable Explain Data for a Site.

Mode	Who Can Access
Viewing mode	Tableau **Viewers**, **Explorers**, and **Creators** who have the Run Explain Data permission capability can run and explore Explain Data explanations in viewing mode.
Editing mode	Tableau **Creators** can run Explain Data when editing a view in Tableau Desktop, Tableau Cloud, or Tableau Server. **Explorers** who have the Run Explain Data permission cap-

Mode	Who Can Access
	ability and editing permissions can run Explain Data when editing a workbook in Tableau Cloud or Tableau Server. Creators and Explorers who have editing permissions can open new worksheets for further analysis. They also can use Explain Data Settings to control who can use Explain Data and what they can see.

Control who can use Explain Data and what they can see

A combination of settings must be enabled to make Explain Data available in editing mode and viewing mode in Tableau Cloud and Tableau Server.

Editing mode

Requirements for authors to run Explain Data or edit Explain Data settings in editing mode:

- Site setting: **Availability of Explain Data** set to **Enable**. Enabled by default.
- Site role: Creator, Explorer (can publish)
- Permissions: **Run Explain Data** capability set to **Allowed**. Unspecified by default. If you open a workbook (Tableau version 2022.1 or earlier) that used this permission in Tableau version 2022.2 or later, you will need to reset the Run Explain Data capability to Allowed.

Note: The **Download Full Data** capability for a Creator or Explorer (can publish) controls whether they see the View Full Data option in Extreme Values explanations. Viewers are always denied the Download Full Data capability. However, all users can see record-level details when the Extreme Values explanation type is enabled in Explain Data settings.

Creators and Explorers with editing permissions and the Run Explain Data permission capability can access **Explain Data Settings**, which provide options for controlling:

- The explanation types that are displayed in the Data Guide pane.
- The fields that are included in, or excluded from statistical analysis.

These options are set for the entire workbook and can only be set in the Explain Data Settings dialog box.

Viewing mode

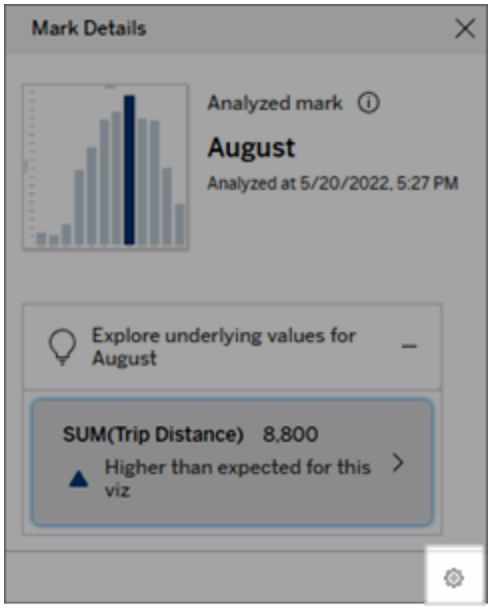
Requirements for all users to run Explain Data in viewing mode:

- Site setting: **Availability of Explain Data** set to **Enable**. Enabled by default.
- Site role: Creator, Explorer, or Viewer
- Permissions: **Run Explain Data** capability set to **Allowed**. Unspecified by default. If you open a workbook (Tableau version 2022.1 or earlier) that used this permission in Tableau version 2022.2 or later, you will need to reset the Run Explain Data capability to Allowed.

Note: To see explanations of Detected Outliers in the Data Guide, users of a viz must have the Explain Data permission allowed for the workbook or view. The owner of the workbook will need to open the permissions settings for this workbook in Tableau Server or Tableau Cloud and allow the Explain Data permission to that user.

Open the Explain Data Settings dialog box

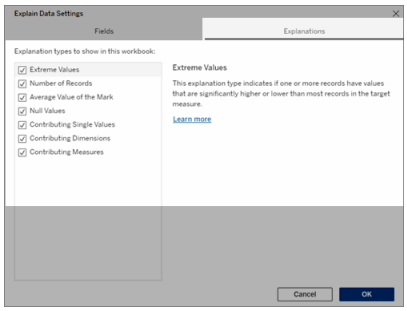
1. From the **Analysis** menu, choose **Explain Data Settings**. Or, in the Data Guide pane, click the settings icon (bottom right).



Include or exclude explanation types displayed by Explain Data

Creators and Explorers who have editing permissions can choose to exclude (or include) explanation types displayed for all workbook users.

1. In the Explain Data Settings dialog box, click the **Explanation Types** tab.



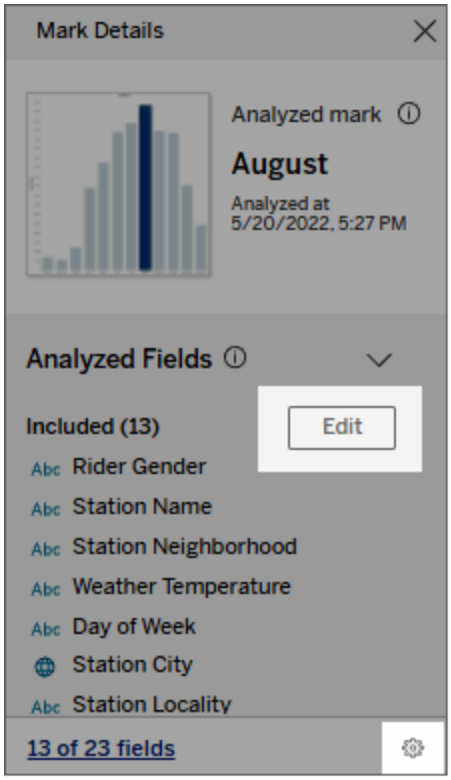
2. In the list of explanation types, select or clear an explanation type.
3. Click **OK**.

Test the setting by saving and closing the published workbook, and then opening a view from the workbook in viewing mode. Select a mark that typically has Extreme Value explanations, and then run Explain Data to check the explanation results.

Include or exclude fields used for statistical analysis

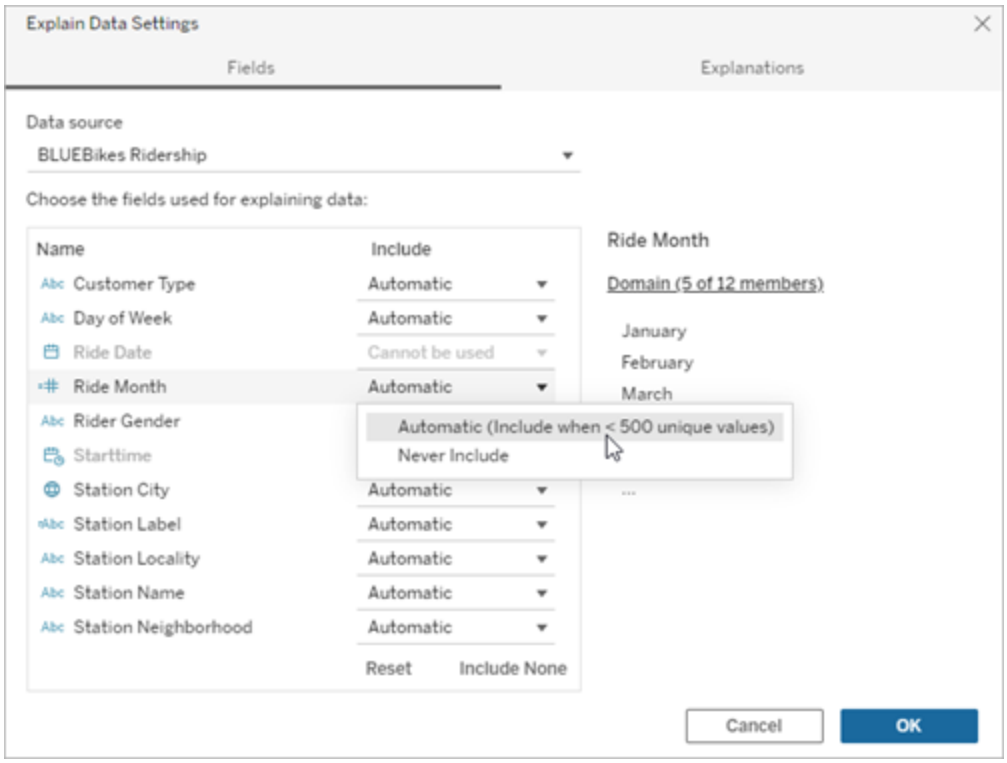
Creators or Explorers who have editing permissions can choose to exclude (or include) fields that are eligible for analysis.

1. In the Data Guide pane (bottom right), choose the settings icon. Or, choose the **Edit** button in the Analyzed Fields view.



2. In the Explain Data Settings dialog box, click the **Fields** tab.
3. In the list of fields under **Include**, click the drop-down arrow and select **Automatic** to include an eligible field every time Explain Data runs for that workbook.

   Note that fields must have less than 500 unique values to be included in the analysis.



Select **Never Include** to explicitly exclude the field.
Select **Include None** to run a statistical analysis on the data without considering fields.
Select **Reset** to return to the default settings.

4. Click **OK**.

Test the setting by saving the published workbook. Select a mark, and then run Explain Data to check the explanation results.

Configure Tableau to allow users to share explanations via email and Slack

Tableau administrators can control whether explanations can be shared in viewing mode via email or Slack to other Tableau users.

Follow these steps to allow notifications and sharing via email and Slack in Tableau Cloud or Tableau Server:

1. Click **Settings**.
2. On the General tab, scroll down to **Manage Notifications**.
3. For Collaboration, select **Share** for **On Tableau**, **Email**, and **Slack**.

To share explanations via Slack, the Tableau app must be set up for your Slack workspace. Sharing explanations with Slack is enabled by default in Tableau Cloud.

In Tableau Server, an administrator will need to set up the Tableau app for Slack. For more information, see Integrate Tableau with a Slack Workspace.

How Explain Data Works

Use Explain Data as an incremental, jumping-off point for further exploration of your data. The possible explanations that it generates help you to see the different values that make up or relate to an analyzed mark in a view. It can tell you about the characteristics of the data points in the data source, and how the data might be related (correlations) using statistical modeling. These explanations give you another tool for inspecting your data and finding interesting clues about what to explore next.

Note: Explain Data is a tool that uncovers and describes relationships in your data. It can't tell you what is causing the relationships or how to interpret the data. **You are the expert on your data.** Your domain knowledge and intuition is key in helping you decide what characteristics might be interesting to explore further using different views.

For related information on how Explain Data works, and how to use Explain Data to augment your analysis, see these Tableau Conference presentations:

- From Analyst to Statistician: Explain Data in Practice (1 hour)
- Leveraging Explain Data (45 minutes)

What Explain Data is (and isn’t)

Explain Data is:

- A tool and a workflow that leverages your domain expertise.
- A tool that surfaces relationships in your data and recommends where to look next.
- A tool and a workflow that helps expedite data analysis and make data analysis more accessible to a broader range of users.

Explain Data is not:

- A statistical testing tool.
- A tool to prove or disprove hypotheses.
- A tool that is giving you an answer or telling you anything about causality in your data.

When running Explain Data on marks, keep the following points in mind:

- **Consider the shape, size, and cardinality of your data**. While Explain Data can be used with smaller data sets, it requires data that is sufficiently wide and contains enough marks (granularity) to be able to create a model.

- **Don't assume causality**. Correlation is not causation. Explanations are based on models of the data, but are not causal explanations.

  A correlation means that a relationship exists between some data variables, say A and B. You can't tell just from seeing that relationship in the data that A is causing B, or B is causing A, or if something more complicated is actually going on. The data patterns are exactly the same in each of those cases and an algorithm can't tell the difference between each case. Just because two variables seem to change together doesn't necessarily mean that one causes the other to change. A third factor could be causing them both to change, or it may be a coincidence and there might not be any causal relationship at all.

  However, you might have outside knowledge that is not in the data that helps you to identify what's going on. A common type of outside knowledge would be a situation where the data was gathered in an experiment. If you know that B was chosen by flipping a coin, any consistent pattern of difference in A (that isn't just random noise) must be caused by B. For a longer, more in-depth description of these concepts, see the article Causal inference in economics and marketing by Hal Varian.

How explanations are analyzed and evaluated

Explain Data runs a statistical analysis on a dashboard or sheet to find marks that are outliers, or specifically on a mark you select. The analysis also considers possibly related data points

from the data source that aren't represented in the current view.

Explain Data first predicts the value of a mark using only the data that is present in the visualization. Next, data that is in the data source (but not in the current view) is considered and added to the model. The model determines the range of predicted mark values, which is within one standard deviation of the predicted value.

What is an expected range?

The expected value for a mark is the median value in the expected range of values in the underlying data in your viz. The expected range is the range of values between the 15th and 85th percentile that the statistical model predicts for the analyzed mark. Tableau determines the expected range each time it runs a statistical analysis on a selected mark.

Possible explanations are evaluated on their explanatory power using statistical modeling. For each explanation, Tableau compares the expected value with the actual value.

value	Description
Higher than expected / Lower than expected	If an expected value summary says the mark is *lower than expected* or *higher than expected*, it means the aggregated mark value is outside the range of values that a statistical model is predicting for the mark. If an expected value summary says the mark is *slightly lower* or *slightly higher* than expected, or *within the range of natural variation*, it means the aggregated mark value is within the range of predicted mark values, but is lower or higher than the median.
Expected Value	If a mark has an expected value, it means its value falls within the expected range of values that a statistical model is predicting for the mark.
Random Variation	When the analyzed mark has a low number of records, there may not be enough data available for Explain Data to form a statistically significant explanation. If the mark's value is outside the expected range, Explain Data can't determine whether this unexpected value is being caused by random variation or by a meaningful difference in

value	Description
	the underlying records.
No Explanation	When the analyzed mark value is outside of the expected range and it does not fit a statistical model used for Explain Data, no explanations are generated.

Models used for analysis

Explain Data builds models of the data in a view to predict the value of a mark and then determines whether a mark is higher or lower than expected given the model. Next, it considers additional information, like adding additional columns from the data source to the view, or flagging record-level outliers, as potential explanations. For each potential explanation, Explain Data fits a new model, and evaluates how unexpected the mark is given the new information. Explanations are scored by trading off complexity (how much information is added from the data source) against the amount of variability that needs to be explained. Better explanations are simpler than the variation they explain.

Explanation type	Evaluation
Extreme values	Extreme values are aggregated marks that are outliers, based on a model of the visualized marks. The selected mark is considered to contain an extreme value if a record value is in the tails of the distribution of the expected values for the data. An extreme value is determined by comparing the aggregate mark with and without the extreme value. If the mark becomes less surprising by removing a value, then it receives a higher score. When a mark has extreme values, it doesn't automatically mean it has outliers, or that you should exclude those records from the view. That choice is up to you depending on your analysis. The explanation is simply pointing out an interesting extreme value in

Explanation type	Evaluation
	the mark. For example, it could reveal a mistyped value in a record where a banana cost 10 dollars instead of 10 cents. Or, it could reveal that a particular sales person had a great quarter.
Number of records	The number of records explanation models the aggregate sum in terms of the aggregate count; average value of records models it in terms of the aggregate average. The better the model explains the sum, the higher the score. This explanation describes whether the sum is interesting because the count is high or low, or because the average is high or low.
Average value of the mark	This type of explanation is used for aggregate marks that are sums. It explains whether the mark is consistent with the other marks because in terms of its aggregate count or average, noting the relation SUM(X) = COUNT(X) * AVG(X). This explanation describes whether the sum is interesting because the count is high or low, or because the average is high or low.
Contributing Dimensions	This explanation models the target measure of the analyzed mark in terms of the breakdown among categories of the unvisualized dimension. The analysis balances the complexity of the model with how well the mark is explained. An *unvisualized dimension* is a dimension that exists in the data source, but isn't currently being used in the view. This type of explanation is used for sums, counts and averages. The model for unvisualized dimensions is created by splitting out marks according to the categorical values of the explaining column, and then building a model with the value that includes all of the data points in the source visualization. For each row, the model

Explanation type	Evaluation
	attempts to recover each of the individual components that made each mark. The analysis indicates whether the model predicts the mark better when components corresponding to the unvisualized dimension are modeled and then added up, versus using a model where the values of the unvisualized dimension are not known. Aggregate dimension explanations explore how well mark values can be explained without any conditioning. Then, the model conditions on values for each column that is a potential explanation. Conditioning on the distribution of an explanatory column should result in a better prediction.
Contributing Measures	This explanation models the mark in terms of this unvisualized measure, aggregated to its mean across the visualized dimensions. An *unvisualized measure* is a measure that exists in the data source, but isn't currently being used in the view. A Contributing Measures explanation can reveal a linear or quadratic relationship between the unvisualized measure and the target measure.

Disable or Enable Explain Data for a Site

Explain Data is enabled for sites by default, but Tableau administrators may disable it.

1. Go to the **General** site settings.
2. (Tableau Server only) In the **Web Authoring** section, select **Let users edit workbooks in their browser**.
3. In the **Availability of Explain Data** section, select from these options:
    - **Enable** lets Creators and Explorers with the appropriate permissions run Explain Data in editing mode. Lets all users with appropriate permissions run

Explain Data when it is enabled for viewing mode.

- **Disable** prevents all users from running Explain Data or accessing Explain Data settings in workbooks.

4. In Tableau Cloud and Tableau Server 2023.3 or later, to use Explain Data:

   - In the **Availability of Data Guide** section, select **Show**. For more information about Data Guide, see Explore Dashboards with Data Guide.

Use Dashboard Extensions

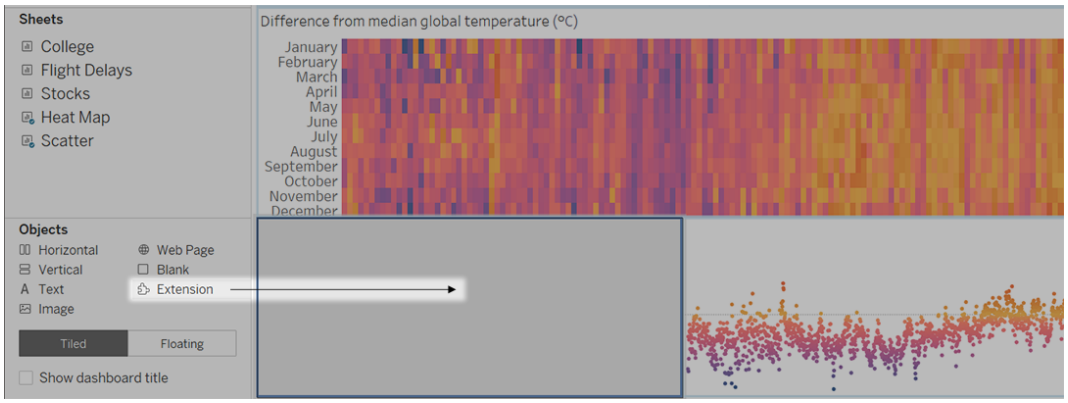
Extensions let you add unique features to dashboards or directly integrate them with applications outside Tableau. Adding extensions is easy; you incorporate them into dashboard layouts just like other dashboard objects.

Extensions expand dashboard functionality with the help of web applications created by third-party developers. If you're a developer and want to create your own extensions, see the Tableau Extensions API documentation on GitHub.

Note: Tableau administrators can turn off dashboard extensions for Tableau Desktop, Tableau Server, and Tableau Cloud.

Add an extension to a dashboard

1. In a Tableau workbook, open a dashboard sheet.
2. From the **Objects** section, drag **Extension** to the dashboard.



3. In the “Add an Extension” dialog box, do either of the following:

    - Search for and select an extension.
    - Click **Access Local Extensions**, and navigate to a .trex file you previously downloaded.

4. If prompted, allow or deny the dashboard extension access to data in the workbook. For more information, see Data security, Network-enabled, and Sandboxed extensions.

    If you allow access, follow any on screen instructions for configuring the extension.

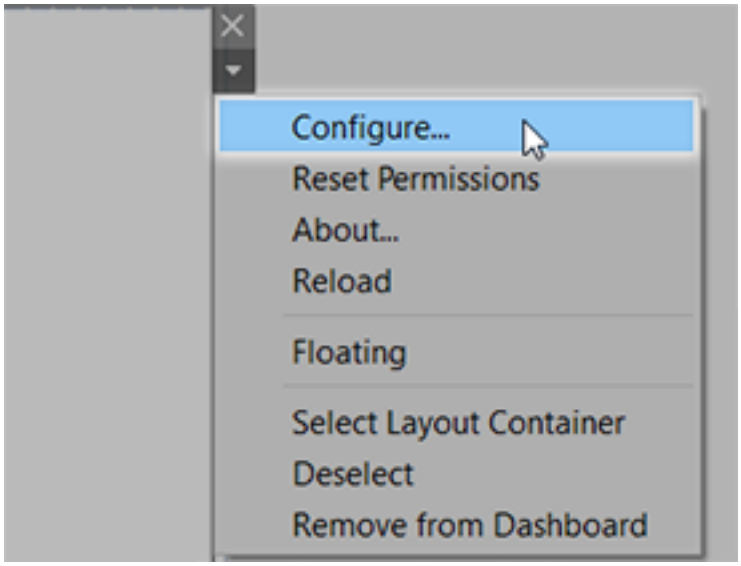
    Note: If you're using Tableau Server or Tableau Cloud, Extension objects will appear blank in prints, PDFs, and images of dashboards (including images in subscription emails).

Configure a dashboard extension

Some dashboard extensions provide configuration options that let you customize features.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **Configure**.

2. Follow the on-screen instructions to configure the extension.



Reload a dashboard extension

If a dashboard extension becomes unresponsive, you might need to reload it, which is similar to refreshing a web page in a browser.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **Reload**.

   The dashboard extension is refreshed and set to its original state.

2. If reloading the extension fails to return it to a useable state, try removing it from the dashboard and adding it again.

Data security, Network-enabled, and Sandboxed extensions

Dashboard extensions are web applications that come in two forms:

- *Network-enabled extensions* run on web servers located outside of your local network.
- *Sandboxed extensions* run in a protected environment without access to any other resource or service on the web.

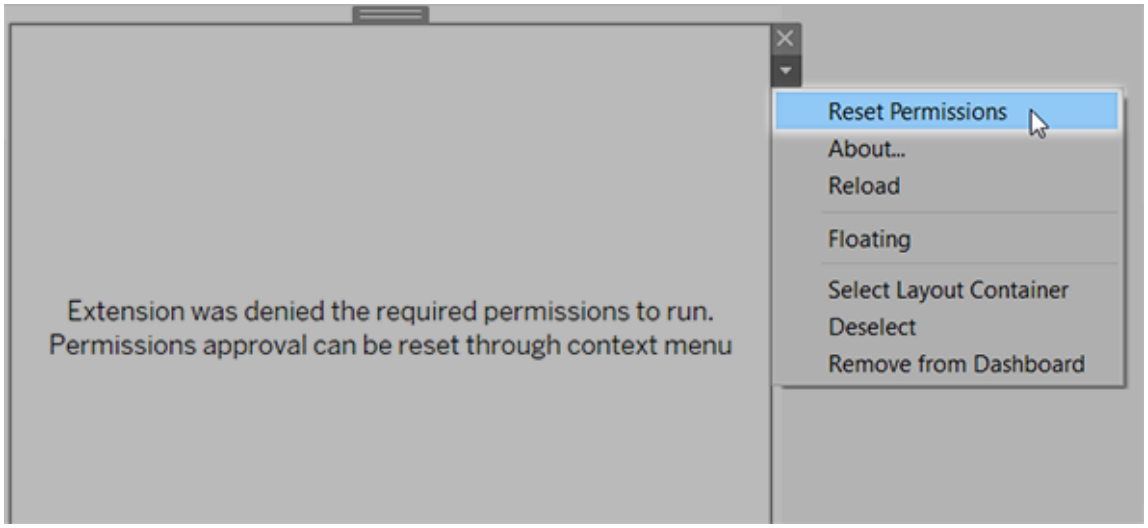
Before adding a Network-enabled extension or viewing a dashboard with one, be certain that you trust the website that hosts it. By default, dashboard extensions use the HTTPS protocol, which guarantees an encrypted channel for sending and receiving data, and ensures some privacy and security.

For more information about data security when using dashboard extensions, see Extension Security - Best Practices for Deployment.

Allow or deny data access to a Network-enabled extension

Depending on how an extension is designed, it can access either visible data in a view, or full underlying data, table and field names from data sources, and information about data source connections. When you add an extension, or view a dashboard with one, you're given an opportunity to allow or deny the extension to run and access this data.

If you're viewing a dashboard with an extension that requires full data access, and that access has been denied, a message appears in place of the extension. If you trust the extension and want to use it, you can reset permissions and allow the extension to run.



1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **Reset Permissions**.

2. Click either **Allow** to let the extension run and access data, or **Deny** to prevent the extension from running.

Ensure that JavaScript is enabled in Tableau Desktop

Dashboard extensions interact with data using the Tableau Extensions API library, a JavaScript library. If you want to use extensions, be sure that JavaScript is enabled in the dashboard security settings:

Choose **Help** > **Settings and Performance** > **Set Dashboard Web View Security** > **Enable JavaScript**.

Ensure that extensions run on Tableau Cloud or Tableau Server

You can add extensions to workbooks you publish from Tableau Desktop or directly in the web-authoring mode of Tableau Cloud and Tableau Server. A Tableau administrator must allow extensions to run on a site and add Network-enabled extensions to a safe list. Administrators should only allow extensions that you have tested and trust.

If you want to use a dashboard extension on Tableau Cloud or Tableau Server, direct your administrator to Manage Dashboard Extensions in Tableau Cloud or Manage Dashboard Extensions in Tableau Server.

Supported web browsers for Sandboxed extensions

Sandboxed extensions run in all browsers supported Tableau Server and Tableau Cloud except Internet Explorer 11.

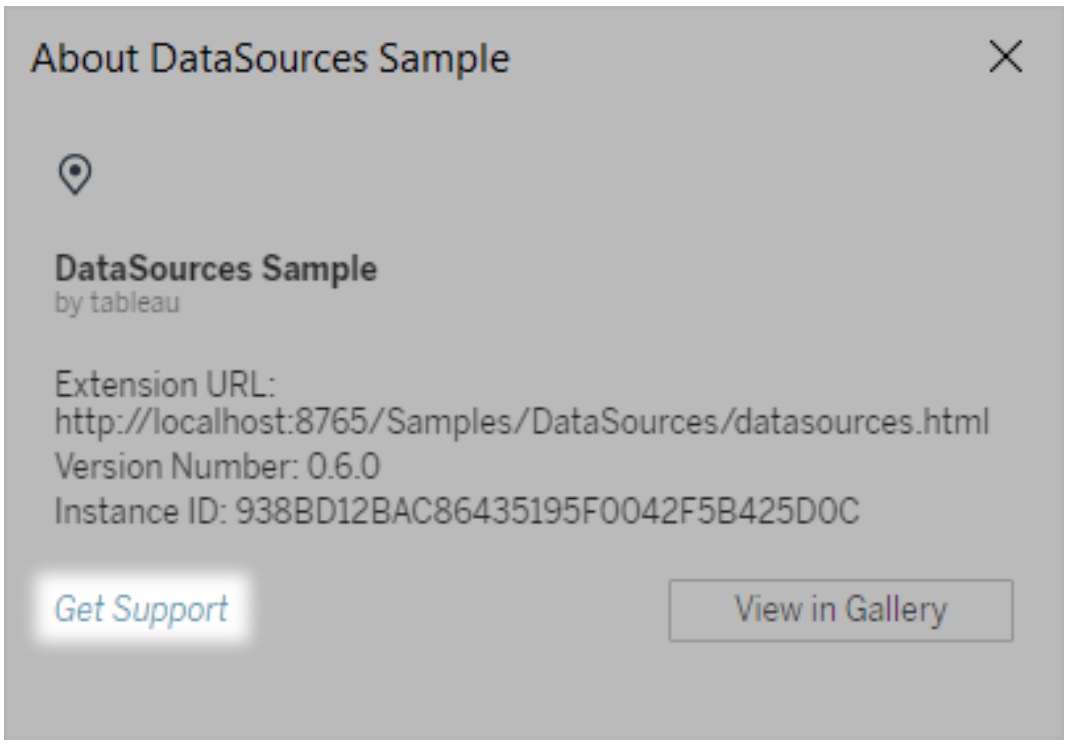
Supported versions of Tableau Server for Sandboxed extensions

You can use Sandboxed extensions in Tableau Server 2019.4 and later.

Get support for dashboard extensions

To get help for an extension, you'll need to contact the developer or company who created it.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **About.**
2. Click **Get Support** to go to the support page of the extension developer.



Note: Tableau doesn't provide support for extensions or for other programs that interface with the Extensions API. However, you can submit questions and ask for help in the Tableau developer community.

Format Animations

Animate visualizations to better highlight changing patterns in your data, reveal spikes and outliers, and see how data points cluster and separate.

Animations visually transition between filter, sort, and zoom settings, different pages, and changes to filter, parameter, and set actions. As visualizations animate in response to these

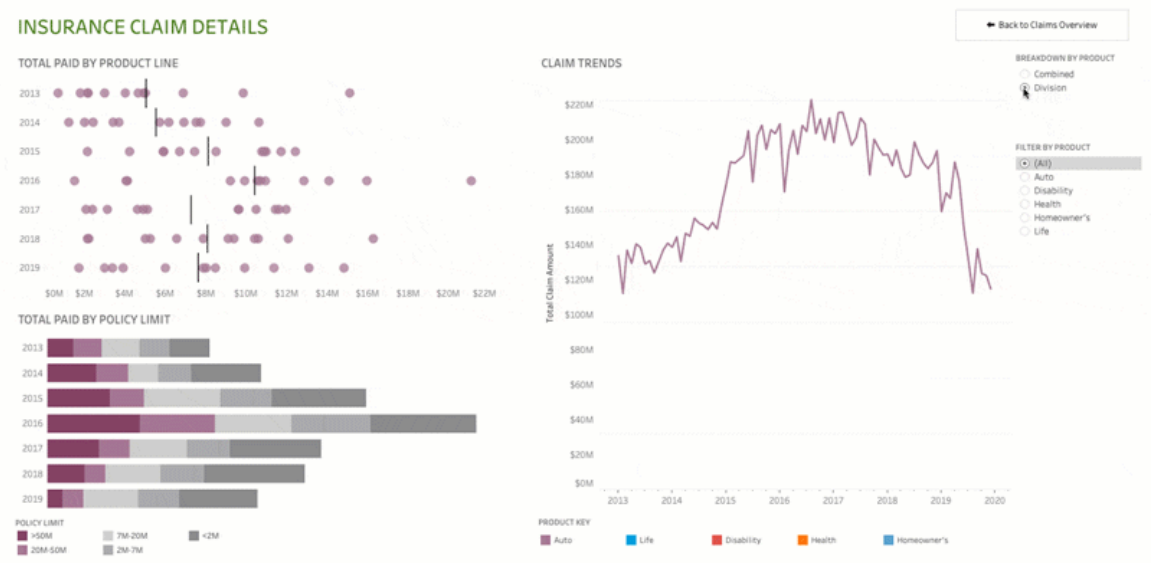
changes, viewers can more clearly see how data differs, helping them make better informed decisions.

Understanding simultaneous and sequential animations

When you author animations, you can choose between two different styles: simultaneous or sequential. Here are examples of each type.

Simultaneous animations

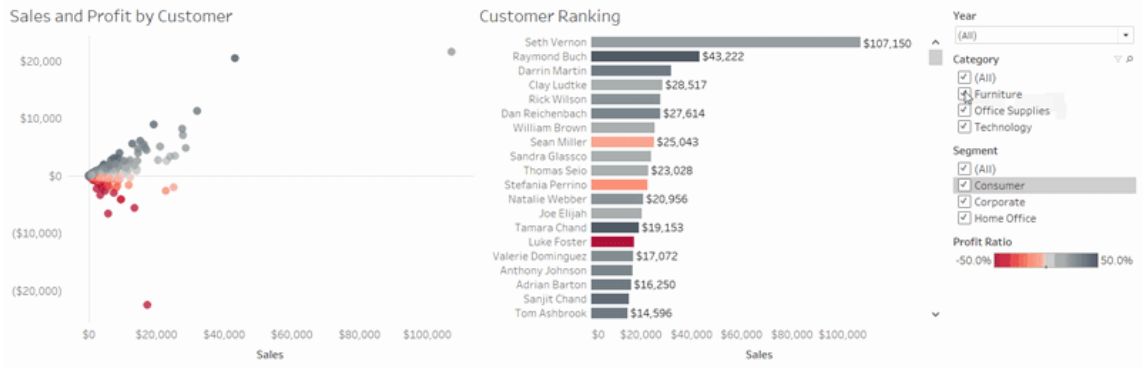
The default simultaneous animations are faster and work well when showing value changes in simpler charts and dashboards.



Click the image above to replay the animation.

Sequential animations

Sequential animations take more time but make complex changes clearer by presenting them step-by-step.



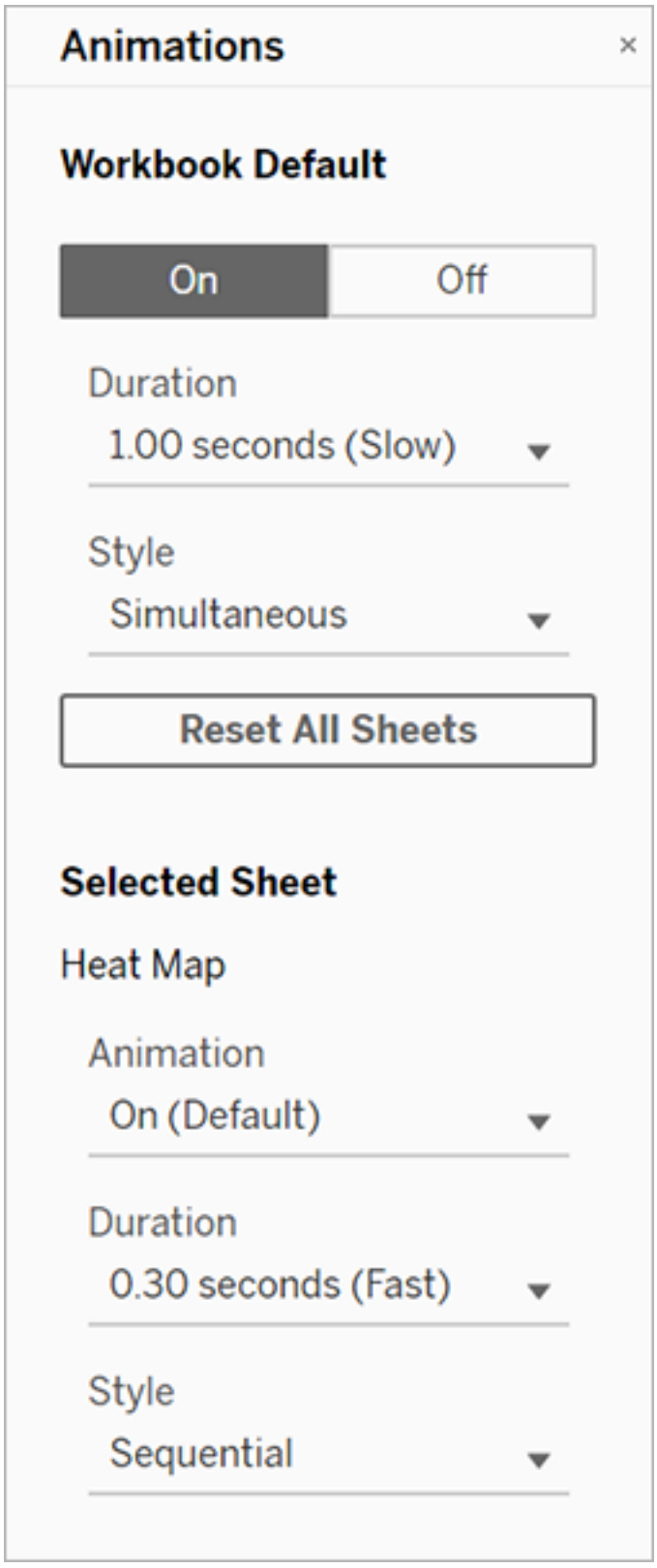
Click the image above to replay the animation.

Animate visualizations in a workbook

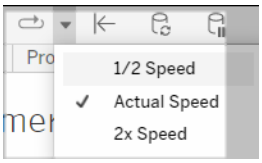
When you create a new workbook, Tableau enables animations for your viz by default. You can turn animations on or off at the user and workbook level.

1. Choose **Format** > **Animations**.
2. If you want to animate every sheet, under **Workbook Default**, click **On**. Then do the following:
    - For **Duration**, choose a preset, or specify a custom duration of up to 10 seconds.
    - For **Style**, choose **Simultaneous** to play all animations at once or **Sequential** to fade out marks, move and sort them, and then fade them in.
3. To override workbook defaults for a particular sheet, change the settings under **Selected Sheet**.

> **Note:** In the Selected Sheet section, “(Default)” indicates a setting that automatically reflects the related Workbook Default setting.


Animations
Workbook Default
On
Off
Duration
1.00 seconds (Slow)
Style
Simultaneous
Reset All Sheets
Selected Sheet
Heat Map
Animation
On (Default)
Duration
0.30 seconds (Fast)
Style
Sequential

To replay an animation, click the **Replay** button in the toolbar. From the **Replay** button, you can also choose the speed at which the animation replays: actual speed, 2x speed, or 1/2 speed.



Reset animation settings for a workbook

You can reset animations to return an entire workbook to the default animation settings. Be aware that this turns animations off by default.

1. Choose **Format** > **Animations**.
2. In the middle of the **Animations** pane, click **Reset All Sheets**.

Completely disable all animations

When you create a new workbook, animations are enabled by default. If you find animations distracting while viewing vizzes, you can completely disable them so they never play. (This isn't a system-wide setting; each user needs to apply it separately.)

- In Tableau Desktop, choose **Help** > **Settings and Performance**, and deselect **Enable Animations**.
- In Tableau Cloud or Tableau Server, click your profile image or initials in the top right corner of the browser, and choose **My Account Settings**. Then scroll down to the bottom of the page, deselect **Enable animations**, and click **Save Changes**.

Note: When animations are disabled, you can still choose **Format** > **Animations** in authoring mode and adjust settings—but they will have no effect.

Format decimals for axes animations

If the number of decimal places for a measure is set to the default, then the number of decimals shown during the axis animation might fluctuate during the axes animation. To avoid this,

format the number of decimal places displayed for a measure. For more information, see Format Numbers and Null Values.

Why animations won't play

Server rendering

Animations won't play if a viz is server-rendered. To ensure that vizzes render on a client computer or mobile device, use these techniques:

- If you're a viz author, reduce viz complexity.
- If you're a Tableau Server administrator, increase the complexity threshold for client-side rendering.

> **Note:** On computers with lower processing power, animations may appear choppy, but users can continue to interact with vizzes without any delays in responsiveness.

Unsupported browsers and features

Animations are supported by all web browsers except Internet Explorer.

The following Tableau features don't animate:

- Maps, polygons, and density marks in web browsers
- Pie and text marks
- Headers
- Forecasts, trends, and reference lines
- Page history trails (If a viz includes these, turn off animations to avoid unexpected behavior.)

Format Numbers and Null Values

You can specify the format for numeric values that display in your viz, including measures, dimensions, parameters, calculated fields, and axis labels. When specifying a number format, you can select from a set of standard formats, such as number, currency, scientific, and per-

centage. You can also define a custom number format with the option to include special characters.

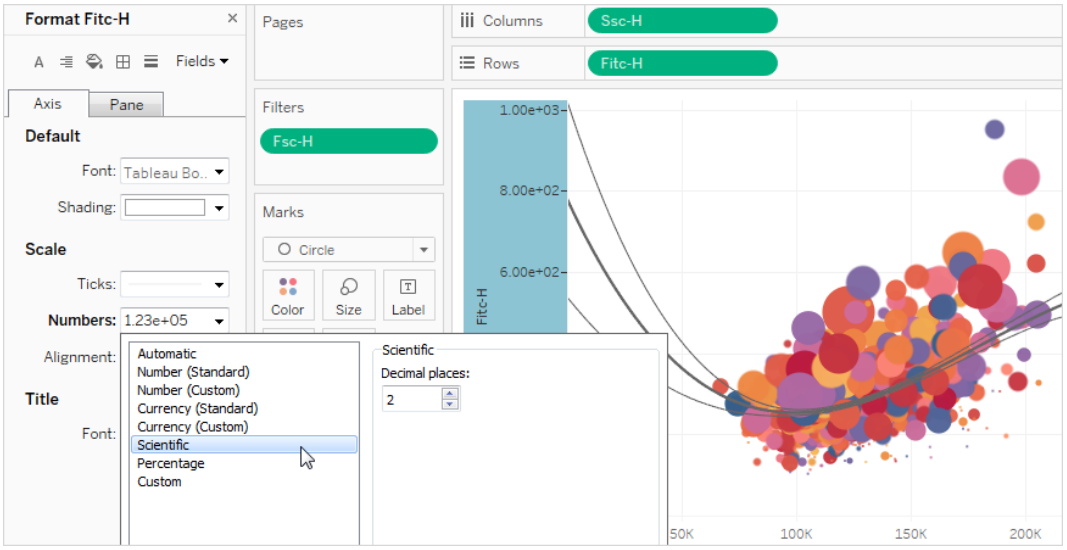
When a measure contains null values, you can use formatting to handle the null values in a different way, such as replacing nulls with zeros or hiding them.

For Tableau Desktop

Specify a number format

1. Right-click (control-click on Mac) a number in the view and select **Format**.
2. In the **Format** pane, click the **Numbers** drop-down menu.
3. Select a number format.

   Some formats require additional settings. For example, if you select **Scientific**, you must also specify the number of decimal places.



Here are the number formats and associated options available in Tableau.

NUMBER FORMAT	FORMAT OPTIONS
Automatic: format is automatically selected based on either the format spe-	None.

cified by the data source or the data contained in the field.	
Number (Standard): format is based on locale selected.	**Locale**: number format changes based on the geographical location selected.
Number (Custom): format is customized to your choice.	**Decimal Places**: the number of decimal places to display. **Negative Values**: how negative values are displayed. **Units**: the number is displayed using the specified units. For example, if the number is 20,000 and the units are thousands, the number will be displayed as 20K. **Prefix/Suffix**: characters that precede and follow each displayed number. **Include thousands separators**: whether the number shows separators every thousand (example: 100,000 vs. 100000).
Currency (Standard): format and currency symbol is based on locale selected.	**Locale**: currency format based on the geographical location selected.
Currency (Custom): format and currency symbol is customized to your choice.	**Decimal Places**: the number of decimal places to display. **Negative Values**: how negative values are displayed. **Units**: the number is displayed using the spe-

	cified units. For example, if the number is 20,000 and the units are thousands, the number is displayed as 20K. **Prefix/Suffix**: characters that precede and follow each displayed number. **Include thousands separators**: whether the number shows separators every thousand (example: 100,000 vs. 100000).
Scientific: numbers are displayed in scientific notation.	**Decimal**: the number of decimal places to display.
Percentage: numbers are displayed as a percentage with the percent symbol. The value of 1 is interpreted as 100% and 0 as 0%	**Decimal**: the number of decimal places to display.
Custom: format is based entirely on what is specified in the format options.	**Custom**: type in the format you want to use, including special characters (optional). See "Define a custom number format" in this topic for details.

Define a custom number format

To apply a custom number format in your viz:

1. Right-click (control-click on Mac) a number in the view and select **Format**.
2. In the **Format** pane, click the **Numbers** drop-down menu and select **Custom**.
3. In the **Format** field, define your formatting preferences using the following syntax: `Positive number format;Negative number format;Zero values`.

When defining your number format code, keep in mind that:

- You can specify the formatting for up to three types of numbers in the following order: positive numbers, negative numbers, and zeros.
- Each number type must be separated by a semicolon (;).
- If you specify only one number type, the format for that type is used for all numbers.
- If you specify two number types, the format for the first type is applied to positive numbers and zeros, while the format for the second type is applied to negative numbers.
- If you skip types in your number format code, you must include a semicolon (;) for each of the missing types.

Custom number format examples

Refer to the following table for examples of commonly used custom number format codes that you can use in your viz.

The syntax has three portions: `<positive number format>;<negative number format>;<zero format>` separated by semicolons.

USE CASE	NUMBER CODE SYNTAX	EXAMPLE OUTPUT
Show only positive values	#,##; ; (note a blank space after the 2nd and 3rd semicolon)	Positive values: 1,234 Negative values: (only the blank space displays) Zero values: (only the blank space displays)
Show only negative values	;-#,##;	Positive values: (nothing displays) Negative values: -1,234 Zero values: (nothing displays)
Show only zero values	;;0;	Positive values: (nothing displays) Negative values: (nothing displays) Zero values: 0

Hide zero values	#,###;-#,###;;	Positive values: 1,234 Negative values: -1,234 Zero values: (nothing displays)
Show negative values in parentheses	#,###;(#,###);;	Positive values: 1,234 Negative values: (1,234) Zero values: (nothing displays)
Add a character prefix to a value	$#,###.##;-$#,###.##;$0;	Positive values: $1,234.56 Negative values: -$1,234.56 Zero values: $0
Add a character suffix to a value	#%;-#%;0%;	Positive values: 12% Negative values: -34% Zero values: 0%
Add text descriptors	"$"#,#" Surplus";"$"-##" Shortage"; "$"0;	Positive values: $1,234 Surplus Negative values: $-1,234 Shortage Zero values: $0

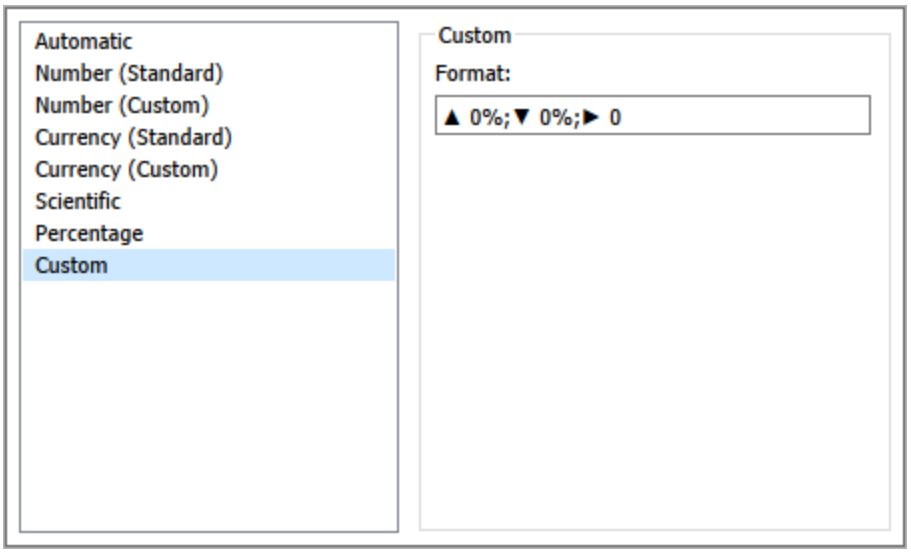
There are several ways to customize your number format to meet your needs. For more code guidance and examples, see Review guidelines for customizing a number format in the Microsoft Knowledge Base.

Note: There are slight deviations between the options described by Microsoft and those available in Tableau. For example, custom number formats that align text within columns aren't relevant in Tableau. Additionally, custom number formats to color code text aren't applicable because you can apply color to text using the Marks card. (For more information on applying color, see Control the Appearance of Marks in the View.) Be sure to only use custom number formats that apply in Tableau.

Include special characters in a custom number format

One of the benefits of custom number formatting is the ability to include special characters or symbols. Symbols can make it easier to quickly understand comparisons between measures and calculations in your viz.

For example, let's say you want to show a month-over-month comparison of profits for three of your company's top-selling products. Rather than using the standard label to show that the profit for these products changed +5%, -2%, and 0% label from last month, you can set the custom number format as follows to show these changes as ▲5%,▼2%, and ►0.



Set the default number format for a field

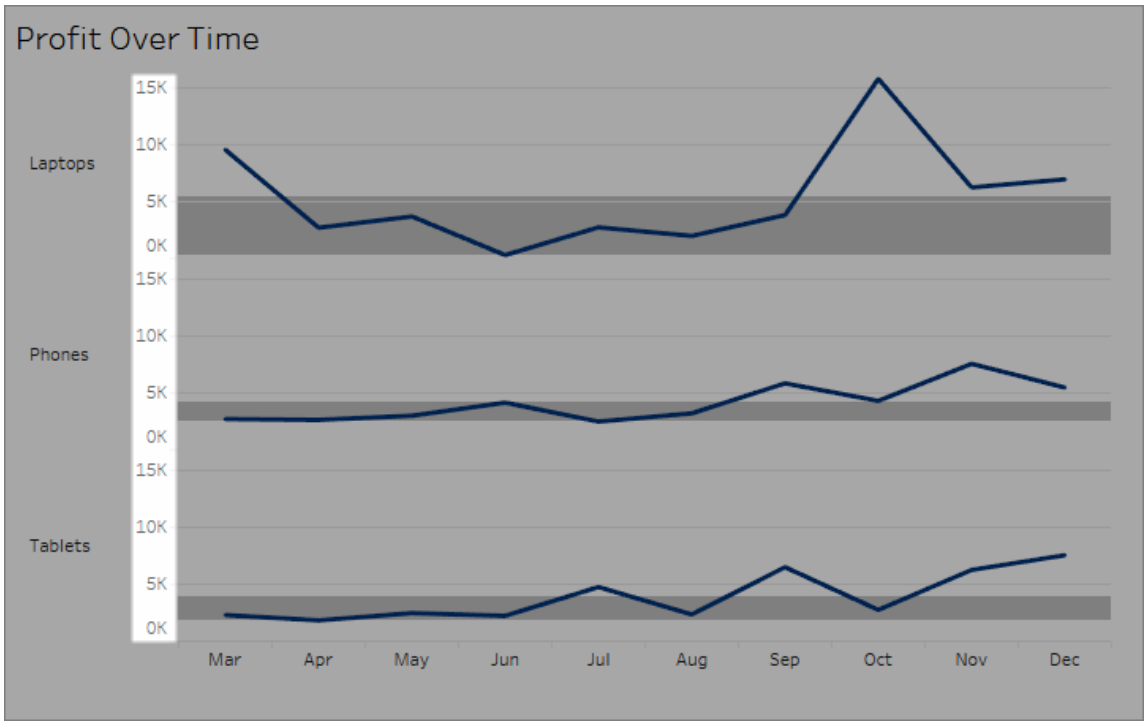
Right-click (control-click on Mac) the field in the **Data** pane and select **Default Properties** > **Number Format**.

In the subsequent dialog box, specify a number format to be used whenever the field is added to the view. The default number format is saved with the workbook. It's also exported when you export the connection information.

Note: Formatting numbers using the **Format** pane overrides any number formatting applied elsewhere.

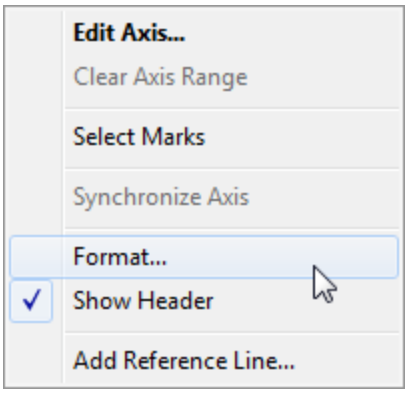
Format a measure as currency

The view in the following image shows profit over time. Notice that the profit figures on the vertical axis aren't formatted as currency.



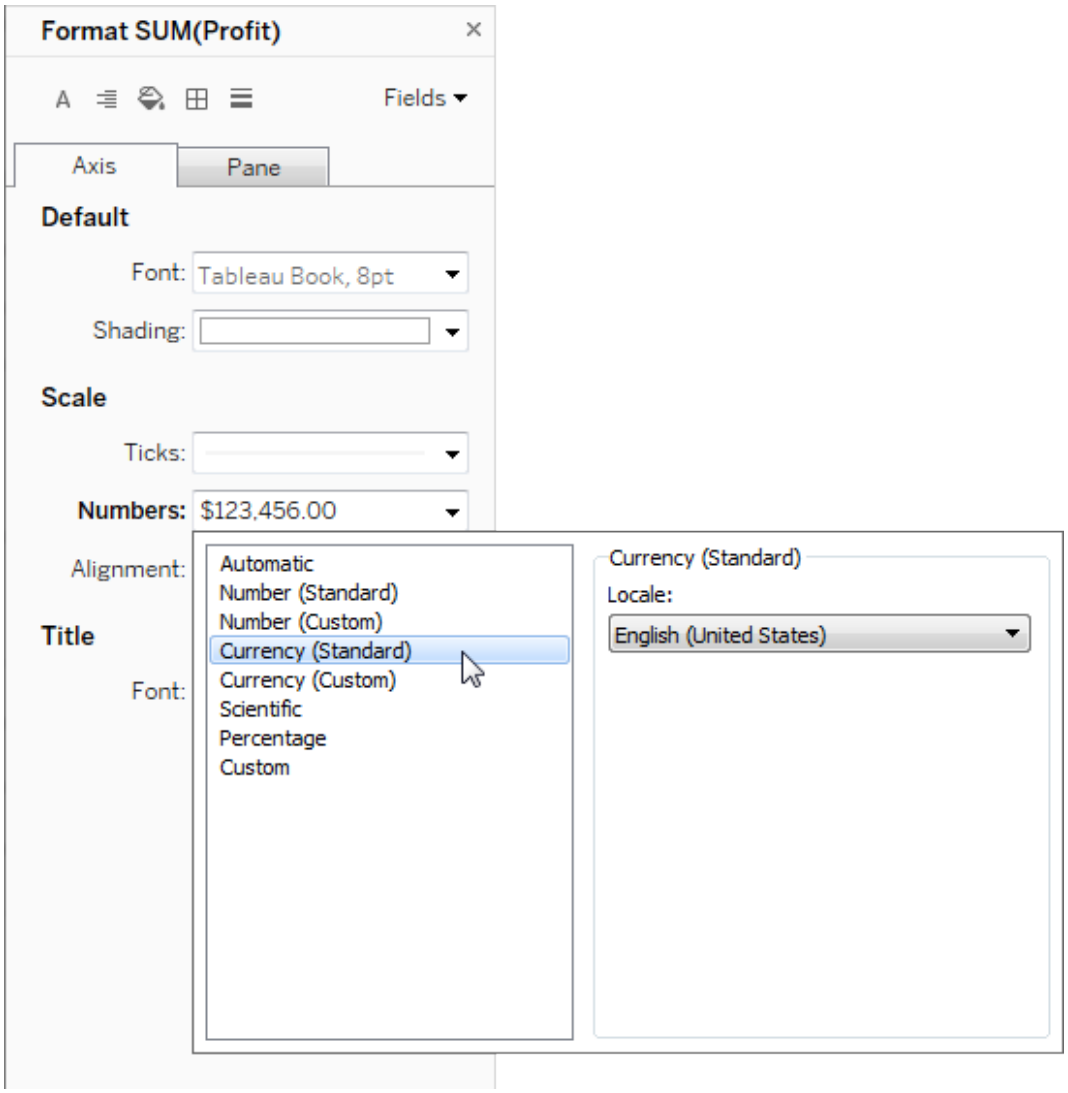
To format the numbers as currency:

1. Right-click the **Profit** axis and choose **Format**.



2. On the **Axis** tab in the **Format** pane, under **Scale**, select the **Numbers** drop-down list, and then select one of the following:

   Currency (Standard) to add a dollar sign and two decimal places to the figures.



Currency (Custom) to specify the number of decimal places, how to show negative values, the units, whether to include a prefix or suffix, and whether to include a separator character.

Use locale to specify number formats

By default, Tableau uses your computer's locale and language settings to format numbers. But you can explicitly set a different locale in the **Format** pane.

The following steps show how to set Swiss German currency, using the same view as in the previous section.

1. Right-click the **Profit** axis and select **Format**.
2. On the **Axis** tab, under **Scale**, select the **Numbers** drop-down list and then select **Currency (Standard)**.
3. In the **Locale** drop-down list, items appear in a **Language (Country)** format. For this example, select **German (Switzerland)**. The view updates to show the sales figures as Swiss Francs, formatted for the German language.

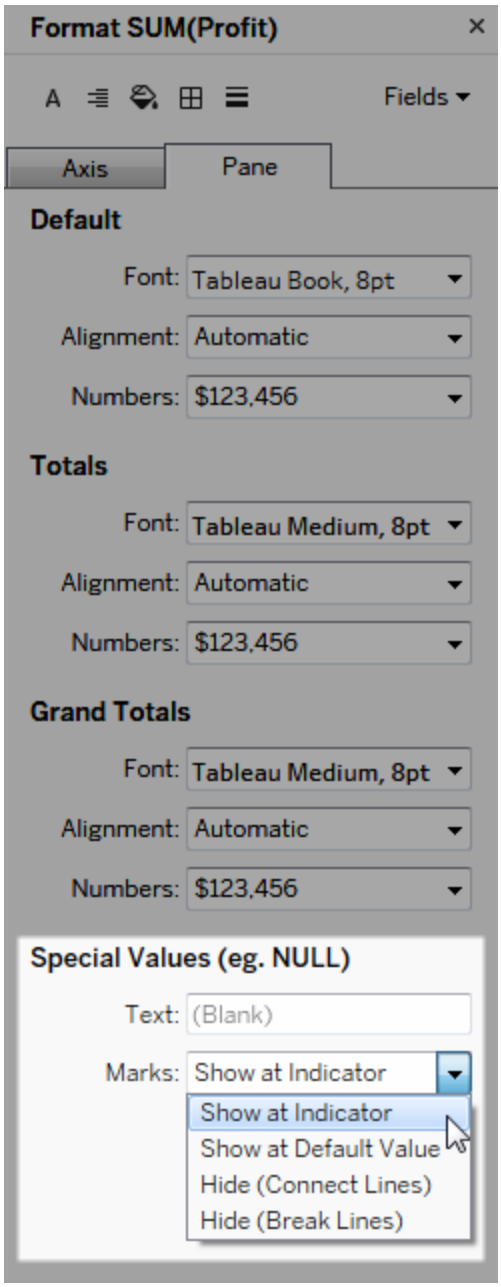
Tip: You can change the default currency setting so that every time you drag the **Profit** measure to a view it uses the settings you want. In the **Data** pane, right-click **Profit** (or other monetary measure), and select **Default Properties** > **Number Format**. Then format the field as shown above.

Format null values

When a measure contains null values, they're usually plotted in a view as zero. However, sometimes that changes the view and you'd rather just suppress null values altogether. You can format each measure to handle null values in a unique way.

To format null values for a specific field:

1. Right-click the field in the view that has the null value (Control-click on a Mac) and choose **Format**.
2. Go to the **Pane** tab.
3. In the **Special Values** area, specify whether to show the null value using an indicator in the lower right corner of the view, plot it at a default value (such as zero for number fields), hide the value but connect the line, or hide and break the line to indicate that a null value exists.



4. If you specify text in the **Text** field, it appears in the view for a null value when mark labels are turned on. See Show and Hide Mark Labels.

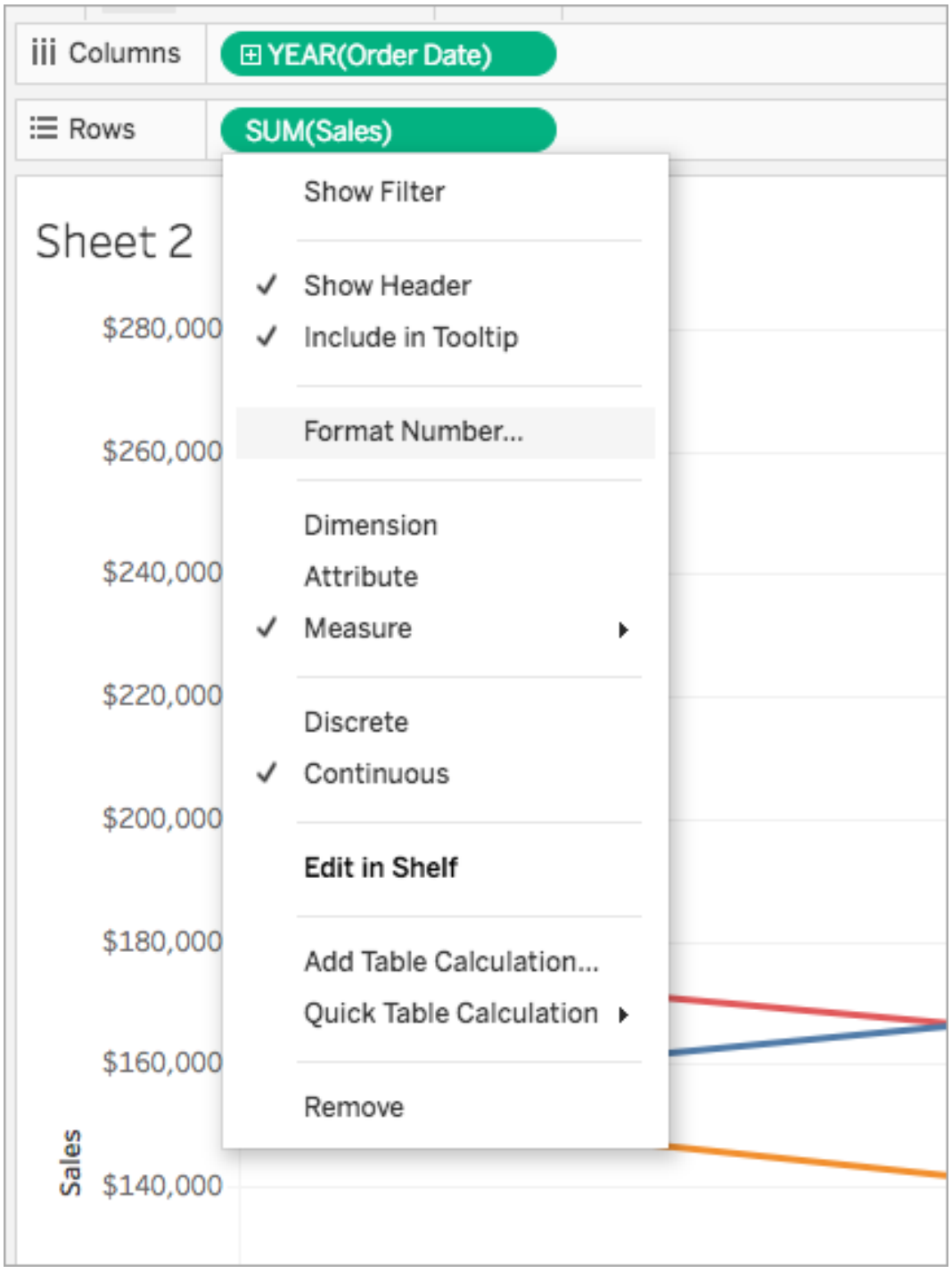
Note: The Special Values area isn't available for dimensions or discrete measures.

For Tableau Server or Tableau Cloud

Specify a number format

When authoring a view on the web, you can specify the number format for a field used in the view.

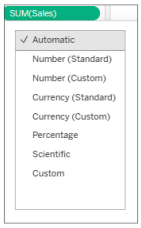
1. In web editing mode, right-click a measure in the view and select **Format Number**.



2. In the dialog box that appears, select a number format.

   Some formats provide additional settings. For example, if you select **Currency**, you can also specify the number of decimal places, as well as the units, and whether or not to include separators, such as commas.

   In this example, Sales is formatted as a Currency with zero decimal places and thousand (k) units. Sales numbers in the view update with these settings. Labels and tooltips update as well.



   Here are the number formats and associated options available in Tableau.

NUMBER FORMAT	FORMAT OPTIONS
Automatic: format is automatically selected based on either the format specified by the data source or the data contained in the field.	None.
Number (Custom): format is customized to your choice.	**Decimal Places**: the number of decimal places to display. **Units**: the number is displayed using the specified units. For example, if the number is 20,000 and the units are thousands, the number is displayed as 20K. **Include separators**: whether the number

	shows separators every thousand (example: 100,000 vs. 100000).
Currency (Custom): format and currency symbol is customized to your choice.	**Decimal Places**: the number of decimal places to display. **Units**: the number is displayed using the specified units. For example, if the number is 20,000 and the units are thousands, the number is displayed as 20K. **Include separators**: whether the number shows separators every thousand (example: 100,000 vs. 100000).
Percentage (Custom): numbers are displayed as a percentage with the percent symbol. The value of 1 is interpreted as 100% and 0 as 0%	**Decimal Places**: the number of decimal places to display.
Scientific (Custom): numbers are displayed in scientific notation.	**Decimal Places**: the number of decimal places to display.

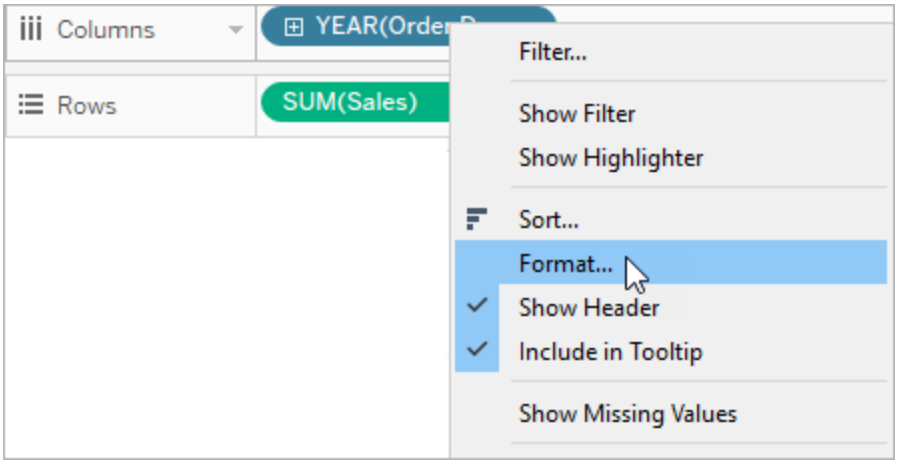
Custom Date Formats

This article discusses using the custom date format field to format dates in a view. For an overview of how Tableau works with dates, see Dates and Times, or Changing Date Levels. For setting date properties for a **data source**, see Date Properties for a Data Source.

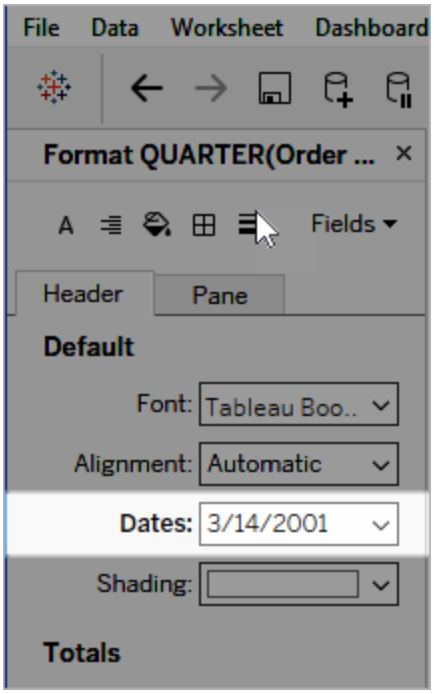
How to find the custom date format field

Format a date field in a view (Tableau Desktop)

To format a date field in the view in Tableau Desktop, right-click (Control-click on a Mac) the field and choose **Format**.



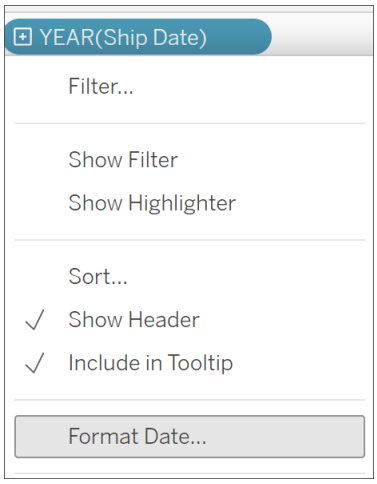
This opens the **Format** panel to the left of your view. Select the **Dates** field.



When you format dates, Tableau presents a list of available formats. Usually, the last item in the list is **Custom**. You can specify a custom date using format symbols listed in the Supported date format symbols table, either alone or in combination.

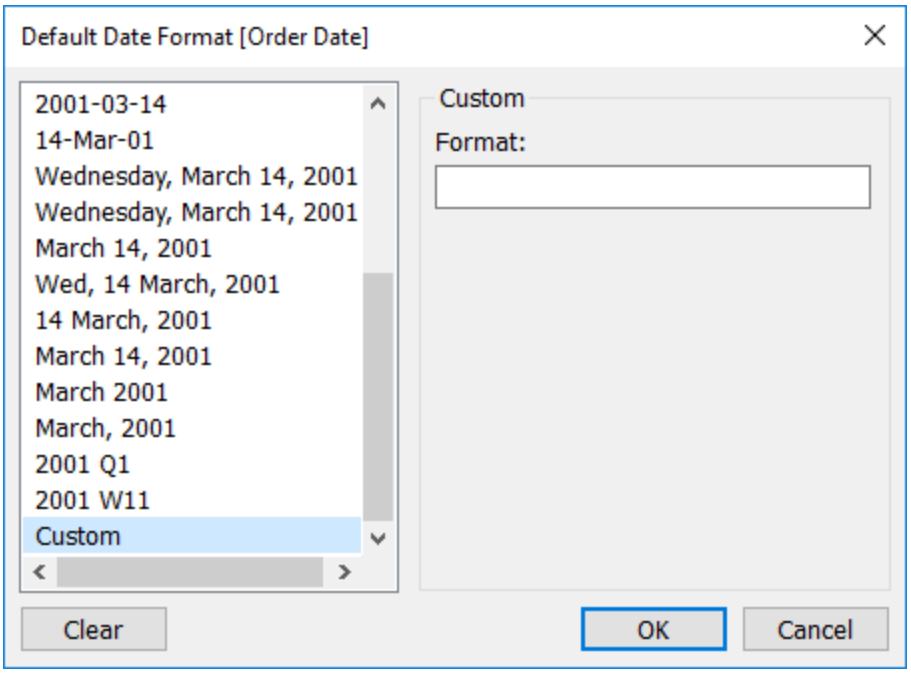
Format a date field in a view (Tableau Cloud and Tableau Server)

To format a date field in the view in Tableau Cloud and Tableau Server, right-click (Control-click on a Mac) the field and choose **Format Date.**



Format a date field in the Data pane (Tableau Desktop only)

To format a date field in the **Data** pane, right-click the field and choose **Default Properties** > **Date Format**.



The date formats in the table are supported when your workbook is connected to a Tableau extract or has a live connection to a data source that also supports the date format. (Refer to your data source's documentation to verify that the date format you want is supported.)

Tableau retrieves date formats from the data source. Tableau Server can also retrieve date formats from the Run As user account on the server that's running Tableau Server.

Note: The following date formats might not be the same as those used with the Type Conversion function. See Convert Strings to Date Fields for more information.

Supported date format symbols

Use the following symbols to construct a custom date format.

Symbol	Description
(:)	Time separator. In some locales, a different character is used to represent the time separator. The time separator separates hours, minutes, and seconds when time values are formatted. The actual character used as the time sep-

	arator in formatted output is determined by your system settings.
(/)	Date separator. In some locales, a different character is used to represent the date separator. The date separator separates the day, month, and year when date values are formatted. The actual character used as the date separator in formatted output is determined by your system settings.
c	Display the date as `ddddd` and display the time as `ttttt`, in that order. Display only date information if there's no fractional part to the date serial number; display only time information if there's no integer portion.
d	Display the day as a number without a leading zero (1-31).
dd	Display the day as a number with a leading zero (01-31).
ddd	Display the day as an abbreviation (Sun, Sat).
dddd	Display the day as a full name (Sunday, Saturday).
ddddd	Display the date as a complete date (including day, month, and year), formatted according to your system's short date format setting. The default short date format is `m/d/yy`.
dddddd	Display a date serial number as a complete date (including day, month, and year) formatted according to the long date setting recognized by your system. The default long date format is `mmmm dd, yyyy`.
aaaa	The same as dddd, only it's the localized version of the string.
w	Display the day of the week as a number (1 for Sunday through 7 for Saturday).
ww	Display the week of the year as a number (1-54).
M	Display the month as a number without a leading zero (1 12). If `m` immediately follows `h` or `hh`, the minute rather than the month is displayed.
MM	Display the month as a number with a leading zero (01-12). If `m` immediately follows `h` or `hh`, the minute rather than the month is displayed.

MMM	Display the month as an abbreviation (Jan-Dec).
MMMM	Display the month as a full month name (January-December).
MMMMM	Display the month as a single letter abbreviation (J-D)
oooo	The same as MMMM, but localized.
q	Display the quarter of the year as a number (1- 4).
y	Display the day of the year as a number (1-366).
yy	Display the year as a 2-digit number (00-99).
yyyy	Display the year as a 4-digit number (100-9999).
h	Display the hour as a number without leading zeros (0-23).
Hh	Display the hour as a number with leading zeros (00-23).
N	Display the minute as a number without leading zeros (0 59).
Nn	Display the minute as a number with leading zeros (00 59).
S	Display the second as a number without leading zeros (0 59).
Ss	Display the second as a number with leading zeros (00 59).
000	Display milliseconds. Use a period character as a separator before specifying milliseconds.
t t t t t	Display a time as a complete time (including hour, minute, and second), formatted using the time separator defined by the time format recognized by your system. A leading zero is displayed if the leading zero option is selected and the time is before 10:00 A.M. or P.M. The default time format is `h:mm:ss`.
AM/PM	Use the 12-hour clock and display an uppercase AM with any hour before noon; display an uppercase PM with any hour between noon and 11:59 P.M.
am/pm	Use the 12-hour clock and display a lowercase AM with any hour before noon; display a lowercase PM with any hour between noon and 11:59 P.M.
A/P	Use the 12-hour clock and display an uppercase A with any hour before noon; display an uppercase P with any hour between noon and 11:59 P.M.

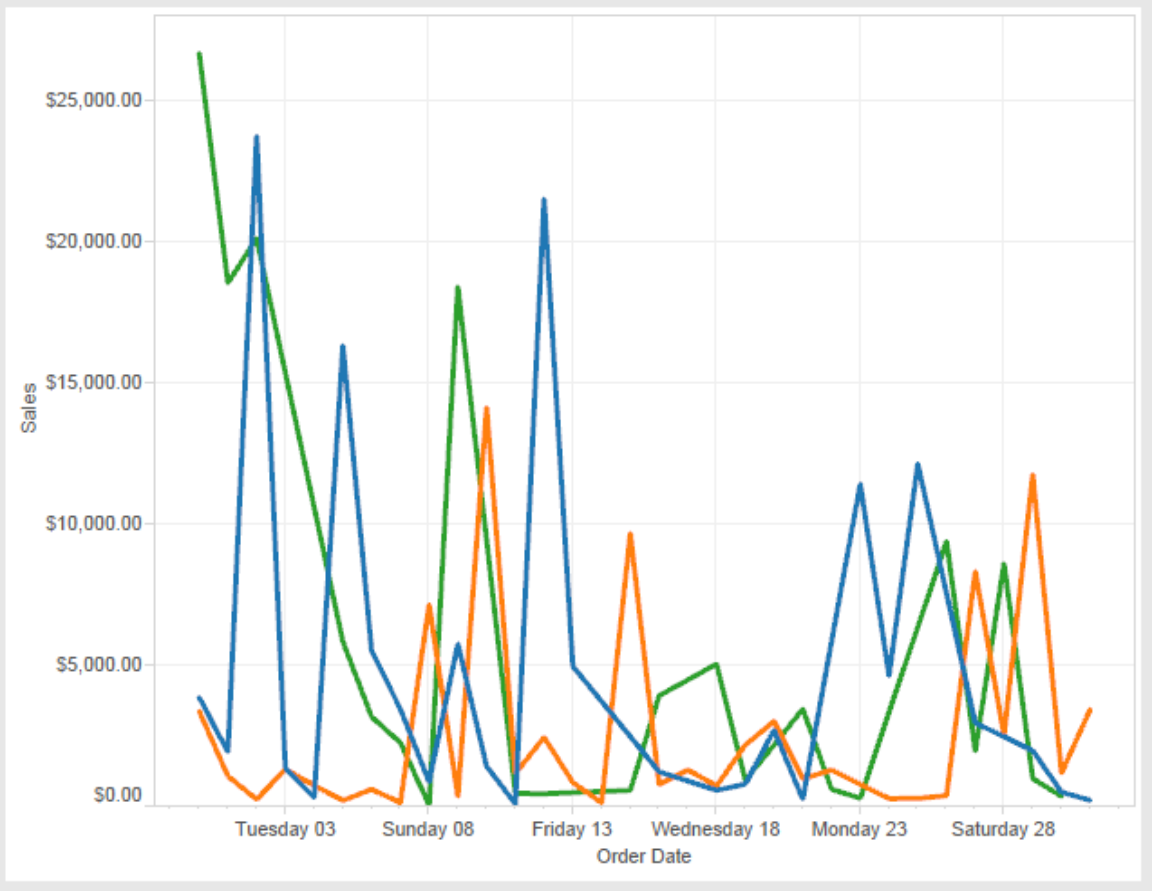
a/p	Use the 12-hour clock and display a lowercase A with any hour before noon; display a lowercase P with any hour between noon and 11:59 P.M.
AMPM	Use the 12-hour clock and display the AM string literal as defined by your system with any hour before noon; display the PM string literal as defined by your system with any hour between noon and 11:59 P.M. AMPM can be either uppercase or lowercase, but the case of the string displayed matches the string as defined by your system settings. The default format is AM/PM.

Custom date format examples

Any of the date format symbols in the table above can be used alone or in combination.

Specifying a custom format yyyy-MM-dd HH:mm:ss.000 would produce dates in the format 2015-05-10 11:22:16.543. Such a format might be appropriate for scientific data.

Specifying a custom format DDDD DD would produce dates that show the Weekday and the Day, as shown below.



Specifying a custom format `yy-mm-dd (dddd)` would produce dates in the format **18-01-04 (Thursday)**.

Specifying a custom format `"Q"1 YYYY` would produce dates that show **Q1 2018**.

Support for Japanese era-based date formats

Tableau supports Japanese emperor-era-based date (Wareki) formats. Here's how to apply an era-based date format to a field in your view:

1. Set your workbook locale to Japanese.
2. Right-click the field in the view for which you want to set a date format.

3. Choose **Format**.

4. In the **Format** pane, from the **Dates** drop-down list, select a format.

If the format that you want isn't listed, you can construct your own date format. To do this, choose **Custom format** in the **Dates** box, then type your format using the Tableau date placeholders. The following era-based year placeholders are available:

Symbol	Description
g	Short era name (such as H for the Heisei era).
gg	Era name (such as 平成).
ggg	Long era name (for Japanese, this is the same as the regular era name).
e	Era-based year, such as 1 for the first year of an era.
ee	Era-based year, such as 01 for the first year of an era. If there's only one digit, then the era-based year will have a zero added to the front.

If your workbook locale isn't Japanese, you can create a custom date format, then insert the language code !ja_JP! in front of your format, so that it looks like this:

```
!ja_JP! gg ee"年"mm"月"dd"日"
```

The language code forces the date to be treated as if it's a Japanese date.

Era-based dates aren't fully supported by the Tableau Server browser view. In particular, if you publish a workbook that contains an interactive filter, the e and g placeholders won't be filled in:

To avoid this issue, don't show era-based dates in interactive filters if your workbook will be viewed in a browser.

Using literal text in a date format

You may want your date format to include some words or phrases, such as **Fiscal Quarter q of yyyy**. However, if you type that text directly into the Tableau format box, it may treat the letters like date parts:

Quarter of Order Date
Fi01/1/2010al 1uarter 1 of 2010
Fi04/1/2010al 2uarter 2 of 2010
Fi07/1/2010al 3uarter 3 of 2010
Fi010/1/2010al 4uarter 4 of 2010

To prevent Tableau from doing this, put double quotes around the letters and words that shouldn't be treated as date parts: `"Fiscal Quarter" q "of" yyyy`.

If you want a literal quote inside of a quoted section, insert this code: `"\""`. For example, the format `"Fiscal "\"" Quarter"` would be formatted as **Fiscal " Quarter**.

Format syntax in DATEPARSE function for extract data sources

If you're using the DATEPARSE function in an extract, use the syntax defined by the Unicode Consortium.

The following table lists the field types that can be represented in the format parameter of the DATEPARSE function. Click the field type to get information about the symbols, field patterns, examples, and descriptions from the Unicode Consortium website.

Unit of time	**Notes**
Era	n/a
Year	All symbols are supported in .hyper extracts except for "U". **Notes:**

Unit of time	Notes
	• Negative values denote a year before Christ (BC). For example, `DATEPARSE('y','-10')` returns the first January of 11BC and `DATEPARSE('y','-0')` returns the first January of 1BC. • When working with the calendar year "y", the pattern "yy" requests the two low-order digits of the year. For numbers < 70, the DATEPARSE function returns the year 2000+x. For numbers >=70, the DATEPARSE function returns the year 1900+x. • When working with "Y" in "ISO week date" based calendars, the year transition occurs on a week boundary and may differ from the calendar year transition. The "Y" designation is used in conjunction with pattern character "w" in ISO year-week calendar. The ISO week date system is effectively a leap week calendar system that is part of the ISO 8601 date and time standard. Similar to "y", negative values for "Y" denote a year before Christ (BC).
Month	All symbols are supported in .hyper extracts except for "l". **Notes:** • The month designations are used in conjunction with "d" for the day number. • In contrast to ICU, .hyper extracts allow values 1–12. Other values cause an error.
Week	All symbols are supported in .hyper extracts except for "W".

Unit of time	Notes
	Notes: • When working with "w", in contrast to ICU, .hyper extracts allow only valid weeks. A year has 52 or 53 weeks (ISO 8601). The DATEPARSE function validates the input. For example, an error occurs for the 53rd week of 2016 because the 53rd week doesn't exist for 2016. • When working with "W", ICU doesn't support this designation, but it's useful for dates like 1st Monday of September.
Day	**Notes:** • When working with "d", in contrast to ICU, .hyper extracts only allow valid day numbers. For example, an error occurs for the 31st of February. • When working with "D", in contrast to ICU, .hyper extracts only allow valid day numbers. For example, an error occurs for the 366th day of 2017.
Hour	Only "h" and "H" symbols are supported in .hyper extracts. **Notes:** • When working with "h", .hyper extract don't allow negative values for this field. Negative values cause an error. • When working with "H", .hyper extracts don't allow negative values for this field. Negative values cause

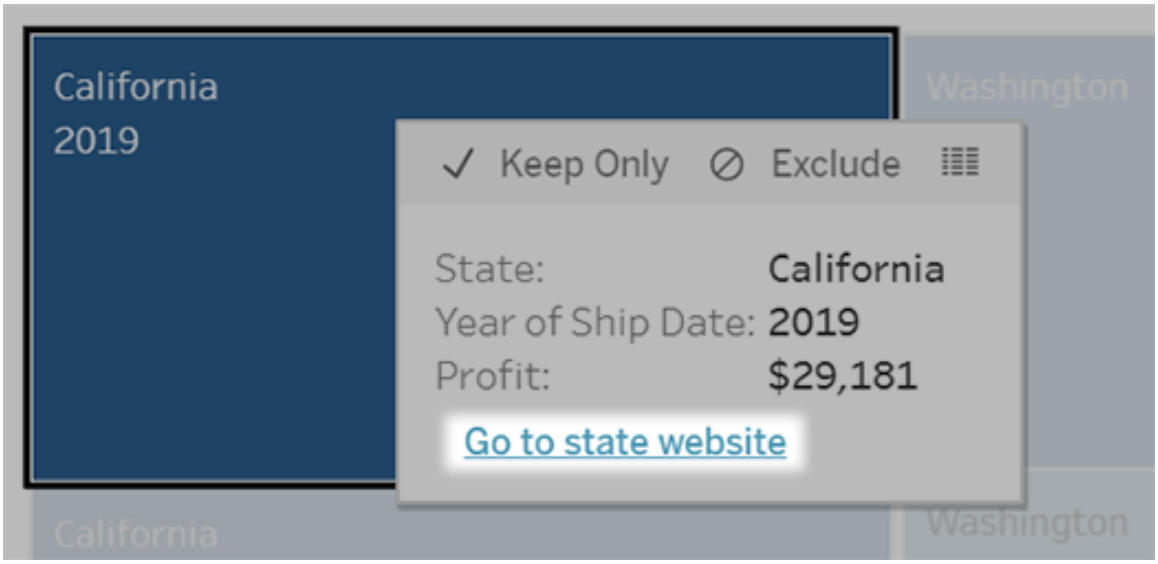
Unit of time	Notes
	an error.
Minute	**Note:** In contrast to ICU, .hyper extracts don't allow negative values for this field. Negative values will cause an error.
Second	**Notes:** • In contrast to ICU, .hyper extracts don't allow negative values for this field. Negative values will cause an error. • When working with "S", `DATEPARSE('ss.SSSS', '12.3456')` returns 1990-01-01 00:00:12:3456 AD.
Quarter	**Note:** In contrast to ICU, .hyper extracts only allow values 1–4. All other values cause an error.
Weekday	**Notes:** • When working with "e" and "ee", in contrast to ICU, .hyper extracts only allow values 1–7. All other values cause an error. • When working with "c..cc", in contrast to ICU, .hyper extracts only allow values 1–7. All other values cause an error.
Period	n/a

URL Actions

A URL action is a hyperlink that points to a web page, file, or other web-based resource outside of Tableau. You can use URL actions to create an email or link to additional information

about your data. To customize links based on your data, you can automatically enter field values as parameters in URLs.

Tip: URL actions can also open in a web page object in a dashboard. See Actions and Dashboards to learn more.



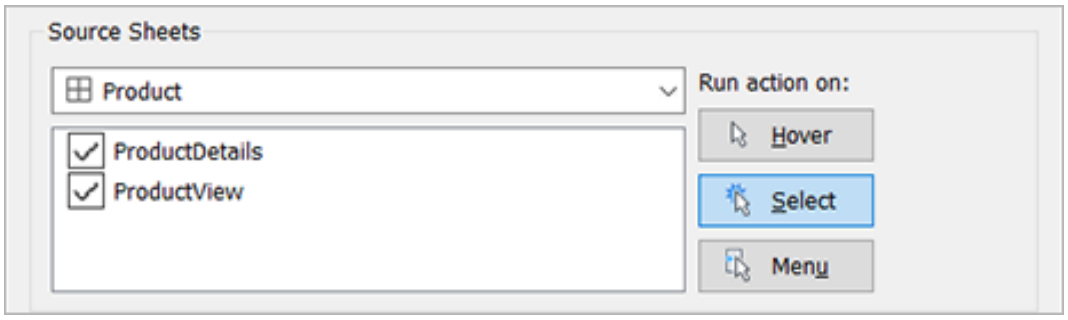
A URL action run from a tooltip menu. The link reflects the action name, not the target URL.

Open a web page with a URL action

1. On a worksheet, select **Worksheet** > **Actions**. From a dashboard, select **Dashboard** > **Actions**.
2. In the Actions dialog box, click **Add Action** and then select **Go to URL**.
3. In the next dialog box, enter a name for the action. To enter field variables in the name, click the **Insert** menu to the right of the **Name** box.

   Note: Give the action a descriptive name, because the link text in the tooltip is the name of the action, not the URL. For example, when linking to more product details, a good name could be "Show More Details".

4. Use the drop-down list to select a source sheet or data source. If you select a data source or dashboard you can select individual sheets within it.



5. Select how users will run the action.

If you choose this option...	The action is run when the user...
Hover	Mouses over a mark in the view. This option works best for highlight actions within a dashboard.
Select	Clicks a mark in the view. This option works well for all types of actions.
Menu	Right-clicks (control-clicks on Mac) a selected mark in the view, then clicks an option in a tooltip (menu). This option works particularly well for URL actions.

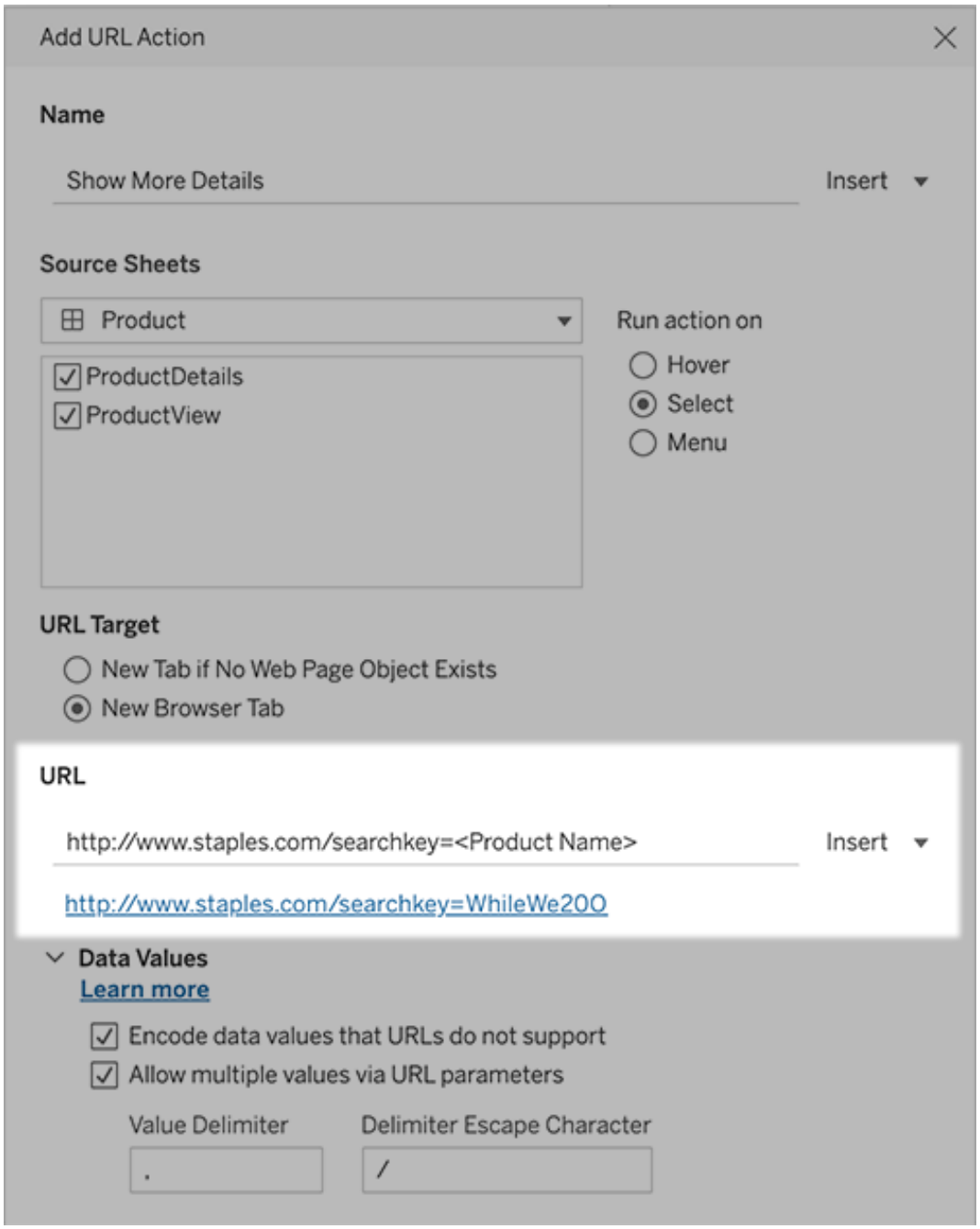
6. For URL Target, specify where the link will open:
   - **New Tab if No Web Page Object Exists** — Ensures that the URL opens in a browser on sheets that lack web page objects. This is a good choice when Source Sheets is set to All or a data source.
   - **New Browser Tab** — Opens in the default browser.
   - **Web Page Object** — (Available only for dashboards with Web Page objects) Opens in the web page object you select.

7. Enter a URL
    - The URL should start with one of the following prefixes:`http`, `https`, `ftp`, `mailto`, `news`, `gopher`, `tsc`, `tsl`, `sms`, or `tel`

      Note: If no prefix is entered, `http://` is automatically appended to the beginning and the URL action will work in Tableau Desktop. However, if a URL action with no prefix is published to Tableau Server or Tableau Cloud, it will fail in the browser. Always provide a fully qualified URL for actions if the dashboard will be published.

      Note: You can specify an ftp address only if the dashboard doesn't contain a web object. If a web object exists, the ftp address won't load.

    - Tableau Desktop also supports local paths like `C:\Example folder-\example.txt`, as well as file URL actions.
    - To enter field and filter values as dynamic values in the URL, click the **Insert** menu to the right of the URL. Be aware that any referenced fields must be used in the view. For details, see Using field and filter values in URLs.



Below the URL you enter is a hyperlinked example you can click for testing.

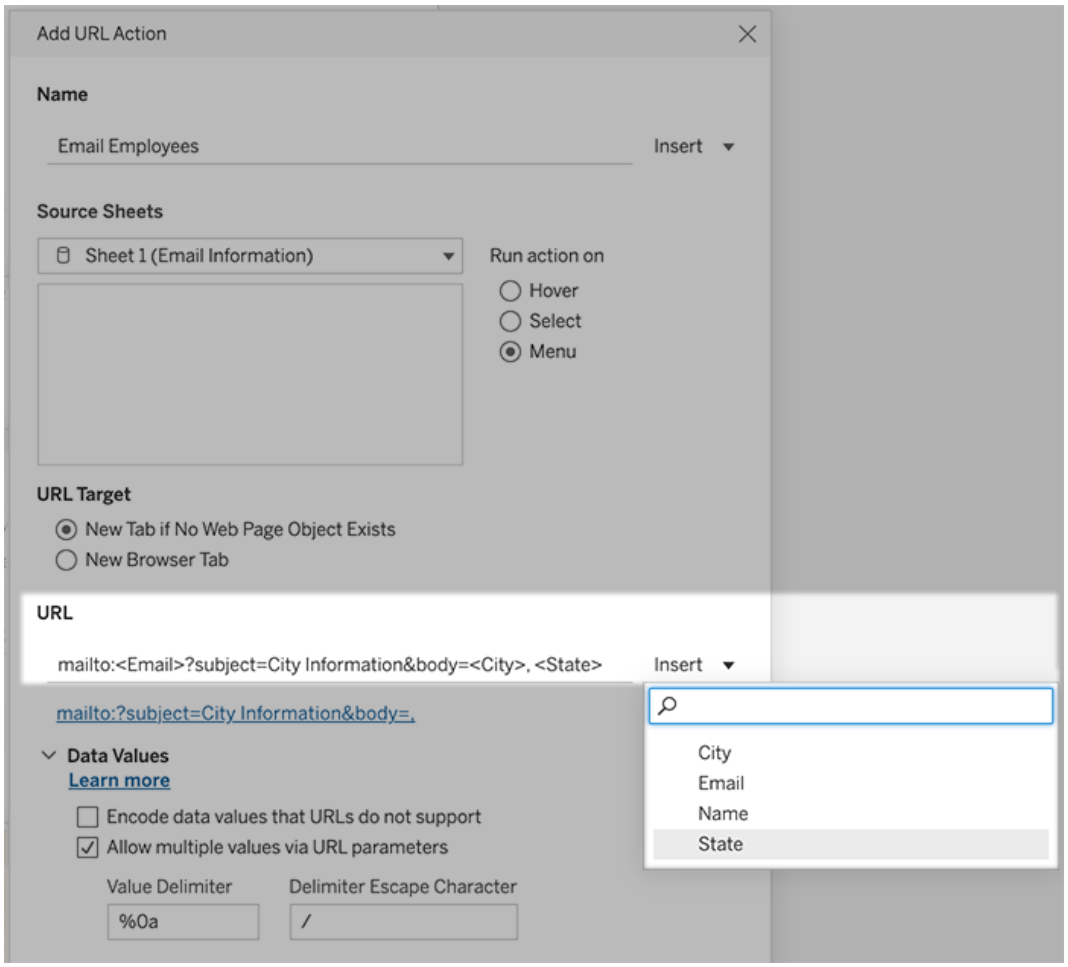
8. (Optional) In the Data Values section, select any of the following options:
    - **Encode Data Values that URLs Do Not Support** — Select this option if your data contains values with characters that browsers don't allow in URLs. For example, if one of your data values contains an ampersand, such as “Sales & Finance,” the ampersand must be translated into characters that your browser understands.
    - **Allow Multiple Values via URL Parameters** — Select this option if you are linking to a web page that can receive lists of values via parameters in the URL. For example, say you select several products in a view and you want to see each product’s details hosted on a webpage. If the server can load multiple product details based on a list of identifiers (product ID or product name), you could use multi-select to send the list of identifiers as parameters.

      When you allow multiple values, you must also define the delimiter escape character, which is the character that separates each item in the list (for example, a comma). You must also define the Delimiter Escape, which is used if the delimiter character is used in a data value.

Create an email with a URL action

1. On a worksheet, select **Worksheet** > **Actions**. From a dashboard, select **Dashboard** > **Actions**.
2. In the Actions dialog box, click **Add Action**, and select **Go to URL**.
3. In the Source Sheets drop-down list, select the sheet that contains the field with the email addresses you want to send to.
4. In the URL box, do the following:
    - Type **mailto:**, and click the **Insert** menu at right to select the data field that contains email addresses.
    - Type **?subject=**, and enter text for the Subject line.
    - Type **&body=**, and click the **Insert** menu at right to select the fields of information that you want to include in the body of the email.

    In the example below, the “Email” field contains the email addresses, the subject is “City Information”, and the body text of the email consists of the city and state data that is associated with the email address.



:

5. (Optional) Display data from your workbook in the body of your email as a vertical list instead of the default horizontal list. For example, suppose you have a horizontal list of cities, such as Chicago, Paris, Barcelona, which you would rather display vertically, like this:

   Chicago

   Paris

   Barcelona

   To make the list vertical, in the Data Values section, do the following:

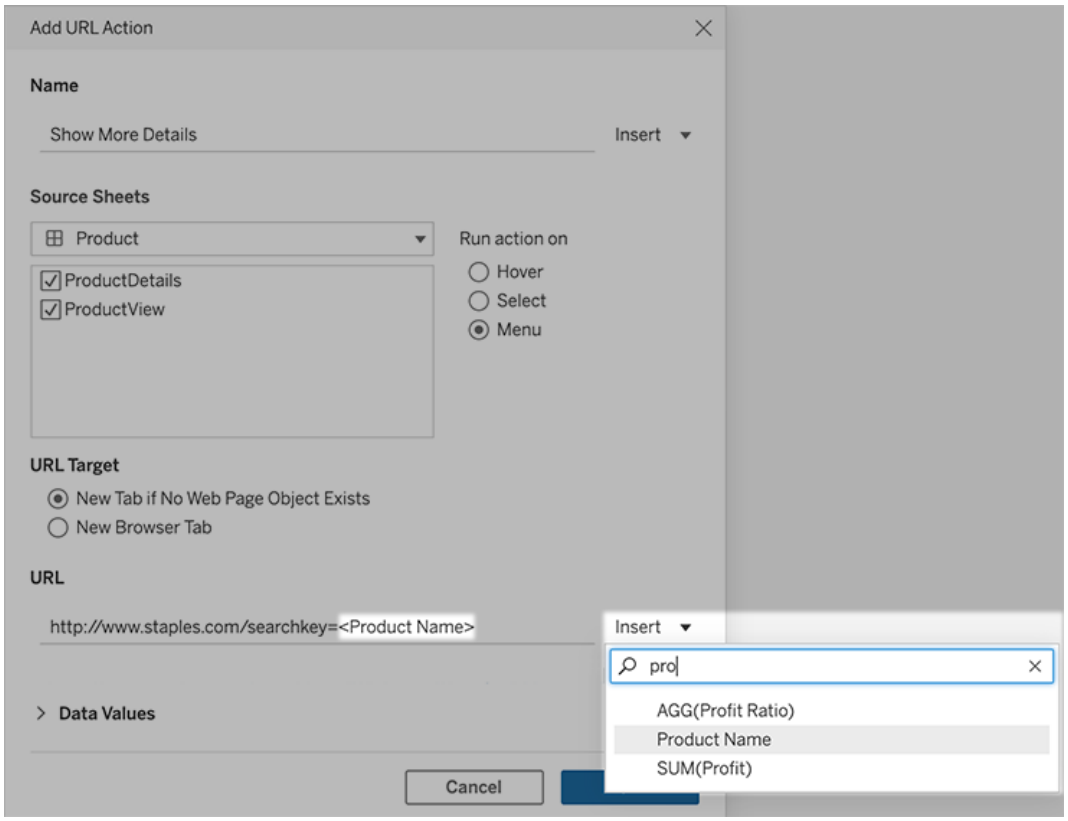
- Deselect **Encode Data Values that URLs Do Not Support**
- Select **Allow Multiple Values via URL Parameters**.
- Type **%0a** in the **Value Delimiter** text box to add line breaks between each item in the list. (These are the URL-encoded characters for a line break.)

Using field and filter values in URLs

When users trigger URL actions from selected marks, Tableau can send field, filter, and parameter values as variables in the URL. For example, if a URL action links to a mapping website, you could insert the address field to automatically open the currently selected address on the website.

1. In the Edit URL Action dialog box, begin typing the URL for the link.
2. Place the cursor where you want to insert a field, parameter, or filter value.
3. Click the **Insert** menu to the right of the text box and select the field, parameter, or filter you want to insert. The variable appears within angle brackets. You can continue adding as many variables as you need.

> **Note:** Any referenced fields must be used in the view. Otherwise, the link won't display in the viz, even if it functions when you click Test Link.



Including aggregated fields

The list of available fields includes only non-aggregated fields. To use aggregated field values as link parameters, first create a related calculated field, and add that field to the view. (If you don't need the calculated field in the visualization, drag it to Detail on the Marks card.)

Inserting parameter values

When inserting parameter values, URL actions send the Display As value by default. To instead send the actual value, add the characters `~na` after the parameter name.

For example, say you have a parameter that includes IP addresses, with Actual Value strings such as 10.1.1.195 and Display As strings with more friendly values such as `Computer A (10.1.1.195)`. To send the actual value, you'd revise the parameter in the URL to look like this: `http://<IPAddress~na>/page.htm`.

Create a Subscription to a View or Workbook

Subscriptions email you an image or PDF snapshot of a view or workbook at regular intervals—without requiring you to sign in to Tableau Server.

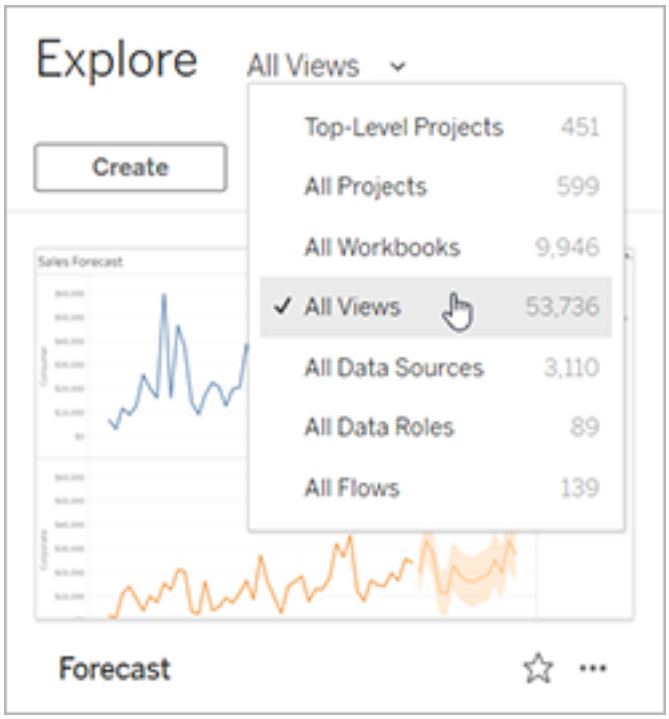
Note: Administrators determine whether subscriptions are turned on for a site.

If Tableau Catalog is turned on for a site, administrators can also determine whether subscription emails include relevant upstream data quality warnings. Tableau Catalog is available as part of the Data Management offering. For more information, see About Tableau Catalog.

Set up a subscription for yourself or others

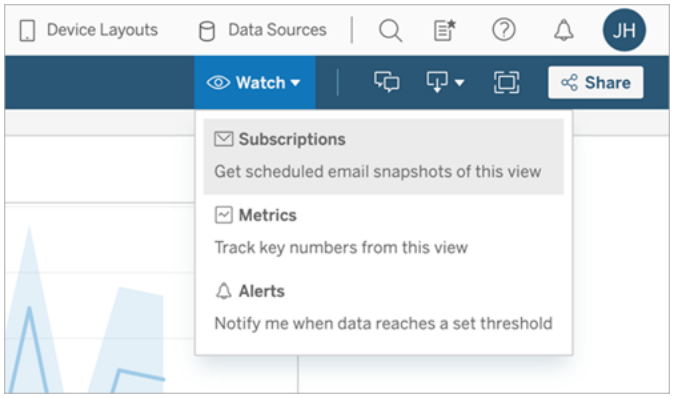
When you open a view in Tableau Server, if you see a subscription icon (✉+) in the toolbar, you can subscribe to that view or to the entire workbook. You can subscribe other users who have permission to view the content if you own a workbook, if you are a project leader with an appropriate site role, or if you are an administrator.

1. From the Explore section of your site, select **All Workbooks** or **All Views**, or open the project that contains the view you want to subscribe to.



2. Open a view either directly, or after opening the containing workbook.

3. On the view toolbar, select **Watch** > **Subscriptions**.



4. Add the Tableau users or groups you want to receive the subscription. To receive a subscription, users must have the View and Download Image/PDF permissions and their accounts must also have email addresses.

If you own the workbook, select **Subscribe me**.

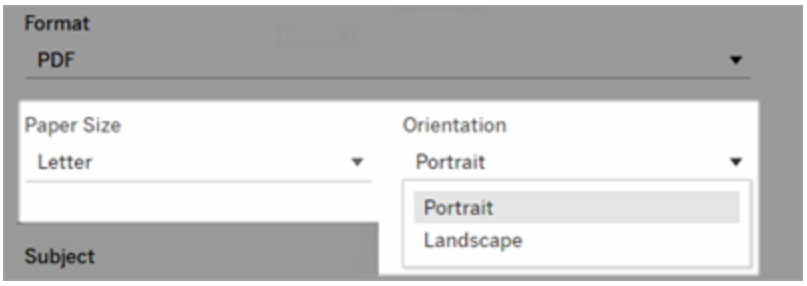
Notes:

- When you subscribe a group, each user is added individually at the time the subscription is created. If more users are added to the group later, you must re-subscribe the group for those new users to receive the subscription. Likewise, users later removed from the group will not have their subscriptions removed automatically unless their permissions to the subscribed view are removed.
- You can't subscribe a group set.

5. Choose whether subscription emails include the current view or the entire workbook. If the view contains data only when high-priority information exists, select **Don't send if view is empty**.

6. Choose the format for your snapshot: as a PNG image, a PDF attachment, or both.

   - If PDFs, choose the paper size and orientation you'd like to receive.



7. To clarify subscription emails, customize the subject line, and add a message.

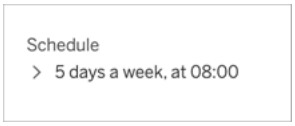
   Note: To update the subscription message, you must unsubscribe from the existing subscription and create a new subscription with a different message. For more information, see Update or unsubscribe from a subscription.

8. When the workbook uses one data extract on a published connection, you can pick a frequency:

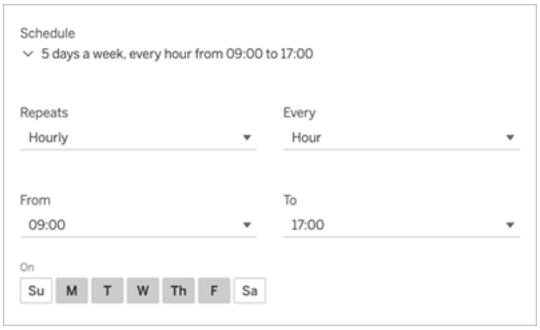
- **When Data Refreshes**: Sends only when data in the view or workbook is refreshed by running refresh schedules.
- **On Selected Schedule**: Pick a schedule for the subscription.

9. If frequency is not set to When Data Refreshes, pick a schedule:

   - Choose from subscription schedules established by your administrator.
   - For sites with custom schedules enabled, click the drop-down arrow to the left of the current settings.

   Then specify a custom schedule that sends subscription emails whenever you wish. (The precise delivery time may vary if server load is high.)

   To change the time zone, click the Time Zone link to go to your account settings page.

10. Click **Subscribe**.

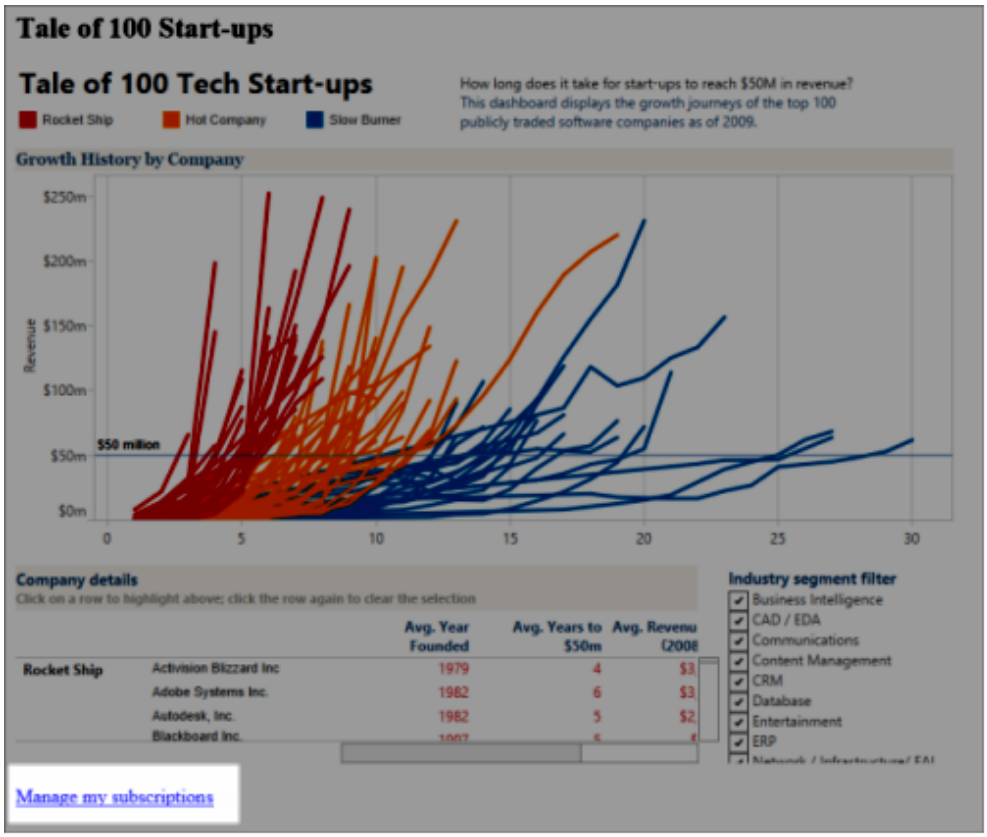
When you receive a subscription email, you can select the image (or the link in the message body for PDF subscriptions) to be taken to the view or workbook in Tableau Server.

Update or unsubscribe from a subscription

You can unsubscribe from an existing subscription, or make changes to a subscription's format, schedule, subject, or empty view mode.

1. Access your Tableau Server account settings by doing one of the following:

   - Click **Manage my subscriptions** at the bottom of a subscription email.



   - Sign in to Tableau Server. At the top of the page, select your user icon, and then

select **My Content**.

2. Click **Subscriptions**.

3. Select the check box next to the view you want to unsubscribe from, click **Actions**, and then click **Unsubscribe**, or select the subscription option you'd like to change.

Resume or delete suspended subscriptions

Sometimes, subscriptions fail because of an issue with the workbook or a problem loading the view. If a subscription fails more than five times, you'll receive a notification email that your subscription has been suspended. There are a few ways to resume a suspended subscription if you're a subscription owner or administrator:

- From the My Content area of Tableau web pages, an icon appears in the Last update column to indicate that the subscription is suspended. Select **...** > **Resume Subscription** to resume.
- From the Subscriptions tab of the affected workbook, an icon appears in the last update column to indicate that the subscription is suspended. Select **...** > **Resume Subscription** to resume.

You'll receive an email notification when the subscription is working again.

See also

Change subscription settings in the Tableau Desktop and Web Authoring Help.

Project-level administration in the Tableau Cloud Help, to learn which site roles allow full Project Leader capabilities.

Use Custom Views

A custom view is a shortcut to a specific state of interaction, such as filter selections and sorting, for a published viz. Custom views don't impact the underlying content. They're a good option if you find yourself adjusting the same filters or zooming into the same data every time you look at a viz.

Custom views aren't the same as web editing, which changes the underlying published content itself. See Edit Tableau Views on the Web.

If the custom views are specifically for filter settings, consider using embedding filter parameters in a shared URL. See Filter a published dashboard by editing the URL from The Data School.

Notes on custom views

- A custom view doesn't modify the content it's built on.
- Deleting the original content deletes its custom views.
- If the original content is updated or republished, the custom view is also updated.

  Tip: Some changes to the original content can break the custom view. See Maintain Content with Custom Views for best practices on modifying content with custom views.

- If a user is removed from the site, any shared custom views they owned are also lost.
- Subscriptions and data-driven alerts based on custom views may be more fragile than those based on the original content.

Note: As of the 2022.3 release, Tableau replaced user names in the custom view URL with IDs. Bookmarked URLs still work but are redirected to the new URL schema. This change is to add more company and user data protection.

Create a custom view

Start by navigating to the individual view. Make whatever changes you want to capture in the custom view, such as selecting marks, filtering data, or changing sorts.

1. When you're ready to save the changes you've made as a custom view, select **Save Custom View** from the toolbar.

   > **Note**: The **Save Custom View** button appears in the toolbar after any changes are made to the current view.

2. In the **Save Custom View** dialog, enter a name for the custom view.
3. (Optional) Select **Make it my default**.
4. (Optional) Select **Make visible to others**. This makes the custom view available to everyone who can see the original content. However, there are several instances when this option isn't available:
   - The user is a Viewer site role.
   - The site's User Visibility setting is set to Limited.
   - The permission capability Share Customized is denied on the workbook.
5. Click **Save**.

Find a custom view

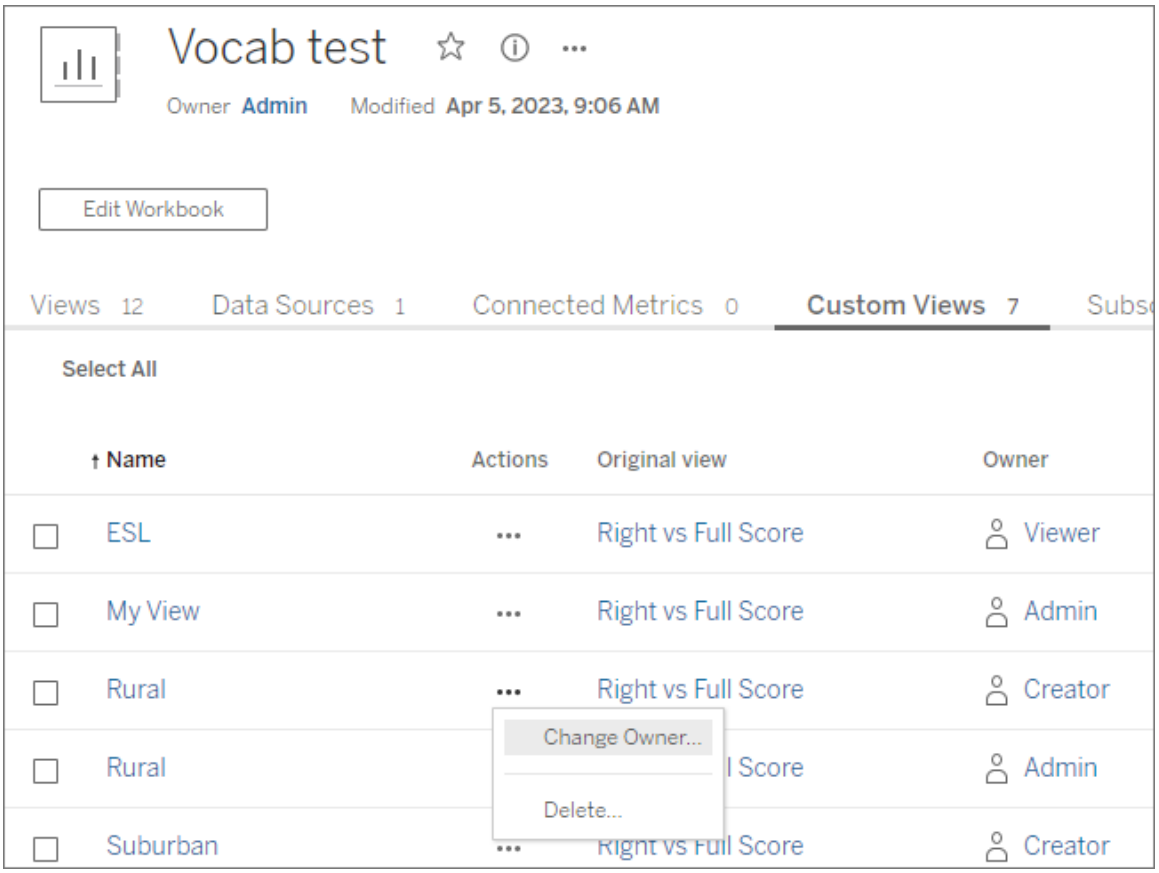
From a view

When you're looking at a viz, you can change to a different custom view by selecting the View icon in the toolbar. If there's room in the toolbar, the name of the custom view you're looking at is shown.

Any custom views you've made, and all visible custom views made by other users, appear in the list.

From the workbook

When you're looking at content at the workbook level, use the **Custom Views** tab to see all the available custom views for that workbook.



Set a default custom view

After you've found or made a custom view, you make it the default you see when you open that viz.

1. Select the **View** icon in the toolbar.
2. Check the **Set this view a your default** option.
3. Close the dialog to save.

The next time you open that viz, you'll land on that custom view.

Share a custom view

By default, custom views are private and only appear for the user who created them.

Note: Users with a Viewer site role can't make custom views visible to others. However, they can share a custom view by copying and sharing the URL.

Users with a site role of Explorer or higher can set a custom view as visible to others. This setting allows anyone with access to the original content to see the custom view.

To change an existing private custom view to be visible to others (or to make a visible view private):

1. Select the **View** icon in the toolbar.
2. Toggle the eye icon to the view you want to share to the desired state.
3. Close the dialog to save.

The eye with a slash  indicates the view is private to you. The eye  indicates the view is visible to others.

Delete a custom view

To delete a custom view:

1. Select the **View** icon in the toolbar.
2. Select the trash icon of the view you want to delete.
3. Confirm that you want to delete the view.

Take care when deleting

If you're the owner of a custom view that's visible to others, remember it's deleted for everyone if you delete it.

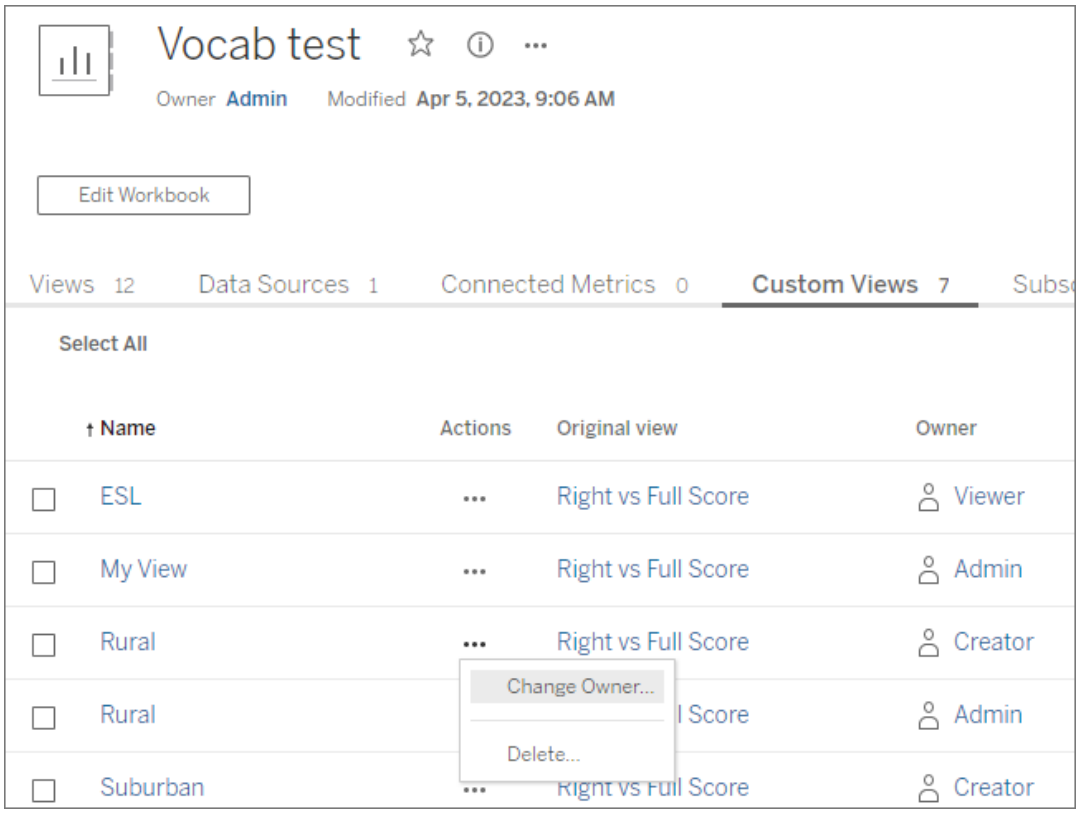
Deleting a custom view also deletes any subscriptions or data-driven alerts based on that custom view.

Manage custom views

Administrators can change ownership for custom views and delete custom views created by other users.

Custom views can be managed for a piece of content or for a specific user.

1. Go to the Custom Views tab for the workbook or user.
2. Use the action menu to change the owner or delete the custom view.



Tip: It's a best practice to change ownership of any custom views belonging to a user before removing them from the site. Deleting a user also deletes their custom views, including public views others may be using.

Safely change content with custom views

If you need to modify a view that has custom views (or the data source the view is built on), be aware that certain changes can break custom views. For more information, see Maintain Content with Custom Views.

Publish Views to Salesforce

Bring your views from Tableau Cloud or Tableau Server right to your Salesforce ecosystem by publishing views to a CRM Analytics app or Salesforce Lightning page.

For more information, see Publish Tableau Content to CRM Analytics in Salesforce Help.

Prerequisites

See a complete list of prerequisites in Salesforce Help, including required licenses, account setup, and permissions.

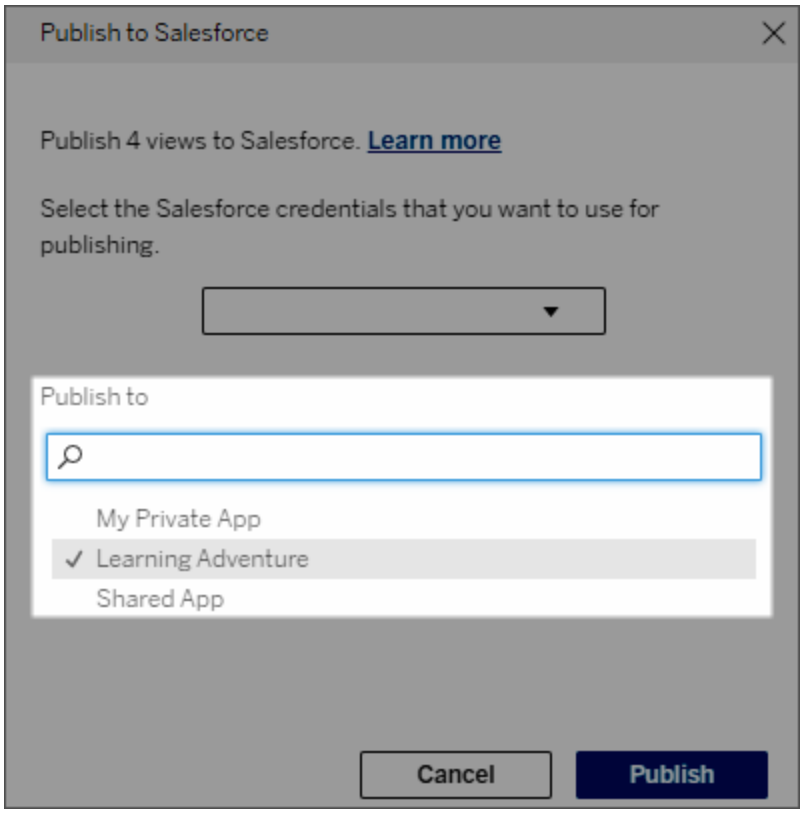
Publish a view to Salesforce

Select one or more views, including dashboards, sheets, and stories. Then, choose a destination from a list of CRM Analytics apps that you have access to edit or manage.

1. Select the view that you want to publish to Salesforce.

   Note: You can select a maximum of 25 views at a time to publish to Salesforce.

2. For **Actions**, select **Publish to Salesforce**.
3. Select the Salesforce credentials that you want to use for publishing.
4. Select the destination app that you want to publish to. You can only see apps that you can edit or manage with the signed-in Salesforce user.



5. Click **Publish**.

Who can see the published view in Salesforce?

When you publish a view to Salesforce, anyone with access to the selected CRM Analytics app or Lightning page can see that the content exists. However, only those signed in with existing Tableau permissions can see the view.

Configure Tableau Lightning Web Components and Single Sign-On (SSO) with Token Authentication

Tableau Lightning web components (LWC) allow Salesforce customers to drag and drop Tableau views and Tableau Pulse metrics onto Salesforce Lightning pages.

- The Tableau View component allows you to add embedded views from Tableau Cloud or Tableau Server.

- The Tableau Pulse component allows you to add embedded Tableau Pulse metrics from Tableau Cloud.

Tableau LWC seamless authentication allows you to view Tableau content using connected app trusted tokens without signing in. Seamless authentication is optional for Tableau View LWC, and is required for Tableau Pulse LWC.

Important:
• Salesforce Console apps do not support the use of Tableau Lightning web components.
• Case Record pages do not support the use of Tableau Lightning web components with Chatter emails.

Add Trusted URL

The Tableau view or Pulse URL that you want to add to your Lightning page must be added as a Trusted URL.

1. From your Salesforce app, select the gear in the top-right corner, and then select **Setup**.
2. On the left navigation pane, enter “Trusted URLs” in the **Quick Find** search bar.
3. Select the **Trusted URLs** settings page.
4. Select **New Trusted URL**.
5. Enter an **API Name** and **URL**, following the instructions on the settings page. **Note**: The URL must begin with https://
6. For CSP Context, select **All**.
7. For CSP Directives, check all boxes.
8. Select **Save**.

Turn on seamless authentication for Tableau LWCs

Tableau View and Tableau Pulse LWCs are available in the Lightning App Builder without any configuration. However, Tableau Pulse LWC requires token authentication to function.

Configure Salesforce settings

The following steps only need to be completed one time by a Salesforce admin:

1. From your Salesforce app, select the gear in the top-right corner, and then select **Setup**.
2. On the left navigation pane, enter “Tableau” in the **Quick Find** search bar.
3. Select the **Tableau Embedding** settings page.
4. Select the checkbox for **Turn on token-based single sign-on authentication**.

   > **Note**: This box must be checked to configure the Tableau Pulse LWC. For the Tableau View LWC, you can choose not to set up token authentication and instead sign in manually when the component loads.
   >
   > To use Tableau View LWC on Mobile, you must turn on token-based authentication and set up seamless authentication.

5. For **Select Tableau User Identity field**, set an org-level user field to authenticate the user in Tableau. You must select the Salesforce user field that corresponds with the Tableau username. The dropdown shows the field value for the current user, or null if no value is defined. If none of the user fields match the Tableau username, select an empty field (for example, Federation ID or a custom field). Then, populate the empty field with the Tableau username for your users.

   > **Note**: The Tableau User Identity field setting applies to all users and doesn’t need to be set on an individual basis.

6. Save your changes.
7. If you're the Tableau admin, keep the Salesforce settings **Tableau Embedding** tab open while you configure Tableau settings in the next section. If you aren’t the admin, share the **Issue URL** and **JWKS URI** with your Tableau admin.

Configure Tableau settings

In one tab, open the Tableau Embedding settings page in your Salesforce org. In another tab, go to your Tableau site and follow these instructions to set up the Connected App.

For Tableau Server, follow these steps:

1. As a Tableau Server admin, sign in to the Tableau Services Manager (TSM) web interface.
2. Navigate to **User Identity & Access**, and then select the **Authorization Server** tab.
3. Select the checkbox for **Enable OAuth access for embedded content**.
4. Enter the **Issue URL** and **JWKS URI**, which you can find on the Salesforce org Tableau Embedding settings page. Use the **Copy** button on the Salesforce org settings page to copy the Issuer URL value, and then paste it into the TSM web interface. Repeat this process for the JWKS URI value.

   > **Note**: The JWKS URI field is marked as optional in the TSM web interface, however this value is required to use Tableau LWC seamless authentication.

5. Select **Save Pending Changes**.
6. Select **Pending Changes** in the upper-right corner of the page, and then select **Apply Changes and Restart** to stop and restart Tableau Server.

For more information, see Register your EAS with Tableau Server.

For Tableau Cloud, follow these steps:

1. Open the Tableau **Settings** page, and then choose the **Connected Apps** tab.
2. From the New Connected App dropdown, select **OAuth 2.0 Trust**.
3. On the Create Connected App dialog, enter the **Issue URL** and **JWKS URI**, which you can find on the Salesforce org Tableau View Embedding settings page. Use the **Copy** button on the Salesforce org settings page to copy the Issuer URL value, and then paste it into the Tableau settings page. Repeat this process for the JWKS URI value.

   Important: The Create Connected App dialog notes the JWKS URI field as optional, however this value is required to use Tableau LWC seamless authentication.

4. Select the checkbox for **Enable connected app**.
5. Select **Create**.

   > **Note:** The Connected App is named External Authorization Server.

For more information, see Register your EAS with Tableau Cloud.

If you want to create a host mapping for this site, leave the Connected Apps tab open. You can use the URL for this page and the Copy Site ID button to populate the host mapping fields in the following section.

Set up or edit host mapping

Follow these steps to create or edit a host mapping.

Tip: The Tableau User Identity field setting applies to all users and doesn't need to be set on an individual basis.

Create a new host mapping

1. From your Salesforce app, select the gear in the top-right corner, and then select **Setup**.
2. On the left navigation pane, enter "Tableau" in the **Quick Find** search bar.
3. Select the **Tableau Embedding** settings page.
4. From the Tableau Host Mapping section, select **Create New**.
5. Fill in the host mapping details:
    a. Tableau site URL: Enter a URL for the Tableau site that you want to map. The URL should contain the site name, unless it's an on-prem installation using the Default site. **Note**: If you want to create a host mapping for this site, leave the Connected Apps tab open. You can use the URL for this page and the Copy Site ID button to populate the host mapping fields in the following section.
    b. Tableau site ID: Enter the site ID for the Tableau site that you want to map. You can use the Copy Site ID button on the Connected App settings page or on the Share dialog.
    c. Tableau site host type: Select Tableau Cloud or Tableau Server.
6. Select **Save**. Or, if you want to return to the Tableau Embedding settings page without saving, select **Cancel**.

Edit a host mapping

You can update the site ID and host type for an existing mapping. If you need to change the site URL, delete the existing mapping, and then create a new one with the correct URL.

1. From the Salesforce app Tableau Embeddings settings page, select Edit next to an existing host mapping.
2. Edit the **Tableau site ID** or **Tableau site host type** fields as needed.

3. Select **Save**. Or, if you want to return to the Tableau Embedding settings page without saving, select **Cancel**.

Add Tableau LWCs to a Lightning page using Lightning App Builder

Tableau LWCs are available on App, Home, and Record Lightning pages only. For more information about Lightning page types and using the Lightning App Builder, see Lightning App Builder in Salesforce Help.

> **Note:** Case Record pages do not support the use of Tableau Lightning web components with Chatter emails.

To add a Tableau View or Tableau Pulse LWC to an existing Lightning page, follow these steps:

1. Navigate to the Lightning page that you want to edit.
2. Select the gear icon in the top right.
3. Select Edit Page.
4. Proceed to the **Add a Tableau LWC to a Lightning page** section below.

To add a Tableau View or Tableau Pulse LWC to a new Lightning page, follow these steps:

1. From your Salesforce app, select the gear in the top-right corner, and then select **Setup**.
2. On the left navigation pane, enter “Lightning App Builder” in the **Quick Find** search bar.
3. Select the **Lightning App Builder** setup page.
4. Select **New**.
5. Select the page type that you want to create. Tableau LWCs are available on App, Home, and Record pages.
6. Select **Next**.
7. Enter a name and select a layout for the new page, and then select **Done**.

Add a Tableau LWC to a Lightning page

1. From the Components list on the left side of the page, drag and drop the Tableau View or Tableau Pulse component onto the page.

2. Configure the LWC:
    - Configure a Tableau View Lightning Web Component
    - Configure a Tableau Pulse Lightning Web Component

Save and activate the page

1. When you've finished adding and configuring a Tableau View or Tableau Pulse LWC, select **Save**.
2. If you've created a new page, you are prompted to activate the page so that it's visible to users. Select **Activate**.
3. On the **Page Settings** tab of the Activation page, enter a name, choose an icon, and select your visibility preference.
4. (Optional) On the **Lightning Experience** tab of the Activation page, you can add the page to various Lightning Experience Apps.
5. (Optional): On the **Mobile Navigation** tab of the Activation page, you can add the page to the Mobile Navigation Menu.
6. Select **Save**.

Embed multiple Tableau views

You can embed more than one Tableau view on a Salesforce Lightning page as long as all views come from the same site. Tableau only supports a single session, and that session is specific to the site. The most recently granted session will wipe out the previous one.

To embed Tableau views from multiple sites, you must create a separate Lightning page that is site-specific.

Tableau LWC single sign-on for Mobile

Note: Tableau View and Pulse LWCs are available on iOS 17.2.1+.

Consider the following best practices to prevent issues for mobile users:

- Lightning page type: Mobile users can access App Pages and Record Pages, but not Home Pages.

> **Note:** Record Pages must be associated with a specific type of record.

- If you use the same page for both desktop and mobile users, select **Activation** to verify that the Lightning page is set as the Org Default for both desktop and phone form factors.
- Consider creating separate Lightning pages for desktop and mobile to provide a tailored visual experience. The height for the Tableau View component is fixed and won't dynamically adjust to different screen sizes.
- To add a scroll bar to a view, select **Show Toolbar** from the Tableau View component properties pane.
- For the App Page type, select **Activation**, and then select the **Lightning Experience** tab. Add your page to the LightningBolt list to make the page easier to find on mobile.
- On mobile, iOS blocks cross-site traffic by default. Open your mobile settings, select Salesforce settings, and then turn on **Allow Cross-Website Tracking**. For more information, see Enable cross-website tracking.

Troubleshooting Tableau View LWC seamless authentication

Verify the Salesforce and Tableau configuration

1. Verify that the **Issuer URL** and **JWKS URI** values match in both Salesforce and Tableau Settings and that JWKS URI ends with **id/keys**.
    - For Tableau Cloud, open the Salesforce Settings Tableau Embedding page in one tab. In another tab, open the Tableau Settings Connected Apps tab. On the Connected Apps tab, select **External Authorization Server**, and then select **Edit**. Verify that the **Issuer URL** and **JWKS URI** values match and that the JWKS URI ends with id/keys.
    - For Tableau Server, open the Salesforce Settings Tableau Embedding page in one tab. Then, sign in to the Tableau Services Manager (TSM) web interface, navigate to **User Identity & Access**, and then open the **Authorization Server** tab.
2. Verify Host Mapping: If you've saved a host mapping, verify that it has the correct site ID and host type.

Verify the JWT token

In the Tableau View Lightning web component property editor, select Debug Mode to verify that the JWT token is working as expected.

1. Open the console logs and copy the token.
2. Go to the jwt.io website and paste the token into the **Encoded** field.
3. Verify the following:
   - The subject (“sub”) matches the Tableau username.
   - For Tableau Cloud, the audience (“aud”) is “tableau+SiteID”.
     For Tableau Server, the audience (“aud”) is “tableau”.
   - The scope (“scp”) includes both “tableau:views:embed” and “tableau-:insights:embed”.
   - The issuer (“iss”) EAS server is accurate.

Verify page activation

Sometimes, a user creates a Lightning page, but it hasn’t been activated or assigned anywhere, so users can’t find it. Select **Activation** to verify that the Lightning page is set as the Org Default for the intended form factors.

Tip: When debugging, it’s helpful to drag and drop a Rich Text component onto your page. Add a brief description of the page type and the view URL that you’re trying to embed. This allows you to be sure that the page being viewed by the end user is the page that the admin is editing.

Confirm that Tableau View LWC is working without seamless authentication (Tableau View LWC only)

1. From the Tableau View pane on your Lightning page, clear the checkbox for **Default Authentication Token**, and then save the changes.
2. If you’re signed in to Tableau in another tab, sign out. Ensure that navigating to the View URL redirects you to the Tableau sign-in page. Do not sign in.
3. Navigate to the Lightning page. The Tableau View LWC should display a **Sign in to Tableau** button.
4. Select **Sign in to Tableau**, and then enter your Tableau credentials to sign in.
   Note: If the view doesn’t load, this indicates a broader issue with authenticating to Tableau.

Error: LWC component version no longer supported (Tableau View LWC only)

To resolve this error, follow these steps:

1. In the Components list, search for “Tableau”, and then drag and drop a new **Tableau View** component onto the page.
2. Copy all properties from the Tableau View pane for the old component over to the new component.
3. Select the delete icon on the old component.

Error: To enable Tableau Pulse LWC, please reach out to your Salesforce admin to configure seamless authentication for Tableau (Tableau Pulse LWC only)

To resolve this error, follow the steps on this page to **Turn on seamless authentication for Tableau LWCs**.

See Also

Troubleshoot Connected Apps

Register EAS to Enable SSO for Embedded Content

Interact with Data in Tableau

This tutorial walks you through some of the basics of viewing and interacting with data visualizations, or views, in Tableau Server.

Tableau is a tool that lets you interact with published visualizations to explore insights, ask questions, and stay on top of your data. Here’s how to get started.

Go ahead. It’s safe to click around

Tableau is built for interaction. What you do to a visualization changes how it looks for you, just for now.

Others will still see the visualization as it originally appeared. And the data used to build it stays the same, too.

1: What is a Tableau Site?

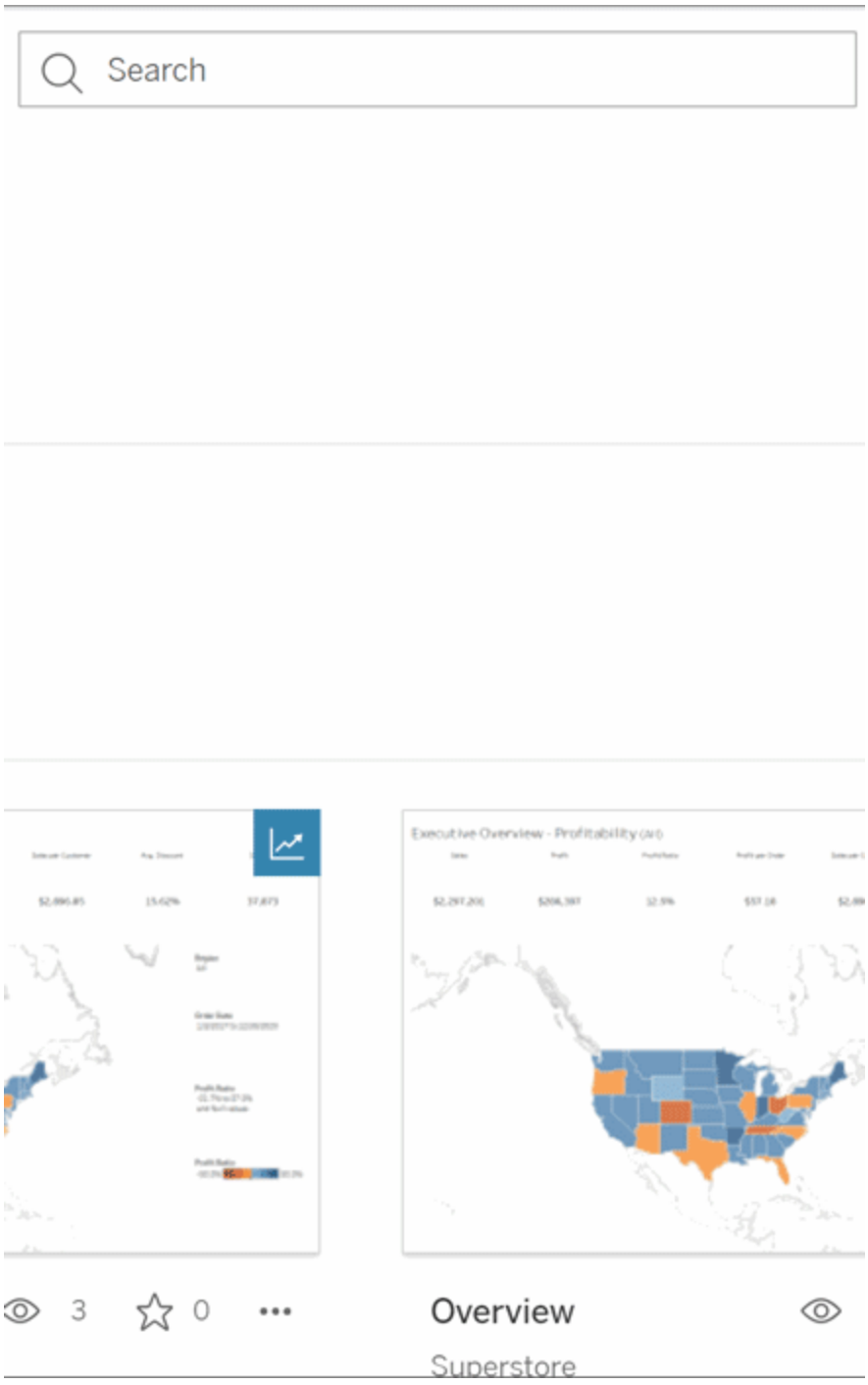
A Tableau Site is a place for your team to share data and data visualizations with each other. You can explore what they've published and made available to you.

When you sign in to your Tableau site, you'll land on the home page.

2: Search for a viz

Tableau calls visualizations on a site Views. Use search to find views or workbooks (a package of views in a single file).

Search results will show all the different content types relevant to your query.



You can select See All for all search results if the views in the quick search aren't what you were looking for, or use the Explore page to browse. There you'll see all the different types of content a Tableau site can host.

3: Interact with Content

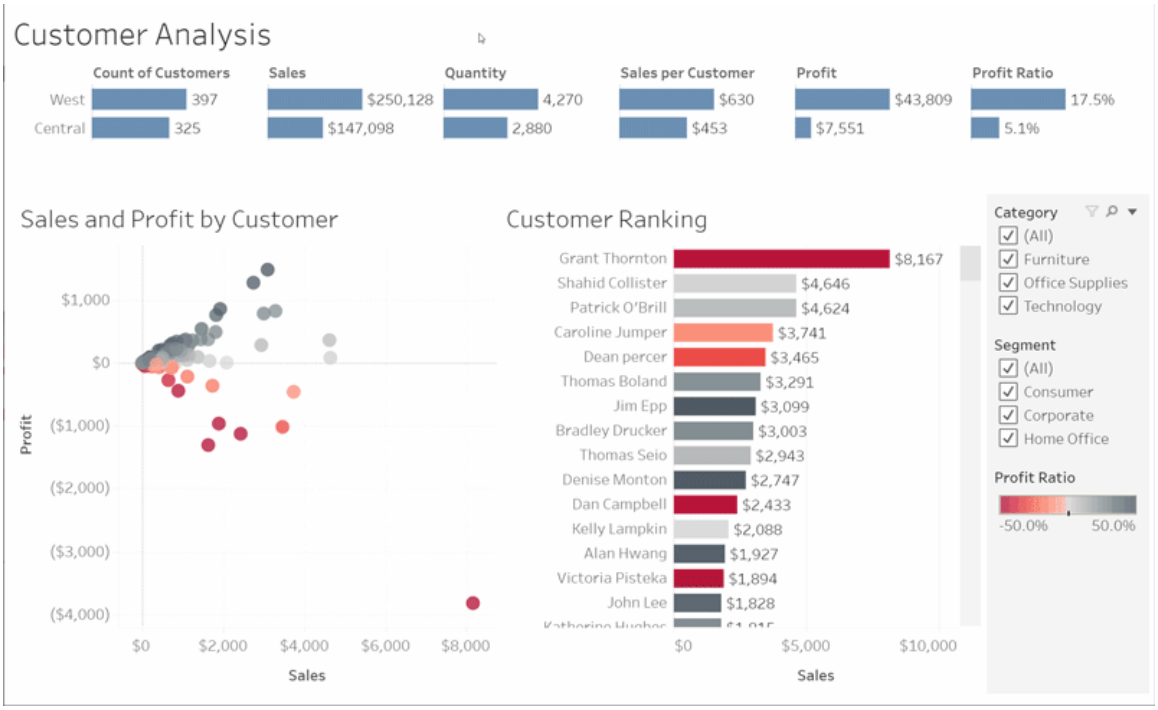
A published view is a canvas for you to interact and understand your data. Remember, you won't hurt or change the underlying data, or change what others see.

Here are some of the tools in your toolbox to find data insights.

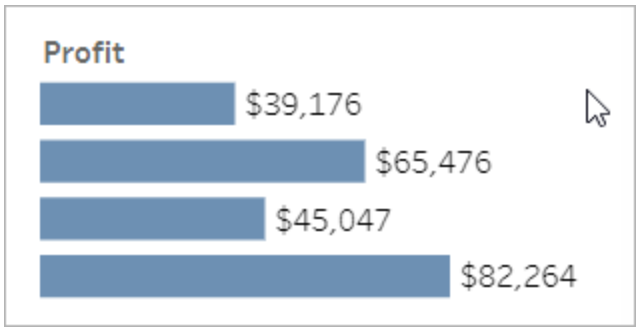
See Details and Sort Data

Now that you know you can click on the data, let's check it out.

As you move the mouse across a view, you might see tooltips that reveal details about each data point, or mark. You can also select multiple marks.

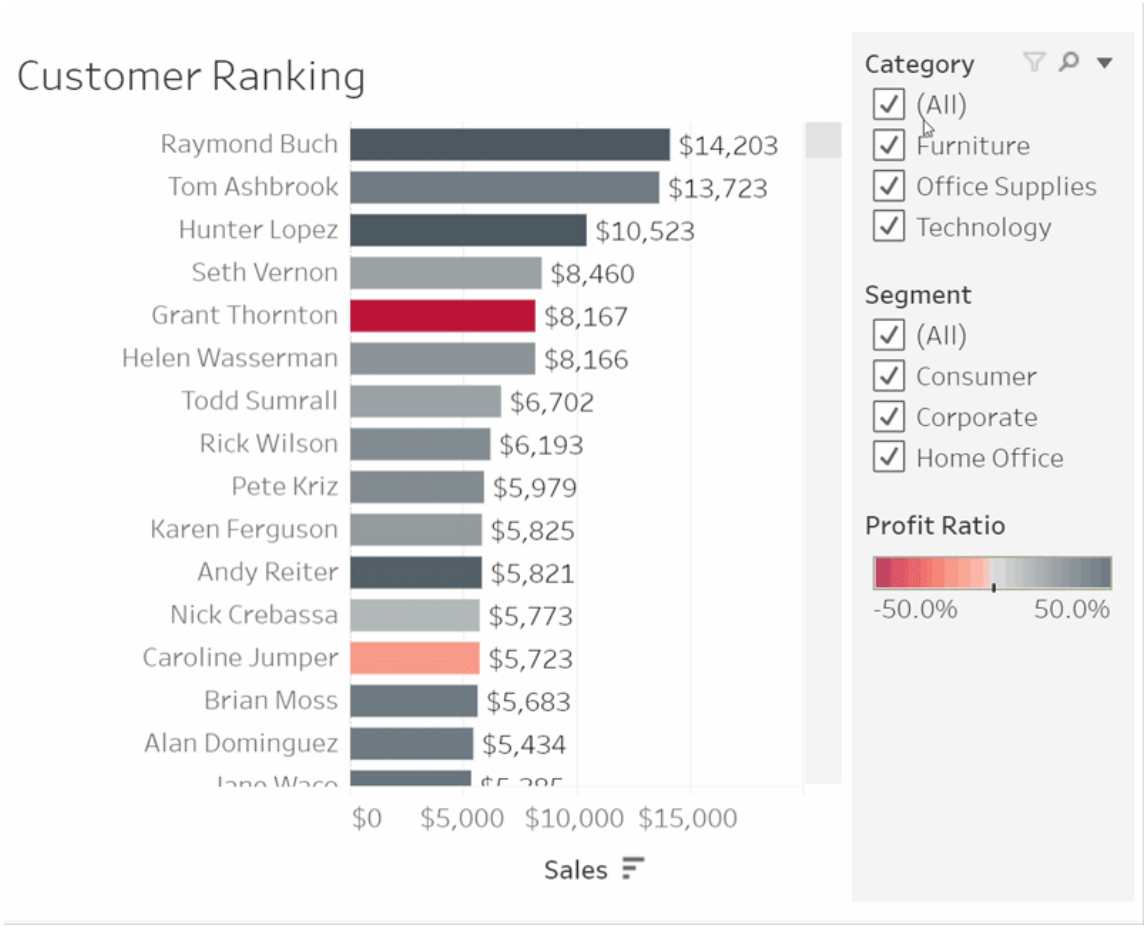


Sort tables alphabetically or numerically by hovering over a column header and clicking the sort icon.



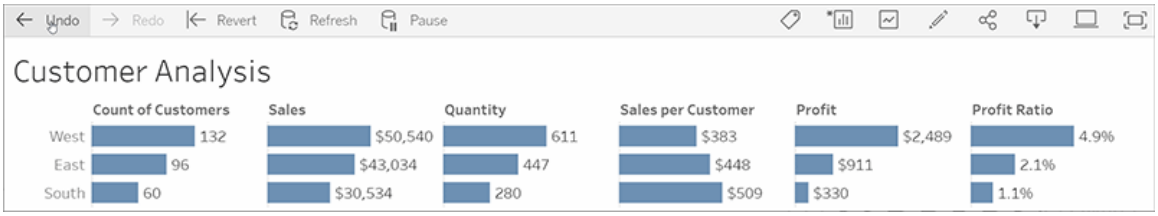
Filter Data

Trim or limit the visible data to a specific area, date, or category.



Undo/Revert

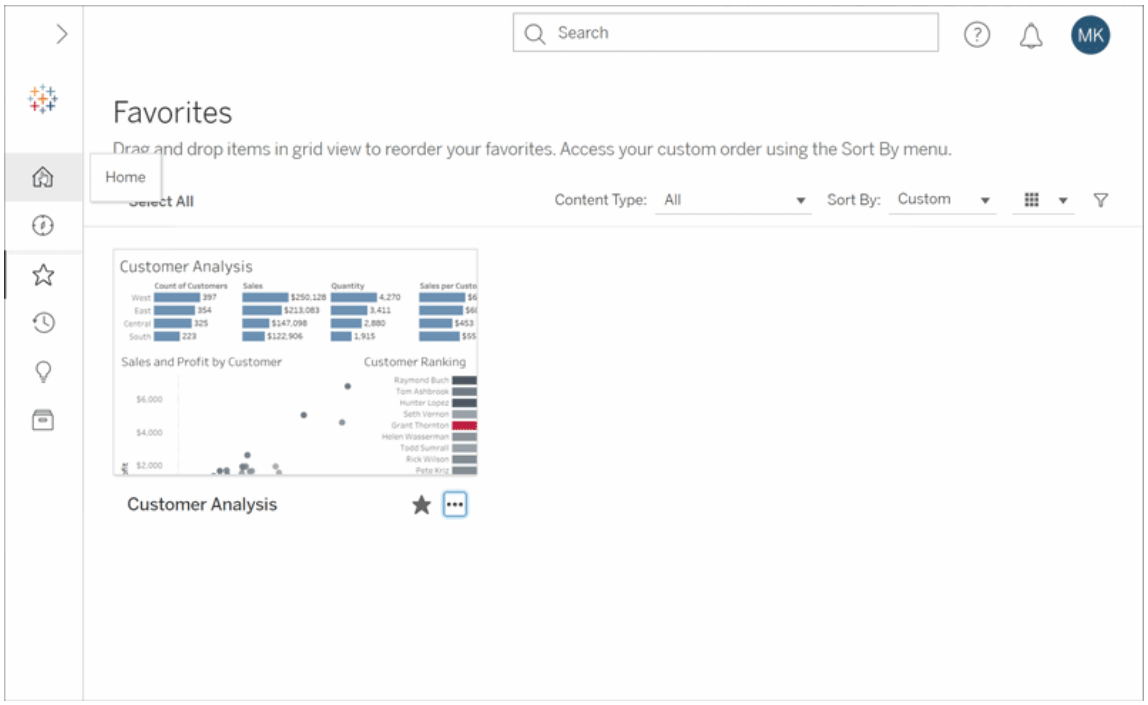
Maybe you didn't mean to exclude everything but one area. Click Undo to remove the last change, or use Revert to undo all your selections.



4: Keep up

This view can automatically update with new data, so you don't need to go searching for new charts to get the latest information. Keep it handy by clicking the star icon to add it to your favorites.

All favorites are added to the Favorites page in the navigation panel. And any recent dashboard or view you explored also appears in the Home page, waiting for you next time.



There's a lot you can do in Tableau Server, and this just explored some of the basics. For more information, see What Can I Do with a Tableau Web View?

Happy exploring!

Select Background Maps

Tableau comes installed with a set of background maps that you can access to create map views.

By default, Tableau maps appear with a **Light** background layer provided by Mapbox maps.

The eight background maps that come installed with Tableau are described in this table.

Background Map	Description
Light	A subtle map that puts the emphasis on the marks while providing geographic context. All areas without data appear in white or light gray.
Normal	A general-purpose map similar to Light. Land areas without data appear in

Background Map	Description
	white or light gray, and bodies of water appear in light blue.
Dark	A subtle map that puts the emphasis on marks while providing geographic context. The inverse of the Light map; areas without data appear in black or dark gray.
Streets	A general-purpose map that includes major road and transit networks.
Outdoors	A general-purpose map that includes terrain and natural features, including bodies of water and parks.
Satellite	A stylized map with global satellite imagery.
Offline	A map that you can use while not connected to the internet. This background map stores the images that make up the map in a cache on your machine for improved performance and offline access. For more information, see the offline maps section.
None	A visualization that displays data between latitude and longitude on a viz type other than a map.

Change your background map:

In Tableau select **Map** > **Background Maps** and then select the background map you want to use.

Change your default background map in Tableau Desktop (feature deprecated)

Note: Changing your default background map is a legacy feature that only works with WMS maps on Tableau Desktop. We don't recommend using this feature.

You can choose to set your default background map to a Web Map Service (WMS) or offline map. To learn more about using a WMS map, see Use Web Map Services (WMS) Servers.

To specify a default background map:

1. In Tableau Desktop, select **Map** > **Background Maps** > **Add WMP Map**… or **Offline**.
2. Select Map > Background Maps > Set as Default to set the selected background map to the default.

The background map is automatically saved as a Tableau Map Source (.tmsd) file and placed in the Mapsources folder of your My Tableau

Repository. It's now the default background map for all new worksheets.

Use the Offline background map

You can create and inspect data in a map view offline using the offline background map that comes with Tableau Desktop.

To use the offline background map:

- In Tableau, select **Map** > **Background Maps** > **Offline**

Note: The offline background map uses map images stored on your machine. You can find these images in the following locations:

- On Windows: `C:\Program Files\Tableau\<Tableau Version>\Local\Maps`
- On Mac: `//Applications/<Tableau Version>.app/Contents/install/local/maps`

There are several actions, however, that require Tableau to retrieve a map image that may not be stored. If the new map image isn't stored on your machine, you won't be able to load the map until you reconnect to the online map that comes with Tableau.

You may need to reconnect to the online map if you would like to do one or more of the following:

- **Turn layers on or off** - if you decide to turn on a layer that isn't stored in the cache, Tableau needs to connect to retrieve the necessary information.

- **Zoom in or out** - zooming in or out on a map requires different map images. If the images at the specified zoom level don't exist in the cache, Tableau needs to retrieve the updated maps.
- **Pan** - panning sometimes requires new map images. If you're working offline and don't have the necessary map images and legends stored in the cache, the new images and legends won't load.

To reconnect to the Tableau map:

- On Tableau Desktop, select **Map** > **Background Maps** > **Tableau**

Note: If you set a map to Offline and then publish the workbook, the published workbook will still use the Offline, stored maps, with all of the functionality and limitations of Offline maps noted earlier.

Create and Troubleshoot Metrics (Retired)

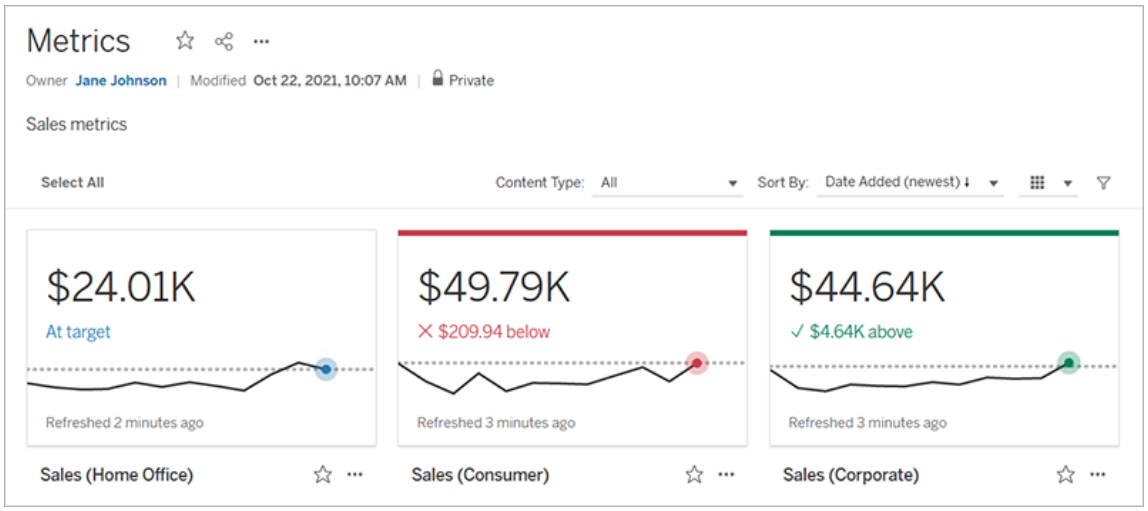
Retirement of legacy metrics

This article is about Tableau's legacy metrics feature, which was retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2. In October 2023, Tableau retired the ability to embed legacy metrics in Tableau Cloud and in Tableau Server version 2023.3.

Tableau Pulse introduces a new way to track metrics. With Tableau Pulse, the metrics you create are used to generate insights about your data. These data insights are sent directly to users who follow the metrics, so they can learn about changes to their data in their flow of work. For more information, see Create Metrics with Tableau Pulse.

If you have legacy metrics that you want to keep, note the data source, measure, and time dimension for those metrics and recreate them in Tableau Pulse. Legacy metrics won't be automatically migrated to Tableau Pulse.

Metrics provide a fast way to stay informed about your data. Because metrics update automatically and display their current value in the grid and list view of your content, you can check all the key numbers you care about in seconds.



At their most basic level, metrics show the value of an aggregate measure, like the sum of sales. More complex metrics can include timelines, comparisons, and statuses that provide an easy to understand indicator of how you're performing relative to a prior point in time or a value you have defined.

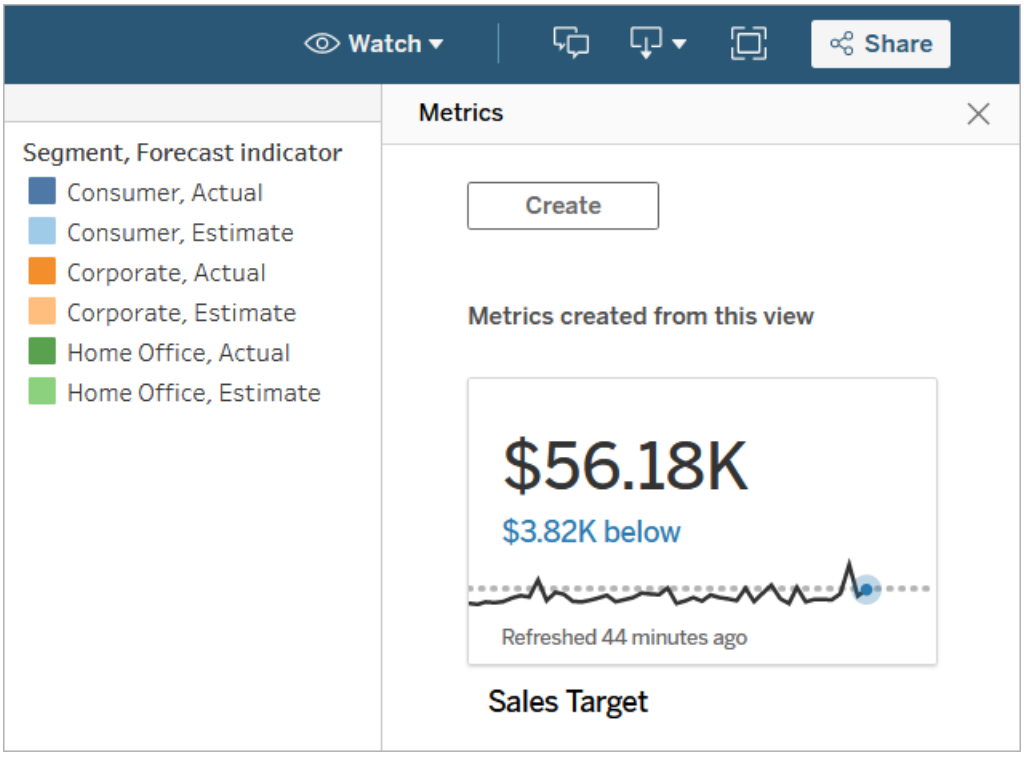
If you have a set of dashboards that you regularly check, create metrics for the numbers that you want to monitor, then track them in one place by adding them to your favorites or a collection, or by creating them in the same project. That way, you don't need to load and filter the dashboards unless you want to dig deeper into your data.

Find metrics on your site

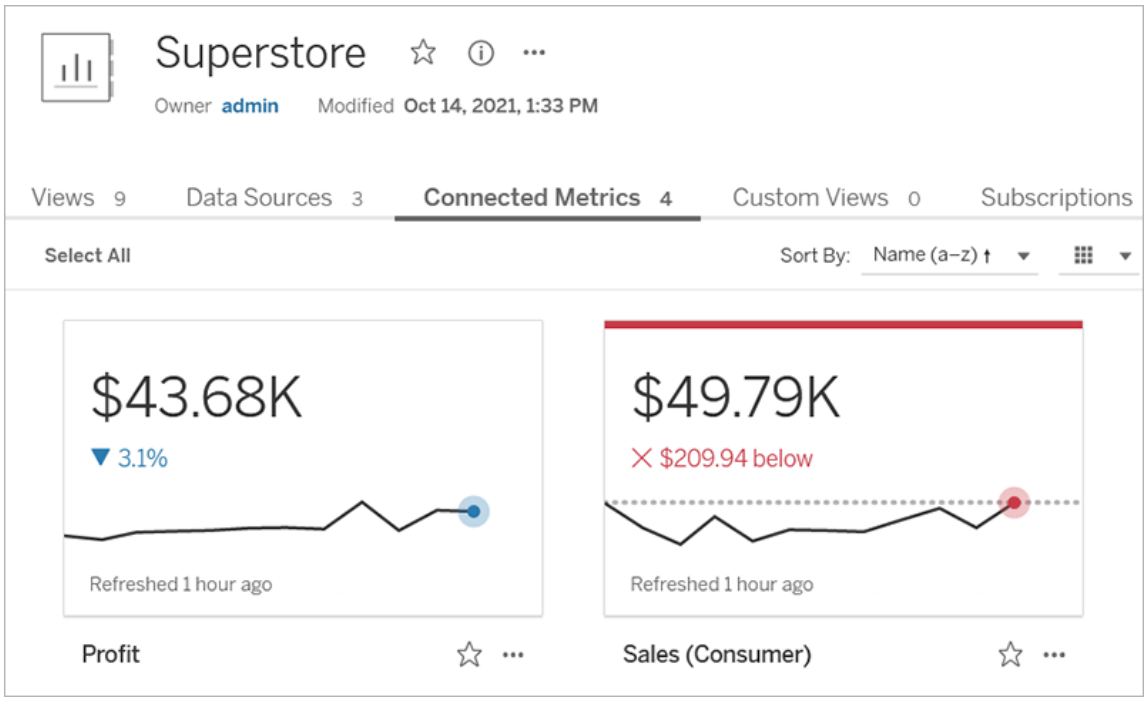
There are a few ways to find metrics on your Tableau site. To browse all the metrics that you have permission to view, navigate to the Explore page, then select **All Metrics** from the content type menu.

If you're looking for metrics related to a particular view or workbook, check the connected metrics for that content. To see connected metrics for a view, open the view, then click **Watch** >

Metrics in the view toolbar. The metrics displayed are ordered from the newest creation date to the oldest.



To see connected metrics for all the views in a workbook, navigate to the workbook, then click the **Connected Metrics** tab. You can sort these metrics using the Sort By menu.

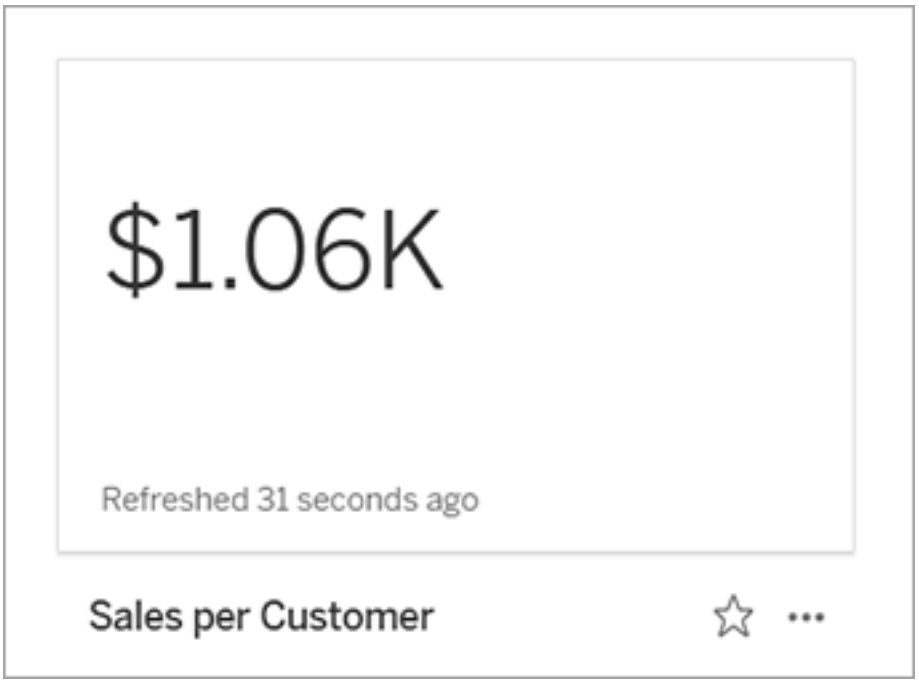


Components of a metric

The only data required to define a metric is an aggregate measure. Metrics are created from a mark in a view, and the measure associated with that mark defines the metric. The measure must be aggregated, because an unaggregated mark will not change over time. For information about dimensions and measures in Tableau, see Dimensions and Measures, Blue and Green.

A metric can optionally be defined by a date dimension, and you can configure a comparison and a status for your metric. Each of these components will add context to the data presented on the metric card.

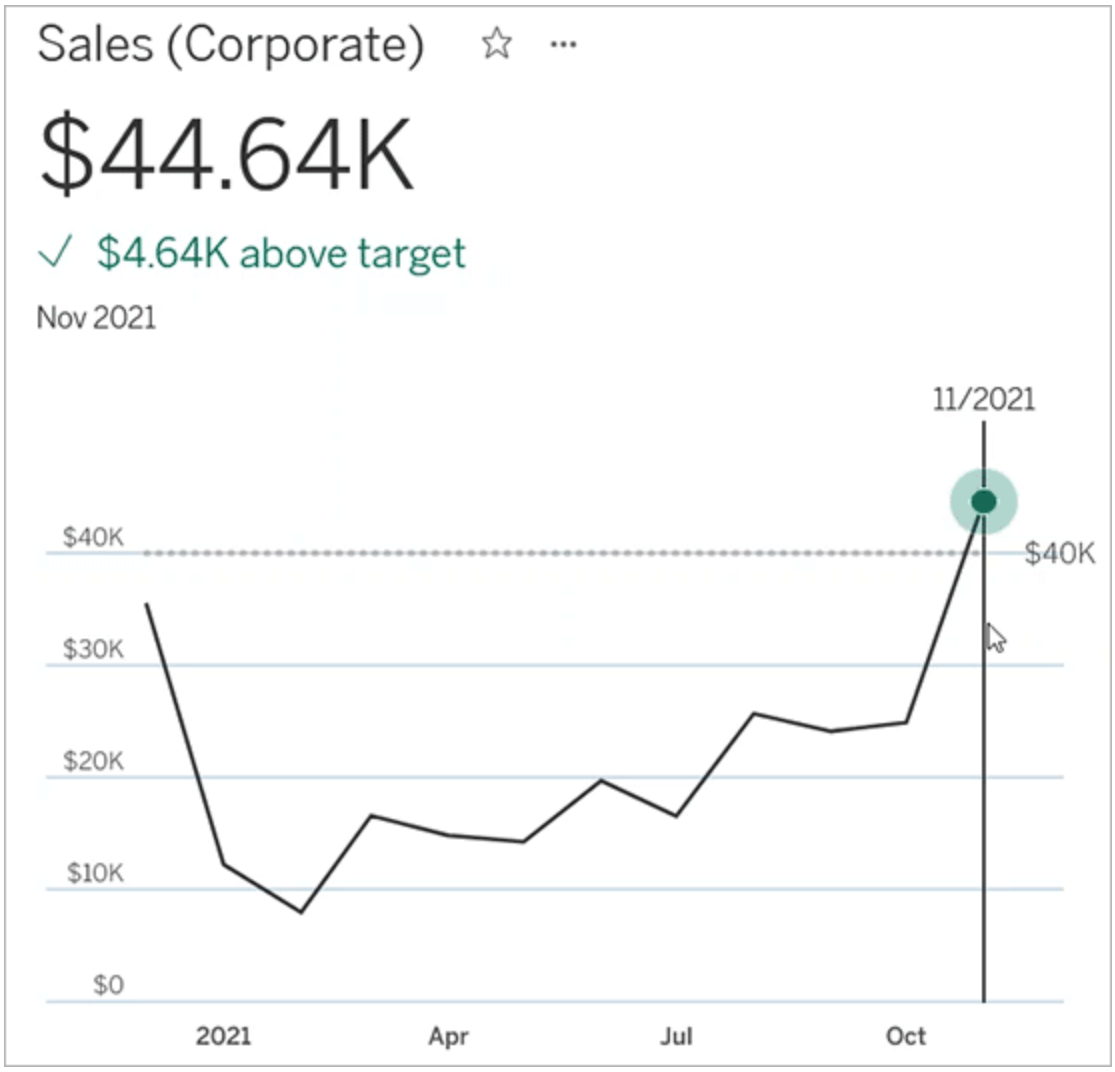
Metrics that have only a measure defining them will appear as a single number. This number will update when the data updates, but there will be no timeline on the card.



Timeline

When you select a mark to define a metric, if the mark has a date dimension associated with it, that dimension becomes part of the metric definition. Metrics with a date dimension show a timeline, and you can configure the historical comparison for the metric. By default, the historical comparison is to the previous mark.

When you open a metric's details page, the timeline shows the value of the measure based on the granularity of the date dimension, for example, daily sales or monthly users. Hover over the points on the timeline to see historical values.

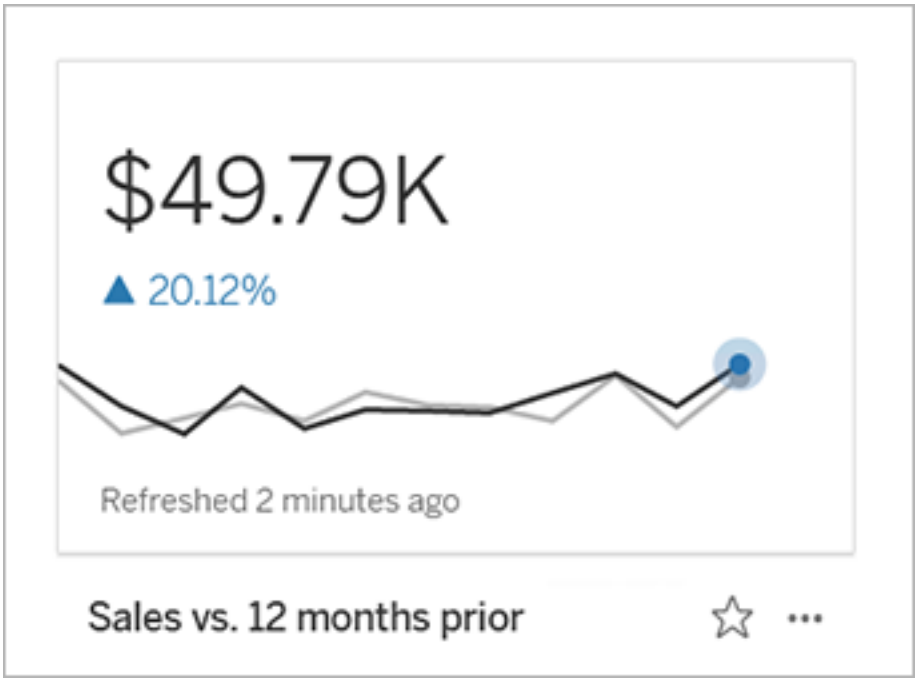


Comparison

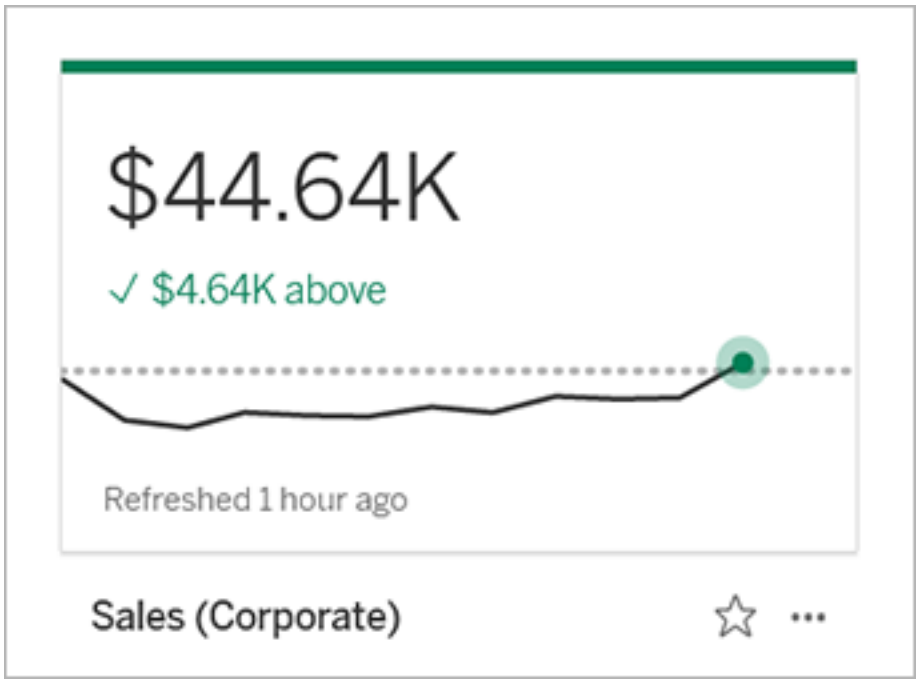
There are two types of comparisons that you can configure for metrics: historical comparisons and constant comparisons. You can configure a historical comparison only if a metric has a date dimension associated with it, but constant comparisons can be added for any type of metric.

A historical comparison is a relative comparison between the current value and a specified number of hours, days, or other unit of time previous. For example, you could set a comparison between the current value for monthly sales and the value from 12 months ago. Every

time data is added to a metric, the historical comparison will adjust relative to the date or time of the new data.

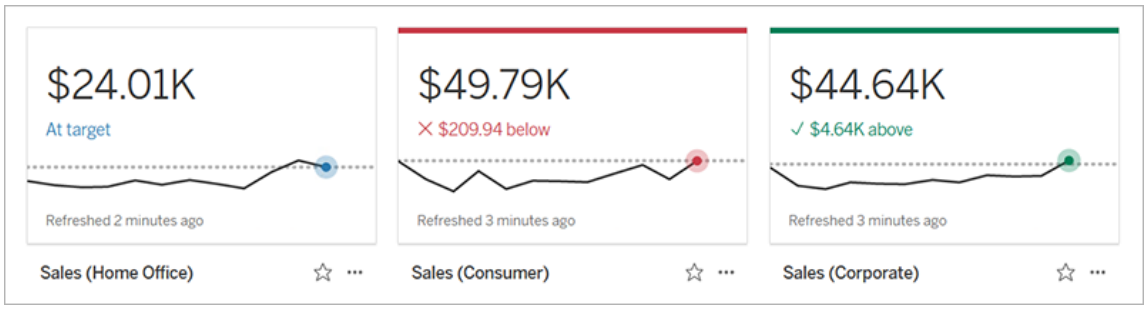


Constant comparisons are to a single value that doesn't change as new data is added. You might set a comparison to represent a threshold to stay above, for example, if you need to maintain a 90% on-time delivery rate. Or you might define a cumulative goal you are working toward, for example, a monthly sales target.



Status

For metrics with a constant comparison, you can define whether being above, below, or at the comparison value is good, bad, or neutral. A metric with a “good” status will display a check mark next to the comparison value, and the metric card will have a green band at the top. A metric with a “bad” status will display an X next to the comparison value, and the metric card will have a red band at the top. Metrics with a “neutral” status appear the same as metrics without a status indicator; there is no icon or color applied to the card.



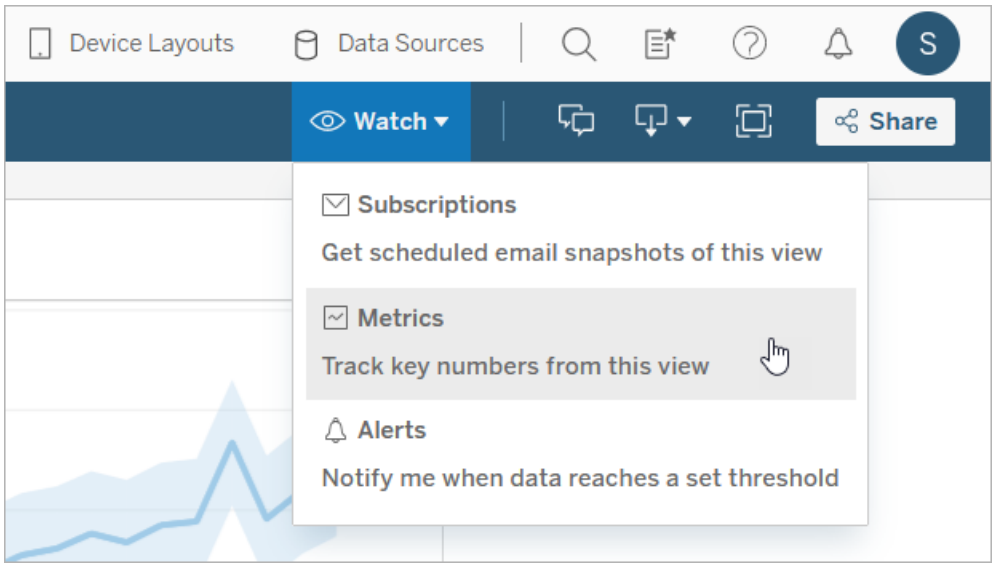
Create a metric from a view

If you have a site role of Creator or Explorer (can publish), and you have the Create/Refresh Metric capability on the relevant workbook, you can create metrics on Tableau Cloud or Tableau Server.

Before you create a metric, check the connected metrics for the view to make sure that the metric you are planning to create doesn't already exist. Instead of creating a duplicate metric, open the existing metric and add it as a favorite.

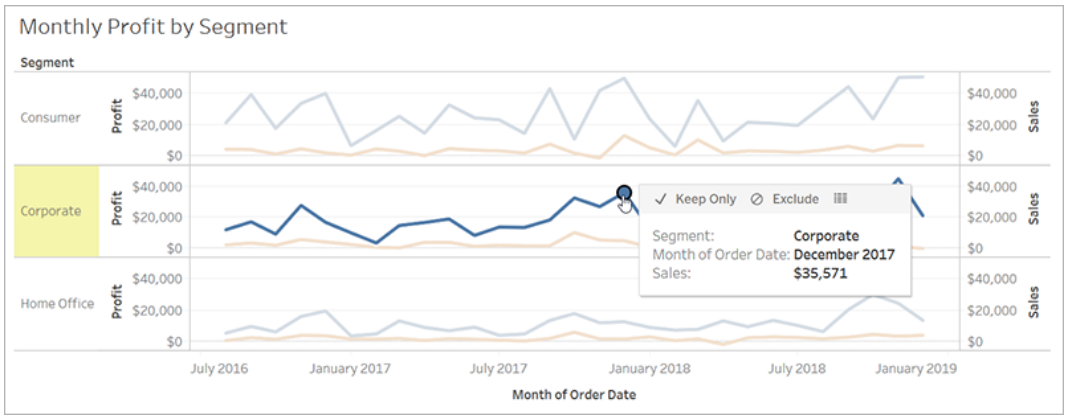
Select the mark to define your metric

1. Navigate to the view that you want to create a metric from.
2. On the view toolbar, select **Watch** > **Metrics**.



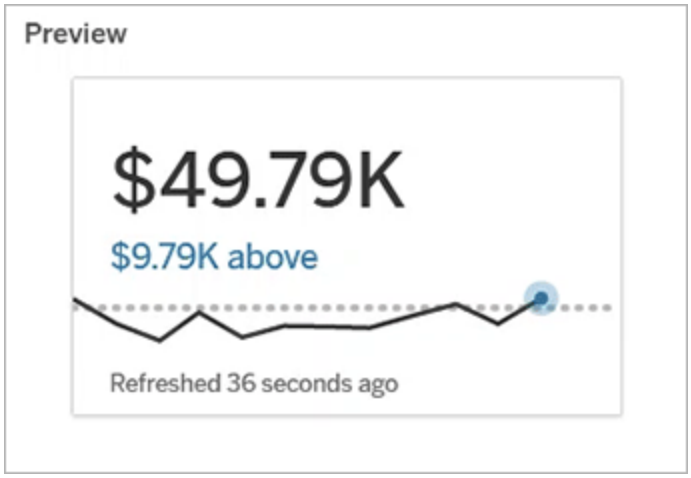
The metrics pane opens.

3. If the pane shows connected metrics, select the **Create** button to enter authoring mode.
4. Select a mark. If you encounter an error, see When you can't create a metric.



The measure associated with this mark defines your metric. Any filters you apply to the mark are applied to your metric. If this mark has a date dimension associated with it, that date dimension also defines your metric, and your metric will display a timeline.

The metrics pane shows a preview of your metric. The value in the preview is the most recent value for the metric, which may differ from the value of the mark you selected if it was not the most recent in the time series. The preview updates as you try different configurations.



Describe and configure your metric

The options available to configure your metric depend on the mark you select and on the type of comparison you choose.

1. The **Name** field is pre-populated based on the mark you select. You can give the metric a different name. A metric must have a unique name within the project it belongs to.

2. Under **Description**, enter an optional message to help others understand your metric. For example, describe filters applied to the metric or indicate the data source used by the metric.

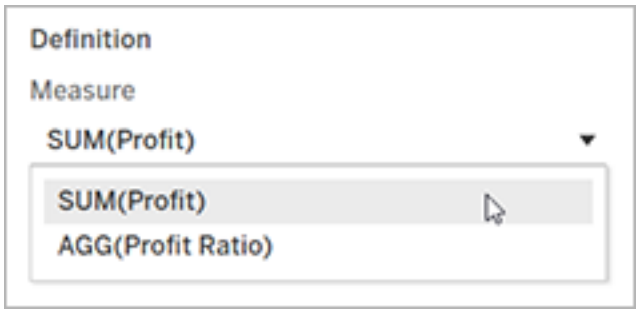
3. For the **Date Range** (only for metrics with a date dimension), select one of the default options, or set a custom range. If your metric has a large number of marks, limiting the date range can make it easier to read the timeline.

4. Select the **Comparison Type** for the metric: historical or constant.

5. For **Historical** comparisons:
   - Enter how far prior you want to compare against. The unit of time for the comparison is the same as the granularity of your data, such as hours or months.
   - Select **Show Comparison Line** to include a second line for the comparison period on the timeline.

6. For **Constant** comparisons:
   - Enter the value to compare against. Don't include commas or symbols in this field. To enter a percentage, simply type the number without the percent sign, for example, enter 25 instead of 0.25 for a target of 25%. When you enter a valid target value, the preview updates to show how far above or below the target the current value is.
   - Set the **Status** for the comparison to indicate whether being above, at, or below the value is good, bad, or neutral. By default, the status is set to neutral. Check the metric preview to see how different statuses affect your metric.

7. Under **Definition** > **Measure**, select the measure to use for your definition from the drop-down. This option appears only if the mark you select has more than one measure associated with it.



Finalize your metric

1. Under **Project**, select **Change Location** to choose a different project for the metric. By default, your metric will be added to the same project that the view belongs to.

   Every metric in a project must have a unique name. Because a metric's name and project are initially set based on the mark you select, you might run into a conflict when you try to save your metric, if another user has already created a metric from that mark. Either choose a different project or name for your metric or, if you want to overwrite the existing metric, see Overwrite a metric.

2. Click the **Create** button.

   A message appears with a link to the metric in the project you added it to.

3. Verify that the permissions for your metric are correct, following the guidance in Set permissions.

   By default, a metric inherits the permissions of the project it is created in. Anyone with access to your metric will be able to see the metric's data, even if they don't have permission to access the connected view or data source.

Now that you've created a metric, you can manage the metric the same way you manage other independent pieces of content on your Tableau site. Though metrics are created from a view, they exist independently of that view, unlike data driven alerts or subscriptions. You can

move the metric to a different project without moving the connected view. For information about managing content on your Tableau site, see Manage Web Content.

Overwrite a metric

Once a metric is created, you can change the name, description, and configuration of the metric, but you can't change how the metric is defined. If you want to change the data that the metric uses, you must overwrite it. In order to overwrite a metric, you need to be the metric owner or be granted the correct permission capability.

1. To overwrite a metric, create a metric with the same name in the same project as the metric you want to overwrite.

   The Overwrite Metric dialog appears.

2. Click the **Overwrite** button.

When you overwrite a metric, the metric continues to appear for those who have added it to their favorites, and any changes made to permissions for the previous metric will apply to the new metric.

When you can't create a metric

If you select a mark on a chart that doesn't support metrics, you'll get an error message explaining why you can't create a metric. The table below summarizes these scenarios.

Reason	Scenarios
You don't have the correct permissions.	• The workbook owner or an administrator has denied the Create/Refresh Metric capability. For more information, see Permissions.
You can't access the complete data.	• Row level security or user filters limit the data you can see. For more information, see Restrict Access at the Data Row Level.
The password for the workbook's data source is not embedded or is no longer	• The workbook prompts for a password. For more information, see Set Credentials for Accessing Your Published Data.

Reason	Scenarios
valid.	
The data isn't at the correct level of granularity.	• The data in the chart isn't aggregated. Metrics use aggregations, such as sum or average. For more information, see Data Aggregation in Tableau. • There are multiple values per cell of data—a result of data blending. For more information, see Troubleshoot Data Blending.
The date dimension is not supported.	• The chart includes both date parts and date values. For more information, see Change Date Levels. • The date dimension uses the ISO 8601 calendar rather than the standard Gregorian calendar. For more information, see ISO-8601 Week-Based Calendar. • The date dimension is aggregated at the custom level of Month / Year or Month / Day / Year. For more information, see Custom Dates.

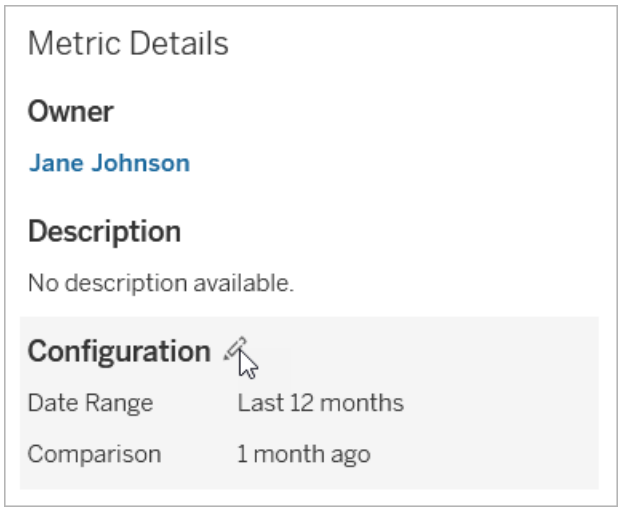
Edit a metric's configuration

Starting in 2022.2, you can change the configuration for a metric. A metric's configuration includes the comparison, date range, and status indicator. The configuration options available depend on the type of metric. Metrics with a timeline allow you to set a historical comparison or a constant comparison. Single number metric support only a constant comparison.

A metric's configuration does not include the metric definition (the measure and date dimension that generate the metric value). If you want to change the definition, overwrite the metric with a new metric.

To edit a metric's configuration, you must have the overwrite capability for the metric.

1. Open the metric details page for the metric you want to edit.
2. Mouse over the configuration section. Click anywhere on the section to enter editing mode.



3. For a timeline metric, set the date range to display on the metric card and details.
4. Select the comparison type. For a constant comparison, set a comparison value and status. For a historical comparison, set how far prior you want to compare against and choose whether to show a comparison line on the timeline.
5. Click **Save**. Your configuration changes will appear to anyone who views the metric.

How metrics refresh

When a metric refreshes, it checks the connected view (the view the metric was created from) for new data. A refresh doesn't necessarily update the value of a metric, because there may be no changes to the data.

Metrics refresh at a frequency either based on an extract's refresh schedule or, for live data, every 60 minutes. The time of the last refresh is displayed on the metric.

Fix failing refreshes

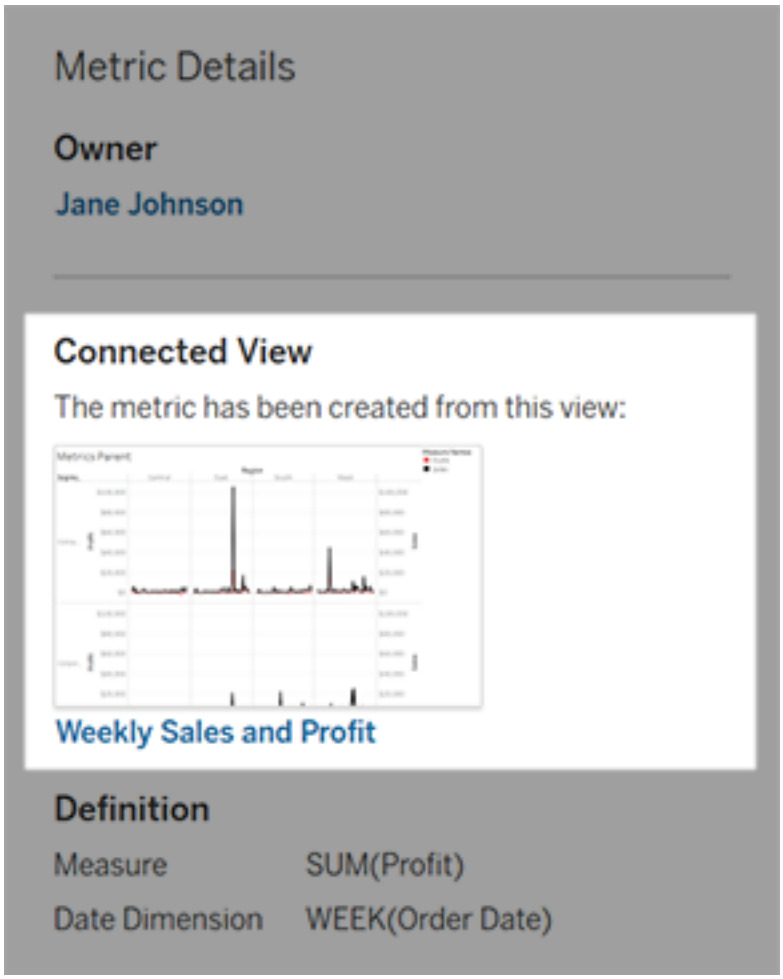
If a metric isn't able to access the connected view or its underlying data, the refresh will fail. If the refresh for your metric fails, you'll receive a notification, which notes the time of the failure and the affected metric.

Metric refreshes may fail for one of the following reasons.

- The connected view was deleted or modified.
- Permissions changed for the connected view.

- The password for the data source is no longer embedded or is no longer valid.
- The metric owner doesn't have the required site role to refresh the metric. A site role of Creator or Explorer (can publish) is required.
- There was a temporary connectivity issue, which will resolve itself.

To identify the cause of the failure, look at the metric details. Make sure that the metric owner has the required site role to refresh the metric. Then inspect the **Connected View**.



If the connected view is still listed

Open the view to investigate the cause of the failure.

If the view loads, check that the measure and (optional) date dimension that define the metric are still present in the view.

- If the view appears to be unchanged, you might no longer have permission to refresh metrics from it. The content owner or a Tableau administrator can change the Create/Refresh Metric permission capability. For more information, see Permissions.
- If the measure is no longer present, the view has been modified so the metric can't connect to the data needed to refresh. The content owner or a Tableau administrator can check the revision history and restore previous versions. For more information, see View Revision History.

If the view doesn't load, but instead prompts for a password or displays an error when connecting to the data source, the password for the data source is not embedded or is no longer valid. The content owner or a Tableau administrator can edit the data source connection to embed the password. For more information, see Edit Connections.

If there is no connected view listed

The view was deleted or you no longer have permission to access the view. Contact your Tableau administrator for assistance.

Resume suspended refreshes

If a refresh fails enough times, the refresh is suspended. You'll receive a notification if the refresh for your metric is suspended.

When a metric refresh is suspended, Tableau no longer attempts to get new data for the metric. Metrics with suspended refreshes continue to present historical data.

If the cause of the failure is fixed, you can resume the refresh.

1. Open the affected metric.
2. On the warning message, click **Resume refresh**.

Tableau attempts to perform the refresh. If this attempt succeeds, you'll receive a notification, and the refresh will resume on schedule. If the attempt doesn't succeed, your refresh remains suspended.

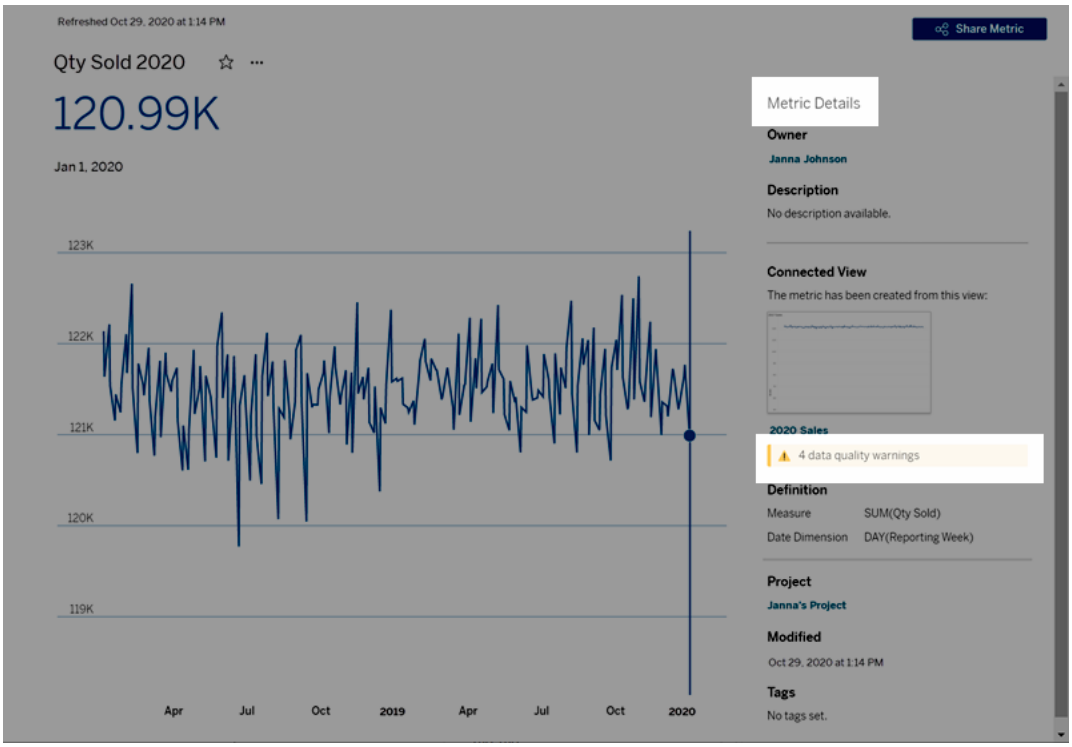
Try overwriting the metric if the connected view is still available. For more information, see Overwrite a metric. Otherwise, you can keep the metric to reference past data or delete the metric.

Note: If the metric refresh is suspended because you don't have the required site role for it to refresh, you won't be able to resume the refresh or delete the metric.

Metrics appear in Tableau Catalog

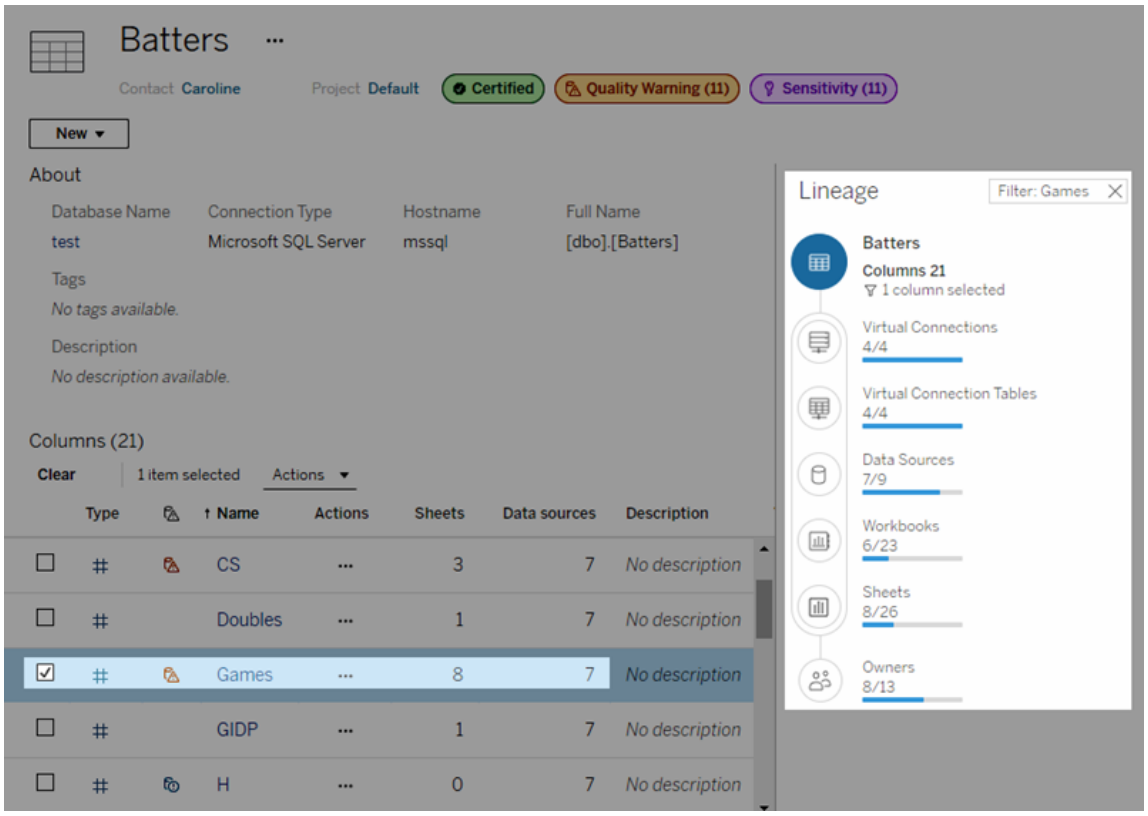
Starting in 2019.3, Tableau Catalog is available with Data Management in Tableau Cloud and Tableau Server. When the Tableau Catalog is enabled in your environment, you can see metrics in the Catalog lineage tool, and metrics affected by data quality warnings display those warnings. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Cloud Help.

When you have metrics defined for the numbers you want to monitor, it's important to know if the data that the metrics rely on are impacted in any way. You can use Catalog in a couple of ways to know this. First, data quality warnings set on the data your metric is based on show on the metric. These warnings appear when you open a metric in Tableau Mobile, and in Tableau Server and Tableau Cloud when you hover over a metric in grid view and on the metric details page, as shown below:



For more information, see "Set a Data Quality Warning" in the Tableau Server or Tableau Cloud Help.

You can also use the lineage tool in Tableau Catalog to see the upstream sources that your metric relies on. When doing an impact analysis, you can see which metrics are affected if a certain column or a table is changed or deprecated, or if certain workbooks are removed. Including metrics in the lineage means that Catalog gives a full picture of the impact changes can have on the assets in your environment.



For more information, see "Use Lineage for Impact Analysis" in the Tableau Server or Tableau Cloud Help.

Set Credentials for Accessing Your Published Data

When you publish a workbook to Tableau Cloud or Tableau Server, you can publish the data source it connects to as part of the workbook (*embedded* into the workbook), or as a separate, standalone data source. In addition, if the data source you're publishing requires authentication, you can customize how credentials are obtained.

The type of authentication to your data source is independent of how people sign in to your Tableau Cloud or Tableau Server site. For example, to give people direct access to the data in a workbook, you would embed a database user's credentials into the data source's con-

nection. But anyone viewing the workbook would still need to be able to sign in to the site on Tableau Cloud or Tableau Server to open your workbook.

This topic describes how to set authentication on data connections as part of the publishing process.

Note: This topic doesn't apply to connections to that don't require authentication, such as text files or Excel files.

Set the authentication type

For many types of connection you can embed a database user's name and password, or use single sign-on (SSO). Specific exceptions are described later in this topic.

The following steps describe how to set authentication as part of publishing a data source or workbook. You can do this for each connection in the data source.

1. In the Publish Workbook dialog box, go to the **Data Sources** area, which lists the workbook's connections, and select **Edit**.
2. In the **Manage Data Sources** popup, after you decide whether to publish the data source separately or as part of the workbook, select an authentication type for each connection in the data source. The available authentication types depend on the connection type, and they can include one or more of the following:
    - **Prompt user**: Users must enter their own database credentials to access the published data when the view or workbook loads.
    - **Embedded password**: The credentials you used to connect to the data will be saved with the connection and used by everyone who accesses the data source or workbook you publish.
    - **Server run as account**: A single Kerberos service account is used to authenticate the user. On Windows this is the account that Tableau Server runs as. On Linux it can be any Kerberos account.
    - **Viewer credentials**: The viewer's credentials are passed through to the database using SSO (usually Kerberos).
    - **Impersonate with embedded account** or **Impersonate with server Run As service account**: Impersonation using embedded credentials connects with the

embedded credentials and then switches to the viewer's identity (only for databases that support this). Impersonation using the Run As service account is similar but first, connects with the Kerberos service account before switching to the viewer's identity.

- **Refresh not enabled** or **Allow refresh access**: These options appear when you publish an extract of cloud data such as from Salesforce, and database credentials are needed to access the underlying data. **Allow refresh access** embeds the credentials in the connection, so that you can set up refreshes of that extract on a regular schedule.

Important: How you want to keep extracted data fresh is also a factor:

- If you want to set up an automatic refresh schedule, you must embed the password in the connection.
- If you're publishing a cloud data connection to Tableau Cloud, the publishing steps alert you if you must add Tableau Cloud to the data provider's authorized list.
- You can't publish an extract that's created from a Kerberos-delegated, row-level-secure data source.

Dropbox, OneDrive connections

For Dropbox and OneDrive, when you publish a data source or workbook and select **Embedded password**, Tableau creates a saved credential and embeds it in the data source or workbook.

Workbook connections to Tableau data sources

When you publish a workbook that connects to a Tableau Cloud or Tableau Server data source, rather than setting the credentials to access the underlying data, you set whether the workbook can access the published data source it connects to. Regardless of the original data type, the choice for server data sources is always **Embedded password** or **Prompt users**.

If you select to prompt users, a user who opens the workbook must have **View** and **Connect** permissions on the data source to see the data. If you select embed password, users can see the information in the workbook even if they don't have View or Connect permissions.

Virtual connections

As of Tableau Cloud and Tableau Server 2022.3 and Tableau Desktop 2022.4, when you publish Tableau content like a data source or workbook that uses a virtual connection and select **Embed password** or **Embed credentials**, the viewer of the content will have your permissions to connect to and query the virtual connection. However, any data policies associated with the virtual connection are always evaluated using the viewer's identity–not yours.

For example, you publish a workbook that uses a virtual connection. To let viewers of the workbook connect to and query data by way of the virtual connection, you embed your permissions to connect to and query the virtual connection. Then, any data policies associated with the virtual connection prevent the viewers of the workbook from accessing any sensitive data.

When evaluating whether the tables in a virtual connection can be viewed and accessed, the identity of the content creator is used. However, when evaluating any data policies associated with the tables in a virtual connection, the viewer's identity is used. And the content creator can only ever embed connect permissions to the virtual connection–not edit permissions.

If you choose not to embed permissions, then only users with permissions to access the workbook or data source and with connect permissions to the virtual connection can access the workbook or data source.

The embed password and embed credentials options for virtual connections don't work in Tableau Cloud 2022.2, Tableau Server 2022.1, and Tableau Desktop 2022.3 and earlier. If you select these options before you upgrade to 2022.3 (for Tableau Cloud and Tableau Server) or 2022.4 (for Tableau Desktop), the options will work as expected after you upgrade. Then, you're able to embed your permissions for querying a virtual connection.

See also

- If you publish to Tableau Server, see Edit Connections in the Tableau Server Help.
- If you publish to Tableau Cloud and the workbook connects to Salesforce, Google Analytics, Google Sheets, Google BigQuery, OneDrive, Dropbox, and QuickBooks Online data, see Refresh Data Using Saved Credentials in the Tableau Cloud Help.

- If you're a Tableau Server administrator looking for more information about authentication, see the Tableau Server help topics, "Authentication" (Windows | Linux) and "Data Connection Authentication" (Windows | Linux).

Explore Dashboards with Data Guide

Have you ever discovered a new Tableau dashboard and wondered what data means or how to use it? Or, have you ever published a new dashboard and wished that you could include instructions about how to use your dashboard?

Data Guide provides helpful information about a dashboard and insights about the data behind it. Data Guide allows dashboard creators to provide more explanatory context for end users–like descriptions and links to resources–directly in the dashboard. And Data Guide automatically surfaces insights powered by Explain Data to help users find outliers and learn about explanations for a mark. Explain the Viz (powered by Explain Data) identifies outlier measures and potential key drivers behind them.

These contextually relevant details can help dashboard users navigate and use new dashboards more easily, allowing users to find insights faster, trust that they're looking at the right data, provide context for data, and establish confidence in their understanding of the viz.

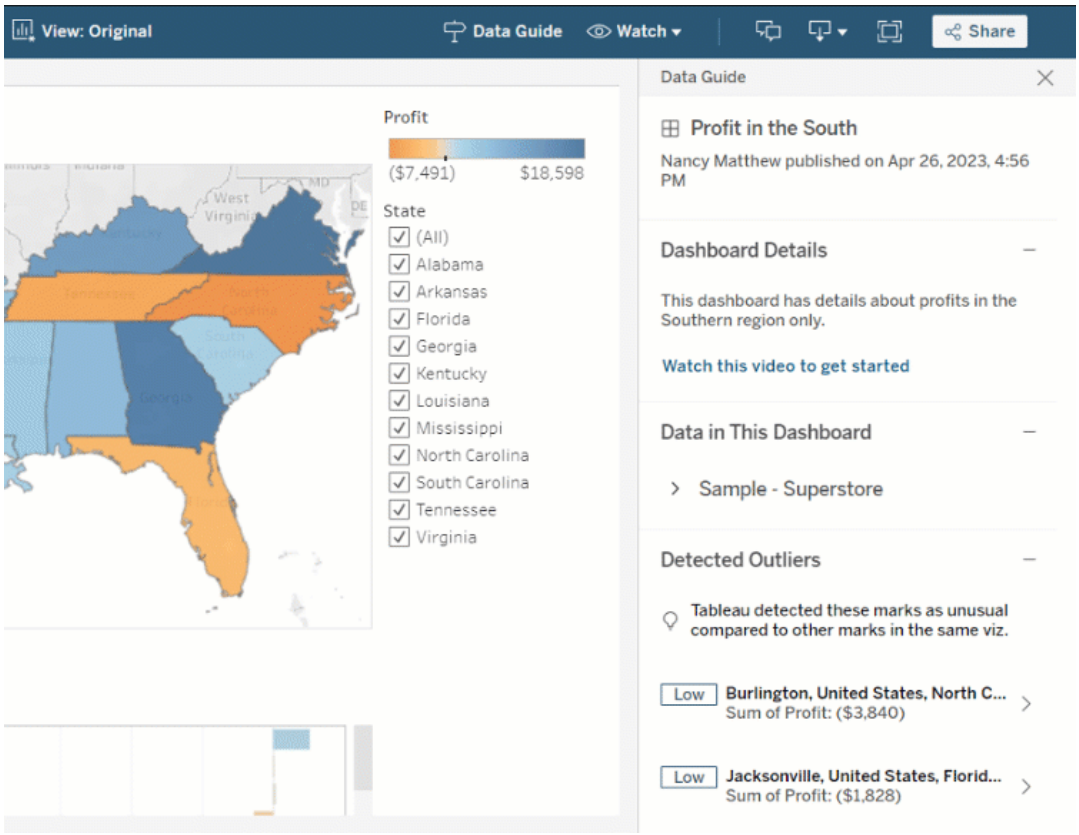
Customize Data Guide as an author

1. From your dashboard, choose the **Edit** button.
2. Choose **Data Guide** in the toolbar.
3. In the Data Guide pane, type a description to help your end users understand the purpose of your dashboard.
4. Select **Add link**, and enter a descriptive label for your link text and the URL.
5. Choose **OK**.
6. Select a viz (a sheet in the dashboard). Data Guide updates automatically so you can add a description and resources that are relevant to each viz in your dashboard.
7. Type a description, and add links relevant to the viz.
8. Choose **OK**.

Tip: To use Data Guide to write custom alt text to improve the accessibility of your vizzes, see Show more text and make it helpful.

Explore Data Guide as a dashboard user

1. From the dashboard, choose **Data Guide** in the toolbar.
2. Read the description of the dashboard and explore resources provided by the dashboard author.
3. Expand **Data in This Dashboard** and **Detected Outliers** to learn more about the underlying data used in the dashboard.
4. Select a viz (an object in the dashboard).
5. Read the description of the viz and explore resources provided by the dashboard author.
6. Expand **Data Summary** and **Detected Outliers** to learn more about data in that viz.
7. Select a mark or multiple marks, such as a bar on a chart or a region on a map, to see information about **Data in This Mark** and **Applied Filters**. Select a single mark to see possible **Explanations** for its value.



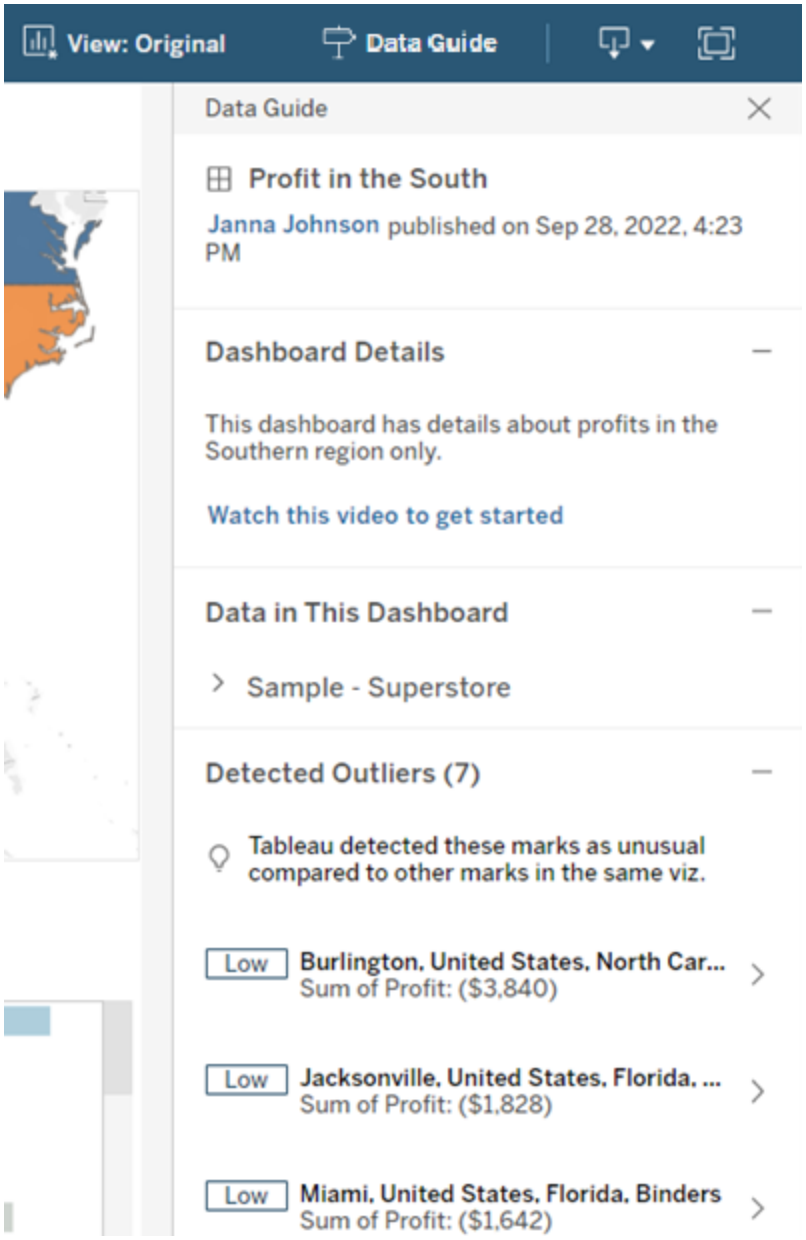
Explore Data Guide at different levels

By default, Data Guide is closed when you open a workbook. And you can open Data Guide in the toolbar. At the top of the Data Guide pane, you see the name of the dashboard or viz that you selected. As a dashboard author, you can write descriptions for both the dashboard and for individual vizzes that make up your dashboard.

Understand dashboard-level details

At the dashboard level, Data Guide:

- Displays the dashboard's name, author, and last published date.
- Can include a description written by the dashboard author and links to related resources, such as videos or wiki pages.
- Lists the data sources used by the dashboard and details about the data, such as which dimensions and measures are used.
- Reveals detected outliers in the dashboard that are identified by Explain Data. For more information, see Get Started with Explain Data.

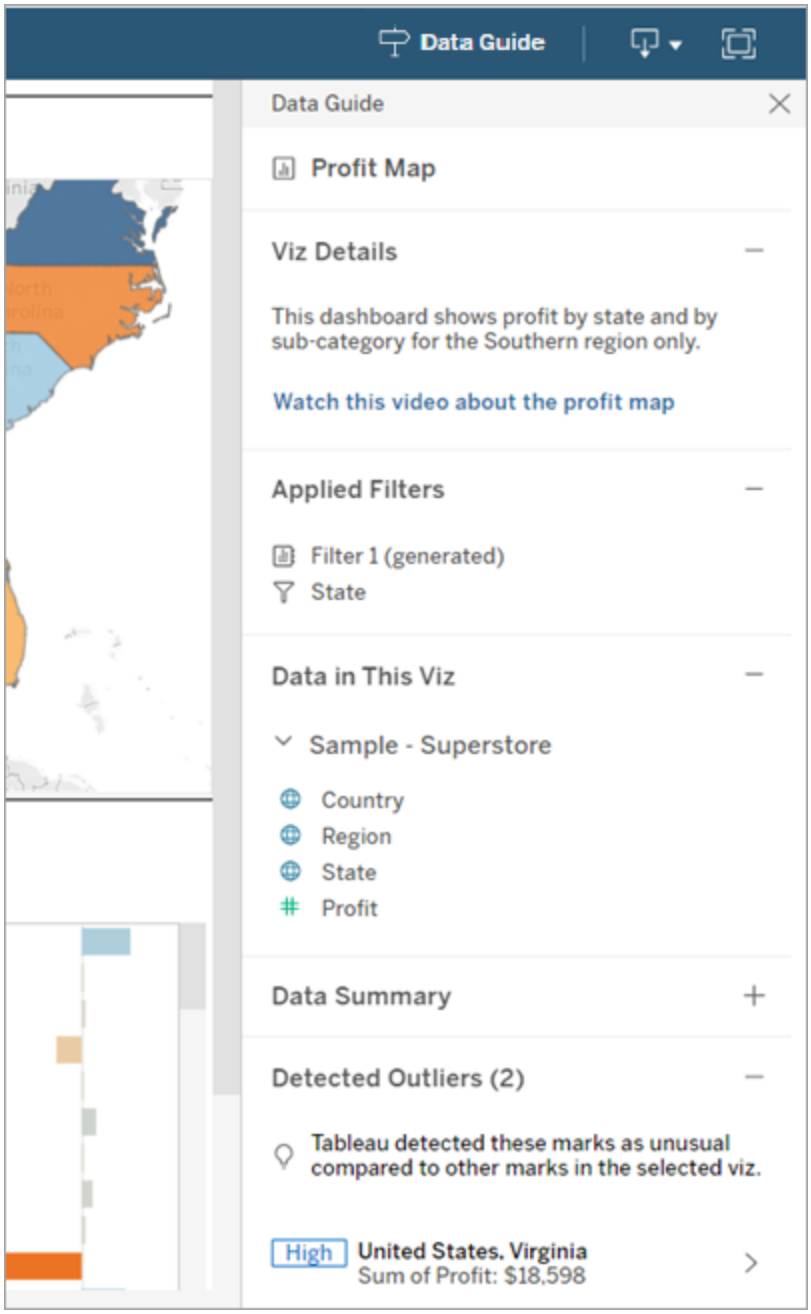


Understand viz-level details

When a user selects a specific viz (dashboard object), Data Guide:

- Displays the viz's name.
- Can include a description written by the dashboard author and links to related resources, such as videos or wiki pages.

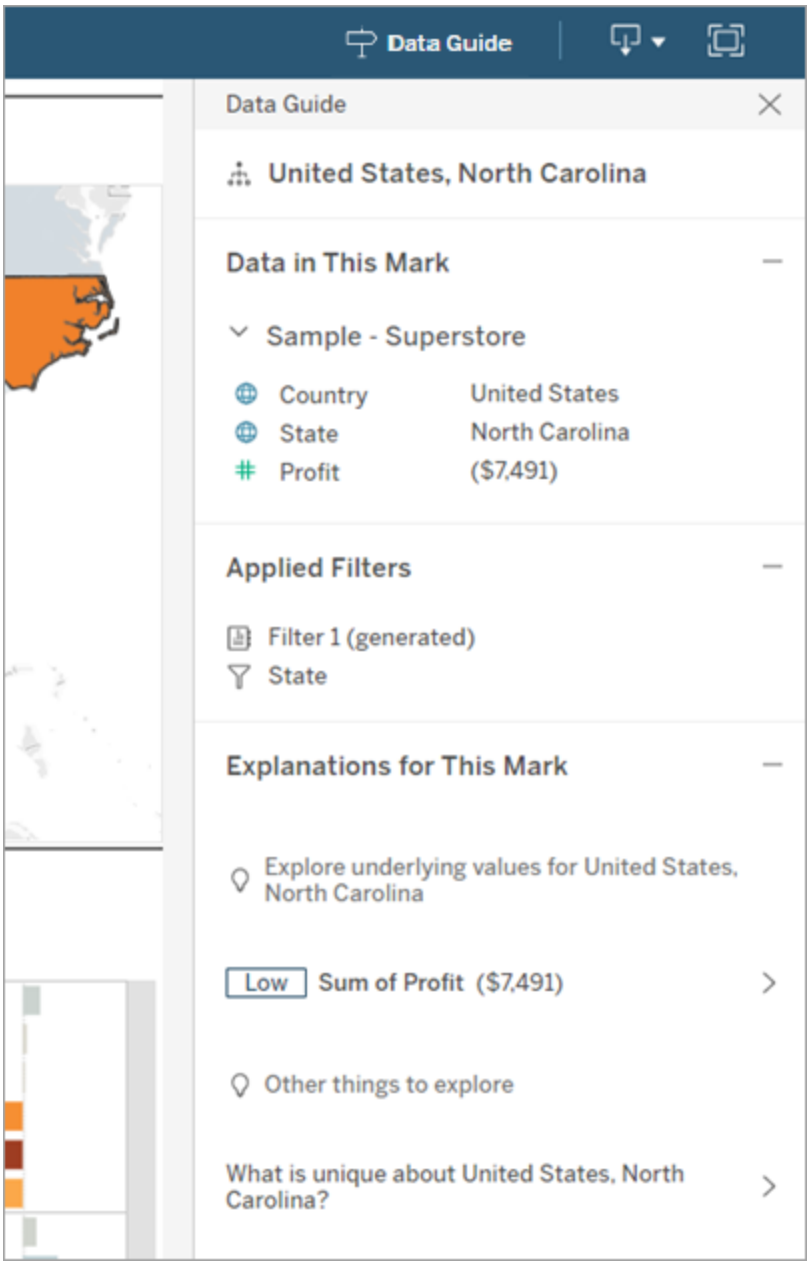
- Lists the data sources used by the dashboard, including the dimensions and measures used.
- Lists applied filters and data used in the viz, so you can identify how the data is being influenced by filters that users interact with. Only filters that a user can change are shown.
- Has a summary of the data used in the viz, including the number of data points and sorting details.
- Reveals detected outliers in the viz that are identified by Explain Data. For more information, see Get Started with Explain Data.



Understand mark-level details

When a user selects a mark (or marks) such as region on a map, Data Guide:

- Displays the name of the mark.
- Lists applied filters and data used in the viz, so you know what's included and what's excluded.
- Reveals detected outliers in the viz that are identified by Explain Data. For more information, see Get Started with Explain Data.



Control Data Guide visibility

By default, Data Guide is closed when you open a workbook. When you open or close Data Guide, it remains in this state for all workbooks on the site. In 2023.1 or later, a server administrator can hide Data Guide for all users. This removes the Data Guide button from the Tableau toolbar on all workbooks on the site. To hide Data Guide:

1. From the left pane, choose **Settings**.
2. From the **General** tab, scroll to **Availability of Data Guide**.
3. Choose **Hide**.
4. Choose **Save**.

Set a Data Freshness Policy for Query Caches and View Acceleration

Understand data freshness for Query Caches

You've built your workbook, and your team loves it. But sometimes people need to click the Refresh button for the most up-to-date data to appear in the viz. You built the workbook using a live connection, so why does the data need to be refreshed? The answer is performance.

To improve performance, Tableau caches the results of queries used to fetch data, so subsequent visits can reuse and return that cached data faster. You can click the Refresh button to retrieve updated data, but this can add to potential performance costs.

To balance data performance and freshness, set a data freshness policy for your workbook. When you set a data freshness policy, your data is refreshed at the time you specify. Tableau won't visualize cached data that doesn't meet the freshness policy you've set.

Understand data freshness for View Acceleration

With the View Acceleration feature, Tableau precomputes selected workbooks to generate views, resulting in significantly reduced load times. A precomputation schedule is created based on the data freshness policy or extract schedule that you set for the selected workbooks to provide data that is both performant and fresh.

To minimize resource consumption, the number of precomputation jobs that you can run is limited to 12 per day. For example, if your data freshness policy is set to less than two hours, the performance benefits of View Acceleration are limited to the first 12 refreshes in a day.

Choose what's best for your workbook

Some people might not want caching so that they always have the freshest data, while other people might want large caches to reduce overhead and improve workbook performance. The first step in setting a data freshness policy is to decide what's right for your business.

Tableau Cloud refreshes cached data every 12 hours by default, and workbook owners can set data freshness policies at the workbook level.

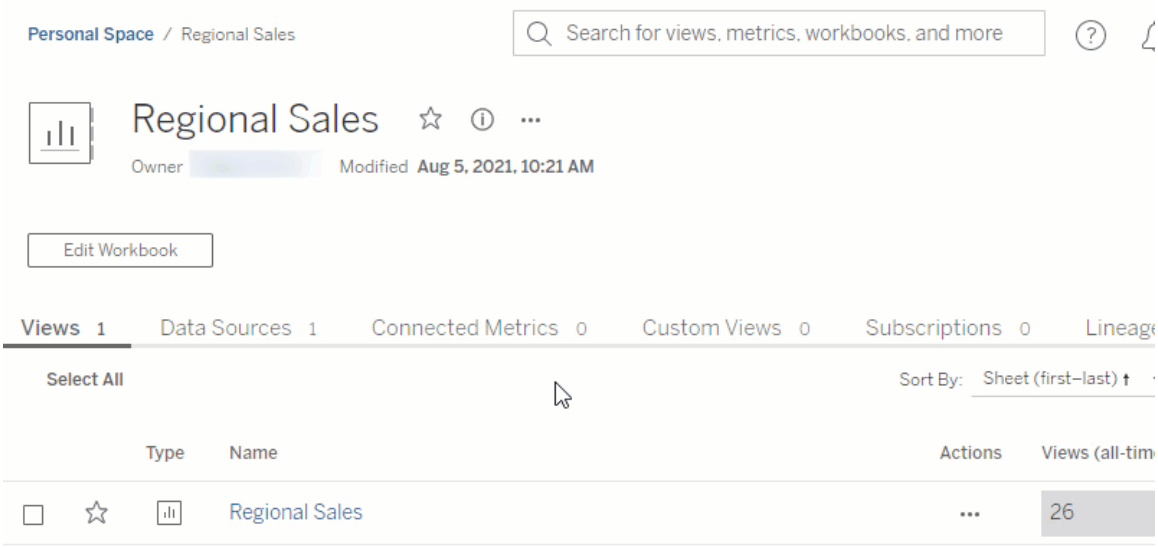
In Tableau Server, server administrators can set a default caching policy for all sites on the server, and workbook owners can set data freshness policies at the workbook level.

> **Note:** Data freshness policies aren't available in Tableau Desktop or for workbooks that use extract and file-based data sources.

Edit a workbook data freshness policy

To edit a workbook data freshness policy, you must be the workbook owner, and the workbook must have a live connection to the data source.

1. Sign in to a site on Tableau Cloud or Tableau Server.
2. From the Home or Explore page, navigate to the workbook you want to set a policy for.
3. Click the details icon ⓘ.
4. From the Workbook Details dialog, click **Edit Data Freshness Policy**.
5. Choose one of the following options:
   Site default (12 hours)
   Always live (Tableau will always get the latest data)
   Ensure data is fresh every
   Ensure data is fresh at
6. Click **OK**.



Site default refreshes your data every 12 hours, which is a great option if your audience regularly uses your dashboard, but doesn't need up-to-the-minute data freshness.

Always live provides the freshest data at all times, which can increase loading time.

Ensure data is fresh every... allows you to specify how often data is refreshed with the granularity of minutes, hours, days, or weeks.

Ensure data is fresh at... allows you to schedule the time and day for data refreshes. If you have an important meeting every Monday, Wednesday, and Friday at 09:00 AM Pacific time, then you can set your data refresh to occur at 08:45 AM every Monday, Wednesday, and Friday, so you have the freshest data when your meeting starts.

Use Dynamic Axis Ranges

It's key that your users understand the range represented in a viz so that they can correctly analyze the data. Especially when analyzing multiple vizzes in a dashboard or multiple worksheets at the same time, it can be easy for users to misinterpret data when the range of the axes is different between vizzes. For example, when two bar charts appear next to each other, the bars in both charts might appear to be equivalent sizes. However, the axes might have very different ranges, making the charts misleading.

To help users understand the range of the axis, in 2023.3 and later, authors can use Dynamic Axis Ranges to set the minimum and maximum values of an axis range by using numeric parameters or date parameters. Then, as users navigate across vizzes, the axes update synchronously. This makes it easier for your users to analyze data across vizzes easily and accurately. And by limiting the range of the data, you can view a subset of data without filtering the underlying data or impacting the moving average of your data.

Dynamic Axis Ranges can also be used to extend or shorten the range of an axis without filtering out the underlying data. This is ideal for showing progress against a goal or showing a moving average.

Supported field types

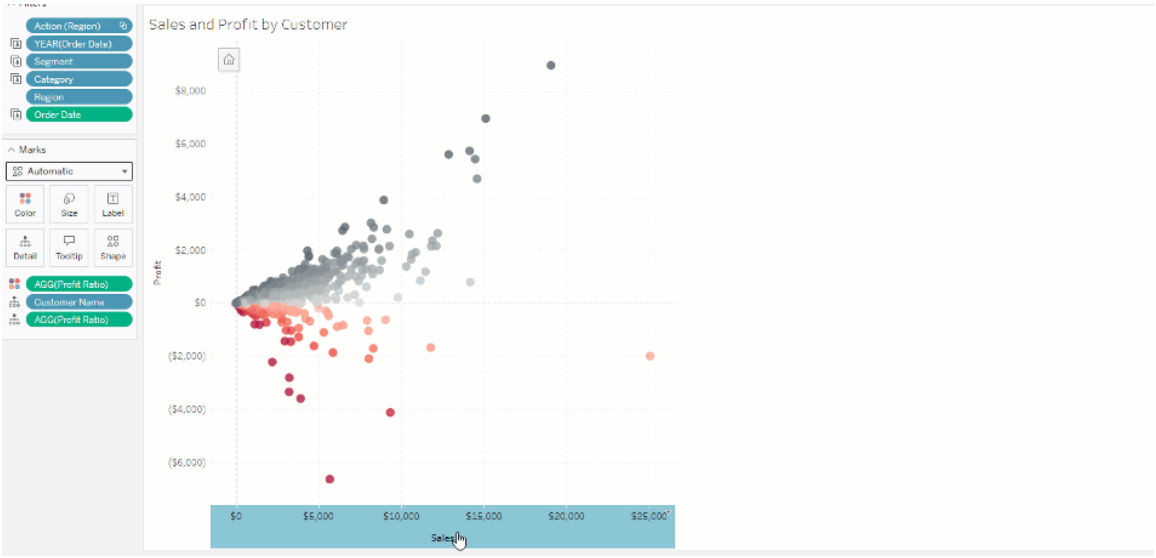
Dynamic Axis Ranges support any parameter that is compatible with the selected continuous axis, for example:

- Numeric parameters
- Temporal parameters (date or dateTime)

Configure a dynamic axis range

1. From a Tableau sheet, create a parameter for your axis.
2. Right-click the axis, and choose **Edit Axis**.
3. For the Range, choose **Custom**. Then, select the parameter that you created for either the start or end axis extent.
4. Close the dialog box.

5. If you're using multiple sheets on a dashboard, repeat these steps for the axes across the different sheets.



Understand limitations and edge cases

Dynamic Axis Ranges aren't updated within Stories. If the parameter used for the axis is deleted, then the most recent parameter value remains as the axis extent. The next time you edit the axis, an error message will prompt you to choose a new parameter to be used.

Use Dynamic Axis Titles

Tableau's flexibility empowers authors to quickly analyze different data sets and visualize data differently for deeper analysis. But with that flexibility, comes the responsibility to communicate accurately the data that is used in a viz. For example, it's critical to communicate the units of measure that change based on a parameter value. Imagine that you're looking at a viz of the weather that shows a forecast of 25 degrees. In Fahrenheit, that forecast might be a great ski trip. But in Celsius, that forecast might call for a trip to the beach.

As an author, you can use Dynamic Axis Titles to update the axis title based on the value of a parameter or a single-value field (for example, an LOD calculation). If you use swap parameters, the axis titles update to match the data being used.

Supported field types

To be used as a Dynamic Axis Title, fields must be:

- A single-valued and a fixed LOD calculation
- A parameter
- A constant calculated field
- A top 1 set

Configure a dynamic axis title

1. From a Tableau sheet, drag a continuous field onto a shelf.
2. Double-click the axis to open the **Edit Axis** dialog.
3. Under the **Axis Titles** section, choose the field you want to use for your axis title from the list.
4. Close the **Edit Axis** dialog.

For a more complex use case, first follow the steps in Example: Swap Measures Using Parameters. Then, follow these steps:

1. Double-click the X axis to open the **Edit Axis** dialog.
2. Under the **Axis Titles** section, select **Parameter**, and then choose the Placeholder 2 selector.
3. Close the **Edit Axis** dialog.
4. Repeat these steps for the Y axis using Placeholder 1.

Now, when the parameters are changed, the axes titles update to reflect the data being displayed.

Understand limitations and edge cases

Dynamic axis titles are cleared when the viz type is changed using Show Me. Also if you use subtitles, then the subtitle appears after the Dynamic Axis Title, just as it does for custom titles. Automatic subtitles are populated only when you have a continuous date value axis that is filtered to a single year with at least two time periods. The dynamic axis title functionality doesn't work when using the worksheet in a story (a sequence of visualizations that work together to convey information).

Use Dynamic Zone Visibility

Dashboard space is valuable, especially when you want to progressively reveal insights about data. With Dynamic Zone Visibility, you can hide or reveal zones (tiled or floating dashboard elements) based on the value of a field or parameter. As you interact, zones on your dashboard appear or disappear. The result is a dynamic dashboard that doesn't compromise your desired layout.

While you can show or hide objects by clicking a button on a dashboard, Dynamic Zone Visibility allows you to show and hide objects automatically. This is ideal for dashboards that are used by different user groups. For example, you might want to show different user groups different zones when they visit your dashboard.

And you can use Dynamic Zone Visibility with Parameter Actions. For example, when a user clicks a mark on a viz, a previously hidden zone appears. This is ideal for complex dashboards because it allows you to choose when deeper levels of data are revealed.

Supported field types

To be used for Dynamic Zone Visibility, a field or parameter must be:

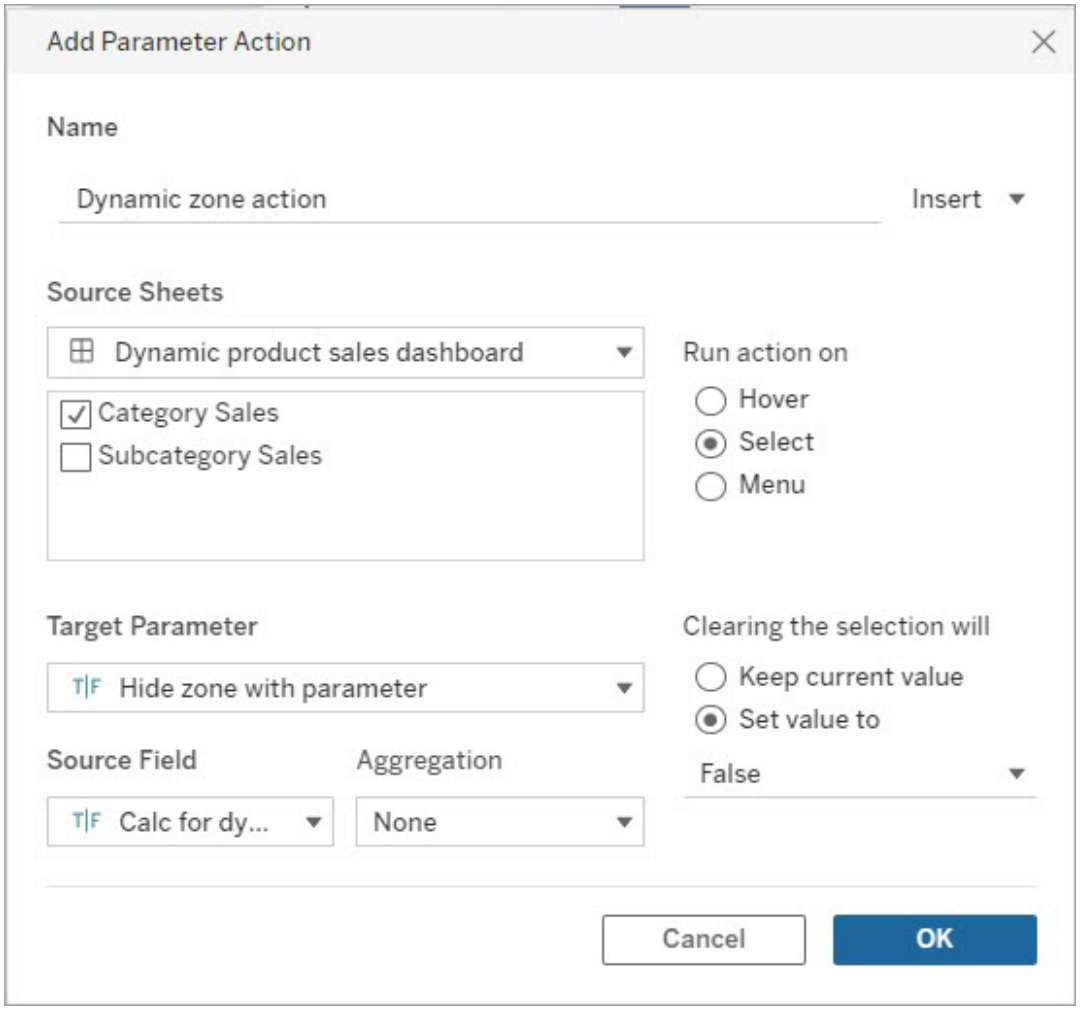
- Boolean.
- Single value.
- Independent of the viz, meaning the field returns a constant value independent of the structure of the viz, such as a fixed level of detail (LOD) calculation.

Configure a dynamic dashboard zone

The following example has two sheets that use Superstore data: the first sheet has a bar chart with Sales by Category, and the second sheet has a bar chart with Sales by Sub-Category. By using Dynamic Zone Visibility, the second sheet is visible only after a mark is clicked in the Sales by Category zone. This example relies on a boolean calculated field, which is used as the source field for a parameter action. For the calculation to be used as the source field for the parameter action, the calculation must be added to the marks card.

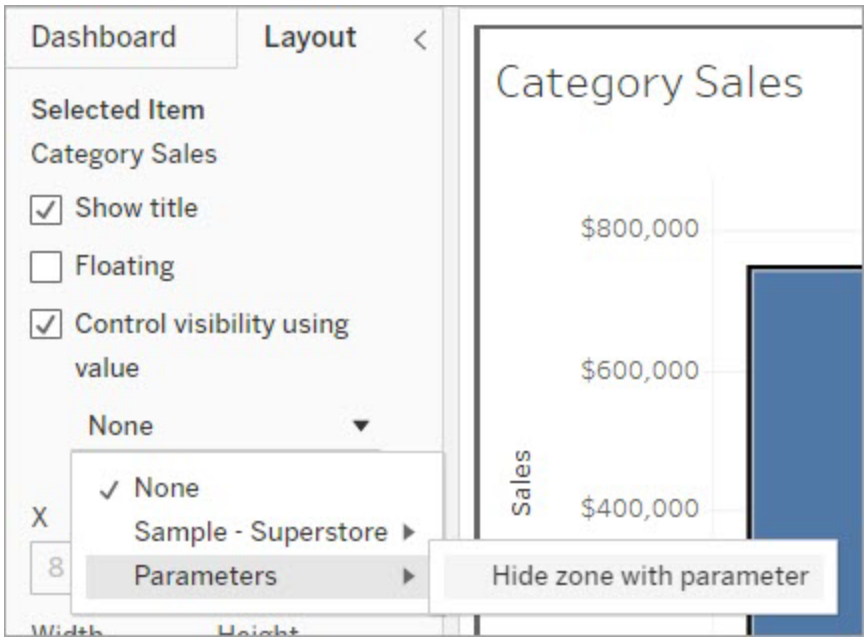
1. From the Sales by Category sheet, create a parameter. In this example, the parameter **Data type** must be set to **Boolean**.
2. From the Sales by Category sheet, create a calculated field. This example uses the following calculation: `True`
3. On the Sales by Category sheet, drag the calculation that you created to **Details** on the marks card.
4. Create a dashboard.
5. Drag the sheet that you always want to be visible onto your dashboard. In our example, we want Sales by Category to be visible.
6. From your dashboard, click the Sales by Category zone (dashboard object). Then, from the Worksheet menu on your dashboard, create a parameter action. This example uses the following **Change Parameter** action:
   The source sheet is set to use the dashboard you created and the Category Sales sheet.
   The **Target Parameter** is the parameter you created.

The **Source Field** is the calculation you created.

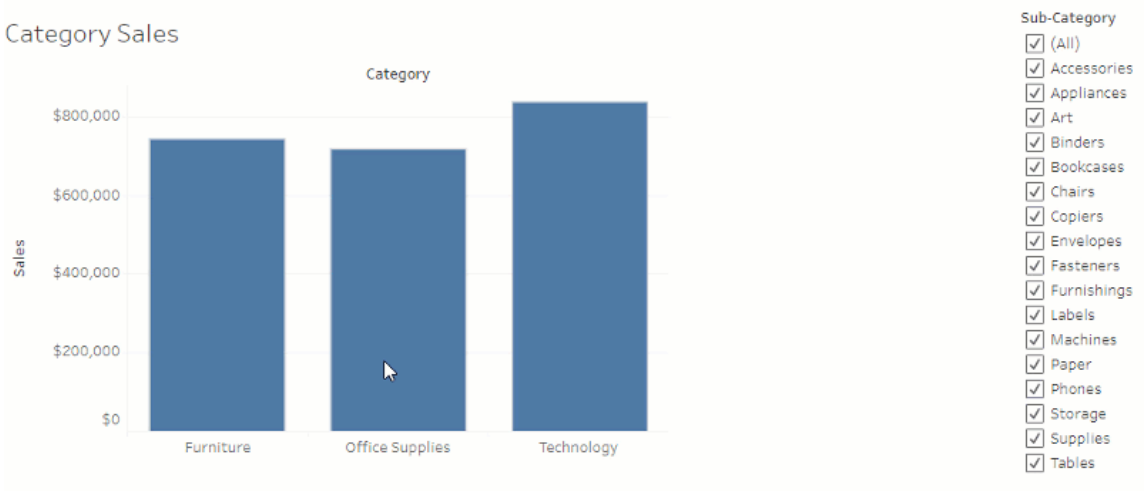


7. Drag the Sales by Sub-Category worksheet onto your dashboard.
8. Click the Sales by Category zone. From the upper right corner, click the dropdown arrow and select **Use as Filter**.
9. Click the Sales by Sub-Category zone, and then click the **Layout** tab.
10. Check the box for **Control visibility using value**.
11. From the dropdown, choose the parameter you created to control zone visibility.

Note: If the option to Control visibility using value doesn't contain the field you want to use, be sure that the field is a supported field type.



Now, when you click a category mark in the Sales by Category zone, the Sales by Sub-Category zone appears in your dashboard.



Manage Server

After installing Tableau Server, you can customize and manage your server. For example, you can manage security, licenses, sites, subscriptions and data-driven alerts, and more.

Security

As a part of managing Tableau Server, you can configure authentication, data security, and network security.

Authentication

Authentication verifies a user's identity. Everyone who needs to access Tableau Server—whether to manage the server, or to publish, browse, or administer content—must be represented as a user in the Tableau Server repository. The method of authentication may be performed by Tableau Server (“local authentication”), or authentication may be performed by an external process. In the latter case, you must configure Tableau Server for external authentication technologies such as Kerberos, SAML, or OpenID. In all cases, whether authentication takes place locally or is external, each user identity must be represented in the Tableau Server repository. The repository manages authorization meta data for user identities.

Looking for Tableau Server on Windows? See Authentication.

Although all user identities are ultimately represented and stored in the Tableau Server repository, you must manage user accounts for Tableau Server in an identity store. There are two, mutually exclusive, identity store options: LDAP and local. Tableau Server supports arbitrary

LDAP directories, but it's been optimized for Active Directory LDAP implementation. Alternatively, if you are not running an LDAP directory, you can use the Tableau Server local identity store. For more information see Identity Store.

As shown in the following table, the type of identity store you implement, in part, will determine your authentication options.

Identity Store	**Authentication Mechanism**								
	Basic	SAML	Site SAML	Kerberos	(Windows only) Automatic Logon (Microsoft SSPI)	OpenID Connect	Connected Apps	Trusted Auth	Mutual SSL
Local	X	X	X			X	X	X	X
Active Directory	X	X		X	X		X	X	X
LDAP	X	X					X	X	X

Access and management permissions are implemented through site roles. Site roles define which users are administrators, and which users are content consumers and publishers on the server. For more information about administrators, site roles, groups, Guest User, and user-related administrative tasks, see Users and Site Roles for Users.

Note: In the context of authentication, it's important to understand that users are not authorized to access external data sources through Tableau Server by virtue of having an account on the server. In other words, in the default configuration, Tableau Server does not act as a proxy to external data sources. Such access requires additional configuration of the data source on Tableau Server or authentication at the data source when the user connects from Tableau Desktop.

Add-on authentication compatibility

Some authentication methods can be used together. The following table shows authentication methods that can be combined. Cells marked with an "X" indicate a compatible authentication set. Blank cells indicate incompatible authentication sets.

	Connected Apps	Trusted Authentication	Server-wide SAML	Site SAML	Kerberos	(Windows only) Automatic Logon (Microsoft SSPI)	Mutual SSL	OpenID Connect
Tableau Connected Apps	N/A		X	X	X		X	X
Trusted Authentication		N/A	X	X	X		X	X
Server-	X	X	N/A	X				

wide SAML								
Site SAML	X	X	X	N/A				
Kerberos	X	X			N/A			
Automatic Logon (Microsoft SSPI)						N/A		
Mutual SSL	X	X					N/A	
OpenID Connect	X	X						N/A
Personal Access Token (PAT)	*	*	*	*	*	*	*	*

* PATs, by design, do not work directly with the authentication mechanism listed in these columns to authenticate to the REST API. Instead, PATs use Tableau Server user account credentials to authenticate to the REST API.

Client authentication compatibility

Authentication handled through a user interface (UI)

Clients	Authentication Mechanism									
	Basic	SAML	Site SAML	Kerberos	(Windows only) Automatic Logon (Microsoft SSPI)	OpenID Connect	Connected Apps	Trusted Auth	Mutual SSL	Personal Access Token (PAT)
Tableau Desktop	X	X	X	X	X	X			X	
Tableau Prep Builder	X	X	X	X	X	X			X	
Tableau Mobile	X	X	X	X (iOS only *)	X **	X			X	
Web	X	X	X	X	X	X	X	X	X	

Brows-ers							***			

* Kerberos SSO isn't supported for Android, but a fall back to user name and password is possible. For more information, see Note 5: Android platform.

** SSPI is not compatible with the Workspace ONE version of the Tableau Mobile app.

*** In embedding workflows only.

Authentication handled programmatically

Cli-ents	**Authentication Mechanism**									
	Bas-ic	SA-ML	Site SA-ML	Ker-beros	(Win-dows only) Auto-matic Logon (Micro-soft SSPI)	Open-ID Con-nect	Con-nected Apps	Trus-ted Auth	Mutu-al SSL	Per-sonal Acces-s Token (PAT)
RES-T API	X						X			X
tab-cmd 2.0	X									X
tab-cmd	X									

Local authentication

If the server is configured to use local authentication, then Tableau Server authenticates users. When users sign-in and enter their credentials, either through Tableau Desktop, tabcmd, API, or web client, Tableau Server verifies the credentials.

To enable this scenario, you must first create an identity for each user. To create an identity, you specify a username and a password. To access or interact with content on the server, users must also be assigned a site role. User identities can be added to Tableau Server in the server UI, using tabcmd Commands, or using the REST API.

You can also create groups in Tableau Server to help manage and assign roles to large sets of related user groups (e.g., “Marketing”).

When you configure Tableau Server for local authentication, you can set password policies and account lockout on failed password attempts. See Local Authentication.

Note: Tableau with multi-factor (MFA) authentication is available for Tableau Cloud only.

External authentication solutions

Tableau Server can be configured to work with a number of external authentication solutions.

Kerberos

You can configure Tableau Server to use Kerberos for Active Directory. See Kerberos.

SAML

You can configure Tableau Server to use SAML (security assertion markup language) authentication. With SAML, an external identity provider (IdP) authenticates the user's credentials, and then sends a security assertion to Tableau Server that provides information about the user's identity.

For more information, see SAML.

OpenID Connect

OpenID Connect (OIDC) is a standard authentication protocol that lets users sign in to an identity provider (IdP) such as Google. After they've successfully signed in to their IdP, they are automatically signed in to Tableau Server. To use OIDC on Tableau Server, the server must be configured to use the local identity store. Active Directory or LDAP identity stores are not supported with OIDC. For more information, see OpenID Connect.

Mutual SSL

Using mutual SSL, you can provide users of Tableau Desktop, Tableau Mobile, and other approved Tableau clients a secure, direct-access experience to Tableau Server. With mutual SSL, when a client with a valid SSL certificate connects to Tableau Server, Tableau Server confirms the existence of the client certificate and authenticates the user, based on the username in the client certificate. If the client does not have a valid SSL certificate, Tableau Server can refuse the connection. For more information, see Configure Mutual SSL Authentication.

Connected apps

Direct trust

Tableau connected apps enable a seamless and secure authentication experience by facilitating an explicit trust relationship between your Tableau Server site and external applications where Tableau content is embedded. Using connected apps also enables a programmatic way to authorize access to the Tableau REST API using JSON Web Tokens (JWTs). For more information, see Use Tableau Connected Apps for Application Integration.

EAS or OAuth 2.0 trust

You can register an external authorization server (EAS) with Tableau Server to establish a trust relationship between your Tableau Server and an EAS using the OAuth 2.0 standard protocol. The trust relationship provides your users with single sign-on experience, through your IdP, to embedded Tableau content. In addition, registering an EAS enables a programmatic way to authorize access to the Tableau REST API using JSON Web Tokens (JWTs). For more information, see Configure Connected Apps with OAuth 2.0 Trust.

Trusted authentication

Trusted authentication (also referred to as "Trusted tickets") lets you set up a trusted relationship between Tableau Server and one or more web servers. When Tableau Server receives requests from a trusted web server, it assumes that the web server has already handled whatever authentication is necessary. Tableau Server receives the request with a redeemable token or ticket and presents the user with a personalized view which takes into consideration the user's role and permissions. For more information, see Trusted Authentication.

LDAP

You can also configure Tableau Server to use LDAP for user authentication. Users are authenticated by submitting their credentials to Tableau Server, which will then attempt to bind to the LDAP instance using the user credentials. If the bind works then the credentials are valid and Tableau Server grants the user a session.

"Binding" is the handshake/authentication step that happens when a client tries to access an LDAP server. Tableau Server does this for itself when it makes various non-authentication related queries (such as importing users and groups).

You can configure the type of bind you want Tableau Server to use when verifying user credentials. Tableau Server supports GSSAPI and simple bind. Simple bind passes credentials directly to the LDAP instance. We recommend that you configure SSL to encrypt the bind communication. Authentication in this scenario maybe be provided by the native LDAP solution, or with an external process, like SAML.

For more information about planning for and configuring LDAP, see Identity Store and External Identity Store Configuration Reference.

Other authentication scenarios

- REST API: Signing In and Out (Authentication)

  Note: REST API does not support SAML single-sign (SSO).

- Mobile device authentication: Single sign-on for Tableau Mobile
- Certificate trust for TSM clients: Connecting TSM clients
- PAM integration for TSM administration: TSM Authentication

Data access and source authentication

You can configure Tableau Server to support a number of different authentication protocols to various different data sources. Data connection authentication may be independent of Tableau Server authentication.

For example, you may configure user authentication to Tableau Server with local authentication, while configuring OAuth or SAML authentication to specific data sources. See Data Connection Authentication.

Local Authentication

If the server is configured to use local identity store, then Tableau Server authenticates users. When users sign-in and enter their credentials, either through Tableau Desktop, tabcmd, API, or web client, Tableau Server verifies the credentials. Tableau user names stored in the identity store are associated with rights and permissions for Tableau Server. After authentication is verified, Tableau Server manages user access (authorization) for Tableau resources.

To use local authentication, you must configure Tableau Server with a local identity store during Setup. You cannot use local authentication if your Tableau Server has been configured with an external identity store (LDAP, Active Directory, etc).

Note: Identity pools, which is a tool designed to complement and support additional user provisioning and authentication options you might need in your organization, supports OpenID Connect (OIDC) authentication only. For more information, see Provision and Authenticate Users Using Identity Pools.

Password storage

When local authentication is used, the user's salted and hashed password is stored in the repository. Passwords are never stored directly, rather, the result of salting and hashing the password is stored. Server uses the PBKDF2 derivation function with the HMAC SHA512 hashing function.

Configure password settings

After you install Tableau Server with local authentication, you can use Tableau Server Manager (TSM) to configure a number of password-related settings:

- Password policies: these policies define the requirement for password structure, such as length, character types, and other requirements.
- Password expiration: enable and specify password expiry.
- Login rate limit: Tableau Server throttles the time between sign-in attempts after users enter 5 incorrect passwords. Users will need to wait a few seconds before attempting another sign-in. If users continue to enter incorrect passwords, then they must wait for exponentially longer periods of time in between sign-in attempts. By default, the maximum time between sign-in attempts is 60 minutes.

  Lock out account access after too many failed attempts. You can specify how many failed attempts users are allowed to enter before they are locked out. For information on how to unlock access to a locked account, see View and manage users on a site .
- User password reset: Enable users to reset passwords. Enabling password reset will configure Tableau Server to display a link on the sign-in page. Users who forget passwords or want to reset a password can click the link to initiate a reset-password workflow. Password reset must be configured using TSM CLI, as described below.

Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click on **User Identity & Access** on the **Configuration** tab and then click **Authentication Method**.

3. Select Local authentication from the drop-down menu to display the password settings.

4. Configure the password settings and then click **Save Pending Changes**.

5. Click **Pending Changes** at the top of the page:

6. Click **Apply Changes and Restart**.

Use the TSM CLI

For the initial configuration of password policies, we recommend that you use the configuration file template below to create a json file. You can also set any single configuration key listed below with the syntax described in tsm configuration set.

1. Copy the following json template to a file. Fill in the key values with your password policy configuration. See the reference section that follows for more information about key options.

```
{
"configKeys": {
 "wgserver.localauth.policies.mustcontainletters.enabled":
false,
  "wgserver.localauth.policies.mustcontainuppercase.enabled":
false,
```

```
  "wgserver.localauth.policies.mustcontainnumbers.enabled":
false,

  "wgserver.localauth.policies.mustcontainsymbols.enabled":
false,

  "wgserver.localauth.policies.minimumpasswordlength.enabled":
false,

  "wgserver.localauth.policies.minimumpasswordlength.value": 8,

  "wgserver.localauth.policies.maximumpasswordlength.enabled":
false,

  "wgserver.localauth.policies.maximumpasswordlength.value":
255,

  "wgserver.localauth.passwordexpiration.enabled": false,

  "wgserver.localauth.passwordexpiration.days": 90,

  "wgserver.localauth.ratelimiting.maxbackoff.minutes": 60,

  "wgserver.localauth.ratelimiting.maxattempts.enabled": false,

  "wgserver.localauth.ratelimiting.maxattempts.value": 5,

  "vizportal.password_reset": false

                }

}
```

2. Run the `tsm settings import -f file.json` to pass the json file with the appropriate values to Tableau Services Manager to configure Tableau Server.

3. Run the `tsm pending-changes apply` command to apply the changes. See tsm pending-changes apply.

Configuration file reference

This section lists all of the options that can be used to configure password polices.

`wgserver.localauth.policies.mustcontainletters.enabled`

Default value: `false`

Require at least one letter in passwords.

`wgserver.localauth.policies.mustcontainuppercase.enabled`

Default value: `false`

Require at least one upper-case letter in passwords.

`wgserver.localauth.policies.mustcontainnumbers.enabled`

Default value: `false`

Require at least one number letter in passwords.

`wgserver.localauth.policies.mustcontainsymbols.enabled`

Default value: `false`

Require at least one special character in passwords.

`wgserver.localauth.policies.minimumpasswordlength.enabled`

Default value: `false`

Enforce minimum-length passwords.

`wgserver.localauth.policies.minimumpasswordlength.value`

Default value: `8`

The minimum number of characters passwords must have. Enter a value between `4` and `255`, inclusive. You must set `wgserver.localauth.policies.minimumpasswordlength.enabled` to `true` to enforce this value.

`wgserver.localauth.policies.maximumpasswordlength.enabled`

Default value: `false`

Enforce maximum-length passwords.

`wgserver.localauth.policies.maximumpasswordlength.value`

Default value: `255`

The maximum number of characters passwords may have. Enter a value between `8` and `225`, inclusive. You must set `wgserver.localauth.policies.maximumpasswordlength.enabled` to `true` to enforce this value.

`wgserver.localauth.passwordexpiration.enabled`

Default value: `false`

Enforce password expiry.

`wgserver.localauth.passwordexpiration.days`

Default value: `90`

The number of days before a password expires. Enter a value between `1` and `365`, inclusive. You must set `wgserver.localauth.passwordexpiration.enabled` to `true` to enforce this value.

`wgserver.localauth.ratelimiting.maxbackoff.minutes`

Default value: `60`

Maximum time between sign-in attempts after a user enters multiple incorrect passwords. Enter a value between `5` and `1440`, inclusive.

`wgserver.localauth.ratelimiting.maxattempts.enabled`

Default value: `false`

Enforce account lock out after 5 incorrect passwords are entered. To change the number of incorrect passwords that will trigger account lock out, you set `wgserver.localauth.ratelimiting.maxattempts.value`.

`wgserver.localauth.ratelimiting.maxattempts.value`

Default value: `5`

The number of incorrect passwords that a user may enter to trigger account lock out. Enter a value between `5` and `100`, inclusive. You must set `wgserver.localauth.ratelimiting.maxattempts.enabled` to `true` to enforce this value.

vizportal.password_reset

Default value: `false`

Enable users to reset passwords. Tableau Server must be configured to send email for this feature to operate. See Configure SMTP Setup.

SAML

SAML (Security Assertion Markup Language) is an XML standard that allows secure web domains to exchange user authentication and authorization data. You can configure Tableau Server to use an external identity provider (IdP) to authenticate users over SAML 2.0. No user credentials are stored with Tableau Server, and using SAML enables you to add Tableau to your organization's single sign-on environment.

You can use SAML server wide, or you can configure sites individually. Here's an overview of those options:

- **Server-wide SAML authentication**. A single SAML IdP application handles authentication for all Tableau Server users. Use this option if your server has only the Default site, as it is unnecessary to configure site specific SAML in this case. You may also use

Server-wide SAML in multisite environments, but users are limited to a single IdP to across all sites.

- **Server-wide local authentication and site-specific SAML authentication**. In a multi-site environment, users who are not enabled for SAML authentication at the site level can sign in using local authentication.
- **Server-wide SAML authentication and site-specific SAML authentication**. In a multi-site environment, all users authenticate through a SAML IdP configured at the site level, and you specify a server-wide default SAML IdP for users that belong to multiple sites.

If you want to use site-specific SAML, you must configure server-wide SAML before you configure individual sites. Server-side SAML does not need to be enabled for site-specific SAML to function, but it must be configured.

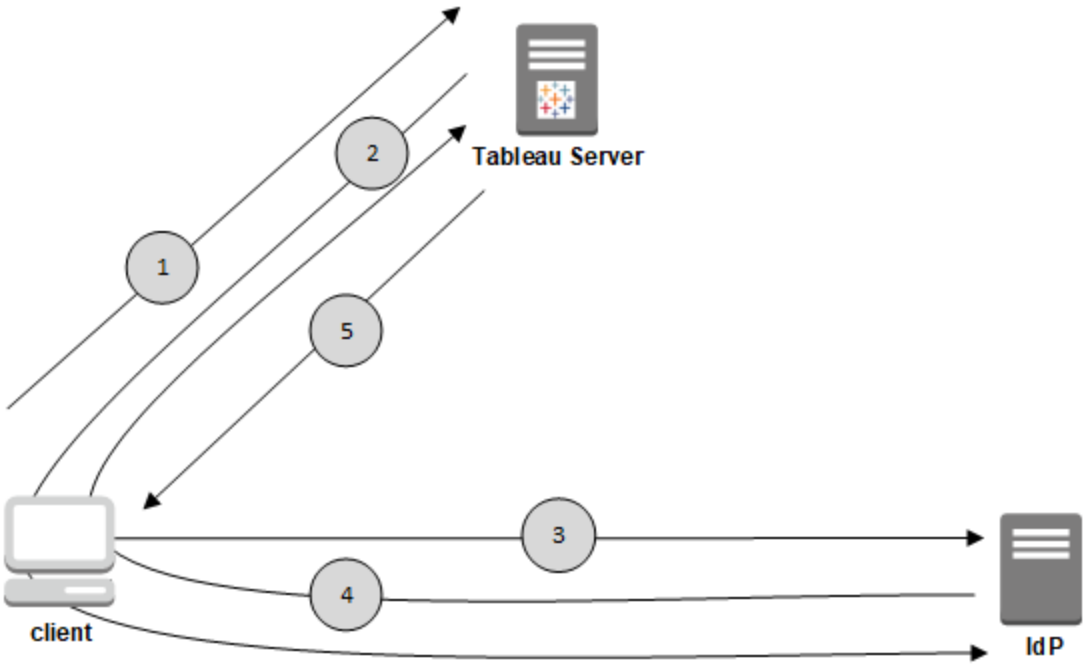
User authentication through SAML does not apply to permissions and authorization for Tableau Server content, such as data sources and workbooks. It also does not control access to underlying data that workbooks and data sources connect to.

Notes:

- Tableau Server supports both service provider initiated and IdP initiated SAML in browsers and in the Tableau Mobile app. SAML connections from Tableau Desktop must be service provider initiated.
- Identity pools, which is a tool designed to complement and support additional user provisioning and authentication options you might need in your organization, supports OpenID Connect (OIDC) authentication only. For more information, see Provision and Authenticate Users Using Identity Pools.

Authentication overview

The following image shows the steps to authenticate a user with single sign-on in a typical service provider initiated flow:



1. User navigates to the Tableau Server sign-in page or clicks a published workbook URL.
2. Tableau Server starts the authentication process by redirecting the client to the configured IdP.
3. The IdP requests the user's username and password from the user. After the user submits valid credentials, the IdP authenticates the user.
4. The IdP returns the successful authentication in the form of a SAML Response to the client. The client passes the SAML Response to Tableau Server.
5. Tableau Server verifies that the username in the SAML Response matches a licensed user stored in the Tableau Server Repository. If a match is verified, then Tableau Server responds to the client with the requested content.

SAML Requirements

Before you configure SAML on Tableau Server, make sure your environment meets the requirements.

Important: SAML configurations, both with the IdP and on Tableau Server, are case sensitive. For example, URLs configured with the IdP and on Tableau Server must match exactly.

Certificate and identity provider (IdP) requirements

To configure Tableau Server for SAML, you need the following:

- **Certificate file.** A PEM-encoded x509 certificate file with a **.crt** extension. This file is used by Tableau Server, not the IdP. If you have an SSL certificate, it is possible in some circumstances to use the same certificate with SAML. For more information, see Using SSL certificate and key files for SAML later in this article.

  Tableau Server requires a certificate-key pair to sign the request that is sent to the IdP. This reduces the threat of a man-in-the-middle attack given the difficulty of spoofing a signed request. Additionally, Tableau Server verifies that the AuthNResponse it receives from the trusted IdP. Tableau Server verifies the AuthNResponse by using the signature produced by the IdP. The IdP certificate metadata is provided to Tableau Server as part of the initial SAML configuration process.

  Signed requests are not always necessary for all IdPs. By default, Tableau Server requires signed requests. We recommend this configuration to ensure a more secure communication transmission with the IdP. Work with your IdP team to understand if disabling signed requests is necessary. To disable signed requests see samlSettings Entity.

- **Signature algorithm.** The certificate must use a secure signature algorithm, for example SHA-256. If you attempt to configure Tableau Server for SAML with a certificate that uses SHA-1 signature hash, Tableau Server will reject the certificate. You can configure Tableau Server to accept the less-secure SHA-1 hash by setting the tsm wgserver.saml.blocklisted_digest_algorithms configuration key.

- **RSA key and ECDSA curve sizes.** The Tableau Server certificate must have an RSA key strength of 2048, and the IdP certificate must have either an RSA key strength of 2048 or ECDSA curve size of 256.

You can configure Tableau Server to accept the less-secure sizes by setting the respective configuration keys, wgserver.saml.min_allowed.rsa_key_size and wgserver.saml.min_allowed.elliptic_curve_size.

- **Certificate key file.** An RSA or DSA private key file that has the **.key** extension. RSA keys must be in in PKCS#1 or PKCS#8 format.

  Password protection requirements are as follows:

  - The PKCS#1 RSA key file cannot be password protected.
  - To use a password-protected key file, you must configure SAML with a RSA PKCS#8 file. **Note:** A PKCS#8 file with a null password is not supported.
  - Password-protected key files are not supported in site-specific SAML deployments.

  Summary of support

Key file format	Server-Wide SAML Support	Site-Level SAML Support
PKCS#8 RSA	Yes	No
PKCS#8 (no/null password)	No	No
PKCS#1 RSA	Yes	Yes
PKCS#1 RSA (password)	No	No
PKCS#1 DSA (password)	No	No

- **IdP must sign SAML assertions with a secure signature algorithm.** By default, Tableau Server will reject SAML assertions signed with the SHA-1 algorithm. You can configure Tableau Server to accept assertions signed with the less-secure SHA-1 hash

by setting the tsm wgserver.saml.blocklisted_digest_algorithms configuration key.

- **IdP account that supports SAML 2.0 or later.** You need an account with an external identity provider. Some examples are PingFederate, SiteMinder, and Open AM.
- **IdP provider that supports import and export of XML metadata.** Although a manually created metadata file might work, Tableau Technical Support cannot assist with generating the file or troubleshooting it.
- **Username:** Required. The IdP configuration must include the "username" attribute or claim and the corresponding SAML configuration attribute on Tableau Server must be set to "username" as well.

SSL off-loading

If your organization terminates SSL connections from the IdP at a proxy server before sending the authentication request to Tableau Server, then you may need to make a proxy configuration. In this scenario, SSL is "off-loaded" at the proxy server, which means the https request is terminated at the proxy server and then forwarded to Tableau Server over http.

Tableau Server validates the SAML response message returned from the IdP. Since SSL is off-loaded at the proxy, Tableau Server will validate with the protocol that it receives (http), but the IdP response is formatted with https, so validation will fail unless your proxy server includes the X-Forwarded-Proto header set to `https`. See Configure Tableau Server to work with a reverse proxy server and/or load balancer.

Using SSL certificate and key files for SAML

If you are using a PEM-encoded x509 certificate file for SSL, you can use the same file for SAML. For SSL, the certificate file is used to encrypt traffic. For SAML, the certificate is used for authentication.

In addition to the requirements listed in Certificate and identity provider (IdP) requirements above, to use the same certificate for both SSL and SAML, the certificate must also meet the following condition to work for SAML:

- Confirm that the certificate includes only the certificate that applies to Tableau Server and not any other certificates or keys.

  To do this, you can create a backup copy of the certificate file, and then open the copy in a text editor to review its contents.

User management requirements

When you enable SAML, user authentication is performed outside of Tableau, by the IdP. However, user management is performed by an identity store: either an external identity store (Active Directory or LDAP) or by Tableau Server in a local identity store. For more information about planning for user management with Tableau Server, see Identity Store.

When you configure the identity store during Setup, you must select the option that reflects how you want to use SAML. Note, if you want to use site-specific SAML, you must configure server-wide SAML before you configure individual sites.

- **For site-specific SAML:** If you have multiple sites on Tableau Server and want to set up each site for a particular IdP or IdP application (or configure some sites not to use SAML), configure Tableau Server to manage user with a local identity store. For site-specific SAML, Tableau Server relies on the IdP for authentication and does not use passwords.
- **For server-wide SAML:** If you configure server-wide SAML with a single IdP, you can configure Tableau Server to use the local identity store or an external identity store.
- **For both server-wide SAML authentication and site-specific SAML authentication:**
  - **When using a local identity store**, it is important that you use a username that has email address formatting. Using a complete email address helps to guarantee the uniqueness of the username in Tableau Server, even when two users have the same email prefix but have different email domains. To ensure uniqueness in identities, leverage full email address formatting across both systems or upgrade Tableau Server to version 2022.1.x or later and run the identity migration

background job.

- **In a multi-site environment**, all users authenticate through a SAML IdP configured at the site level. In this scenario, you specify a server-wide default SAML IdP for users who belong to multiple sites. To configure this scenario, Tableau Server must be configured with a local identity store.

- **Ignore domain when matching SAML username attribute**. Beginning in Tableau Server versions 2021.4.21, 2022.1.17, 2022.3.9, and 2023.1.5, you can configure Tableau Server to ignore the domain portion of the username attribute when matching the identity provider (IdP) user name to a user account on Tableau Server. For example, the username attribute in the IdP might be `alice@example.com` to match a user named `alice` in Tableau Server. Ignoring the domain portion of the username attribute might be useful if you already have users defined in Tableau Server that match the prefix portion of the username attribute but not the domain portion of the username attribute.

  Important: We do not recommend ignoring the domain name without taking precautions. Specifically, verify that user names are unique across the configured domains that you've created in your IdP. Configuring Tableau Server to ignore the domain name has the potential to result in unintended user sign-in. Consider the case where your IdP has been configured for multiple domains (e.g., `example.com` and `tableau.com`). If two users with the same first name, but different user accounts (e.g., `alice@tableau.com` and `alice@example.com`) are in your organization, then you could have a mapping mismatch.

  To configure Tableau Server to ignore domain names in the username attribute from the IdP, set `wgserver.ignore_domain_in_username_for_matching` to `true`. For more information, see wgserver.ignore_domain_in_username_for_matching.

  Notes:

- This command only works in Tableau Server deployments that are in `legacy-identity-mode` or deployments that have not been updated through the identity migration to use the Identity Service.
- When you change the tsm command to ignore the domain name in the username attribute, all user names in Tableau Server must have a domain name.

Note: The REST API and tabcmd do not support SAML single-sign (SSO). To sign in, you must specify the name and password of a user who has been created on the server. The user may be managed by the local identity store or an external identity store, depending on how you have configured Tableau Server. REST API or tabcmd calls will have the permissions of the user you sign in as.

SAML compatibility notes and requirements

- **Matching usernames**: The user name stored in Tableau Server must match the configured user name attribute sent by the IdP in the SAML assertion. By default, Tableau Server expects the incoming assertion to contain an attribute called "username" with that user's information. For example, if the user name for Jane Smith is stored in PingFederate as **jsmith**, it must also be stored in Tableau Server as **jsmith**.

  When configuring SAML during authentication

  If you are configuring SAML as part of the initial Tableau Server setup, make sure the account you plan to use exists in your IdP before you run setup. During Tableau Server setup you create the server administrator account.

  When running multiple domains

  If you use an Active Directory or LDAP external identity store and you are running in multiple domains (that is, users belong to multiple domains, or your Tableau Server installation includes multiple domains), the IdP must send both the user name *and* domain attributes for a user in the SAML assertion. Both these user name and domain attributes

must match exactly the user name and domain stored in Tableau Server. Do *one* of the following:

- Set `domain\username` in the username field
- Set domain in the domain field and set user name in the username field

When setting the domain attribute, you can use the fully qualified domain name (FQDN) or the shortname.

Where the domain isn't specified, it will be considered the default domain.

For more information, see Support for multiple domains and the "Match Assertions" section in the **Use TSM CLI** tab of Configure Server-Wide SAML.

- **Signature algorithm**: Tableau Server uses SHA256 signature algorithm.
- **Single Log Out (SLO)**: Tableau Server supports both service provider (SP)-initiated SLO and identity provider (IdP)-initiated SLO for both server-wide SAML and site-specific SAML.
- **External authentication types**: Tableau Server supports using one external authentication type at a time.
- **Mutual SSL**: Tableau Server does not support mutual SSL (two-way SSL) and SAML together. If you want to use mutual SSL, you can configure it on the IdP.
- **Assertions encoding**: Assertions must be UTF-8 encoded.
- **Encryption and SAML assertions**:
    - **Server-wide SAML:** When Tableau Server is configured for server-wide SAML, Tableau Server supports encrypted assertions from the IdP. Encryption assertions are enabled by the certificate that you upload as part of the initial configuration for server-wide SAML. SAML requests and responses can be sent over HTTP or HTTPS.

    - **Site-specific SAML:** When Tableau Server is configured for site-specific SAML, Tableau Server does not support encrypted assertions from the IdP. However, all SAML requests and responses are sent over HTTPS to secure communication with the IdP. HTTP requests and responses are not supported.

- **User identity in Tableau Server for tabcmd users**: As described in User management requirements section above, to use tabcmd, you must sign in as a user defined on the server. You cannot use SAML accounts with **tabcmd**.

- **Using SAML SSO with Tableau Desktop**: By default, Tableau Desktop allows SP-initiated SAML authentication.

  If your IdP does not support this functionality, you can disable SAML sign-in for Tableau Desktop using the following command:

  ```
  tsm authentication saml configure --desktop-access disable
  ```

  For more information, see tsm authentication saml <commands>.

- **Distributed installations**: TSM versions of Tableau Server (2018.2 and newer) use the Client File Service to share files in a multi node cluster. After you have configured SAML on the initial node in your cluster, the Client File Service will distribute certificate and key files to the other nodes.

- **Login URL**: For users to be able to sign in, your IdP must be configured with SAML Login endpoint that sends a POST request to the following URL:

  `https://<tableauserver>/wg/saml/SSO/index.html`.

- **Logout URL**: To enable users to sign out after signing in with SAML (single logout, or SLO), your IdP must be configured with a SAML Logout endpoint that sends a POST request to the following URL:

  `https://<tableauserver>/wg/saml/SingleLogout/index.html`.

Note: Tableau Server supports both service provider (SP)-initiated SLO and identity provider (IdP)-initiated SLO for both server-wide SAML and site-specific SAML.

- **Post-logout redirect URL**: By default, when a user signs out of Tableau Server, the sign-in page is displayed.

  To display a different page after sign-out, use the `tsm authentication saml configure` command with the `-su` or `--signout-url` option.

  - To specify an absolute URL, use a fully-qualified URL starting with `http://` or `https://`, as in this example:

    ```
    tsm authentication saml configure -su https://example.com
    ```

  - To specify a URL relative to the Tableau Server host, use a page starting with a / (slash):

    ```
    tsm authentication saml configure -su /ourlogoutpage.html
    ```

- **Active Directory Federation Service (AD FS)**: You must configure AD FS to return additional attributes for Tableau authentication with SAML. The **Name ID** and **username** attributes can be mapped to the same AD attribute: **SAM-Account-Name**.

- **AuthNContextClassRef** : AuthNContextClassRef is an optional SAML attribute that enforces validation of certain authentication "contexts" in IdP initiated flows. You can set comma-separated values for this attribute with TSM. When this attribute is set, Tableau Server validates that the SAML response contains at least one of the values listed. If the SAML response does not contain one of the configured values, authentication will be rejected, even if the user has successfully authenticated with the IdP.

  Leaving this optional attribute blank will result in default behavior: any successfully authenticated SAML response will result in a user being granted a session within Tableau Server.

This value is only evaluated for server-wide SAML. If site-SAML is configured, the `AuthNContextClassRef` attribute will be ignored.

To set this value with TSM web interface, see Configure Server-Wide SAML.

To set this value with tsm configuration set, use the key, `wgserver.saml.authcontexts`, to set a comma-separated list of values.

To set this value with a JSON configuration file, see samlSettings Entity.

Using SAML SSO with Tableau client applications

Tableau Server users with SAML credentials can sign in to the server from Tableau Desktop or the Tableau Mobile app. For full compatibility, we recommend that the Tableau client application version matches that of the server. To connect using site-specific SAML, users must run version 10.0 or later of the Tableau client application.

Connecting to Tableau Server from Tableau Desktop or Tableau Mobile uses a service provider (SP) initiated connection.

Redirecting authenticated users back to Tableau clients

When a user signs in to Tableau Server, Tableau Server sends a SAML request (`AuthnRequest`) to the IdP, which includes the Tableau application's **RelayState** value. If the user has signed in to Tableau Server from a Tableau client such as Tableau Desktop or Tableau Mobile, it's important that the RelayState value is returned within the IdP's SAML response back to Tableau.

When the RelayState value is not returned properly in this scenario, the user is taken to their Tableau Server home page in the web browser, rather than being redirected back to the application they signed in from.

Work with your Identity Provider and internal IT team to confirm that this value will be included as part of the IdP's SAML response, and then preserved by any network appliance (such as a proxy or load balancer) that resides between your IdP and Tableau Server.

XML data requirements

As part of SAML configuration, you exchange XML metadata between Tableau Server and the IdP. This XML metadata is used to verify a user's authentication information when the user initiates the Tableau Server sign-in process.

Tableau Server and the IdP each generates its own metadata. Each set of metadata must contain the information described in the following list. If either set is missing information, errors can occur when you configure SAML or when users try to sign in.

- **HTTP POST and HTTP REDIRECT:** Tableau Server supports HTTP POST and HTTP REDIRECT requests for SAML communications. In the SAML metadata XML document that is exported by the IdP, the `Binding` attribute can be set to HTTP-POST or HTTP-REDIRECT.
- When the `Binding` attribute set to `HTTP-POST`, the SAML metadata that Tableau Server and the IdP each export must contain the following elements.
    - The element that specifies the URL that the IdP redirects to after successful authentication. This is required in the Service Provider metadata, not the Identity Provider metadata.

      ```
      <md:AssertionConsumerService Bind-
      ing="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Loca-
      tion="https://<tableau-server>/wg/saml/SSO/index.html
      index="0" isDefault="true"/>
      ```

      For Site SAML, the `Location` endpoint is `/samlser-vice/public/sp/metadata?alias=<site alias>`.
    - The logout endpoint element appears in Tableau Server metadata and specifies the URL that the IdP will use for Tableau Server's logout endpoint. If this element is not in the IdP metadata, Tableau Server cannot negotiate a logout endpoint with the IdP and the SAML Logout feature will not be available within Tableau

Server:

```
<md:SingleLogoutService Bind-
ing="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Loca-
tion="https://SERVER-NAME:9031/idp/slo"
```

- Verify that the metadata XML from the IdP includes a **SingleSignOnService** element, in which the binding is set to `HTTP-POST`, as in the following example:

```
<md:SingleSignOnService Bind-
ing="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Loca-
tion="https://SERVER-NAME:9031/idp/SSO.saml2"/>
```

- This element should appear in IdP metadata and specifies the URL that Tableau Server will use for the IdP's logout endpoint.

```
 <md:SingleLogoutService Bind-
ing="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Loca-
tion="https://SERVER-NAME:9031/idp/slo"/>
```

- **Attribute named *username***: You must configure the IdP to return an assertion that includes the `username` attribute in the `saml:AttributeStatement` element. The assertion's attribute type must be `xs:string` (it should *not* be typed as `xs:any`).

  The following example shows what this might look like.

```
<saml:Assertion assertion-element-attributes>
  <saml:Issuer>issuer-information</saml:Issuer>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    ...
  </Signature>
  <saml:Subject>
    ...
```

```
  </saml:Subject>
  <saml:Conditions condition-attributes >
    ...
  </saml:Conditions>
  <saml:AuthnStatement authn-statement-attributes >
    ...
  </saml:AuthnStatement>

  <saml:AttributeStatement>
    <saml:Attribute Name="username" NameFormat-
t="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
    <saml:AttributeValue xmlns:x-
s="http://www.w3.org/2001/XMLSchema" xmlns:x-
si="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">
          user-name
    </saml:AttributeValue>
    </saml:Attribute>
  </saml:AttributeStatement>
</saml:Assertion>
```

By default, Tableau Server will read the `username` attribute in the AuthNResponse returned from the Idp. However, some IdPs may return a different attribute that is intended to identify the user.

To change the SAML attribute that passes the `username` value, run the following TSM command:

```
tsm authentication saml map-assertions --user-name <USER-NAME>.
```

See tsm authentication.

- **Dynamic group membership using SAML assertions:**

Beginning in Tableau Server 2024.2, if SAML (or site SAML) is configured and the capability's setting enabled (server-wide or site-level), you can dynamically control group membership through custom claims included in the SAML XML response sent by the identity provider (IdP).

When configured, during user authentication, the IdP sends the SAML assertion that contains two custom group membership claims: group (`https://tableau.com/groups`) and group names (for example, "Group1" and "Group2") to assert the user into. Tableau validates the assertion and then enables access to the groups and the content whose permissions are dependent on those groups.

For more information, see Dynamic group membership using assertions.

Example SAML XML response

```
<saml2p:Response
  xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
    .....
    .....
  <saml2:Assertion
    .....
    .....
    xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
    <saml2:AttributeStatement
              xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
              <saml2:Attribute
              Name="https://tableau.com/groups"
                      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-form-
at:unspecified">
                      <saml2:AttributeValue
                              xmlns:xs="http://www.w3.org/2001/XMLSchema"
                              xmlns:xsi="http://www.w3.org/2001/XMLSchema-ins
                              xsi:type="xs:string">Group1
                      </saml2:AttributeValue>
                      <saml2:AttributeValue
                              xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```
                                    xmlns:xsi="http://www.w3.org/2001/XMLSchema-ins
                                    xsi:type="xs:string">Group2
                            </saml2:AttributeValue>
                    <saml2:Attribute>
                </saml2:AttributeStatement>
            </saml2:Assertion>
        </saml2p:Response>
```

Configure Server-Wide SAML

Configure server-wide SAML when you want all single sign-on (SSO) users on Tableau Server to authenticate through a single SAML identity provider (IdP), or as the first step to configuring site-specific SAML in a multi-site environment.

If you have configured server-wide SAML and are ready to configure a site, see Configure Site-Specific SAML.

The SAML configuration steps we provide make the following assumptions:

- You are familiar with the options for configuring SAML authentication on Tableau Server, as described in the SAML topic.
- You have verified that your environment meets the SAML Requirements, and obtained the SAML certificate files described in those requirements.

Before you begin

As part of your disaster recovery plan, we recommend keeping a backup of certificate and IdP files in a safe location off of the Tableau Server. The SAML asset files that you upload to Tableau Server will be stored and distributed to other nodes by the Client File Service. However, these files are not stored in a recoverable format. See Tableau Server Client File Service.

Note: If you use the same certificate files for SSL, you could alternatively use the existing certificate location for configuring SAML, and add the IdP metadata file to that directory

> when you download it later in this procedure. For more information, see Using SSL certificate and key files for SAML in the SAML requirements.

If you are running Tableau Server in a cluster, then the SAML certificates, keys, and metadata file will be automatically distributed across the nodes when you enable SAML.

Use the TSM web interface

This procedure requires that you upload the SAML certificates to TSM so that they are properly stored and distributed in the server configuration. The SAML files must available to the browser on the local computer where you are running the TSM web interface in this procedure.

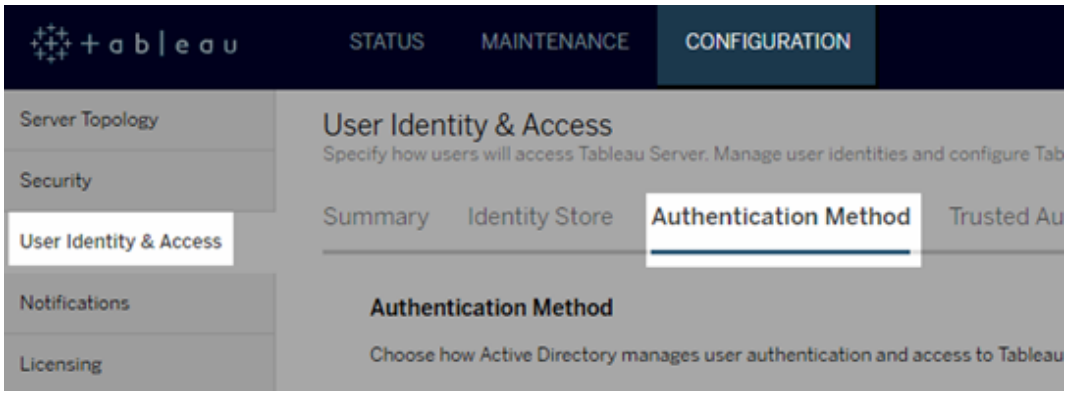
If you have gathered and saved the SAML files to the Tableau Server as recommended in the previous section, then run the TSM web interface from the Tableau Server computer where you copied the files.

If you are running the TSM web interface from a different computer, then you will need to copy all SAML files locally before proceeding. As you follow the procedure below, browse to the files on the local computer to upload them to TSM.

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. On the **Configuration** tab, select **User Identity & Access**, and then select the **Authentication Method** tab.



3. For **Authentication Method**, select **SAML**.

4. In the SAML section that appears, complete Step 1 in the GUI, entering the following settings (do not yet select the check box to enable SAML for the server):

   - **Tableau Server return URL**—The URL that Tableau Server users will access, such as https://tableau-server.

     Using https://localhost or a URL with a trailing slash (for example, http://tableau_server/) is not supported.

   - **SAML entity ID**—The entity ID uniquely identifies your Tableau Server installation to the IdP.

     You can enter your Tableau Server URL again here. If you plan to enable site-specific SAML later, this URL also serves as the base for each site's unique ID.

   - **SAML certificate and key files**— Click **Select File** to upload each of these files.

     If you are using a PKCS#8 passphrase-protected key file, you must enter the passphrase with TSM CLI:

     ```
     tsm configuration set -k wgserver.saml.key.passphrase -v
     <passphrase>
     ```

After you provide the information required in Step 1 in the GUI, the **Download XML Metadata File** button in Step 2 in the GUI becomes available.

5. Now select the **Enable SAML authentication for the server** check box above Step 1 in the GUI.

6. Complete the remaining SAML settings.

   a. For Steps 2 and 3 in the GUI, exchange metadata between Tableau Server and the IdP. (Here's where you might need to check in with the IdP's documentation.)

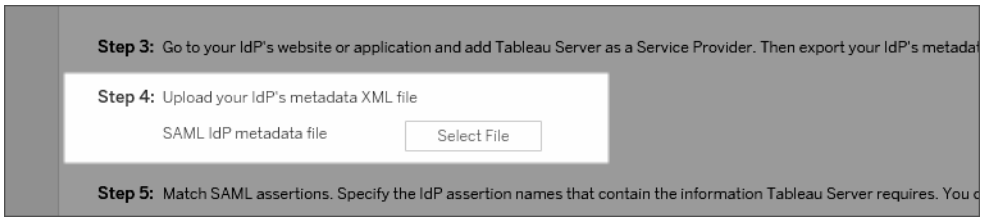
      Select **Download XML Metadata File**, and specify the file location.

      For other IdPs, go to your IdP account to add Tableau Server to its applications (as a service provider), providing the Tableau metadata as appropriate.

      Follow the instructions in the IdP's website or documentation to download the IdP's metadata. Save the .xml file to the same location that holds your SAML certificate and key files. For example:

      ```
      /var/opt/tableau/tableau_server/data/saml/idp-metadata.xml
      ```

   b. Return to the TSM web UI. For Step 4 in the GUI, enter the path to the IdP metadata file, and then click **Select File**.

   c. For Step 5 in the GUI: In some cases, you may need to change the assertion values in the Tableau Server configuration to match the assertion names that are passed by your IdP.

You can find assertion names in the IdP's SAML configuration. If different assertion names are passed from your IdP, then you must update Tableau Server to use the same assertion value.

Tip: “Assertions” are a key SAML component, and the concept of mapping assertions can be tricky at first. It might help to put this in a tabular-data context, in which the assertion (attribute) name is equivalent to a column heading in the table. You enter that “heading” name, rather than an example of a value that might appear in that column.

d. For Step 6 in the GUI, select the Tableau applications in which you want to give users a single sign-on experience.

> **Note**: The option to disable mobile access is ignored by devices running Tableau Mobile app version 19.225.1731 and higher. To disable SAML for devices running these versions you must disable SAML as a client login option on Tableau Server.

e. For the SAML sign-out redirect, if your IdP supports single logout (SLO), enter the page you want to redirect users to after they sign out, relative to the path you entered for the Tableau Server return URL.

f. (Optional) For Step 7 in the GUI, do the following:

- Add a comma-separated value for the `AuthNContextClassRef` attribute. For more information about how this attribute is used, see SAML compatibility notes and requirements.
- Specify a domain attribute if not sending the domain as part of the username (i.e., `domain\username`). For more information, see When running multiple domains.

7. Click **Save Pending Changes** after you've entered your configuration information.

8. Click **Pending Changes** at the top of the page:

9. Click **Apply Changes and Restart**.

Use the TSM CLI

Before you begin

Before you begin, do the following:

- Go to your IdP's website or application, and export the IdP's metadata XML file.

  Confirm that the metadata XML from the IdP includes a **SingleSignOnService** element, in which the binding is set to `HTTP-POST`, as in the following example:

  ```
  <md:SingleSignOnService Bind-
  ing="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Loca-
  tion="https://SERVER-NAME:9031/idp/SSO.saml2"/>
  ```

- Gather the certificate files and place them on Tableau Server.

  In the Tableau Server folder, create a new folder named SAML, and place copies of the SAML certificate files in that folder. For example:

  ```
  /var/opt/tableau/tableau_server/data/saml
  ```

Step 1: Configure return URL, SAML entity ID, and specify certificate and key files

1. Open the command prompt shell and configure the SAML settings for the server (replacing placeholder values with your environment path and file names).

```
tsm authentication saml configure --idp-entity-id https://t-
ableau-server --idp-metadata /var/opt/tableau/tableau_server-
/data/saml/<metadata-file.xml> --idp-return-url
https://tableau-server --cert-file /var/opt/tableau/tableau_
server/data/saml/<file.crt> --key-file /var/-
opt/tableau/tableau_server/data/saml/<file.key>
```

For more information, see `tsm authentication saml configure`.

2. If you are using a PKCS#8 key that is protected with a passphrase, enter the passphrase as follows:

   ```
   tsm configuration set -k wgserver.saml.key.passphrase -v <pass-
   phrase>
   ```

3. If SAML is not already enabled on Tableau Server; for example, you're configuring it for the first time, or you have disabled it, enable it now:

   ```
   tsm authentication saml enable
   ```

4. Apply the changes:

   ```
   tsm pending-changes apply
   ```

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Step 2: Generate Tableau Server metadata and configure the IdP

1. Run the following command to generate the required XML metadata file for Tableau server.

   ```
   tsm authentication saml export-metadata -f <file-name.xml>
   ```

   You can specify a file name, or omit the `-f` parameter to create a default file named `saml-metadata.xml`.

2. On your IdP's website or in its application:

   - Add Tableau Server as a Service Provider.

     Refer to your IdP's documentation for information about how to do this. As part of the process of configuring Tableau Server as a Service Provider, you will import the Tableau Server metadata file you generated from the `export-metadata` command.

   - Confirm that your IdP uses **username** as the attribute to verify users.

Step 3: Match assertions

In some cases, you may need to change the assertion values in the Tableau Server configuration to match the assertion names that are passed by your IdP.

You can find assertion names in the IdP's SAML configuration. If different assertion names are passed from your IdP, then you must update Tableau Server to use the same assertion value.

Tip: "Assertions" are a key SAML component, and the concept of mapping assertions can be tricky at first. It might help to put this in a tabular-data context, in which the assertion (attribute) name is equivalent to a column heading in the table. You enter that "heading" name, rather than an example of a value that might appear in that column.

The following table shows the default assertion values and the configuration key that stores the value.

Assertion	Default value	Key
Username	`username`	`wgserver.saml.idpattribute.username`
Display name	`displayName`	Tableau does not support this attribute type.
Email	`email`	Tableau does not support this attribute type.
Domain	(not mapped by default)	`wgserver.saml.idpattribute.domain`

To change a given value, run the `tsm configuration set` command with the appropriate key:value pair.

For example, to change the `username` assertion to the value, `name`, run the following commands:

```
tsm configuration set -k wgserver.saml.idpattribute.username -v
name

tsm pending-changes apply
```

Alternatively, you can use the `tsm authentication saml map-assertions` command to change a given value.

For example, to set the domain assertion to a value called `domain` and specify its value as "example.myco.com," run the following commands:

```
tsm authentication saml map-assertions --domain example.myco.com

tsm pending-changes apply
```

Optional: Disable client types from using SAML

By default, both Tableau Desktop and the Tableau Mobile app allow SAML authentication.

If your IdP does not support this functionality, you can disable SAML sign-in for Tableau clients using the following commands:

```
tsm authentication saml configure --desktop-access disable
```

```
tsm authentication saml configure --mobile-access disable
```

> **Note**: The `--mobile-access disable` option is ignored by devices running Tableau Mobile app version 19.225.1731 and higher. To disable SAML for devices running these versions you must disable SAML as a client login option on Tableau Server.

```
tsm pending-changes apply
```

Optional: Add AuthNContextClassRef value

Add a comma-separated value for the `AuthNContextClassRef` attribute. For more information about how this attribute is used, see SAML compatibility notes and requirements.

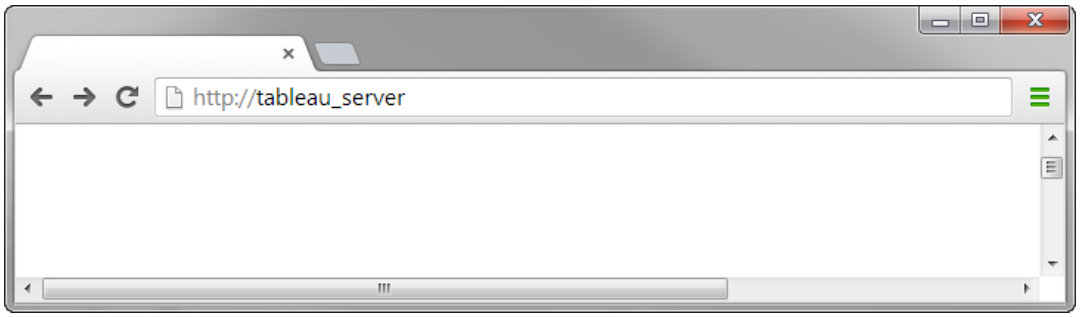
To set this attribute run the following commands:

```
tsm configuration set -k wgserver.saml.authcontexts -v <value>
```

```
tsm pending-changes apply
```

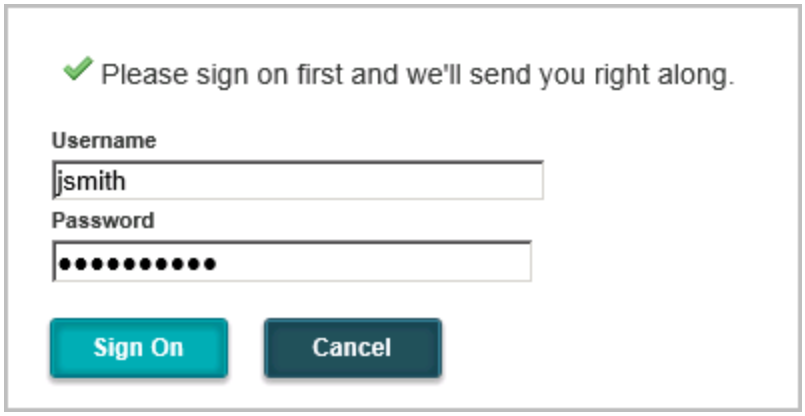
Test the configuration

1. In your web browser, open a new page or tab, and enter the Tableau Server URL.



The browser redirects you to the IdP's sign-in form.

2. Enter your single sign-on user name and password.



The IdP verifies your credentials and redirects you back to your Tableau Server start page.

Configure SAML with Salesforce IdP on Tableau Server

This topic provides road-map steps that describe how to configure SAML on Tableau Server with the Salesforce IdP.

This topic also explains how to enable Lightning Web Component (LWC). The LWC allows Salesforce administrators to embed a Tableau visualization within a Lightning page. When SSO is configured for Tableau Viz LWC on Tableau Server, the user experience is seamless: after the user signs into Salesforce, embedded Tableau views will work without further authentication to Tableau Server.

Enable Salesforce as a SAML Identity Provider

If you have not yet configured Salesforce as an IdP, then follow the procedure, Enable Salesforce as a SAML Identity Provider, on the Salesforce Help site.

During the process to enable Salesforce as a SAML IdP, you will either specify a certificate or Salesforce will generate a self-signed certificate for use with SAML. Download this certificate

(.crt file) and the associated provider metadata file (.xml). You will need these assets in the next step.

Configure SAML on Tableau Server

Using the certificate and metadata files that you downloaded in the previous step, follow the procedure in Configure Server-Wide SAML.

As part of the configuration process, you will generate a SAML Entity ID and a login URL for Tableau Server. You will need these assets for the next step.

(Optional) After you have configured server-wide SAML, you can configure site-specific SAML on Tableau Server. See Configure Site-Specific SAML.

Add Tableau Server as a Connected App in Salesforce

Follow the procedure, Integrate Service Providers as Connected Apps with SAML 2.0, on the Salesforce Help site.

In this process, you will create a new connected app (Tableau Server). Some important details follow:

- Select "Enable SAML"
- Enter the Entity ID and the login URL that you generated when configuring Tableau in the previous section. By default, login URL is `https://<tableauserver>/wg/saml/SSO/index.html`.
- For the IdP certificate, be sure to select the same certificate that you specified or generated when you enabled Salesforce as a SAML provider.
- To allow users to sign in to Tableau Server from your org, manage access to your connected app by assigning the appropriate profiles or permission sets.

Enable Lightning Web Component

To enable LWC for SAML SSO on Tableau Server, you must enable in-frame authentication. Before you enable LWC, upgrade to the latest maintenance release of Tableau Sever. Two additional version details are important:

- If you are not running the latest maintenance release, and your users are running Chrome browsers to access Salesforce Lightning, then review the Tableau KB article,

Embedded Views Fail to Load After Updating to Chrome 80.

- If you are using LWC with site-specific SAML on Tableau Server, then you must be running the Tableau Server 2020.4 or later.

After you have configured SAML on Tableau Server, run the following TSM commands to enable in-frame authentication:

```
tsm configuration set -k wgserver.saml.iframed_idp.enabled -v true

tsm pending-changes apply
```

Embed Tableau Views into Salesforce

After you have configured Tableau Server for SSO, you can then install the LWC in your Salesforce org and embed Tableau views. See Embed Tableau Views into Salesforce.

Configure SAML for Tableau Viz Lightning Web Component

Tableau provides a Lightning Web Component (LWC) for embedding a Tableau visualization within a Salesforce Lightning page.

This topic describes how to enable a SSO experience for embedded Tableau visualizations in a Salesforce Lightning page. SSO for the Tableau Viz LWC scenario requires SAML configuration. The SAML IdP used for Tableau authentication must be either the Salesforce IdP or same IdP that is used for your Salesforce instance.

In this scenario, Salesforce administrators can drag-and-drop Tableau Viz LWC into the Lightning page to embed a visualization. Any view that is available to them on Tableau Server can be displayed in the dashboard by entering the embedded URL to the view.

When single sign-on (SSO) is configured for Tableau Viz LWC on Tableau Server, the user experience is seamless: after the user signs into Salesforce, embedded Tableau views will work without further authentication to Tableau Server.

When SSO is not configured, then users will need to reauthenticate with Tableau Server to view embedded visualizations from Tableau Server.

Requirements

- The SAML IdP used for Tableau authentication must be either the Salesforce IdP or same IdP that is used for your Salesforce instance. See Configure SAML with Salesforce IdP on Tableau Server.
- SAML must be configured on Tableau Server. See Configure Server-Wide SAML, or Configure Site-Specific SAML.
- SAML must be configured for Salesforce.
- Install the Tableau Viz Lightning Web Component. See Embed Tableau Views into Salesforce.

Configuring the authentication workflow

You may need to make additional configurations to optimize the sign-in experience for users who access Lightning with embedded Tableau views.

If a seamless authentication user experience is important, then you will need to make some additional configurations. In this context, “seamless” means that users who access the Salesforce Lightning page where Tableau Viz LWC SSO has been enabled, will not be required to perform any action to view the embedded Tableau view. In the seamless scenario, if the user is logged into Salesforce, then embedded Tableau views will be displayed with no additional user action. This scenario is enabled by *in-frame authentication*.

For a seamless user experience you will need to enable in-frame authentication on Tableau Server and at your IdP. The sections below describe how to configure in-frame authentication.

On the other hand, there are scenarios where users are interacting with the Lightning page that will require them to click a “Sign in” button to view the embedded Tableau view. This scenario, where a user must take another action to view the embedded Tableau view, is called pop-up authentication.

Pop-up authentication is the default user experience if you do not enable in-frame authentication.

Enable in-frame authentication on Tableau Server

Before you enable in-frame authentication on Tableau Server, you must have already configured and enabled SAML on Tableau Server.

Run the following TSM commands to enable in-frame authentication:

```
tsm configuration set -k wgserver.saml.iframed_idp.enabled -v true

tsm pending-changes apply
```

Note: Clickjack protection is enabled by default on Tableau Server. When you enable in-frame authentication, clickjack protection is temporarily disabled for the in-frame authentication session. You should evaluate the risk of disabling clickjack protection. See Clickjack Protection.

Tableau Server Versioning

For the best user experience, run the latest maintenance release of Tableau Server.

If you are not running the latest maintenance release, and your users are running Chrome browsers to access Salesforce Lightning, then review the Tableau KB article, Embedded Views Fail to Load After Updating to Chrome 80.

Enable in-frame authentication with your SAML IdP

As described above, a seamless authentication user experience with Salesforce Mobile requires IdP support for in-frame authentication. This functionality may also be referred to as “iframe embedding” or “framing protection” at IdPs.

Salesforce safelist domains

In some cases, IdPs only allow enabling in-frame authentication by domain. In those cases, set the following Salesforce wildcard domains when you enable in-frame authentication:

```
*.force
```

`*.visualforce`

Salesforce IdP

Salesforce IdP supports in-frame authentication by default. You do not need to enable or configure in-frame authentication in the Salesforce configuration. However, you must run the TSM command on Tableau Server as described above.

Okta IdP

See *Embed Okta in an iframe*, in the Okta Help Center topic, General customization options.

Ping IdP

See the Ping support topic, How to Disable the "X-Frame-Options=SAMEORIGIN" Header in PingFederate.

OneLogin IdP

See *Framing protection*, in the OneLogin Knowledge Base article, Account Settings for Account Owners.

ADFS and EntraID IdP

Microsoft has blocked all in-frame authentication and it cannot be enabled. Instead, Microsoft only supports pop-up authentication in a second window. As a result, pop up behavior can be blocked by some browsers, which will require users to accept pop ups for the `force.com` and `visualforce.com` sites.

Salesforce Mobile App

If your users primarily interact with Lightning on the Salesforce Mobile App, then you should be aware of the following scenarios:

- The Salesforce Mobile App requires that you configure SSO/SAML to view embedded Tableau.

- The Salesforce Mobile App requires in-frame authentication. Pop-up authentication does not work. Instead, users on the Salesforce Mobile App will see the Tableau sign-in button but will not be able to sign to Tableau.
- Mobile App will not work on ADFS and Azure AD IdP.
- The Mobile App uses OAuth tokens to enable SSO. There are scenarios where the OAuth token refreshes and logs users out, requiring users to log back in. To learn more, see the Tableau KB article, Tableau Viz Lightning Web Component On Salesforce Mobile App Prompts for Sign-in.
- The SSO behavior differs according to the version of Salesforce Mobile App (iOS vs Android) and the the IdP:

IdP	Mobile OS	SSO behavior
Salesforce IdP	Android	SSO works initially, but users will need to sign-in after some time.
	iOS	
External IdP	Android	SSO does not work. Users will need to manually sign-in. (SSO must still be configured to enable users access to embedded Tableau views).
	iOS	SSO works initially, but users will need to sign-in after some time.

Configure SAML with Azure AD IdP on Tableau Server

You can configure Azure AD as a SAML identity provider (IdP), and add Tableau Server to your supported single sign-on (SSO) applications. When you integrate Azure AD with SAML and Tableau Server, your users can sign in to Tableau Server using their standard network credentials.

Before you begin: Prerequisites

Before you can configure Tableau Server and SAML with Azure AD, your environment must have the following:

- SSL certificate encrypted using SHA-2 (256 or 512 bit) encryption, and that meets the additional requirements listed in the following sections:
    - SSL certificate requirements
    - SAML Certificate and identity provider (IdP) requirements
- If your users are signing in from a domain that's not the default domain, review SAML Requirements and User Management in Deployments with External Identity Stores to ensure the domain attribute value is set and defined to avoid any sign in issues later on.

Step 1: Verify SSL connection to Azure AD

Azure AD requires an SSL connection. If you haven't done so yet, complete the steps in Configure SSL for External HTTP Traffic to and from Tableau Server, using a certificate that meets the requirements as specified above.

Alternatively, if Tableau Server is configured to work with a reverse proxy or load balancer where SSL is being terminated (commonly referred to as SSL off-loading), then you don't need to configure external SSL.

If your organization uses Azure AD App proxy, see the section below, Azure AD App Proxy.

Step 2: Configure SAML on Tableau Server

Complete the steps in Configure Server-Wide SAML through downloading the Tableau Server metadata to an XML file. At that point, return here and continue to the next section.

Step 3: Configure Azure AD claim rules

The mapping is case sensitive and requires exact spelling, so double-check your entries. The table here shows common attributes and claim mappings. You should verify the attributes with your specific Azure AD configuration.

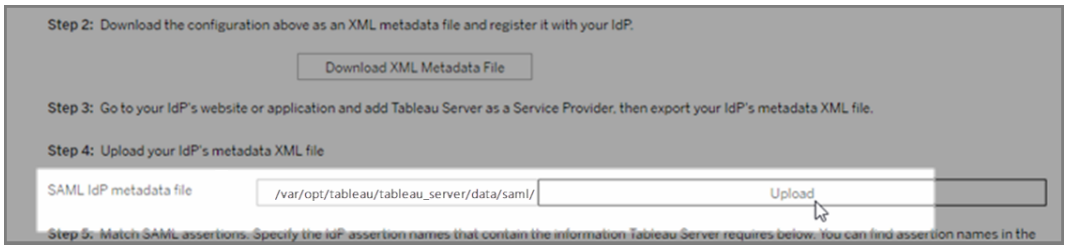
LDAP Attribute	Outgoing Claim Type
onpremisessamaccountname	username

Given-Name	firstName **Note:** This is optional.
Surname	lastName **Note:** This is optional.
netbiosname	domain **Note:** This is only required if you have users signing in from a domain that's not the default domain.

In some organizations, Azure AD as a SAML IdP is used in with Active Directory as the identity store for Tableau Server. In this case, `username` is usually the sAMAccountName name. See Microsoft's documentation for identifying the sAMAccountName attribute within Azure AD to map to the `username` attribute.

Step 4: Provide Azure AD metadata to Tableau Server

1. Return to the TSM web UI, and navigate to **Configuration** > **User Identity & Access** > **Authentication Method** tab.
2. In Step 4 of the SAML configuration pane, enter the location of the XML file you exported from Azure AD, and select **Upload**.



3. Complete the remaining steps (matching assertions and specifying client type access) as specified in Configure Server-Wide SAML. Save and apply changes.
4. Perform the following steps if this isn't the first time configuring SAML:

a. Stop Tableau Server, open TSM CLI, and run the following commands.

```
tsm configuration set -k wgserver.saml.sha256 -v true

tsm authentication saml configure -a -1
```

b. Apply the changes:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Azure AD App Proxy

If you're running Azure AD App Proxy in front of Tableau Server and SAML is enabled, then you'll need to make an additional configuration to Azure AD App Proxy.

Tableau Server can only accept traffic from one URL when SAML is enabled. However, by default, Azure AD App Proxy sets an external URL and an internal URL.

You must set both of these values to the same URL in your custom domain. For more information, see the Microsoft documentation, Configure custom domains with Azure AD Application Proxy.

Troubleshooting

Azure AD App Proxy

In some cases, links to views render internally but fail externally when traffic is crossing an Azure AD App Proxy. The issue arises when there's a pound sign (#) in the URL and users are

accessing the link with a browser. The Tableau Mobile app is able to access URLs with a pound sign.

User session timeouts appear to be ignored

When Tableau Server is configured for SAML, users might experience sign in errors because the IdP maximum authentication age setting is set to a value greater than Tableau's maximum authentication age setting. To resolve this issue, you can use the tsm configuration set option `wgserver.saml.forceauthn` to require the IdP to reauthenticate the user each time Tableau redirects the authentication request, even if the IdP session for the user is still active.

For example, when the Azure AD setting `maxInactiveTime` is greater than Tableau Server's setting `maxAuthenticationAge`, Tableau redirects the authentication request to the IdP who subsequently sends Tableau an assertion that the user is already authenticated. However, because the user was authenticated outside of Tableau Server's `maxAuthenticationAge`, Tableau rejects the user authentication. In cases like this, you can do one or both of the following:

- Enable the `wgserver.saml.forceauthn` option to require the IdP to reauthenticate the user every time Tableau redirects the authentication request. For more information, see wgserver.saml.forceauthn.
- Increase Tableau Server's `maxAuthenticationAge` setting. For more information, see "a, --max-auth-age <max-auth-age>" in the tsm authentication topic.

AppID mismatch

When reviewing the vizportal.log file, you might see "*The intended audience doesn't match the recipient*" error.

To resolve this issue, ensure the appID matches what is sent. Azure will automatically append "SPN" to the appID when using the application ID with the app that is being used. You can change the value in the Tableau SAML settings by adding "SPN:" prefix to the application ID.

For example: SPN:myazureappid1234

Configure SAML with AD FS on Tableau Server

You can configure Active Directory Federation Services (AD FS) as a SAML identity provider, and add Tableau Server to your supported single sign-on applications. When you integrate AD FS with SAML and Tableau Server, your users can sign in to Tableau Server using their standard network credentials.

Prerequisites

Before you can configure Tableau Server and SAML with AD FS, your environment must have the following:

- A server running Microsoft Windows Server 2008 R2 (or later) with AD FS 2.0 (or later) and IIS installed.
- We recommend that you secure your AD FS server (for example, using a reverse proxy). When your AD FS server is accessible from outside your firewall, Tableau Server can redirect users to the sign in page hosted by AD FS.
- SSL certificate encrypted using SHA-2 (256 or 512 bit) encryption, and that meets the additional requirements listed in the following sections:
    - SSL certificate requirements
    - SAML Certificate and identity provider (IdP) requirements

Step 1: Verify SSL connection to AD FS

AD FS requires an SSL connection. If you haven’t done so yet, complete the steps in Configure SSL for External HTTP Traffic to and from Tableau Server, using a certificate that meets the requirements as specified above.

Alternatively, if Tableau Server is configured to work with a reverse proxy or load balancer where SSL is being terminated (commonly referred to as SSL off-loading), then you do not need to configure external SSL.

Step 2: Configure SAML on Tableau Server

Complete the steps in Configure Server-Wide SAML through downloading the Tableau Server metadata to an XML file. At that point, return here and continue to the next section.

Step 3: Configure AD FS to accept sign-in requests from Tableau Server

> **Note:** These steps reflect a third-party application and are subject to change without our knowledge.

Configuring AD FS to accept Tableau Server sign-in requests is a multi-step process, starting with importing the Tableau Server XML metadata file to AD FS.

1. Do one of the following to open the **Add Relying Party Trust Wizard**:

   Windows Server 2008 R2:

   a. Select **Start menu**> **to Administrative Tools**> **AD FS 2.0**.

   b. In **AD FS 2.0**, under **Trust Relationships**, right-click the **Relying Party Trusts** folder, and then click **Add Relying Party Trust**.

   Windows Server 2012 R2:

   a. Open **Server Manager**, and then on the **Tools** menu, click **AD FS Management**.

   b. In **AD FS Management**, on the **Action** menu, click **Add Relying Party Trust**.

2. In the **Add Relying Party Trust Wizard**, click **Start**.

3. On the **Select Data Source** page, select **Import data about the relying party from a file**, and then click **Browse** to locate your Tableau Server XML metadata file. By default, this file is named **samlspmetadata.xml**.

4. Click **Next**, and on the **Specify Display Name** page, type a name and description for the relying party trust in the **Display name** and **Notes** boxes.
5. Click Next to skip the **Configure Multi-factor Authentication Now** page.
6. Click Next to skip the **Choose Issuance Authorization Rules** page.
7. Click Next to skip the **Ready to Add Trust** page.
8. On the **Finish** page, select the **Open the Edit Claim Rules dialog for this relying party trust when the wizard closes** check box, and then click **Close**.

Next, you'll work in the **Edit Claim Rules** dialog, to add a rule that makes sure the assertions sent by AD FS match the assertions Tableau Server expects. At a minimum, Tableau Server needs an email address. However, including first and last names in addition to email will ensure the user names displayed in Tableau Server are the same as those in your AD account.

1. In the **Edit Claim Rules** dialog box, click **Add Rule**.
2. On the **Choose Rule Type** page, for **Claim rule template**, select **Send LDAP Attributes as Claims**, and then click **Next**.
3. On the **Configure Claim Rule** page, for **Claim rule name**, enter a name for the rule that makes sense to you.
4. For **Attribute store**, select **Active Directory**, complete the mapping as shown below, and then click **Finish**.

   The mapping is case sensitive and requires exact spelling, so double-check your entries. The table here shows common attributes and claim mappings. Verify attributes with your specific Active Directory configuration.

LDAP Attribute	Outgoing Claim Type
SAM-Account-Name	Name ID

SAM-Account-Name	username
Given-Name	firstName
Surname	lastName

If you are running AD FS 2016 or later, then you must add a rule to pass through all claim values. If you are running an older version of AD FS, skip to the next procedure to export AD FS metadata.

1. Click **Add Rule**.
2. Under **Claim rule template,** choose **Pass Through or Filter an Incoming Claim.**
3. Under **Claim rule name**, enter Windows.
4. On the **Edit Rule - Windows** pop-up:
    - Under **Incoming claim type**, select **Windows account name**.
    - Select **Pass through all claim values**.
    - Click **OK**.

Now you will export AD FS metadata that you'll import to Tableau Server later. You will also make sure the metadata is configured and encoded properly for Tableau Server, and verify other AD FS requirements for your SAML configuration.

1. Export AD FS Federation metadata to an XML file, and then download the file from **https://<*adfs server name*>/federationmetadata/2007-06/FederationMetadata.xml**.

2. Open the metadata file in a text editor like Sublime Text or Notepad++, and verify that it is correctly encoded as UTF-8 without BOM.

   If the file shows some other encoding type, save it from the text editor with the correct encoding.

3. Verify that AD FS uses forms-based authentication. Sign-ins are performed in a browser window, so you need AD FS to default to this type of authentication.

Edit **c:\inetpub\adfs\ls\web.config**, search for the tag **,** and move the line so it appears first in the list. Save the file so that IIS can automatically reload it.

> **Note:** If you don't see the **c:\inetpub\adfs\ls\web.config** file, IIS is not installed and configured on your AD FS server.

4. (Optional) This step is required only if AD FS is configured as an IDP for site-specific SAML. This step is not required if AD FS is configured as the IDP for server-wide SAML.

   Configure an additional AD FS relying party identifier. This allows your system to work around any AD FS issues with SAML logout.

   Do one of the following:

   Windows Server 2008 R2:

   a. In **AD FS 2.0**, right-click on the relying party you created for Tableau Server earlier, and click **Properties**.

   b. On the **Identifiers** tab, in the **Relying party identifier** box, enter **https://<tableauservername>/public/sp/metadata** and then click **Add**.

   Windows Server 2012 R2:

   a. In **AD FS Management**, in the **Relying Party Trusts** list, right-click on the relying party you created for Tableau Server earlier, and click **Properties**.

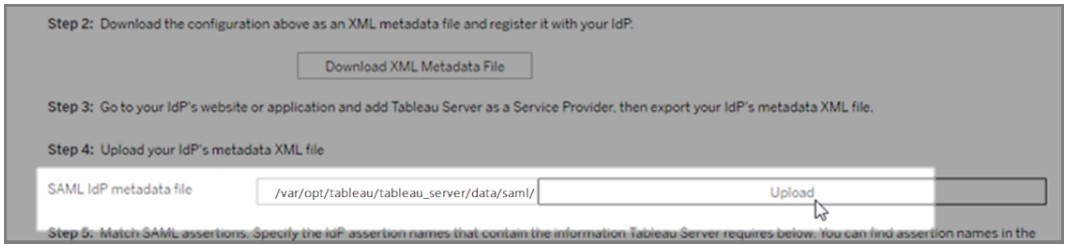
   b. On the **Identifiers** tab, in the **Relying party identifier** box, enter **https://<tableauservername/public/sp/metadata** and then click **Add**.

   > **Note**: AD FS can be used with Tableau Server for a single relying party to the same instance. AD FS cannot be used for multiple relying parties to the same instance,

> for example, multiple site-SAML sites or server-wide and site SAML configurations.

Step 4: Provide AD FS metadata to Tableau Server

1. Return to the TSM web UI, and navigate to **Configuration** > **User Identity & Access** > **Authentication Method** tab.
2. In Step 4 of the SAML configuration window, enter the location of the XML file you exported from AD FS, and select **Upload**.



3. Complete the remaining steps (matching assertions and specifying client type access) as specified in Configure Server-Wide SAML.
4. Save and apply changes.
5. Perform the following steps if this is not the first time configuring SAML:
   a. Stop Tableau Server, open TSM CLI, and run the following commands:

      ```
      tsm configuration set -k wgserver.saml.sha256 -v true
      ```

      ```
      tsm authentication saml configure -a -1
      ```

   b. Apply the changes:

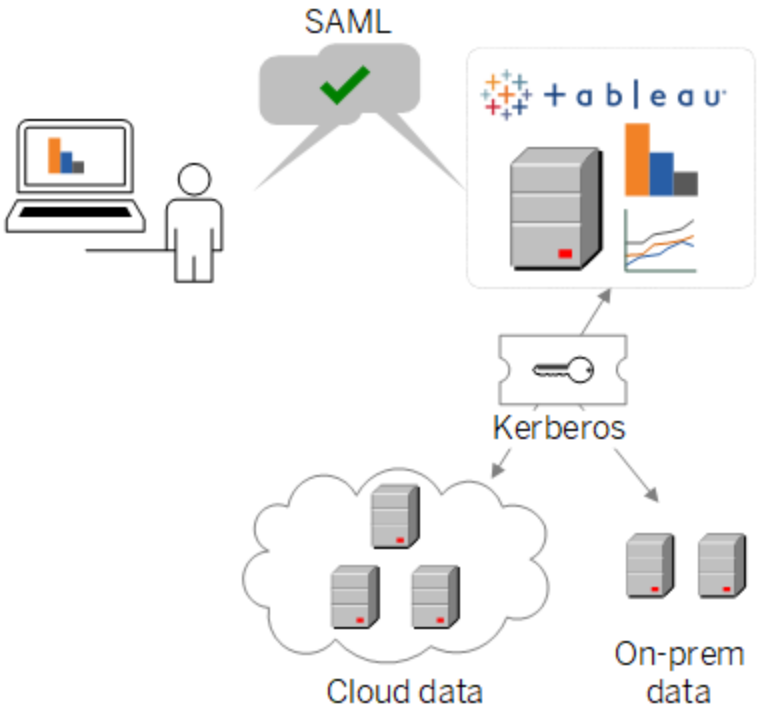
      ```
      tsm pending-changes apply
      ```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Use SAML SSO with Kerberos Database Delegation

In a Windows Active Directory (AD) environment, you can enable SAML single sign-on (SSO) to Tableau Server, along with Kerberos database delegation. This provides authorized users direct access to Tableau Server, as well as to the underlying data defined in their published workbooks and data sources.

Overview of the process



In a typical scenario:

1. One of your Tableau analysts publishes a dashboard to Tableau Server. That dashboard contains a connection to a Hadoop cluster, for example, that is configured to accept Kerberos credentials.

   Then the workbook publisher sends a link to colleagues for review.

2. When a colleague clicks the link, Tableau Server authenticates the user through the SAML SSO process. Then it looks at the user's authorization scheme, and if allowed, uses the Tableau Server keytab to accesses the underlying database on behalf of the user. This populates the dashboard with the Hadoop data that the user is authorized to see.

Configure Tableau Server for SAML with Kerberos

Using SAML with Kerberos works inherently when you complete the processes to enable each separately:

1. Configure Tableau Server for SAML, as described in Configure Server-Wide SAML.

2. Configure Tableau Server and your underlying databases to accept Kerberos credentials, as described in Enable Kerberos Delegation and related articles.

Configure Site-Specific SAML

Use site-specific SAML in a multi-site environment when you want to enable single sign-on, and you also use multiple SAML identity providers (IdPs) or IdP applications. When you enable site SAML, you can specify the IdP or IdP application for each site, or configure some sites to use SAML and the others to use the default server-wide authentication method.

If you want all server users to use SAML and sign in through the same IdP application, see Configure Server-Wide SAML.

Prerequisites for enabling site-specific SAML

Before you can enable SAML single sign-on at the site level, complete the following requirements:

- The Tableau Server identity store must be configured for local identity store.

  You cannot configure site-specific SAML if Tableau Server is configured with an external identity store such as Active Directory or OpenLDAP.

- Make sure your environment and your IdP meet the general SAML Requirements.

  Some features are supported only in server-wide SAML deployments, including but not limited to:

  - Password-protected key files, which are not supported in site-specific SAML deployments.

- You must configure server-wide SAML before you configure site-specific SAML. You do not need to enable server-wide SAML, but site-specific SAML requires the server-wide configuration. See Configure Server-Wide SAML.

- Note the location of the SAML certificate files. You will provide this when you Configure the server to support site-specific SAML.

  For more information, see Put metadata and certificate files in place in the topic on configuring server-wide SAML.

- Add Tableau Server as a service provider to your IdP. You can find this information in the documentation the IdP provides.

- Confirm that the system clocks of the computer hosting the site-SAML IdP and the computer hosting Tableau Server are within 59 seconds of each other. Tableau Serer does not have a configuration option to adjust the response skew (time difference) between the Tableau Server computer and the IdP.

Server-wide settings related to site-specific SAML

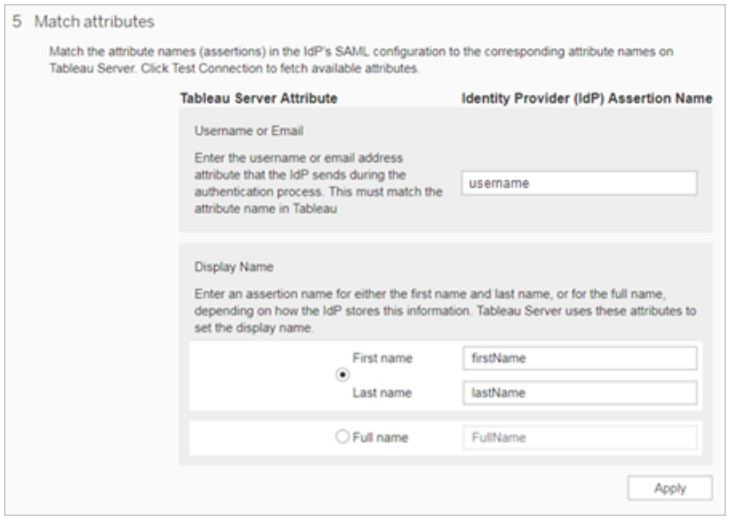
Return URL and entity ID: In the settings for configuring site-specific SAML, Tableau provides a site-specific return URL and entity ID based on these settings. The site-specific return URL and entity ID cannot be modified. These configurations are set by TSM as described in Configure Server-Wide SAML.

Authentication age and response skew: Server-wide settings, maximum authentication age and response skew, do not apply to site-specific SAML. These configurations are hard-coded:

- The maximum authentication age refers to how long an authentication token from the IdP is valid after it is issued. The hard-coded maximum authentication age site-specific SAML is 24 days.
- The response skew is the maximum number of seconds difference between Tableau Server time and the time of the assertion creation (based on the IdP server time) that still allows the message to be processed. The hard-coded site-specific value for this is 59 seconds.

Username: Required. In addition to the server-wide SAML configuration attribute, the site-specific SAML configuration attribute must be set to "username."

> **Note:** For site-specific SAML to operate successfully with a server-wide SAML default, the username attribute configured for server-wide SAML with the wgserver.saml.idpattribute.username configuration key must be "username". The IdP used for server-wide SAML must deliver the username in an attribute named "username."



HTTP POST and HTTP REDIRECT: For site-specific SAML, Tableau Server supports HTTP-POST, HTTP-REDIRECT, and HTTP-POST-SimpleSign.

Configure the server to support site-specific SAML

After you complete the prerequisites listed above, you can run the following commands to configure the server to support site-specific SAML.

1. Configure Server-Wide SAML. At a minimum, you must run the following TSM command (if you have already configured server-wide SAML, skip to Step 2):

   ```
   tsm authentication saml configure --idp-entity-id <tableau-server-entity-id> --idp-return-url <tableau-server-return-url> --cert-file <path-to-saml-certificate.crt> --key-file <path-to-saml-keyfile.key>
   ```

2. Enable site SAML. Run the following commands:

   ```
   tsm authentication sitesaml enable

   tsm pending-changes apply
   ```

About the commands

The `sitesaml enable` command exposes the **Authentication** tab on each site's **Settings** page in the Tableau Server web UI. After you configure the server to support site SAML, you can continue to Configure SAML for a site to work through the settings on the **Authentication** tab.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

If you want to review the commands and settings that will be carried out when you run `pending-changes apply`, you can run the following command first:

```
tsm pending-changes list --config-only
```

Configure SAML for a site

This section takes you through the configuration steps that appear on the **Authentication** tab in the Tableau Server Settings page. In a self-hosted Tableau Server installation, this page appears only when support for site-specific SAML is enabled at the server level.

Note: To complete this process, you will also need the documentation your IdP provides. Look for topics that refer to configuring or defining a service provider for a SAML connection, or adding an application.

Step 1: Export metadata from Tableau

To create the SAML connection between Tableau Server and your IdP, you need to exchange required metadata between the two services. To get metadata from Tableau Server, choose one of the following methods. See the IdP's SAML configuration documentation to confirm the correct option.

- Select **Export metadata** button to download an XML file that contains the Tableau Server SAML entity ID, Assertion Consumer Service (ACS) URL, and X.509 certificate.

  The entity ID is site-specific, and based on the server-wide entity ID that you specified when you enabled site SAML on the server. For example, if you specified `https://tableau_server`, you might see the following entity ID for the site:

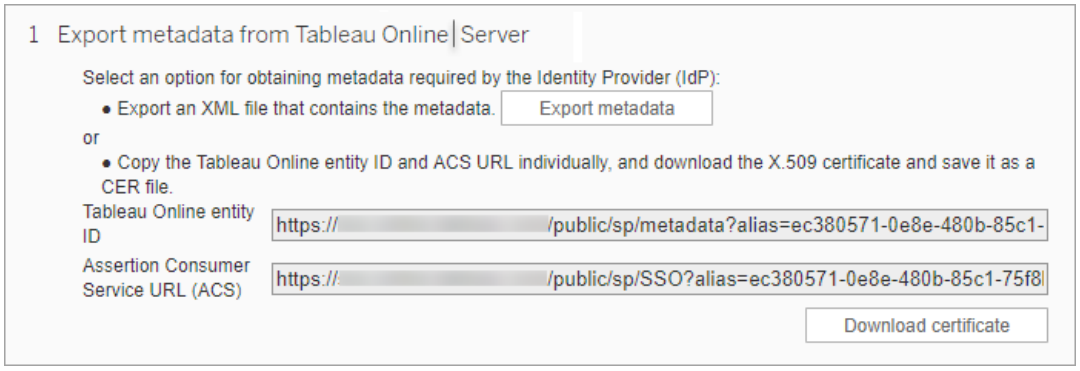
  ```
  https://tableau_server-
  /samlservice/public/sp/metadata?alias=48957410-9396-430a-967c-
  75bdb6e002a0
  ```

  You cannot modify the site-specific entity ID or ACS URL that Tableau generates.

- Select **Download certificate** if your IdP expects the required information in a different way. For example, if it wants you to enter the Tableau Server entity ID, ACS URL, and

X.509 certificate in separate locations.

The following image has been edited to show that these settings are the same in Tableau Cloud and Tableau Server.



Step 2 and Step 3: External steps

For step 2, to import the metadata you exported in step 1, sign in to your IdP account, and use the instructions provided by the IdP's documentation to submit the Tableau Server metadata.

For step 3, the IdP's documentation will guide you also in how to provide metadata to a service provider. It will instruct you to download a metadata file, or it will display XML code. If it displays XML code, copy and paste the code into a new text file, and save the file with a .xml extension.

Step 4: Import IdP metadata to the Tableau site

On the **Authentication** page in Tableau Server, import the metadata file that you downloaded from the IdP or configured manually from XML it provided.

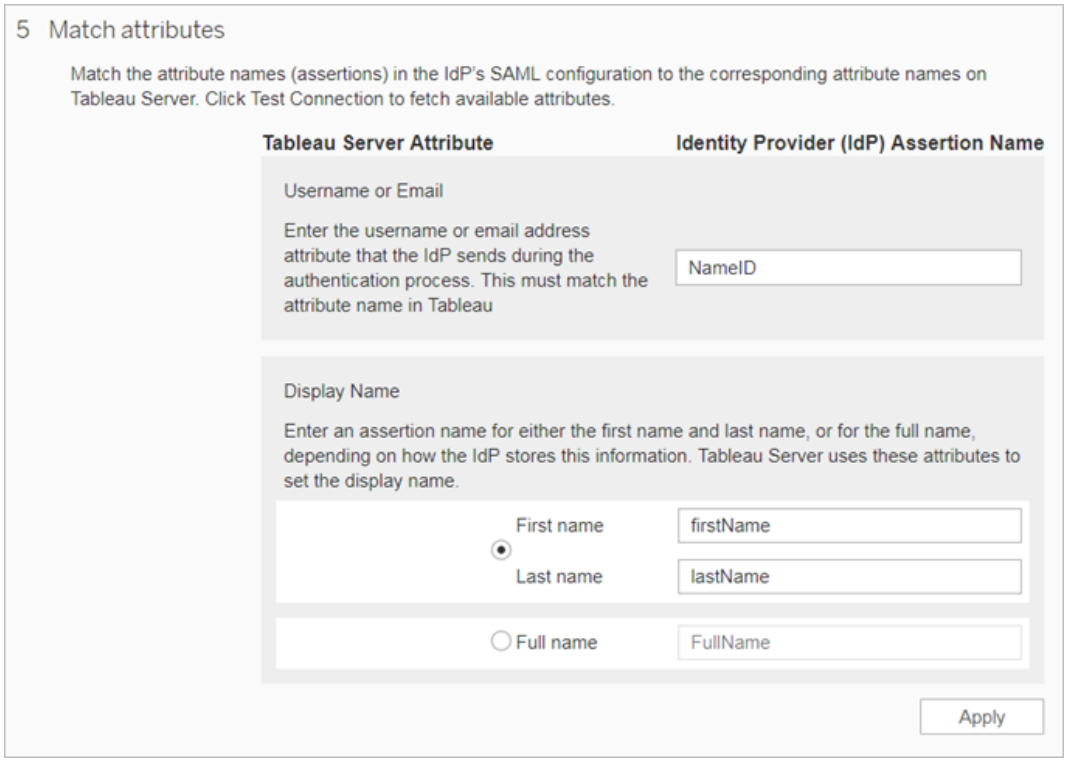
Note: If editing the configuration, you will need to upload the metadata file so Tableau knows to use the correct IdP entity ID and SSO service URL.

Step 5: Match attributes

Attributes contain authentication, authorization, and other information about a user. In the **Identity Provider (IdP) Assertion Name** column, provide the attributes that contain the information Tableau Server requires.

- **Username or Email:** (Required) Enter the name of the attribute that stores user names or email addresses.
- **Display name:** (Optional) Some IdPs use separate attributes for first and last names, and others store the full name in one attribute. If you're using SAML with local authentication, the display name attribute isn't synchronized with the SAML IdP.

  Select the button that corresponds to the way your IdP stores the names. For example, if the IdP combines first and last name in one attribute, select **Display name**, and then enter the attribute name.



Step 6: Manage users

Select existing Tableau Server users, or add new users you want to approve for single sign-on.

When you add or import users, you also specify their authentication type. On the Users page, you can change users' authentication type any time after adding them.

For more information, see Add Users to a Site or Import Users and Set the User Authentication Type for SAML.

Important: Users that authenticate with site-specific SAML can belong only to one site. If a user needs to access multiple sites, set their authentication type to the server default. Depending on how site-specific SAML was configured by the server administrator, the server default is either local authentication or server-wide SAML.

Step 7: Troubleshooting

Start with the troubleshooting steps suggested on the Authentication page. If those steps do not resolve the issue, see Troubleshoot SAML.

Update SAML Certificate

After you have configured SAML authentication, you may need to periodically update the certificate. In some cases, you may need to change the certificate for operational changes in your IT environment. In either case, you must use TSM or the Site Authentication page to update the SAML certificate that has already been configured.

Below are the steps to update the certificate and key files for server-wide and site-specific SAML implementations.

Update certificate for server-wide SAML

To change or update the certificate (and the corresponding key file if required) for server-wide SAML, follow the steps below:

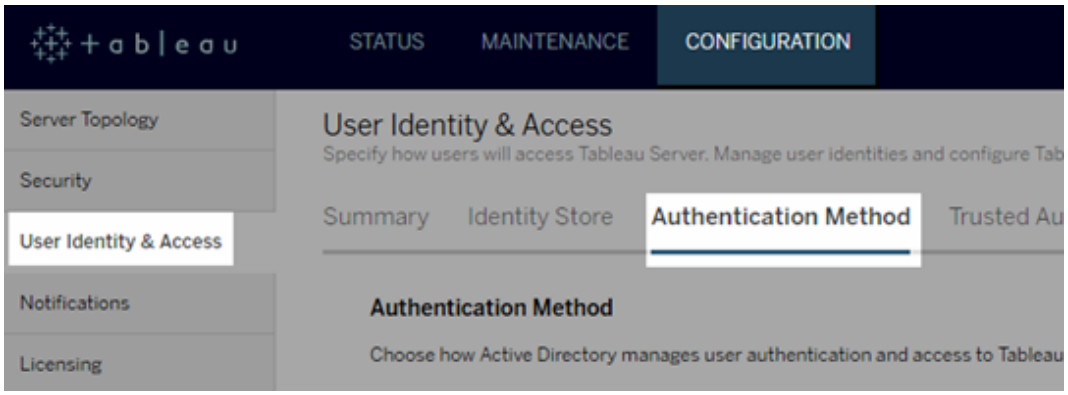
1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Stop Tableau Server.

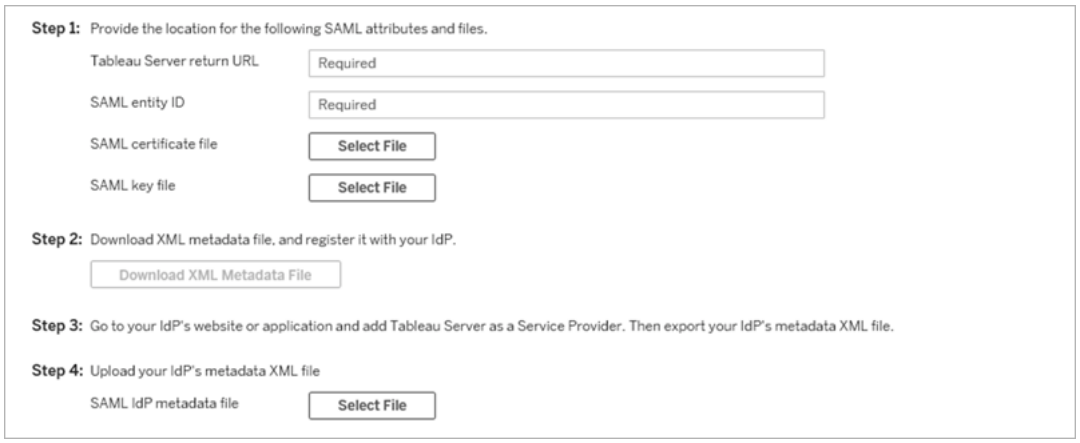
   You can do this either from the TSM web UI, by clicking **Tableau Server is running**, and selecting **Stop Tableau Server**, or from the command line, using the `tsm stop` command.

3. On the **Configuration** tab, select **User Identity & Access**, and then select the **Authentication Method** tab.



4. For **Authentication Method**, select **SAML**.

5. Complete Step 1 - Step 4 in the GUI to update the SAML certificate file and exchange metadata between Tableau Server and your IdP.



6. Click **Save Pending Changes** after you've entered your configuration information.
7. Click **Pending Changes** at the top of the page:

8. Click **Apply Changes and Restart**.

After you change the certificate, you must run `tsm pending-changes apply` to restart Tableau Server services. We also recommend restarting any other services on the computer that use the SAML certificate. If you are changing a root certificate on the operating system, you must reboot the computer.

Update certificate for site-specific SAML

The certificate used for Tableau site metadata is provided by Tableau and not configurable. To update the certificate for site-specific SAML, you must upload a new certificate to your IdP and re-exchange the metadata with Tableau Server.

1. Sign in to the site as a server or site administrator, and select **Settings** > **Authentication**.
2. Under Authentication types, select **Edit connection** to expand the UI.
3. Open a new tab or window, and sign in to your IdP account.

4. Use the instructions provided by the IdP's documentation to upload a new SAML certificate.
5. Download the new XML metadata file to provide to Tableau Server.
6. Return to the **Authentication** page in Tableau Server, and in Step 4 of the UI, import the metadata file that you downloaded from the IdP.
7. Click the **Apply** button.

Troubleshoot SAML

This topic provides information about resolving issues that can occur when you configure SAML authentication.

SAML and Enable Automatic Logon

If you are using SAML and if Tableau Server is also configured to use Active Directory, do not also select **Enable automatic logon**. **Enable automatic logon** and SAML cannot both be used on the same server installation.

HTTP status 500 error when configuring SAML

Under some circumstances you might get an HTTP status 500 error and see the following error after enabling SAML and navigating to the Tableau Server URL in a browser:

```
org.opensaml.saml2.metadata.provider.MetadataProviderException:
User specified binding is not supported by the Identity Provider
using profile urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser
```

To help resolve this error, make sure of the following:

- The IdP URL for the SSO profile specified in the SAML tab is correct.
- The IdP URL for the SSO profile provided while creating the service provider in the IdP is correct.

- The IdP is configured to use `HTTP-POST` requests. (Redirect and SOAP are not supported.)

If any of these settings were not correct, make appropriate updates and then perform the SAML configuration steps again, starting with generating and exporting the XML metadata document from Tableau Server.

If these settings are correct, but you still see the error, examine the metadata XML that is produced by Tableau Server and by the IdP, as described in SAML Requirements.

Signing in from the command line

SAML is not used for authentication when you sign in to Tableau Server using tabcmd or the Tableau Data Extract command line utility (provided with Tableau Desktop), even if Tableau Server is configured to use SAML. These tools require the authentication configured when Tableau Server was originally installed (either local authentication or AD).

Login fails: Failed to find the user

Login fails with the following message:

```
>Login failure: Identity Provider authentication successful for user
<username from IdP>. Failed to find the user in Tableau Server.
```

This error typically means that there is a mismatch between the usernames stored in Tableau Server and provided by the IdP. To fix this, make sure that they match. For example, if Jane Smith's username is stored in the IdP as `jsmith` it must be stored in Tableau Server as `jsmith`.

Login fails: SSL offloading

Logon fails with the following message:

```
Unable to Sign In - Invalid username or password.
```

Additionally, the vizportal logs (set to `debug` mode) contain the following message:

```
DEBUG com.tableau.core.util.RemoteIP - Found header null in X-
FORWARDED-PROTO
```

> **Note:** To log SAML-related events, `vizportal.log.level` must be set to `debug`. For more information, see Change Logging Levels.

This combination of messages indicates a misconfiguration of an external proxy server that is offloading SSL for the connection to Tableau Server. To resolve this issue, see the KB article, "Unable to Sign In" and "Invalid username or password" Error With SAML After Upgrading.

SAML error log

SAML authentication takes place outside Tableau Server, so troubleshooting authentication issues can be difficult. However, login attempts are logged by Tableau Server. You can create a snapshot of log files and use them to troubleshoot problems. For more information, see Log File Snapshots (Archive Logs).

> **Note:** To log SAML-related events, `vizportal.log.level` must be set to `debug`. For more information, see Change Logging Levels.

Check for SAML errors in the following files in the unzipped log file snapshot: `\vizportal\vizportal-<n>.log`

The application process (vizportal.exe) handles authentication, so SAML responses are logged by that process.

Trailing slash

On the SAML tab, confirm that the **Tableau Server return URL** does not end with a trailing slash

Correct: **http://tableau_server**

Incorrect: http://tableau_server/

Confirm connectivity

Confirm that the Tableau Server you are configuring has either a route-able IP address or a NAT at the firewall that allows two-way traffic directly to the server.

You can test your connectivity by running telnet on Tableau Server and attempting to connect with the SAML IdP. For example: `C:\telnet 12.360.325.10 80`

The above test should connect you to the HTTP port (80) on the IdP and you should receive an HTTP header.

Multiple domains

On the SAML tab, confirm that the Tableau Server **Domain** attribute will detect the domain in the `domain\username` format in the SAML assertion by leaving it blank.

Correct: <empty>

Incorrect: yourdomain.com

Kerberos

Kerberos is a three-way authentication protocol that relies on the use of a trusted third-party network service called the Key Distribution Center (KDC) to verify the identity of computers and provide for secure connections between the computers through the exchange of *tickets*. These tickets provide mutual authentication between computers or services, verifying that one has permission to access the other.

Tableau Server supports Kerberos authentication in an Active Directory Kerberos environment, with authentication to Tableau Server being handled by Kerberos.

Notes:

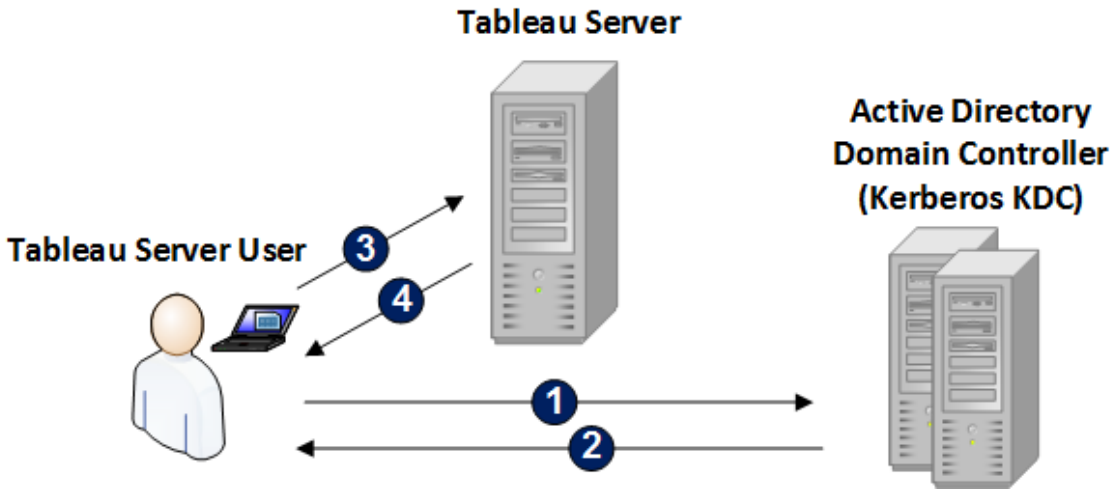
- The Kerberos support in Tableau Server is for user authentication. It does not handle internal permissions and authorization related to Tableau Server content, such as workbooks.
- Identity pools, which is a tool designed to complement and support additional user provisioning and authentication options you might need in your organization, supports

OpenID Connect (OIDC) authentication only. For more information, see Provision and Authenticate Users Using Identity Pools.

How Kerberos works

When you configure Tableau Server for Kerberos in an Active Directory (AD) environment, the AD domain controller also serves as the Kerberos Key Distribution Center (KDC) and issues Ticket Granting Tickets to the other nodes in the domain. Users authenticated by the KDC do not have to authenticate further when connecting to Tableau Server.

The following is a diagram of the authentication workflow.



1. User logs into their Active Directory domain.
2. The Kerberos KDC authenticates the user and sends a Ticket Granting Ticket (TGT) to the user's computer.
3. The user connects to Tableau Server in Tableau Desktop or in a web browser.
4. Tableau Server authenticates the user.

Kerberos Requirements

You can configure Kerberos authentication for Tableau Server running in Active Directory environments.

General requirements

- External Load Balancer/Proxy Server: If you are going to use Tableau Server with Kerberos in an environment that has external load balancers (ELBs) or proxy server, you need to set these up before you configure Kerberos in the Tableau Server Configuration utility. See Configuring Proxies and Load Balancers for Tableau Server.
- iOS Browser Support: An iOS user can use Kerberos authentication with mobile Safari if a Configuration Profile specifying the user's Kerberos identity is installed. See Configuring an iOS Device for Kerberos Support in the Tableau Mobile Help. For more information about browser support for Kerberos SSO, see Tableau Client Support for Kerberos SSO.
- Tableau Server supports constrained delegation for authentication to data sources. In this scenario, the Tableau data access account is specifically granted rights to the target database SPNs. Unconstrained delegation is not supported.
- The supported data sources (SQL Server, MSAS, PostgreSQL, Hive/Impala, and Teradata) must be configured for Kerberos authentication.
- A keytab file that is configured with the service provider name for the Tableau Server for user authentication. For more information, see Understanding Keytab Requirements.
- Beginning in Tableau Server 2021.2.25, 2021.3.24, 2021.4.19, 2022.1.15, 2022.3.7, and 2023.1.3 (or later), ensure keytab files are created with AES-128 or AES-256 ciphers. RC4 and 3DES ciphers are no longer supported. For more information, see "Tableau Server could not authenticate you automatically" in the Tableau Knowledge Base.

Active Directory requirements

You must meet the following requirements to run Tableau Server with Kerberos in an Active Directory environment:

- Tableau Server must use Active Directory (AD) for authentication.
- The domain must be an AD 2003 or later domain for Kerberos connections to Tableau Server.
- Smart Card Support: Smart cards are supported when users sign into their workstations with a smartcard and this results in a Kerberos TGT being granted to the user from Active Directory.
- Single-Sign On (SSO): Users must be granted a Kerberos Ticket Granting Ticket (TGT) from Active Directory when they sign into their computers. This is standard behavior for domain-joined Windows computers and standard for Mac computers that use AD as their network account server. For more information on using Mac computers and Active Directory, see Join your Mac to a network account server in the Apple Knowledge Base.

Kerberos delegation

For Kerberos delegation scenarios the following are required:

- If the domain is AD 2003 or later, single domain Kerberos delegation is supported. The users, Tableau Server, and backend database must be on the same domain.
- If the domain is AD 2008, there is limited cross domain support. Users from other domains can be delegated if the following conditions are met. Tableau Server and the backend database must be on the same domain, and a two way trust is required between the domain where Tableau Server resides and the user's domain.
- If the domain is 2012 or later, full cross-domain delegation is supported. AD 2012 R2 is

preferred because it has a dialog for configuring constrained delegation, while 2012 non-R2 requires manual configuration.

Understanding Keytab Requirements

Kerberos authentication relies on credentials that are stored in specially formatted files called keytab files. You may need to generate keytab files for your Tableau Server deployment. This topic describes the keytab files that Tableau Server uses to access various services in a typical organization. You may need to generate keytabs for Tableau Server to integrate into the following services:

- User authentication (SSO) in Windows Active Directory
- Data source delegation
- Operating system
- Directory service

Important: Beginning in Tableau Server 2021.2.25, 2021.3.24, 2021.4.19, 2022.1.15, 2022.3.7, and 2023.1.3 (or later), ensure keytab files are created with AES-128 or AES-256 ciphers. RC4 and 3DES ciphers are no longer supported. For more information, see "Tableau Server could not authenticate you automatically" in the Tableau Knowledge Base.

If your organization includes IT professionals who handle identity, authentication, and/or security, then you should work with them to create a plan for generating appropriate keytabs for your Tableau Server deployment.

User authentication (SSO) in Windows Active Directory

If you will be using Active Directory as the identity store for Tableau Server, and you want users to authenticate with Kerberos SSO, then you will need to generate a keytab file for Tableau Server.

Tableau is running on...	**Need to manually generate a keytab?**
Windows in Active Directory domain	Yes
Linux in Active Directory domain	Yes

Windows or Linux in non-Active Directory environment	Kerberos SSO is not a supported scenario.

Follow these recommendations (for Windows and Linux versions of Tableau Server):

- Create a service account in your directory for Tableau Server.
- Create a keytab specifically for the Tableau Server service account. Do not reuse the keytab file that the computer account/OS uses to authenticate. You may use the same keytab for Kerberos SSO as you use for the directory authentication in the scenario above.
- You must create service principal names (SPN) in Active Directory for the Tableau Server service.
- Use the batch file in the next section to create the SPNs and the keytab file.
- After you have created the SPNs, upload the keytab file as described in Configure Kerberos.

Batch file: Set SPN and create keytab in Active Directory

You can use a batch file to set the service principal names (SPN) and create a keytab file. These operations are a part of the process to enable Kerberos SSO for Tableau Server (on Windows or Linux) running in Active Directory.

In previous versions of Tableau Server (before 2018.2), the configuration script was generated from the Tableau Server Configuration utility.

To generate a configuration script, copy and paste the following batch file contents into a text file. The batch file creates service principal names (SPN) for Tableau Server and will create a keytab file for the user you specify in the file.

Follow the directions in the file contents. After you have finished customizing the file, save it as a .bat file.

This file must be run in an Active Directory domain by a Domain admin, who will be prompted for the service account password of the account you specify in the file.

The batch file uses the Windows set, setspn, and ktpass commands.

> **Note**: The batch file below is self-documented. However, if you do not have experience with Kerberos and generating keytab files, we recommend that you read the Microsoft blog post, All you need to know about Keytab files, before proceeding. Environmental details in your organization may require additional configuration of the ktpass command. For example, you must determine what to set for the `/crypto`parameter. We recommend specifying a single `/crypto` value that is required by your KDC. See the Microsoft article, ktpass for the full list of supported values for the `/crypto` parameter.

Creating a keytab file for user authentication in Active Directory must be performed on a Windows computer as specified here. Creating this keytab file on a Linux computer is not supported.

SPN and keytab batch file contents

Beginning in Tableau Server 2022.3, 2022.1.8, 2021.4.12, 2021.3.17, 2021.2.18, 2021.1.20, and 2020.4.23

```
@echo off
setlocal EnableDelayedExpansion

REM *******

REM This script generates the Service Principal Names (SPNs) and
keytab files required for
REM Kerberos SSO with Apache.
REM This script executes set, setspn, and ktpass commands included
in any Windows Server
REM Operating System from 2003 on.
REM Before running this script you must enter configuration
```

```
information for the setspn and
REM ktpass commands.
REM Elements that require your configuration information are
enclosed in as such:
REM  ! -- and --!.
REM After you customize this file, save it as a .bat file, and run
on a domain-joined
REM computer.
REM This script must be run by a Domain admin.

REM **********

REM The following set command will prompt the domain admin for cre-
dentials of the
REM Tableau Server service account.
REM This account must be a valid domain user account.
REM If the password contains a literal \" (blackslash - double
quote), all backslashes
REM immediately before the double quote must be
REM duplicated when typed for the password to work, e.g. if pass-
word contains
REM  \" replace with \\", if passwords contains \\" replace with
\\\\"

set /p adpass= "Enter password for the Tableau Server service
account."
set adpass=!adpass:"=\"!

REM **********

REM The following setspn commands create the SPN in the domain.
REM More information on setspn can be found here:
REM http://technet.microsoft.com/en-us/library/cc731241(WS.10).aspx

REM Enter the canonical FQDN and the host names for Tableau Server
followed by the
```

```
REM Tableau Server service account name.
REM Use this syntax: HTTP/hostname domain\service_account_name.
REM The example below shows syntax for a computer named "tableau01"
in the "example.lan"
REM domain, with service account, "tab-serv-account":
REM setspn -s HTTP/tableau01 example\tab-serv-account
REM setspn -s HTTP/tableau01.example.lan example\tab-serv-account
REM DNS and AD are not case sensitive, but the keytab files are.
Verify that host names
REM match letter case as stored in DNS.
REM Use Windows Server's DNS Manager utility to verify host name
case.

REM **********

echo Creating SPNs...
setspn -s HTTP/!--replace with canonical host name and service
account --!
setspn -s HTTP/!--replace with canonical FQDN and service account -
-!

REM **********

REM The following commands create the keytab file in the same dir-
ectory where the
REM bat file is run. More information on ktpass can be found here:
REM https://docs.microsoft.com/en-us/windows-server-
/administration/windows-commands/ktpass
REM Note: keytab files are case-sensitive.
REM The realm following the FQDN should be all uppercase.
REM Syntax is:
REM ktpass /princ HTTP/!--FQDN--!@!--Kerberos_Realm--! /pass !ad-
pass!
REM /pttype KRB5_NTPRINCIPAL /crypto !--cipher--! /out
keytabs\kerberos.keytab
REM Best practice: specify the /crypto value that is required by
```

```
your KDC.
REM Options for /crypto = {DES-CBC-CRC|DES-CBC-MD5|AES256-
SHA1|AES128-SHA1|All}
REM Do not specify /crypto All because it will result in a keytab
that contains ciphers that are not supported
REM and cause errors.
REM When using AES256-SHA1 OR AES128-SHA1, the /mapuser option must
be included
REM in the ktpass command to ensure the keytab file is mapped prop-
erly to the user. For example:
REM ktpass /princ HTTP/!--FQDN--!@!--Kerberos_Realm--! /pass !ad-
pass! /ptype KRB5_NT_PRINCIPAL /mapuser <domain\username> /crypto
AES256-SHA1 /out keytabs\kerberos.keytab
REM The following example shows the ktpass syntax with the
example.lan configuration from above:
REM ktpass /princ HTTP/!--FQDN--!@!--Kerberos_Realm--! /pass !ad-
pass! /ptype KRB5_NT_PRINCIPAL /crypto DES-CBC-CRC /out keyt-
abs\kerberos.keytab

REM **********

echo Creating Keytab files in %CD%\keytabs
mkdir keytabs
ktpass /princ HTTP/!--FQDN--!@!--Kerberos_Realm--! /pass !adpass!
/ptype KRB5_NT_PRINCIPAL /crypto DES-CBC-CRC /out keyt-
abs\kerberos.keytab
```

For earlier versions of Tableau Server

```
@echo off
setlocal EnableDelayedExpansion

REM *******

REM This script generates the Service Principal Names (SPNs) and
keytab files required for
```

```
REM Kerberos SSO with Apache.
REM This script executes set, setspn, and ktpass commands included
in any Windows Server
REM Operating System from 2003 on.
REM Before running this script you must enter configuration inform-
ation for the setspn and
REM ktpass commands.
REM Elements that require your configuration information are
enclosed in as such:
REM  ! -- and --!.
REM After you customize this file, save it as a .bat file, and run
on a domain-joined
REM computer.
REM This script must be run by a Domain admin.

REM **********

REM The following set command will prompt the domain admin for cre-
dentials of the
REM Tableau Server service account.
REM This account must be a valid domain user account.
REM If the password contains a literal \" (blackslash - double
quote), all backslashes
REM immediately before the double quote must be
REM duplicated when typed for the password to work, e.g. if password
contains
REM  \" replace with \\", if passwords contains \\" replace with
\\\\"

set /p adpass= "Enter password for the Tableau Server service
account."
set adpass=!adpass:"=\"!

REM **********

REM The following setspn commands create the SPN in the domain.
```

```
REM More information on setspn can be found here:
REM http://technet.microsoft.com/en-us/library/cc731241(WS.10).aspx

REM Enter the canonical FQDN and the host names for Tableau Server
followed by the
REM Tableau Server service account name.
REM Use this syntax: HTTP/hostname domain\service_account_name.
REM The example below shows syntax for a computer named "tableau01"
in the "example.lan"
REM domain, with service account, "tab-serv-account":
REM setspn -s HTTP/tableau01 example\tab-serv-account
REM setspn -s HTTP/tableau01.example.lan example\tab-serv-account
REM DNS and AD are not case sensitive, but the keytab files are.
Verify that host names
REM match letter case as stored in DNS.
REM Use Windows Server's DNS Manager utility to verify host name
case.

REM **********

echo Creating SPNs...
setspn -s HTTP/!--replace with canonical host name and service
account --!
setspn -s HTTP/!--replace with canonical FQDN and service account -
-!

REM **********

REM The following commands create the keytab file in the same dir-
ectory where the
REM bat file is run. More information on ktpass can be found here:
REM https://docs.microsoft.com/en-us/windows-server-
/administration/windows-commands/ktpass
REM Note: keytab files are case-sensitive.
REM The realm following the FQDN should be all uppercase.
REM Syntax is:
```

```
REM ktpass /princ HTTP/!--FQDN--!@!--Kerberos_Realm--! /pass !ad-
pass!
REM /pttype KRB5_NTPRINCIPAL /crypto !--cipher--! /out
keytabs\kerberos.keytab
REM Best practice: specify the /crypto value that is required by
your KDC.
REM Options for /crypto = {DES-CBC-CRC|DES-CBC-MD5|RC4-HMAC-
NT|AES256-SHA1|AES128-SHA1|All}
REM Specifying /crypto All will result in passwords stored with RC4
cipher, which is
REM no longer considered secure.
REM When using AES256-SHA1 OR AES128-SHA1, the /mapuser option must
be included
REM in the ktpass command to ensure the keytab file is mapped prop-
erly to the user. For example:
REM ktpass /princ HTTP/!--FQDN--!@!--Kerberos_Realm--! /pass !ad-
pass! /ptype KRB5_NT_PRINCIPAL /mapuser <domain\username> /crypto
AES256-SHA1 /out keytabs\kerberos.keytab
REM The following example shows the ktpass syntax with the
example.lan configuration from above:
REM ktpass /princ HTTP/!--FQDN--!@!--Kerberos_Realm--! /pass !ad-
pass! /ptype KRB5_NT_PRINCIPAL /crypto DES-CBC-CRC /out keyt-
abs\kerberos.keytab

REM **********

echo Creating Keytab files in %CD%\keytabs
mkdir keytabs
ktpass /princ HTTP/!--FQDN--!@!--Kerberos_Realm--! /pass !adpass!
/ptype KRB5_NT_PRINCIPAL /crypto DES-CBC-CRC /out keyt-
abs\kerberos.keytab
```

Operating system

If your organization uses Kerberos for authentication, then the computer where Tableau Server is running must be authenticated with the Kerberos realm in which it's running.

Tableau is running on...	Need to manually generate a keytab?
Windows in Active Directory domain	No
Linux in Active Directory domain	Yes
Windows or Linux in non-Active Directory environment	Yes

If you are running Tableau Server on Windows, and the computer is joined to the Active Directory, then you do not need to manage or generate a keytab file for the operating system.

If you are running Tableau Server on Linux in a Kerberos realm (MIT KDC or Active Directory), then you will need to generate a keytab file specifically for the computer operating system. The keytab you create for the computer should be specifically for OS authentication. Do not use the same keytab file for OS authentication that you will be using for the other services described later in this topic.

Directory service

If your organization uses a directory service, such as LDAP or Active Directory, to manage user identity, then Tableau Server requires read-only access to the directory.

Alternatively, you can configure Tableau Server to manage all accounts by installing with a local identity store. In this case, you do not need a keytab.

The following table summarizes keytab requirements:

Tableau is running on...	Directory service	Need to manually generate a keytab?
Windows in AD domain	Active Directory	No
Windows	LDAP (GSSAPI bind)	Yes
Linux	Active Directory or LDAP	Yes

	(GSSAPI bind)	
Windows or Linux	Active Directory or LDAP (Simple bind)	No
Windows or Linux	Local identity store	No keytab required.

If you need to manually generate a keytab for this scenario, then you will use it for GSSAPI bind to the directory. Follow these recommendations:

- Create a service account in your directory for Tableau Server.
- Create a keytab specifically for the Tableau Server service account. Do not reuse the keytab file that the computer account/OS uses to authenticate.
- Upload the keytab file as part of the json configuration of the Tableau Server identity store. See identityStore Entity.

As part of your disaster recovery plan, we recommend keeping a backup of the keytab and conf files in a safe location off of the Tableau Server. The keytab and conf files that you add to Tableau Server will be stored and distributed to other nodes by the Client File Service. However, the files are not stored in a recoverable format. See Tableau Server Client File Service.

Datasource delegation

You can also use Kerberos delegation to access data sources in an Active Directory. In this scenario, users can be authenticated to Tableau Server with any supported authentication mechanism (SAML, local authentication, Kerberos, etc), but can access datasources that are enabled by Kerberos.

Follow these recommendations:

- The computer account for Tableau Server (Windows or Linux) must be in Active Directory domain.

- The keytab file that you use for Kerberos delegation can be the same keytab that you use for Kerberos user authentication (SSO).
- The keytab must be mapped to the service principal for Kerberos delegation in Active Directory.
- You may use the same keytab for multiple data sources.

For more information, see the following configuration topics:

- Tableau Server on Linux: Enable Kerberos Delegation
- Tableau Server on Windows: Enabling Kerberos Delegation

Configure Kerberos

You can configure Tableau Server to use Kerberos. This allows you to provide a single sign-on (SSO) experience across all the applications in your organization. Before you configure Tableau Server for Kerberos make sure your environment meets the Kerberos Requirements.

> **Note:** Kerberos constrained delegation for SSO to Tableau Server is not supported. (Constrained delegation for data sources is supported.) For more information, see Single-Sign On (SSO) in Kerberos Requirements.

To configure Kerberos, you must first enable Kerberos, and then specify a keytab file for user authentication. The keytab file you specify must be configured with the service provider name for the Tableau Server for user authentication. If you are using Kerberos authentication for data sources, those credentials should be included in the single keytab file that you will specify during Kerberos configuration on Tableau Server.

As part of your disaster recovery plan, we recommend keeping a backup of the keytab file in a safe location off of the Tableau Server. The keytab file that you add to Tableau Server will be stored and distributed to other nodes by the Client File Service. However, the file is not stored in a recoverable format. See Tableau Server Client File Service.

Use the TSM web interface

1. Open TSM in a browser:

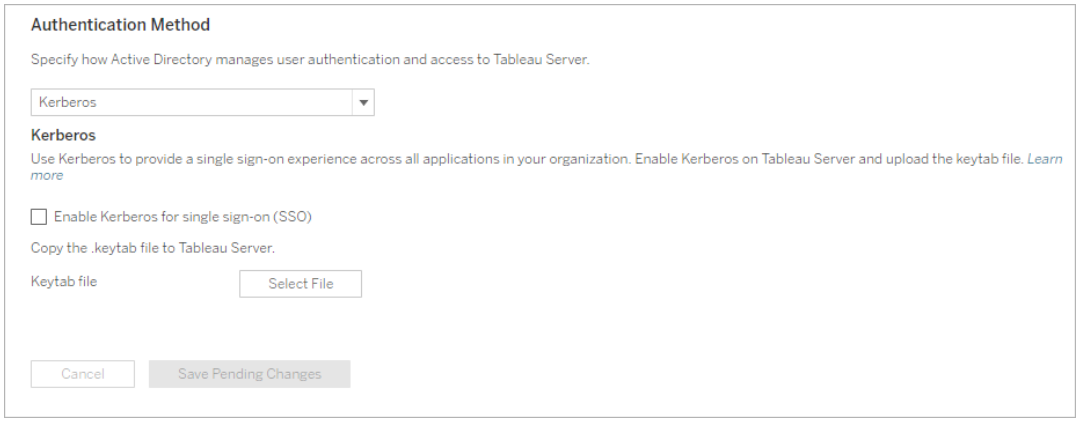
   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click User **Identity & Access** on the **Configuration** tab and then click **Authentication Method**.

3. Under **Authentication Method**, select **Kerberos** in the drop-down menu.

4. Under Kerberos, select **Enable Kerberos for single sign-on (SSO)**.

5. To copy the keytab file to the server, click **Select File**, and then browse to the file on your computer.



6. Click **Save Pending Changes** after you've entered your configuration information.

7. Click **Pending Changes** at the top of the page:

8. Click **Apply Changes and Restart**.

Use the TSM CLI

1. Copy the keytab file to the computer running Tableau Server and run the following command to set permissions on the file:

   ```
   chmod 644 "/path/keytab_file"
   ```

   If you are running Tableau Server on in a distributed cluster deployment, then you will need to manually distribute the keytab file to each node and then set the permissions. Copy the keytab file to the same directory on each node in the cluster. After you have copied the keytab file to each node and set permissions on the file, then run the following TSM commands on one node. The configuration will propagate to each node.

2. Type the following command to specify the location and name of the keytab file:

   ```
   tsm authentication kerberos configure --keytab-file <path-to-keytab_file>
   ```

3. Type the following command to enable Kerberos:

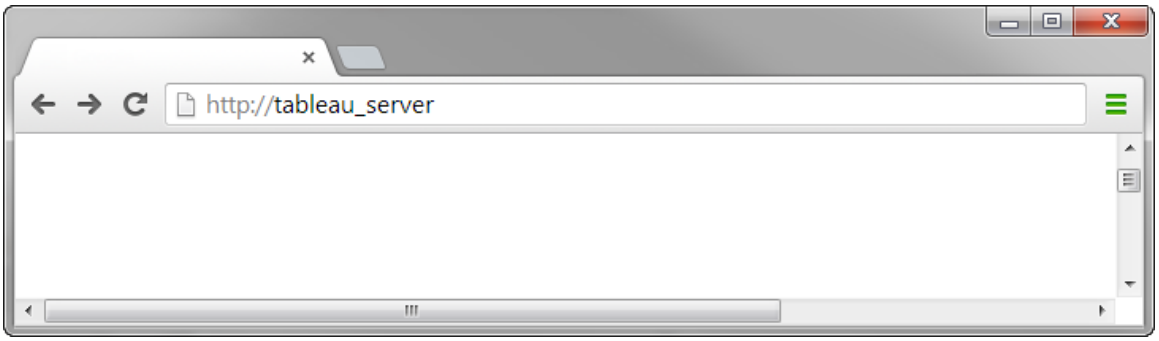
   ```
   tsm authentication kerberos enable
   ```

4. Run `tsm pending-changes apply` to apply changes.

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Confirm your SSO configuration

Once Tableau Server has restarted, test your Kerberos configuration from a web browser on a different computer by typing the Tableau Server name in the URL window:



You should be automatically authenticated to Tableau Server.

Tableau Client Support for Kerberos SSO

This article describes some requirements for and nuances with using Kerberos single sign-on (SSO) with Tableau Server, depending on the particular Tableau client and operating system. Tableau clients covered in this article include common web browsers, Tableau Desktop, and the Tableau Mobile app.

General browser client support

To use browser-based Kerberos Single Sign-on (SSO), the following must be true:

- Kerberos must be enabled on Tableau Server.
- The user must have a user name and password to sign in to Tableau Server.

  Note: When Kerberos SSO fails, users can fall back on their user name and password credentials, if a fall back is set up.

- The user must be authenticated to Active Directory through Kerberos on the client computer or mobile device. Specifically, this means that they have a Kerberos Ticket Granting Ticket (TGT).

Tableau Desktop and browser clients

On Windows or the Mac, you can use Kerberos SSO to sign in to Tableau Server from the following versions of Tableau Desktop or browser. Where noted, additional configuration is required.

Windows

- Tableau Desktop 10.3 or later supported.
- Internet Explorer - supported, may require configuration - see Note 1
- Chrome - supported, may require configuration -see Note 1
- Firefox - requires configuration - see Note 2
- Safari - not supported

Mac OS X

- Tableau Desktop 10.3 or newer
- Safari - supported
- Chrome - see Note 3
- Firefox - see Note 2
- Internet Explorer - not supported

Tableau Mobile app clients

On a iOS or Andoid device, you can use the following Tableau Mobile or mobile browser versions to use Kerberos authentication to Tableau Server:

iOS

- Tableau Mobile app- see Note 4
- Safari - see Note 4
- Chrome - not supported

Android - see Note 5

- Tableau Mobile app
- Chrome

Operating system and browser-specific notes

The following notes describe configuration requirements or issues with specific operating system and client combinations.

Note 1: Internet Explorer or Chrome on Windows desktop

Kerberos SSO is supported in both Internet Explorer and Chrome, but it requires configuration in **Windows Internet Options**:

1. Enable **Integrated Windows Authentication**.
2. Verify that Tableau Server URL is in the local intranet zone.

   Internet Explorer can sometimes detect intranet zones and configure this setting. If it has not detected and configured the Tableau Server URL, you must manually add the URL to the local intranet zone.

To enable Integrated Windows Authentication:

1. In Windows Control Panel, open **Internet Options**.
2. On the **Advanced** tab scroll down to the **Security** section.
3. Select **Enable Integrated Windows Authentication**.
4. Click **Apply**.

To verify or add the Tableau Server URL to the local intranet zone:

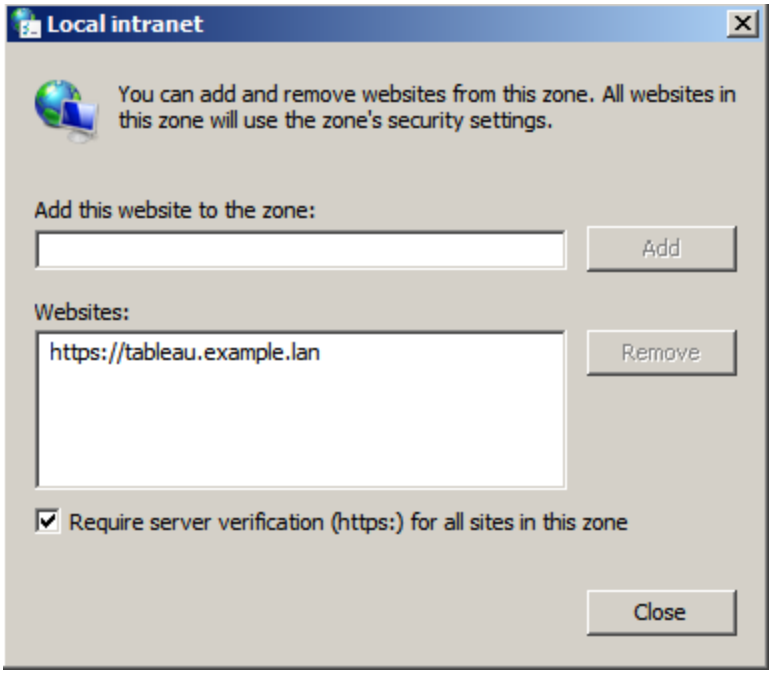
1. In Windows Control Panel, open **Internet Options**.
2. On the **Security** tab, select **Local intranet**, and then click **Sites**.
3. On the **Local intranet** dialog box, click **Advanced**.

   In the **Websites** field, look for the internal Tableau Server URL.

   In some organizations, IT administrators will use a wildcard (*) to specify internal URLs. For example, the following URL includes all servers in the internal `example.lan` namespace in the local intranet zone:

   ```
   https://*.example.lan
   ```

The following image shows a specific URL of `https://tableau.example.lan`.



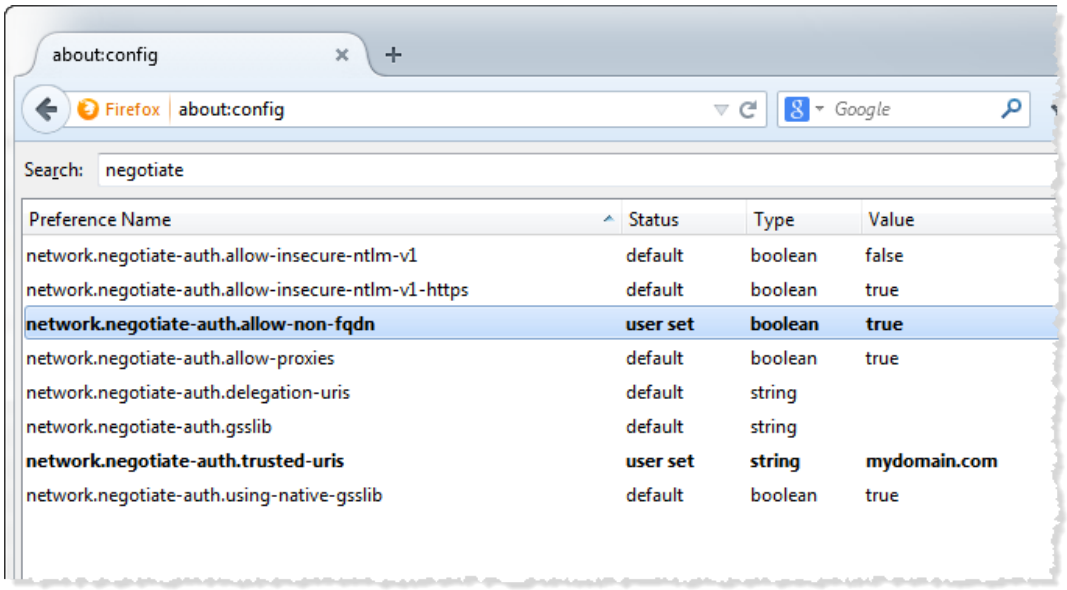
4. If the Tableau Server URL or a wildcard URL is not specified in the **Websites** field, enter the Tableau Server URL in the **Add the website to the zone** field, click **Add**, and then click **OK**.

   If the Tableau Server URL is already listed in **Websites**, you can simply close the dialog.

Note 2: Firefox on Windows or Mac OS X desktop

You can use Firefox with Kerberos SSO on either Windows or Mac to sign in to Tableau Server. To do this, you must complete the following steps to configure Firefox to support Kerberos:

1. In Firefox, enter `about:config` in the address bar.
2. Click **I'll be careful, I promise** when warned about changing advanced settings.
3. Enter `negotiate` in the **Search** box.



4. Double-click **network.negotiate-auth.allow-non-fqdn**, and then set the value to **true**.
5. Double-click **network.negotiate-auth.trusted-uris** and enter the Tableau Server fully qualified domain name (FQDN). For example, `tableau.example.com`.

Note 3: Chrome on Mac OS X desktop

According to Chrome documentation, Kerberos SSO works on a Mac when you launch Chrome from a terminal window with the following command:

```
open -a "Google Chrome.app" --args --auth-server-whitel-
ist="tableauserver.example.com"
```

where `tableauserver.example.com` is the URL for Tableau Server in your environment.

However, we have found inconsistent results in our testing. Therefore, if you want to use Kerberos SSO on a Mac, we recommend that you use Safari or Firefox. For more information, see the *Integrated Authentication* section at HTTP authentication on The Chromium Projects site.

Note: Users can still use Chrome on Mac OS X to sign in to Tableau Server, but they might be prompted to enter their user name and password (single sign-on may not work).

Note 4: Mobile Safari or Tableau Mobile on iOS

Kerberos SSO is supported if iOS is configured for Kerberos. The iOS device must have a Kerberos authentication configuration profile installed. This is usually done by an enterprise IT group. Tableau Support cannot assist with configuring iOS devices for Kerberos. See the authentication topic in the *Tableau Mobile Deployment Guide*.

Note 5: Android platform

Kerberos SSO is not supported on the Tableau Mobile app on the Android operating system. You can still use your Android device and the Tableau Mobile app or a supported mobile browser to connect to Tableau Server if Kerberos has a fall back set up for when SSO fails to accept user name and password authentication. In this scenario, rather than authenticating with Kerberos, users will be prompted to enter their credentials when accessing Tableau Server.

More information

- *Tableau Mobile Deployment Guide*: Control authentication and access for Tableau Mobile
- See *Web Browsers* under Tableau Server Tech Specs

Troubleshoot Kerberos

The troubleshooting suggestions in this topic are divided into issues related to single sign-on (SSO) on the server and issues with the delegated data sources.

Also see the Tableau Community wiki page, Testing Database Kerberos Configuration On Linux.

Single sign-on to Tableau Server

In a Kerberos SSO environment, a user signing in to Tableau Server from a web browser or Tableau Desktop might see a message indicating that Tableau Server can't sign them in automatically (using single sign-on). It suggests that they provide a Tableau Server user name and password instead.

Troubleshooting sign-in errors on the client computer

- **Enter the user name and password**—To check the user's general access to Tableau Server, sign in by entering the user's name and password.

  If these credentials fail, the user might not be a user on Tableau Server. For Kerberos SSO to work, the user must be able to access Tableau Server, and they must be granted a Ticket Granting Ticket (TGT) by Active Directory (as described in the **TGT** item later in this list).

- **Check other users' SSO credentials**—Try to connect with SSO to Tableau Server using other user accounts. If all users are affected, the problem might be in the Kerberos

configuration.

- **Use a computer other than the server computer**—Kerberos SSO does not work when you sign in to Tableau Server on localhost. Clients must connect from a computer other than the Tableau Server computer.
- **Use a server name, not IP address**—Kerberos SSO does not work if you enter an IP address as the Tableau Server name. In addition, the server name you use to access Tableau Server must match the name used in the Kerberos configuration (see Key table entry, below).
- **Confirm that the client has TGT**—The client computer must have a TGT (Ticket Granting Ticket) from the Active Directory domain. Constrained delegation, with the proxy granting a ticket, is not supported.

  To confirm the client computer has a TGT, do the following:

  - On Windows, open a command prompt and type the following: `klist tgt`
  - On the Mac, open a terminal window and type the following: `klist`

  The output should show a TGT for the user/domain trying to authenticate to Tableau Server.

  The client computer might not have a TGT in the following circumstances:

  - The client computer is using a VPN connection.
  - The client computer is not joined to the domain (for example, it is a non-work computer being used at work).
  - The user signed in to the computer with a local (non-domain) account.
  - The computer is a Mac that is not using Active Directory as a network account server.

- **Confirm browser version and settings**—For web browser sign-in, make sure the browser is supported for Kerberos and, if necessary, is configured correctly.

    - Internet Explorer (IE) and Chrome work “out of the box” on Windows.
    - Safari works “out of the box” on the Mac.
    - Firefox requires additional configuration.

    For more information, see Tableau Client Support for Kerberos SSO.

Troubleshooting sign-in errors on the server

If you cannot solve the problem from the client computer, your next steps are to troubleshoot on the computer running Tableau Server. The administrator can use the request ID to locate the sign-in attempt in the Apache logs on Tableau Server.

- **Log files**—Check the Apache error.log for an error with the exact time/date of the failed sign-in attempt.
- **Key table entry**—If the error.log entry includes the message, “No key table entry matching HTTP/<servername>.<domain>.<org>@”, for example:

    ```
    [Fri Oct 24 10:58:46.087683 2014] [:error] [pid 2104:tid 4776]
    [client 10.10.1.62:56789] gss_acquire_cred() failed: Unspe-
    cified GSS failure. Minor code may provide more information (,
    No key table entry found matching HTTP/servername.domain.com@)
    ```

    This error is a result of a mismatch of any of the following:

    - **Tableau Server URL** - The URL used by the client computer to access the server.

        This is the name that you type into Tableau Desktop or a browser address bar. It could be a shortname (`http://servername`) or a fully-qualified domain name (`http://servername.domain.com`)

- **DNS reverse lookup** for the server IP address.

  This looks up a DNS name using an IP address.

  At a command prompt type:

  ```
  ping servername
  ```

  with the IP address returned by pinging the server, do a reverse DNS lookup type:

  ```
  nslookup <ip address>
  ```

  The nslookup command will return network information for the IP address. In the *Non-authoritative answer* portion of the response, verify that the fully qualified domain name (FQDN) matches the following configured values:

  - The Kerberos .keytab file
  - Service Principal Name (SPN) for the server

  For more information about configuring these values, see Understanding Keytab Requirements.

Verify Kerberos configuration script

You may need to modify the ktpass command that you used to generate the keytab file for environmental variables. Review the troubleshooting steps in the Knowledge Base article, Unable to Generate Kerberos Script Configuration for Tableau Server.

Data source SSO

Delegated data source access failures

Check the vizqlserver log files for "workgroup-auth-mode."

Look for "workgroup-auth-mode" in the log files. It should say "kerberos-impersonate" not "as-is".

Kerberos delegation multi-domain configuration

Tableau Server can delegate users from other Active Directory domains. If your database uses MIT Kerberos, you may need to adjust your Kerberos principal to database user mapping. Specifically, you will need to update krb5.conf with rules for each Kerberos realm that users will connect from. Use the `auth_to_local` tag in the `[realms]` section to map principal names to local user names.

For example, consider a user, `EXAMPLE\jsmith`, whose Kerberos Principal is `jsmith@EXAMPLE.LAN`. In this case, Tableau Server will specify a delegated user, `jsmith@EXAMPLE`. Tableau Server will use the Active Directory legacy domain alias as the Kerberos Realm.

The target database may already have a rule such as the following to map the user, `jsmith@EXAMPLE.LAN` to the database user, `jsmith`.

```
EXAMPLE.LAN = {
  RULE:[1:$1@$0](.*@EXAMPLE.LAN)s/@.*//
  DEFAULT
}
```

To support delegation, you must add another rule to map jsmith@EXAMPLE to a database user:

```
EXAMPLE.LAN = {
   RULE:[1:$1@$0](.*@EXAMPLE.LAN)s/@.*//
   RULE:[1:$1@$0](.*@EXAMPLE)s/@.*//
   DEFAULT
}
```

See the MIT Kerberos Documentation topic, krb5.conf, for more information.

Cross-domain constrained delegation

In some cross-domain scenarios where the KDC is running on a Windows Server prior to Windows 2012, delegation may fail. Errors you may see include:

- SQL Server Network Interfaces: The system cannot contact a domain controller to service the authentication request. Please try again later.
- SQL Server Native Client: Cannot generate SSPI context.
- The Domain Controller returns: `KRB-ERR-POLICY error with a status STATUS_CROSSREALM_DELEGATION_FAILURE (0xc000040b).`

Cross-domain refers to a scenario where Tableau Server is running in a different domain than the data source with different service accounts. For example:

- Tableau Server runs on DomainA with DomainA service account.
- SQL Server runs on DomainB with DomainB service account.

Traditional constrained delegation only works if both servers are in the same domain. The user can come from other domains.

If you are seeing the errors noted above, then to enable this scenario, your Active Directory administrator should remove any traditional constrained delegation which is configured on the delegating account. Removing delegation can be achieved with Active Directory management tools or by removing the values associated with the Active Directory property, `msDS-AllowedToDelegateTo`.

If you want to preserve an existing single domain delegation alongside cross-domain delegation, you must configure both using resource-based constrained delegation.

For more information about Kerberos and constrained delegation, see the Microsoft topic, Kerberos Constrained Delegation Overview.

Web authoring

There are two web authoring scenarios that do not support Kerberos delegation: "Connect to data on the web" and "Create datasource on the web." feature does not support delegation yet. Specifically, if you create a datasource that uses Kerberos on the with web authoring, the data source will use Run As service account authentication. If you want to use Kerberos delegation to create a datasource, then you must publish withTableau Desktop. For more information on Run As service account, see Enable Kerberos Service Account Access.

Configure Mutual SSL Authentication

Using mutual SSL, you can provide users of Tableau Desktop, Tableau Mobile, and other approved Tableau clients a secure, direct-access experience to Tableau Server. With mutual SSL, when a client with a valid SSL certificate connects to Tableau Server, Tableau Server confirms the existence of the client certificate and authenticates the user, based on the user name in the client certificate. If the client does not have a valid SSL certificate, Tableau Server can refuse the connection.

You can also configure Tableau Server to fall back to username/password authentication if mutual SSL fails. Additionally, a user can log in using the REST API with a username and password (if one exists) whether or not fallback authentication is configured.

User authentication session time limits

When users log in with mutual SSL, the authentication session is governed by the same method that governs the Tableau Server global authentication session configuration.

For clients that connect to Tableau Server using a web browser, configuration of the global authentication session is described in the *Security Hardening Checklist*, see 9. Verify session lifetime configuration.

Sessions for connected clients (Tableau Desktop, Tableau Mobile, Tableau Prep Builder, and Bridge) use OAuth tokens to keep users logged in by re-establishing a session. By default, OAuth client tokens reset after a year. If a client token has not been used in 14 days, then it will expire. You can change these values by setting the `refresh_token.absolute_expiry_in_seconds` and `refresh_token.idle_expiry_in_seconds` options. See tsm configuration set Options.

Certificate usage

Before you enable and configure mutual SSL, you must configure external SSL. External SSL authenticates Tableau Server to the client and encrypts the session using the certificate and key that is required when you configure external SSL.

For mutual SSL, an additional certificate file is required. The file is a concatenation of CA certificate files. The file type must be `.crt`. A "CA" is a *certificate authority* that issues certificates to the client computers that will connect to Tableau Server. The action of uploading the CA certificate file establishes a trust, which enables Tableau Server to authenticate the individual certificates that are presented by the client computers.

As part of your disaster recovery plan, we recommend keeping a backup of the certificate and revocation (if applicable) files in a safe location off of the Tableau Server. The certificate and revocation files that you add to Tableau Server will be stored and distributed to other nodes by the Client File Service. However, the files are not stored in a recoverable format. See Tableau Server Client File Service.

RSA key and ECDSA curve sizes

The CA certificate used for mutual SSL must either have an RSA key strength of 2048, or ECDSA curve size of 256.

.You can configure Tableau Server to accept the less-secure sizes by setting the respective configuration keys:

- ssl.client_certificate_login.min_allowed.rsa_key_size
- ssl.client_certificate_login.min_allowed.elliptic_curve_size

See tsm configuration set Options.

Client certificate requirements

Users authenticating to Tableau Server with mutual SSL must present a client certificate that meets minimum security requirements.

Signing algorithm

Client certificates must use a SHA-256 or greater signing algorithm.

Tableau Server configured for mutual SSL authentication will block authentication of users with client certificates that use the SHA-1 signing algorithm.

Users who attempt to log in with SHA-1 client certificates encounter an "Unable to sign in" error, and the following error will be visible in the VizPortal logs:

```
Unsupported client certificate signature detected: [certificate Signature Algorithm name]
```

You can configure Tableau Server to accept the less secure SHA-1 signing algorithm by setting the ssl.client_certificate_login.blocklisted_signature_algorithms tsm configuration option.

RSA key and ECDSA curve sizes

The client certificate used for mutual SSL must either have an RSA key strength of 2048, or ECDSA curve size of 256.

Tableau Server will fail mutual authentication requests from client certificates that do not meet these requirements. You can configure Tableau Server to accept the less-secure sizes by setting the respective configuration keys:

- ssl.client_certificate_login.min_allowed.rsa_key_size
- ssl.client_certificate_login.min_allowed.elliptic_curve_size

See tsm configuration set Options.

Use the TSM web interface

1. Configure SSL for External HTTP Traffic to and from Tableau Server.
2. Open TSM in a browser:

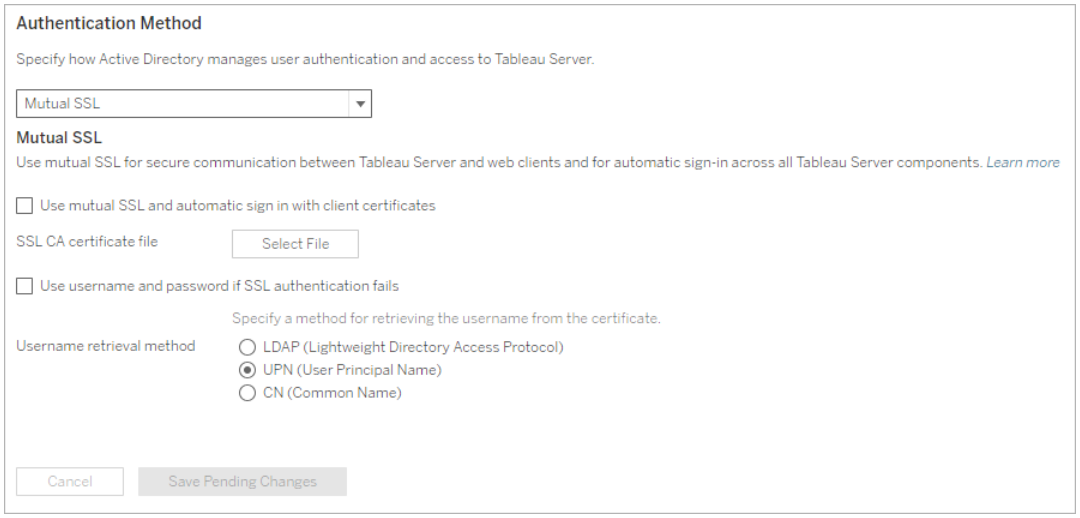
   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.
3. On the **Configuration** tab, select **User Identity & Access** > **Authentication Method**.
4. Under **Authentication Method**, select **Mutual SSL** in the drop-down menu.
5. Under Mutual SSL, select **Use mutual SSL and automatic sign in with client certificates**.
6. Click **Select File** and upload your certificate authority (CA) certificate file to the server.

   The file (.crt) is an all-in-one file that includes certificates of CAs that are used for client authentication. The file you upload must be a concatenation of the various PEM-encoded certificate files, in order of preference.
7. Enter remaining SSL configuration information for your organization.

   Username format: When Tableau Server is configured for mutual SSL, the server gets the user name from the client certificate, so it can establish a direct sign-in for the client user. The name that Tableau Server uses depends on how Tableau Server is configured for user authentication:

   - Local Authentication—Tableau Server uses the UPN (User Principal Name) from the certificate.
   - Active Directory (AD)—Tableau Server uses LDAP (Lightweight Directory Access Protocol) to get the user name.

Alternatively, you can set Tableau Server to use the CN (Common Name) from the client certificate.



8. Click **Save Pending Changes** after you've entered your configuration information.

9. Click **Pending Changes** at the top of the page:

10. Click **Apply Changes and Restart**.

Use the TSM CLI

Step 1: Require SSL for external server communication

To configure Tableau Server to use SSL for external communication between Tableau Server and web clients, run the `external-ssl enable` command as follows, providing the names for the server certificate's .crt and .key files:

```
tsm security external-ssl enable --cert-file <file.crt> --key-
file <file.key>
```

- For `--cert-file` and `--key-file`, specify the location and file name where you saved the server's CA-issued SSL certificate (.crt) and key (.key) files.
- The above command assumes the you are signed in as a user that has the **Server Administrator** site role on Tableau Server. You can instead use the `-u` and `-p` parameters to specify an administrator user and password.
- If the certificate key file requires a passphrase, include the `--passphrase` parameter and value.

Step 2: Configure and enable mutual SSL

Add mutual authentication between the server and each client, and allow for Tableau client users to be authenticated directly after the first time they provide their credentials.

1. Run the following command:

   ```
   tsm authentication mutual-ssl configure --ca-cert <certificate-file.crt>
   ```

   For `--ca-cert`, specify the location and file name of the certificate authority (CA) certificate file.

   The file (.crt) is an all-in-one file that includes certificates of CAs that are used for client authentication. The file you upload must be a concatenation of the various PEM-encoded certificate files, in order of preference.

2. Run the following commands to enable mutal SSL and apply the changes:

   ```
   tsm authentication mutual-ssl enable
   ```

   ```
   tsm pending-changes apply
   ```

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart

behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Additional options for mutual SSL

You can use `mutual-ssl configure` to configure Tableau Server to support the following options.

For more information, see tsm authentication mutual-ssl <commands>.

Fallback authentication

When Tableau Server is configured for mutual SSL, authentication is automatic and clients must have a valid certificate. You can configure Tableau Server to allow a fallback option, to accept user name and password authentication.

```
tsm authentication mutual-ssl configure -fb true
```

Tableau Serer accepts username and password authentication from REST API clients, even if the above option is set to `false`.

User name mapping

When Tableau Server is configured for mutual SSL, the server authenticates the user directly by getting the user name from their client certificate. The name that Tableau Server uses depends on how the server is configured for user authentication:

- **Local Authentication**—uses the UPN (User Principal Name) from the certificate.
- **Active Directory (AD)**—uses LDAP (Lightweight Directory Access Protocol) to get the user name.

You can override either of these defaults to set Tableau Server to use the common name.

```
tsm authentication mutual-ssl configure -m cn
```

For more information, see Mapping a Client Certificate to a User During Mutual Authentication

Certificate Revocation List (CRL)

You might need to specify a CRL if you suspect that a private key has been compromised, or if a certificate authority (CA) did not issue a certificate properly.

```
tsm authentication mutual-ssl configure -rf <revoke-file.pem>
```

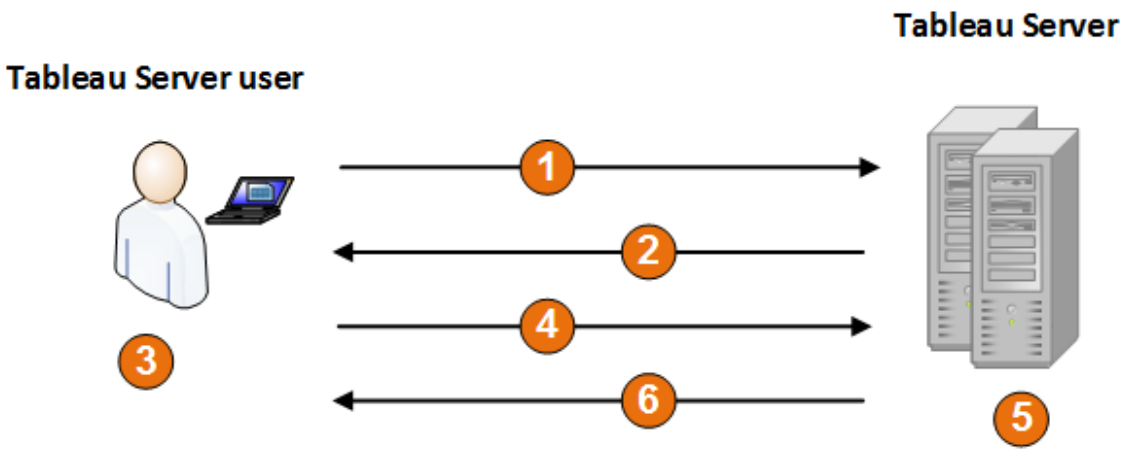
How Mutual SSL Authentication Works

Mutual (or two-way) SSL authentication provides a combination of an encrypted data stream, mutual authentication of both server and client, and direct access convenience. To use mutual SSL with Tableau Server, you need the following:

- External SSL configured on Tableau Server.
- A trusted CA-issued SSL certificate for Tableau Server. The file is a concatenation of CA certificate files. A "CA" is a *certificate authority* that issues certificates to the client computers that will connect to Tableau Server. The action of uploading the CA certificate file establishes a trust, which enables Tableau Server to authenticate the individual certificates that are presented by the client computers.
- A certificate on each client that will connect to Tableau Server.
- A Tableau Server configured to use mutual SSL.

Tableau Server and the client verify that each other has a valid certificate, and Tableau Server authenticates the user, based on the user name in the client certificate.

The following image shows a little more detail about the sequence of events that occurs with mutual SSL.



1. The user navigates to Tableau Server.
2. Tableau Server sends its SSL certificate to the client computer.
3. The client computer verifies the Tableau Server certificate.
4. The client computer sends its certificate to Tableau Server.
5. Tableau Server verifies the client certificate.
6. Tableau Server references the user name in the client certificate to authenticate the user.

Mapping a Client Certificate to a User During Mutual Authentication

When you use mutual (two-way) SSL authentication, the client presents its certificate to Tableau Server as part of the authentication process. Tableau Server then maps user information in the client certificate to a known user identity. The strategy that Tableau Server uses to perform client mapping depends on the content of your organization's client certificates.

This topic discusses the ways information in a client certificate can map to a user identity and how to change the way Tableau Server performs that mapping. To understand how the mapping happens and whether you need to change it, you must know how client certificates are structured in your organization.

- User-name mapping options
- Change the certificate mapping
- Address user-name ambiguity in multi-domain organizations

User-name mapping options

Tableau Server uses one of the following approaches to map a client certificate to a user identity:

- **Active Directory**. If Tableau Server is configured to use Active Directory for user authentication, when Tableau Server receives a client certificate, it passes the certificate to Active Directory, which maps the certificate to an Active Directory identity. Any explicit user name information in the certificate is ignored.

  Note: This approach requires client certificates to be published for the user accounts in Active Directory.

- **User principal name (UPN)**. A client certificate can be configured to store the user name in the user principal name field. Tableau Server reads the UPN value and maps it to a user in Active Directory or to a local user.
- **Common name (CN)**. A client certificate can be configured to store the user name in the common name field of the certificate. Tableau Server reads the CN value and maps it to a user in Active Directory or to a local user.

If you configure the server for Active Directory authentication and UPN or CN user-name mapping, put the user name in one of the following formats:

`username`, `domain/username`, or `username@domain`.

For example: `jsmith`, `example.org/jsmith`, or `jsmith@example.org`.

If the server uses local authentication, the format of the name in the UPN or CN fields is not predetermined, but the name in the field must match a user name on the server.

Change the certificate mapping

You use the tsm authentication mutual-ssl <commands> commands to map a client certificate to a user identity in Tableau Server:

```
tsm authentication mutual-ssl configure -m <value>
```

Possible values are `ldap` for Active Directory mapping, `upn` for UPN mapping, or `cn` for CN mapping.

When you first install and configure Tableau Server, the server sets the default user-name mapping to match the server's authentication type:

- If the server is configured to use Active Directory, it also uses Active Directory for mapping the certificate to the user identity.
- If the server is configured to use local authentication, the server gets the user-name value from the UPN field in the certificate.

If the default behavior for how Tableau Server maps a user name to an identity is not correct for your server configuration, run the following set of commands to change the mapping to use the CN value:

```
tsm authentication mutual-ssl configure -m cn

tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Address user-name mapping ambiguity in multi-domain organizations

Under some circumstances, the user name in a certificate's UPN or CN field can be ambiguous. This ambiguity can lead to unexpected results when the user name is mapped to a user identity on the server.

For example, if Tableau Server is presented with a user name that does not include a domain, the server maps the user name to an identity using the default domain. This can cause an incorrect user-name mapping, potentially assigning a user a different user's identity and permissions.

This can occur particularly in environments where the following conditions apply:

- Your organization supports multiple Active Directory domains.
- The server is configured to use Active Directory authentication.
- The server is configured to use UPN or CN mapping.
- Some users have the same user name but different domains. For example, `jsmith@example.org` and `jsmith@example.com`.
- The user name in the certificate's UPN or CN fields does not include the domain as part of the user name—for example, it shows `jsmith`.

To avoid incorrect user-name mapping, make sure the client certificates include fully qualified user names with the domain, using the format `jsmith@example.org` or `example.org/jsmith`.

OpenID Connect

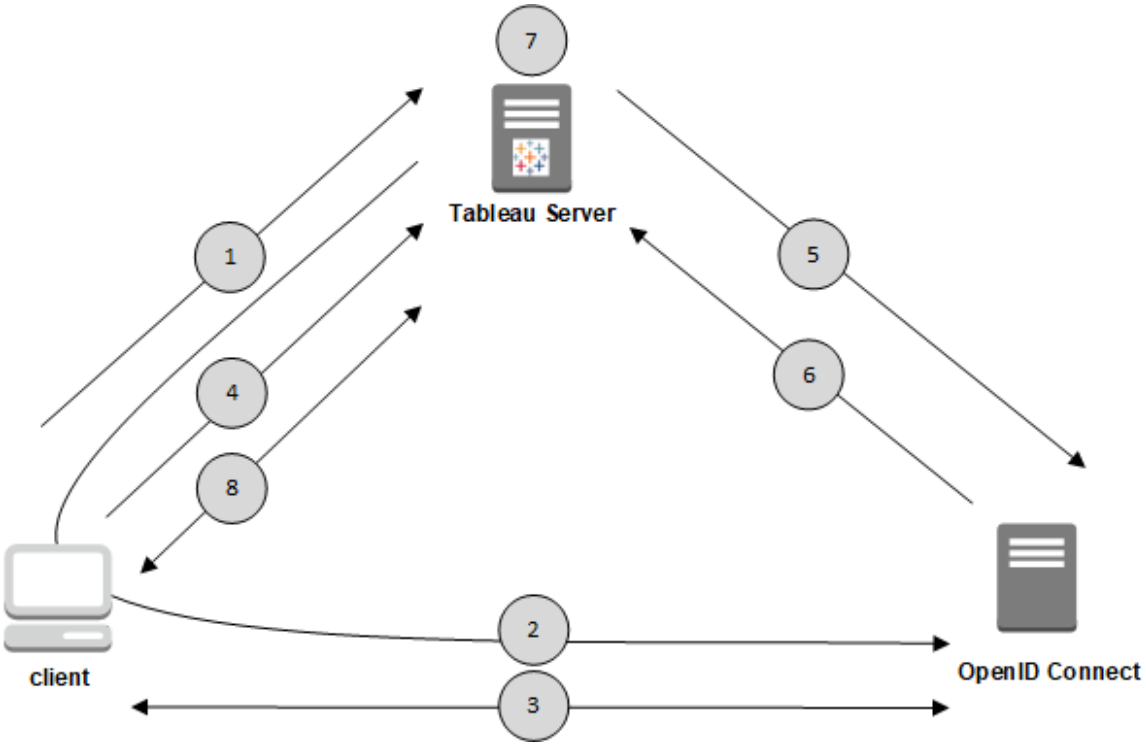
You can configure Tableau Server to support OpenID Connect (OIDC) for single sign-in (SSO). OIDC is a standard authentication protocol that lets users sign in to an identity provider (IdP) such as Google or Salesforce. After they've successfully signed in to their IdP, they are automatically signed in to Tableau Server.

Configuring OIDC involves several steps. The topics in this section provide general information about using Tableau Server with OIDC, and provide a sequence for configuring the IdP and Tableau Server.

Note: Unless otherwise noted, information about OIDC authentication applies to both OIDC authentication configured in TSM during Tableau Server setup or OIDC authentication configured with identity pools.

Authentication overview

This section describes the OpenID Connect (OIDC) authentication process with Tableau Server.



1. A user attempts to log in to Tableau Server from a client computer.

2. Tableau Server redirects the request for authentication to the IdP gateway.

3. The user is prompted for credentials and successfully authenticates to the IdP. The IdP responds with a redirect URL back to Tableau Server. The redirect URL includes an authorization code for the user.

4. The client is redirected to Tableau Server and presents the authorization code.

5. Tableau Server presents the client's authorization code to the IdP along with its own client credentials. Tableau Server is also a client of the IdP. This step is intended to prevent spoofing or man-in-the-middle attacks.

6. The IdP returns an access token and an ID token to Tableau Server.

- JSON Web Token (JWT) validation: By default Tableau Server performs a validation of the IdP JWT. During discovery, Tableau Server retrieves the public keys specified by the `jwks_uri` in the IdP configuration discovery document. Tableau Server validates the ID token for expiry and then verifies the JSON web signature (JWS), the issuer (IdP), and the client ID. You can learn more about the JWT process in the OpenID documentation, 10. Signatures and Encryption, and the IETF proposed standard, JSON Web Token. We recommend leaving JWT validation enabled, unless your Idp does not support it.

- The ID token is a set of attribute key-pairs for the user. The key-pairs are called *claims*. Here is an example IdP claim for a user:

```
"sub"                    :
"7gYhRR3HiRRCaRcgvY50ubrtjGQBMJW4rXbpPFp-
g2cptHP62m2sqowM7G1LwjN5"
"email"                  : "alice@example.com",
"email_verified"         : true,
"name"                   : "Alice Adams",
"given_name"             : "Alice",
"family_name"            : "Adams",
"phone_number"           : "+359 (99) 100200305",
"profile"                : "https://tableau.com/users/alice"
```

7. Tableau Server identifies the user from the IdP claims and completes the authentication request from step 1. Tableau Server searches for the user's account record stored in the repository by matching the "sub" (subject identifier) to identify the correct user account. If no user account is stored with the sub claim value, then Tableau Server searches for a username in the repository that matches the "email" claim from the IdP. When a username match succeeds, Tableau Server will store the corresponding sub claim to the user's record in the repository. Tableau Server can be configured to use different claims for this process. See Requirements for Using OpenID Connect.

8. Tableau Server authorizes the user.

How Tableau Server works with OpenID Connect

OpenID Connect (OIDC) is a flexible protocol that supports many options for the information that's exchanged between a service provider (here, Tableau Server) and an IdP. The following list provides details about the Tableau Server implementation of OIDC. These details can help you understand what types of information Tableau Server sends and expects, and how to configure an IdP.

- Tableau Server supports only the OpenID Authorization Code Flow as described in the OpenID Connect final specification in the OpenID Connect documentation.
- Tableau Server relies on using discovery or a provider URL to retrieve the OpenID provider metadata. Alternatively, you can host a static discovery document on Tableau Server. For more information see Configure Tableau Server for OpenID Connect.
- Tableau Server supports the `client_secret_basic` and `client_secret_post` client authentication.
- Tableau Server expects a `kid` value in the `id_token` attribute's JOSE Header. This value is matched with one of the keys found in the JWK Set document, whose URI is specified by the `jwks_uri` value in the OpenID discovery document. A `kid` value must be present even if there is only one key in the JWK Set document.

- Tableau Server does include OpenID support for the JWK `x5c` parameter or for using X.509 certificates.
- By default, Tableau Server ignores proxy settings and sends all OpenID requests directly to the IdP.

  If Tableau Server is configured to use a forward proxy to connect to the internet, then you must make additional changes as described in Configure Tableau Server for OpenID Connect.

Requirements for Using OpenID Connect

This topic describes the requirements to use OpenID Connect with Tableau Server.

Note: The TSM authentication configuration commands apply only to OIDC authentication configured in TSM during Tableau Server setup. To make OIDC authentication configuration changes for identity pools, you can use the Update Authentication Configuration endpoint using Tableau REST OpenAPI.

Summary of requirements

- IdP account
- Local identity store
- IdP claims - mapping users
- Authentication context

IdP account

You must have access to an identity provider (IdP) that supports the OpenID Connect (OIDC) protocol. You must also have an account with the IdP. OpenID Connect is supported by many identity providers. The OIDC protocol is an open and flexible standard, and as such, not all implementations of the standard are identical. As you configure Tableau Server for OIDC, work with your IdP.

The Google IdP implementation has been extensively tested with Tableau Server and is the model IdP for the configuration documented in these topics.

Local identity store

To use OpenID Connect on Tableau Server, one of the following must be true:

- **If configuring OIDC in TSM during Tableau Server setup**, Tableau Server must be configured to use a local identity store. The server must be configured so that you explicitly create users on the Tableau Server, rather than importing them from an external directory such as Active Directory. Managing users with an external identity store is not supported with OpenID.
- **If configuring OIDC using identity pools**, OIDC can be configured with 1) a local identity store or 2) AD or LDAP is the identity store configured in TSM during Tableau Server setup.

IdP claims - mapping users

To sign in successfully to Tableau Server, a given user must be provisioned in OpenID and then mapped to a user account on Tableau Server. OpenID uses a method that relies on *claims* to share user account attributes with other applications. Claims include user account attributes such as email, phone number, given name, etc. To understand how Tableau Server maps IdP claims to user accounts, see OpenID Connect.

Tableau Server relies on the IdP claims to map user accounts from the IdP to those hosted on Tableau Server. By default, Tableau Server expects the IdP to pass the email claim. Depending on your IdP, you may need to configure Tableau Server to use a different IdP claim.

If you are using Google as an IdP, then use the default, `email` claim to map IdP identities to Tableau Server user accounts. If you are not using Google as an IdP, then work with your IdP to determine the claim for which you should configure Tableau Server.

Default: using email claim to map users

By default, the user's user name in Tableau Server must match the `email` claim in the IdP ID token. Therefore, in the default configuration, you must use email addresses (also referred to

as UPN) as the username in Tableau Server. If you use Google as the IdP, the user name in Tableau Serverr must be the user's Gmailaddress (`alice@gmail.com`). Using a complete email address helps to guarantee the uniqueness of the user name in Tableau Server, even when two users have the same email but are on different email hosts.

Note: When you create a user identity in Tableau Server, you specify a user name, password, and optionally an email address. For using OpenID Connect in the default configuration, the user name (expressed as an email address) is the value that must match the user's name in the IdP. The optional email address in the Tableau Server user identity is not used for OpenID authentication.

Ignoring the domain name

You can configure Tableau to ignore the domain portion of an email address when matching the IdP `email` claim to a user account on Tableau Server. In this scenario, the `email` claim in the IdP might be `alice@example.com`, but this will match a user named `alice` in Tableau Server. Ignoring the domain name might be useful if you already have users defined in Tableau Server that match the user names portion of the `email` claim, but not the domain portions.

Important: We do not recommend ignoring the user domain name without taking precautions. Specifically, verify that user names are unique across the configured domains that you've created in your IdP.

Setting Tableau Server to ignore the user domain name has the potential to result in unintended user log on. Consider the case where your IdP has been configured for multiple domains (`example.com` and `tableau.com`). If two users with the same first name, but different user accounts (`alice@tableau.com` and `alice@example.com`) are in your organization, then the first one to complete the OpenID provisioning sequence will claim the `sub` mapping in the IdP. If the wrong user is mapped, then the other user will be unable to log on until the associated `sub` value is reset.

To configure Tableau Server to ignore domain names in user names from the IdP, set `tsm authentication openid configure --ignore-domain` to `true`. For more information, see tsm authentication openid <commands>.

When you change the tsm authentication openid configure `--ignore-domain` option to ignore the domain in user names, all user names in Tableau Server must have a domain name.

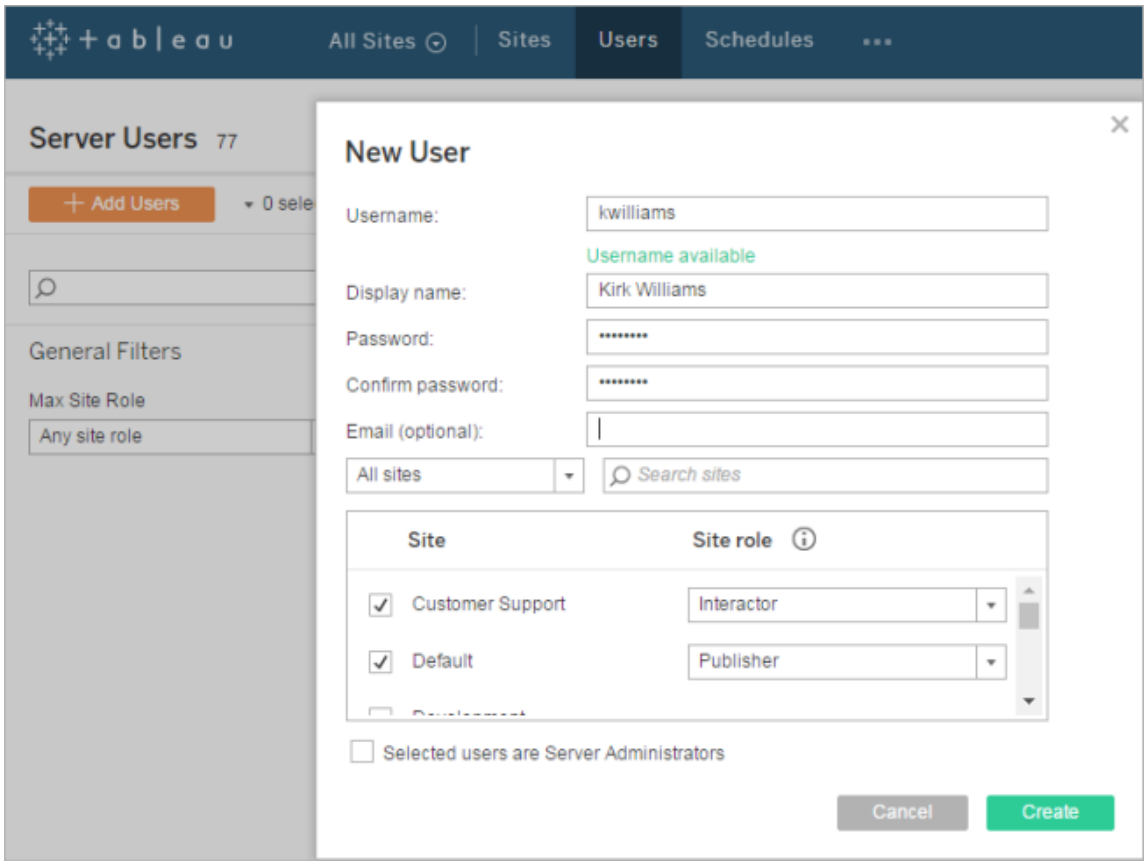
Using custom claims to map users

As referenced in OpenID Connect, the `sub` claim is often included in IdP claims. Typically, the `sub` claim is a unique string that identifies a given user account. The benefit of using a `sub` claim is that it will not change, even if you or another admin updates other user attributes or IdP claims (email, phone number, etc) associated with that account. By default, Tableau Server identifies and verifies OpenID users according to the `sub` claim in the IdP ID token.

The OpenID `sub` claim value must be mapped to the corresponding user in Tableau Server. Since the `sub` claim is an arbitrary string, a different claim is used to associate accounts during the first sign-in session. The first time a user signs in to Tableau Server with OpenID, Tableau will match the OpenID user account to a corresponding user account in Tableau Server. By default, Tableau will use the IdP claim, `email`, to identify the Tableau user. Tableau will then update that user's record with the `sub` claim from OpenID. Since the ID token always includes the `sub` claim along with other claims, on subsequent sessions, Tableau will identify that user with the `sub` claim only.

For some organizations, mapping user names with the email address is not reliable or not supported by the IdP. Beginning with Tableau Server 10.2, you can map user accounts from any arbitrary IdP claim to the Tableau Server username.

The IdP claim you are using must map exactly to a corresponding Tableau Server username. In the example below, the username is `kwilliams`.



To change the IdP claim that is used to map identity on Tableau Server, use the `tsm authentication openid map-claims --user-name` command. For more information, see tsm authentication openid <commands>.

Changing the `sub` claim

As described above, the `sub` claim is the identifier that Tableau Server uses to identify users after the initial mapping session. The `sub` claim is written to the corresponding user account in Tableau Server. If your IdP does not provide a `sub` claim, then you can specify an arbitrary claim to use instead. Like `sub`, the claim value you specify must be unique and should not change when other user claims are updated.

To specify a different IdP claim for default sub claim, use the use the `tsm authentication openid map-claims --id` command. For more information, see tsm authentication openid <commands>.

Where `arbitraryClaim` is the name of the IdP claim that you want to use as the replacement for the `sub` claim.

Authentication context

If your OpenID Connect IdP requires a specific authentication context, you can specify a list of essential and voluntary ACR values using the `vizportal.openid.essential_acr_values` and `vizportal.openid.voluntary_acr_values` configuration keys. For more information, see tsm configuration set Options.

Configure the Identity Provider for OpenID Connect

This topic provides information about configuring an identity provider (IdP) to use OpenID Connect (OIDC) with Tableau Server. This is one step in a multi-step process. The following topics provide information about configuring and using OIDC with Tableau Server.

1. OpenID Connect Overview
2. Configure the Identity Provider for OpenID Connect (you are here)
3. Configure Tableau Server for OpenID Connect
4. Signing In to Tableau Server Using OpenID Connect

Configure the IdP

Before you can use OpenID Connect with Tableau Server, you must have an account with an identity provider (IdP) and a project or application with the IdP. When you configure Tableau Server, you will need to be able to provide the following information:

- Client ID. This is the identifier that the IdP assigned to your application.
- Client secret. This is a token that is used by Tableau to verify the authenticity of the response from the IdP. This value is a secret and should be kept securely.
- Configuration URL. This is the URL at the provider's site that Tableau Server should send authentication requests to.

Redirect URL

Some IdPs will require a redirect URL for your Tableau Server.

You can manually construct your URL for the IdP using the following syntax:

```
<protocol>://<host>/vizportal/api/web/v1/auth/openIdLogin
```

For example, `https://tableau.example.com/vizportal/api/web/v1/auth/openIdLogin`.

Example IdP process

The following procedure provides an outline of the steps that you follow with the provider. As an example, the procedure discusses using Google as a provider. However, each provider has a somewhat different flow, so the specifics of the steps (and their order) might vary depending on your provider.

1. Register at the provider's developer site and sign in. For example, for Google, you can go to the Developers Console at this URL: https://console.developers.google.com
2. Create a new project, application, or relying party account.
3. In the developer dashboard, follow the steps for getting an OAuth 2.0 client ID and client secret. Record these values for later.

   Note: Keep the client secret in a secure place.

4. On the developer site, find the URL of the endpoint that the IdP uses for OpenID Connect discovery. For example, Google uses the URL https://accounts.google.com/.well-known/openid-configuration. Record this URL for later.

   Alternatively, if your IdP has provided you with a static discovery document, copy that file to a local directory on the Tableau Server for later.

Configure Tableau Server for OpenID Connect

This topic describes how to configure Tableau Server to use OpenID Connect (OIDC) for single-sign on (SSO). This is one step in a multi-step process. The following topics provide information about configuring and using OIDC with Tableau Server.

1. OpenID Connect Overview
2. Configure the Identity Provider for OpenID Connect
3. Configure Tableau Server for OpenID Connect (you are here)
4. Signing In to Tableau Server Using OpenID Connect

Notes:

- Before you perform the steps described here, you must configure the OpenID identity provider (IdP) as described in Configure the Identity Provider for OpenID Connect.
- The procedures described in this topic apply to OIDC authentication configured in TSM during Tableau Server setup and not OIDC authentication configured with identity pools. For more information about identity pools, see Provision and Authenticate Users Using Identity Pools.

Use the TSM web interface

1. Open TSM in a browser:

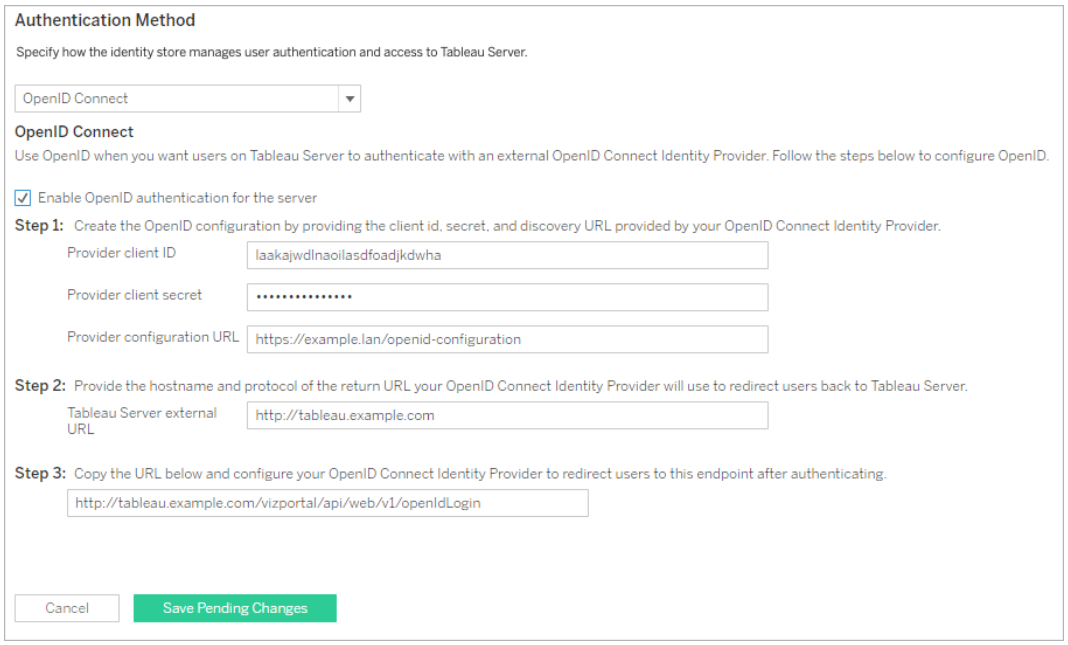
   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click User **Identity & Access** on the **Configuration** tab and then click **Authentication Method**.

3. Under **Authentication Method**, select **OpenID Connect** in the drop-down menu.

4. Under OpenID Connect, select **Enable OpenID authentication for the server**.

5. Enter the OpenID configuration information for your organization:



Note: If your provider relies on a configuration file hosted on the local computer (rather than a file hosted at a public URL), you can specify the file with the tsm authentication openid <commands>. Use the `--metadata-file <file_path>` option to specify a local IdP configuration file.

6. Click **Save Pending Changes** after you've entered your configuration information.

7. Click **Pending Changes** at the top of the page:

8. Click **Apply Changes and Restart**.

Use the TSM CLI

The procedure in this section describes how to use TSM command line interface to configure OpenID Connect. You can also use a configuration file for the initial configuration of OpenID Connect. See openIDSettings Entity.

1. Use the `configure` command of tsm authentication openid <commands> to set the following required options:

   `--client-id <id>`: Specifies the provider client ID that your IdP has assigned to your application. For example, `"laakjwdlnaoiloadjkwha"`.

   `--client-secret <secret>`: Specifies the provider client secret. This is a token that is used by Tableau to verify the authenticity of the response from the IdP. This value is a secret and should be kept securely. For example, `"fwahfkjaw72123="`.

   `--config-url <url>` or `--metadata-file <file_path>`: Specifies location of provider configuration json file. If the provider hosts a public json discovery file, then use `--config-url`. Otherwise, specify a path on the local computer and file name for `--metadata-file` instead.

   `--return-url <url>`: The URL of your server. This is typically is the public name of your server, such as `"http://example.tableau.com"`.

   For example, run the command:

   ```
   tsm authentication openid configure --client-id "laakjwdlna-
   oiloadjkwha" --client-secret "fwahfkjaw72123=" --config-url
   "https://example.com/openid-configuration" --return-url
   "http://tableau.example.com"
   ```

There are additional, optional configurations that you can set for Open ID Connect using either openIDSettings Entity or tsm authentication openid <commands>. In addition, if you need to configure IdP claim mapping, see Options for openid map-claims.

2. Type the following command to enable Open ID Connect:

   ```
   tsm authentication openid enable
   ```

3. Run `tsm pending-changes apply` to apply changes.

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Signing In to Tableau Server Using OpenID Connect

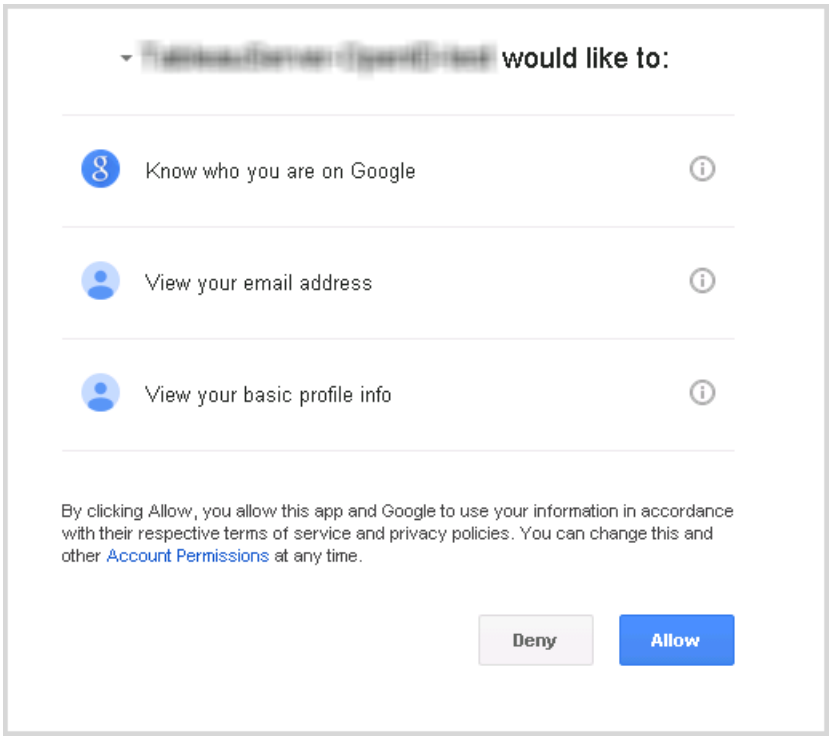
This topic provides information about signing in to Tableau Server using OpenID Connect. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- OpenID Connect
- Configure the Identity Provider for OpenID Connect
- Configure Tableau Server for OpenID Connect
- Signing In to Tableau Server Using OpenID Connect (you are here)

Signing in using OpenID Connect

After Tableau Server has been configured to use OpenID Connect, users who access the server and aren't already signed in are redirected to the IdP site, where they are prompted to sign in. Users enter the credentials that they have with the IdP. In many cases, the user is also

asked to authorize the IdP to share information with Tableau Server, as in the following example:



When a user signs in using OpenID Connect, the IdP sends a unique user identifier (known in OpenID as the sub value) as part of the information that's redirected to Tableau Server. This sub value is associated with the user's Tableau user identity.

Restricting sign-in to server administrators for command-line tools

Command-line tools for working with Tableau Server (`tabcmd`, TSM, and `tableau.com`) do not support sign-in using OpenID Connect. When OpenID Connect is enabled for the server, these tools still require sign-in using a Tableau Server username and password.

Even if users normally authenticate using OpenID Connect, each user has a Tableau Server username and password. This means that users could use command-line tools like `tabcmd`. As a security measure, you can make sure that *only* server administrators can use command-line tools. To do this, use `tsm configuration set` to set `wgserver.authentication.restricted` to `true`. When this setting is `true`, only server

administrators can sign in to Tableau Server using a username and password; all other users *must* sign in to the server using a single sign-on (SSO) option like OpenID Connect. The effect is that users who are not administrators also cannot then use command-line tools. To make this change, run the following sequence of TSM commands:

```
tsm configuration set -k wgserver.authentication.restricted -v true

tsm pending-changes apply
```

OpenID Connect Authentication Request Parameters

The OpenID authentication request sent from Tableau Server passes information using a limited set of parameters, as listed in this topic. If your OpenID IdP requires parameters that are not in the following list, it is not compatible for use with Tableau Server.

- `scope`. This value specifies a profile that tells the IdP what user information claims to return. This value can be configured by a Tableau Server administrator. The default value is "openid email profile". For more information, see Configure the scope value later in this document.
- `response_type`. OpenID Connect supports multiple flows. This value tells the IdP which flow Tableau Server expects. Tableau supports only the authorization code flow, and the value is always set to "code".
- `client_id`. This value specifies the server's ID (**Provider client ID** in the Tableau Server Configuration dialog box), which lets the IdP knows where the request came from. It is provided by the IdP when the service is registered. The value is configurable by a Tableau Server administrator.
- `redirect_uri`. This value specifies the URL that the IdP redirects to after the user has authenticated using OpenID Connect. The URL must include the host and protocol (for example, `http://example.tableau.com`), but Tableau provides the URL endpoint.
- `nonce`. Tableau Server generates a nonce value to verify that the client that it redirected to matches the entity that comes back from the IdP.

Configure the scope value

The `scope` value indicates to the IdP the information that Tableau Server requests about the user. By default, Tableau Server sends the value "openid profile email". This indicates that Tableau uses OpenID to authenticate (this part of the `scope` attribute value must always be included) and that Tableau Server is requesting the user profile and email information during the exchange of the user authorization code.

If this default scope is not appropriate for your scenario, you can have Tableau Server request custom information about the user. To do so, you configure the IdP with a custom profile (for example, something like "tableau-scope"). You can then configure Tableau Server to send the scope request using the name of the custom profile.

To change the scope value that Tableau Server requests, use the following TSM CLI command:

```
tsm authentication openid configure --custom-scope-name custom-scope-name
```

Notes:

- Tableau Server always includes "openid" as part of the scope value (even if you don't include it in the `custom_scope` setting).
- The TSM authentication configuration commands apply only to OIDC authentication configured in TSM during Tableau Server setup. To make OIDC authentication configuration changes for identity pools, you can use the Update Authentication Configuration endpoint using Tableau REST OpenAPI.

Changing IdPs in Tableau Server for OpenID Connect

This topic provides information about changing an identity provider (IdP) if you have configured Tableau Server to use OpenID Connect.

Change providers

You might decide to change the IdP that Tableau Server is configured to use. To do so, you follow the procedure that you used to configure the first IdP: establish an account, get a customer

ID and secret, configure Tableau Server with that information, and provide the IdP with the redirect URL for Tableau Server. For more information, see Configure Tableau Server for OpenID Connect.

Reset user identifiers

However, you also need to perform an additional step: you must clear any user identifiers (`sub` values or claims) that have already been associated with Tableau Server users. The new IdP will have different `sub` values for each user, and you must clear the existing ones so that Tableau Server can store a new `sub` value when the user signs in using the new IdP.

To clear sub values for users, use the `tabcmd reset_openid_sub` command. You can reset (that is, clear) `sub` values for an individual user, as in the following example:

```
tabcmd reset_openid_sub --target-username jsmith
```

You can also clear the sub value for all users using this command:

```
tabcmd reset_openid_sub --all
```

Note: Clearing user identifiers for members of an identity pool is not supported.

Troubleshoot OpenID Connect

Use the following topics to troubleshoot OpenID Connect (OIDC) issues in Tableau Server.

OIDC protocol is supported by many identity providers. The OIDC protocol is an open and flexible standard, and as such, not all implementations of the standard are identical. Most issues that administrators encounter when configuring Tableau Server for OIDC are the result of how different identity providers implement OIDC. If you encounter errors as you set up OIDC with Tableau Server, we recommend that you work with your IdP to resolve them.

Enabling enhanced OpenID logging

To efficiently troubleshoot OpenID Connect issues in Tableau Server, enable enhanced logging by setting the logging level to debug, and full logging for OpenID using the

`vizportal.openid.full_server_request_logging_enabled` configuration key to `true` using these TSM commands:

```
tsm configuration set -k vizportal.log.level -v debug

tsm configuration set -k vizportal.openid.full_server_request_log-
ging_enabled -v true

tsm pending-changes apply
```

After completing your troubleshooting, we recommend setting the values of both configuration keys back to their defaults to limit the information collected in logs and to reduce log file sizes. For details on resetting configuration keys to defaults, see Resetting a configuration key to default.

Note: Enhanced logging for identity pools isn't supported. However, `vizportal.log.level debug` logging is supported.

Signing in from the command line

Even if Tableau Server is configured to use OIDC, it isn't used if you sign in to Tableau Server using tabcmd, the Tableau REST API, or the Tableau Data Extract command line utility (provided with Tableau Desktop).

Login failed

Login can fail with the following message:

```
Login failure: Identity Provider authentication unsuccessful for
user <username from IdP>. Failed to find the user in Tableau Cloud.
```

This error typically means that there's a mismatch between a username stored in Tableau Server and the username provided by the IdP. To fix this, make sure that they match. For example, if Jane Smith's username is stored in the IdP as `jsmith` it must be stored in Tableau Server as `jsmith` as well.

Error 69: "Unable to Sign In"

An error 69 might be returned when you attempt to sign in to Tableau Server with a web browser and receive an error, "Unable to Sign In. Sign in failed. Contact your Tableau Server administrator." The URL that returns this message is `https://example.-com/#/error/signin/69?redirectPath=%2`.

If you receive this error, check with your IDP provider to verify if the IdP is expecting `client_secret_post` instead of `client_secret_basic` (the Tableau default).

If the IdP is expecting `client_secret_post`, then you must set the `viz-portal.openid.client_authentication` parameter to `client_secret_post`.

For example, if you receive this error and you have configured OIDC for the Salesforce IdP, then you must set the `vizportal.openid.client_authentication` parameter.

See tsm configuration set Options for more information.

OpenID error log

OpenID authentication takes place outside Tableau Server, so troubleshooting authentication issues can be difficult. However, sign-in attempts are logged by Tableau Server. You can create a snapshot of log files and use them to troubleshoot problems. For more information, see Tableau Server Logs and Log File Locations.

> **Note**: To log OpenID-related events, `vizportal.log.level` must be set to `debug` with tsm configuration set Options.

Check for OpenID errors in the following files in the unzipped log file snapshot:

`\vizportal\vizportal-<n>.log`

User not found

An error "user not found" might be returned if the "sub" claims have changed after initial login of users. You can verify this issue if you see the following in the vizportal logs: `Possible conflicting or stale account: <username> A different user already owns this account.`

If you continue to see this issue, reset the "sub" claims for that user or for all users on Tableau Server. For more information, see Reset user identifiers.

Trusted Authentication

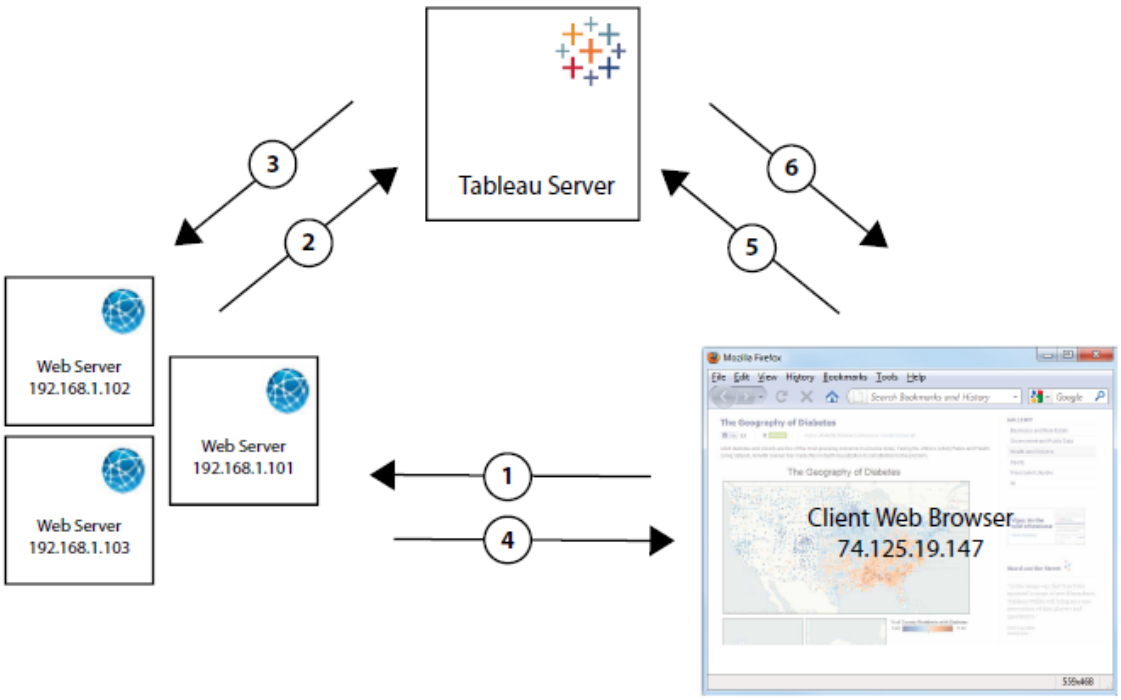
When you embed Tableau Server views into web pages, everyone who visits the page must be a licensed user on Tableau Server. When users visit the page they are prompted to sign in to Tableau Server before they can see the view. If you already have a way of authenticating users on the webpage or within your web application, you can avoid this prompt and save your users from having to sign in twice by setting up trusted authentication.

Trusted authentication simply means that you have set up a trusted relationship between Tableau Server and one or more web servers. When Tableau Server receives requests from these trusted web servers it assumes that your web server has handled whatever authentication is necessary.

> **Note:** Client browsers must be configured to allow third-party cookies if you want to use trusted authentication with embedded views.

How Trusted Authentication Works

The diagram below describes how trusted authentication works between the client's web browser, your web server(s) and Tableau Server.



User visits the webpage: When a user visits the webpage with the embedded Tableau Server view, the webpage sends a GET request to your web server for the HTML for that page.

Web server POSTS to Tableau Server: The web server sends a POST request to the trusted Tableau Server (for example, `https://<server_name>/trusted`, not `https://<server_name>`). That POST request must have a `username` parameter. The `username` value must be the username for a licensed Tableau Server user. If Tableau Server is hosting multiple sites and the view is on a site other than the Default site, then the POST request must also include a `target_site` parameter.

Tableau Server creates a ticket: Tableau Server checks the IP address or host name of the web server (192.168.1.XXX in the above diagram) that sent the POST request. If the web server is listed as a trusted host then Tableau Server creates a ticket in the form of a unique string. Tickets must be redeemed within three minutes after they are issued. Tableau

Server responds to the POST request with that ticket. Or if there is an error and the ticket cannot be created, then Tableau Server responds with a value of `-1`. The server must have an IPv4 address. IPv6 addresses are not supported. For more information, see Ticket Value of -1 Returned from Tableau Server.

4 **Web server passes the URL to the browser**: The web server constructs the URL for the view and inserts it into the HTML for the page. The ticket is included (for example, `https://<server_name>/trusted/<unique_ticket>/views/<view_name>`). The web server passes the HTML back to the client's web browser.

5 **Browser requests view from Tableau Server**: The client web browser sends a GET request to Tableau Server that includes the URL with the ticket.

6 **Tableau Server redeems the ticket**: Tableau Server redeems the ticket, creates a session, logs the user in, removes the ticket from the URL, and then sends the final URL for the embedded view to the client.

The session allows the user to access any of the views that the user would have if they logged onto the server. In the default configuration, users authenticated with trusted tickets have restricted access such that only views are available. They cannot access workbooks, project pages, or other content hosted on the server.

To change this behavior, see the `wgserver.unrestricted_ticket` option at tsm configuration set Options.

How is a trusted ticket stored?

Tableau Server stores trusted tickets in the Tableau Server repository using the following process:

1. Tableau Server generates a two-part ticket: the first part is a Base64-encoded unique ID (UUID) and the second part is a 24-character random secret string.
2. Tableau Server hashes the secret string and stores it with the unique ID in the repository. Hashing takes the secret string as input, and uses an algorithm to compute a

unique string. This unique string protects the security of the secret string from unauthorized users.
3. Tableau Server sends the Base64 UUID and the original 24-character random string to the client.
4. The client returns the Base64 UUID and the original 24-character secret string to Tableau Server as part of the request for a view.
5. Tableau Server locates the string pair with the Base64 UUID, and then hashes the secret string to verify that it matches the hash stored in the repository.

This process ensures that any trusted ticket content stored on Tableau Server cannot be used to impersonate users or access content protected by authentication. However, because the full trusted ticket is sent over HTTP between Tableau Server and the client, the process relies on secure and encrypted transmission of HTTP data. Therefore, we recommend that you only deploy trusted tickets over SSL/TLS or another layer of network encryption.

Add Trusted IP Addresses or Host Names to Tableau Server

The first step in setting up trusted authentication is to configure Tableau Server to recognize and trust requests from one or more web servers:

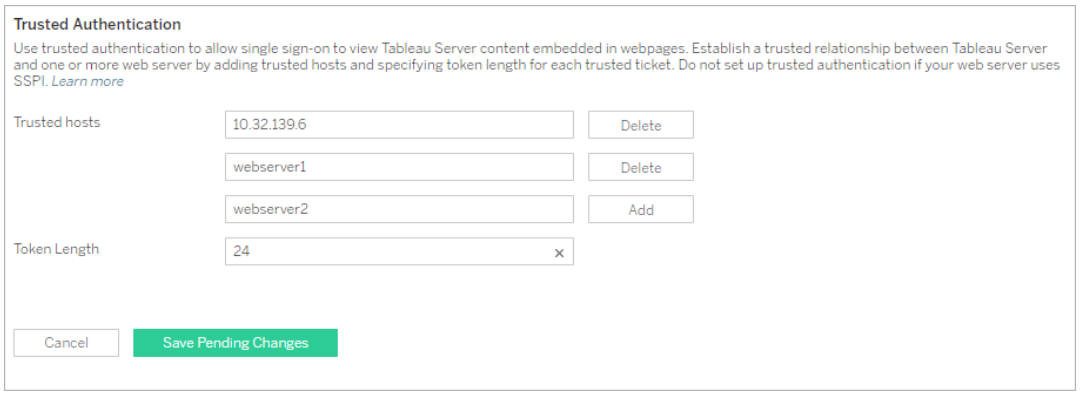
Use the TSM web interface

1. Open TSM in a browser:

   https://<server_name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click User **Identity & Access** on the **Configuration** tab and then click **Trusted Authentication**.

3. Under **Trusted Authentication**, for each trusted host, enter the name or IP address and then click **Add**:



Notes:
The values you specify completely overwrite any previous setting. Therefore, you must include the full list of hosts if you want to amend an existing list.

Static IP addresses are required: The web servers you specify must use static IP addresses, even if you use host names.

If you have one or more proxy servers between the computer that is requesting the trusted ticket (one of those configured in Step 2 as shown at Trusted Authentication) and Tableau Server, you also need to add them as trusted gateways using the `tsm configuration set gateway.trusted` option. See Configuring Proxies and Load Balancers for Tableau Server for steps.

4. Enter a value in **Token Length** (Optional).

   The token length determines the number of characters in each trusted ticket. The default setting of 24 characters provides 144 bits of randomness. The value can be set to any integer between 9 and 255, inclusive.

5. Click **Save Pending Changes** after you've entered your configuration information.

6. Click **Pending Changes** at the top of the page:

7. Click **Apply Changes and Restart**.

Use the TSM CLI

1. Enter the following command:

   ```
   tsm authentication trusted configure -th <trusted IP address or host name>
   ```

   In the command above, `<trusted IP address>` should be a comma-separated list of the IPv4 addresses or host names of your web server(s), with each host name or IP address in quotes.

   > **Note**: The values you specify completely overwrite any previous setting. Therefore, you must include the full list of hosts in the `tsm authentication trusted configure -th` command. (You cannot amend the list of hosts by running the `tsm authentication trusted configure -th` command repeatedly.)

   For example:

   ```
   tsm authentication trusted configure -th "192.168.1.101", "192.168.1.102", "192.168.1.103"
   ```

   or

   ```
   tsm authentication trusted configure -th "webserv1", "webserv2", "webserv3"
   ```

   > **Notes:**
   > Each host name or IP address in the list must be in double-quotes, followed by a

comma and one space after each comma.
The web servers you specify must use static IP addresses, even if you use host names.

2. If you have one or more proxy servers between the computer that is requesting the trusted ticket (one of those configured in Step 2 as shown at Trusted Authentication) and Tableau Server, you also need to add them as trusted gateways using the `tsm configuration set gateway.trusted` option. See Configuring Proxies and Load Balancers for Tableau Server for steps.

3. Type the following command to save the changes to all the server configuration files:

   ```
   tsm pending-changes apply
   ```

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

   There are other optional trusted authentication configurations (legacy support, logging, and timeout settings) that you can make by passing a json file to Tableau Server. See trustedAuthenticationSettings Entity.

Next, you need to configure your web server to receive tickets from Tableau Server.

Get a Ticket from Tableau Server

After you've added trusted IP addresses to Tableau Server, you're ready to configure your web server to get tickets from Tableau Server via POST requests (Step 2 in the diagram). The POST request must be sent to `http://<server_name>/trusted`. For example, the POST request must be sent to `http://tabserv/trusted` not `http://tabserv`.

Note: If SSL is enabled you must use https instead of http. For example: `https://tabserv/trusted`.

Here's the data you can use in a POST request to Tableau Server:

- **`username=<username>`** (required): The username for a licensed Tableau Server user. If you are using Local Authentication the username can be a simple string (for example, `username=jsmith`). If you are using Active Directory with multiple domains you must include the domain name with the user name (for example, `username=MyCo\jsmith`).
- **`target_site=<site id>`** (required if view not on Default site): Specifies the site containing the view if Tableau Server is running multiple sites and the view is on a site other than the Default site (for example, `target_site=Sales`). The value you use for `<site id>` should be the Site ID that was provided when the site was created. This value is case sensitive. If the **Site ID** is `SAles`, then the `target_site=SAles`.
- **`client_ip=<IP address>`** (optional): Used to specifiy the IP address of the computer whose web browser is accessing the view (for example, `client_ip=123.45.67.891`). It is not the IP address of the web server making the POST request of Tableau Server. If you decide to use this parameter, see Optional: Configure Client IP Matching for more information.

Tableau Server's response to the POST request will be a unique string (the ticket). If Tableau Server isn't able to process the request, the return will be `-1`. See Ticket Value of -1 Returned from Tableau Server for tips on how to correct this. Also, in order for users to successfully authenticate when they click an embedded view, their browsers must be configured to allow third-party cookies.

The ticket format changed in Tableau Server 10.2. The ticket format is now a string composed of two parts. Each part is a 128 bit string that is encoded before it is returned to the client. The first part is a universally unique ID (UUID v4) that is Base64-encoded. The second part is a 24-character secure random string. The concatenation of these parts can be expressed as

Base64(UUIDv4):SecureRandomString. An example of a ticket might look like this: `9D1ObyqDQmSIOyQpKdy4Sw==:dg62gCsSE0QRArXNTOp6mlJ5`.

Next, you need to add code that allows the web server to construct an URL for the view that includes the view's location and the ticket.

Display the View with the Ticket

After you create the POST request, you need to write code that provides the web server with the view's location and the ticket from Tableau Server. It will use this information to display the view. How you specify it depends on whether the view is embedded, and if Tableau Server is running multiple sites.

Tableau Server View Examples

Here's an example of how to specify a view that users only access via Tableau Server (the view is not embedded):

```
http://<server_name>/trusted/<unique_ticket>/views/<workbook_
name>/<view_name>
```

If Tableau Server is running multiple sites and the view is on a site other than the Default site, you need to add `t/<site_id>` to the path. For example:

```
http://<server_name/trusted/<unique_ticket>/t/Sales/views/<workbook_
name>/<view_name>
```

Use the same capitalization that you see in the Tableau Server URL.

Embedded View Examples

Here are some examples of how to specify embedded views. Because there are two approaches you can take with embed code, both ways are provided below. Regardless of which you use, there is some information unique to trusted authentication that you must provide. For more information, search for "Writing Embed Code" in the Tableau Server Help.

Note: The examples below use embed code parameters. For more information, see Embed Code Parameters in the Tableau Help.

Script Tag Examples

This example uses the `ticket` object parameter:

```
<script type="text/javascript" src-
c="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="800" height="600" style-
e="display:none;">
    <param name="name" value="MyCoSales/SalesScoreCard" />
    <param name="ticket" value-
e="9D1ObyqDQmSIOyQpKdy4Sw==:dg62gCsSE0QRArXNTOp6mlJ5" />
</object>
```

Here's what the above example looks like for a multi-site Tableau Server, where the view is published on the `Sales` site:

```
<script type="text/javascript" src-
c="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="800" height="600" style-
e="display:none;">
    <param name="site_root" value="/t/Sales" />
    <param name="name" value="MyCoSales/SalesScoreCard" />
    <param name="ticket" value-
e="9D1ObyqDQmSIOyQpKdy4Sw==:dg62gCsSE0QRArXNTOp6mlJ5" />
</object>
```

Instead of using `ticket`, you can use the `path` parameter to state the full path of the view explicitly. When `path` is used, you do not also need the `name` parameter, which is usually a required parameter in Tableau JavaScript embed code:

```
<script type="text/javascript" src-
c="http://myserver/javascripts/api/viz_v1.js"></script>
```

```
<object class="tableauViz" width="900" height="700" style-
e="display:none;">
    <param name="path" value-
="tru-
sted/9D1ObyqDQmSIOyQpKdy4Sw-
w==:dg62gCsSE0QRArXNTOp6mlJ5/views/MyCoSales/SalesScoreCard" />
</object>
```

Here's the same example, but for a multi-site server. Note that `/t/<site_id>` is used here:

```
<script type="text/javascript" src-
c="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="900" height="700" style-
e="display:none;">
     <param name="path" value-
="tru-
sted/9D1ObyqDQmSIOyQpKdy4Sw-
w==:dg62gCsSE0QRArXNTOp6mlJ5/t/Sales/views/MyCoSales/SalesScoreCard"
/>
</object>
```

Iframe Tag Example

```
<iframe src-
="h-
ttp://t-
abserver-
/trus-
ted/9D1ObyqDQmSIOyQpKdy4Sw-
w==:dg62gCsSE0QRArXNTOp6mlJ5/views/workbookQ4/SalesQ4?:embed=yes"
width="800" height="600"></iframe>
```

Optional: Configure Client IP Matching

By default, Tableau Server does not consider the client web browser IP address when it creates or redeems tickets. To change this, you need to do two things: specify an IP address using the `client_ip` parameter in the POST request that obtains the ticket, and follow the steps below to configure Tableau Server to enforce client IP address matching.

1. Open TSM CLI and type the following command:

   ```
   tsm configuration set -k wgserver.extended_trusted_ip_checking
   -v true
   ```

2. Then type the following command:

   ```
   tsm pending-changes apply
   ```

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Test Trusted Authentication

The steps below provide a method you can use to test retrieving a trusted ticket from your web server. This simple test can help evaluate connectivity between the web server and Tableau Server, and whether or not trusted authentication has been configured correctly.

> **Important**: The test code provided in this topic runs client-side in the browser to provide a quick, visual validation that trusted authentication is configured correctly on Tableau Server. If the client browser loading the html page is not on the trusted web server, you may have to temporarily trust the client IP. In practice, you should never trust client IP addresses as part of your trusted authentication configuration. All trusted IP addresses are able to request tickets as any user including your administrator. In a production environment, all trusted authentication flows should run server-side, between Tableau Server and the trusted web server(s) only.

Because the test is run on a client browser, the test isn't an exact replica of the communication path in a production environment. After successfully running the test, we recom-

mend that you then test requesting tickets for your users with a server-side post request for final verification.

An alternative testing method is to run a trusted ticket generator to test your configuration. The following URL references a trusted ticket generator that is not supported by Tableau. However the generator has been used by many customers to test their trusted ticket configuration: https://github.com/mkannan-tsi/Trusted-Ticket-Generator.

Step 1: Add a test user

Create a user on the Tableau Server that you can use to test trusted ticket functionality. See Add Users to Tableau Server. Add that user to a site on the server, and set the user's site role to **Explorer**.

Step 2: Create a test HTML page

Paste the following code into a new .html file that you save on the Tableau Server machine where you're performing the test from. You can change the labels and style attributes as you prefer.

```
<html>
<head>
<title>Trusted Ticket Requester</title>
<script type="text/javascript">
  function submitForm(){
    document.getElementById('form1').action =
    document.getElementById('server').value + "/trusted";
  }
</script>
<style type="text/css">
  .style1 {width: 100%;}
  .style2 {width: 429px;}
  #server {width: 254px;}
</style>
```

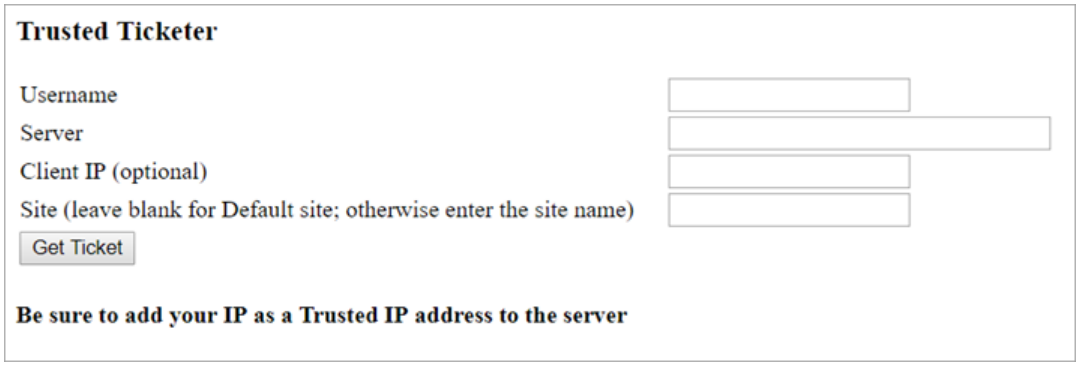
```
</head>
<body>
<h3>Trusted Ticketer</h3>
<form method="POST" id="form1" onSubmit="submitForm()">
  <table class="style1">
    <tr>
      <td class="style2">Username</td>
      <td><input type="text" name="username" value="" /></td>
    </tr>
    <tr>
      <td class="style2">Server</td>
      <td><input type="text" id="server" name="server" value-
e="https://" /></td>
    </tr>
    <tr>
      <td class="style2">Client IP (optional)</td>
      <td><input type="text" id="client_ip" name="client_ip"
value="" /></td>
    </tr>
    <tr>
      <td class="style2">Site (leave blank for Default site; oth-
erwise enter the site name)</td>
      <td><input type="text" id="target_site" name="target_site"
value="" /></td>
    </tr>
    <tr>
      <td class="style2"><input type="submit" name="submittable"
value="Get Ticket" /></td>
      <td> </td>
    </tr>
  </table>
```

```
</form>
<h4>Be sure to add your IP as a Trusted IP address to the server-
</h4>
</body>
</html>
```

Step 3: Retrieve a trusted ticket from Tableau Server

The following procedure will return a trusted ticket from Tableau Server.

1. Open the web page that you created in the previous step.



   This operation requires JavaScript, so the web browser might prompt you to allow scripts to run.

2. In the text boxes, enter the following:

   - **Username**: The test user that was created in Step 1.
   - **Server**: the address of your Tableau Server, e.g., `https://<server_name>`.
   - **Client IP (optional)**: The IP address of the user's computer, if it's configured for client trusted IP matching.
   - **Site**: The name of the Tableau Server site that the test user is a member of.

3. Click **Get Ticket**. One of the following will be returned:

   - **A unique ticket**: A trusted ticket is a string composed of a base64-encoded UUID and a 24-character random string, for example, `9D1OlxmDQmSIOyQpKdy4Sw==:dg62gCsSE0QRArXNTOp6mlJ5`.

- **-1**: If the value, `-1` is returned, the configuration contains an error. See Ticket Value of -1 Returned from Tableau Server.

Step 4: Test access with trusted ticket

Now that you have a ticket, you can use it to access content on Tableau Server.

Construct a URL with the unique ticket that you generated in the previous step to verify access with the trusted ticket. The URL syntax is different if you are accessing a Tableau Server with a single site vs a server that hosts multiple sites.

Default site server url

```
https://<server_name>/trusted/<unique_ticket>/views/<workbook_
name>/<view_name>
```

Non-default site server url

```
https://<server_name>/trusted/<unique_ticket>/t/<site_name>/views/<-
workbook_name>/<view_name>
```

Variables in the URLs are indicated by angle brackets (< and >). All other syntax is literal.

Troubleshoot Trusted Authentication

This section includes some common issues and errors you might encounter when you're configuring trusted authentication.

A common source for trusted authentication errors are misconfiguration with a proxy server or load balancer. If your Tableau Server operates behind a reverse proxy server or a load balancer, see Configure Tableau Server to work with a reverse proxy server and/or load balancer and Add a Load Balancer.

Trusted authentication information is written to `/var/opt/tableau/tableau_server-/data/tabsvc/logs/vizqlserver/vizql-*.log`.

To increase the logging level from `info` to `debug`, run the following commands:

```
tsm configuration set -k vizqlserver.trustedticket.log_level -v
debug
tsm pending-changes apply
```

To test your trusted authentication deployment, see Test Trusted Authentication.

See also

For more troubleshooting information for specific errors, see the following topics accessible from the **Other articles in this section** below:

Request for ticket by web server

- Ticket Value of -1 Returned from Tableau Server
- HTTP 401 - Not Authorized
- HTTP 404 - File Not Found
- Invalid User (SharePoint or C#)

Viewer redeeming ticket

- Attempting to Redeem the Ticket from the Wrong IP Address
- Cookie Restriction Error

Navigating between several embedded views

- An error occurred communicating with the server (403)

Ticket Value of -1 Returned from Tableau Server

Tableau Server returns -1 for the ticket value if it cannot issue the ticket as part of the trusted authentication process. Before troubleshooting this scenario, be sure to set the log level for trusted authentication to `debug` as specified in Troubleshoot Trusted Authentication.

The exact reason for this message is written to the `vizqlserver_node*-*.log.*` files in the following folder:

```
/var/opt/tableau/tableau_server/data/tabsvc/logs/vizqlserver
```

Here are some things to confirm:

- **All web server host names or IP addresses are added to trusted hosts**

  The log error, "`Invalid request host: <ip_address>`" may indicate that the IP address or host name for the computer sending the POST request is not in the list of trusted hosts on Tableau Server. See Add Trusted IP Addresses or Host Names to Tableau Server to learn how to add IP addresses or host names to this list.

- **IP addresses are IPv4**

  If you are using IP addresses to specify trusted hosts, they must be in Internet Protocol version 4 (IPv4) format. An IPv4 address looks like this: 123.456.7.890. IPv6 addresses (for example, fe12::3c4a:5eab:6789:01c%34) are not supported as a way of inputting trusted hosts.

- **Username in POST request is a valid Tableau Server user**

  The username you send in the POST request must be a licensed Tableau Server user. You can see a list of users by signing in to Tableau Server as an administrator.

  The following log errors indicate a user POST issue:

  - "`Missing username and/or client_ip`"
  - "`Invalid user: <username>`"
  - "`Unlicensed user is not allowed: <username>`"

  Username in POST request includes domain

  If Tableau Server is configured to use Local Authentication, the username that you send in the POST can be a simple string. However, if the server is configured for Active Directory you must include the domain name with the user name (domain\username). For example, the username parameter might be: `username=dev\jsmith`. A common error log for this scenario is "`Invalid user: <username>`".

- **Content-Type is specified**

If you are designing an ASP.NET or C# application, you need to declare the content type in your HTTP request. For example:

```
http.setRequestHeader("Content-Type","application/x-www-form-urlencoded;charset=UTF-8")
```

If you do not specify content type and Tableau Server returns a -1, the log files contain the error: "`missing username and/or client_ip`".

HTTP 401 - Not Authorized

If you receive a **401- Not Authorized error**, it could be for one of the following reasons:

- You may have configured Tableau Server to use Active Directory with SSPI. If your web server uses SSPI, you do not need to set up trusted authentication. You can embed views and your users will have access to them as long as they are licensed Tableau server users and members of your Active Directory. For more information, see tsm authentication sspi <commands>.

  Note: SSPI can only be configured on Tableau Server for Windows.
- Or a **302- Redirect error** after you have deployed Tableau Server, then it's likely the trusted ticket code you have written to construct the URL for the client has not been updated to account for the two-part ticket URL format. For more information, see Get a Ticket from Tableau Server.

HTTP 404 - File Not Found

You may receive this error if your program code references a Tableau Server URL that does not exist. For example, your web server may construct an invalid URL that cannot be found when the webpage tries to retrieve it.

Another cause for this error is if you did not enter the trusted web servers as specified in Add Trusted IP Addresses or Host Names to Tableau Server. If you have already entered the trusted web servers, verify that the IP addresses or host names are correct.

Invalid User (SharePoint or C#)

You may encounter this error if you've configured Tableau Server for trusted authentication.

The example code for the SharePoint .dll references the following GET request:

```
SPContext.Current.Web.CurrentUser.Name
```

The above request will return the display name of the current Windows Active Directory user. If you want to use the login ID, then you will need to change the code to:

```
SPContext.Current.Web.CurrentUser.LoginName
```

After you make the change, recompile the SharePoint .dll.

Attempting to Redeem the Ticket from the Wrong IP Address

When you've configured Tableau Server for trusted authentication, you may encounter issues with redeeming the ticket.

This issue can occur when Tableau Server is configured to enforce client IP address matching. Because the client web browser IP address is not considered by default when redeeming the ticket, make sure that the client's web browser IP address that is sent in the POST request to Tableau Server is the same as when the browser tries to retrieve the embedded view.

For example, in the Trusted Authentication diagram, if the POST request in step 3 sends the parameter `client_ip=74.125.19.147`, then the GET request in step 5 must come from that same IP address.

For more information, see Optional: Configure Client IP Matching to learn how to configure Tableau Server to enforce client IP address matching.

Cookie Restriction Error

When a user signs in to Tableau Server, a session cookie is stored in their local browser. The stored cookie is how Tableau Server maintains that the signed in user has been authenticated and can access the server. Because the cookie is set with the same domain or sub-domain as

the browser's address bar, it is considered a first-party cookie. If a user's browser is configured to block first-party cookies, they will be unable to sign in to Tableau Server.

When a user signs in to Tableau Server via an embedded view, or in an environment where trusted authentication has been configured, the same thing happens: a cookie is stored. In this case, however, the browser treats the cookie as a third-party cookie. This is because the cookie is set with a domain that's different from the one shown in the browser's address bar. If a user's web browser is set to block third-party cookies, authentication to Tableau Server will fail. To prevent this from occurring, web browsers must be configured to allow third-party cookies.

An error occurred communicating with the server (403)

If Tableau Server is configured for trusted authentication, you may receive this error after opening a new view in a browser and attempting to navigate back to views you'd opened earlier.

Tableau Server provides protection against unauthorized reuse of VizQL sessions through the tsm configuration set Options `vizqlserver.protect_sessions`, which is set to `true` by default. Because Tableau Server is configured for trusted authentication, you might not need this level of protection. To disable this option, you can use tsm configuration set Options to change `vizqlserver.protect_sessions` to `false`. Beginning in 2024.2.0, Tableau Server always prevents VizQL sessions from being reused after the original user signs out.

Personal Access Tokens

Personal access tokens (PATs) provide you and your Tableau Server users the ability to create long-lived authentication tokens. PATs enable you and your users to sign in to Tableau REST API without requiring hard-coded credentials (username and password) or interactive sign-in. For more information about using PATs with Tableau REST API, see Signing In and Out (Authentication) in the Tableau REST API Help.

We recommend creating PATs for automated scripts and tasks that are created with the Tableau REST API:

- **Improve security**: Personal access tokens reduce risk in the event credentials are compromised. In the case where Tableau Server uses Active Directory or LDAP as an identity store, you can reduce the impact of credential compromise by using a personal access token for automated tasks. If a token gets compromised or is used in automation that is failing or posing a risk, you can just revoke the token. You do not need to rotate or revoke the user's credentials.

- **Audit and track**: As an administrator, you can review Tableau Server logs to track when a token is used, what sessions are created from that token, and the actions that are performed in those sessions. You can also determine if a session and the related tasks were performed from a session that was generated from a token or from an interactive signin.

- **Manage automation**: A token can be created for each script or task that is run. This allows you to silo and review automation tasks across your organization. Additionally, by using tokens then password resets or metadata changes (username, email, etc.) on user accounts will not disrupt automation as it would when credentials are hard-coded into the scripts.

Notes:

- To use PATs with tabcmd, install the compatible version of tabcmd from https://tableau.github.io/tabcmd/.
- PATs are not used for generic client access to the Tableau Server web UI or TSM.
- Configuring PATs expiration and disabling or limiting users access to PATs creation from the UI is available in Tableau Cloud only.
- PATs are automatically revoked when a user's authentication method is changed.

Understand personal access tokens

When a personal access token (PAT) is created, it is hashed then stored in the repository. After the PAT is hashed and stored, the PAT secret is shown once to the user and then no longer accessible after the users dismisses the dialog. Therefore, users are instructed to copy the PAT to a safe place and to handle it as they would a password. When the PAT is used at

run-time, Tableau Server compares the PAT presented by the user to the hashed value stored in the repository. If a match is made, then an authenticated session is started.

In the context of authorization, the Tableau Server session that is authenticated with a PAT has the same access and privileges as the PAT owner.

Note: Users can't request concurrent Tableau Server sessions with a PAT. Signing in again with the same PAT, whether at the same site or a different site, will terminate the previous session and result in an authentication error.

Server administrator impersonation

Beginning with version 2021.1, you can enable Tableau Server PAT impersonation. In this scenario, PATs that are created by server administrators can be used for user impersonation when using the Tableau REST API. Impersonation is useful in scenarios where you are embedding end-user-specific Tableau content within your application. Specifically, impersonation PATs allow you to build applications that query as a given user and retrieve content that the user is authorized for within Tableau Server, without hard-coding any credentials.

For more information, see Impersonating a User Tableau REST API Help.

Enable Tableau Server to accept personal access tokens during impersonation sign-in requests

By default, Tableau Server does not allow impersonation for server administrator PATs. You must enable the server-wide setting by running the following commands.

```
tsm authentication pat-impersonation enable [global options]

tsm pending-changes apply
```

Important: After you have run the commands, all PATs created by server administrators (including preexisting PATs) can be used for impersonation. To bulk-revoke all existing server administrator PATs, you can post the `DELETE /api/{api-version}/auth/serverAdminAccessTokens` URI. For more information, see Impersonating a User in the Tableau REST API Help.

Create personal access tokens

Users must create their own PATs. Administrators cannot create PATs for users.

Users with accounts on Tableau Server can create, manage, and revoke personal access tokens (PATs) on the **My Account Settings** page. See ManageYour Account Settings in the Tableau User Help for more information.

Note: A user can have up to 10 PATs.

Change personal access tokens expiry

Personal access tokens (PATs) expire if they are not used after 15 consecutive days. If they are used more frequently than every 15 days, PATs expire after one year. After a year, new PATs must be created. Expired PATs will not display on the **My Account Settings** page.

You can change PATs expiration period by using the refresh_token.absolute_expiry_in_seconds option with the `tsm configuration set` command.

Revoke a personal access token

As an administrator, you can also revoke a user's PAT. A user can also revoke their own personal access tokens (PATs) on the **My Account Settings** page using the procedure described in the Manage Your Account topic in the Tableau User Help.

1. Sign in to Tableau Server as a server or site administrator.
2. Locate the user whose PAT you want to revoke. For more information about navigating Server Admin pages and locating users, see View, Manage, or Remove Users.
3. Click the user's name to open their profile page.
4. On the user's profile page, click the **Settings** tab.
5. In the **Personal Access Tokens** section, identify the PAT that you want to revoke and then click **Revoke**.
6. In the dialog box, click **Delete**.

Track and monitor personal access token usage

All personal access token (PAT)-related actions are logged in the Tableau Server Application Server (vizportal) service. To locate PAT-related activities, filter log entries containing the string, `RefreshTokenService`.

A PAT is stored in this format :`Token Guid: <TokenID(Guid)>`, where the TokenID is a base64 encoded string. The secret value is not included in the logs.

For example:

```
Token Guid: 49P+CxmARY6A2GHxyvHHAA== (e3d3fe0b-1980-458e-80d8-61f1-
caf1c700).
```

The following is an example snippet of two log entries. The first entry shows how a user is mapped to a PAT. The second entry shows a refresh event for the same PAT:

```
RefreshTokenService - Issued refresh token to the following user:
jsmith. Token Guid: 49P+CxmARY6A2GHxyvHHAA== (e3d3fe0b-1980-458e-
80d8-61f1caf1c700)
RefreshTokenService - Redeemed refresh token. Token Guid:
49P+CxmARY6A2GHxyvHHAA== (e3d3fe0b-1980-458e-80d8-61f1caf1c700)
```

To locate key operations, filter log entries containing the string, `OAuthController`.

Use Tableau Connected Apps for Application Integration

Beginning with Tableau Server 2022.1, Tableau connected apps enable a seamless and secure authentication experience by facilitating an explicit trust relationship between your Tableau Server site and external applications where Tableau content is embedded. , Tableau connected apps extended its capabilities to support REST API authorization. And as of October 2023, REST API authorization using connected apps is respected by the Tableau Metadata API.

Note: Tableau connected apps and Salesforce connected apps are different and offer different functionality. Today, Tableau connected apps are optimized for embedding Tableau views and metrics in external applications and used to authorize access to the Tableau REST API. (In

October 2023, Tableau retired the ability to embed metrics in Tableau Cloud and Tableau Server version 2023.3.).

There are two types of connected apps you can configure: direct trust or OAuth 2.0 trust.

Direct trust

Using *direct trust*, you can:

- Restrict access to which content can be embedded and where that content can be embedded
- Provide users the ability to access embedded content using single sign-on (SSO) without having to integrate with an identity provider (IdP)
- Provide users the ability to authenticate directly from your external application
- Programmatically authorize access to the Tableau REST API and Tableau Metadata API (starting in Tableau Server October 2023) on users' behalf using JSON Web Token (JWT)
- Scope Tableau REST API capabilities users or applications can perform
- Enable additional features like:
    - Group assertions (beginning in Tableau Server 2024.2)

For more information about this connected app type, see Configure Connected Apps with Direct Trust.

OAuth 2.0 trust

Using *OAuth 2.0 trust*, you can:

- Restrict access to which content can be embedded and where that content can be embedded

- Provide users the ability to access embedded content using single sign-on (SSO) through your identity provider (IdP)
- Provide access using standard OAuth 2.0 standard protocol
- Programmatically authorize access to Tableau REST API (and the Metadata API starting in Tableau Server 2023) on users' behalf
- Scope Tableau REST API capabilities users or applications can perform
- Enable additional features like:
  - Group assertions (beginning in Tableau Server 2024.2)

For more information about this connected app type, see Configure Connected Apps with OAuth 2.0 Trust.

Configure Connected Apps with Direct Trust

Beginning with Tableau Server version 2022.1, Tableau connected apps enable a seamless and secure authentication experience by facilitating an explicit trust relationship between your Tableau Server site and external applications.

Notes:

- Connected apps functionality, without UI, for Tableau Server became available in Tableau Server version 2021.4 through the Connected App methods in the Tableau REST API.
- To enable embedding through connected apps, Tableau Server must be configured to use SSL for HTTP traffic.
- In order for the session token to be valid, the clocks of the external application and the server that hosts the external application must be set to Coordinated Universal Time (UTC). If either clock uses a different standard, the connected app will not be trusted.

How Tableau connected apps work with direct trust

The trust relationship between your Tableau Server site and external application is established and verified through an authentication token in the JSON Web Token (JWT) standard, which

uses a shared secret provided by the Tableau connected app and signed by your external application.

Key components of a connected app

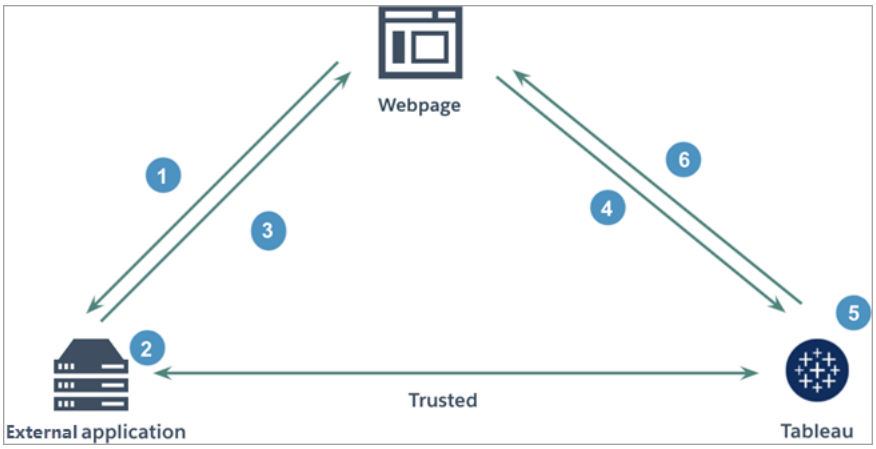
The following components of the connected work together with the JWT in your external application to authenticate users and display embedded content.

- **Secrets:** Secrets are keys shared by Tableau and your external application. They are used in signatures that form the JWT. A secret is required when using connected apps for embedding authentication or REST API authorization. Secrets can be created in a connected app, do not expire, and remain valid until deleted.
- **Domain allowlist (embedding workflows only):** You can specify a list of allowed domains in each connected app. Tableau content embedded through a connected app is only allowed under the specified domains. This helps ensure that content is exposed under the domains that are secured and approved by your business.
- **Access level (embedding workflows only):** You can specify what content can be embedded through a connected app by associating a connected app with one project or all projects. If you specify one project, only the content in the selected project can be embedded through the connected app. If you want to specify multiple projects, you must use the Tableau REST API.

Connected app workflow

Embedding workflows

The diagram below illustrates how authentication works between your external application (web server and webpage) and Tableau connected app.



1. **User visits the webpage:** When a user visits the embedded content on a webpage, webpage sends a GET request to your external application to retrieve the HTML on that webpage.
2. **External application constructs an authentication token:** The external application constructs a JWT, which contains a secret from the connected app (see Step 3 below for additional JWT requirements) and the scope of user access for the embedded content. The secret is signed by the external application and is used for verification of the trust relationship in a later step.
3. **External application responds with authentication token:** The external application responds to the page with the JWT in the embedded content's URL called by the webpage.
4. **Webpage requests content from Tableau**: With the attempt to load the embedded content, the webpage calls the embedded content's URL, which sends a GET request to Tableau.
5. **Tableau validates the token:** Tableau receives the JWT and verifies the trust relationship with the external application by identifying the connected app and shared secret used in the JWT. Then Tableau creates a session for the user. The session not only respects the embedding scopes defined in the JWT, but also the restrictions specified in the connected app, including the allowed domains and allowed projects.
6. **Tableau returns the content based on the restricted embedding context:** The embedded content only loads when the page is under an allowed domain and the content is published to an allowed project (if applicable). The authenticated user can only interact with the embedded content by the scope defined in the JWT.

Step 1: Create a connected app

Create a connected app from Tableau Server's Settings page.

1. As a server admin, sign in to Tableau Server.
2. From the left pane, select **Settings** > **Connected Apps**.
3. Click the New Connected App button drop-down arrow and select **Direct Trust**.

   Note: If you're using Tableau Server 2023.3 or earlier, click **New Connected App** button.
4. In the Create Connected App dialog box, do *one* of following:

   - For *REST API authorization workflows (including Metadata API workflows that use the REST API for authentication)*, in the Connected app name text box, enter a name for the connected app and click the **Create** button.

     Note: You can ignore **Access level** and **Domain allowlist** when configuring a connected app for REST API and Metadata API authorization.

   - For *embedding workflows*, do the following:

     i. In the Connected app name text box, enter a name for the connected app.

     ii. From the Applies to drop-down menu, select **All project** or **Only one project** to control which views or metrics can be embedded. If you select the "Only one project" option, select the specific project to scope to. For more information about these two options, see Access level (embedding workflows only).

     Notes:

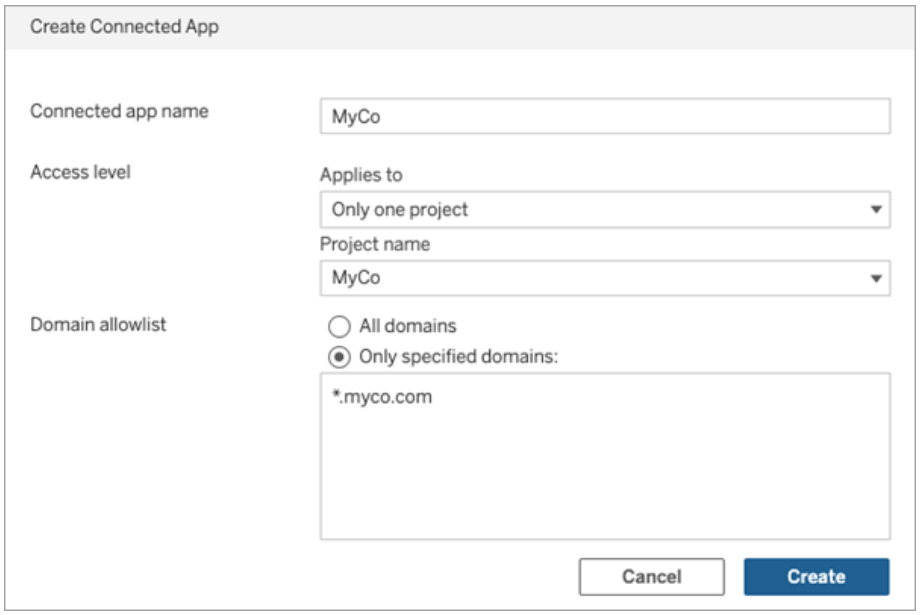
     - In Tableau Server 2023.3, Tableau retired the ability to embed metrics.

- Beginning with Tableau Server 2024.2, you can specify multiple projects using the Tableau REST API. For more information, see Create Connected App and Update Connected App methods in the REST API Help.

iii. In the Domain allowlist, specify the domains using the rules described in Domain formatting below to control where views or metrics can be embedded.

Important: We recommend using the domain allowlist as a security best practice to ensure Tableau content is only embedded in locations that you allow.

iv. When finished, click the **Create** button.



5. Next to the connected app's name, click the actions menu and select **Enable**. For security purposes, a connected app is set to disabled by default when created.

6. Make note of the connected app's ID, also known as the client ID, to use in Step 3 below.

Step 2: Generate a secret

You can generate a total of two secrets for each connected app. The second secret can be used for secret rotation purposes to help protect against issues if a secret is compromised.

1. On the detail page of the connected app you created in Step 1, click the **Generate New Secret** button.
2. Make note of the secret ID and secret value to use in Step 3 below.

Step 3: Configure the JWT

After you've generated a secret, you want to enable your external application to send a valid JWT. JWT is a standard used to securely transfer information between two parties. The JWT is signed by your external application to securely send information to Tableau Server. The

JWT references the connected app, the user that the session is being generated for, and the level of access the user should have.

A valid JWT includes the following information:

- Connected app ID, also known as the client ID, from Step 1
- Secret ID and secret value generated in Step 2
- Registered claims and header:

Claim	Name	Description or required value
"kid"	Secret ID	Required (in header). The connected app's secret key identifier.
"iss"	Issuer	Required (in header). Unique issuer URI that identifies the trusted connect app and its signing key.
"alg"	Algorithm	Required (in header). JWT signing algorithm. Only HS256 is supported.
"sub"	Subject	User name of the authenticated Tableau Server user.
"aud"	Audience	Value must be: "tableau".
"exp"	Expiration Time	A valid JWT must not be expired. The expiration time (in UTC) of the JWT must be within the configured maximum validity period. The maximum validity period can be configured using the tsm viz-

		portal.oauth.connected_apps.max_expiration_period_in_minutes command.
"jti"	JWT ID	Required as a claim. The JWT ID claim provides a unique identifier for the JWT and is case sensitive.
"scp" **Important:** Do not use "scope".	Scope	For *embedding workflows*, supported values include: "tableau:views:embed" "tableau:views:embed_authoring"Added in Tableau Server 2022.3 "tableau:metrics:embed" (Retired in Tableau Server 2023.3) "tableau:ask_data:embed" (Added in Tableau Server 2023.1) **Notes:** • Values must be passed as a list type. • For tableau-:views:embed and tableau-:views:embed_authoring, the scope respects users' permissions already configured in Tableau Server and allows

		users to interact with the tools in the embedded view if available in the original view. • We recommend the embed code exclude the toolbar parameter. For more information see Known issues (embedding workflows only) below. For *REST API authorization workflows*, see REST API methods that support JWT authorization. For *Metadata API workflows that use the REST API for authentication*, the only supported scope is `tableau-:content:read`.
`https://tableau.com/groups`		For *embedding workflows* only. Value must match the name of one or more groups in Tableau Server. For more information, see the Dynamic group membership (embedding workflows only) section below.

Example JWTs

Here are example JWTs in both Java and Python languages. The Java and Python examples use the nimbus-jose-jwt library and the PyJWT library, respectively.

Java

```
import com.nimbusds.jose.*;
import com.nimbusds.jose.crypto.*;
import com.nimbusds.jwt.*;

import java.util.*;

...

String secret = "secretvalue";
       String kid = "connectedAppSecretId";
       String clientId = "connectedAppClientId";
       List<String> scopes = new
ArrayList<>(Arrays.asList("tableau:views:embed"));
       String username = "username";
       JWSSigner signer = new MACSigner(secret);
       JWSHeader header = new
JWSHeader.Builder(JWSAlgorithm.HS256).keyID(kid).customParam("iss",
clientId).build();
       JWTClaimsSet claimsSet = new JWTClaimsSet.Builder()
               .issuer(clientId)
               .expirationTime(new Date(new Date().getTime() + 60 * 1000)) //ex-
pires in 1 minute
               .jwtID(UUID.randomUUID().toString())
               .audience("tableau")
               .subject("username")
               .claim("scp", scopes)


               .build();
       SignedJWT signedJWT = new SignedJWT(header, claimsSet);
```

```
        signedJWT.sign(signer);
        model.addAttribute("token", signedJWT.serialize());
```

Python

```
import jwt

token = jwt.encode(
        {
                "iss": connectedAppClientId,
                "exp": datetime.datetime.utcnow() + datetime.timedelta(minutes=5),
                "jti": str(uuid.uuid4()),
                "aud": "tableau",
                "sub": user,
                "scp": ["tableau:views:embed", "tableau:metrics:embed"]

        },
                connectedAppSecretKey,
                algorithm = "HS256",
                headers = {
                'kid': connectedAppSecretId,
                'iss': connectedAppClientId
         }
  )
```

After you've configured the JWT, when the code is run by your external application, it will generate a token.

Step 4: Next steps

For embedding workflows

After the JWT has been configured, you must add embed code to your external application. Ensure that you include the valid JWT you configured in Step 3 above in the web component that your external application calls.

For more information about embedding Tableau content, see one or both of the following:

- Embed metrics, see Embed Metrics into Webpages topic in the Tableau Help. (In Tableau Server 2023.3, Tableau retired the ability to embed metrics.)
- Embed Tableau views and metrics using the Tableau Embedding API v3.

Note: For users to successfully authenticate when they access embedded content, browsers must be configured to allow third-party cookies.

For REST API authorization workflows

After the JWT has been configured, you must add the valid JWT to the REST API Sign In request for authorized access. For more information, see Access Scopes for Connected Apps.

For Metadata API workflows

After the JWT has been configured, you must add the valid JWT to the REST API Sign In request. For more information, see Access Scopes for Connected Apps.

Manage a connected app

The Connected Apps page is where you can manage all the connected apps for your site. You can perform tasks such creating, deleting, and disabling connected apps; and revoking or generating new secrets if existing secrets have been compromised.

1. As a server admin, sign in to Tableau Server.
2. From the left pane, select **Settings** > **Connected Apps**.
3. Select the check box next to the connected app you want to manage and do one or more of the following:
    - **Generate a new secret** according to the rotation timeline specified by your organization's security policies. To generate an additional secret, click on the name of the connected app and then click the **Generate New Secret** button. A connected app can have a maximum of two secrets. Both secrets can be active at the same time, do not expire, and remain valid until deleted.

- **Review the connected app details** by clicking the name of the connected app to see when the connected app was created, its ID, project and domain scopes, and its secrets.

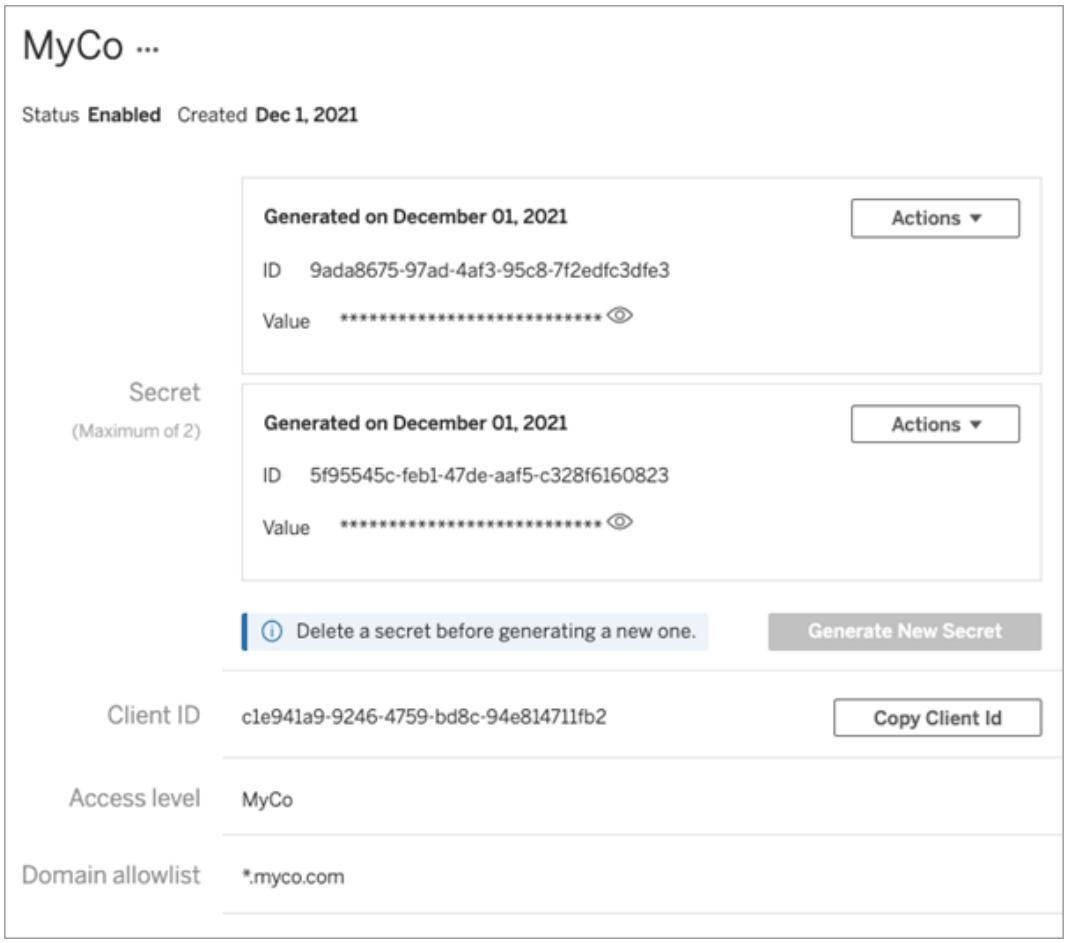
- **Change the project scope or domain**, in the Actions menu, select **Edit**. Make your changes and click **Update**.

  Note: If you change the project or domain scopes and the embedded content doesn't exist in either the new project or new domain, the embedded view or metric is unable to display and users will see an error when accessing the embedded content.

- **Delete a secret** by clicking the connected app's name. On the connected app's page, click **Actions** next to the secret and select **Delete**. In the confirmation dialog box, select **Delete** again.

  Note: If the connected app's secret is being used by an external application, the embedded view or metric is unable to display after the secret is deleted. For more information, see Effects of disabling or deleting a connected app, or deleting a secret below.

- **Disable a connected app**, in the Actions menu, select **Disable**. If the connected app is being used by an external application, the embedded view or metric is unable to display after the connected app is disabled. For more information, see Effects of disabling or deleting a connected app, or deleting a secret below.



Effects of disabling or deleting a connected app, or deleting a secret

To display embedded content to your user or enable REST API access through a connected app, the connected app must be enabled and its secret generated. If the connect app is being used in your external application and is either disabled or deleted, or its secret deleted or replaced, users will get a 403 error.

To avoid this issue, ensure the connect app is enabled and the JWT is using the correct secret ID and value.

Access level (embedding workflows only)

You can select one of two project types when configuring a connected app's access level. The access level controls which content can be embedded.

- **All projects:** This option enables the content in all projects to be embedded
- **Only one project:** This option enables only the content in the specified project to be embedded. If the specified project contains nested projects, embedding content in those nested projects is not enabled.

About multiple projects

Starting in Tableau Server 2024.2, you can enable the content in multiple projects for a connected app using the Tableau REST API only. To specify which projects, use the "project IDs" in either the Create a Connected App or Update a Connected App methods.

Note: When multiple projects are configured for your connected app, Tableau displays **Multiple projects** for the connected app's access level. If you select either **Only one project** or **All projects** and update the connected app, the "Multiple projects" option will no longer be visible. If you need to configure the connected app for multiple projects again, you must use the REST API.

Domain allowlist rules (embedding workflows only)

The connected app’s domain allowlist enables you to restrict access to embedded Tableau content to all domains or some domains; or exclude some domains or block all domains.

Important: We recommend using the domain allowlist as a security best practice to ensure Tableau content is only embedded in locations that you allow.

Domain options

You can select one of two options when configuring a connected app’s domain allowlist:

- **All domains:** As the default option, this option enables unrestricted access to embedded content.

- **Only specific domains:** This option gives you the ability to scope down access to embedded content. If you use this option, follow the formatting rules specified in the following section, Domain formatting.

Domain formatting

In the domain allowlist text box, you can enter one or more domains using the formatting examples below.

Note: Domain formatting rules also apply when using the Connect App methods in the Tableau REST API.

Here are some formatting examples based on common scenarios:

To specify...	Example	Embedding access
Range of domains	*.myco.com	Embedded content is accessible from all subdomains under myco.com.
All ports	myco.com:*	Embedded content is access from all ports in myco.com.
Specific port	myco.com:8080	Embedded content is accessible from port 8080 in myco.com only.
Multiple discrete domains	myco.com events.myco.com ops.myco.com	Embedded content is accessible from all three domains. **Note:** When specifying multiple domains, type each domain on a new line or separate domains with a space. For the REST API, domains must be separated by a space.
Secure traffic only	https:	Embedded content is securely accessible regardless of domain.
Secure traffic to all ports for	https:*myco.com:*	Embedded content is securely accessible from all ports on all subdomains under

a range of domains		myco.com.
No domains	[no domains]	Access to embedded content is blocked.

Dynamic group membership (embedding workflows only)

Beginning in Tableau Server 2024.2, if connected apps are configured and the capability's setting is enabled, you can dynamically control group membership through custom claims included in the JWT sent by the external application.

When configured, during user authentication, the external application sends the JWT that contains two custom claims for group membership: group (`https://tableau.com/groups`) and group names (for example, "Group1" and "Group2") to assert the user into. Tableau validates the JWT and then enables access to the groups and the content whose permissions are dependent on those groups.

For more information, see Dynamic group membership using assertions.

Known issues (embedding workflows only)

There are a couple of known issues when using connected apps that will be addressed in a future release.

- **Toolbar features:** When embedded content has the toolbar parameter defined, not all toolbar features will work. To work around this issue, we recommend you hide the toolbar parameter like in the example below.

```
<tableau-viz id='tab-viz' src='https://<your_server>/t/<your_
site>/...'
       toolbar='hidden'>
</tableau-viz>
```

- **Published data sources:** Published data sources set to **Prompt User** for database credentials will not display. To work around this issue, if possible, we recommend data source owners embed their database credentials instead.
- **Embedded views on multiple sites:** In Tableau Server 2023.1 and earlier, switching between views on different sites in the same browser causes error **1008: Could not fetch secret for connect app**. To work around this issue, upgrade to Tableau Server 2023.3 or later.
- **Ask Data objects in embedded dashboards:** Ask Data objects in embedded dashboards will not load. (In Tableau Server 2024.2 will retire Ask Data.)
- **Metrics and domain allowlists:** Embedded metrics views will display despite access restrictions that might be specified in the connected apps' domain allowlists. **Note:** Metrics data accessed from toolbars of embedded views will work as expected. (In Tableau Server 2023.3, Tableau retired the ability to embed metrics.)

Troubleshoot

You can refer to Troubleshoot Connected Apps - Direct Trust for errors that might be associated with the connected app and suggested troubleshooting steps.

Configure Connected Apps with OAuth 2.0 Trust

As a Tableau Server admin, you can register one or more external authorization servers (EASs) to establish a trust relationship between your Tableau Server and the EAS using the OAuth 2.0 standard protocol.

Important:

- Some of the procedures in this topic require configuration with third party software and services. We've made a best effort to verify the procedures to enable the EAS feature on Tableau Server. However, third-party software and services might change or your organization might differ. If you encounter issues, refer to your third-party documentation for authoritative configuration details and support.
- To enable embedding through EAS, Tableau Server must be configured to use SSL for HTTP traffic.

- In order for the session token to be valid, the clocks of the external application and the server that hosts the external application must be set to Coordinated Universal Time (UTC). If either clock uses a different standard, the connected app will not be trusted.

How Tableau connected apps work with OAuth 2.0 trust

The trust relationship between your Tableau Server site and external application is established and verified through an authentication token in the JSON Web Token (JWT) standard.

When embedded Tableau content is loaded in your external application, Authorization Code Flow, OAuth flow is used. After users successfully sign in to the IdP, they are then automatically signed in to Tableau Server. Follow the steps described below to register your EAS with Tableau Server .

Key components of a connected app

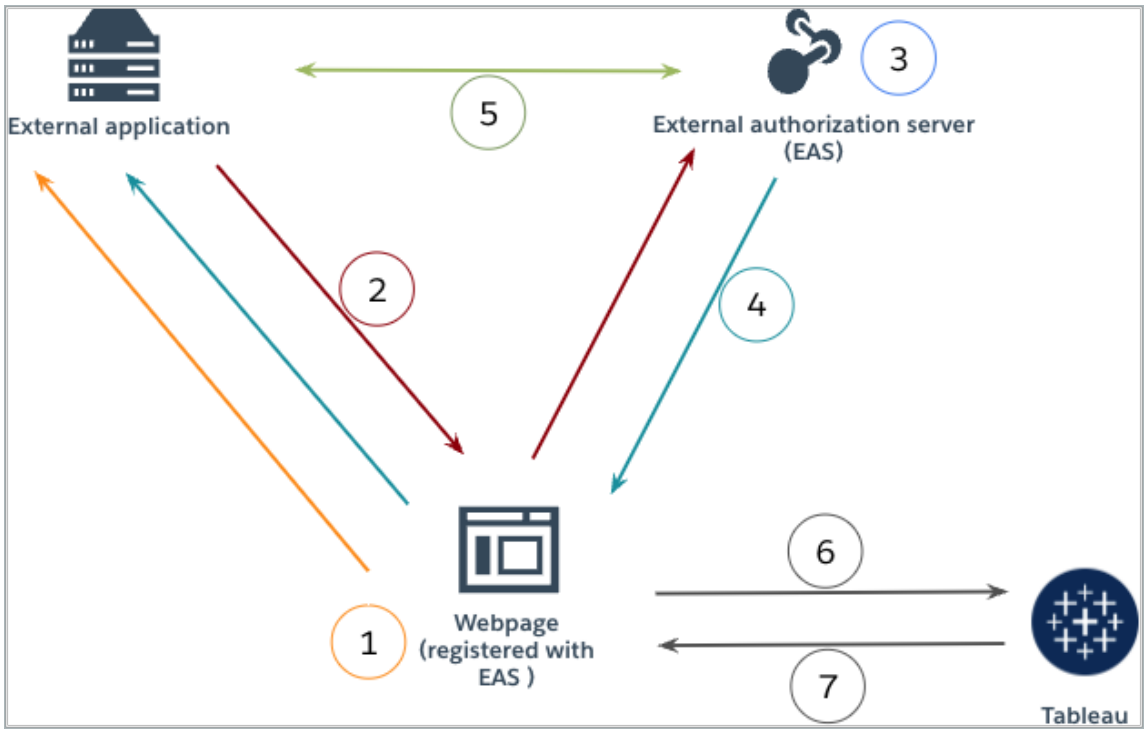
The following components of the connected work together with the JWT in your external application to authenticate users and display embedded content.

- **External authorization server (EAS):** The server, typically your IdP, that functions as the interface between the user and the external application. The server authenticates and authorizes user access to the protected Tableau content.
- **Issuer URL:** The URL that uniquely identifies the EAS instance.

Connected app workflow

Embedding workflows

The diagram below illustrates how authentication works between your external authorization server (EAS), external application (web server and webpage), and Tableau connected app.



1. **User visits the webpage:** When a user visits the embedded content on a webpage, the webpage sends a GET request to the external application.
2. **External application redirects request to EAS:** External application responds with a webpage that redirects to the external authorization server (EAS).
3. **User authenticates with EAS:** User authenticates and authorizes with the EAS.
4. **EAS responds to webpage with authorization code:** The EAS responds to the page with an authorization code and redirects back to the webpage.
5. **EAS converts authorization code to JWT:** Webpage calls the EAS to convert the authorization code to a JWT, which the webpage puts into the embedded content's URL.
6. **Webpage requests content from Tableau:** Webpage loads the iFrame and sends a GET request to Tableau.

7. **Tableau validates the token:** Tableau validates the JWT in the URL with the signature and responds with the content and respects the embedding scopes defined in the JWT.

Create a connected app

Step 1: Before you begin

To register an EAS with Tableau Server , you must have an EAS already configured. In addition, the EAS must send a valid JSON Web Token (JWT) that contains the registered claims and header listed in the table below.

Claim	Name	Description or required value
"kid"	Key ID	Required (in header). A unique key identifier from the identity provider.
"iss"	Issuer	Required (in header or as a claim). Unique issuer URIthat identifies the trusted connect app and its signing key.
"alg"	Algorithm	Required (in header). JWT signing algorithm. Supported algorithm names are listed in the Class JWSAlgorithm page in the javadoc.io documentation. The signing algorithm can be configured using the vizportal.oauth.external_authorization_server.blocklisted_jws_algorithms command.
"sub"	Subject	User name of the authenticated Tableau Server user.

"aud"	Audience	Value must be: "tableau"
"exp"	Expiration Time	A valid JWT must not be expired. The expiration time (in UTC) of the JWT must be within the configured maximum validity period. Maximum validity period can be configured using vizportal.oauth.external_authorization_server.max_expiration_period_in_minutes command.
"jti"	JWT ID	The JWT ID claim provides a unique identifier for the JWT and is case sensitive.
"scp"	Scope	For *embedding workflows*, supported values include: "tableau:views:embed" "tableau:views:embed_authoring" (Added in Tableau Server 2022.3) "tableau:metrics:embed" (Retired in Tableau Server 2023.3) "tableau:ask_data:embed"(Added in Tableau Server 2023.1. Will be retired in Tableau Server 2024.2) **Notes:** • Values must be passed

		as a list type. • For `tableau-:views:embed`, the scope respects users' permissions already configured in Tableau Server and allows users to interact with the tools in the embedded view if available in the original view. • We recommend the embed code exclude the toolbar parameter. For more information see Known issues (embedding workflows only) below. For *REST API authorization workflows*, see REST API methods that support JWT authorization. For *Metadata API workflows that use the REST API for authentication*, the only supported scope is `tableau-:content:read`.
`https://tableau.com/groups`	Dynamic group membership	For *embedding workflows* only. Value must match the name of one or more groups in Tableau

		Server. For more information, see Dynamic group membership (embedding workflows only) section below.

Note: The JWT claims above are documented in the Registered Claim Names section in the documentation distributed by the Internet Engineering Task Force (IETF) organization.

Step 2: Register your EAS with Tableau Server

By registering your EAS with Tableau Server, you establish a trust relationship between the EAS and Tableau Server. This means when users access Tableau content embedded in your external application, they are redirected to authenticate with the IdP. The EAS generates the authentication token, which is passed to Tableau Server for verification. After the trust relationship is verified, access to the embedded content is granted users.

Note: Some EAS support the option to display a consent dialog that asks for users' approval for the application to access Tableau content. To ensure the best experience for your users, we recommend you configure your EAS to automatically consent to the external application's request on users' behalf.

About site-level EAS

Beginning in Tableau Server 2024.2, you can configure site-level EAS. To register an EAS at the site-level, connected apps must be enabled in Tableau Server Manager (TSM).

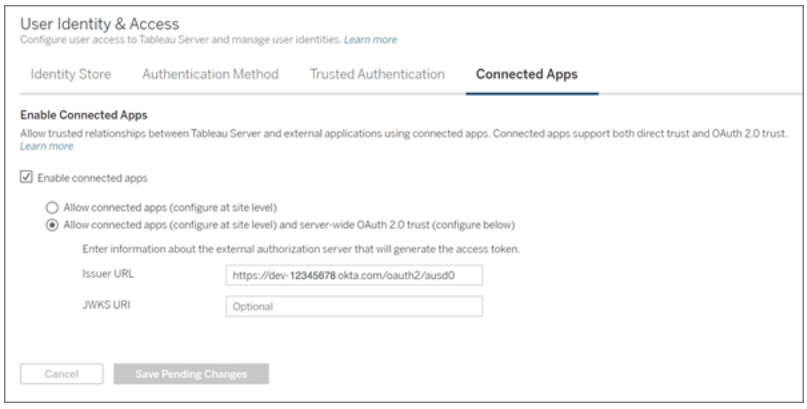
Server-wide EAS

There are two ways you can register server-wide EAS: using the TSM web UI or using TSM CLI.

After registering the EAS, the established trust relationship applies to all sites on Tableau Server.

Option 1: Using TSM web UI

1. As a Tableau Server admin, sign in to the Tableau Services Manager (TSM) web UI. For more information, see Sign in to Tableau Services Manager Web UI.
2. Do *one* of the following:
   - In Tableau Server 2024.2 and later, navigate to User Identity & Access page > **Connected Apps** tab.
   - In Tableau Server 2023.3 and earlier, navigate to User Identity & Access page > **Authorization Server** tab.
3. Do *one* of the following:
   - In Tableau Server 2024.2 and later:
     a. Select the **Enable connected apps** check box.
     b. Select the second radio button, **Allow connected apps (configure at site level) and server-wide OAuth 2.0 trust (configure below)**.
     c. In the **Issuer URL** text box, paste the issuer URL of the EAS.
     d. Click the **Save Pending Changes** button.

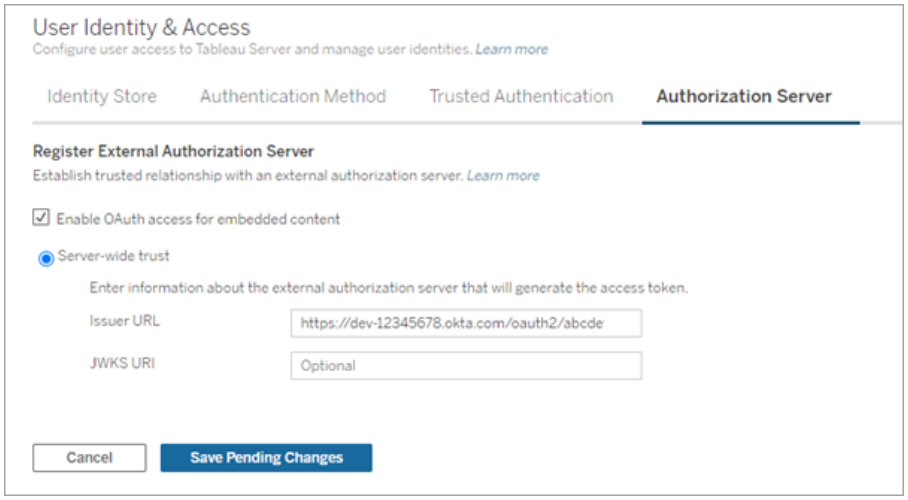


   - In Tableau Server 2023.3 and earlier:

a. Select the **Enable OAuth access for embedded content** check box.

b. In the **Issuer URL** text box, paste the issuer URL of the EAS.

c. Click the **Save Pending Changes** button.

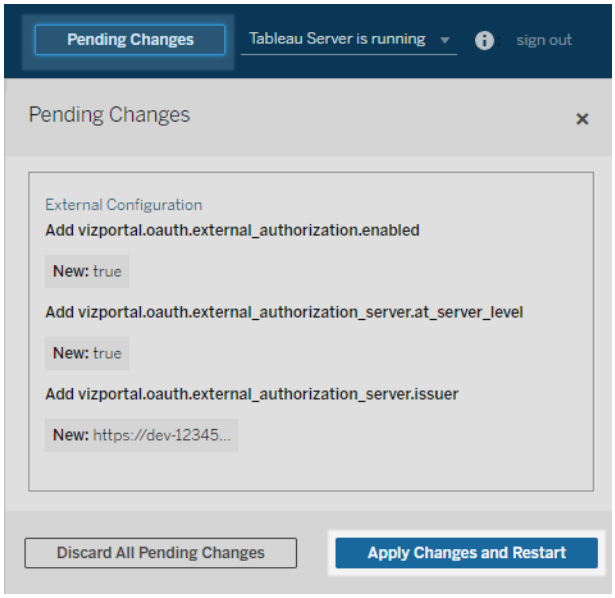


4. When finished, do the following:

a. In the upper-right corner of the page, click the **Pending Changes** button.

b. In the bottom-right corner of the page, click the **Apply Changes and Restart**

button to stop and restart Tableau Server.



Option 2: Using TSM CLI

1. Open a command prompt as an admin on the initial node (where TSM is installed) in the cluster.
2. Run the following commands:

```
tsm configuration set -k vizportal.oauth.external_author-
ization.enabled -v true
tsm configuration set -k vizportal.oauth.external_author-
ization_server.issuer -v "<issuer_url_of_EAS>"
tsm restart
```

Site-level EAS

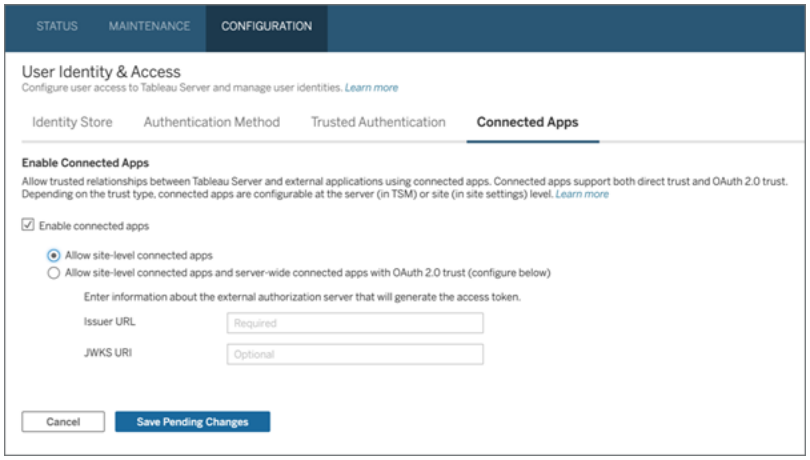
Beginning in Tableau Server 2024.2, you can register one or more EASs for a site. After registering the EAS at the site level, the established trust relationship applies to the site only.

Note: A prerequisite to configuring site-level EAS is that connected apps is enabled in TSM.

Step 1: Enable connected apps

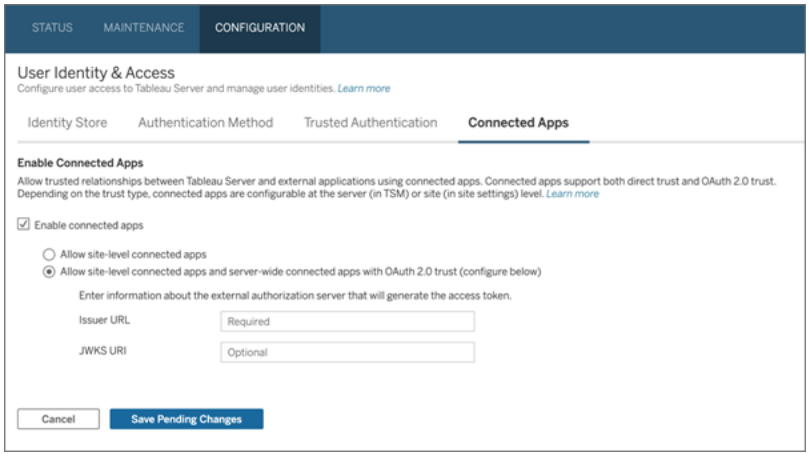
1. As a Tableau Server admin, sign in to the Tableau Services Manager (TSM) web UI. For more information, see Sign in to Tableau Services Manager Web UI.
2. Navigate to User Identity & Access page > **Connected Apps** tab.
3. Select the **Enable connected apps** check box.
4. Do *one* of the following:

    - Select the first radio button, **Allow connected apps (configure at site level)** to enable registering EASs at the site-level only.



    - (Default) Select the second radio button, **Allow connected apps (configure at site level) and server-wide OAuth 2.0 trust (configure below)** to enable registering EASs at both the site-level and server-wide. If you choose this option, ensure the issuer URL specified at the site-level is different from the server-wide

issuer URL.



5. Click the **Save Pending Changes** button.
6. When finished, do the following:
    a. In the upper-right corner of the page, click the **Pending Changes** button.
    b. In the bottom-right corner of the page, click the **Apply Changes and Restart** button to stop and restart Tableau Server.

Step 2: Register the EAS

1. As a Tableau Server admin, sign in to Tableau Server.
2. From the left pane, select **Settings** > **Connected Apps**.
3. Click the New Connected App button drop-down arrow and select **OAuth 2.0 Trust**.
4. In the Create Connected App dialog box, do the following:
    a. In the **Name** text box, enter a name for the connected app.
    b. In the **Issuer URL** text box, paste the issuer URL of the EAS.

c. Select the **Enable connected app**. For security purposes, a connected app is set to disabled by default when created.

d. When finished, click the **Create** button.

5. After the connected app is created, copy the connected app's site ID. The site ID is used for the JWT's "aud" (Audience) claim described in Step 1 above.

Step 3: Next steps

For embedding workflows

After configuring Tableau Server to use your EAS, you must add embed code to your external application. Ensure that you include the valid JWT generated by your EAS, as described in Step 1, in the web component that your external application calls.

For more information about embedding Tableau content, see one or both of the following:

- Embed metrics, see Embed Metrics into Webpages topic in the Tableau Help. (In Tableau Server 2023.3, Tableau retired the ability to embed metrics.)

- Embed Tableau views and metrics using the Tableau Embedding API v3.

Note: For users to successfully authenticate when they access embedded content, browsers must be configured to allow third-party cookies.

Control where content can be embedded using domain allowlist for embedding

Starting in Tableau Server 2023.3, you and your users can control whether Tableau content can be embedded without restriction or restricted to certain domains using the Update Embedding Settings for Site method in Tableau REST API.

By default, the `unrestrictedEmbedding` site setting for embedding is set to `true` to allow unrestricted embedding. Alternatively, you and your users can set the setting to `false` and specify the domains where Tableau content in external applications can be embedded using the `allowList` parameter.

For more information, see one or both of the following:

- Update Embedding Settings for Site in the Tableau REST API Help
- Tableau Site Setting for Embedding in the Tableau Embedding API v3 Help.

For REST API authorization workflows

After the JWT has been configured, you must add the valid JWT to the REST API Sign In request for authorized access. For more information, see Access Scopes for Connected Apps.

For Metadata API workflows

After the JWT has been configured, you must add the valid JWT to the REST API Sign In request. For more information, see Access Scopes for Connected Apps.

Manage a connected app

Dynamic group membership (embedding workflows only)

Beginning in Tableau Server 2024.2, if connected apps are configured and the capability's setting is enabled, you can dynamically control group membership through custom claims

included in the JWT sent by the external application.

When configured, during user authentication, the external application sends the JWT that contains two custom claims for group membership: group (`https://tableau.com/groups`) and group names (for example, "Group1" and "Group2") to assert the user into. Tableau validates the JWT and then enables access to the groups and the content whose permissions are dependent on those groups.

For more information, see Dynamic group membership using assertions.

Known issues (embedding workflows only)

There are a couple of known issues when using connected apps that will be addressed in a future release.

- **Toolbar features:** When embedded content has the toolbar parameter defined, not all toolbar features will work. To work around this issue, we recommend you hide the toolbar parameter like in the example below.

  ```
  <tableau-viz id='tab-viz' src='https://<your_server>/t/<your_
  site>/...'
         toolbar='hidden'>
  </tableau-viz>
  ```

- **Published data sources:** Published data sources set to **Prompt User** for database credentials will not display. To work around this issue, if possible, we recommend data source owners embed their database credentials instead.
- **Embedded views on multiple sites:** In Tableau Server 2023.1 and earlier, switching between views on different sites in the same browser causes error **1008: Could not fetch secret for connect app**. To work around this issue, upgrade to Tableau Server 2023.3 or later.

Troubleshoot

When embedded content fails to display in your external application or Tableau REST API authorization fails, you can use a browser's developer tools to inspect and identify error codes that might be associated with the EAS feature enabled on Tableau Server .

Refer to the table below to review the description of the error code and potential resolution.

Error code	Summary	Description	Potential resolution or explanation
5	SYSTEM_USER_NOT_FOUND	Tableau user could not be found	To resolve this issue, verify the '`sub`' (Subject) claim value in the JWT is "username" for the authenticated Tableau Server. This value is case sensitive.
16	LOGIN_FAILED	Login failed	This error is typically caused by one of the following claim issues in the JWT: • The '`exp`' (Expiration Time) exceeds the default maximum validity period. To resolve this issue, review registered claims required for a valid JWT and ensure the correct value is used. To change the maximum validity period, you can use the viz-portal.oauth.external_authorization_server.max_expiration_period_in_minutes command.

			• The '`sub`' (Subject) is calling an unknown user. To resolve this issue, verify the '`sub`' claim is "username". This value is case sensitive.
67	FEATURE_NOT_ENABLED	On-demand access is not supported	On-demand access is available through licensed Tableau Cloud sites only.
10081	COULD_NOT_RETRIEVE_IDP_METADATA	Missing EAS metadata endpoint	To resolve this issue, verify the EAS is configured correctly and the correct issuer is being called.
10082	AUTHORIZATION_SERVER_ISSUER_NOT_SPECIFIED	Missing issuer	To resolve this issue, verify the correct issuer is being called. To change the issuer URL, you can use the vizportal.oauth.external_authorization_server.issuer command.
10083	BAD_JWT	JWT header contains issues	The '`kid`' (Secret ID) or '`clientId`' (Issuer) claims are missing from the JWT header. To resolve this issue, ensure this information is included.
10084	JWT_PARSE_ERROR	JWT contains issues	To resolve this issue, verify the following: • The '`aud`' (Audience) value referenced in the JWT uses the "tableau" value. This value is case sensitive.

			• The 'aud' (Audience) and 'sub' (Subject) are included in the JWT.
10085	COULD_NOT_ FETCH_JWT_ KEYS	JWT could not find keys	Could not find the secret. To resolve this issue, verify the correct issuer is being called. To change the issuer URL, you can use the viz-portal.oauth.external_authorization_server.issuer command.
10087	BLOCKLISTED_ JWS_ ALGORITHM_ USED_TO_SIGN	Issue with the JWT signing algorithm	To resolve the issue, you can remove the signing algorithm. For more information, see viz-portal.oauth.external_authorization_server.blocklisted_jws_algorithms.
10088	RSA_KEY_SIZE_ INVALID	Issue with JWT signing requirements	To resolve this issue, verify with the EAS or IdP the JWT is being signed with an RSA key size of 2048.
10091	JTI_ALREADY_ USED	Unique JWT required	The JWT has already been used in the authentication process. To resolve this issue, the EAS or IdP must generate a new JWT.
10092	NOT_IN_DOMAIN_ ALLOW_LIST	Domain of the embedded content is not specified	To resolve this issue, ensure the unrestrictedEmbedding setting is set to true or domainAllowlist parameter includes

			the domains where Tableau content is embedded using the Update Embedding Settings for Site method in the Tableau REST API.
10094	MISSING_ REQUIRED_JTI	Missing JWT ID	To resolve this issue, verify the '`jti`' (JWT ID) is included in the JWT.
10096	JWT_ EXPIRATION_ EXCEEDS_ CONFIGURED_ EXPIRATION_ PERIOD		The '`exp`' (Expiration Time) exceeds the default maximum validity period. To resolve this issue, review registered claims required for a valid JWT and ensure the correct value is used. To change the maximum validity period, you can use the vizportal.oauth.external_author-ization_server.max_expiration_ period_in_minutes command.
10097	SCOPES_ MALFORMED	Issues with scopes claim	This error can occur when the '`scp`' (Scope) claim is either missing from the JWT or not passed as a list type. To resolve this issue, verify '`scp`' is included in the JWT and passed as a list type. For troubleshooting help with a JWT, see Debugger on the auth0 site.
10098	JWT_UNSIGNED_ OR_ENCRYPTED	JWT is unsigned or encrypted	Tableau does not support an unsigned or encrypted JWT.

10099	SCOPES_ MISSING_IN_JWT	Missing scopes claim	The JWT is missing the required '`scp`' (Scope) claim. To resolve this issue, verify '`scp`' is included in the JWT. For troubleshooting help with a JWT, see Debugger on the auth0 site.
10100	JTI_ PERSISTENCE_ FAILED	Unexpected JWT ID error	There was an unexpected error with the '`jti`' (JWT ID). To resolve this issue, a new JWT with a new '`jti`' must be generated.
10103	JWT_MAX_SIZE_ EXCEEDED	JWT exceeds maximum size	This error can occur when JWT size exceeds 8000 bytes. To resolve this issue, make sure that only the necessary claims are being passed to Tableau Server.

Access Scopes for Connected Apps

Starting with Tableau Server version 2022.3, using Tableau connected apps, you can programmatically call and access the Tableau REST API through your custom application on behalf of Tableau Server users. Access to the REST API is enabled by a JSON Web Token (JWT) defined as part of the initial Sign In request. The JWT must contain scopes that define the REST API methods that are available to your custom application and its users through the connected app.

Authorize access to the REST API using connected apps to:

- Enhance security—using a JWT as a bearer token is inherently more secure than storing and managing admin user passwords through .env files in vaults
- Enhance efficiency—using a JWT as a bearer token enables a simplified impersonation with one request to the Sign In endpoint instead of two requests

- Extend and automate complex Tableau integrations and backend queries—such as dynamic content retrieval and advanced filtering

Scope actions

Connected apps use scopes that grant access to content or administrative actions through the REST API methods that support JWT authorization (below). A scope is a colon-separated string that starts with the namespace `tableau`, followed by the Tableau resource that access is being granted to, such as `datasources`, and ends with the action that is allowed on the resource, such as `update`.

The action a scope can take include:

- `create`
- `read`
- `run`
- `update`
- `download`
- `delete`

For example, a scope that allows your custom application to call the Update Data Source method looks like:

```
tableau:datasources:update
```

Scope types

The type of scope you use depends on the content or administrative action that you want to enable. Scopes generally fall into one of the following types: content read, individual, wild-card, and cross-category.

- **Content read scope:** The content read scope, `tableau:content:read`, enables supported GET methods for Tableau content. When you use this scope, you enable actions across REST API categories. More specifically, using this scope you enable GET methods for data sources, metrics, views, workbooks, projects, and sites. Starting in Tableau Server 2023.3, you also specify this scope in a JWT that will be used to

create a credentials token for use with the Metadata API.

Note: To enable GET methods for administrative actions, like users and groups, you can use their individual scopes.

- Individual scopes: To enable supported content and administrative actions, you can use their individual scopes. An individual scope is generally associated with a single method and REST API category.

  Examples:

  - To enable publish or update a data source action, you can use the individual `tableau:datasources:create` or `tableau:datasources:update` scope, respectively.
  - For administrative actions like add or remove users, you can use the individual `tableau:users:create` or `tableau:users:delete` scope, respectively.

  Note: There are some individual scopes that can enable actions across REST API categories. For example, `tableau:views:download` enables actions in the view data and workbooks REST API categories.

- Wildcard (*) scopes: For certain scopes, you can replace the action with the wildcard character (*) to enable supported actions within a specific REST API category.

  Examples:

  - You can use the `tableau:projects:*` wildcard scope to enable the create, delete, update actions in the projects REST API category.
  - You can use the `tableau:users:*` wildcard scope to enable the get/list, add, delete, update actions in the users REST API category.
  - You can use the `tableau:tasks:*` wildcard scope to enable the get/list, add, delete, update and run actions of extract and subscriptions REST API categories. In addition, this scope enables update data source (if an extract) and update workbook.

- Cross-category scopes: In addition to the content read scope, there are a few additional scopes that, if used, enable supported actions across different REST API categories.

Examples:

- If using the `tableau:tasks:run` scope, you enable actions in the data sources and workbooks REST API categories.
- Again, if using the `tableau:views:download` scope, you enable actions in the view data and workbook REST API categories.
- If using permissions scopes like `tableau:permissions:update` or `tableau:permissions:delete`, you enable actions in the data sources, workbooks, and projects REST API categories.

Summary of how to authorize REST API access

The following list summarizes the steps to request access to the REST API through a JWT:

1. **Create a connected app** using one of the following methods:
   - Configure Connected Apps with Direct Trust
   - Configure Connected Apps with OAuth 2.0 Trust
2. **Generate a valid JWT**—at runtime your custom application will generate a valid JWT, configured with the scopes you have included
3. **Make a Sign In request**—your custom application will make a Sign In request using the JWT to return a Tableau credentials token and site ID (LUID)
4. **Use the Tableau access token in subsequent requests**—in subsequent REST API calls, use 1) the Tableau credentials token as the `X-Tableau-Auth` header value and 2) the site ID (LUID) in the request URI

Example

For example, suppose you create a connected app using direct trust. Using direct trust, your custom application that calls the REST API generates a valid JWT using the client ID and client secret generated by the connected app.

Scopes in the JWT

To successfully authorize access to the REST API, the JWT must also contain the scopes that define the REST API capabilities. For example, to enable various data source-related methods, you might include the following scopes in the JWT:

```
"tableau-
:con-
tent:read","t-
ableau-
:data-
sources:cre-
ate","t-
ableau-
:data-
sources:update","tableau:datasources:download","tableau:tasks:run"
```

Or

```
"tableau:content:read","tableau:datasources:*","tableau:tasks:run"
```

Note: Scope values must be passed as a list type.

Sign In Request URI

To make a call to the REST API, your custom application must first make a Sign In request to generate a Tableau credentials token.

```
POST https://myco/api/3.17/auth/signin
```

Request body

To authorize REST API access using a JWT, the Sign In request body must contain the valid JWT like the example below.

```
<tsRequest>
   <credentials jwt-
="eyJ-
pc3MiOiI4ZTFiNzE3Mi0zOWMzLTRhMzItODg3ZS1mYzJiNDExOWY1NmQiLCJh-
bGciOiJIUzI1NiIsImt-
pZCI6ImIwMTE1YmY5LTNhNGItNGM5MS1iMDA5LWNmMGMxNzBiMWE1NiJ9.eyJh-
dWQiOiJ0YWJsZWF1Ii-
```

```
wic3ViI-
joicm1vaGFuQHRhYmxlYXUuY29tIi-
wic2NwIjp-
bInRhYmxlYXU6c2l0ZXM6cmVhZCJdLCJp-
c3MiOiI4ZTFiNzE3Mi0zOWMzLTRhMzItODg3ZS1mYzJiNDExOWY1NmQiLCJleHAiOjE-
2NDg2Njg0Mzk-
sIm-
p0aSI6IjY1ZWFmMmYxLTNmZTgtNDc5Ny1hZmRiLTMyODMzZDVmZGJkYSJ9.mUv2o4gt-
BTrMVLEXY5XTpzDQTGvfE2LGi-3O2vdGfT8">
    <site contentUrl="mycodotcom"/>
   </credentials>
</tsRequest>
```

Response body

The Sign In request produces the following response body, which includes the Tableau credentials token.

```
<tsResponse>
   <credentials token="12ab34cd56ef78ab90cd12ef34ab56cd">
    <site id="9a8b7c6d5-e4f3-a2b1-c0d9-e8f7a6b5c4d" contentUrl=""/>
    <user id="9f9e9d9c-8b8a-8f8e-7d7c-7b7a6f6d6e6d" />
   </credentials>
</tsResponse>
```

After the Tableau access token is generated, add the Tableau credentials token to the header of all subsequent REST API requests.

Header

```
X-Tableau-Auth:12ab34cd56ef78ab90cd12ef34ab56cd
```

All subsequent REST API requests using the Tableau access token are then bounded by the scopes in the JWT.

REST API methods that support JWT authorization

The following scopes can be associated with the connected app to define access and methods your custom application can have to the REST API on users' behalf.

Notes:

- For other REST API capabilities not listed in the table below, you can use other authorization mechanisms to access the methods. For more information, see Authentication Methods in the Tableau REST API Help.
- Both the Sign In and Sign Out methods are supported by JWT authorization but do not require scopes to use beginning in Tableau Server 2023.3.
- For scopes supported by the Embedding API v3, see one of the following:
  - Configure Connected Apps with Direct Trust
  - Configure Connected Apps with OAuth 2.0 Trust

Wildcard (*) scopes

Wildcard scopes use the wildcard character (*) instead of a specific action, to enable multiple supported actions within a specific REST API category. These include:

Scope	Methods enabled
`tableau:datasources:*`	Enables create, update, and update connection data source methods.
`tableau:metrics:*`	Enables query, update, and delete metrics actions.
`tableau:workbooks:*`	Enables publish, update, download, and preview image workbook actions.
`tableau:groups:*`	Enables create, query, update, and delete groups actions.
`tableau:projects:*`	Enables create, delete, and update projects methods.
`tableau:users:*`	Enables get/list, add, delete, and update users methods.

Scope	Methods enabled
`tableau:tasks:*` **Note:** This scope is also cross-category.	Enables get/list, add, delete, update and run methods for extracts and subscription tasks. Enables update methods for data sources for workbooks.

Cross-category scopes

Cross-category scopes enable multiple supported actions across multiple REST API categories. These include:

Scope	Methods enabled
`tableau:content:read`	Enables read/list methods for Tableau content, including data sources, metrics, views, workbooks, projects, and sites.
`tableau:tasks:run`	Enables run methods for data sources, workbooks, and extracts.
`tableau:views:download`	Enables download methods for view data and workbooks.
`tableau:tasks:*` **Note:** This scope is also wildcard.	Enables get/list, add, delete, update and run methods for extracts and subscription tasks. Enables update methods for data sources for workbooks.

Individual scopes

Method	Scope	Description
(Methods without scopes)	(None)	When no scopes are defined in the

Method	Scope	Description
		JWT, access to the REST API is denied.
Sign in	(No scope needed)	Signs you in as a user on the specified site on Tableau Server.
Sign out	(No scope needed)	Signs you out of the current session.
(Content read scope)	`tableau:content:read`	Enables read/list actions for Tableau content: data sources, metrics, views, workbooks, and projects.
Labels		
Delete Label	`tableau:labels:delete`	Deletes a data label by its LUID.
Delete Labels	`tableau:labels:delete`	Deletes the data labels on one or more assets.

Method	Scope	Description
Get Label	`tableau:labels:read`	Gets a data label by its LUID.
Get Labels	`tableau:labels:read`	Displays information about the data labels on one or more assets.
Update Label	`tableau:labels:update`	Updates a label by its LUID.
Update Labels	`tableau:labels:update`	Creates or updates labels on one or more assets.
Data sources		
(All `tableau-:datasources:` methods)	`tableau:datasources:*`	Enables create data source, update data source, and update data source connection methods.
Publish data source	`tableau:datasources:create`	Publish a data source to a site or append

Method	Scope	Description
		data to an existing published data source.
Query data source	`tableau:content:read`	Get information about a published data source.
Query data sources	`tableau:content:read`	Get information about all published data source on a site.
Query data source connections	`tableau:content:read`	Get server address, port, user name, or password information about a published data source.
Update data source	`tableau:datasources:update`	Update owner, project or certification status of the data source.
Update data source connection	`tableau:datasources:update`	Update server address, port, user name, or password of the data

Method	Scope	Description
		source connection.
Update data source now	`tableau:tasks:run`	Run extract refresh.
Extracts		
(All `tableau:tasks:` methods)	`tableau:tasks:*`	Enables create, delete, get, list, run, and update refresh actions for extracts, subscriptions, update data source (for data sources with extracts), and update workbook methods.
List extract refresh tasks in site	`tableau:tasks:read`	List the extract refreshes tasks configured for in a site.
Run extract refresh task	`tableau:tasks:run`	Runs an extract refresh task.
Flows		

Method	Scope	Description
Publish flow	`tableau:flows:create`	Publish a flow.
Metrics **Retirement of the legacy metrics feature** Tableau's legacy metrics feature was retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2. In October 2023, Tableau retired the ability to embed legacy metrics in Tableau Cloud and in Tableau Server version 2023.3. With Tableau Pulse, we've developed an improved experience to track metrics and ask questions of your data. For more information, see Create Metrics with Tableau Pulse to learn about the new experience and Create and Troubleshoot Metrics (Retired) for the retired feature.		
(All `tableau:metrics:` methods)	`tableau:metrics:*`	Enables query, update, and delete metrics actions.
(All `tableau:metrics:` methods)	`tableau:metrics:*`	Enables query, update, and delete metrics actions.
Get metric	`tableau:content:read`	Get a metric.
Delete metric	`tableau:metrics:delete`	Delete a metric.
List metrics	`tableau:content:read`	Get list of metrics for a site.
Query metrics data	`tableau:metrics:download`	Get under-

Method	Scope	Description
		lying data of a metric in comma-separated value (.csv) format.
Update metric	`tableau:metrics:update`	Update owner, project, suspend status, and name of the metric.
Subscriptions		
(All `tableau:tasks:` methods)	`tableau:tasks:*`	Enables create, delete, get, list, run, and update refresh actions for extracts, subscriptions, update data source (for data sources with extracts), and update workbook methods.
Create subscription	`tableau:tasks:create`	Create a subscription.
Delete subscription	`tableau:tasks:delete`	Delete a subscription.

Method	Scope	Description
Get subscription	`tableau:tasks:read`	Gets the details of a subscription.
List subscriptions	`tableau:tasks:read`	Lists subscriptions in a site.
Update subscription	`tableau:tasks:update`	Updates a subscription.
Views		
Delete custom view	`tableau:views:update`	Delete the specified custom view.
Get custom view	`tableau:content:read`	Get the details of a specified custom view.
Get custom view image	`tableau:views:download`	Download a .png format image file of a specified custom view.
Get view	`tableau:content:read`	Get details about a view.
Get view by path	`tableau:content:read`	Get details for all views on a site using the specified name.

Method	Scope	Description
List custom views	`tableau:content:read`	Get a list of custom views on a site.
Query view data	`tableau:views:download`	Get a view rendered in comma-separated value (.csv) format.
Query view PDF	`tableau:views:download`	Get a view as a PDF (.pdf) file.
Query view image	`tableau:views:download`	Get a view as an image (.png) file.
Query views for site	`tableau:content:read`	Get all views for a site.
Query views for workbook	`tableau:content:read`	Get all views for the specified workbook.
Query view preview image	`tableau:views:download`	Get the thumbnail image (.png) of the view.
Update custom view	`tableau:views:update`	Change the owner or name of an existing custom view.

Method	Scope	Description
Workbooks		
(All `tableau-:workbooks:` methods)	`tableau:workbooks:*`	Enables publish, update, download, and preview image workbook actions.
Publish workbook	`tableau:workbooks:create`	Publish a workbook (.twb or .twbx).
Query workbook	`tableau:content:read`	Get a specified workbook and its details.
Query workbook for site	`tableau:content:read`	Get a list of workbooks published to a site.
Query workbook preview image	`tableau:workbooks:download`	Get the thumbnail image (.png) of the workbook.
Update workbook	`tableau:workbooks:update`	Modify an existing workbook.
Update workbook connection	`tableau:workbooks:update`	Update the connection information.

Method	Scope	Description
Update workbook now	`tableau:tasks:run`	Initiate a workbook refresh outside of a scheduled task.
Publish		
Append to file upload	`tableau:file_uploads:create`	Upload a block of data and append it to the data that is already uploaded - to be used after an upload has been initiated using the "initiate file upload" method.
Initiate file upload	`tableau:file_uploads:create`	Initiate the upload process of a file.
Download		
Download data source	`tableau::datasources:download`	Download the data source (.tdsx).
Download view crosstab Excel	`tableau:views:download`	Download an Excel (.xlsx) file containing

Method	Scope	Description
		crosstab data from the view.
Download workbook	`tableau:workbooks:download`	Download a workbook (.twb or .twbx).
Download workbook revision	`tableau:workbooks:download`	Download a specific version of the workbook (.twb or .twbx).
Download workbook PDF	`tableau:views:download`	Download a PDF (.pdf) file containing images of the sheets in the workbook.
Download workbook PowerPoint	`tableau:views:download`	Download a PowerPoint (.pptx) file containing slides of the sheets in the workbook.
Users		
(All `tableau:users` methods)	`tableau:users:*`	Enables add, query, update, and remove users

Method	Scope	Description
		actions.
Add user to group	`tableau:groups:update`	Add a user to a group.
Add user to site	`tableau:users:create`	Add a user and assign the user to a site.
Get users in group	`tableau:groups:read`	Get a list of users in a group.
Get users on site	`tableau:users:read`	Get all users on a site.
Query user on site	`tableau:users:read`	Get a user on a site.
Remove users from group	`tableau:groups:update`	Remove a user from a group.
Remove user from site	`tableau:users:delete`	Remove the user from a site.
Groups		
(All `tableau:groups:` methods)	`tableau:groups:*`	Enables create, query, update, and delete groups actions.
Create group	`tableau:groups:create`	Create a group.

Method	Scope	Description
Delete group	`tableau:groups:delete`	Delete a group.
Get groups for user	`tableau:users:read`	Get a list of groups that a user belongs to.
Query groups	`tableau:groups:read`	Get a list of groups on a site.
Update group	`tableau:groups:update`	Update a group.
Projects		
(All `tableau-:projects:` methods)	`tableau:projects:*`	Enables create, update, and delete projects actions.
Create project	`tableau:projects:create`	Create a project.
Delete project	`tableau:projects:delete`	Delete a project.
Query project	`tableau:content:read`	Get a list of projects.
Update project	`tableau:projects:update`	Update the name, description, or project hierarchy of the project.

Method	Scope	Description
Permissions		
(All `tableau-:permissions:` methods)	`tableau:permissions:*`	Enables add, query, update, delete permissions actions.
Add data source permissions	`tableau:permissions:update`	Add permissions to a data source for a Tableau Server user or group.
Add default permissions	`tableau:permissions:update`	Add default permission capabilities to a user or group, for metric, flow, workbook, data source, data role, or lens resources in a project.
Add project permissions	`tableau:permissions:update`	Add permissions to a project for a user or group
Add view permissions	`tableau:permissions:update`	Add permissions to a view for a user or group.

Method	Scope	Description
Add workbook permissions	`tableau:permissions:update`	Add permissions to a specified workbook for a user or group.
Delete data source permissions	`tableau:permissions:delete`	Delete default permission capabilities of a user or group, for metric, flow, workbook, data source, data role, or lens resources in a project.
Delete default permissions	`tableau:permissions:delete`	Delete default permission capabilities of a user or group, for metric, flow, workbook, data source, data role, or lens resources in a project.
Delete project permissions	`tableau:permissions:delete`	Delete the project permission for a user or group.

Method	Scope	Description
Delete view permissions	`tableau:permissions:delete`	Delete the view permission for a user or group.
Delete workbook permissions	`tableau:permissions:delete`	Delete the workbook permission for a user or group.
Query data source permissions	`tableau:permissions:read`	Get a list of permissions for the data source.
Query default permissions	`tableau:permissions:read`	Get default permission capabilities of users and groups for metrics, workbooks, and data sources.
Query project permissions	`tableau:permissions:read`	Get a list of permissions for the project.
Query view permissions	`tableau:permissions:read`	Get a list of permissions for the view.
Query workbook permissions	`tableau:permissions:read`	Get a list of permissions for the workbook.

Method	Scope	Description
Site		
(All `tableau:sites:` methods)	`tableau:sites:*`	Enables create, query, update, and delete sites actions.
Create site	`tableau:sites:create`	Create a site on Tableau Server.
Get recently viewed site	`tableau:content:read`	Get views and workbooks details on the most recently created, updated, or accessed by the signed in user.
Query sites	`tableau:sites:read`	List all sites on Tableau Server.
Query views for site	`tableau:content:read`	List all views on a site.
Update site	`tableau:sites:update`	Update a site.

401001 - signin error

If you encounter error 401001, the **Sign In** response body is appended with one of the following additional connected apps-specific error codes: 16, 10084, or 10085.

For example, in the following response body, "10084" is the connected apps error code you can use to help troubleshoot issues with signing in to Tableau Server using a JWT for REST API authorization.

```
<error code="401001">
  "summary": "Signin Error",
  "detail": "Error signing in to Tableau Cloud (10084)"
</error>
```

To help resolve the issue, refer to the description of the applicable error code and its potential causes.

- **16: Could not find user**—this error can occur because the incorrect "`sub`" (user name) was specified
- **10084: Could not parse access token**—this error can occur for the following reasons:
  - JWT is invalid or there was an unexpected problem
  - Incorrect "`aud`" (audience) was specified
  - For direct trust, there was a problem with signing the secret
- **10085: Could not fetch secret to verify signature for client ID**—this error can occur for the following reasons:
  - Incorrect client ID in "`iss`" specified
  - For direct trust, incorrect "`kid`" (secret ID) was specified
  - For EAS, unable to fetch keys from the JWKSource

401002 - unauthorized access error

If you encounter error 401002 and have confirmed that you have the appropriate permissions to make the request, ensure the scope included in the JWT is correct and matches the request you're trying to make. For a list of endpoints and supported scopes, see the REST API methods that support JWT authorization section above.

Troubleshoot Connected Apps - Direct Trust

When embedded content fails to display in your custom application or Tableau REST API authorization fails, you can use a browser's developer tools to inspect and identify error codes that might be associated with the Tableau connected app that's used to display the embedded content.

Note: In order for the session token to be valid, the clocks of the external application and the server that hosts the external application must be set to Coordinated Universal Time (UTC). If either clock uses a different standard, the connected app will not be trusted.

Refer to the table below to review the description of the error code and potential resolution.

Error code	Summary	Description	Potential resolution or explanation
5	SYSTEM_USER_NOT_FOUND	Tableau user could not be found	To resolve this issue, verify the '`sub`' (Subject) claim value in the JWT is "username" for Tableau Server. This value is case sensitive.
16	LOGIN_FAILED	Login failed	This error is typically caused by one of the following claim issues in the JWT: • The '`exp`' (Expiration Time) exceeds the default maximum validity period. To resolve this issue, review

			registered claims required for a valid JWT and ensure the correct value is used. To change the maximum validity period, you can use the viz-portal.oauth.connected_apps.max_expiration_period_in_minutes command. • The 'sub' (Subject) is calling an unknown user. To resolve this issue, verify the 'sub' claim is "username". This value is case sensitive.
67	FEATURE_NOT_ENABLED	On-demand access is not supported	On-demand access is available through licensed Tableau Cloud sites only.
126	CONNECTED_APP_NOT_FOUND	The connected app could not be found	To resolve this issue, verify the connected app is enabled and the correct client ID (also known as the connect app ID) is referenced in the JWT.
127	CONNECTED_APP_SECRET_NOT_FOUND	The connected app's secret could not be found	To resolve this issue, verify the correct connected app's secret ID and secret value are referenced in the JWT.
128	CONNECTED_APP_SECRET_LIMIT_EXCEEDED	Maximum limit for secrets has been reached	A maximum of two secrets are allowed for a connected app. This error can occur when there's an attempt to create a third secret. To resolve this issue, delete a secret from the connected app

			before creating a new one.
133	INVALID_ CONNECTED_ APP_DOMAIN_ SAFELIST	Domain allowlist contains one or more invalid characters	This error can occur when the domain allowlist contains one or more invalid characters.
10083	BAD_JWT	JWT header contains issues	The '`kid`' (Secret ID) or '`clientId`' (Issuer) claims are missing from the JWT header. To resolve this issue, ensure this information is included.
10084	JWT_PARSE_ ERROR	JWT contains issues	To resolve this issue, verify the following: • The '`aud`' (Audience) value referenced in the JWT uses the "tableau" value. This value is case sensitive. • The '`aud`' (Audience) and '`sub`' (Subject) are included in the JWT. • Review IssueTime or ensure there's no clock mismatch between the machine hosting the connected app and Tableau Cloud.
10085	COULD_NOT_ FETCH_JWT_ KEYS	JWT could not find keys	Could not find the secret. To resolve this issue, verify the correct 'kid' (Secret ID) is used in the JWT header.
10087	BLOCKLISTED_ JWS_	Issue with the JWT signing	To resolve the issue, you can

	ALGORITHM_ USED_TO_SIGN	algorithm	remove the signing algorithm. For more information, see viz-portal.oauth.external_author-ization_server.blocklisted_jws_ algorithms.
10089	CONNECTED_ APP_NOT_ FOUND	Could not find connected app	To resolve this issue, ensure the issuer is calling the correct con-nected app ID (also known as the client ID).
10090	CONNECTED_ APP_DISABLED	Connected app is disabled	The connected app used to verify trust is disabled. To resolve this issue, enable the connected app.
10091	JTI_ALREADY_ USED	Unique JWT required	The JWT has already been used in the authentication process. To resolve this issue, a new JWT must be generated.
10092	NOT_IN_ DOMAIN_ ALLOW_LIST	Domain of the embedded content is not specified	To resolve this issue, ensure the `unrestrictedEmbedding` set-ting is set to `true` or `domainAl-lowlist` parameter includes the domains where Tableau content is embedded using the Update Embedding Settings for Site method in the Tableau REST API.
10094	MISSING_ REQUIRED_JTI	Missing JWT ID	To resolve this issue, verify the '`jti`' (JWT ID) is included in the JWT.
10096	JWT_ EXPIRATION_ EXCEEDS_ CONFIGURED_ EXPIRATION_ PERIOD	Issue with expiration time	The '`exp`' (Expiration Time) exceeds the default maximum validity period. To resolve this issue, review registered claims required for a valid JWT and

			ensure the correct value is used. To change the maximum validity period, you can use the viz-portal.oauth.external_author-ization_server.max_expiration_period_in_minutes command.
10097	SCOPES_ MALFORMED	Issues with scopes claim	This error can occur when the '`scp`' (Scope) claim is either miss-ing from the JWT or not passed as a list type. To resolve this issue, verify '`scp`' is included in the JWT and passed as a list type. For troubleshooting help with a JWT, see Debugger on the auth0 site.
10098	JWT_ UNSIGNED_OR_ ENCRYPTED	JWT is unsigned or encrypted	Tableau does not support an unsigned or encrypted JWT.
10099	SCOPES_ MISSING_IN_ JWT	Missing scopes claim	The JWT is missing the required '`scp`' (scope) claim. To resolve this issue, verify '`scp`' is included in the JWT. For troubleshooting help with a JWT, see Debugger on the auth0 site.
10100	JTI_ PERSISTENCE_ FAILED	Unexpected JWT ID error	There was an unexpected '`jti`' (JWT ID) error. To resolve this issue, a new JWT with a new '`jti`' must be generated.
10103	JWT_MAX_SIZE_ EXCEEDED	JWT exceeds maximum size	This error can occur when JWT size exceeds 8000 bytes. To resolve this issue, make sure that only the necessary claims are being passed to Tableau Server.

10105	ORIGIN_ HEADER_NOT_ A_VALID_URI	Invalid Origin header	This error can occur because 1) a URL is specified in the domain allowlist and 2) the Origin header does not contain a valid URL.

Data Connection Authentication

You can configure data connection authentication using Kerberos, OAuth, and single sign-on.

More information

- Tableau Server on Linux - Connecting to a Windows Shared Directory (Tableau Community)
- Setting an Oracle Connection to Use TNSNames.ora or LDAP.ora (Tableau Support)

Enable Kerberos Delegation

Kerberos delegation enables Tableau Server to use the Kerberos credentials of the viewer of a workbook or view to execute a query on behalf of the viewer. This is useful in the following situations:

- You need to know who is accessing the data (the viewer's name will appear in the access logs for the data source).
- Your data source has row-level security, where different users have access to different rows.

Supported data sources

Tableau supports Kerberos delegation with the following data sources:

- Cloudera: Hive/Impala
- Denodo
- Hortonworks
- Oracle
- PostgreSQL
- Spark

- SQL Server
- Teradata
- Vertica

MSAS is not supported on Linux platforms.

Requirements

Kerberos delegation requires Active Directory.

- The Tableau Server identity store must be configured to use Active Directory.
- The computer where Tableau Server is installed must be joined to the Active Directory domain.
- MIT Kerberos KDC is not supported.

Web authoring and user Kerberos authentication

When configuring Connect to Data for a given target, you may select Integrated or Windows authentication as the preferred authentication method. However, for web authoring scenarios, the default behavior will be to use the Kerberos service account (“Run As” account) instead.

To enable user credentials in web authoring scenarios with Kerberos delegation, you must make an additional configuration using TSM. Run the following commands:

```
tsm configuration set -k native_api.We-
bAuthoringAuthModeKerberosDelegation -v true
```

```
tsm pending-changes apply
```

After making this configuration, Kerberos Delegation becomes the default operation when selecting integrated authentication with web authoring. However, this setting will not prevent content creators from accessing the service account. Creators can still publish content that connects with the Run As service account, using Tableau Desktop or other methods.

For more information on Run As service account, see Enable Kerberos Service Account Access.

Configuration process

This section provides an example of the process to enable Kerberos delegation. The scenario also includes example names to help describe the relationships between the configuration elements.

1. Tableau Server will need a Kerberos service ticket to delegate on behalf of the user that is initiating the call to the database. You must create a domain account that will be used to delegate to the given database. This account is referred to as the Run As service account. In this topic, the example user configured as the delegation/Run As account is `tabsrv@example.com`.

   The account must be configured with Active Directory User and Computers on a Windows Server that is connected to the user domain:

   - Open the **Properties** page for the Run As service account, click the **Delegation** tab and select **Trust this user for delegation to specified services only** and **Use any authentication protocol**.

2. Create a keytab file for the Run As service account.

   For example, the following commands create a keytab (`tabsrv-runas.keytab`) using the ktutil tool:

   ```
   sudo ktutil
   ```

   ```
   ktutil:  addent -password -p tabsrv@EXAMPLE.COM -k 2 -e <encryption scheme>
   ```

   Encryption schemes for this command include `RC4-HMAC`, `aes128-cts-hmac-sha1-96`, and `aes256-cts-hmac-sha1-96`. Consult your IT team for the correct encryption scheme for your environment and data source.

   ```
   ktutil:  wkt tabsrv-runas.keytab
   ```

   Tableau Server will use the Run As service account and the associated keytab to authenticate and make a direct connection to the database.

3. Copy the keytab into the Tableau Server data directory and set proper ownership and permissions. If you are running a multi-node deployment, then you must run the following commands on each node in the cluster.

```
mkdir /var/opt/keytab
sudo cp -p tabsrv-runas.keytab /var/opt/keytab
sudo chown $USER /var/opt/keytab/tabsrv-runas.keytab

chgrp tableau /var/opt/keytab/tabsrv-runas.keytab

chmod g+r /var/opt/keytab/tabsrv-runas.keytab
```

4. Run the following TSM commands to enable Kerberos delegation, set the delegation service account, and associate the keytab file with the service account:

```
tsm configuration set -k wgserver.delegation.enabled -v true
tsm configuration set -k native_api.datasource_impersonation_
runas_principal -v tabsrv@EXAMPLE.COM
tsm configuration set -k native_api.datasource_impersonation_
runas_keytab_path -v /var/opt/keytab/tabsrv-runas.keytab
tsm configuration set -k native_api.protocol_transition_a_d_
short_domain -v false
tsm configuration set -k native_api.protocol_transition_upper-
case_realm -v true
```

   In some cases, TSM may return an error mentioning `--force-keys`. If you get this error, run the command again with the `--force-keys` parameter appended to the argument.

5. Run the following TSM command apply the changes to Tableau Server:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

6. Enable delegation for data connections:

   - **SQL Server**—See Enabling Kerberos Delegation for SQL Server in the Tableau Community.
   - **PostgreSQL**—See Enabling Kerberos Delegation for PostgreSQL in the Tableau Community.
   - **Teradata**—See Enabling Kerberos Delegation for Teradata in the Tableau Community.
   - **Denodo**—See Enabling Kerberos Delegation for Denodo on Linux in the Tableau Community.
   - **Oracle**—See Enable Kerberos Delegation for Oracle in the Tableau Community.
   - **Cloudera**—See Enable Kerberos Delegation for Hive/Impala in the Tableau Community.
   - **Vertica**—See Enabling Kerberos Delegation for Vertica in the Tableau Community.

   See also

   Troubleshoot Kerberos

Enable Kerberos Delegation for JDBC Connectors

As of version 2020.2, Tableau Server supports Kerberos delegation for JDBC connectors.

Kerberos delegation enables Tableau Server to use the Kerberos credentials of the viewer of a workbook or view to execute a query on behalf of the viewer. This is useful in the following situations:

- You need to know who is accessing the data (the viewer's name will appear in the access logs for the data source).
- Your data source has row-level security, where different users have access to different rows.

Supported data sources

Tableau supports JDBC Kerberos RunAs authentication with the following data sources:

- Oracle
- PostgreSQL

Both native and JDBC-based connectors use the same configuration on Tableau Server on Linux. See Enable Kerberos Delegation.

Enable Kerberos Run As Authentication for JDBC Connectors

As of version 2020.2, Tableau Server supports Kerberos authentication for JDBC connectors.

You can configure Tableau Server to use a Kerberos service account to access a database. In this scenario, Tableau Server connects to databases with a service account, also referred to as a "Run As service account". This scenario is referred to as "Run As authentication"

To use Run As authentication on Tableau Server you must first create a workbook or datasource in Tableau Desktop that uses integrated authentication. When you publish to Tableau Server you will get the option to use Run As authentication. When creating a datasource with Web Authoring, Run As authentication is the default operation if you select integrated authentication.

Supported data sources

Tableau supports JDBC Kerberos delegation with the following data sources:

- Oracle
- PostgreSQL

Both native and JDBC-based connectors use the same configuration on Tableau Server on Linux. To configure Run As authentication see Enable Kerberos Service Account Access.

OAuth Connections

Tableau Server supports OAuth for a number of different connectors. In many cases, OAuth functionality doesn't require additional configuration on Tableau Server.

From Tableau, when users sign in to data with a connector that uses OAuth, users are redirected to the authentication provider's sign in page. After users provide their credentials and authorize Tableau to access their data, the authentication provider sends Tableau an **access token** that uniquely identifies Tableau and the users. This access token is used to access data on users' behalf. For more information, see Overview of the OAuth process below.

Using OAuth-based connections provides the following benefits:

- **Security:** Your database credentials are never known to or stored in Tableau Server, and the access token can be used only by Tableau on behalf of users.
- **Convenience:** Instead of having to embed your data source ID and password in multiple places, you can use the token provided for a particular data provider for all published workbooks and data sources that access that data provider.

  Note: For live connections to Google BigQuery data, each workbook viewer can have a unique access token that identifies the user, rather than sharing a single username and password credential.

Overview of the OAuth process

The following steps describe a workflow in the Tableau environment that calls the OAuth process.

1. A user takes an action that requires access to a cloud-based data source.

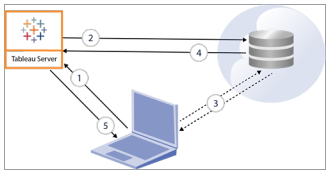
   For example, you open a workbook that's published to Tableau Server.

2. Tableau directs the user to the cloud data provider's sign in page. The information that is sent to the data provider identifies Tableau as the requesting site.

3. When the user signs in to the data, the provider prompts the user to confirm their authorization for Tableau Server to access the data.

4. Upon the user's confirmation, the data provider sends an access token back to Tableau Server.

5. Tableau Server presents the workbook and data to the user.



Note: Single use refresh tokens are not supported for OAuth connections to Tableau at this time. In most cases, you can set up your identity provider (such as Okta or Redshift IDC) to use rolling refresh tokens instead. For more information, see your provider's OAuth documentation.

The following user workflows can use the OAuth process:

- Creating a workbook and connecting to the data source from Tableau Desktop or from Tableau Server.
- Publishing a data source from Tableau Desktop.
- Signing in to Tableau Server from an approved *client*, such as Tableau Mobile or Tableau Desktop.

Default saved credential connectors

Saved credentials refers to the functionality where Tableau Server stores user tokens for OAuth connections. This allows users to save their OAuth credentials to their user profile on Tableau Server. After they've saved the credentials, they won't be prompted when they subsequently publish, edit, or refresh when accessing the connector.

> **Note**: When editing Tableau Prep flows on the web, you may still be prompted to reauthenticate.

The following connectors use saved credentials by default and don't require additional configuration on Tableau Server.

- Anaplan
- Box
- Dropbox
- Esri ArcGIS Server
- Google Ads, Google Drive
- LinkedIn Sales Navigator
- Marketo
- OneDrive (Additional configuration is required starting with 2022.3)
- Oracle Eloqua
- ServiceNow ITSM
- Snowflake - To use "private link" requires additional configuration. For more information, see Configure Snowflake OAuth for Partner Applications on the Snowflake website and Configure OAuth for Snowflake Connections.

The following connectors can use saved credentials with additional configuration by the server administrator.

- Azure Data Lake Storage Gen2, Azure Synapse, Azure SQL Database, Databricks, OneDrive and SharePoint Online, and SharePoint Lists (JDBC)

  For more information, see Configure Azure AD for OAuth and Modern Authentication.

- Dremio

For more information, see Set Up OAuth for Dremio.

- Google Analytics, Google BigQuery, Google Sheets (deprecated in Tableau version 2022.1)

  For more information, see Set up OAuth for Google.

  > **Note:** If Tableau Server isn't listed in the Accessed Apps list in the Google admin console, you can manually add a new app to the list using its client ID. To create a client ID, see Change Google OAuth to Saved Credentials.

- Intuit QuickBooks Online

  For more information, see Set Up OAuth for Intuit QuickBooks Online.

  OneDrive (Starting with 2022.3)

  For more information, see Configure Custom OAuth for a site

- Salesforce

  For more information, see Change Salesforce.com OAuth to Saved Credentials.

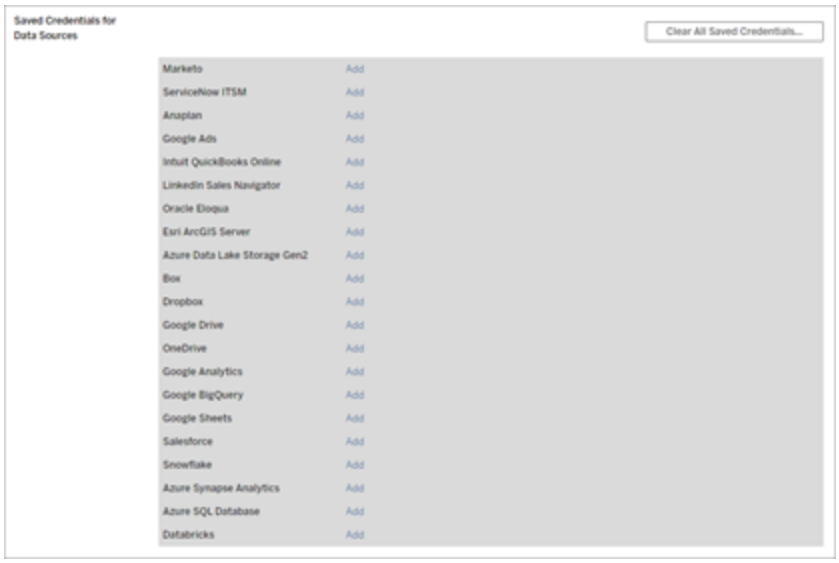
- Salesforce CDP

  For more information, see Connect Tableau Server to the Salesforce Data Cloud.

- Snowflake

  Starting with version 2024.2. For more information, see Change Snowflake OAuth to Saved Credentials.

All supported connectors are listed under **Saved Credentials for Data Sources** on users' **My Account Settings** page on Tableau Server. Users manage their saved credentials for each connector.



Access tokens for data connections

You can embed credentials based on access tokens with data connections, to enable direct access after the initial authentication process. An access token is valid until a Tableau Server user deletes it, or the data provider revokes it.

It’s possible to exceed the number of access tokens your data source provider allows. If that's the case, when a user creates a token, the data provider uses the length of time since last access to decide which token to invalidate to make room for the new one.

Access tokens for authentication from approved clients

By default, Tableau Server sites allow users to access their sites directly from approved Tableau clients, after users provide their credentials the first time they sign in. This type of authentication also uses OAuth access tokens to store the users' credentials securely.

For more information, see Disable Automatic Client Authentication.

Default-managed keychain connectors

Managed keychain refers to the functionality where OAuth tokens are generated for Tableau Server by the provider and shared by all users in the same site. When a user first publishes a

data source, Tableau Server prompts the user for the data source credentials. Tableau Server submits the credentials to the data source provider, which returns OAuth tokens for Tableau Server to use on behalf of the user. On subsequent publishing operations, the OAuth token stored by Tableau Server for the same class and username is used so that the user isn't prompted for the OAuth credentials. Should the data source password change, then the preceding process is repeated and the old token is replaced by a new token on Tableau Server.

Additional OAuth configuration on Tableau Server isn't required for the default-managed keychain connectors:

- Google Analytics, Google BigQuery, and Google Sheets (deprecated in Tableau version 2022.1)
- Salesforce

Token limits and storage

Google has a 50 token limit per user per client application (in this scenario, Tableau Server is the client application). Because the OAuth token is stored on Tableau Server and reused by the user, the user is unlikely to exceed the token limit.

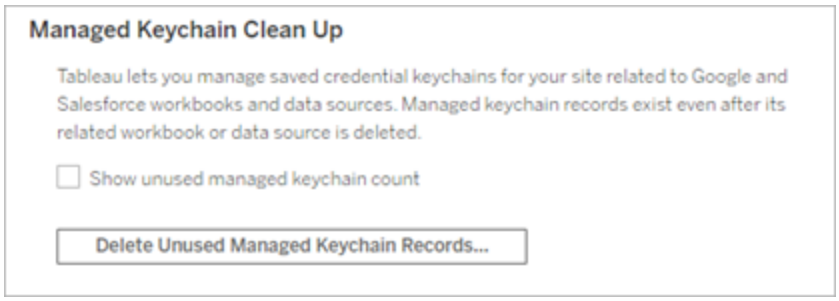
All user tokens are encrypted at rest when stored on Tableau Server. See Manage Server Secrets for more information.

Removing unused keychain records

A managed keychain record contains connection attributes like dbClass, username, and OAuth secret attributes. All managed keychain records for a given site are merged, encrypted, and stored in PostgreSQL.

Records are persisted even for workbooks and data sources that have been removed. Over time, these records can grow to large sizes, which may cause issues.

We recommend purging the unused keychain records periodically as a regular maintenance task. You can view the number of records and unused records stored on each site. You can also delete unused records.



To access **Managed Keychain Clean Up**, sign in to the Tableau Server admin pages, navigate to the site where you want to delete unused records, and click **Settings**.

Scenario limitations with managed keychain

Three scenarios aren't supported when using managed keychain OAuth with Tableau Server:

- Prompting for OAuth credentials on live connections. Users must embed credentials on live connections with managed-keychain OAuth
- Editing the OAuth data source connection on Tableau Server
- Web authoring

Convert managed keychain to saved credentials

You can convert the connectors that use managed keychain to use saved credentials by configuring Tableau Server with an OAuth client ID and secret for each connector. By converting these connectors to saved credentials, users are able to manage their credentials for each connector type on the **My Account Settings** page on Tableau Server. Additionally, live connection prompts, editing connections, and web authoring are also supported.

Configure a custom OAuth for a site

For a subset of connectors, you can configure site-level OAuth by configuring custom OAuth clients. For more information, see one of the following topics:

- For Azure Data Lake Storage Gen2, Azure SQL Database, Azure Synapse, Databricks, OneDrive and SharePoint Online, and SharePoint Lists (JDBC), see Configure custom OAuth for a site.
- For Dremio, see Set Up OAuth for Dremio.
- For Google Analytics, Google BigQuery, Google Sheets (deprecated in Tableau version 2022.1), see Configure custom OAuth for a site.
- For Salesforce, see Configure custom OAuth for a site.
- For Salesforce CDP, see Connect Tableau Server to the Salesforce Data Cloud.
- For Snowflake, see Option 2: Configure OAuth for Snowflake Connections by Site.

Allow Saved Access Tokens

After you configure Tableau Server for OAuth, you can decide to allow users to manage their own OAuth credentials, or you want to manage them centrally. If you want users to manage their own, you need to enable user profile settings from the server.

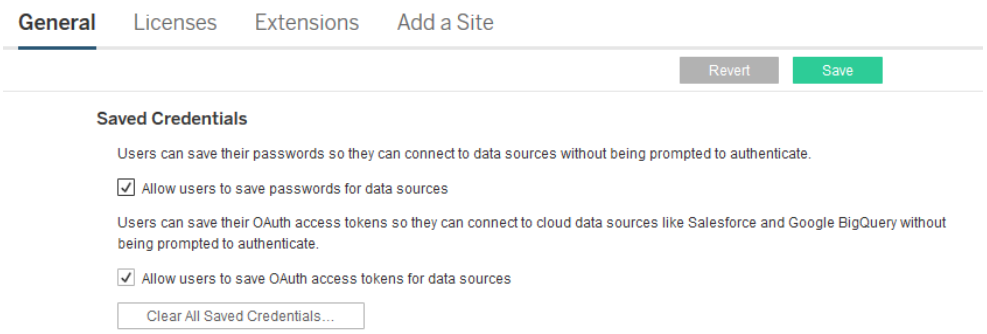
Note: If you have not yet configured your server to enable OAuth data connections, see the related topics listed below.

Note: Single use refresh tokens are not supported for OAuth connections to Tableau at this time. In most cases, you can set up your identity provider (such as Okta or Redshift IDC) to use rolling refresh tokens instead. For more information, see your provider's OAuth documentation.

1. Sign in to Tableau Server as a server administrator.
2. **Single-site**: Click **Settings** > **General**.

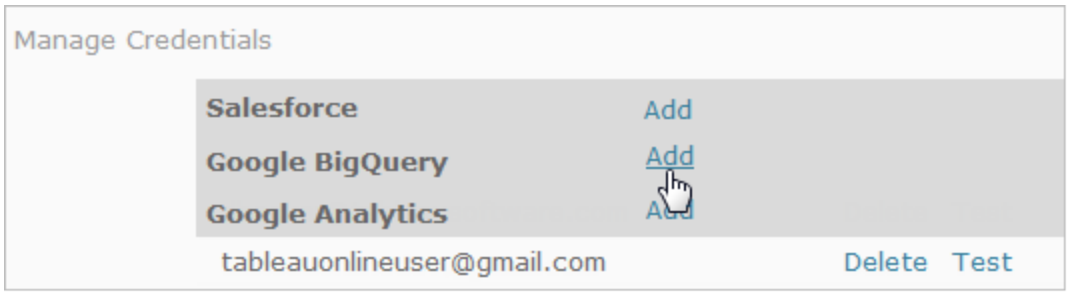
   Multisite: In the site menu, click **Manage All Sites** and then click **Settings** > **General**.
3. In the **Saved Credentials** section, select the following:

   - **Allow users to save passwords for data sources** (allows users to save their individual credentials with data sources).
   - **Allow users to save OAuth access tokens for data sources**



4. Click **Save**.

After you select these check boxes, users will see a **Manage Credentials** section in their profile settings, where they can add access tokens for OAuth data connections.



Managing credentials centrally

Server administrators alternatively can manage OAuth credentials centrally. This can work well, for example, if multiple users work from the same data, and you have a dedicated user account for your data provider.

To manage credentials centrally, you do the following:

- Clear the check boxes described in the preceding procedure.
- Edit connection information as data sources are published.

  When you edit the connection, you embed credentials that use an OAuth access token instead of an individual's user name and password.

When the settings for saving passwords and access tokens are not enabled, the Manage Credentials section is excluded from users' profile settings.

See also

Set up OAuth for Google

Change Salesforce.com OAuth to Saved Credentials

Configure OAuth for Snowflake Connections

Set Up OAuth for Intuit QuickBooks Online

Change Salesforce.com OAuth to Saved Credentials

By default, the Salesforce.com connector uses a managed keychain for OAuth tokens that are generated for Tableau Server by the data provider and shared by all users in the same site. You can configure Tableau Server with saved client ID and client secret. There are three scenarios where you might want to do this:

- **Salesforce connector**—If you're using the Salesforce connector, you can configure Tableau Server with an OAuth client ID and secret, so the connector can use saved

credentials.

- **Write to CRM Analytics**—If you're writing Tableau Prep flow data to Salesforce CRM Analytics (version 2022.3 and later), configure Tableau Server with an OAuth client ID and secret, so the flow can run in Tableau Server using saved credentials.
- **Einstein Discovery**—If you are integrating Einstein Discovery extensions with Tableau Server, you need to do this OAuth client ID and secret configuration. The ability to integrate Einstein Discovery and Tableau Server was added in version 2021.1.0. For more information, see Configure Einstein Discovery Integration.

This topic describes how to set up your Salesforce.com data sources and Einstein Discovery extensions for OAuth saved credentials. Complete these steps for each Tableau Server instance.

For more information about managed keychain and saved credentials, see OAuth Connections

Notes:

- The Salesforce connector requires managed keychain (default), server-wide OAuth, or site-specific OAuth.
- To use saved credentials for a site, server-wide OAuth must be configured first.
- Server-wide OAuth can be used whether site-wide OAuth is configured.
- If using site-specific OAuth, each site must be configured individually.
- To support live connection prompts, editing connections, and web authoring, convert managed keychain to saved credentials to avoid errors.

Summary of steps

Set up OAuth by following these general steps:

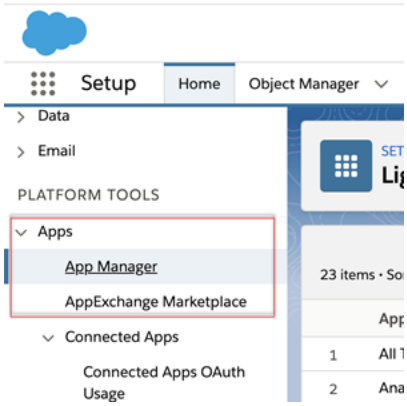
1. Create a Connected App in Salesforce.
2. Use the information you obtained in step 1 to configure Tableau Server.
3. (Optional) Configure site-specific OAuth.

Step 1: Create a Salesforce connected app

Note: This procedure documents the process in Salesforce Lightning. If you are using the traditional interface, the navigation may be different but the configuration is the same.

1. Sign in to your Salesforce.com developer account, click your user name in the upper-right, and then select **Setup**.

2. In the left navigation column, under **Apps**, select **App Manager**.



3. In the Connected Apps section, click **New Connected App**.

4. In **Basic Information**, give the app a name, tab through the api field so it will self-populate in the correct format, and enter a contact email for the app.

5. In the **API [Enable OAuth Settings]** section, select **Enable OAuth Settings**.

6. In the new OAuth settings that appear, for **Callback URL**, type the fully qualified domain name of your server, using the `https` protocol, and append the following text to the URL: `auth/add_oauth_token`.

   For example:

   ```
   https://www.your_tableau_server.com/auth/add_oauth_token
   ```

7. Move the following items from **Available OAuth Scopes** to **Selected OAuth Scopes**:

   - **Access the identity URL service (id, profile, email, address, phone)**
   - **Manage user data via APIs (api)**
   - **Perform requests any time (refresh_token, offline access)**

8. Click **Save**.

After you save the app, Salesforce populates the API section with the following IDs that you will use to configure Tableau Server:

- **Consumer Key**
- **Consumer Secret**
- **Callback URL**

Step 2: Configure Tableau Server for Salesforce.com OAuth

Once your connected app is created in Salesforce and you have the Customer Key, Customer Secret, and the Callback URL, you can configure Tableau Server for Salesforce data connections and outputs, and Einstein Discovery.

1. On the Tableau Server computer, at a command prompt, run the following commands:

   ```
   tsm configuration set -k oauth.salesforce.client_id -v <your_
   customer_key>
   ```

   ```
   tsm configuration set -k oauth.salesforce.client_secret -v
   <your_customer_secret>
   ```

   ```
   tsm configuration set -k oauth.salesforce.redirect_uri -v
   <your_redirect_URL>
   ```

2. (Optional) To change the default login server, type the following command:

   ```
   tsm configuration set -k oauth.salesforce.server_base_url -v
   <URL>
   ```

3. Enter the following command to apply changes:

   ```
   tsm pending-changes apply
   ```

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Configure custom OAuth for a site

You can configure a custom Salesforce OAuth client for a site.

Consider configuring a custom OAuth client to 1) override an OAuth client if configured for the server or 2) enable support for securely connecting to data that requires unique OAuth clients.

When a custom OAuth client is configured, the site-level configuration takes precedence over any server-side configuration and all new OAuth credentials created use the site-level OAuth client by default. No Tableau Server restart is required for the configurations to take effect.

Important: Existing OAuth credentials established before the custom OAuth client is configured are temporarily usable but both server administrators and users must update their saved credentials to help ensure uninterrupted data access.

1: Prepare the OAuth client ID, client secret, and redirect URL

Before you can configure the custom OAuth client, you need the information listed below. After you have this information prepared, you can register the custom OAuth client for the site.

- **OAuth client ID and client secret:** First register the OAuth client with the data provider (connector) to retrieve the client ID and secret generated for Tableau Server.
- **Redirect URL:** Note the correct redirect URL. You will need this during the registration process in **Step 2** below.

  https://<your_server_name>.com/auth/add_oauth_token

  For example, https://example.com/auth/add_oauth_token

2: Register the OAuth client ID and client secret

Follow the procedure described below to register the custom OAuth client to the site.

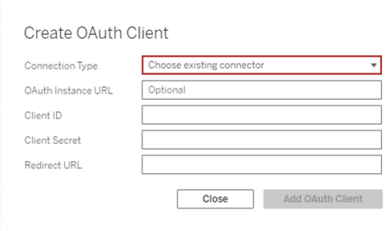
1. Sign in to your Tableau Server site using your admin credentials and navigate to the **Settings** page.
2. Under OAuth Clients Registry, click the **Add OAuth Client** button.
3. Enter the required information, including the information from **Step 1** above:

a. For **Connection Type**, select the connector whose custom OAuth client you want to configure.

b. **OAuth Instance URL** is required if multiple OAuth clients are being registered. Otherwise, it is optional.

c. For **Client ID**, **Client Secret**, and **Redirect URL**, enter the information you prepared in **Step 1** above.

d. Click the **Add OAuth Client** button to complete the registration process.



4. (Optional) Repeat step 3 for all supported connectors.
5. Click the **Save** button at the bottom or top of the Settings page to save changes.

3: Validate and update saved credentials

To help ensure uninterrupted data access, you (and your site users) must delete the previous saved credentials and add it again to use the custom OAuth client for the site.

1. Navigate to your **My Account Settings** page.
2. Under **Saved Credentials for Data Sources**, do the following:

   a. Click **Delete** next to the existing saved credentials for the connector whose custom OAuth client you configured in **Step 2** above.

   b. Next to connector name, click **Add** and follow the prompts to 1) connect to the custom OAuth client configured in **Step 2** above and 2) save the latest credentials.

4: Notify users to update their saved credentials

Make sure you notify your site users to update their saved credentials for the connector whose custom OAuth client you configured in **Step 2** above. Site users can use the procedure described in Update saved credentials to update their saved credentials.

Manage access tokens

After you configure the server for OAuth, you can allow users to manage their own access tokens in their profile settings, or you can manage the tokens centrally. For more information, see Allow Saved Access Tokens.

Configure Azure AD for OAuth and Modern Authentication

The Azure Synapse, Azure SQL Database, Azure Databricks, Azure Data Lake Gen2, OneDrive and SharePoint Online, and SharePoint Lists (JDBC) connectors support authentication through Azure AD by configuring an OAuth client for Tableau Server.

Note: Single use refresh tokens are not supported for OAuth connections to Tableau at this time. In most cases, you can set up your identity provider (such as Okta or Redshift IDC) to use rolling refresh tokens instead. For more information, see your provider's OAuth documentation.

Note: OAuth support for Azure AD is only supported with Microsoft SQLServer driver 17.3 and later.

Step 1: Register OAuth client for Azure

Follow the steps below to register and configure an OAuth application for Azure under a specific Azure tenant.

1. Sign in to the Azure portal.
2. If you have access to multiple tenants, select the tenant in which you want to register an application.
3. Search for and select **Azure Active Directory**.
4. Under **Manage**, select **App registrations,**and then select **New registration**.
5. Enter “Tableau Server OAuth”, or similar value as the **Name**.
6. Under the **Supported account types** field in the app registration page, select who can use this application.

   > **Note**: If you want to use the Client ID & Client Secret of your application for accounts under a different tenant, choose the second option (Multitenant).

7. Under **Redirect Uri** (optional) field, choose **Web**, and then enter the internet address of your server appended by the string, `/auth/add_oauth_token`.

   For example: `https://your_server_url.com/auth/add_oauth_token`

8. Select **Register**. After the registration completes, the Azure portal displays the app registration's Overview pane, which includes its Application (client) ID. Also referred to as *Client ID*, this value uniquely identifies your application in the Microsoft identity platform.
9. Copy the value, it will be used as the `[your_client_id]` field in the following steps.
10. Select **Certificates & secrets** on the left bar, and then choose**New client secret**.
11. Add a description of the secret.
12. Select **Client secret lifetime.**
13. Choose **Add** and then copy the secret. The secret will be used as `[your_client_secret]` in the following steps.
14. Select **API permissions** on the left bar.
15. Choose **Add permissions**.
16. Select **Microsoft Graph**.
17. Choose **Delegated permissions**.
18. Under **Select permissions**, select all OpenId permissions (email, offline_access, openid, and profile).
19. Choose **Add permissions**.
20. Add additional permissions. Follow the steps below for the connector(s) that you are enabling:

- Azure SQL database
    a. Click **Add a permission**.
    b. Select **My APIs**.
    c. Click **Azure SQL Database**, and then **Delegated permissions**.
    d. Select **user_impersonation**, and then click **Add permissions**.
- OneDrive and SharePoint Online
    a. Click **Add a permission**.
    b. Select **Microsoft Graph**.
    c. Click **Delegated permissions**.
    d. Under **Select permissions**, in the filter search box, enter and then add the following permissions:
        - Files.Read.All
        - Sites.Read.All
        - User.Read
- SharePoint lists (JDBC)
    a. Click **Add a permission**.
    b. Select **Microsoft Graph**.
    c. Click **Delegated permissions**.
    d. Under **Select permissions**, in the filter search box, enter and then add User.Read permission.
    e. Click **Add a permission** again.
    f. Select **SharePoint**.
    g. Click **Delegated permissions**.
    h. Expand the **AllSites** section, and then select and add the AllSites.Manage permission.

Step 2: Configure Tableau Server for Azure

Configuring Tableau Server requires running a Tableau Server Manager (TSM) command. Azure Data Lake Storage Gen2 requires a different set of commands than the common command that is run for Azure Synapse, Azure SQL Database, or Databricks.

Configure default OAuth client for Azure Data Lake Storage Gen2

To configure Tableau Server for Data Lake Storage Gen2, you must have the following configuration parameters:

- **Azure OAuth client ID:** The client ID is generated from the procedure in Step 1. Copy this value for `[your_client_id]` in the first tsm command.
- **Azure OAuth client secret:** The client secret is generated from the procedure in Step 1. Copy this value for `[your_client_secret]` in the second tsm command.
- **Tableau Server URL**: Enter your Tableau Server URL, such as `https://myco.com`. Copy this value for `[your_server_url]` in the third tsm command.

Run the following tsm commands to configure Tableau Server OAuth for Azure Data Lake Storage Gen2:

- `tsm configuration set -k oauth.azuredatalake_storage_gen2.client_id -v [your_client_id] --force-keys`
- `tsm configuration set -k oauth.azuredatalake_storage_gen2.client_secret -v [your_client_secret] --force-keys`
- `tsm configuration set -k oauth.azuredatalake_storage_gen2.redirect_uri -v http://[your_server_url]/auth/add_oauth_token --force-keys`
- `tsm pending-changes apply`

Configure default client for Azure Synapse, Azure SQL Database, or Databricks

To configure Tableau Server, you must have the following configuration parameters:

- **Azure OAuth client ID:** Generated from the procedure in Step 1. Copy this value for `[your_client_id]` in the tsm command.
- **Azure OAuthClient secret:** Generated from the procedure in Step 1. Copy this value for `[your_client_secret]` in the second tsm command.
- **Tableau Server URL:** This is your Tableau Server URL, such as `https://myserver.com`. Copy this value for `[your_server_url]` in the third tsm command.

- **Configuration ID:** The value for the `oauth.config.id` parameter in the following tsm command. Valid values:
  - Azure Synapse: `azure_sql_dw`
  - Azure SQL Database: `azure_sqldb`
  - Databricks: `databricks`

Run the following tsm commands to configure Azure AD for Azure Synapse, Azure SQL Database, or Databricks. For example, to set up Azure Synapse:

```
tsm configuration set -k oauth.config.clients -v "[{\"oauth.config.id\":\"azure_sql_dw\", \"oauth.config.client_id\":\"[your_client_id]\", \"oauth.config.client_secret\":\"[your_client_secret]\", \"oauth.config.redirect_uri\":\"[your_server_url]/auth/add_oauth_token\"}]" --force-keys

tsm pending-changes apply
```

Configure a default OAuth client for OneDrive and SharePoint Online

To configure Tableau Server for OneDrive and SharePoint Online, you must have the following configuration parameters:

- **Azure OAuth client ID**: The client ID is generated from the procedure in Step 1. Copy this value for [your_client_id] in the first tsm command.
- **Azure OAuth client secret**: The client secret is generated from the procedure in Step 1. Copy this value for [your_client_secret] in the second tsm command.
- **Tableau Server URL:** This is your Tableau Server URL, such as https://myco.com. Copy this value for [your_server_url] in the third tsm command.

Run the following tsm commands to configure Tableau Server OAuth for OneDrive and SharePoint Online:

- `tsm configuration set -k oauth.onedrive_and_sharepoint_online.client_id -v [your_client_id] --force-keys`

- `tsm configuration set -k oauth.onedrive_and_sharepoint_online.client_secret -v [your_client_secret] --force-keys`
- `tsm configuration set -k oauth.onedrive_and_sharepoint_online.redirect_uri -v http://[your_server_url]/auth/add_oauth_token --force-keys`
- `tsm pending-changes apply`

Configure a default OAuth client for SharePoint Lists (JDBC)

To configure Tableau Server for SharePoint Lists (JDBC), you must have the following configuration parameters:

- **Azure OAuth client ID**: The client ID is generated from the procedure in Step 1. Copy this value for [your_client_id] in the first tsm command.
- **Azure OAuth client secret**: The client secret is generated from the procedure in Step 1. Copy this value for [your_client_secret] in the first tsm command.
- **Tableau Server URL:** This is your Tableau Server URL, such as https://myco.com. Copy this value for [your_server_url] in the first tsm command.

Run the following tsm commands to configure Tableau Server OAuth SharePoint Lists (JDBC):

- `tsm configuration set -k oauth.config.clients -v "[{\"oauth.config.id\":\"cdata_sharepoint\", \"oauth.config.client_id\":\"[your_client_id]\", \"oauth.config.client_secret\":\"[your_client_secret]\", \"oauth.config.redirect_uri\":\"[your_server_url]/auth/add_oauth_token\"}]" --force-keys`
- `tsm pending-changes apply`

Configure a default OAuth client for OneDrive (deprecated)

To configure Tableau Server for OneDrive (deprecated), you must have the following configuration parameters:

- **Azure OAuth client ID:** The client ID is generated from the procedure in Step 1. Copy this value for [your_client_id] in the first tsm command.
- **Azure OAuth client secret:** The client secret is generated from the procedure in Step 1. Copy this value for [your_client_secret] in the second tsm command.

- **Tableau Server URL:** This is your Tableau Server URL, such as https://myco.com. Copy this value for [your_server_url] in the third tsm command.

To continue run the following tsm commands to configure Tableau Server OAuth for OneDrive (deprecated):

- `tsm configuration set -k oauth.onedrive.client_id -v [your_client_id] --force-keys`
- `tsm configuration set -k oauth.onedrive.client_secret -v [your_client_secret] --force-keys`
- `tsm configuration set -k oauth.onedrive.redirect_uri -v http://[your_server_url]/auth/add_oauth_token --force-keys`
- `tsm pending-changes apply`

Server Restart Scenarios

After you configure a default OAuth client, the following scenarios can occur.

- A restart prompt appears if the pending changes require a server restart.
- You can suppress the prompt using the `--ignore-prompt` option, but this doesn't stop the restart.
- If the changes don't require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Setting multiple connectors

If you have multiple connectors to set, you must include all of them in a single command. For example:

```
tsm configuration set -k oauth.config.clients -v "[{\"oauth.config.id\":\"azure_sql_dw\", \"oauth.config.client_id\":\"[your_client_id]\", \"oauth.config.client_secret\":\"[your_client_secret]\", \"oauth.config.redirect_uri\":\"[your_server_url]/auth/add_oauth_token\"}, {\"oauth.config.id\":\"azure_sqldb\", \"oauth.config.client_id\":\"[your_client_id]\", \"oauth.config.client_secret\":\"[your_client_secret]\", \"oauth.config.redirect_uri\":\"[your_server_url]/auth/add_oauth_token\"},
```

```
{\"oauth.config.id\":\"databricks\", \"oauth.config.client_id\":\"
[your_client_id]\", \"oauth.config.client_secret\":\"[your_client_
secret]\", \"oauth.config.redirect_uri\":\"[your_server_url]/au-
th/add_oauth_token\"}]" --force-keys

tsm pending-changes apply
```

Configure custom OAuth for a site

You can configure custom Azure Data Lake Storage Gen2, Azure Synapse, Azure SQL Database, Databricks OAuth, OneDrive and Sharepoint online, and Sharepoint Lists (JDBC) clients for a site.

Consider configuring a custom OAuth client to 1) override an OAuth client if configured for the server or 2) enable support for securely connecting to data that requires unique OAuth clients.

When a custom OAuth client is configured, the site-level configuration takes precedence over any server-side configuration and all new OAuth credentials created use the site-level OAuth client by default. No Tableau Server restart is required for the configurations to take effect.

Important: Existing OAuth credentials established before the custom OAuth client is configured are temporarily usable but both server administrators and users must update their saved credentials to help ensure uninterrupted data access.

1: Prepare the OAuth client ID, client secret, and redirect URL

Before you can configure the custom OAuth client, you need the information listed below. After you have this information prepared, you can register the custom OAuth client for the site.

- **OAuth client ID and client secret:** First register the OAuth client with the data provider (connector) to retrieve the client ID and secret generated for Tableau Server.
- **Redirect URL:** Note the correct redirect URL. You will need this during the registration process in **Step 2** below.

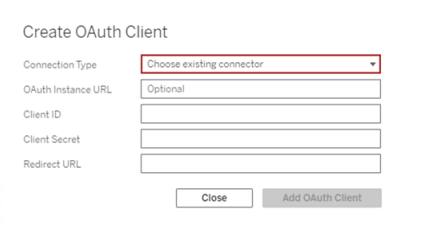
https://<your_server_name>.com/auth/add_oauth_token

For example, https://example.com/auth/add_oauth_token

2: Register the OAuth client ID and client secret

Follow the procedure described below to register the custom OAuth client to the site.

1. Sign in to your Tableau Server site using your admin credentials and navigate to the **Settings** page.
2. Under OAuth Clients Registry, click the **Add OAuth Client** button.
3. Enter the required information, including the information from **Step 1** above:
    a. For **Connection Type**, select the connector whose custom OAuth client you want to configure.
    b. **OAuth Instance URL** is required if multiple OAuth clients are being registered. Otherwise, it is optional.
    c. For **Client ID**, **Client Secret**, and **Redirect URL**, enter the information you prepared in **Step 1** above.
    d. Click the **Add OAuth Client** button to complete the registration process.



4. (Optional) Repeat step 3 for all supported connectors.
5. Click the **Save** button at the bottom or top of the Settings page to save changes.

3: Validate and update saved credentials

To help ensure uninterrupted data access, you (and your site users) must delete the previous saved credentials and add it again to use the custom OAuth client for the site.

1. Navigate to your **My Account Settings** page.
2. Under **Saved Credentials for Data Sources**, do the following:
    a. Click **Delete** next to the existing saved credentials for the connector whose custom OAuth client you configured in **Step 2** above.
    b. Next to connector name, click **Add** and follow the prompts to 1) connect to the custom OAuth client configured in **Step 2** above and 2) save the latest credentials.

4: Notify users to update their saved credentials

Make sure you notify your site users to update their saved credentials for the connector whose custom OAuth client you configured in **Step 2** above. Site users can use the procedure described in Update saved credentials to update their saved credentials.

Configure OAuth for Snowflake Connections

There are multiple ways you can configure OAuth for Snowflake connections, depending on which version of Tableau you are using and how many sites you are updating. This topic covers configuration for each available option.

- For all versions of Tableau up to and including version 2024.1, the Tableau Snowflake connector by default uses an OAuth proxy hosted in AWS (GALOP), which uses a common client ID and secret.
- Starting with Tableau 2020.4, you can optionally configure Tableau Server to use a new OAuth service that runs in the same location as that instance of Tableau. This requires providing your own client ID and secret, which is referred to as custom OAuth.
- Starting in 2024.2, the GALOP proxy will be deprecated and the instructions below for setting up custom OAuth will be **required**, as it will now use the local OAuth service for authentication.

- For Tableau Desktop and Tableau Cloud versions 2024.3 and beyond, you can configure a 3rd party IdP (external OAuth) for Snowflake. For more information, see External OAuth for Snowflake.

The benefits provided by Custom OAuth include:

- Improved security
- You can use OAuth in isolated environments that cannot connect to the OAuth Proxy (GALOP).
- You don't have to safe-list the GALOP IP addresses to run the OAuth flow in AWS PrivateLink or Azure Private Link VPCs.

Register OAuth Client With Snowflake

To use a custom OAuth setup in Tableau Server, you must first register your OAuth client and obtain a Client ID and Client Secret to complete the configuration. For Tableau Server versions 2024.2 and later, this step is required, regardless of which configuration option you are using. To register a custom OAuth client with Snowflake, follow the steps described in Configure Snowflake OAuth for Custom Clients.

After you register, you'll use the following Snowflake parameters to configure Tableau Server:

- Account instance URL
- Client ID
- Client secret
- Redirect URL

Note: The Redirect URL is the same when entered both on the Snowflake and Tableau sides. The format is:
https://your_server_url.com/auth/add_oauth_token
For example, https://example.com/auth/add_oauth_token

Option 1: Configure OAuth for Snowflake Connections using TSM

We recommend using this option when you need to update several sites at once.

Note: This configuration option is not available for use on Tableau Cloud.

1. (Versions 2024.1 and earlier) On the Tableau Server computer, run the following command to enable the Snowflake OAuth service:

   ```
   tsm configuration set -k native_api.enable_snowflake_
   privatelink_on_server -v true
   ```

   Note: For versions 2024.2 and newer, skip step 1 regardless of whether a Snowflake private connection is being used or not.

2. Copy, paste, and customize the following command in a text editor:

   Note: If you're making these configuration changes in Tableau Server 2021.1 and later, note that the format of oauth.snowflake.clients value has changed.

   ```
   tsm configuration set -k oauth.snowflake.clients -v " [{\"oau-
   th.snowflake.instance_url\":\"ht-
   tps://account.snowflakecomputing.com\",
   \"oauth.snowflake.client_id\":\"client_id_string\", \"oau-
   th.snowflake.client_secret\":\"client_secret_string\", \"oau-
   th.snowflake.redirect_uri\":\"http://your_server_
   url.com/auth/add_oauth_token\" }]"
   ```

The `oauth.snowflake.clients` key takes an array of key pairs. Each element in the key pair must be encapsulated by double quotes. Double quotes must be escaped as `\"`.

To specify multiple account instance URLs, separate each additional OAuth client wrapped in braces ({}) with a comma (,), as in this example:

```
tsm configuration set -k oauth.snowflake.clients -v " [{\"oau-
th.snowflake.instance_url\":\"ht-
tps://ac-
count.snowflakecomputing.com\",\"oauth.snowflake.client_
id\":\"client_id_string1\",\"oauth.snowflake.client_
secret\":\"client_secret_string1\",\"oauth.snowflake.redirect_
uri\":\"http://your_server_url.com/auth/add_oauth_token\" },
{\"oauth.snowflake.instance_url\":\"ht-
tps://ac-
count2.snowflakecomputing.com\",\"oauth.snowflake.client_
id\":\"client_id_string2\",\"oauth.snowflake.client_
secret\":\"client_secret_string2\",\"oauth.snowflake.redirect_
uri\":\"http://your_server_url.com/auth/add_oauth_token\" }]"
```

Replace the values for each key:

- Account instance URL:`oauth.snowflake.instance_url`
- Client ID: `oauth.snowflake.client_id`
- Client secret: `oauth.snowflake.client_secret`
- Redirect URL: `oauth.snowflake.redirect_uri`

Note: Before running the command, verify the syntax carefully. TSM won't validate this input.

Copy the command into TSM CLI and run the command.

3. Enter the following command to apply the changes:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Option 2: Configure OAuth for Snowflake Connections by Site

Note: This option is also available for Tableau Cloud starting with version 2024.2.

You can configure a custom Snowflake OAuth client at the site level by using the Tableau Server UI.

Consider configuring a custom OAuth client to 1) override an OAuth client if configured for the server or 2) enable support for securely connecting to data that requires unique OAuth clients.

When a custom OAuth client is configured, the site-level configuration takes precedence over any server-side configuration and all new OAuth credentials created use the site-level OAuth client by default. No Tableau Server restart is required for the configurations to take effect.

Important: Existing OAuth credentials established before the custom OAuth client is configured are temporarily usable but both server administrators and users must update their saved credentials to help ensure uninterrupted data access.

1: Prepare the OAuth client ID, client secret, and redirect URL

Before you can configure the custom OAuth client, you need the information listed below. After you have this information prepared, you can register the custom OAuth client for the site. For

more information, see the section **Register OAuth Client With Snowflake**, above.

- **OAuth client ID and client secret:** First register the OAuth client with the data provider (connector) to retrieve the client ID and secret generated for Tableau Server.
- **Redirect URL:** Note the correct redirect URL. You will need this during the registration process in **Step 2** below.

  https://<your_server_name>.com/auth/add_oauth_token

  For example, https://example.com/auth/add_oauth_token

2: Register the OAuth client ID and client secret

Follow the procedure described below to register the custom OAuth client to the site.

1. (Versions 2024.1 and earlier) On the Tableau Server computer, run the following command to enable the Snowflake OAuth service:

   ```
   tsm configuration set -k native_api.enable_snowflake_
   privatelink_on_server -v true
   ```

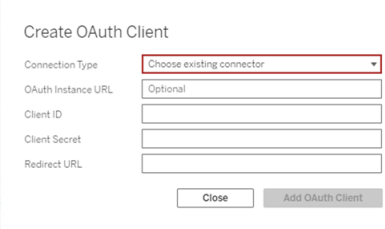
   Note: For versions 2024.2 and newer, skip step 1 regardless of whether a Snowflake private connection is being used or not.

2. Sign in to your Tableau Server site using your admin credentials and navigate to the **Settings** page.
3. Under OAuth Clients Registry, click the **Add OAuth Client** button.
4. Enter the required information, including the information from **Step 1** above:
   a. For **Connection Type**, select the connector whose custom OAuth client you want to configure.

b. **OAuth Instance URL** is required if multiple OAuth clients are being registered. Otherwise, it is optional.

c. For **Client ID**, **Client Secret**, and **Redirect URL**, enter the information you prepared in **Step 1** above.

d. Click the **Add OAuth Client** button to complete the registration process.



5. (Optional) Repeat step 3 for all supported connectors.
6. Click the **Save** button at the bottom or top of the Settings page to save changes.

3: Validate and update saved credentials

To help ensure uninterrupted data access, you (and your site users) must delete the previous saved credentials and add it again to use the custom OAuth client for the site.

1. Navigate to your **My Account Settings** page.
2. Under **Saved Credentials for Data Sources**, do the following:

   a. Click **Delete** next to the existing saved credentials for the connector whose custom OAuth client you configured in **Step 2** above.

   b. Next to connector name, click **Add** and follow the prompts to 1) connect to the custom OAuth client configured in **Step 2** above and 2) save the latest credentials.

4: Notify users to update their saved credentials

Make sure you notify your site users to update their saved credentials for the connector whose custom OAuth client you configured in **Step 2** above. Site users can use the procedure described in Update saved credentials to update their saved credentials.

Connect Tableau Server to the Salesforce Data Cloud

> Note: Data Cloud was previously called Customer Data Platform.

Tableau Server (version 2023.3 and later)

The Salesforce Data Cloud connector was released for Tableau Desktop and Tableau Prep in 2023.2, for Tableau Cloud in June 2023, and for Tableau Server in 2023.3. This connector seamlessly connects Tableau to Data Cloud, and is available for Tableau Desktop, Tableau Cloud, Tableau Server, and Tableau Prep. Compared to the earlier Customer Data Platform connector, the Salesforce Data Cloud connector is simpler to set up, recognizes Data Spaces, presents clearer object labels, and is powered by accelerated queries. See the steps below.

> Note: The Customer Data Platform connector was deprecated in Tableau Server 2023.3 and can't be used for new connections starting in Tableau Server 2024.2. Existing workbooks, data sources, and other assets that use the Customer Data Platform connector will continue to work until the connector is completely removed, typically 1-2 releases after deprecation. To ensure that existing assets continue to function, Tableau strongly recommends customers modify existing assets to use the Salesforce Data Cloud connector.

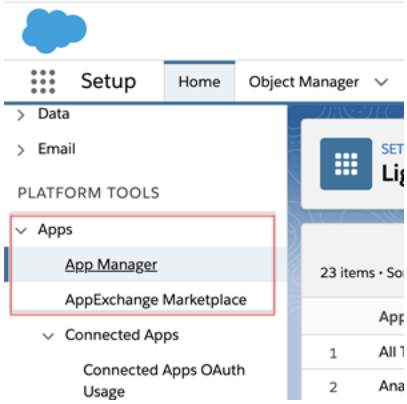
Step 1: Create a Salesforce connected app

The Salesforce Data Cloud connector requires you to create a connected app in Salesforce.

Both the Salesforce Data Cloud connector and the Salesforce connector can use the same connected app. Therefore, if you are already using the Salesforce connector, you only need to add the three Customer Data Platform scopes listed in step 7 to your existing connected app.

1. Sign in to your Salesforce.com developer account, click your user name in the upper-right, and then select **Setup**.

2. In the left navigation column, under **Apps**, select **App Manager**.



3. In the Connected Apps section, click **New Connected App**.

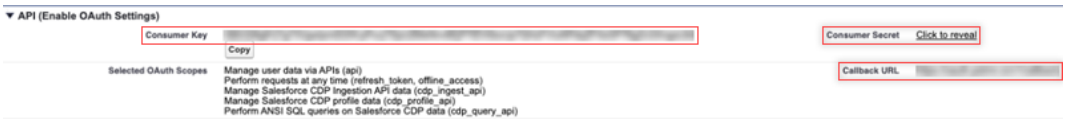
4. In **Basic Information**, give the app a name, tab through the APIfield so it will auto-populate in the correct format, and enter a contact email for the app.
5. In the **API [Enable OAuth Settings]** section, select **Enable OAuth Settings**.
6. In the new OAuth settings that appear, for **Callback URL**, type the fully qualified domain name of your server, using the `https` protocol, and append the following text to the URL: `auth/add_oauth_token`.
   For example:
   `https://www.your_tableau_server.com/auth/add_oauth_token`
7. Move the following items from **Available OAuth Scopes** to **Selected OAuth Scopes**:
   - **Access the identity URL service (id, profile, email, address, phone)**
   - **Manage user data via APIs (api)**
   - **Perform requests any time (refresh_token, offline access)**

- **Perform ANSI SQL queries on Customer Data Platform data (cdp_query_api)**
- **Manage Customer Data Platform profile data (cdp_profile_api)**
- **Manage Customer Data Platform Ingestion API data (cdp_ingest_api)**

8. Click **Save**.

After you save the app, Salesforce populates the API section with the following IDs that you will use to configure Tableau Server:

- **Consumer Key**
- **Consumer Secret**
- **Callback URL**



Step 2: Configure Tableau Server for Salesforce.com OAuth

Once your connected app is created in Salesforce and you have the Customer Key, Customer Secret, and the Callback URL, you can configure Tableau Server for Salesforce data connections and outputs, and Einstein Discovery.

1. On the Tableau Server computer, at a command prompt, run the following commands:
   ```
   tsm configuration set -k oauth.salesforce.client_id -v <your_customer_key>
   tsm configuration set -k oauth.salesforce.client_secret -v <your_customer_secret>
   tsm configuration set -k oauth.salesforce.redirect_uri -v <your_redirect_URL>
   ```
2. (Optional) To change the default login server, type the following command:
   ```
   tsm configuration set -k oauth.salesforce.server_base_url -v <URL>
   ```
3. Enter the following command to apply changes:
   ```
   tsm pending-changes apply
   ```
   If the pending changes require a server restart, the **pending-changes apply** command will display a prompt to let you know a restart will occur. This prompt displays

even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Configure custom OAuth for a site

For setup instructions, see the OAuth Connections topic.

Tableau Server (version 2023.1 and earlier)

Warning: The Customer Data Platform was deprecated in October 2023 and is unavailable in Tableau Server 2024.2 and later. To ensure that assets continue to function, Tableau recommends that customers modify existing assets to use the Salesforce Data Cloud connector.

This section describes how to connect from Tableau Server to the Salesforce Customer Data Platform through OAuth authentication.

The steps described in this section are required to use Salesforce Customer Data Platform data in Tableau Server.

Step 1: Set up the connector

1. Download the latest Salesforce Customer Data Platform connector (Salesforce_CDP.taco file) from the Tableau Exchange Connectors site.
2. Move the.taco file to the Tableau connector folder:

- Windows: `C:\Users[Windows User]\Documents\My Tableau Repository\Connectors`
- Linux: `/opt/tableau/connectors` OR `/var/opt/tableau_server-/data/tabsvc/vizqlserver/Connectors/`

3. Restart Tableau Server.

Step 2: Install the Customer Data Platform JDBC driver

Note: Tableau version 2023.1 for Server is only compatible with JDBC driver version 18 and above.

1. Download the latest JDBC driver (Salesforce-CDP-jdbc-[*version*].jar file) from the Salesforce CDP GitHub site: https://github.com/forcedotcom/Salesforce-CDP-jdbc/releases
2. Move the downloaded Salesforce-CDP-jdbc-[*version*].jar file to the following location:

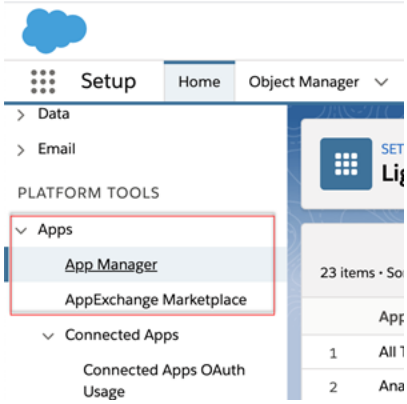
- Windows: `C:\Program Files\Tableau\Drivers`
- Linux: `/opt/tableau/tableau_driver/jdbc`

Step 3: Create a Salesforce connected app

After creating the API scopes, use the following procedure to create a Salesforce connected app to handle OAuth delegation from Tableau Server.

1. Sign in to your Salesforce Customer Data Platform account as an admin, click your username in the upper-right, and then select **Setup**.

2. In the left pane, under Apps, select **App Manager**.



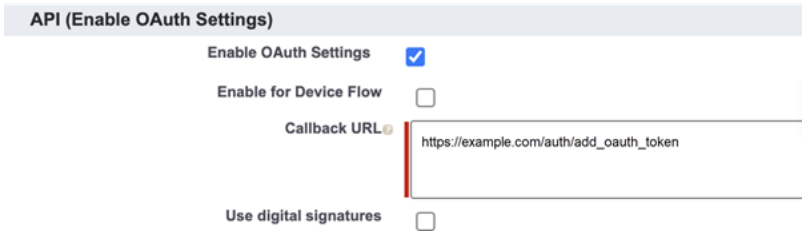
3. In the Connected Apps section, click **New Connected App**.

4. In **Basic Information**, give your connected app a name (for example, Example.com), tab through the API field so it self-populates in the correct format, and enter a contact email address for the app.

5. In the **API [Enable OAuth Settings]** section, select **Enable OAuth Settings**, and then do the following:

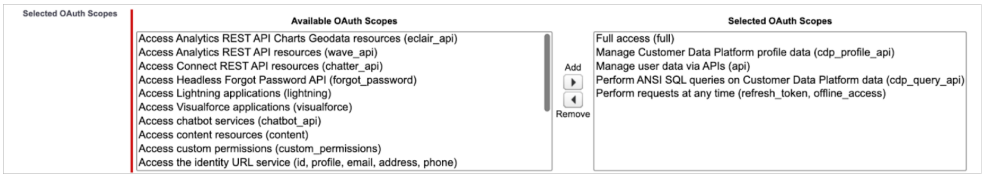
    a. In the new OAuth settings that appear, for **Callback URL**, type the fully qualified domain name (FQDN) of your Tableau Server, using the `https` protocol, and append the following text to the URL: `/auth/add_oauth_token`.

    For example: `https://example.com/auth/add_oauth_token`



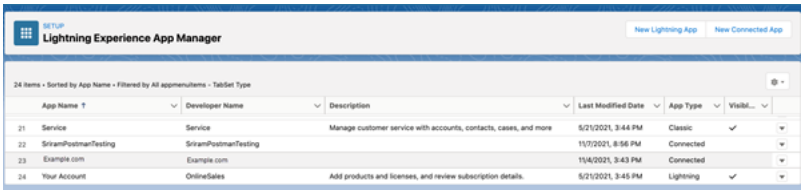
b. Move the following items from **Available OAuth Scopes** to **Selected OAuth Scopes**:

- **Manage user data via APIs (api)**
- **Perform requests on your behalf at any time (refresh_token, offline_access)**
- **Manage Customer Data Cloud profile data (cdp_profile_api)**
- **Perform ANSI SQL queries on Customer Data Platform data (cdp_query_api)**



6. When finished, click **Save**.

7. Go to your **App Manager** list, navigate to your connected app, click the dropdown arrow, and then select **Manage**.



After you save the app, the **API (Enable OAuth Settings)** section is populated with the following IDs that you'll use to configure Tableau Server:

- **Consumer Key**
- **Consumer Secret**
- **Callback URL**

> **Note:** Save your Consumer Key, Secret, and Callback URL for use later.

Use OAuth with the Customer Data Platform

Consider configuring a custom OAuth client to 1) override an OAuth client if configured for the server or 2) enable support for securely connecting to data that requires unique OAuth clients.

After the connected app is created in Salesforce and you have the Consumer Key, Consumer Secret, and the Callback URL, you can configure Tableau Server for the Customer Data Platform OAuth connections. To get started gather the following information.

- **Consumer Key:** The Consumer Key, also known as the client ID in Tableau, is generated from the procedure at the end of Step 4. Use this value for `[your_consumer_key]` in the following tsm command.
- **Consumer Secret:** The Consumer Secret, also known as the client secret in Tableau, is generated from the procedure at the end of Step 4. Use this value for `[your_consumer_secret]` in the following tsm command.
- **Callback URL:** The Callback URL, also know as the redirect URL in Tableau, is you Tableau Server URL `https://example.com` and "`/auth/add_oauth_token`" appended to it. Use this value for `[your_callback_url]` in the following tsm command.
- **Configuration ID:** The value for the `oauth.config.id` parameter you use in the following tsm: `customer_360_audience`

Use TSM Commands for OAuth Setup

Run the following tsm commands to configure OAuth.

```
tsm configuration set -k oauth.config.clients -v "[{\"oau-
th.config.id\":\"customer_360_audience\", \"oauth.config.client_
```

```
id\":\"[your_consumer_key]\", \"oauth.config.client_secret\":\"
[your_consumer_secret]\", \"oauth.config.redirect_uri\":\"[your_
callback_url]\"}]" --force-keys

tsm pending-changes apply
```

Setting multiple connectors

If you have multiple connectors to set, you must include all of them in a single command. For example:

```
tsm configuration set -k oauth.config.clients -v "[{\"oau-
th.config.id\":\"custom_360_audience\", \"oauth.config.client_
id\":\"[your_consumer_key]\", \"oauth.config.client_secret\":\"
[your_consumer_secret]\", \"oauth.config.redirect_uri\":\"[your_
callback_url]\"}, {\"oauth.config.id\":\"dremio\", \"oau-
th.config.client_id\":\"[your_client_id]\", \"oauth.config.client_
secret\":\"[your_client_secret]\", \"oauth.config.redirect_uri\":\"
[your_server_url]/auth/add_oauth_token\"}, {\"oau-
th.config.id\":\"azure_sql_dw\", \"oauth.config.client_id\":\"
[your_client_id]\", \"oauth.config.client_secret\":\"[your_client_
secret]\", \"oauth.config.redirect_uri\":\"[your_server_url]/au-
th/add_oauth_token\"}, {\"oauth.config.id\":\"azure_sqldb\", \"oau-
th.config.client_id\":\"[your_client_id]\", \"oauth.config.client_
secret\":\"[your_client_secret]\", \"oauth.config.redirect_uri\":\"
[your_server_url]/auth/add_oauth_token\"}]" --force-keys

tsm pending-changes apply
```

Step 1: Register OAuth client ID and client secret

Complete the following procedure to register the custom OAuth client to your site.

1. Sign into Tableau Server using your site admin credentials and navigate to the **Settings** page.

2. Under **OAuth Clients Registry**, select the **Add OAuth Client** button.
3. For **Connection Type**, select Customer Data Platform.
4. For OAuth Provider, select **Custom IDP**.
5. Enter the **Client ID**.
6. Enter the **Client Secret**.
7. Enter the **Redirect URL**.
8. For Choose OAuth Config File, select the **Choose a file** button to upload the config file.
9. Select the **Add OAuth Client** button to complete the registration process.
10. Select the Save button at the bottom or top of the Settings page to save changes.

Step 2: Validate and update saved credentials

To help ensure uninterrupted data access, you (and your site users) must delete any previous saved credentials and add them again.

1. Navigate to your My Account Settings page.
2. Under **Saved Credentials for Data Sources**, select **Delete** next to the existing saved credentials.
3. Next to the same connector, select **Add**.
4. Follow the prompts to connect to the Customer Data Platform connector.
5. Select **Save**.

Step 3: Notify users to update their saved credentials

Make sure you notify your site users to update their saved credentials for the Customer Data Platform connector. Site users can use the procedure described in Manage Saved Credentials for Data Connections to update their saved credentials.

See also

- Salesforce Help: Set Up Tableau in Customer Data Platform
- Salesforce Help: Enable Customer Data Platform in Tableau
- Salesforce Help: Using Customer Data Platform Data in Tableau

External OAuth for Snowflake

Starting in Tableau 2024.3, you can use OAuth 2.0/OIDC to federate identity from an external identity provider to Snowflake.

Depending on the identity provider, there are different steps needed to configure the integration. This is a high-level overview intended to guide your configuration without providing the necessary details you'll find in your identity provider documentation. It is assumed you are familiar with configuring OAuth and understand the technical details required in setting up authentication with an external identity provider.

Configure IDP on Snowflake

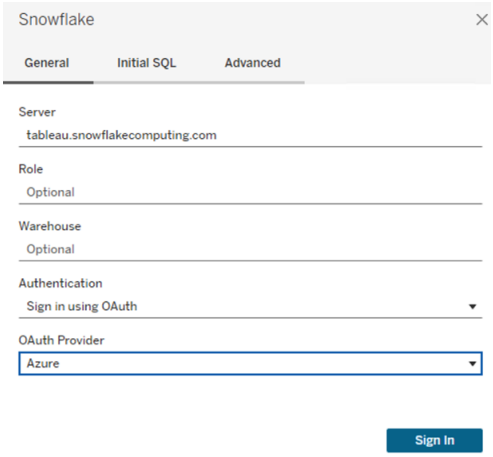
For information on configuring your IDP, see External OAuth overview in Snowflake's help system.

Configure the IDP on Tableau

1. Create OAuth clients on the IDP for Tableau Desktop, and on Tableau Cloud or Tableau Server. The Desktop client enables PKCE and uses `http://localhost` redirects.
2. Create the Tableau OAuth config file. For details on how to do this, see OAuth Configuration and Usage on github, and examples here. We welcome additional examples for other IDPs.
    A. Be sure to prefix the Tableau OAuth config IDs with “custom_”.
    B. If your IDP supports dynamic localhost port, disable `OAUTH_CAP_FIXED_PORT_IN_CALLBACK_URL`. If your IDP does not support this, make sure to add several localhost callback URLs to the allowlist in the config file and on the IDP.
3. Install the new Tableau OAuth configuration files in the OAuthConfigs folder associated with each application on desktop hosts (Tableau Desktop, Tableau Prep Builder, Tableau Bridge), and on each Tableau Server and Tableau Cloud site that will be using OAuth via site settings page. For more details, see Custom OAuth Configs on Desktop and Site Level OAuth Clients.

Connect to Snowflake

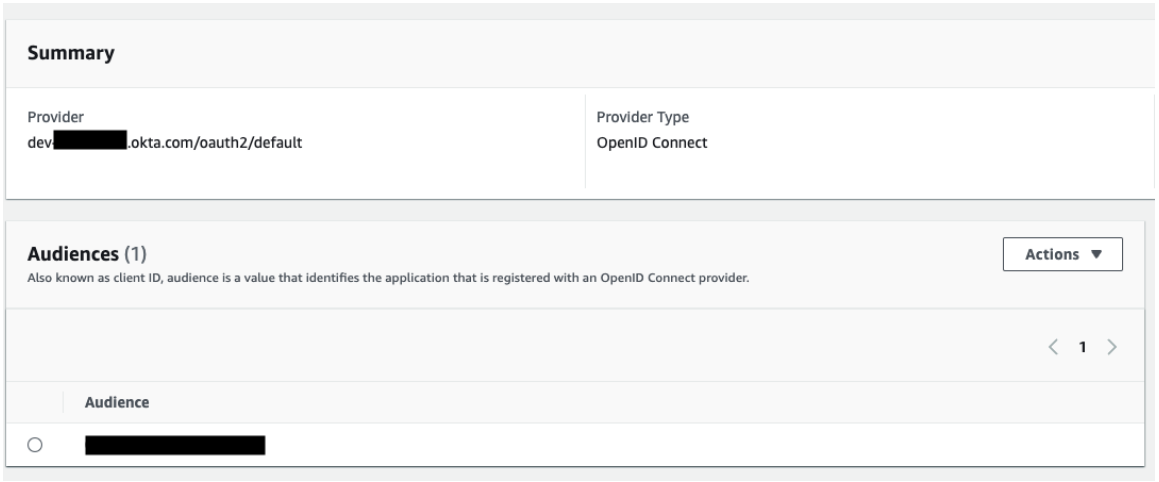
When connecting, you must select OAuth and choose the OAuth configuration installed earlier.



Okta

If using Okta it's better to use a "custom authorization server" rather than the "org authorization server." The custom authorization servers are more flexible. There's a custom authorization server created by default, which is called "default". The authorization URL should look like this:

```
https://${yourOktaDomain}/oauth2/{authServerName}/v1/authorize
```



Hyper Query Processing (Beta)

> **Note:** Since Hyper Query Processing is a Beta release the name can change before general release.

The Customer Data Platform (CDP) connector is now powered with the Hyper Query Processing Engine. It supports interactive analytics with fast data query processing and simplifies the connect-to- data experience for faster data exploration. The Hyper query processing engine speeds up query time when querying Customer Data Platform data from Tableau with live connections and Tableau data extracts.

Beta Feature: Hyper Query Processing beta feature offers extra functionality at no cost. You can opt to try this service with your sole discretion. Any use of the Beta Service is subject to the applicable Beta Services Terms provided at Agreements and Terms.

Enable Hyper Query Processing in Tableau Server

Complete the following steps to enable Hyper Query Processing.

1. Connect to Salesforce CDP.

2. Follow the directions on the screen to install the hyper jdbc driver.

`-Salesforce-cdp-hyper-beta-1.15.0.jar` or download the .jar file from here.

3. Move the downloaded file to any of the following directory paths depending on your system.

Windows - `C:\Program Files\Tableau\Drivers`

macOS - `/Users/[user]/Library/Tableau/Drivers`

4. After you've moved the driver to the correct path, connect again. Your Customer Data Platform connector is ready to use in Tableau Server.

Note: The Hyper Query Processing support is a beta release. It's available for Tableau Server. Tableau Desktop and Tableau Online will be supported from December 2022. These dates are subject to change.

See Also

- Tableau Help: Connect Tableau Server to the Customer Data Platform
- Salesforce Help: Set Up Tableau in Customer Data Platform
- Salesforce Help: Enable Customer Data Platform in Tableau
- Salesforce Help: Using Customer Data Platform Data in Tableau

Set Up Amazon Redshift IAM OAuth

Starting in Tableau 2023.3.2, you can use OAuth 2.0/OIDC to federate identity from an external identity provider to Amazon Redshift.

These instructions are for the older AWS IAM service. For IAM IDC integration see Set Up Amazon Redshift IAM Identity Center OAuth.

Depending on the identity provider, there are different steps needed to configure the integration. This is a high-level overview. Tableau cannot provide detailed instructions for how to configure AWS or the IDP, but the general approach is described below.

Note: Single use refresh tokens are not supported for OAuth connections to Tableau at this time. In most cases, you can set up your identity provider (such as Okta or Redshift IDC) to use rolling refresh tokens instead. For more information, see your provider's OAuth documentation.

Step 1: Configure the IDP

1. Create OAuth clients on the IDP for Tableau Desktop, and Tableau Server or Tableau Cloud. The Desktop client should enable `PKCE` and use `http://localhost` redirects.

2. Add custom claims to use for authorization to roles. In particular, if you are using original IAM, you may want to add claims for `DbUser` and `DbGroups`. These can be used in your IAM policies later.

3. Create the Tableau OAuth config files. See documentation on GitHub, and examples here. We welcome examples for other IDPs.

    a. Be sure to prefix the Tableau OAuth config IDs with "`custom_`".

    b. If your IDP supports dynamic localhost port then disable `OAUTH_CAP_FIXED_PORT_IN_CALLBACK_URL`. If your IDP does not support this, make sure to add several localhost callback URLs to the allowlist in the config file and on the IDP.

4. Install the new Tableau OAuth configuration files in the `OAuthConfigs` folder associated with each application on desktop hosts (Tableau Desktop, Tableau Prep Builder, Tableau Bridge), and on each Tableau Server and Tableau Cloud site that will be using OAuth.

Configure IDP on AWS

1. Create the IDP model on the AWS. See Amazon docs Web Identity Federation and Create OIDC Identity Provider.

2. Create roles and policies specifically for the IDP. See Create Role for OIDC in the AWS docs.

Configure Roles for Redshift Users

Attach the policies needed for Redshift. You may use custom claims from the token to authorize to roles. There are several examples with SAML in the AWS documentation. These can be easily adapted to OAuth. In the case of OAuth, the claims are just "`DbUser`", "`DbGroups`", etc.

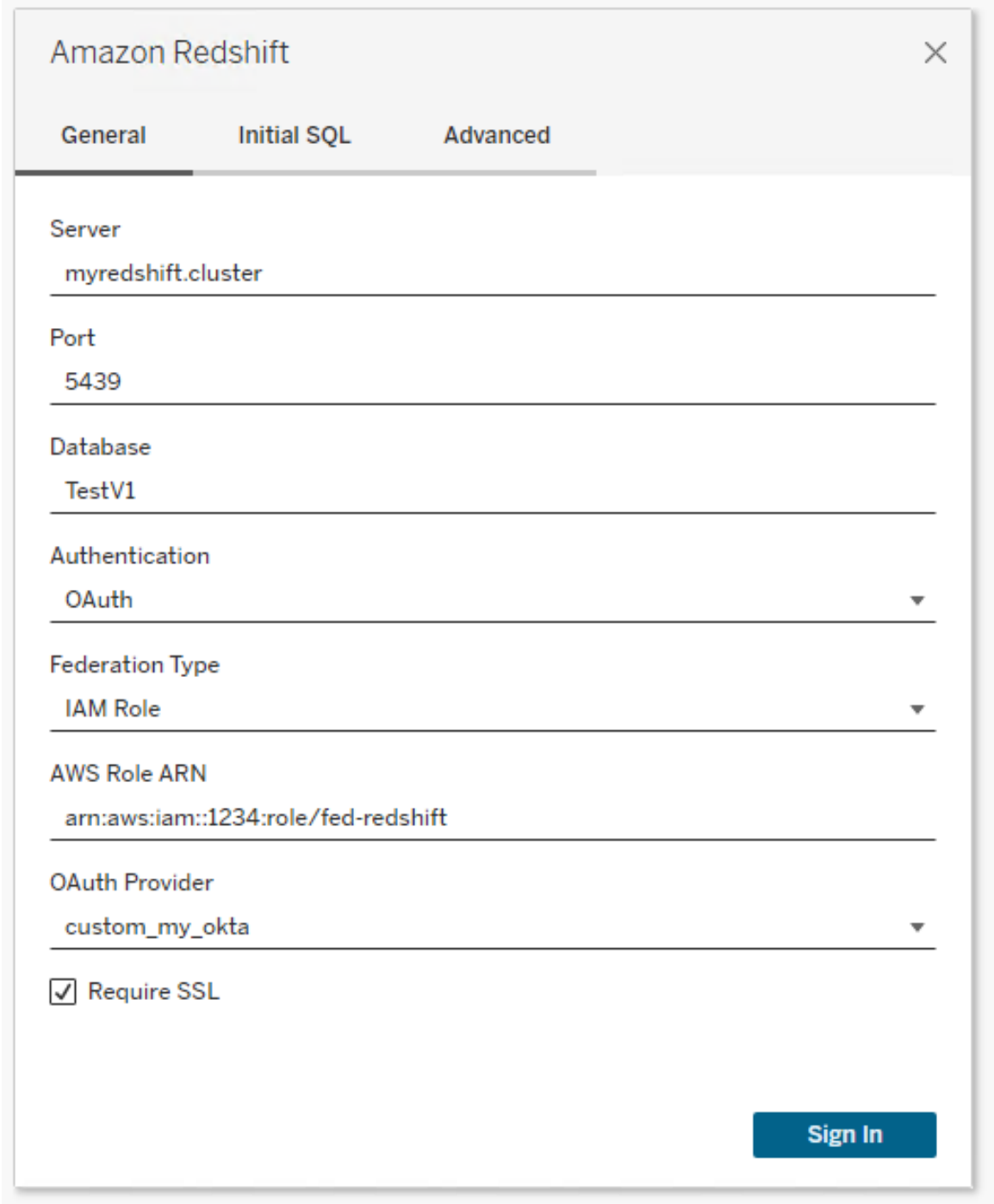
Here is an example of the policy from the AWS documentation:

```
{
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": "redshift:GetClusterCredentials",
        "Resource": [
            "arn:aws:redshift:us-west-1:123456789012:dbname:cluster-
identifier/dev",
            "arn:aws:redshift:us-west-1:123456789012:dbuser:cluster-
identifier/${redshift:DbUser}",
            "arn:aws:redshift:us-west-1:123456789012:cluster-
:cluster-identifier"
        ],
        "Condition": {
            "StringEquals": {
                "aws:userid": "AROAJ2UCCR6DPCEXAMPLE:${red-
shift:DbUser}@example.com"
            }
        }
    },
    {
        "Effect": "Allow"
        "Action": "redshift:CreateClusterUser",
        "Resource": "arn:aws:redshift:us-west-1:12345:d-
buser:cluster-identifier/${redshift:DbUser}"
    },
    {
        "Effect": "Allow",
```

```
        "Action": "redshift:JoinGroup",
        "Resource": "arn:aws:redshift:us-west-1:12345:d-
bgroup:cluster-identifier/my_dbgroup"
    },
    {
        "Effect": "Allow",
        "Action": [
            "redshift:DescribeClusters",
            "iam:ListRoles"
        ],
        "Resource": "*"
    }
]
}
```

Connect to Redshift

The user must specify the **role ARN** to assume, and select the OAuth config installed earlier.



When properly configured, the user will be redirected to the IDP to authenticate and authorize tokens for Tableau. Tableau will receive openid and refresh tokens. AWS is able to validate the token and signature from the IDP, extract the claims from the token, look up the mapping of

claims to IAM role, and either permit or block Tableau from assuming the role on the user's behalf. (in other words, AssumeRoleWithWebIdentity).

Tokens

By default Redshift OAuth IAM passes the ID token to the driver. For on-premise customers, including those using Tableau Bridge, you may use a TDC file to pass the access token instead.

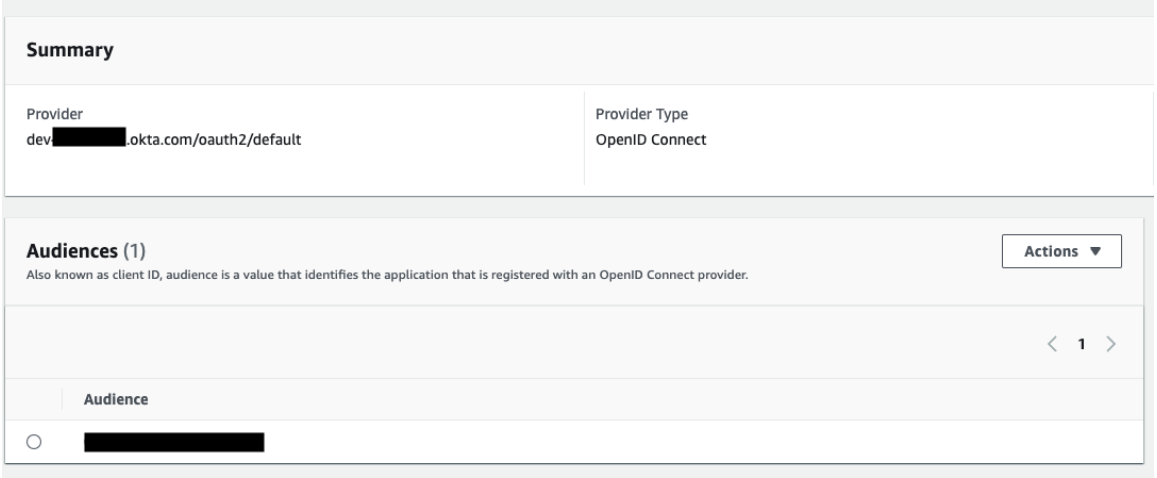
```
<connection-customization class='redshift' enabled='true' ver-
sion='10.0'>
       <vendor name='redshift' />
       <driver name='redshift' />
       <customizations>
              <customization name='CAP_OAUTH_FEDERATE_ACCCESS_TOKEN' value-
='yes'/>
       </customizations>
</connection-customization>
```

For more information about configuring and installing .tdc files, see Customize and Tune a Connection and Using a .tdc File with Tableau Server.

Okta

If using Okta it's better to use a "custom authorization server" rather than the "org authorization server." The custom authorization servers are more flexible. There is a custom authorization server created by default, which is called "default". The authorization URL should look like this:

```
https://${yourOktaDomain}/oauth2/{authServerName}/v1/authorize
```



Update Driver

For Redshift OAuth using the original IAM service, you may use either:

- Redshift ODBC v1 driver starting with version 1.59, which can be downloaded from https://docs.aws.amazon.com/redshift/latest/mgmt/configure-odbc-connection.html.
- Redshift ODBC v2 driver starting with version 2.0.1.0, which can be downloaded from https://github.com/aws/amazon-redshift-odbc-driver/tags. Note there is no v2 driver for OSX.

Troubleshooting

The best way to diagnose errors is to remove Tableau from the picture. You can instead test using the driver manager or a similar tool. This is just for troubleshooting - you shouldn't use a DSN or the "Other ODBC" connector for regular usage of this feature. To help ensure a valid test, the parameters should be the same as shown below, except for the cluster information, database, token, and namespace.

If you see an error message about invalid/expired token coming from the driver on the first connection (it will have a SQLState error code like [28000] or [08001] in the error message), then Tableau successfully completed the OAuth flow, and failed in the driver. This means there is a

misconfiguration on either the AWS side or the IDP side. There may also be permissions or authorization errors returned from the driver, which is also out of Tableau's control.

Before you begin testing, you first need to get an access token (the default for IAM IDC) or refresh token (if customized) to send to the driver.

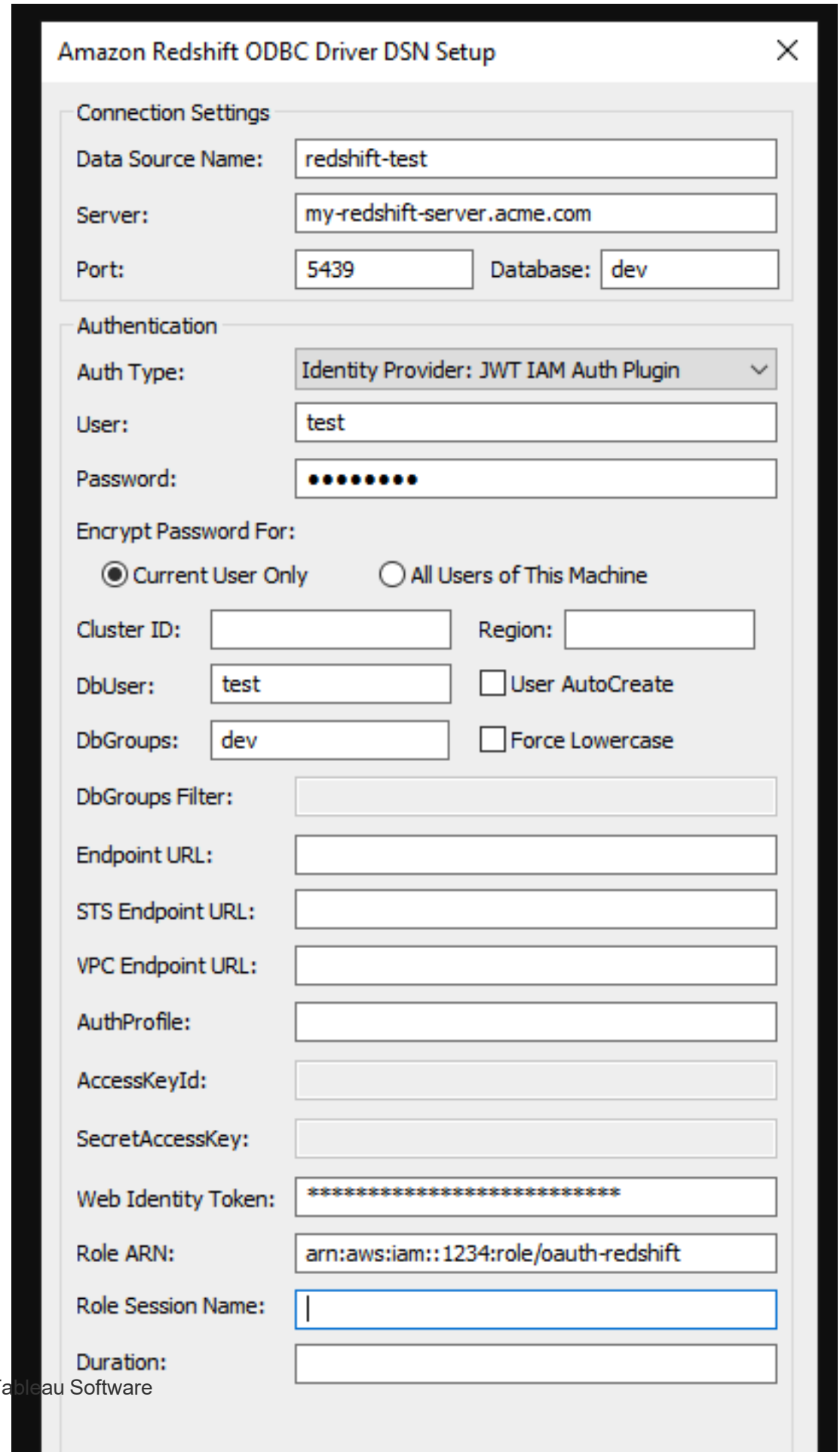
Here is an example with Okta. Almost all IDPs have a way to do this which is quite similar. Note that to use this flow you need to have enabled resource owner password grant type. Substitute the IDP URL, client secret, client ID, username, and password.

```
curl -X POST "https://OKTA_URL/v1/token" \
-H 'accept: application/json' \
-H "Authorization: Basic $(echo -n 'CLIENTID:CLIENTSECRET' |
base64)" \
-H "Content-Type: application/x-www-form-urlencoded" \
-d "grant_type-
e=password&username=USER&password=PASSWORD&scope=openid"
```

Once you have the token, you can use a DSN to test. For Windows, you can use the ODBC driver manager. On Mac you can use the iODBC driver manager UI. On Linux you can use the isql command line tool that is included with Tableau Server in the customer-bin folder.

Tableau recommends you do not use other plugins to test, because they may not work in a server environment. They either use a fixed AWS profile, or require direct access to a browser.

Below is an example of using the ODBC driver manager on Windows.


Amazon Redshift ODBC Driver DSN Setup
Connection Settings
Data Source Name: redshift-test
Server: my-redshift-server.acme.com
Port: 5439
Database: dev
Authentication
Auth Type: Identity Provider: JWT IAM Auth Plugin
User: test
Password: ••••••••
Encrypt Password For:
Current User Only
All Users of This Machine
Cluster ID:
Region:
DbUser: test
User AutoCreate
DbGroups: dev
Force Lowercase
DbGroups Filter:
Endpoint URL:
STS Endpoint URL:
VPC Endpoint URL:
AuthProfile:
AccessKeyId:
SecretAccessKey:
Web Identity Token: **********
Role ARN: arn:aws:iam::1234:role/oauth-redshift
Role Session Name:
Duration:

Set Up Amazon Redshift IAM Identity Center OAuth

Starting in Tableau 2023.3.2, you can use OAuth 2.0/OIDC to federate identity from an external identity provider to Amazon Redshift.

These instructions are for the newer AWS IAM IDC service. For original IAM integration see Set Up Amazon Redshift IAM OAuth.

Depending on the identity provider, there are different steps needed to configure the integration. This is a high-level overview. Tableau cannot provide detailed instructions for how to configure AWS or the IDP, but this is the general approach.

For some detailed examples of implementing authentication with Redshift, see "Integrate Tableau and Okta with Amazon Redshift using AWS IAM Identity Center" and "Integrate Tableau and Microsoft Entra ID with Amazon Redshift using AWS IAM Identity Center".

Note: Single use refresh tokens are not supported for OAuth connections to Tableau at this time. In most cases, you can set up your identity provider (such as Okta or Redshift IDC) to use rolling refresh tokens instead. For more information, see your provider's OAuth documentation.

Step 1: Configure the IDP

1. Create OAuth clients on the IDP for Tableau Desktop and Tableau Server or Tableau Cloud. The Desktop client should enable `PKCE` and use `http://localhost` redirects.
2. Add any required custom claims to use for authorization to roles.
3. Create the Tableau OAuth config files. See documentation on GitHub, and examples. We welcome examples for other IDPs.

a. Be sure to prefix the Tableau OAuth config IDs with "`custom_`".

b. If your IDP supports dynamic localhost port then disable `OAUTH_CAP_FIXED_PORT_IN_CALLBACK_URL`. If it does not, make sure to add several localhost callback URLs to the allowlist in the config file and on the IDP.

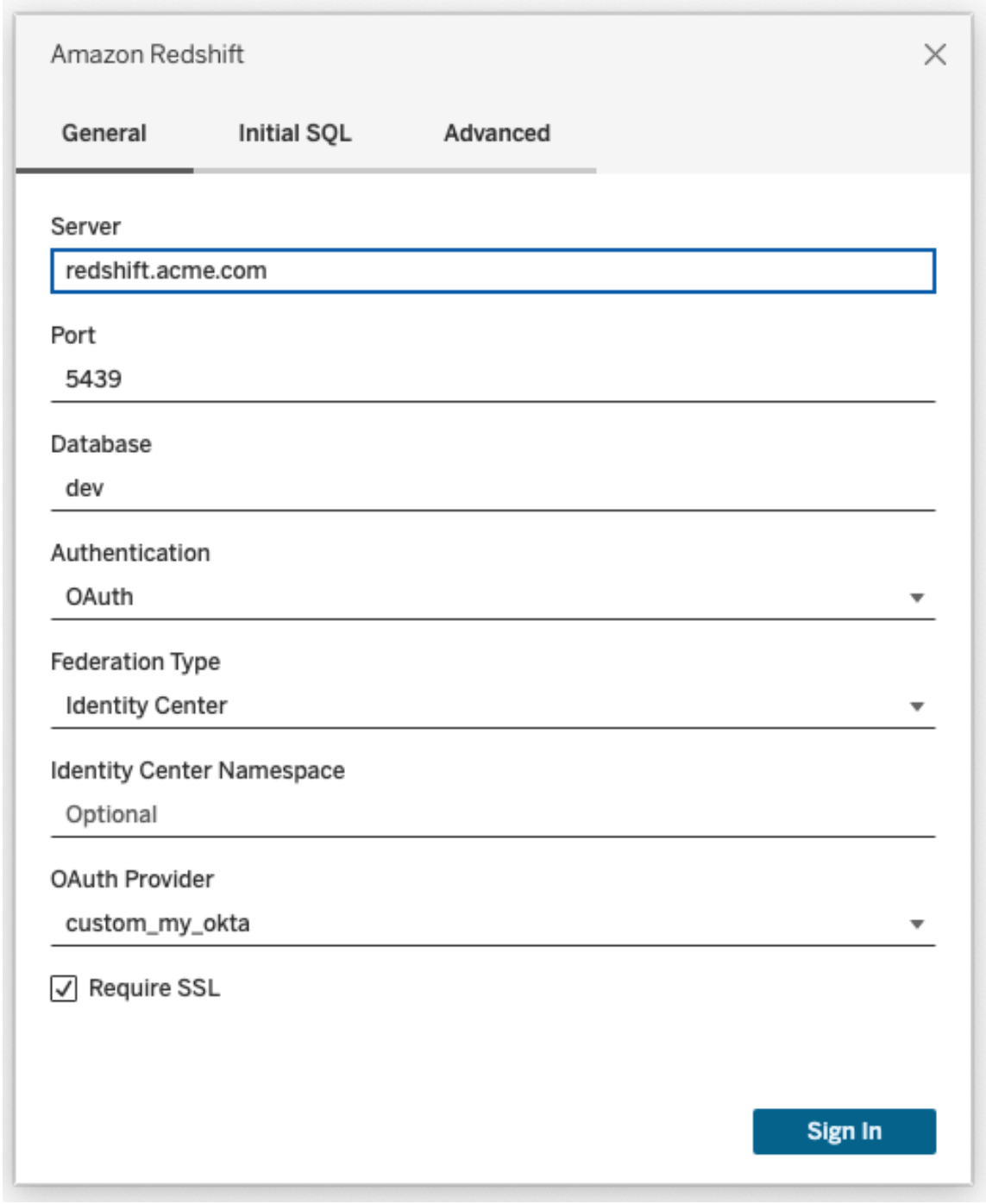
4. Install the new Tableau OAuth configuration files in the `OAuthConfigs` folder associated with each application on desktop hosts (Tableau Desktop, Tableau Prep Builder, Tableau Bridge), and on each Tableau Server and Tableau Cloud site that will be using OAuth.

Step 2: Configure IDP and Roles on AWS

See your AWS documentation for information on doing this.

Step 3: Connect to Redshift

1. Connect to Redshift.
2. Select OAuth for **Authentication**.
3. Select Identity Center for **Federation Type**.
4. (Optional) Specify the **Identity Center Namespace** if necessary.



When correctly configured, you will be redirected to the IDP to authenticate and authorize tokens for Tableau. Tableau will receive an access token and refresh tokens. It will send the access token to the driver for authentication.

Tokens

By default Redshift OAuth to IAM IDC passes the access token to the driver. For on-premise customers, including those using Tableau Bridge, you may use a TDC file to pass the ID token instead.

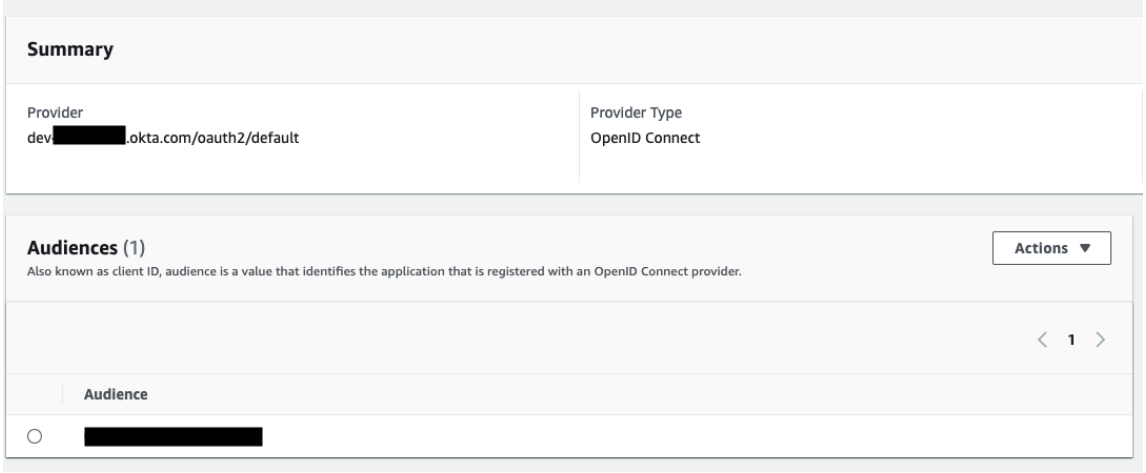
```
<connection-customization class='redshift' enabled='true' ver-
sion='10.0'>
       <vendor name='redshift' />
       <driver name='redshift' />
       <customizations>
              <customization name='CAP_OAUTH_FEDERATE_ID_TOKEN' value='yes'/>
       </customizations>
</connection-customization>
```

For more information about configuring and installing .tdc files, see Customize and Tune a Connection and Using a .tdc File with Tableau Server.

Okta

If you are using Okta, it's better to use a "custom authorization server" instead of the "org authorization server." The custom authorization servers are more flexible. A custom authorization server is created by default and called "default". The authorization URL should look like this:

```
https://${yourOktaDomain}/oauth2/{authServerName}/v1/authorize
```



Update the driver

For Redshift OAuth using the IAM IDC service, you need to use at least version 2.x of the ODBC driver. Download the latest version of the Redshift ODBC driver found on https://-github.com/aws/amazon-redshift-odbc-driver/tags. Note that there is no v2 driver yet for OSX.

Troubleshooting Redshift IAM IDC OAuth

The best way to diagnose errors is to remove Tableau from the picture. You can instead test using the driver manager or a similar tool. This is just for troubleshooting - you shouldn't use a DSN or the "Other ODBC" connector for regular usage of this feature. To help ensure a valid test, the parameters should be the same as shown below, except for the cluster information, database, token, and namespace.

If you see an error message about invalid/expired token coming from the driver on the first connection (it will have a SQLState error code like [28000] or [08001] in the error message), then Tableau successfully completed the OAuth flow, and failed in the driver. This means there is a misconfiguration on either the AWS side or the IDP side. There may also be permissions or authorization errors returned from the driver, which is also out of Tableau's control.

Before you begin testing, you first need to get an access token (the default for IAM IDC) or refresh token (if customized) to send to the driver.

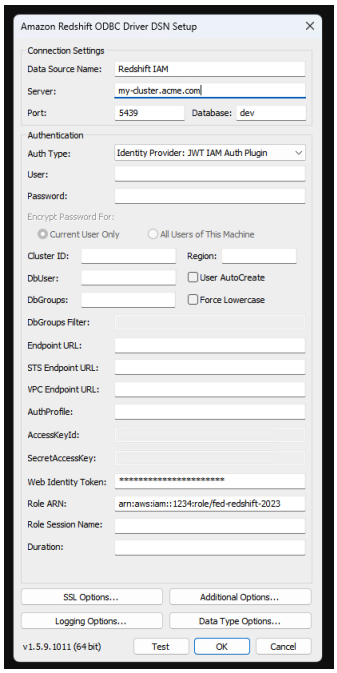
Here is an example with Okta. Almost all IDPs have a way to do this which is quite similar. Note that to use this flow you need to have enabled resource owner password grant type. Substitute the IDP URL, client secret, client ID, username, and password.

```
curl -X POST "https://OKTA_URL/v1/token" \
-H 'accept: application/json' \
-H "Authorization: Basic $(echo -n 'CLIENTID:CLIENTSECRET' |
base64)" \
-H "Content-Type: application/x-www-form-urlencoded" \
-d "grant_type-
e=password&username=USER&password=PASSWORD&scope=openid"
```

Once you have the token, you can use a DSN to test. For Windows, you can use the ODBC driver manager. On Linux you can use the isql command line tool that is included with Tableau Server in the customer-bin folder.

Tableau recommends you do not use other plugins to test, because they may not work in a server environment. They either use a fixed AWS profile, or require direct access to a browser.

Below is an example of using the ODBC driver manager on Windows.



Set Up OAuth for Dremio

This topic describes how to set up Dremio data sources for OAuth authentication. Complete these steps for each Tableau Server instance.

Setting up OAuth for Dremio consists of the following tasks:

1. Register an OAuth client with Dremio.
2. Use the information you obtained in Step 1 to configure Tableau Server for Dremio OAuth.
3. (Optional) Configure site-specific OAuth.

Step 1: Register OAuth client in Dremio

Use the Identity Providers topic in the Dremio documentation to configure a Dremio-supported IdP to get the OAuth client ID and secret configuration parameters needed to configure Tableau Server for Dremio OAuth.

Step 2: Configure Tableau Server for Dremio OAuth

To configure Tableau Server for Dremio OAuth, you will use the parameters listed below in the tsm command that follows.

- **Dremio client ID:** The client ID is generated from the registration process in Step 1. Copy this value for `[your_client_id]` in the tsm command.
- **Dremio client secret:** The client secret is generated from the procedure in Step 1. Copy this value for `[your_client_secret]` in the tsm command.
- **Tableau Server URL:** This is your Tableau Server URL, such as `https://myco.com`. Copy this value for `[your_server_url]` in the tsm command.
- **Configuration ID:** This is the value for the `oauth.config.id` parameter you will use in the tsm command: `dremio`

Run the following tsm commands to configure OAuth for Dremio:

```
tsm configuration set -k oauth.config.clients -v "[{\"oau-
th.config.id\":\"dremio\", \"oauth.config.client_id\":\"[your_cli-
ent_id]\", \"oauth.config.client_secret\":\"[your_client_secret]\",
\"oauth.config.redirect_uri\":\"[your_server_url]/auth/add_oauth_
token\"}]" --force-keys

tsm pending-changes apply
```

Setting multiple connectors

If you have multiple connectors to set, you must include all of them in a single command. For example:

```
tsm configuration set -k oauth.config.clients -v "[{\"oau-
th.config.id\":\"dremio\", \"oauth.config.client_id\":\"[your_cli-
ent_id]\", \"oauth.config.client_secret\":\"[your_client_secret]\",
\"oauth.config.redirect_uri\":\"[your_server_url]/auth/add_oauth_
token\"}, {\"oauth.config.id\":\"customer_360_audience\",
```

```
\"oauth.config.client_id\":\"[your_client_id]\", \"oau-
th.config.client_secret\":\"[your_client_secret]\", \"oau-
th.config.redirect_uri\":\"[your_server_url]/auth/add_oauth_
token\"}, {\"oauth.config.id\":\"azure_sql_dw\", \"oau-
th.config.client_id\":\"[your_client_id]\", \"oauth.config.client_
secret\":\"[your_client_secret]\", \"oauth.config.redirect_uri\":\"
[your_server_url]/auth/add_oauth_token\"}, {\"oau-
th.config.id\":\"azure_sqldb\", \"oauth.config.client_id\":\"[your_
client_id]\", \"oauth.config.client_secret\":\"[your_client_
secret]\", \"oauth.config.redirect_uri\":\"[your_server_url]/au-
th/add_oauth_token\"}]" --force-keys

tsm pending-changes apply
```

Configure custom OAuth for a site

You can configure custom Dremio OAuth for a site.

Consider configuring a custom OAuth client to 1) override an OAuth client if configured for the server or 2) enable support for securely connecting to data that requires unique OAuth clients.

When a custom OAuth client is configured, the site-level configuration takes precedence over any server-side configuration and all new OAuth credentials created use the site-level OAuth client by default. No Tableau Server restart is required for the configurations to take effect.

Important: Existing OAuth credentials established before the custom OAuth client is configured are temporarily usable but both server administrators and users must update their saved credentials to help ensure uninterrupted data access.

1: Prepare the OAuth client ID, client secret, and redirect URL

Before you can configure the custom OAuth client, you need the information listed below. After you have this information prepared, you can register the custom OAuth client for the site.

- **OAuth client ID and client secret:** First register the OAuth client with the data provider (connector) to retrieve the client ID and secret generated for Tableau Server.
- **Redirect URL:** Note the correct redirect URL. You will need this during the registration process in **Step 2** below.

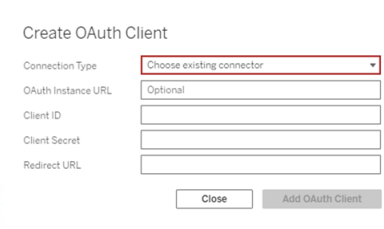
  https://<your_server_name>.com/auth/add_oauth_token

  For example, https://example.com/auth/add_oauth_token

2: Register the OAuth client ID and client secret

Follow the procedure described below to register the custom OAuth client to the site.

1. Sign in to your Tableau Server site using your admin credentials and navigate to the **Settings** page.
2. Under OAuth Clients Registry, click the **Add OAuth Client** button.
3. Enter the required information, including the information from **Step 1** above:
   a. For **Connection Type**, select the connector whose custom OAuth client you want to configure.
   b. **OAuth Instance URL** is required if multiple OAuth clients are being registered. Otherwise, it is optional.
   c. For **Client ID**, **Client Secret**, and **Redirect URL**, enter the information you prepared in **Step 1** above.
   d. Click the **Add OAuth Client** button to complete the registration process.



4. (Optional) Repeat step 3 for all supported connectors.
5. Click the **Save** button at the bottom or top of the Settings page to save changes.

3: Validate and update saved credentials

To help ensure uninterrupted data access, you (and your site users) must delete the previous saved credentials and add it again to use the custom OAuth client for the site.

1. Navigate to your **My Account Settings** page.
2. Under **Saved Credentials for Data Sources**, do the following:
    a. Click **Delete** next to the existing saved credentials for the connector whose custom OAuth client you configured in **Step 2** above.
    b. Next to connector name, click **Add** and follow the prompts to 1) connect to the custom OAuth client configured in **Step 2** above and 2) save the latest credentials.

4: Notify users to update their saved credentials

Make sure you notify your site users to update their saved credentials for the connector whose custom OAuth client you configured in **Step 2** above. Site users can use the procedure described in Update saved credentials to update their saved credentials.

Set Up OAuth for Dropbox

This topic describes how to set up your Dropbox data sources for OAuth authentication. Complete the steps for each Tableau Server instance.

Setting up OAuth for Dropbox consists of the following tasks:

1. Create a new app in your Dropbox developer portal App console.
2. Use the information you get as part of creating the new app to configure your server.
3. (Optional) Configure site-specific OAuth.

Step 1: Create a new app

1. Sign in to your Dropbox developer console, and then choose **App console**.
2. Click the **Create app** button.
3. Configure and name your app and choose the **Create app** button.
4. After the app is created, navigate to its Permissions tab and ensure that the files.-content.read permission is selected.
5. Navigate to the **Settings** tab and add a Redirect URI using the internet address for your Tableau Server.
6. Add the following text to the end of the URI: auth/add_oauth_token. For example:
7. https://your_server_url.com/auth/add_oauth_token
8. Copy the app key, app secret, and redirect URI from the **Settings** tab.

Step 2: Configure Tableau Server for Dropbox

On the Tableau Server computer, open the bash shell, and run the following tsm commands:

```
tsm configuration set -k oauth.dropbox.redirect_uri -v <your_author-
ized_redirect_uri>

tsm configuration set -k oauth.dropbox.client_id -v <your_app_key>

tsm configuration set -k oauth.dropbox.client_secret -v <your_app_
secret>

tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command displays a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this doesn't change the restart behavior. If the changes dont require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Configure custom OAuth for a site

You can configure custom Dropbox OAuth for a site.

Consider configuring a custom OAuth client to 1) override an OAuth client if configured for the server or 2) enable support for securely connecting to data that requires unique OAuth clients.

When a custom OAuth client is configured, the site-level configuration takes precedence over any server-side configuration and all new OAuth credentials created use the site-level OAuth client by default. No Tableau Server restart is required for the configurations to take effect.

Important: Existing OAuth credentials established before the custom OAuth client is configured are temporarily usable but both server administrators and users must update their saved credentials to help ensure uninterrupted data access.

1: Prepare the OAuth client ID, client secret, and redirect URL

Before you can configure the custom OAuth client, you need the information listed below. After you have this information prepared, you can register the custom OAuth client for the site.

- **OAuth client ID and client secret:** First register the OAuth client with the data provider (connector) to retrieve the client ID and secret generated for Tableau Server.
- **Redirect URL:** Note the correct redirect URL. You will need this during the registration process in **Step 2** below.

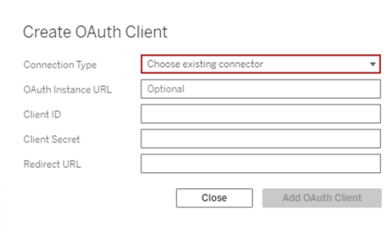
  https://<your_server_name>.com/auth/add_oauth_token

  For example, https://example.com/auth/add_oauth_token

2: Register the OAuth client ID and client secret

Follow the procedure described below to register the custom OAuth client to the site.

1. Sign in to your Tableau Server site using your admin credentials and navigate to the **Settings** page.
2. Under OAuth Clients Registry, click the **Add OAuth Client** button.
3. Enter the required information, including the information from **Step 1** above:
    a. For **Connection Type**, select the connector whose custom OAuth client you want to configure.
    b. **OAuth Instance URL** is required if multiple OAuth clients are being registered. Otherwise, it is optional.
    c. For **Client ID**, **Client Secret**, and **Redirect URL**, enter the information you prepared in **Step 1** above.
    d. Click the **Add OAuth Client** button to complete the registration process.



4. (Optional) Repeat step 3 for all supported connectors.
5. Click the **Save** button at the bottom or top of the Settings page to save changes.

3: Validate and update saved credentials

To help ensure uninterrupted data access, you (and your site users) must delete the previous saved credentials and add it again to use the custom OAuth client for the site.

1. Navigate to your **My Account Settings** page.
2. Under **Saved Credentials for Data Sources**, do the following:

a. Click **Delete** next to the existing saved credentials for the connector whose custom OAuth client you configured in **Step 2** above.

b. Next to connector name, click **Add** and follow the prompts to 1) connect to the custom OAuth client configured in **Step 2** above and 2) save the latest credentials.

4: Notify users to update their saved credentials

Make sure you notify your site users to update their saved credentials for the connector whose custom OAuth client you configured in **Step 2** above. Site users can use the procedure described in Update saved credentials to update their saved credentials.

Set up OAuth for Google

By default, the Google Analytics, Google BigQuery, and Google Sheets (deprecated in Tableau version 2022.1) connectors use a managed keychain for OAuth tokens that are generated for Tableau Server by the provider and shared by all users on the same site.

You can convert the connectors that use managed keychain to use saved credentials by configuring Tableau Server with an OAuth client ID and secret for each connector.

This topic describes how to set up your Google Analytics, Google BigQuery, and Google Sheets connections for OAuth with saved credentials. Complete these steps for each Tableau Server instance.

> **Note:** Google Drive connections use saved credentials by default and, starting in Tableau 2022.3, require Tableau Server to be set up with an OAuth client ID and secret for Google.

For more information about managed keychain and saved credentials, see OAuth Connections

Notes:

- All Google-based connectors require managed keychain (default), server-wide OAuth, or site-specific OAuth.
- To use saved credentials for a site, server-wide OAuth must be configured first.
- Server-wide OAuth can be used whether site-wide OAuth is configured.
- If using site-specific OAuth, each site must be configured individually.
- To support live connection prompts, editing connections, and web authoring, convert managed keychain to saved credentials to avoid errors.

Summary of steps

Set up OAuth by following these general steps:

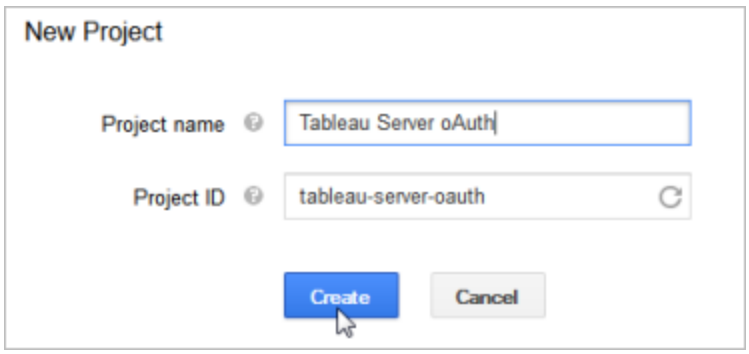
1. Enable API access and create an access token from Google.
2. Use the information you obtained in step 1 to configure Tableau Server.
3. (Optional) Configure site-specific OAuth.
4. Create and edit a Google data source.

Obtain a client ID and enable Google APIs

Note These steps reflect the settings in the Google Cloud Platform console at the time of this writing. For more information, see Using OAuth 2.0 for Web Server Applications in the Google Developers Console Help.

1. Sign in to Google Cloud Platform, and then click **Go to my console**.
2. On the dropdown menu, **Select a Project**, select **Create project**.
3. In the new project form that appears, complete the following:
    - Give the project a meaningful name that reflects the Tableau Server instance for which you'll use this project.
    - Determine whether you want to change the project ID.

> **Note** After you create the project, you won't be able to change the project ID. For more information, click the question mark icons.



4. Open the new project, navigate to **APIs & Services** > **OAuth consent screen**. and select the User Type.
5. Click the **OAuth consent screen** tab and then enter a meaningful name for the **Product name** shown to users.
6. Click **Credentials** and click the **Create Credentials** tab, then click **OAuth client ID**.
7. On the **Create OAuth client ID** screen, fill out the required fields. Follow the steps to authorize your OAuth tokens:
    - Select **Web Application**.
    - Enter a client **Name**.
    - For **Authorized JavaScript Origins**, click **ADD URI** and enter the Tableau Server domain name using HTTP or HTTPs.
    - For **Authorized redirect URIs**, click **ADD URI** and replace the example text with the Internet address for your Tableau Server, and add the following text to

the end of it: **auth/add_oauth_token**. For example:

```
https://your_server_url.com/auth/add_oauth_token
```

8. Copy the Authorized Redirect URI, and paste it in a location that you can access from your Tableau Server computer.
9. Click **Create** .
10. Copy the following values that Google returns, and paste them in a location that you can access from your Tableau Server computer:
    - Client ID
    - Client secret
11. In **APIs & services**, verify that **BigQuery API**, **Google Drive API** (to enable Google Sheets), or **Analytics API** is enabled. To enable APIs, click **ENABLE API** at the top of the page.

    Note: To establish a connection between Tableau Server and Google Analytics 4, you must enable both the Google Analytics Admin API and the Google Analytics Data API in the Google console. By adding these APIs, you can prevent any potential permissions errors that may arise during the process.

Configure Tableau Server for Google OAuth

Using the information you obtained by completing the steps in Obtain a client ID and enable Google APIs, configure your Tableau Server:

- On the Tableau Server computer, open the shell and run the following commands to specify the access token and URI:

  ```
  tsm configuration set -k oauth.google.client_id -v <your_client_ID>
  ```

```
tsm configuration set -k oauth.google.client_secret -v <your_
client_secret>

tsm configuration set -k oauth.google.redirect_uri -v <your_
authorized_redirect_URI>

tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Configure custom OAuth for a site

You can configure a custom Google OAuth client for a site.

Consider configuring a custom OAuth client to 1) override an OAuth client if configured for the server or 2) enable support for securely connecting to data that requires unique OAuth clients.

When a custom OAuth client is configured, the site-level configuration takes precedence over any server-side configuration and all new OAuth credentials created use the site-level OAuth client by default. No Tableau Server restart is required for the configurations to take effect.

Important: Existing OAuth credentials established before the custom OAuth client is configured are temporarily usable but both server administrators and users must update their saved credentials to help ensure uninterrupted data access.

1: Prepare the OAuth client ID, client secret, and redirect URL

Before you can configure the custom OAuth client, you need the information listed below. After you have this information prepared, you can register the custom OAuth client for the site.

- **OAuth client ID and client secret:** First register the OAuth client with the data provider (connector) to retrieve the client ID and secret generated for Tableau Server.
- **Redirect URL:** Note the correct redirect URL. You will need this during the registration process in **Step 2** below.

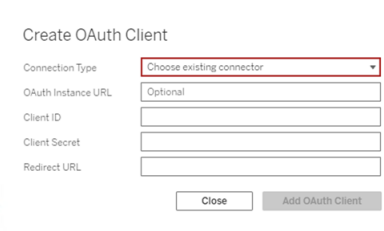
  https://<your_server_name>.com/auth/add_oauth_token

  For example, https://example.com/auth/add_oauth_token

2: Register the OAuth client ID and client secret

Follow the procedure described below to register the custom OAuth client to the site.

1. Sign in to your Tableau Server site using your admin credentials and navigate to the **Settings** page.
2. Under OAuth Clients Registry, click the **Add OAuth Client** button.
3. Enter the required information, including the information from **Step 1** above:
   a. For **Connection Type**, select the connector whose custom OAuth client you want to configure.
   b. **OAuth Instance URL** is required if multiple OAuth clients are being registered. Otherwise, it is optional.
   c. For **Client ID**, **Client Secret**, and **Redirect URL**, enter the information you prepared in **Step 1** above.
   d. Click the **Add OAuth Client** button to complete the registration process.



4. (Optional) Repeat step 3 for all supported connectors.
5. Click the **Save** button at the bottom or top of the Settings page to save changes.

3: Validate and update saved credentials

To help ensure uninterrupted data access, you (and your site users) must delete the previous saved credentials and add it again to use the custom OAuth client for the site.

1. Navigate to your **My Account Settings** page.
2. Under **Saved Credentials for Data Sources**, do the following:
    a. Click **Delete** next to the existing saved credentials for the connector whose custom OAuth client you configured in **Step 2** above.
    b. Next to connector name, click **Add** and follow the prompts to 1) connect to the custom OAuth client configured in **Step 2** above and 2) save the latest credentials.

4: Notify users to update their saved credentials

Make sure you notify your site users to update their saved credentials for the connector whose custom OAuth client you configured in **Step 2** above. Site users can use the procedure described in Update saved credentials to update their saved credentials.

Create and edit Google data source

Next, you must publish the Google data sources to the server. For example, see the Tableau Desktop topic, Google BigQuery.

After you've published the data sources, the final step is to edit the data source connection to use the embedded access token that you configured earlier. See Edit Connections on Tableau Server.

Managing access tokens

After you configure the server for OAuth, you can allow users to manage their own access tokens in their profile settings, or you can manage the tokens centrally. For more information, see Allow Saved Access Tokens.

Set Up OAuth for Intuit QuickBooks Online

This topic describes how to set up your Intuit QuickBooks Online data sources for OAuth authentication. Complete these steps for each Tableau Server instance.

Setting up OAuth for QuickBooks Online consists of the following tasks:

1. Create a Connected App on the Intuit developer platform.
2. Use the information you get as part of the Connected App to configure your server.
3. (Optional) Configure site-specific OAuth.

Step 1: Create an Intuit app

1. Sign in to your Intuit developer account, and then click **My Apps**.
2. In the **Just start coding** section, click **Select APIs**.
3. Select **Accounting** and click **Create App**.
4. In the **Get your app ready for submission** section, click the link to get your production keys.

   Important: You must use production keys rather than development keys.
5. Copy the app token, OAuth consumer key, and OAuth consumer secret.

Step 2: Configure Tableau Server for Intuit QuickBooks Online

- On the Tableau Server computer, open the bash shell and run the following commands:

  ```
  tsm configuration set -k oauth.quickbooks.oauth_callback_uri -v
  http://YOUR-SERVER/auth/add_oauth_token
  ```

  ```
  tsm configuration set -k oauth.quickbooks.consumer_key -v
  <your_consumer_key>
  ```

  ```
  tsm configuration set -k oauth.quickbooks.consumer_secret -v
  <your_consumer_secret>
  ```

  ```
  tsm pending-changes apply
  ```

  If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Managing access tokens

If you run an extract refresh job for your QuickBooks Online data source, Tableau Server attempts to renew access tokens for you. To help ensure that your access tokens do not expire, run your extract refresh jobs more than once a month. Otherwise, the access tokens from QuickBooks Online expire and your extract refresh jobs fail. If your access tokens do expire, you can edit your saved credentials from the **Settings** page.

The saved credentials can be managed centrally or by your users. For more information, see Allow Saved Access Tokens.

Troubleshoot OAuth Connections

This topic provides information about resolving issues that can occur when you configure OAuth data connections.

Conflict error

In some cases, users may receive an error when attempting to connect with OAuth. The first sentence of the error message is:

The server encountered an internal error or misconfiguration and was unable to complete your request.

This error indicates that the fully qualified domain name (FQDN) of the Tableau Server needs to be added to the allowlist redirect key on Tableau Server.

When users are accessing a Tableau Server by the local host name (https://tableau) and the OAuth data provider is responding to the public DNS name (https://data.example.com), Tableau Server must associate the external FQDN with the local server name. The local host name is the server name in the URL that users enter when accessing Tableau Server from the internal network.

To fix this error, run tsm configuration set with the `oauth.whitelisted.redirect_to_origin_host` key option. This key takes a value pair, `"internal_host,FQDN1,FQDN2"`. For example, the following commands set the local host name to *tableau* and the FQDN to *tableau.example.com*:

```
tsm configuration set -k oauth.whitelisted.redirect_to_origin_host -
v "tableau,tableau.example.com"

tsm pending-changes apply
```

In the case where multiple public URLs are used to access the internal Tableau Server, add additional FQDNs to the command, separated by commas, for example:

```
tsm configuration set -k oauth.whitelisted.redirect_to_origin_host
-v "tableau,tableau.example.com,tableau2.example.com"
```

If you need to edit an existing allowlist redirect configuration, you must enter the full mapping set. You cannot truncate or append existing configuration keys.

Configure SAP HANA SSO

You can configure Tableau Server to use SAML delegation to provide a single sign-on (SSO) experience for SAP HANA. This scenario is not dependent on SAML authentication to Tableau Server. You do not need to use SAML sign on with Tableau Server in order to use HANA SAML delegation. You can sign in to Tableau Server using whatever method you choose.

With SAML delegation for SAP HANA, Tableau Server functions as an identity provider (IdP).

Before you begin

Configuring SAML delegation with SAP HANA requires configuration on both Tableau Server and on SAP HANA. This topic provides configuration information about configuring Tableau Server. Before you configure Tableau Server, you must complete the following:

- Acquire a SAML certificate and key file for Tableau Server.
    - The certificate file must be a PEM-encoded x509 certificate with the file extension .crt or .cert. This file is used by Tableau Server and must also be installed on HANA.
    - The private key must be a DER-encoded private key file, in PKCS#8 format, that is not password protected and has the file extension .der. This file is only used by Tableau Server.

- Install the certificate in HANA. To avoid `libxmlsec` errors in HANA, we recommend configuring in-memory certificate store on SAP HANA. For more information, see this SAP support topic.
- Install the latest version of SAP HANA driver (minimum version is 1.00.9) on Tableau Server.
- Configure network encryption from Tableau Server to SAP HANA (recommended).

For more information about generating the certificate/key pair, encrypting the SAML connection, and configuring SAP HANA, see How to Configure SAP HANA for SAML SSO with Tableau Server in the Tableau Community.

Configure Tableau Server SAML for SAP HANA

The following procedure describes how to configure SAML for SAP HANA on Tableau Server using `tsm data-access`. You can also configure SAML for SAP HANA using the sapHanaSettings Entity.

If you are running Tableau Server in a distributed deployment, run the following procedure on the initial node.

1. Place certificate files in a folder named `saml`. For example:

   ```
   /var/opt/saml
   ```

2. Run the following commands to specify the location of the certificate and key files:

   ```
   tsm data-access set-saml-delegation configure --cert-key <cert-key> --cert-file <cert-file>
   ```

   Where `<cert-key>` and `<cert-file>` are file paths to the private key and certificate file, respectively.

   For example,

```
tsm data-access set-saml-delegation configure --cert-key /var/-
opt/saml/hana_pkey_pkcs8.der --cert-file /var/opt/saml/hana_cer-
t.pem
```

You can specify other options. For example, you can specify user name format and how credentials are normalized. See tsm data-access.

3. Run the following commands to enable delegation:

```
tsm data-access set-saml-delegation enable

tsm configuration set -k wgserver.sap_hana_sso.enabled -v true

tsm configuration set -k wgserver.delegation.enabled -v true
```

4. When you have finished, run `tsm pending-changes apply`.

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Enable Kerberos Service Account Access

You can configure Tableau Server to use a Kerberos service account to access a database. In this scenario, Tableau Server connects to databases with a service account, also referred to as a "RunAs account".

To use RunAs authentication on Tableau Server you must first create a workbook or datasource that uses integrated authentication. When users publish to Tableau Server they will get the option for RunAs authentication. If you create a datasource with Tableau Server web authoring that uses integrated authentication, the datasource will use RunAs authentication by default.

> **Note**: *Integrated authentication* is also referred to as *Windows Authentication* on some connectors. In both cases, Tableau Server uses Kerberos authentication.

Data Access with the RunAs Service Account

To use RunAs authentication, the RunAs account requires read and query permissions to external databases. As designed, Tableau Server users with the *Creator* role or the *Explorer (Can Publish)* role have full access to the RunAs account for queries to external databases.

For example, a user with the Creator role can view all databases that have been granted access to the RunAs service account. They can also list tables and run Custom SQL.

If the Creator-user specifies the database host name and selects Integrated Authentication when creating a new data source with web authoring, then databases that have been granted RunAs access will be displayed to the user.

View access to database assets are not restricted to users who connect to Tableau Server with web authoring. Sophisticated users, who have the same roles noted above and who have knowledge of database server names, could create workbooks with Tableau Desktop that display databases that have been granted RunAs access.

Recommendations

Whether user access to databases in these scenarios is acceptable must be assessed by your organization. Generally, reducing the usage and scope of the RunAs service account will reduce the likelihood of inadvertent user access to database content. However, reducing the usage and scope of the RunAs service account may also impose more credential management to you and your users.

Evaluate the following recommendations in context of your business needs and data access policies.

- Firstly, be sure that you trust all users who have Creator roles or Explorer (Can Publish) roles. You will rely on these users to perform actions in Tableau with integrity.

- If you cannot trust all of your users who have publishing rights on data sources that are accessed by the RunAs service account, then you should consider embedding credentials for those data sources.
- If a data source is not set up for automated extract refreshes, that is, the data source is primarily accessed as a live connection, then you may be able to use Kerberos Delegation. For requirements, see Enable Kerberos Delegation.

Requirements

- MIT Kerberos is not supported.
- The RunAs service account must have read access to the target database.

Configuration process

This section provides an example of the process to enable Kerberos service account access.

1. Create a domain user account to act as the RunAs service account. This account must have read access to the target database.

   In the example here, the RunAs service account is User principal named `tabsrv@example.com`.

2. Create a keytab file for the RunAs service account.

   For example, the following commands create a keytab (`tabsrv-runas.keytab`) using the ktutil tool:

   ```
   ktutil
   ```

   ```
   ktutil:  addent -password -p tabsrv@EXAMPLE.COM -k 2 -e <encryption scheme>
   ```

   Encryption schemes for this command include `RC4-HMAC`, `aes128-cts-hmac-sha1-96`, and `aes256-cts-hmac-sha1-96`. Consult your IT team for the correct encryption scheme for your environment and data source.

   ```
   ktutil:  wkt tabsrv-runas.keytab
   ```

Tableau Server will use the RunAs service account and the associated keytab to authenticate and make a direct connection to the database.

3. Copy the keytab into the Tableau Server data directory and set proper ownership and permissions. The keytab should be readable by the unprivileged user. The default unprivileged user created by Tableau Setup is `tableau`.

   If you are running a multi-node deployment, then you must run the following commands on each node in the cluster:

   ```
   mkdir /var/opt/tableau/tableau_server/keytab
   sudo cp -p tabsrv-runas.keytab /var/opt/tableau/tableau_server-
   /keytab
   sudo chown $USER /var/opt/tableau/tableau_server/keytab/tabsrv-
   runas.keytab
   chgrp tableau /var/opt/tableau/tableau_server/keytab/tabsrv-
   runas.keytab
   chmod g+r /var/opt/tableau/tableau_server/keytab/tabsrv-
   runas.keytab
   ```

4. Run the following TSM commands to enable RunAs access, set the RunAs service account, and associate the keytab file with the service account.

   ```
   tsm configuration set -k features.RunAsAuthLinux -v true --
   force-keys
   tsm configuration set -k native_api.datasource_runas_principal
   -v tabsrv@EXAMPLE.COM --force-keys
   tsm configuration set -k native_api.datasource_runas_keytab_
   path -v /var/opt/tableau/tableau_server/keytab/tabsrv-
   runas.keytab --force-keys
   ```

5. Run the following TSM command apply the changes to Tableau Server deployment:

   ```
   tsm pending-changes apply
   ```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Enable Kerberos Run As Authentication for JDBC Connectors

As of version 2020.2, Tableau Server supports Kerberos authentication for JDBC connectors.

You can configure Tableau Server to use a Kerberos service account to access a database. In this scenario, Tableau Server connects to databases with a service account, also referred to as a "Run As service account". This scenario is referred to as "Run As authentication"

To use Run As authentication on Tableau Server you must first create a workbook or datasource in Tableau Desktop that uses integrated authentication. When you publish to Tableau Server you will get the option to use Run As authentication. When creating a datasource with Web Authoring, Run As authentication is the default operation if you select integrated authentication.

Supported data sources

Tableau supports JDBC Kerberos delegation with the following data sources:

- Oracle
- PostgreSQL

Both native and JDBC-based connectors use the same configuration on Tableau Server on Linux. To configure Run As authentication see Enable Kerberos Service Account Access.

SQL Server Impersonation

Impersonation in the context of Tableau Server means allowing one user account to act on behalf of another user account. You can configure Tableau and Microsoft SQL Server to per-

form database user impersonation, so that the SQL Server database account used by Tableau Server queries on behalf of SQL Server database users, who are also Tableau users.

The main benefit of this feature is it allows administrators to implement and control their data security policy in one place: their databases. When Tableau users access a view with a live connection to a SQL Server database, the view only displays what the users' database permissions authorize them to see. An additional benefit is that the users don't have to respond to a database sign-in prompt when they open the view. Also, workbook publishers don't have to rely on user-specific filters to restrict what's seen in views.

Impersonation Requirements

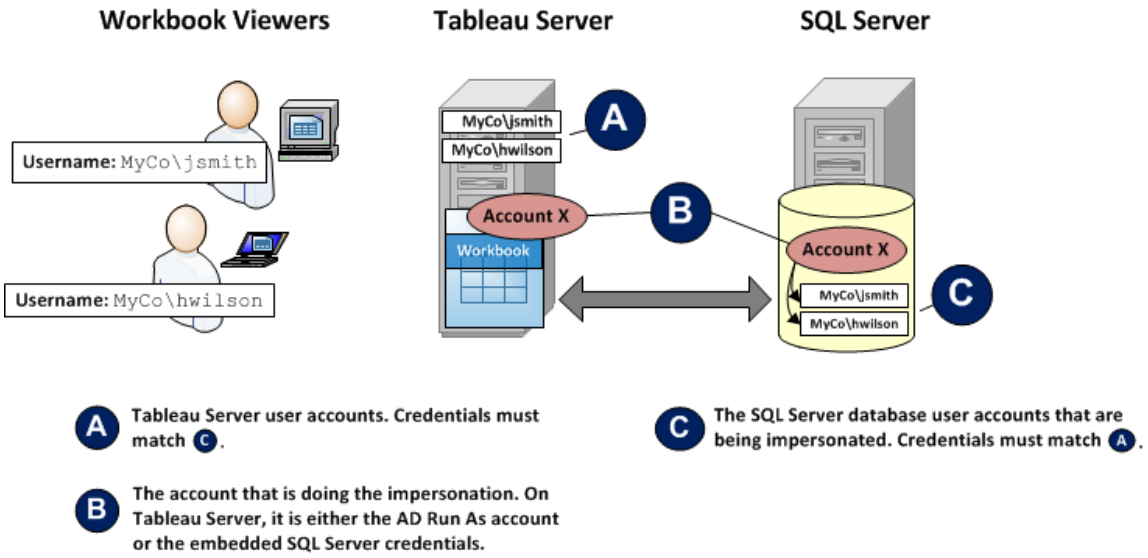
Here’s what you need to use feature:

- **Live connections to SQL Server only:** Impersonation can only be used for views that have a live connection to a SQL Server database, version 2005 or newer.
- **Individual database accounts:** Each person who’ll be accessing the view must have an explicit, individual account in the SQL Server database to which the view connects. Members of an Active Directory (AD) group cannot be impersonated. For example, if Jane Smith is a member of the AD group Sales, and her database administrator adds the Sales AD group to the SQL Server database, Jane cannot be impersonated.
- **Matching credentials and authentication type:** The credentials of each Tableau user's account and their Tableau user authentication type must match their credentials and authentication type in the SQL Server database. For example, if Jane Smith’s Tableau Server user account is `MyCo\jsmith`, the username on the SQL Server database must also be `MyCo\jsmith`. SQL Server must be using Windows Integrated Authentication.
- **SQL Server prerequisites:** In SQL Server you should have a data security table, a view that enforces data security, and you should require that your database users use the view.

- **SQL IMPERSONATE account:** You need a SQL Server database account that has IMPERSONATE permission for the above database users. This is either an account with the sysadmin role or one that has been granted IMPERSONATE permission for each individual user account (see the MSDN article on EXECUTE AS). The SQL Server account must be one of the following:
    - The Tableau Server Run As service account. See Enable Kerberos Service Account Access.
    - The workbook publisher's account. See Impersonate with Embedded SQL Credentials.

How Impersonation Works

Here's an illustration of how database user impersonation works:



In the above illustration, Jane Smith (MyCo\jsmith) is a West Coast sales representative and Henry Wilson (MyCo\hwilson) covers the East. In the SQL Server database, the account permissions for Jane's account, MyCo\jsmith, only give her access to West Coast data. Henry's account, MyCo\hwilson, can only access data for the East Coast.

A view has been created that displays data for the entire country. It has a live connection to a SQL Server database. Both users sign in to Tableau Server and click the view. Tableau Server connects to SQL Server using a database account with IMPERSONATE permission for each user's database account. This account acts on behalf of each user's database account.

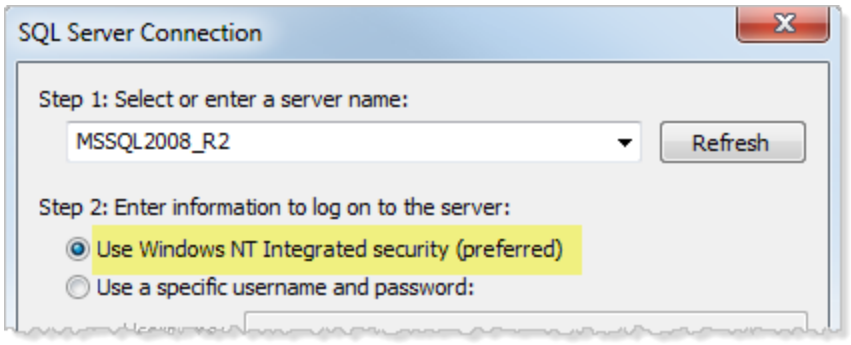
When the view displays, it is restricted by each user's individual database permissions: Jane sees only the West Coast sales data, Henry sees only the East Coast data.

Impersonate with a Run As Service Account

Impersonating via a Run As service account is the recommended way to perform impersonation. The Run As service account is an Active Directory user account the Tableau Server service can run under on the machine hosting Tableau Server. This same account must have IMPERSONATE permission for the database user accounts in SQL Server. From a data security standpoint, using the Tableau Server Run As service account for impersonation gives the administrator the most control.

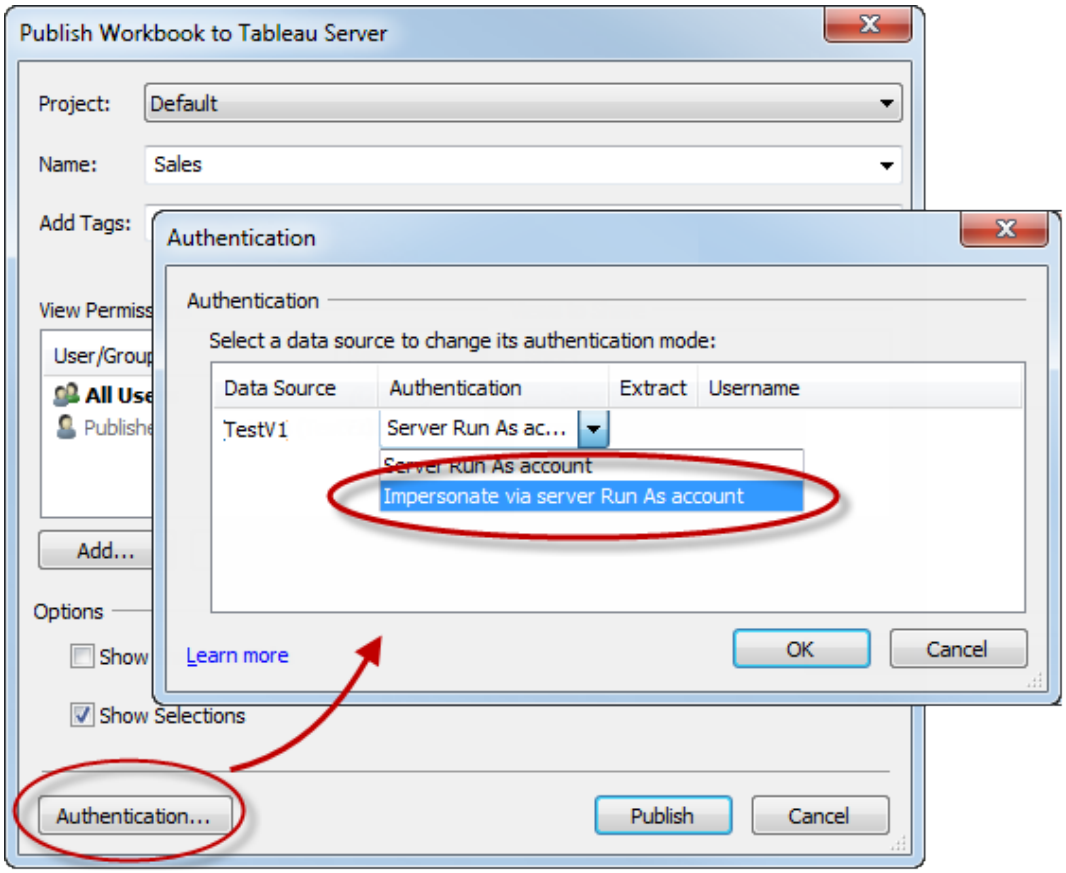
To set up impersonation with a Run As User account:

1. Enable Kerberos Service Account Access.
2. Create a workbook in Tableau Desktop. When you create the data connection, select **Use Windows NT Integrated security** for the workbook's live connection to a SQL Server database:



3. In Tableau Desktop, publish the workbook to Tableau Server (**Server > Publish Workbook**).

4. In the Publish dialog box, click Authentication, then in the Authentication dialog box, select **Impersonate via server Run As account** from the drop-down list:



5. Click **OK**.

6. Test the connection by signing into Tableau Server as a user. When you click a view, you should not be prompted for database credentials and you should only see the data the user is authorized to see.

Impersonate with Embedded SQL Credentials

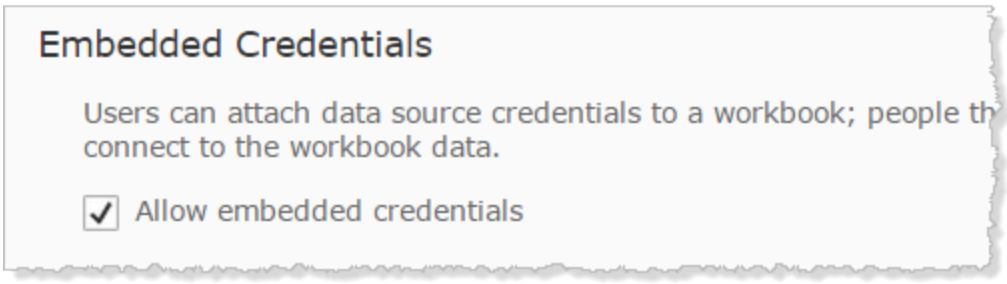
You can also perform impersonation by having the person who publishes a view embed their SQL Server account credentials in the view. Tableau Server can be running under any type of

account, but it will use these credentials, supplied by the publisher, to connect to the database.

This may be the right choice for your site if the account that handles the impersonation cannot be an Active Directory (AD) account and if you're comfortable giving workbook publishers an account with a potentially high permission level on SQL Server.

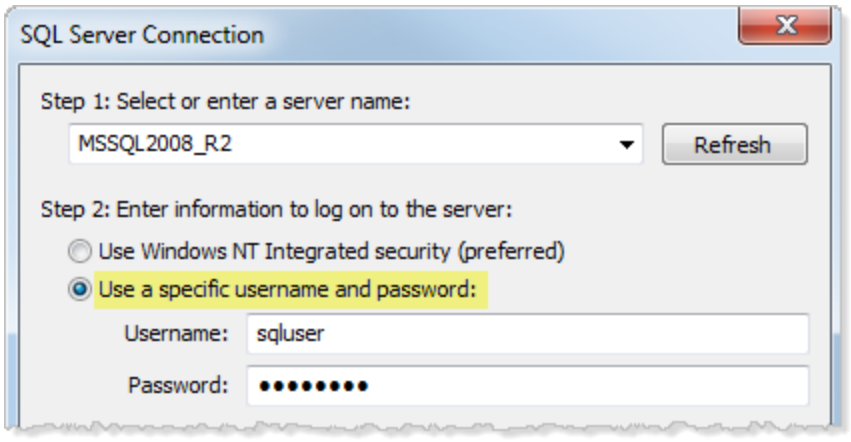
Note:

To use this approach, Embedded Credentials must be enabled on the server Settings page in Tableau Server:



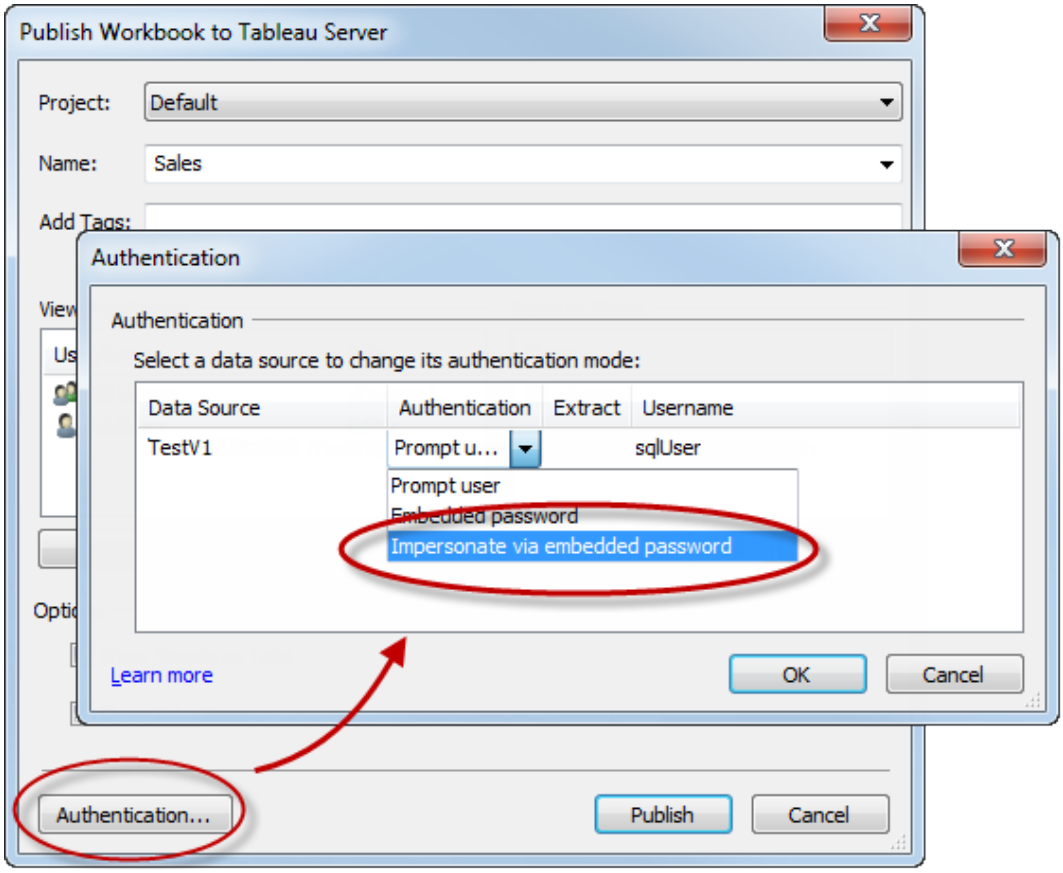
To impersonate with the workbook publisher's SQL account:

1. In Tableau Desktop, create a workbook. When you create the data connection, select Use a specific username and password for the workbook's live connection to a SQL Server database:



2. Publish the workbook to Tableau Server (**Server > Publish Workbook**).

3. In the Publish dialog box, click Authentication, then in the Authentication dialog box, select **Impersonate via embedded password** from the drop-down list:



4. Click **OK**.

5. Test the connection by signing in to Tableau Server as a user. When you click a view, you should not be prompted for database credentials and you should only see the data the user is authorized to see.

Configure a Custom TSM Administration Group

This topic describes how to configure a custom TSM administration group.

By default, the Tableau Server installation process creates a group called `tsmadmin`. Users in this group are authorized as administrators of TSM. You can change this default group during installation. See Help Output for initialize-tsm Script.

If you have already installed Tableau Server and wish to change the group used for TSM administration, follow the procedure in this topic.

You can configure Tableau Server to use an arbitrary group as a custom TSM administration group. Once configured, any user that is a member of the custom group will be able to administer TSM.

Step 1: Create the new group

Create a new group on your computer. Do not change the permissions on the default group as created by Linux.

Step 2: Configure Tableau Server

The custom TSM administration group name is stored in the `tsm.authorized.groups` configuration key. If you want to specify a group name (other than `tsmadmin`) then you will need to update the `tsm.authorized.groups` configuration key and then restart Tableau Server.

Use the tsm configuration set command to set the group name value. For example, to change the TSM administrative group name to `myadmingroup`, run the following commands:

```
tsm configuration set -k tsm.authorized.groups -v myadmingroup

tsm pending-changes apply
```

Notes

- Setting the `tsm.authorized.groups` configuration key overwrites any existing value stored on that key.
- If you have users in the existing `tsmadmin` group, and you overwrite the `tsm.authorized.groups` configuration key with a new value, then the users in the existing `tsmadmin` group will no longer be authorized for TSM.
- You can specify multiple groups by entering a comma-separated list of group names as the value.

Step 3: Add users to the new group

After you finished setting `tsm.authorized.groups`, any user in the new group(s) that you have specified will have full TSM administrative rights on Tableau Server.

Authorization

Authorization refers to how and what users can access on Tableau Server after authentication has been verified. Authorization includes:

- What users are allowed to do with content hosted on Tableau Server, including projects, sites, workbooks, and views.
- What users are allowed to do with the data sources that are managed by Tableau Server.
- What tasks users are allowed to perform to administer Tableau Server, such as configuring server settings, runing command line tools, creating sites, and other tasks.

Authorization for these actions is managed by Tableau Server and determined by a combination of the user's site role and permissions associated with specific entities such as workbooks and data sources.

Site roles

Site roles define who is an administrator. Administrators can be assigned at the site or server level. For non-admins, site roles indicate the maximum level of access a user can have on a given site, subject to permissions set on content assets. For example, if one user is assigned the Viewer site role, and another the CreatorCreator

For more information about site roles, see Set Users' Site Roles.

Permissions

Permissions determine whether a given user is allowed or denied to perform a specific action on a specific content asset.

As an administrator setting up Tableau Server, it's important that you understand how permissions are evaluated. Understanding the Tableau permissions process will enable you to

set up and configure permissions on sites, projects, and other assets so that you can control how content and data is shared, published, viewed, extracted, and imported.

Four important concepts to understand about permissions in Tableau are:

- **Permissions are asset-based**. Permissions are assigned to individual content assets (projects, data sources, workbooks) and are granted to users or groups.
- **Permissions are implicitly denied, and non-admin users must explicitly be allowed to access content**. The process by which Tableau Server determines the "allow" or "deny" permission is explained in detail in Permissions.
- **Permissions inheritance exists only in locked projects and in workbooks with tabbed views**. When content permissions are locked to the top-level project, workbooks, views, and data sources in the entire project hierarchy will use the default permissions set at the top-level project. In workbooks saved with the option **Show sheets as tabs**, views inside those workbooks use the workbook permissions. For more information, see Permissions.
- **In a project that is not locked, initial permissions are a one-time copy of the container item's permissions**. A data source or workbook starts with the default permissions, but authorized users can subsequently edit permissions on those assets. For more information on default permissions and projects, see Permissions.

Tableau Server provides a flexible permissions infrastructure that allows you to manage access to all content for countless scenarios. For more detailed information, see Permissions.

Data access and external authorization

There are scenarios where Tableau Server and Desktop rely on external authorization to enable access to data. For example:

- Users connecting to external data sources might require authorization that is outside the scope of Tableau Server's authority. If users publish an external data source, Tableau Server will manage access and capabilities of that data source. But if users embed an external data source in a workbook, it's up to the user who publishes the workbook to determine how other users who open the workbook will authenticate with the underlying data that the workbook connects to.
- Running Tableau Server in an organization with Active Directory, where Tableau has been configured with a Run As user account, results in a dependency on Active Directory and NTFS for authorization. For example, if you configure Tableau Server to use

the Run As account to impersonate users connecting to SQL, then object-level authorization is reliant on NTFS and Active Directory.
- How users authenticate and are authorized by specific database solutions can differ. As noted, Tableau Server can be configured to provide access authorization when a data source is configured, but some databases will authorize access according to their own authentication scheme.

Data Security

Tableau provides several ways for you to control which users can see which data. For data sources that connect to live databases, you can also control whether users are prompted to provide database credentials when they click a published view. The following three options work together to achieve different results:

- **Database login account:** When you create a data source that connects to a live database, you choose between authenticating to the database through Windows NT or through the database's built-in security mechanism.
- **Authentication mode:** When you publish a data source or a workbook with a live database connection, you can choose an **Authentication mode**. Which modes are available depends on what you choose above.
- **User filters:** You can set filters in a workbook or data source that control which data a person sees in a published view, based on their Tableau Server login account.

The table below outlines some dependencies with the above options:

Database Connection Options		*Data Security Questions*		
Database login account uses...	**Authentication mode**	**Is database security possible per Tableau Server user?**	**Are user filters the only way to restrict which data each user sees?**	**Are web caches shared among users?**
Active Directory credentials (Windows Authentication)	Kerberos service account	No	Yes	Yes
	Impersonate via server Kerberos service account	Yes	No*	No
	Viewer enters their credentials	Yes	No*	No
User name and password	Prompt user: Viewers are prompted for their database credentials when they click a view. Credentials can be saved.	Yes	No	No
	Embedded credentials: The workbook or data source publisher can embed their database credentials.	No	Yes	Yes
	Impersonate via embedded password: Database cre-	Yes	No*	No

Database Connection Options		*Data Security Questions*		
Database login account uses...	**Authentication mode**	**Is database security possible per Tableau Server user?**	**Are user filters the only way to restrict which data each user sees?**	**Are web caches shared among users?**
	dentials with impersonate permission are embedded.			

* Because it can create unexpected results, Tableau recommends that you not use this authentication mode with user filters.

User filters, the embedded credentials option and the impersonation modes have similar effects—when users click a view, they are not prompted for database credentials and they see only the data that pertains to them. However, user filters are applied in the workbook by authors, and the impersonation authentication modes rely on security policies defined by administrators in the database itself.

Overview of Row-Level Security Options in Tableau

Sometimes you want to filter data based on the user that is requesting it. For example:

- You want regional salespeople to see sales figures only for their region.
- You want sales managers to see statistics only for salespeople that report to them.
- You want students to see visualizations based only on their own test scores.

An approach to filtering data this way is called row-level security (RLS). There are multiple methods to accomplish row-level security both inside and outside of Tableau, each with its own pros and cons.

Create a user filter and map users to values manually

The simplest way to achieve row-level security in Tableau is through a user filter where you manually map users to values. For example, you could manually map a user named “Alice” to the value “East” so that she only sees rows in the data source where the “Region” column is “East”.

This method is convenient but high maintenance, and attention must be paid to security. It must be done per-workbook, and you must update the filter and republish the data source as your user base changes. When you publish an asset with this type of user filter, you need to set permissions so that users cannot save or download it and remove the filter, thereby gaining access to all of the data.

For more information, see Create a user filter and map users to values manually in the Tableau Desktop and Web Authoring help.

Create a dynamic user filter using a security field in the data

Using this method, you create a calculated field that automates the process of mapping users to data values. This method requires that the underlying data include the security information you want to use for filtering. For example, using a calculated field, the USERNAME() function, and a “Manager” column in the data source, you could determine if the user requesting the view is a manager and adjust the data in the view accordingly.

Because filtering is defined at the data level and automated by the calculated field, this method is less error prone than mapping users to data values manually. When you publish an asset with this type of user filter, you need to set permissions so that users cannot save or download it and remove the filter, thereby gaining access to all of the data.

For more information, see Create a dynamic filter using a security field in the data in the Tableau Desktop and Web Authoring help.

Use a data policy

Starting in Tableau 2021.4, when Data Management is enabled in Tableau Server or Tableau Cloud, users with a Creator license can implement row-level security through data policies on

virtual connections. Because virtual connections are centralized and reusable, you can manage row-level security for each connection in one place, safely and securely, across all content that uses that connection.

Unlike the above solutions for row-level security in Tableau, this method doesn't carry the same risk of exposing information if an author neglects to properly secure permissions on the workbook or data source, because the policy is enforced on the server for every query.

Row-level security through virtual connection data policies was developed to address shortcomings of other row-level security solutions. We recommend this solution in most situations where it's an option.

For more information on row-level security using data policies on virtual connections, see About Virtual Connections and Data Policies.

Use existing RLS in the database

Many data sources have mechanisms for RLS built in. If your organization has already put effort into building row-level security in a data source, you may be able to take advantage of your existing RLS.

It is not necessarily easier or better to implement a built-in RLS model vs. building it with Tableau in mind; these techniques are generally leveraged when an organization has already invested in these technologies and they want to take advantage of that investment, or when they need to apply the same security policies to other database clients in addition to Tableau.

The main benefit of using built-in RLS is that administrators can implement and control their data security policy in one place: their databases.

For more information, see Row-Level Security in the Database.

Row-level security option comparison

RLS option	Useful when	Pros	Cons
Manual user filter	• You are doing a proof of concept	• Simple at small scales	• High-maintenance

	or testing user filtering functionality • You are creating a static workbook to use with an unchanging group of users • You understand the data security risk of having the permissions set incorrectly	• Easy to understand mapping • Good for testing	• Need to update filter and republish as user base changes • Permissions must be secured to prevent users from seeing unfiltered data • Must be replicated in every workbook
Dynamic user filter	• You don't have a Data Management license • The data contains information you can use to filter it • You understand the data security risk of having the permissions set incorrectly	• Relatively easy to set up	• Permissions must be secured to prevent users from seeing unfiltered data • Must be replicated in every workbook or data source
Data policy	• You have a Data Management license • The data contains information you can use to filter it • Ease of data security is a significant concern	• Centralized • Secure • Low-maintenance • Responsibilities for security and analytics can be separated	• Data Management license required

RLS in the database	• Your database has an existing RLS security built into the database • You aren't using extracts	• Might already be built into your organization's database • Policies can be applied to database clients other than Tableau	• Must use live queries • Might have limitations or requirements. Your IT team can identify them

Which row-level security option should I use?

Does your organization have a preferred RLS solution in the database that works for this project?	→ Yes →	See Row-Level Security in the Database
↓ No ↓		
Do you have a Data Management license?	→ Yes →	See About Virtual Connections and Data Policies
↓ No ↓		
Is this a proof of concept, a basic user filter test, or a static workbook with unchanging users?	→ Yes →	See Use a manual user filter
↓ No ↓		
See Use a dynamic user filter		

RLS Best Practices for Data Sources and Workbooks

Row-level security (RLS) in Tableau restricts the rows of data a certain user can see in a workbook. This differs from Tableau permissions, which control access to content and feature functionality. For example, permissions control whether a user can comment on or edit a

workbook, while row-level security enables two users viewing the same dashboard to see only the data each user is allowed to see.

There are several ways to implement RLS in Tableau. For example, you can set RLS at the data source or workbook level, or you can set RLS at the connection level using a virtual connection with a data policy (requires Data Management). See the Overview of Row-Level Security Options in Tableau for details about alternatives.

> **Note**: This topic focuses on RLS best practices for data sources and workbooks. For more in-depth examples of the concepts outlined in this topic, refer to the whitepaper Best Practices for Row Level Security with Entitlement Tables or How to Set Up Your Database for Row Level Security in Tableau on the blog *Tableau and Behold*.

RLS workflow

For live connections and multi-table extracts, the basic RLS workflow is:

1. The user is identified by logging into Tableau Server or Tableau Cloud
    - This requires a distinct username per user and secure single sign-on (SSO)
    - Active Directory, LDAP, or the Tableau REST API can be used to synchronize user names and establish permissions
2. The set of data entitlements for the user is retrieved from all possible data entitlements
    - This requires a data structure that can link entitlements to the Tableau username
3. The data is filtered by the entitlements for that user
    - This often requires using user functions in a calculated field
4. The published, filtered data is used to build content
    - Using a published (rather than embedded) data source with a data source filter ensures the RLS cannot be modified by downloading or web editing the workbook

How the joins, calculated fields, and filters are set up depends on the structure of the data and how users are managed.

Entitlement tables

Any unique combination of attributes that the data can be filtered on is an entitlement. Most commonly, there are separate tables for specifying the entitlements themselves and mapping those entitlements to users or user roles. Denormalizing is recommended from a performance standpoint because joins are expensive operations.

The entitlements view, consisting of the entitlements mapped to users or roles, is joined with the data. A user-based data source filter is then applied, acting as a WHERE clause that brings in only the entitlements—and therefore the appropriate data rows—for the relevant user. (Query optimization should ensure the filtering occurs before joining when the query is processed to minimize data duplication. For more information, see Performance and processing order of operations.)

Entitlement table models

Generally, there are two models for representing entitlements:

Full mapping to the deepest level of granularity

- Entitlements are defined fully for every column.
- There is one row in the mapping table for every possible entitlement the user has.
- This model requires fewer join clauses.

Sparse entitlements

- Entitlements are defined for every level of hierarchy, with NULL used to represent an “all” state.
- There is a single row in the mapping table for a particular level in the entitlement hierarchy, which vastly reduces the number of entitlement rows for users at high levels in a hierarchy.
- This model requires more complex joins and filters.

Users and roles

Combinations of entitlements are commonly represented as *roles*, which are then linked to users in a many-to-many mapping table. This allows for easily changing or removing a user from the role, while still maintaining a record of the role and its entitlements.

Alternatively, a many-to-many mapping table can be created that instead assigns users directly to entitlements as opposed to going through joining a role table. It will require managing the values more directly in the table but does eliminate a join.

> **Note**: The user values associated with a role or entitlement need to match the username or full name on the Tableau site in order to take advantage of the user functions in Tableau Desktop.

Joins

Regardless of the model used to represent the entitlements, it is advisable to join all entitlements and mapping tables together into a single denormalized entitlements view. While at first this will cause a “blowup” (highly duplicative) version of the entitlements, the data source filter on the user will reduce it back down. You will also want this view if you plan on using an extract.

The deepest granularity method can have a performance benefit when everything is hierarchical—you only need to do a single join on the deepest level of the hierarchy. This only works if all of the attributes at the lowest level are distinct. If there is a chance for duplication (for example, a Central sub-region in more than one region), then you’ll need to join on all the columns to achieve the effect of a distinct key value.

The actual details and their performance characteristics depend on the data system and require testing. For example, using a single key could potentially improve the performance because the join is then only executing on one column, but correctly indexing all of the columns may give equal performance when other factors are taken into consideration.

Implement row-level security

Deepest granularity

After the denormalized view of mapped entitlements is created, an inner join is set up between the view and the data in the Tableau data connection dialog. The data can remain in a traditional star schema. Alternatively, the dimension and fact tables can be materialized together

into two views. Multi-table extracts will build extract tables to match the joins, so creating the two views will simplify the resulting extract. The SQL will follow this basic pattern:

```
SELECT *
FROM data d INNER JOIN entitlements e ON
d.attribute_a = e.attribute_a AND
d.attribute_b = e.attribute_b AND ...
WHERE e.username = USERNAME()
```

Sparse entitlements

If your entitlements more closely resemble the sparse entitlements model, then the custom SQL to join the data to the entitlements would be a little more complex because of the NULL values. Conceptually, it would look like the following:

```
SELECT *
FROM data d
INNER JOIN entitlements e ON
(e.region_id = d.region_id OR ISNULL(e.region_id) AND
(e.sub_region_id = d.sub_region_id OR ISNULL(e.sub_region_id) AND
(e.country_id = d.country_id OR ISNULL(e.country_id)
```

Without using custom SQL, this can be done with a cross join and additional filters in Tableau Desktop. Create a join calculation on both sides of the join dialog that simply consists of the integer 1 and set them equal. This joins every row from the data table with every row in the entitlements table.

Then you need a calculation (or individual calculations) to account for the levels in the hierarchy. For example, you could have several calculations that follow this format: `[region_id] = [region_id (Entitlements View)] OR ISNULL([region_id (Entitlements View)]`

Or you could have a combined calculation for all levels in one:

```
([region_id] = [region_id (Entitlements View)] OR ISNULL([region_id
(Entitlements View)])
AND
```

```
([sub_region_id] = [sub_region_id (Entitlements View)] OR ISNULL
([sub_region_id (Entitlements View)])
AND
([country_id] = [country_id (Entitlements View)] OR ISNULL([country_
id (Entitlements View)])
```

The ISNULL function matches any entitlement column to all items in the other column. As always with RLS, these calculations should be added as data source filters.

Data source filter

For both approaches, once the entitlements are correctly joined with the data, a filter needs to be set up to limit the data for a specific user. A calculated field should be created with a user function. For example, a simple Boolean comparison of whether the user listed in the User-name field is the same as the username of the person logged into the Tableau site: `[User-name] = USERNAME()`

This calculation should be used as a data source filter (with TRUE selected).

If the data source is embedded and a user has permissions to web edit or download the workbook, then the RLS is nonexistent since the filters enforcing it can be easily removed. The Tableau data source should be published separately as opposed to being left embedded in the workbook.

All access with deepest granularity

There is also a common scenario in which there are two access levels within the organization: people who can see everything (“all access”) or people with some reasonably definable subset of entitlements. This is most commonly seen for embedded applications—the organization hosting the data can see everything, but each client can only see their own data. In this case, you need a way to return the full data for the “all access” users, while maintaining the deepest granularity joins for all other users.

For this technique, you will use Tableau groups to create an override using a calculation in the join condition.

1. Create a group for users who should see all the data (here called All Access)
2. From the fact view, create a left join with two join conditions
    - The first join condition should be on the column that represents the deepest level of granularity
    - The second join condition should be two calculations:
        - On the left side (the fact view), for the calculation, enter `True`
        - On the right side (the entitlements view), the calculation should be: `IF ISMEMBEROF('All Access') THEN False ELSE True END`
3. On a sheet, create a calculation structured as: `[Username] = USERNAME() OR ISMEMBEROF(['All Access'] ([Entitlements View)])`
4. Create a data source filter on the username calculation

If a user is a member of the All Access group, then the join becomes a left join on `True = False`. This means there are no matches at all in the entitlements view, so the entire fact view is returned with NULLs for the columns from the entitlements view (zero duplication). In the case where the user is not part of the All Access group, the `True = True` join condition doesn't change anything and the join will function as expected.

The user calculation used as a data source filter is true for all rows when the group override is working, or it will filter down to only the user's deepest granularity in the hierarchy.

Performance and processing order of operations

When a visualization is viewed in Tableau Desktop, Tableau Server, or Tableau Cloud, Tableau sends an optimized query to the RDBMS which then processes the query and sends results back to Tableau to render the visualization with the resulting data. The order of operations for when joins, calculations, and filters are carried out depends on the query optimizer and how the query is executed.

Live connections

When using a live connection to a data source in Tableau, the performance of the query execution is dependent on the query optimizer which translates the incoming SQL into an efficient plan for retrieving the data.

There are two ways the query can be processed:

1. Filter the entitlement rows to the user then join to the fact table
2. Join the entitlements to the fact table then filter to the user's rows

In an ideal situation, the query optimizer will ensure the database processes the query by *filtering then joining*. If a user is entitled to everything, this means the maximum number of rows processed will be the number of rows in the data table.

If the database processes the query by *joining then filtering*, there may be duplication of data. The maximum number of rows processed will be the number of users entitled to see that particular row times each row in the data table.

It will be clear if this second scenario happens: your queries take a long time to finish, you get errors, or there is an indication of performance issues in the database. Your total data volume will expand exponentially, which could cause inordinate system strain on the backend.

Extracts

When the data source in Tableau is a live connection, Tableau sends every query that is necessary to render a particular viz or dashboard to the RDBMS. When the data source is an extract, the process of querying data from the underlying data source only happens at extract creation and refresh. All of the individual queries for visualizations are answered by the extract engine from the extract file.

The same order of operations issue is present when building single table extracts. However, the "blowup" will happen both on the underlying data source and within the resulting extract itself.

Considerations with extracts

Starting in Tableau 2018.3, the data engine can create a multi-table extract and RLS can be implemented as described above. Using multiple table extracts reduces the time it takes to generate an extract with many-to-many relationships by not materializing the join.

The extract should be built with a *data object* and an *entitlements object*. This is the simplest storage in the extract and results in the best performance.

- The *data object* is the table, view or custom SQL query that represents the denormalized combination of the fact and necessary dimension tables
- The *entitlements object* is a denormalized table, view or custom SQL query of whatever entitlements are necessary to filter the data at the most granular level, which requires:
    - A column for username matching the exact usernames in Tableau Server or Tableau Cloud
    - A row for each of the most granular entitlements to the data object

This format is laid out in the deepest granularity method above. Multi-table extracts use the same method, with the caveat that only two data objects are being joined and any field-specific filtering is already applied within the object.

Because multiple table extracts have extract filters disabled, you can filter either in the views or tables you connect to in the data source, or define the filters in custom SQL objects in the Tableau data connection dialog.

> **Note**: As with live connections, if the data source is embedded and a user has permissions to web edit or download the workbook, then the RLS is nonexistent since the filters enforcing it can be easily removed. The extract should be published separately as opposed to being left embedded in the workbook.

Single table extracts

The following method is only recommended when using a version of Tableau prior to 2018.3—multiple table extracts are preferable if available.

Single table extracts materialize any joins you build when constructing the Tableau data source and stores everything as a single table through one query, the results of which are transformed in a single table in the extract file. This denormalization carries the risk of causing massive data duplication, as every row that was allocated to more than one entitlement or user would be duplicated as a result of the many-to-many relationship.

To prevent this duplication:

1. Create a Security Users Field that contains the usernames for that entitlement
    - for example, a value may be “bhowell|mosterheld|rdugger”
2. Use the CONTAINS() function within Tableau to correctly identify individual users
    - For example, `CONTAINS([Security Users Field], USERNAME())`

This method obviously has some caveats. It requires that you go from your entitlements in rows to a single column separated correctly using SQL, and that column can only contain so many characters. Partial matches can be trouble, and you need to use separators that will never be valid in the IDs themselves. Although it is performant within the Tableau Data Engine, as a string calculation it will be very slow for most databases. This limits your ability to switch back to a live connection.

Alternatively, you can take different extracts per “role” or entitlement level, so that only the data appropriate to that person or level is contained within the extract, but this will require processes to appropriately permission and leverage template publication within Tableau Server, generally via the APIs.

Use built-in row-level security in a database

Many databases have mechanisms for RLS built in. If your organization has already put effort into building row-level security in a database, you might be able to take advantage of your existing RLS. It's not necessarily easier or better to implement a built-in RLS model vs. building it with Tableau in mind; these techniques are generally leveraged when an organization has already invested in these technologies and they want to take advantage of the investment. The main benefit of using built-in RLS is that administrators can implement and control their data security policy in one place: their databases. For more information, see Row-Level Security in the Database.

Row-Level Security in the Database

If your organization has already put effort into building out row-level security (RLS) in a database, you might be able to use one of the following techniques to take advantage of your existing RLS. In order to leverage the database’s security models, live connections are required. Additionally, these techniques are likely not available in Tableau Cloud; the Tableau username

for Tableau Cloud is a unique email address that is not typically the user identity on the database side.

It is not necessarily easier or better to implement a built-in RLS model vs. building it with Tableau in mind; these techniques are generally leveraged when an organization has already invested in these technologies and they want to take advantage of the investment.

Note: For information on the alternatives you can use to implement row-level security in Tableau, see an Overview of Row-Level Security Options in Tableau.

Impersonation (Microsoft SQL Server)

Microsoft SQL Server (and a few related systems) can be configured so that users of the database only have access to views with RLS filters built in, either using Security Junction Tables or views built by the DBA. Tableau can take advantage of this using a concept called “impersonation.”

When publishing a Tableau data source containing an MS SQL Server connection to Tableau Server, there are two authentication options available to take advantage of impersonation. The menu you see will depend on whether you logged into the SQL Server using network authentication or by entering username/password credentials.

To enable RLS filtering for any user who can access the published data source in Tableau Server, either the AD Run-As Account or the embedded SQL server credentials must have permission to EXECUTE AS for all of the Tableau users in the database that will be accessing the dashboard or data source. All Tableau users must exist in the database server as users, with SELECT rights for the Views you are trying to connect to (and have RLS applied to). See Impersonation Requirements for the comprehensive list of requirements.

Kerberos and constrained delegation

Constrained delegation within Tableau Server using Kerberos operates similarly to impersonation in that it allows Tableau Server to use the Kerberos credentials of the view of a work-

book or view to execute a query on behalf of the viewer, so if RLS is set up on the database, the viewer of the workbook will see only their data.

To see the comprehensive list of databases where Kerberos delegation is supported, see Enable Kerberos Delegation. Active Directory is required; the computer where Tableau Server is installed must be joined to the Active Directory domain. The authentication method specified when publishing the data source must be **viewer credentials**.

Note that Kerberos can be leveraged for RLS when using Microsoft Analysis Services.

OLAP Cubes

OLAP Cube connections in Tableau do not have the equivalent of a data source filter, which is required for the entitlements table-based RLS method in Tableau, or access to the USERNAME() function. For these reasons, Kerberos and constrained delegation is a recommended approach to RLS with OLAP databases, which allows Tableau to leverage user filtering that has already been implemented on the OLAP Server side.

If the users viewing the dashboard will not be part of the domain, then the manual approach to creating user filters is possible. However, because the User Filter Set generated cannot be added as a data source filter, and will instead exist on the filters shelf, it is important that Web Editing and Download Workbook functionality is not permissible for any published views using this method.

SAML delegation and SAP HANA

If Tableau Server is configured to use Configure SAP HANA SSO to provide a single sign-on experience, the viewer credentials are used to execute the query as that user, which will operate within whatever security is applied on the user level. The authentication method specified when publishing the data source must be **viewer credentials**.

Initial SQL to force a user-specific session (Oracle VPD)

Initial SQL enables you to specify a SQL command that is run when the connection is made to the database for the purpose of setting up temporary tables to use during the session or to set up a custom data environment.

For Oracle VPD, you can set up a session specific to a user by running a particular stored procedure or function to set the context of the database connection to match the Tableau user's username:

```
begin
DBMS_SESSION.SET_IDENTIFIER([TableauServerUser]);
end;
```

The same high-level requirements hold true for using this for RLS as with impersonation; the DBA must set up VPD and all of the associated users to exist on the database.

On MS SQL Server, you could force an EXECUTE as command (however, this is similar to what Tableau does with impersonation already) :

```
EXECUTE AS USER = [TableauServerUser] WITH NO REVERT;
```

Note: If the data source is embedded and a user has permissions to web edit or download the workbook, then the RLS is nonexistent since the initial SQL enforcing it can be easily removed. The data source should be published separately instead of being embedded in the workbook.

Comparison matrix for row-level security methods

Method	Useful when	Pros	Cons
Entitlements table (Recommended)	• There is an existing concept of entitlements in the database • The organization is setting up Row Level Security for the first time	• Easy to test, update, maintain, and scale • Works for both live connections and extracts in version 2018.3+	• Requires creating and maintaining entitlements table • Could require selecting and creating appropriate keys to optimize for performance

CONTAINS() with extracts	• Implementing RLS in extracts prior to version 2018.3	• Allows you to take advantage of extract efficiencies	• Requires mapping all users to a single column • Difficult to switch back to live connections because of string calculation
Impersonation	• Every user accessing the data will exist as a user in your SQL server (Usually, internal deployments)	• Security is handled and maintained in one place—the database	• Requires every person accessing the view to exist as a user within your SQL Server • Only works for Microsoft SQL Server
Kerberos	• All necessary databases are set up for Kerberos delegation and RLS is set up on the database (usually internal deployments)	• The viewer's name appears on the access logs for the database • Security is handled and maintained in the database	• Tableau must be configured to use LDAP-Active Directory • Tableau Server must be joined to the AD domain • Every user must exist within your AD

			domain
Initial SQL	• The database supports initial SQL and RLS is set up on the database side	• Allows the passing of Tableau parameters at load time • Dedicated connection that can't be shared with other users • Users must exist within database to execute query as user	• Not all databases support initial SQL • Potential performance implications because of restricted cache sharing

Manage Server Secrets

Tableau Server needs to store a number of secrets it uses to perform various functions, typically securing internal communication, communicating with other applications or the operating system, or providing secure communication with clients. In this context, the term *secret* may refer to a password, a token, or other string that is used to authenticate one entity to another.

There are two categories of secrets that are required to run Tableau Server. They differ according to how the secrets are generated:

- **Secrets that are generated by administrators**. These include credentials and associated secrets for the Run As User account and the SMTP credentials used by Tableau Server.
- **Secrets that are automatically generated by various processes in the system**. For example, a secret is required to protect communication between the Cluster Con-

troller and ZooKeeper processes. And a number of different passwords are required for each service and programmatic user that communicates with Postgres.

Most secrets are encrypted while at rest. When a secret is needed, it is decrypted at run time.

This topic describes how secrets storage works and what you need to do to properly manage storage of secrets on Tableau Server.

Understanding how secrets storage works

During installation Tableau Server generates and stores a master key in a Java keystore. The master key is used to encrypt a configuration encryption key that is used across the system.

Whenever a new secret is created or updated, the secret is encrypted with the configuration encryption key. The encrypted value is then stored with its corresponding configuration parameter in a YAML file on the server. Parameters that hold an encrypted value use the format, `ENC(<encrypted string>)`, where `<encrypted string>` is a Base64-encoded encrypted string.

At run time, when a given secret needs to be accessed, the encrypted values are read into memory and decrypted with the configuration encryption key.

In the case of pending changes, where secrets are entered during a configuration change, the entire transaction is encrypted. In this scenario, after you enter a secret and then save the pending change, the secret is transmitted to the Coordination Service (over encrypted SSL). The Coordination Service encrypts the secret and stores it until the pending changes are applied. When changes are applied, the secret (still encrypted) is promoted to the current configuration version.

Tableau Server encrypts secrets using 256-bit AES in GCM mode. The keys used for secure storage are different than the asset keys that are used to encrypt embedded database credentials before they are stored in the repository.

Who has access to the master key?

In a default installation, the Java keystore for Tableau Server will be replicated into the `/tabsvc/keystores` folder for each service on that node under `/var/opt/tableau/tableau_`

`server/data/tabsvc/config`.

For example,

`/var/opt/tableau/tableau_server/data/tabsvc/config/tabadminagent_<version_number>/tabsvc/keystores/tableauserver.jks`.

If you use a custom install directory, then the keystore files will be found under

`<install directory>/tableau_server/data/tabsvc/config/<service name_#.version_number>/tabsvc/keystores`

By default, the following users and groups have access to this directory:

- root
- tableau (user)
- members of the 'tableau' group

Import and export configuration information

Tableau Services Manager introduces the capability to import and export configuration information using tsm settings export.

Note: This version of Tableau Server does not support restoring configuration information from a backup. Instead, we recommend using the export and import configuration commands to backup and restore configuration information.

While configuration secrets are encrypted when stored on disk internally, when the configuration is exported to a file, secrets are written into the file in plain text. It is up to the administrator to take measures to protect this file. There are a variety of options available:

- Write the file to an encrypted file system.
- Write the file to a directory that is restricted to specific users or groups by file system permissions.
- Encrypt the output file.

Securing secrets for import and export operations

This section describes how to PGP encrypt the backup output. With this method, you will create a named pipe and provide it as the file argument, then use that as input to gpg encrypt it. The advantage is the secrets are never written to disk in plain text. gpg must be available and a private key available. An example of this method is shown in the following section.

The example in this section describes one way of handing secrets to store them on a separate computer.

See the following external references for more information:

- File encryption on the command line
- mkfifo(1) - Linux man page

Example: encrypt and export

The following is an example of how to secure the file when exporting the configuration.

```
mkfifo -m 600 /tmp/secure1 && (gpg --symmetric --batch --yes --pass-
phrase-file ~/.secrets/pgppassphrase.txt --cipher-algo AES256 --out-
put encrypted.enc < /tmp/secure1 &) && tsm settings export -f
/tmp/secure1 && rm /tmp/secure1
```

The details of this operation are:

- Create a named pipe with access limited by file permissions to rw for current user.

  ```
  mkfifo -m 600 /tmp/secure1
  ```

- Call gpg to encrypt the data sent to the named pipe, backgrounding it to a separate process. It will block waiting for data. The result will be a file containing the encrypted data.

  ```
  gpg --symmetric --batch --yes --passphrase-file ~/.secret-
  s/pgppassphrase.txt --cipher-algo AES256 --output encrypted.enc
  < /tmp/secure1 &
  ```

- Call tsm to export the configuration, providing the named pipe as the file argument.

```
tsm settings export -f /tmp/secure1
```

- Delete the named pipe.

  ```
  rm /tmp/secure1
  ```

The encrypted data is in the file "encrypted.enc."

Example: decrypt and import

The following is an example of how to decrypt and import the configuration.

```
mkfifo -m 600 /tmp/secret2 && (gpg --decrypt --batch --yes --pass-
phrase-file ~/.secrets/pgppassphrase.txt encrypted.enc > /tm-
p/secret2 &) && tsm settings import -f /tmp/secret2 && rm
/tmp/secret2
```

The details of this operation are:

- Create a named pipe with access limited by file permissions to rw for current user.

  ```
  mkfifo -m 600 /tmp/secure2
  ```

- Decrypt the configuration and send it to the named pipe. Background this to a separate process, it will block waiting to be read.

  ```
  gpg --decrypt --batch --yes --passphrase-file ~/.secret-
  s/pgppassphrase.txt encrypted.enc > /tmp/secret2 &
  ```

- Execute the tsm configuration import command, logging in as needed.

  ```
  tsm settings import -f /tmp/secret2
  ```

- Delete the named pipe.

  ```
  rm /tmp/secure1
  ```

The pending configuration contains the imported configuration.

Run tsm pending-changes apply to commit changes. If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Cluster nodes

When adding a new node to your Tableau Server cluster, you will first need to generate the node configuration file (tsm topology). The node configuration file contains a copy of the master keystore file used for encrypting the configuration secrets.

> **Important**: We strongly recommend that you take additional measures to secure the node configuration file when exporting a configuration file with secrets.

When installing and configuring Tableau Server on the new node, you will need to provide the node configuration file to the `initialize-tsm` command. You can use a similar technique as described above to decrypt the contents of the file that was previously encrypted and send it via a named pipe to the `initialize-tsm` command.

Secrets storage event logging

The following events related to secrets storage are logged:

- Generating new encryption keys
- Encryption key is rolled or changed
- Encrypting a new value in the configuration file

For more information about log files and where they are stored, see Work with Log Files.

Managing secrets

As a Tableau Server administrator the most important task related to secrets storage is to periodically update secrets. In some cases (server troubleshooting or auditing), you may need to retrieve a password.

For other operations, such as upgrading versions, backing up and restoring, or adding new nodes to a cluster—as noted above—Tableau Server manages secrets storage and related processes automatically.

Updating secrets

You should update secrets periodically, according to your company's security policy.

To update the master key and automatically generated secrets, run tsm security regenerate-internal-tokens.

Retrieving passwords

In some cases, you may need to retrieve a password for troubleshooting or other operations. For example, you may need the Postgres readonly user credentials that are generated and encrypted by Tableau Server. In these cases, you can run a tsm command that will retrieve and decrypt the password for you.

To retrieve a password, open Command Prompt and issue a `tsm configuration get` command for one of the parameters listed in the table below.

For example, to retrieve a password for the readonly Postgres user, type the following command:

```
tsm configuration get -k pgsql.readonly_password
```

The command will return the password in clear text:

```
$ tsm configuration get -k pgsql.readonly_password

password
```

Configuration Parameter	Description
clustercontroller.zookeeper.password	Password for cluster controller to connect to zookeeper.
indexandsearchserver.client.password	Password for logging into Index and Search Server.

indexandsearchserver.ssl.admin.cert.bytes	Admin certificate that is used for administrative access to the Index and Search Server. The admin certificate is used to generate the node certificate.
indexandsearchserver.ssl.admin.key.file_bytes	Certificate key for administrative access to the Index and Search Server.
indexandsearchserver.ssl.node.cert.bytes	Certificate that is used for Index and Search Server node-to-node communication.
indexandsearchserver.ssl.node.key.file_bytes	Certificate key that is used for Index and Search Server node-to-node communication.
indexandsearchserver.ssl.root.cert.bytes	Certificate that is used to sign the admin and node certificates . This certificate is used by TSM for health check and by NLP to connect to Index and Search Server.
indexandsearchserver.ssl.root.key.file_bytes	Certificate key for root certificate.
filestore.zookeeper.password	Password for filestore to connect to zookeeper.
hyper.connection.init_password	Password used to initialize the Hyper database for user `tableau_internal_user` and is then used for connecting to Hyper.
jdbc.password	Password for the rails Postgres user.
kms.persistent_store	A collection of master encryption keys (MEKs) used by the Key Management System.

maestro.rserve.password	Password for connecting to an external Rserve instance used by Tableau Prep Conductor for running flows that have nodes with R scripts.
maestro.tabpy.password	Password for connecting to an external TabPy (Python server) instance used by Tableau Prep Conductor for running flows that have nodes with Python scripts.
oauth.google.client_secret	Client secret of the Google Cloud Platform account.
oauth.quickbooks.consumer_secret	Consumer secret of the Intuit developer account.
oauth.salesforce.client_secret	Client secret of the Salesforce developer account.
pgsql.adminpassword	Password for the tblwgadmin Postgres user. **Note**: Although the configuration parameter is encrypted in Tableau's configuration files (tabsvc.yml, workgroup.yml), this password is stored in plain text in files used by SAML.
pgsql.readonly_password	Password for the readonly Postgres user.
pgsql.remote_password	Password for the tableau Postgres user.
redis.password	Password for Redis.

	Note: Although the configuration parameter is encrypted in Tableau's configuration files (tabsvc.yml, workgroup.yml), the configuration will still be in plain text in the redis.conf file that is consumed by the Redis application. Redis does not support encrypted/secured passwords.
servercrashupload.proxy_server_password	Password for custom proxy server used to upload crash reports.
service.runas.password	Password of the Run As users. Stored temporarily.
ssl.cert.file_bytes	The content of one of the three SSL certificate files uploaded by the administrator. The certificate files are required to enable secure external connections to Tableau Server.
ssl.chain.file_bytes	The chain file(s) for the certificates uploaded by the administrator for external SSL.
ssl.key.file_bytes	Key file(s) for the certificates uploaded by the administrator for external SSL.
ssl.key.passphrase	Optional passphrase used to protect the external SSL key.
svcmonitor.notification.smtp.password	SMTP Server password supplied by the administrator through TabConfig.exe.
tabadminservice.password	Password for the service that allows

	server admins to download log files through the web interface.
vizportal.openid.client_secret	This is the password ("provider client secret") used for OpenID Connect SSO.
vizqlserver.external_proxy_password	Password used to authenticate to an external proxy.
wgserver.domain.password	Password used to bind to Active Directory.
wgserver.saml.key.passphrase	Passphrase used to access the PKCS#8 SAML key file.
zookeeper.tsm.password	Password that TSM uses to connect to Zookeeper coordination service

Extension Security - Best Practices for Deployment

The following information is for IT officers and administrators, Tableau server and site administrators, and anyone who is interested in managing dashboard and viz extensions and the security of their data and business. The suggestions for deployment are intended for companies that have a mix of users who are on Tableau Desktop and Tableau Server or Tableau Cloud.

Security for extensions in Tableau

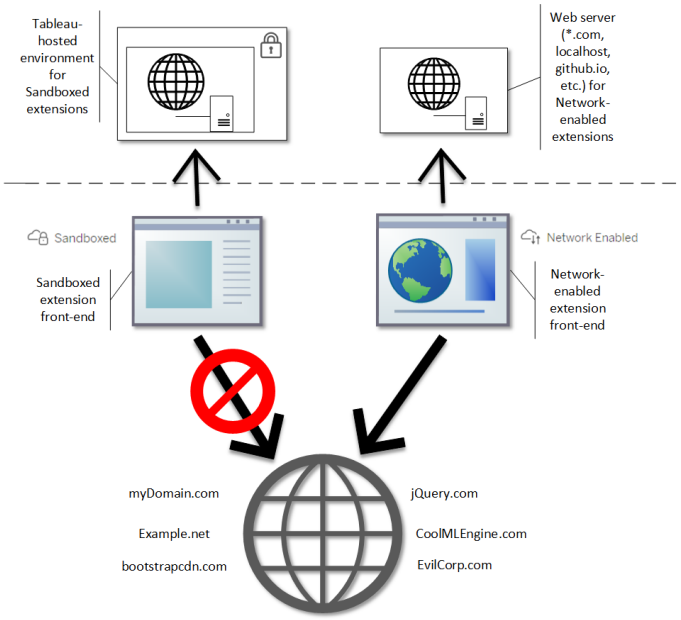
Extensions are web applications that could be hosted inside your network, or outside on a third-party server, or in a secure *sandboxed* environment hosted by Tableau. Extensions can interact with other components in the dashboard and potentially have access to the visible and underlying data in the workbook (through a well-defined API). Tableau supports two types of extensions:

Network-enabled extensions

Network-enabled extensions are hosted on web servers that are located inside or outside of your local network and have full access to the web. Network-enabled extensions can connect with other applications and services. Network-enabled extensions offer new capabilities to Tableau, such as new types of data visualizations, natural language generation, and support for write-back scenarios. Network-enabled extensions have full access to the web, which means that while they can offer rich features and experiences by being able to connect to outside resources, they should be evaluated before deploying or adopting.

Sandboxed extensions

Sandboxed extensions run in a protected environment without access to any other resource or service on the web. Sandboxed extensions are hosted by Tableau and provide the most security and eliminate the risk of data exfiltration. To safeguard against cyber-attacks, the Sandboxed extensions environment and hosting service has undergone extensive penetration testing by a 3rd-party consultant.



You can use Sandboxed and Network-enabled extensions in Tableau Desktop, Tableau Server, and Tableau Cloud. Tableau Server and Tableau Cloud provide the most control over the extensions your users can run.

Potential security risks with Network-enabled extensions

Because extensions are web applications there's the potential that a Network-enabled extension could be vulnerable to certain types of malicious attacks, which in turn could present a risk to your computer or data. The Open Web Application Security Project (OWASP) annually identifies the most critical web application security risks. These risks include the following:

- SQL injection
- Cross-site scripting (XSS)
- Sensitive data exposure

These risks could compromise the extension if the developers of the extension don't properly validate and handle user inputs, or if they generate dynamic queries to access sensitive databases. As you evaluate the extensions that you want to allow in Tableau, be sure to consider how they manage authentication, data access, or user input, and how they mitigate security risks.

Mitigating the security threats with Network-enabled extensions

Understanding what an extension does is a first step to identifying the risks for your enterprise. Often, a dashboard or viz extension doesn’t access underlying data in the workbook and all the JavaScript code runs in the context of the browser running on the user’s computer. In these cases, no data leaves the computer even though the extension might be hosted on a third-party site outside of your domain. Some extensions allow you to connect Tableau with other applications you have already deployed in your domain.

Tableau provides security measures and security requirements for extensions. These requirements are enabled for Tableau Desktop, Tableau Server, and Tableau Cloud.

- All extensions must use the HTTP Secure (HTTPS) protocol.
- By default, anyone using a dashboard with a Network-enabled extension will be prompted and asked to allow the extension permission to run. The extension must request permission if it accesses underlying data.

- To run on Tableau Server or Tableau Cloud, the URL of the Network-enabled extension must be added to a safe list. The server administrator manages this list for Tableau Server; the site administrator manages this list for Tableau Cloud.
- On Tableau Server and Tableau Cloud, the server or site administrator (respectively) can control whether the prompt appears for each Network-enabled extension.

For more information, see Manage Dashboard and Viz Extensions in Tableau Server.

Manage extensions using Tableau

Extensions provide a way to add unique features to dashboards and new visualizations to worksheets. You can use extensions to directly integrate the dashboard with applications outside of Tableau. While extensions open up a world of possibilities, there are instances where you need or want to maintain control of how extensions are deployed in your company or enterprise. In this respect, extensions are no different from any other software that you intend to use. Before you deploy software applications in your company you should thoroughly test and verify that the software works as expected and is secure. The same is true for extensions.

First, determine what level of access your users should have, and identify the extensions you want to use (or conversely, the extensions you don't want to use). Then use the controls and features within Tableau to restrict and curate the dashboard and viz extensions users have access to.

- Do you need to restrict who can add or use extensions in Tableau Desktop? See Recommendations for Tableau Desktop
- Do you need to restrict or control the extensions your users have access to? See Recommendations for Tableau Server and Tableau Cloud.

Recommendations for Tableau Desktop

You have a range of options for deploying Tableau Desktop in your company. You can allow unrestricted access to Sandboxed and Network-enabled extensions, or you can put limits and restrictions on who has access to extensions and under what circumstances.

By default, Tableau Desktop users have unrestricted access to Sandboxed and Network-enabled extensions. You can use two options during installation to change the default settings.

- Turn off all extensions (`DISABLEEXTENSIONS`)
- Turn off Network-enabled extensions (`DISABLENETWORKEXTENSIONS`).

Note: You can change these settings after Tableau Desktop installation by editing the Registry (Windows) or running a script (Mac) on each Desktop. See Turn off dashboard extensions.

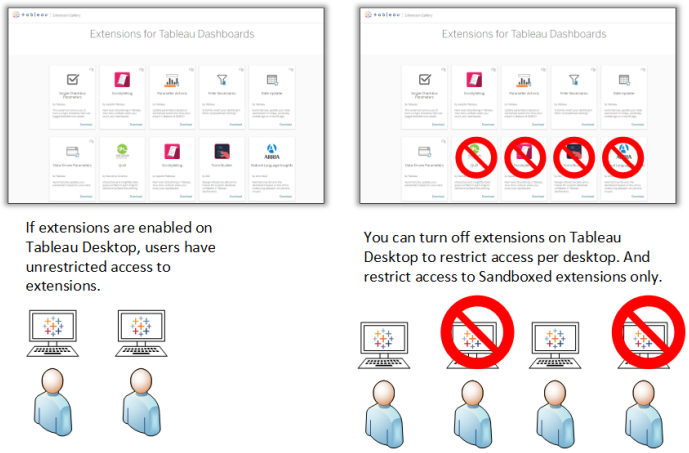
Deployment scenarios

Using the installation settings, you can deploy Tableau Desktop in several ways.

- **Allow all extensions** - In this deployment scenario, you choose to trust Tableau authors to select the Sandboxed and Network-enabled extensions they want to use. If you want to empower your Tableau Desktop users with the greatest flexibility, use the default installation settings. Using the default settings, Tableau Desktop users have unrestricted access to Sandboxed and Network-enabled extensions. The default settings are: `DISABLEEXTENSIONS=0` and `DISABLENETWORKEXTENSIONS=0`. See Install Tableau Desktop from the Command Line.
- **Only allow Sandboxed extensions** - In this scenario, you know that Sandboxed extensions are safe and you want to allow them, but you aren't sure about Network-enabled extensions and want to prevent their use. To turn off support for Network-enabled extensions, set the `DISABLENETWORKEXTENSIONS` property (`DISABLENETWORKEXTENSIONS=1`). Keep the default setting for enabling extensions (`DISABLEEXTENSIONS=0`). See Install Tableau Desktop from the Command Line.
- **No extensions allowed** - In this scenario, you don't want to allow users to use extensions of either type, Network-enabled or Sandboxed. In this case, turn support for all extensions off by using the `DISABLEEXTENSIONS` property (`DISABLEEXTENSIONS=1`). See Install Tableau Desktop from the Command Line .

Use a combination of settings You might have some users who need and should have unrestricted access to all extensions, and others for whom access to Sandboxed extensions is sufficient, and then finally a set of users who need no access to extensions at all. Because the extension options are set per desktop, you can configure your deployment for specific users and their use cases.

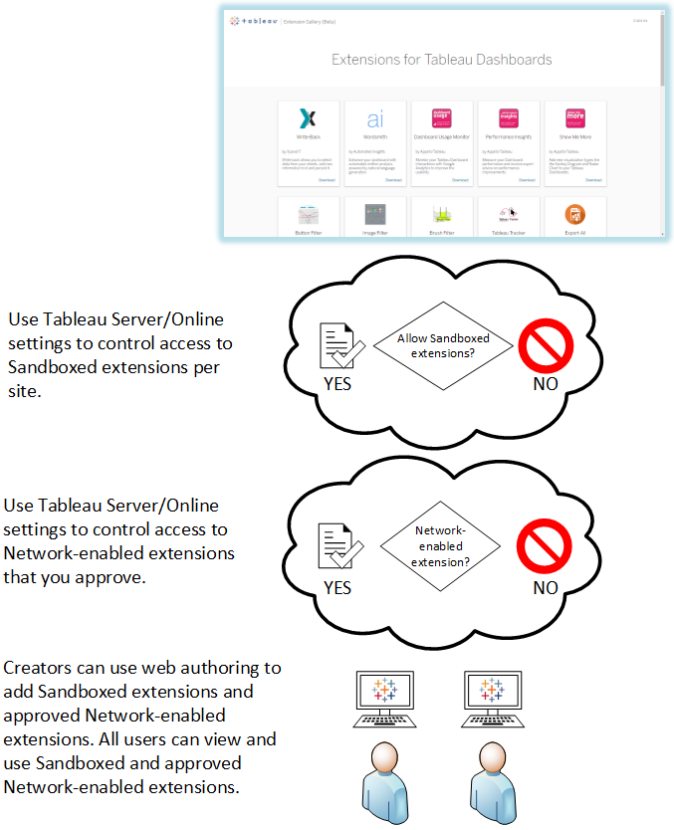
Web authoring - If Tableau Server or Tableau Cloud are available for your users, they can use web authoring to access extensions. In web authoring, the server or site settings for extensions apply. In this scenario, the server and site administrators can determine which extensions to allow users access to. Administrators can use the server and site settings to restrict access to Sandboxed extensions only, or to restrict access to Sandboxed extensions and the Network-enabled extensions that have been added to a safe list.



Recommendations for Tableau Server and Tableau Cloud

If your users have access to Tableau Server or Tableau Cloud, you can use the built-in security controls to put limits and restrictions on the extensions that can be used and under what circumstances. If you've turned off extensions on Tableau Desktop, you can still allow users to

add extensions in web authoring, but you can limit the number of extensions that can be used to just ones you approve of.



Trust Sandboxed extensions and the Network-enabled extensions on the safe list

Starting with Tableau 2019.4, only Sandboxed extensions are allowed to run by default. Network-enabled extensions aren't allowed unless they've been added to the safe list. Administrators can add Network-enabled extensions to the settings page for the site (**Settings** > **Extensions** > **Enable specific Extensions**).

Note To make the safe list the default behavior for extensions in Tableau 2018.2 and Tableau 2018.3, you must change the settings for the site. On the Extensions settings page, under **Default behavior for Extensions**, clear the **Enable unknown**

extensions... option. In Tableau Server 2019.1, Tableau 2019.2, and Tableau 2019.3, by default, no extensions are allowed to run unless they've been added to the safe list.

Checklist for the safe list:

- Does the extension come from a source that you know and trust?
- Check the URL of the extension. Does the URL look suspicious or contain dubious domain names?
- Does the extension require access to full (underlying data) or summary data? See Understand data access.
- Test the extensions before allowing broad use. See Test extensions for security. See Test Network-enabled extensions for security.

Add extensions to the safe list:

- See Add extensions to the safe list and configure user prompts.

Block specific extensions from running on Tableau Server

On Tableau Server, you can block specific extensions by adding their URL to the block list. This is useful if you have multiple sites that are configured differently for extensions. For example, if you have a test site where you want to be able to test internal or third-party extensions, you might have enabled the default behavior for extensions, where unlisted extensions are allowed to run provided they do not access the underlying data in the workbook. Adding an extension to the block list prevents it from inadvertently being used on the test site.

- Add the URL of the extensions that you don't want to allow to the blocked list. This option is only available on Tableau Server. See Block specific extensions.

Turn off extensions for a site

By default, extensions are enabled on Tableau Server and Tableau Cloud. On Tableau Server, the server administrator can turn off extensions for a site. On Tableau Cloud, the site administrator can turn off extensions for the site. On Tableau Server, the server administrator can turn off extensions completely, which overrides the site settings. You should not have to change this setting on the server or for the site, as you can control the Network-enabled

extensions that you want to allow on the safe list. You can also control the settings for Sandboxed extensions, which are allowed by default.

- To disable extensions on a site (Tableau Server, Tableau Cloud), change the site settings that enables users to run extensions on the site. See Control extensions and access to data.

Show or hide user prompts to run Network-enabled extensions

When you add a Network-enabled extension to the safe list, you can configure whether users see prompts by default when they are adding the extension to a dashboard, or when they are interacting with a view that has the extension. The prompt tells users details about the Network-enabled extension and whether the extension has access to full data. The prompt gives users the ability to allow or deny the extension from running. You can hide this prompt from users, allowing the extension to run immediately. When enabled for a site, Sandboxed extensions are allowed by default and do not prompt users.

Turn off Sandboxed extensions

Starting in Tableau 2019.4, Sandboxed extensions are enabled for Tableau Server and Tableau Cloud by default. Sandboxed extensions run in a protected environment and are hosted by Tableau. Administrators can control whether to let users run Sandboxed extensions on a site. Sandboxed extensions don't need to be added to the safe list. When Sandboxed extensions are allowed, users are able to freely add Sandboxed extensions to dashboards and are able to open and use dashboards that contain Sandboxed-extensions. If you need to block a Sandboxed extension, a server administrator can add the Sandboxed extension to a global block list. If you need to turn off Sandboxed extensions completely, you can change the default setting for the site. If you change the default setting for Sandboxed extensions, only the extensions (including Sandboxed extensions) that are on the safe list are allowed to run.

Tableau Server Key Management System

Tableau Server has three Key Management System (KMS) options that allow you to enable encryption at rest. One is a local option that is available with all installations of Tableau

Server. Two additional options require Advanced Management capabilities, but allow you to use a different KMS.

> **Important:** As of September 16, 2024, Advanced Management is no longer available as an independent add-on option. Advanced Management capabilities are only available if you previously purchased Advanced Management, or if you purchase certain license editions - either Tableau Enterprise (for Tableau Server or Tableau Cloud) or Tableau+ (for Tableau Cloud).

Beginning in version 2019.3, Tableau Server added these KMS options:

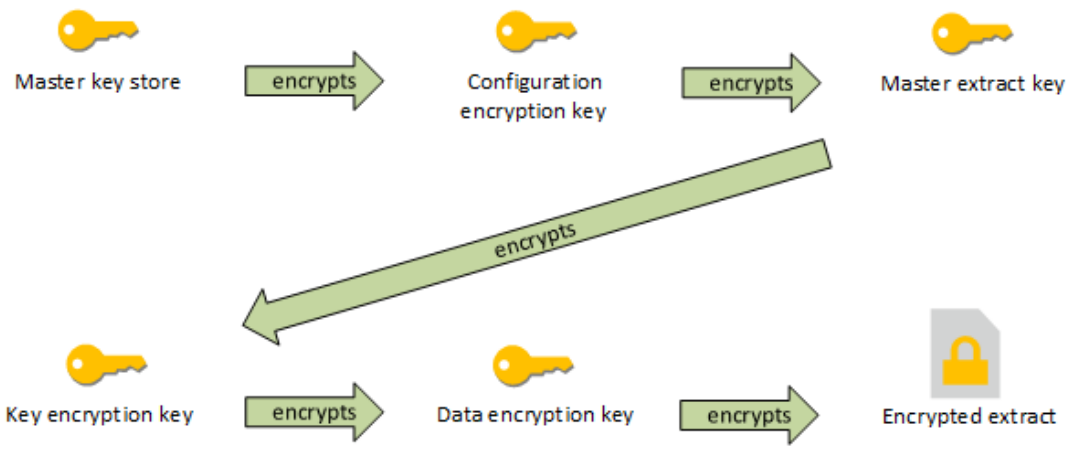
- A local KMS that is available with all installations. This is described below.
- An AWS-based KMS that comes as part of Advanced Management. For details, see AWS Key Management System.

Beginning in version 2021.1, Tableau Server added another KMS option:

- An Azure-based KMS that comes as part of Advanced Management. For details, see Azure Key Vault.

Tableau Server local KMS

The Tableau Server local KMS uses the secret storage capability described in Manage Server Secrets to encrypt and store the master extract key. In this scenario, the Java keystore serves as the root of the key hierarchy. The Java keystore is installed with Tableau Server. Access to the master key is managed by native file system authorization mechanisms by the operating system. In the default configuration, the Tableau Server local KMS is used for encrypted extracts. The key hierarchy for local KMS and encrypted extracts is illustrated here:



Troubleshoot configuration

Multi-node misconfiguration

In a multi-node setup for AWS KMS, the `tsm security kms status` command may report healthy (OK) status, even if another node in the cluster is misconfigured. The KMS status check only reports on the node where the Tableau Server Administration Controller process is running and does not report on the other nodes in the cluster. By default the Tableau Server Administration Controller process runs on the initial node in the cluster.

Therefore, if another node is misconfigured such that Tableau Server is unable to access the AWS CMK, those nodes may report Error states for various services, which will fail to start.

If some services fail to start after you have set KMS to the AWS mode, then run the following command to revert to local mode: `tsm security kms set-mode local`.

Regenerate RMK and MEK on Tableau Server

To regenerate the root master key and the master encryption keys on Tableau Server, run the `tsm security regenerate-internal-tokens` command.

Extract Encryption at Rest

Extract encryption at rest is a data security feature that allows you to encrypt .hyper extracts while they are stored on Tableau Server.

Tableau Server administrators can enforce encryption of all extracts on their site or allow users to specify to encrypt all extracts associated with particular published workbooks or data sources.

Limitations

Before they can be encrypted, older .tde file extracts must be upgraded to .hyper file extracts. This happens automatically as a part of the encryption job. For more information about the impacts of extract upgrade, see Extract Upgrade to .hyper Format.

Temporary files and cache files are not encrypted at rest with this feature.

Workbooks (.twb) and data source files (.tds) are not encrypted with this feature. These files will contain metadata such a database table column names and formatting instructions. In certain cases, they may contain some row-level data if it is included in filters.

Other data files, such as Excel or JSON files, are not encrypted with this feature unless they are converted to extracts before being published.

When extracts are downloaded from the server they are decrypted.

Performance Overview

Increase in Backgrounder Load

You may see a slight to moderate increase in backgrounder load when you turn on encryption at rest. Encryption and decryption are computationally intensive operations. Encryption at rest alters existing backgrounder jobs and introduces new jobs to run on backgrounder. The overall increase in backgrounder load depends on the number and size of affected extracts and how often the scenarios below apply.

- **Initial publishing**: When publishing workbooks or data sources using extracts that should be encrypted, the encryption happens on the server's backgrounders.
- **Extract refreshes from Tableau Server**: Full and incremental refreshes of encrypted extracts on Tableau Server will consume slightly more CPU.
- **Extract refreshes from Tableau Bridge and third-party applications (e.g., Informatica, Alteryx)**: These flows will require new encryption jobs, scheduled on the backgrounders for any refreshed extract, resulting in a slight to moderate increase in backgrounder load.
- **Encrypting and decrypting extracts in already published workbooks and data sources**: If the site setting for encryption at rest is set to **Enable**, users might choose to encrypt or decrypt extracts in already published workbooks and data sources on Tableau server. Depending on the number and size of extracts, this will add slight to moderate load on the backgrounders.
- **Changing a site's encryption mode**: When switching a site's setting for encryption at rest to **Disable** or **Enforce**, the backgrounder will, respectively, decrypt or encrypt all existing extracts on the site. Depending on the number and size of extracts, this may significantly increase the load on backgrounders until all extracts are unencrypted or encrypted.
- **Rotating encryption keys**: Rotating encryption keys results in the backgrounders re-encrypting all existing extracts published on that site, using fresh encryption keys. Depending on the number and size of extracts, this may significantly increase the load on backgrounders until all extracts are re-encrypted.

If running at or over capacity, consider:

- Adding additional backgrounder processes and resources.
- Letting users encrypt individual workbooks and data sources instead of enforcing encryption for the whole site or disable encryption at rest for sites where it isn't necessary. Note that scheduled and ad hoc extract refreshes will take precedence over encryption and decryption jobs.

Increase in Viz Load Time and Worker Load

Query performance, for example, when loading or interacting with a viz or dashboard, will require the data being decrypted once, when loaded from disk to memory. This will result in a slight increase in viz load time and CPU consumption on worker nodes for the first user loading a workbook. This will not affect other users accessing those workbooks at the same time because the data will already be decrypted in memory.

Impact on Backup and Restore

Encrypted extracts in backups remain encrypted. The size of backup files (.tbks) may increase up to 50-100% due to the ineffectiveness of compression on encrypted extracts. The size increase depends, among other factors, on the number of extracts that are encrypted. The time to restore a backup that contains encrypted extracts might increase slightly due to the time to exchange encryption keys.

If your Tableau Server installation has mostly or only encrypted extracts, consider disabling compression during backups to significantly improve the time backups take. To learn more about TSM backup, see tsm maintenance backup.

Enforce Encryption at Rest on a Site

Tableau Server administrators can enforce encryption of all extracts on their site.

1. In a web browser, sign in to Tableau Server as a server administrator.
2. Go to the site you want to configure.
3. Click **Settings**.
4. Scroll down to the Extract Encryption at Rest section.
   Click **Enforce** to encrypt all extracts that are published and stored on the site.
   Encrypting all existing extracts stored on the site may take a while.
5. Click **Save**

Enable Encryption at Rest on a Site

Tableau Server administrators can allow users to specify to encrypt all extracts associated with particular published workbooks or data sources.

1. In a web browser, sign in to Tableau Server as a server administrator.
2. Go to the site you want to configure.
3. Click **Settings**.
4. Scroll down to the Extract Encryption at Rest section.
5. Click **Enable** to allow users to optionally encrypt extracts on the site.
   Changing to Enable will cancel pending decryption jobs and pending encryption jobs. No encryption jobs are created.
6. Click **Save**

Disable Encryption at Rest on a Site

1. In a web browser, sign in to Tableau Server as a server administrator.
2. Go to the site you want to configure.
3. Click **Settings**.
4. Scroll down to the Extract Encryption at Rest section.
5. Click **Disable** to not allow encrypted extracts on the site.
   Changing to Disable will decrypt all existing encrypted extracts. Decrypting all extracts stored on the site may take a while.
6. Click **Save**

View Extract Encryption Mode for All Sites

1. On a multi-site server, click **Manage all sites** on the site menu.

   > **Note:** The **Manage all sites** option only displays when you are signed in as a server administrator.

2. Click **Sites**.
3. The encryption mode of each site is displayed in the **Extract encryption at rest** column.

Encrypt or Decrypt Extracts for a Published Workbook or Data Source

Note: The option to encrypt or decrypt the extracts associated with particular published workbook or data source is only available when the site setting for encryption at rest is set to

Enable. When a site is set to Disable, all content is not encrypted. When a site is set to Enforce, all content is encrypted.

Note: You must be the owner or administrator.

1. Go to the published workbook or published data source page.
2. Click the dropdown menu that says **Encrypted Extract** or **Unencrypted Extract**.
3. Select **Unencrypted**.
   You will see a message that says, “Decrypting extract.”
   -or-
   Select **Encrypted**.
   An encryption job is started.

Alternatively, you can encrypt or decrypt extracts on the card view action menu, list view action menu, and action menu in the header section.

Encrypt or Decrypt Multiple Items

1. Go to the Data Sources page.
2. Select the check box beside one or more data sources.
3. In the upper-left of the Data Sources page, click **Actions**.
4. Click **Encrypt** or **Decrypt**.

View Encryption Status for a Single Item

1. Sign in to the site.
2. Go to a single data source page.
   -or-
   Go to a single workbook page for a workbook containing embedded data sources.
3. The encryption status is displayed on the page.

Filter Data Sources by Encryption Status

1. In the site, click **Explore**.
2. At the top-right, click the Explore: Top-level Projects dropdown menu and select **All Data Sources**.
3. Click the filter icon.
4. Scroll down to the “Live or extract” section and select a filtering option: All, Live, Extracts, Unencrypted Extracts, Encrypted Extracts, Currently Encrypting, or Currently Decrypting.

5. Select the checkbox beside “Include .tde and .hyper files” if you want to include “Live to .tde file” and “Live to .hyper file” connections in your filter results.

Filter Workbooks by Encryption Status

1. In the site, click **Explore**.
2. At the top-right, click the Explore: Top-level Projects dropdown menu and select **All Workbooks**.
3. Click the filter icon.
4. Scroll down to the “Live or extract” section and select a filtering option: All, Live, Extracts, Published, Unencrypted Extracts, Encrypted Extracts, Currently Encrypting, or Currently Decrypting.
5. Select the checkbox beside “Include .tde and .hyper files” if you want to include “Live to .tde file” and “Live to .hyper file” connections in your filter results.
   Any workbooks that have at least one connection that matches the filter selection will be displayed.

View Status of Encrypt or Decrypt Extracts Background Tasks

1. In the site, click **Site Status**.
2. Click **Background Tasks for Non Extracts** to see completed and pending background task details.
   Note: **Background Tasks for Non Extracts** includes all tasks not related to extract refreshes, so it includes encryption jobs.
3. In the Task menu, select **Encrypt Extracts** or **Decrypt Extracts** and click **Apply**.
4. In the Time Range menu, select a range.
   You see "Encrypt Extracts" or "Decrypt Extracts" background tasks for all of your extract-based published data sources and workbooks.

The tabcmd Utility

The tabcmd command-line utility has commands and options to control extract encryption. For more information, see the tabcmd documentation.

Specify the extract encryption mode when you create a site

```
tabcmd createsite <site-name> --extract-encryption-mode [enforced |
enabled | disabled]
```

Specify the extract encryption mode when you edit a site

```
tabcmd editsite <site-name> --extract-encryption-mode [enforced |
enabled | disabled]
```

Get the extract encryption mode when you list sites

```
tabcmd listsites --get-extract-encryption-mode
```

Encrypt extracts when you publish a workbook, data source, or extract to the server

```
tabcmd publish "filename.hyper" --encrypt-extracts
```

Decrypt all extracts on a site

> **Note:** Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

```
tabcmd decryptextracts <site-name>
```

Encrypt all extracts on a site

> **Note:** Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

```
tabcmd encryptextracts <site-name>
```

Reencrypt all extracts on a site with new encryption keys

You must specify a site.

Note: Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

```
tabcmd reencryptextracts <site-name>
```

For more information, see reencryptextracts.

Tableau Server Rest API

With the Tableau Server REST API you can manage Tableau Server resources programmatically. You can use this access to create your own custom applications or to script interactions with Tableau Server resources.

To learn more, see Extract Encryption Methods.

Network Security

There are three main network interfaces in Tableau Server:

- **Client to Tableau Server**: The client can be a web browser, Tableau Mobile, Tableau Desktop, or the tabcmd utility.
- **Tableau Server to your database(s)**: To refresh data extracts or handle live database connections, Tableau Server needs to communicate with your database(s).
- **Server component communication**: This applies to distributed deployments only.

In most organization, Tableau Server is also configured to communicate with the internet and with a SMTP server.

Client to Tableau Server

A Tableau Server client can be a web browser, a device running Tableau Mobile, Tableau Desktop, or tabcmd commands. Communications between Tableau Server and its clients use

standard HTTP requests and responses. We recommend configuring Tableau Server for HTTPS for all communications. When Tableau Server is configured for SSL, all content and communications between clients are encrypted using SSL, and the HTTPS protocol is used for requests and responses.

By default, passwords are communicated from browsers and tabcmd to Tableau Server using 1024-bit public/private key encryption. This level of encryption is not considered robust enough for secure communications. Additionally, this method, where a public key is sent to the recipient in the clear and without network layer authentication is susceptible to man-in-the-middle attacks.

To adequately secure network traffic from clients to Tableau Server, you must configure SSL with a certificate from a trusted certificate authority.

See Configure SSL for External HTTP Traffic to and from Tableau Server.

Client access from the Internet

We recommend a gateway proxy server to enable secure client access from the internet to your Tableau Server. We do not recommend running Tableau Server in a DMZ or otherwise outside your protected, internal network.

Configure a reverse proxy server, with SSL enabled, to handle all inbound traffic from the internet. In this scenario, the reverse proxy is the only external IP address (or range of addresses if multiple reverse proxies are load-balancing inbound requests) that Tableau Server will communicate with. Reverse proxies are transparent to requesting clients, thereby obfuscating Tableau Server network information and simplifying client configuration.

For configuration information, see Configuring Proxies and Load Balancers for Tableau Server.

Clickjack Protection

By default, Tableau Server has *clickjack protection* enabled. This helps prevent certain types of attacks in which the attacker overlays a transparent version of a page on top of an innocuous-looking page in order to lure a user into clicking links or entering information. With

clickjack protection enabled, Tableau Server imposes certain restrictions on embedding views. For more information, see Clickjack Protection.

Tableau Server to your database

Tableau Server makes dynamic connections to databases to process result sets and refresh extracts. It uses native drivers to connect to databases whenever possible and relies on a generic ODBC adapter when native drivers are unavailable. All communications to the database are routed through these drivers. As such, configuring the driver to communicate on non-standard ports or provide transport encryption is part of the native driver installation. This type of configuration is transparent to Tableau.

When a user stores credentials for external data sources on Tableau Server, they are stored encrypted in Tableau Server's internal database. When a process uses those credentials to query the external data source, the process retrieves the encrypted credentials from the internal database and decrypts them in process.

Tableau Server to the internet

In some cases, where users connect to external data sources, such as the Tableau map servers, then Tableau Server will need to connect to the internet. We recommend that you run all components of Tableau inside your protected network. Therefore, connections to the internet may require that you configure Tableau Server to use a forward proxy.

Tableau Server to a SMTP server

You can configure Tableau Server to send event notifications to administrators and users. As of version 2019.4, Tableau Server supports TLS for the SMTP connection. See Configure SMTP Setup.

Communication with the repository

You can configure Tableau Server to use Secure Sockets Layer (SSL) for encrypted communications on all traffic that is exchange with the Postgres repository to and from other

server components. By default, SSL is disabled for communications between server components and the repository.

For more information, see tsm security repository-ssl enable

Server component communication in a cluster

There are two aspects to communication between Tableau Server components in a distributed server installation: trust and transmission. Each server in a Tableau cluster uses a stringent trust model to ensure that it is receiving valid requests from other servers in the cluster. Computers in the cluster running a gateway process accept requests from third parties (clients), unless they are fronted by a load balancer, in which case the load balancer receives the requests. Servers not running a gateway process only accept requests from other trusted members of the cluster. Trust is established by an allowlist of IP address, port, and protocol. If any of these are invalid, the request is ignored. All members of the cluster can communicate with each other.

When a user stores credentials for external data sources on Tableau Server, they are stored encrypted in Tableau Server's internal database. When a process uses those credentials to query the external data source, the process retrieves the encrypted credentials from the internal database and decrypts them in process.

Clickjack Protection

Tableau Server includes protection against clickjack attacks. *Clickjacking* is a type of attack against web pages in which the attacker tries to lure users into clicking or entering content by displaying the page to attack in a transparent layer over an unrelated page. In the context of Tableau Server, an attacker might try to use a clickjack attack to capture user credentials or to get an authenticated user to change settings on your server. For more information about clickjack attacks, see Clickjacking on the Open Web Application Security Project website.

Note: Clickjack protection was available in previous versions of Tableau Server, but was disabled by default. New installations of Tableau Server 9.1 and later will always have clickjack protection on unless you explicitly disable it.

Effects of clickjack protection

When clickjack protection is enabled on Tableau Server, the behavior of pages loaded from Tableau Server changes in the following ways:

- Tableau Server adds the `X-Frame-Options: SAMEORIGIN` header to certain responses from the server. In the current versions of most browsers, this header prevents the content from being loaded into an **<iframe>** element, which helps prevent clickjacking attacks.
- The top-level page from Tableau Server cannot be loaded in **<iframe>** elements. This includes the sign-in page. One consequence is that you cannot host Tableau Server pages in an application that you create.
- Only views can be embedded.
- If an embedded view requires data source credentials, a message is displayed in the **<iframe>** element with a link to open the view in a secure window where the user can safely enter credentials. Users should always verify the address of the opened window before entering credentials.
- Views can be loaded only if they include the `:embed=y` parameter in the query string, as in this example:

  ```
  http://<server>/views/Sales/CommissionModel?:embed=y
  ```

  Note: When clickjack protection is enabled, embedded views that use the URL copied from the browser address bar might not load. These view URLs usually

> contain the hash symbol (`#`) after the server name (for example, `http://myserver/#/views/Sales/CommissionModel?:embed=y`) are blocked when clickjack protection is enabled on Tableau Server.

Disabling clickjack protection

You should leave clickjack protection enabled unless it is affecting how your users work with Tableau Server. If you want to disable clickjack protection, use the following `tsm` commands:

1. `tsm configuration set -k wgserver.clickjack_defense.enabled -v false`
2. `tsm pending-changes apply`

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

HTTP Response Headers

Tableau Server supports some of the response headers specified in the OWASP Secure Headers Project.

This topic describes how to configure the following response headers for Tableau Server:

- HTTP Strict Transport Security (HSTS)
- Referrer-Policy
- X-Content-Type-Options
- X-XSS-Protection

Tableau Server also supports the Content Security Policy (CSP) standard. CSP configuration is not covered in this topic. See Content Security Policy.

Configuring response headers

All response headers are configured with the tsm configuration set command.

When you are finished configuring response headers, run tsm pending-changes apply.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

HTTP Strict Transport Security (HSTS)

HSTS forces clients connecting to Tableau Server to connect with HTTPS. For more information see the OWASP entry, HTTP Strict Transport Security (HSTS).

Options

gateway.http.hsts

Default value: `false`

The HTTP Strict Transport Security (HSTS) header forces browsers to use HTTPS on the domain where it is enabled.

gateway.http.hsts_options

Default value: `"max-age=31536000"`

By default, HSTS policy is set for one year (31536000 seconds). This time period specifies the amount of time in which the browser will access the server over HTTPS.

Referrer-Policy

Beginning in 2019.2, Tableau Server includes the ability to configure Referrer-Policy HTTP header behavior. This policy is enabled with a default behavior that will include the origin

URL for all "secure as" connections (policy `no-referrer-when-downgrade`). In previous versions, the Referrer-Policy header was not included in responses sent by Tableau Server. For more information about the various policy options that Referrer-Policy supports, see the OWASP entry, Referrer-Policy.

Options

gateway.http.referrer_policy_enabled

Default value: `true`

To exclude the Referrer-Policy header from responses sent by Tableau Server, set this value to `false`.

gateway.http.referrer_policy

Default value: `no-referrer-when-downgrade`

This option defines the referrer policy for Tableau Server. You may specify any of the policy value strings listed in the Referrer-Policy table on the OWASP page.

X-Content-Type-Options

The X-Content-Type-Options response HTTP header specifies that the MIME type in the Content-Type header should not be changed by the browser. In some cases, where MIME type is not specified, a browser may attempt to determine the MIME type by evaluating the characteristics of the payload. The browser will then display the content accordingly. This process is referred to as "sniffing." Misinterpreting the MIME type can lead to security vulnerabilities.

For more information see the OWASP entry, X-Content-Type-Options.

Option

gateway.http.x_content_type_nosniff

Default value: `true`

The X-Content-Type-Options HTTP header is set to 'nosniff' by default with this option.

X-XSS-Protection

The HTTP X-XSS-Protection response header is sent to the browser to enable cross-site scripting (XSS) protection. The X-XSS-Protection response header overrides configurations in cases where users have disabled XXS protection in the browser.

For more information see the OWASP entry, X-XSS-Protection.

Option

gateway.http.x_xss_protection

Default value: `true`

The X-XSS-Protection response header is enabled by default with this option.

Content Security Policy

Tableau Server supports the Content Security Policy (CSP) standard. CSP is intended to be an additional layer of security against cross-site scripting and other malicious web-based attacks. CSP is implemented as a HTTP response header that allows you to specify where external resources, such as scripts and images, can be safely loaded from.

See the Mozilla website for more information about CSP.

Configure and enable CSP

CSP is configured and enabled using the tsm configuration set Options command. If you are running Tableau Server in a distributed deployment, run these commands on the initial node in the cluster. The configuration will be applied across the cluster after you run tsm pending-changes apply.

Step 1: Set default directives

Tableau Server includes the set of default directives in the table below.

To set a directive, use the following tsm syntax:

```
tsm configuration set -k content_security_policy.-
directive.<directive_name> -v "<value>"
```

For example, to set the `connect_src` directive, run the following command:

```
tsm configuration set -k content_security_policy.directive.connect_
src -v "* unsafe-inline"
```

Option	Default value	Description
content_security_policy.-directive.default_src	'none'	Serves as a fallback for the other fetch directives. Valid values for `default_src`.
content_security_policy.-directive.connect_src	*	Restricts the URLs which can be loaded using script interfaces. Valid values for `connect_src`.
content_security_policy.directive.script_src	*	Specifies valid sources for JavaScript. Valid values for `script_src`.
content_security_policy.directive.style_src	* 'unsafe-inline'	Specifies valid sources for stylesheets. Valid values for `style_src`.
content_security_policy.directive.img_src	* data:	Specifies valid sources of images and favicons.

		Valid values for `img_src`.
content_security_policy.directive.font_src	* data:	Specifies valid sources for fonts loaded using @font-face. Valid values for `font_src`.
content_security_policy.directive.frame_src	* data:	Specifies valid sources for nested browsing contexts loading using elements such as <frame> and <iframe>. Valid values for `frame_src`.
content_security_policy.directive.object_src	data:	Specifies valid sources for the <object>, <embed>, and <applet> elements. Valid values for `object_src`.
content_security_policy.directive.report_uri	/vizql/csp-report	Instructs the user agent to report attempts to violate the CSP. These violation reports consist of JSON documents sent via an HTTP POST request to the specified URI. Valid values for `report_uri`.

Step 2: Add additional directives (optional)

The default directives included with Tableau Server are a subset of directives that are supported by CSP.

For a full list of supported CSP directives, go to https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy.

You can add directives to the existing default set, by using adding the new directive in the content_security_policy.directive namespace. You must include the `--force-keys` parameter when adding new directives. The syntax is as follows:

```
tsm configuration set -k content_security_policy.directive.<new_directive_name> -v "<value>" --force-keys
```

For example, to add the `worker-src` directive, run the following command:

```
tsm configuration set -k content_security_policy.directive.worker-src -v "none" --force-keys
```

Step 3: Specify report-only directives (optional)

You can configure CPS to report some directives and to enforce others. When you set content_security_policy.enforce_enabled to true, then all directives are enforced (even if content_security_policy.report_only_enable is also set to true).

To specify directives as "report-only" and not enforced, add the directives to the report_only_ directive namespace. You must include the `--force-keys` parameter when adding new directives. The syntax is as follows:

```
tsm configuration set -k content_security_policy.report_only_directive.<directive_name> -v "<value>" --force-keys
```

For example, to report only on the `script_src` directive, run the following command:

```
tsm configuration set -k content_security_policy.report_only_dir-
ective.script_src -v " http://*.example.com" --force-keys
```

Step 4: Enable CSP on Tableau Server

After you have configured directives, enable CSP on Tableau Server.

The following options are used to enable enforcement or report only mode for the directives you have set.

Option	Default value	Description
content_security_policy.enforce_ enabled	false	Adds a CSP header to all requests so that any violation will be enforced by the browser.
content_security_policy.report_only_ enabled	true	Adds a CSP header to all requests so that any violation will be recorded in our vizql-client logs, but will not be enforced by the browser.

To enable enforcement of the CSP directives that you've specified, run the following command

```
tsm configuration set -k content_security_policy.enforce_enabled -v
true
```

Step 5: Run tsm pending-changes apply

When you are finished configuring CSP, run tsm pending-changes apply.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the

server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

View CSP report

To view CSP violations for a given viz, load the viz in a browser that includes developer tools. This example uses the Chrome browser.

1. Load a test viz with violations that is hosted on the Tableau Server deployment where you configured CSP.
2. Enter `CTRL+Shift+I` to open the developer tools in Chrome.
3. Click the **Network** tab.
4. In the **Filter** field, enter `csp-report`, and then click **Find All**.
    - If there are no violations then the search will not return any CSP reports.
    - If there are violations, click the Headers tab in the results pane and scroll to the bottom to view **Request Payload**.

SSL

SSL (Secure Sockets Layer) is a standard security technology that establishes an encrypted link between a web server and clients. To use SSL, you need to install an SSL certificate on Tableau Server.

You can configure Tableau Server to use SSL in the following ways:

- Use SSL for external HTTP traffic.
- Use mutual (two-way) SSL between clients (Tableau Desktop, web browsers, and tabcmd.exe) and Tableau Server.
- Use SSL for all HTTP traffic between internal server components and the repository.

If you are using mutual SSL, each client also needs a certificate.

Note: Tableau Server uses SSL for user authentication. SSL is not used to handle permissions and authorization to content (data sources and workbooks) hosted on Tableau Server.

Configure SSL for External HTTP Traffic to and from Tableau Server

You can configure Tableau Server to use Secure Sockets Layer (SSL) encrypted communications on all external HTTP traffic. Setting up SSL ensures that access to Tableau Server is secure and that sensitive information passed between the server and Tableau clients—such as Tableau Desktop, the REST API, analytics extensions, and so on—is protected. Steps on how to configure the server for SSL are described this topic; however, you must first acquire a certificate from a trusted authority, and then import the certificate files into Tableau Server.

Mutual SSL authentication is not supported on Tableau Mobile.

SSL certificate requirements

Acquire an Apache SSL certificate from a trusted authority (for example, Verisign, Thawte, Comodo, GoDaddy). You can also use an internal certificate issued by your company. Wildcard certificates, which allow you to use SSL with many host names within the same domain, are also supported.

When you acquire an SSL certificate for external communication to and from Tableau Server, follow these guidelines and requirements:

- All certificate files must be valid PEM-encoded X509 certificates with the extension `.crt`.
- Use a SHA-2 (256 or 512 bit) SSL certificate. Most browsers no longer connect to a server that presents an SHA-1 certificate.

- In addition to the certificate file, you must also acquire a corresponding SSL certificate key file. The key file must be a valid RSA or DSA private key file (with the extension `.key` by convention).

  You can choose to passphrase-protect the key file. The passphrase you enter during configuration will be encrypted while at rest. However, if you want to use the same certificate for SSL and SAML, you must use a key file that is *not* passphrase protected.

  Important: If your key file is passphrase protected, you must verify that the related cryptographic algorithm is supported by the version of Tableau Server you are running. Tableau Server uses OpenSSL to open password protected key files. As of August 2023, the latest releases of Tableau Server (2021.3.26, 2021.4.21, 2022.1.17, 2022.3.9, 2023.1.5, and newer) run OpenSSL 3.1. Previous versions of Tableau Server ran OpenSSL 1.1. A number of cryptographic algorithms have been retired and are no longer supported in OpenSSL 3.1. If you are using a passphrase protected key file on an older version of Tableau Server that is still running OpenSSL 1.1, review the following Knowledge Base article before you upgrade to the latest version of Tableau Server: Gateway and Prep Conductor failed to start when using External SSL with passphrase to protect the key file after upgrade to Tableau Server 2022.1.17.

- SSL certificate chain file: A certificate chain file is required for Tableau Desktop on the Mac and for Tableau Prep Builder on the Mac and Tableau Prep Builder on Windows. The chain file is also required for the Tableau Mobile app if the certificate chain for Tableau Server is not trusted by the iOS or Android operating system on the mobile device.

  The chain file is a concatenation of all of the certificates that form the certificate chain for the server certificate. All certificates in the file must be x509 PEM-encoded and the file must have a `.crt` extension (not `.pem`).

- For multiple sub-domains, Tableau Server supports wildcard certificates.

- Verify that the domain, host name, or IP address that clients use to connect to Tableau Server is included in the Subject Alternative Names (SAN) field. Many clients (Tableau Prep, Chrome and Firefox browsers, etc) require valid entry in the SAN field to establish a secure connection.

Note: If you plan to configure Tableau Server for single-sign on using SAML, see Using SSL certificate and key files for SAML in the SAML requirements to help determine whether to use the same certificate files for both SSL and SAML.

Configuring SSL for a Cluster

You can configure a Tableau Server cluster to use SSL. If the initial node is the only one running the gateway process (which it does by default), you need to configure SSL only on that node, using the steps described in this topic.

SSL with multiple gateways

A highly available Tableau Server cluster can include multiple gateways, fronted by a load balancer. If you are configuring this type of cluster for SSL, you have the following choices:

- **Configure the load balancer for SSL**: Traffic is encrypted from the client web browsers to the load balancer. Traffic from the load balancer to the Tableau Server gateway processes is not encrypted. No SSL configuration in Tableau Server is required by you. It's all handled by the load balancer.
- **Configure Tableau Server for SSL**: Traffic is encrypted from the client web browsers to the load balancer, and from the load balancer to the Tableau Server gateway processes. For more information, continue to the following section.

Additional configuration information for Tableau Server cluster environments

When you want to use SSL on all Tableau Server nodes that run a gateway process, you complete the following steps.

1. Configure the external load balancer for SSL passthrough.

   Or if you want to use a port other than 443, you can configure the external load balancer to terminate the non-standard port from the client. In this scenario, you would then configure the load balancer to connect to Tableau Server over port 443. For assistance, refer to the documentation provided for the load balancer.

2. Make sure the SSL certificate is issued for the load balancer's host name.

3. Configure the initial Tableau Server node for SSL.

4. If you are using mutual SSL, upload the SSL CA certificate file. See `tsm authentication mutual-ssl <commands>`.

SSL certificate and key files will be distributed to each node as part of the configuration process.

Prepare the environment

When you get the certificate files from the CA, save them to a location accessible by Tableau Server, and note the names of the certificate .crt and .key files and the location where you save them. You will need to provide this information to Tableau Server when you enable SSL.

Configure SSL on Tableau Server

Use the method you're most comfortable with.

Use the TSM web interface

1. Open TSM in a browser:

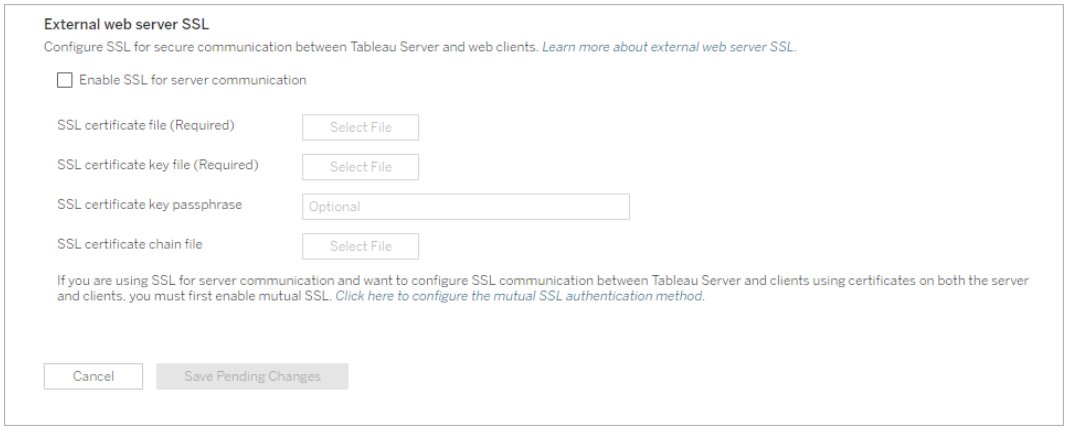
   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. On the **Configuration** tab, select **Security** > **External SSL**.

   > **Note**: If you are updating or changing an existing configuration, click **Reset** to clear the existing settings before proceeding.

3. Under **External web server SSL**, select **Enable SSL for server communication**.

4. Upload the certificate and key files, and if required for your environment, upload the chain file and enter the passphrase key:



   If you are running Tableau Server in a distributed deployment, then these files will be automatically distributed to each appropriate node in the cluster.

5. Click **Save Pending Changes**.

6. Click **Pending Changes** at the top of the page:

7. Click **Apply Changes and Restart**.

Use the TSM CLI

After you have copied the certificate files to the local computer, run the following commands:

```
tsm security external-ssl enable --cert-file <path-to-file.crt> --
key-file <path-to-file.key>

tsm pending-changes apply
```

See the command reference at tsm security external-ssl enable to determine whether you want to include additional options for `external-ssl enable`. Tableau has specific recommendations for the `--protocols` option.

The `external-ssl enable command` imports the information from the .crt and .key files. If you run this command on a node in a Tableau Server cluster, it also distributes the information to any other gateway node.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Port redirection and logging

After the server has been configured for SSL, it accepts requests to the non-SSL port (default is port 80) and automatically redirects to the SSL port 443.

Note: Tableau Server supports only port 443 as the secure port. It cannot run on a computer where another application is using port 443.

SSL errors are logged in the at the following location. Use this log to troubleshoot validation and encryption issues:

```
/var/opt/tableau/tableau_server/data/tabsvc/logs/httpd/error.log
```

Add SSL port to the local firewall

If you are running a local firewall, you must add the SSL port to the firewall on Tableau Server. The example below describes how to configure the firewall running on RHEL/CentOS distributions. The example uses Firewalld, which is the default firewall on CentOS.

1. Start firewalld:

   ```
   sudo systemctl start firewalld
   ```

2. Add port 443 for SSL:

   ```
   sudo firewall-cmd --permanent --add-port=443/tcp
   ```

3. Reload the firewall and verify the settings:

   ```
   sudo firewall-cmd --reload
   ```

   ```
   sudo firewall-cmd --list-all
   ```

Change or update SSL certificate

After you have configured SSL, you may need to periodically update the certificate. In some cases, you may need change the certificate for operational changes in your IT environment. In either case, you must use TSM to replace the SSL certificate that has already been configured for external SSL.

Do not copy a new certificate to the file directory on the operating system. Rather, when you add the certificate with either the TSM web UI or the `tsm security external-ssl enable` command, the certificate file is copied to the appropriate certificate store. In a distributed deployment, the certificate is also copied across the nodes in the cluster.

To change or update the SSL certificate (and the corresponding key file if required), follow the steps in the previous section of this topic, Configure SSL on Tableau Server.

After you change the certificate, you must run `tsm pending-changes apply` to restart Tableau Server services. We also recommend restarting any other services on the computer that use the SSL certificate. If you are changing a root certificate on the operating system, you must reboot the computer.

Example: SSL Certificate - Generate a Key and CSR

Important: This example is intended to provide general guidance to IT professionals who are experienced with SSL requirements and configuration. The procedure described in this article is just one of many available methods you can use to generate the required files. The process described here should be treated as an example and not as a recommendation.

When you configure Tableau Server to use Secure Sockets Layer (SSL) encryption, this helps ensure that access to the server is secure and that data sent between Tableau Server and Tableau Desktop is protected.

> Looking for Tableau Server on Windows? See Example: SSL Certificate - Generate a Key and CSR.

Tableau Server uses Apache, which includes OpenSSL. You can use the OpenSSL toolkit to generate a key file and Certificate Signing Request (CSR) which can then be used to obtain a signed SSL certificate.

Note: Beginning in Tableau Server versions 2021.3.26, 2021.4.21, 2022.1.17, 2022.3.9, 2023.1.5, and later, Tableau Server runs OpenSSL 3.1.

Steps to generate a key and CSR

To configure Tableau Server to use SSL, you must have an SSL certificate. To obtain the SSL certificate, complete the steps:

1. Generate a key file.
2. Create a Certificate Signing Request (CSR).
3. Send the CSR to a certificate authority (CA) to obtain an SSL certificate.
4. Use the key and certificate to configure Tableau Server to use SSL.

You can find additional information on the SSL FAQ page on the Apache Software Foundation website.

Configure a certificate for multiple domain names

Tableau Server allows SSL for multiple domains. To set up this environment, you need to modify the OpenSSL configuration file, openssl.conf, and configure a Subject Alternative Name (SAN) certificate on Tableau Server. See For SAN certificates: modify the OpenSSL configuration file below.

Generate a key

Generate a key file that you will use to generate a certificate signing request.

1. Run the following command to create the key file:

   ```
   openssl genrsa -out <yourcertname>.key 4096
   ```

   Notes:
   - This command uses a 4096-bit length for the key. You should choose a bit length that is at least 2048 bits because communication encrypted with a shorter bit length is less secure. If a value is not provided, 512 bits is used.
   - To create PKCS#1 RSA keys with Tableau Server versions 2021.3.26, 2021.4.21, 2022.1.17, 2022.3.9, 2023.1.5, and later, you must use the additional option `-traditional` when running `openssl genrsa`" command based on the OpenSSL 3.1. For more information about the option, see https://www.openssl.org/docs/man3.1/man1/openssl-rsa.html.

Create a certificate signing request to send to a certificate authority

Use the key file you created in the procedure above to generate the certificate signing request (CSR). You send the CSR to a certificate authority (CA) to obtain a signed certificate.

Important: If you want to configure a SAN certificate to use SSL for multiple domains, first complete the steps in For SAN certificates: modify the OpenSSL configuration file below, and then return to here to generate a CSR.

1. Run the following command to create a certificate signing request (CSR) file:

   ```
   openssl req -new -key yourcertname.key -out yourcertname.csr  -config /opt/tableau/tableau_server-/packages/apache.<version>/conf/openssl.cnf
   ```

2. When prompted, enter the required information.

   Note: For **Common Name**, type the Tableau Server name. The Tableau Server name is the URL that will be used to reach the Tableau Server. For example, if you reach Tableau Server by typing `tableau.example.com` in the address bar of your browser, then `tableau.example.com` is the common name. If the common name does not resolve to the server name, errors will occur when a browser or Tableau Desktop tries to connect to Tableau Server.

Send the CSR to a certificate authority to obtain an SSL certificate

Send the CSR to a commercial certificate authority (CA) to request the digital certificate. For information, see the Wikipedia article Certificate authority and any related articles that help you decide which CA to use.

Use the key and certificate to configure Tableau Server

When you have both the key and the certificate from the CA, you can configure Tableau Server to use SSL. For the steps, see Configure External SSL.

For SAN certificates: modify the OpenSSL configuration file

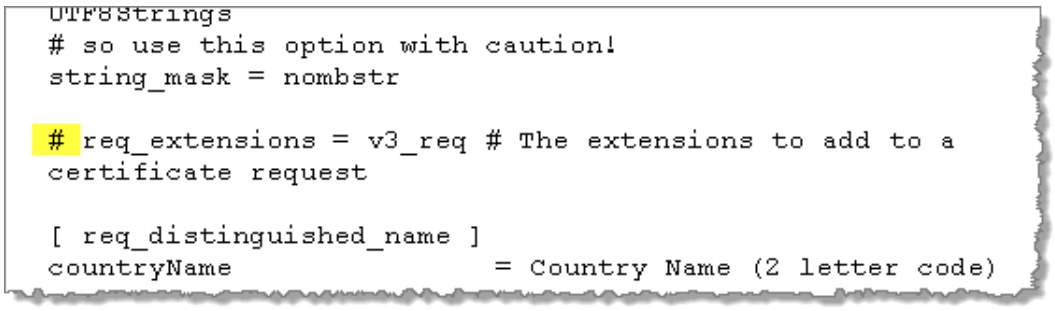
In a standard installation of OpenSSL, some features are not enabled by default. To use SSL with multiple domain names, before you generate the CSR, complete these steps to modify the **openssl.cnf** file.

1. Navigate to the Apache **conf** folder for Tableau Server.

   For example: `/opt/tableau/tableau_server/packages/apache.<version_code>/conf`

2. Open **openssl.cnf** in a text editor, and find the following line: `req_extensions = v3_req`

   This line might be commented out with a hash sign (#) at the beginning of the line.

   
   ```
   UTF8Strings
   # so use this option with caution!
   string_mask = nombstr

   # req_extensions = v3_req # The extensions to add to a
   certificate request

   [ req_distinguished_name ]
   countryName                 = Country Name (2 letter code)
   ```

   If the line is commented out, uncomment it by removing the **#** and **space** characters from the beginning of the line.

3. Move to the **[v3_req]** section of the file. The first few lines contain the following text:

   ```
   # Extensions to add to a certificate request
   basicConstraints = CA:FALSE
   keyUsage = nonRepudiation, digitalSignature, keyEncipherment
   ```

After the **keyUsage** line, insert the following line:

```
subjectAltName = @alt_names
```

If you're creating a self-signed SAN certificate, do the following to give the certificate permission to sign the certificate:

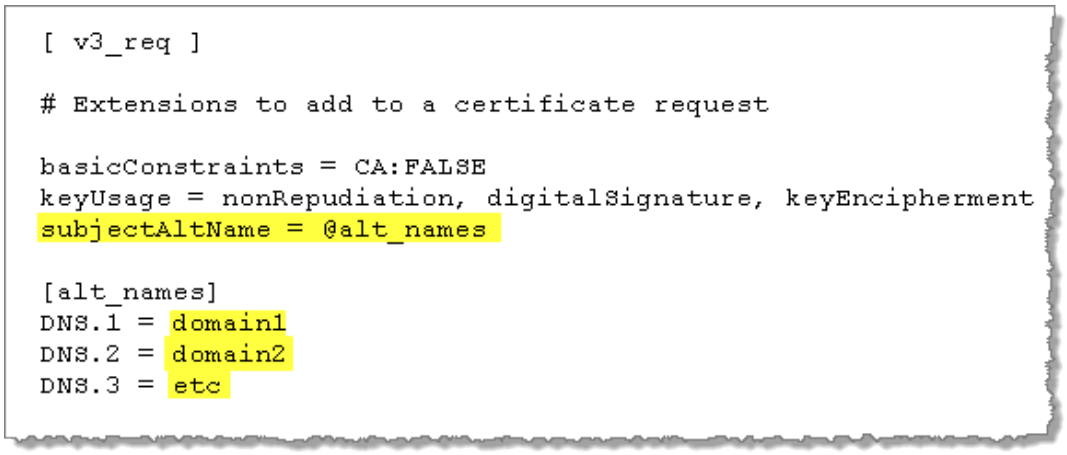
a. Add the `cRLSign` and `keyCertSign` to the **keyUsage** line so it looks like the following: `keyUsage = nonRepudiation, digitalSignature, keyEncipherment, cRLSign, keyCertSign`

b. After the **keyUsage** line, add the following line: `subjectAltName = @alt_names`

4. In the **[alt_names]** section, provide the domain names you want to use with SSL.

```
DNS.1 = [domain1]
DNS.2 = [domain2]
DNS.3 = [etc]
```

The following image shows the results highlighted, with placeholder text that you would replace with your domain names.



5. Save and close the file.

6. Complete the steps in Create a certificate signing request to send to a certificate authority section, above.

Configure SSL for Internal Postgres Communication

You can configure Tableau Server to use SSL (TLS) for encrypted communication between the Postgres repository and other server components. By default, communication that is internal to Tableau Server components is not encrypted.

While you enable support for internal SSL, you can also configure support for direct connections to the repository from Tableau clients, such as Tableau Desktop, Tableau Mobile, REST API, web browsers.

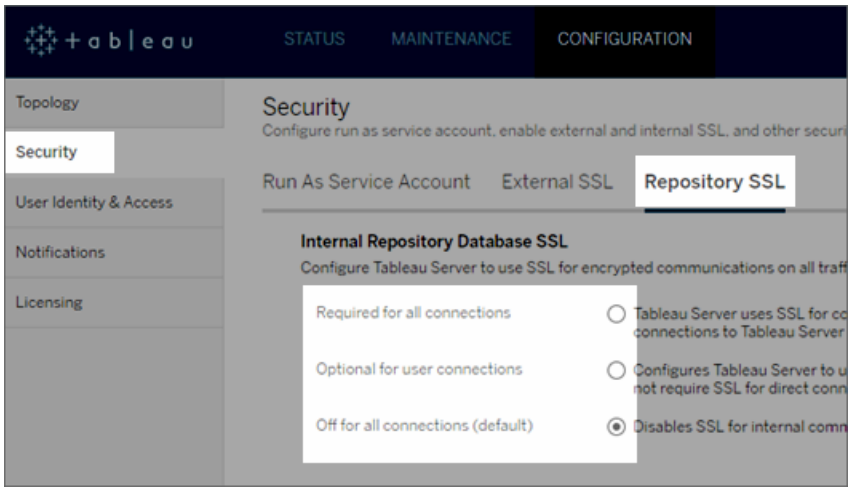
Use the TSM web interface

1. As a server administrator, open TSM in a browser:

   https://<tsm-computer-name>:8850

   For more information, see Sign in to Tableau Services Manager Web UI.

2. On the **Configuration** tab, select **Security** > **Repository SSL**.



3. Select one of the options for using repository SSL.

- **Required for all connections**—uses SSL for internal Tableau Server communication, and requires SSL for Tableau clients and any external (non-Tableau) clients that connect directly to the postgres repository, including those using the **tableau** or **readonly** user.

  > **Important:** Unless you complete the steps in Configure Postgres SSL to Allow Direct Connections from Clients, to place the certificate files in the correct location on the client computers, Tableau clients and external postgres clients will not be able to validate the identity of the Tableau repository by comparing certificates on the client computers with the SSL certificate from the repository computer.

- **Optional for user connections**—When enabled, Tableau uses SSL for internal Tableau Server communication, and supports but does not require SSL for direct connections to the server from Tableau clients and external clients.

- **Off for all connections (default)**—Internal server communication is not encrypted, and SSL is not required for direct connections from clients.

4. Click **OK**.

   The first two options generate the server's certificate files, **server.crt** and **server.key**, and place them in the following location.

   ```
   /var/opt/tableau/tableau_server/data/tabsvc/config/pgsql_<version>/security
   ```

   Use this .crt file if you need to configure clients for direct connections.

Use the TSM CLI

To enable SSL for internal traffic among the server components, run the following commands:

```
tsm security repository-ssl enable
```

```
tsm pending-changes apply
```

What the command does

`repository-ssl enable` generates the server's certificate files, which it places in the following location:

`/var/opt/tableau/tableau_server/data/tabsvc/config/pgsql_<version>/security`

By default, this command sets Tableau Server to require SSL for traffic between the repository and other server components, as well as for direct connections from Tableau clients (including for connections through the **tableau** or **readonly** users).

To complete the configuration, you must also do the steps described in Configure Postgres SSL to Allow Direct Connections from Clients, to place the certificate files in the correct location on the client computers.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Option for repository-ssl enable

If you want to require SSL only for internal Tableau Server communication, and not for direct connections from client apps, use the following option with the `repository-ssl enable` command:

`--internal-only`

Cluster environments

If you run `repository-ssl enable` on a node in a cluster, it copies the required certificate file to the same location on each other node.

For more information about downloading the public certificate for direct connections, see Configure Postgres SSL to Allow Direct Connections from Clients.

Configure Custom SSL Certificate for TSM Controller

The Tableau Server Administration Controller (aka, Controller) is the management component for administration changes to the Tableau Server cluster. By default the Controller runs on the initial (first) node of a Tableau Server cluster. Although it is technically possible to run multiple Controllers in a single Tableau cluster deployment, this is not a recommended practice.

The Controller includes an API that can be managed by various clients: TSM CLI, TSM Web Client, REST clients (curl, postman), etc. Using these clients, Tableau Server administrators can make configuration changes to the server cluster. The Controller, along with Zookeeper, manages and performs the configuration changes across the nodes.

Default TSM SSL functionality

> **Note** As is convention, the term “SSL” is used here when referring to using TLS to secure HTTPS traffic.

By default, the client connection is encrypted with SSL by means of a self-signed certificate that is created by Tableau Server during setup and renewed by the Controller. In addition to encryption, the identity (hostname or IP) of the Controller host machine is validated against the subject name presented in the certificate during the SSL handshake. However, because the certificate is self-signed, the trustworthiness of the certificate is not absolute .

In the case of CLI connection to the Controller, the inability to absolutely trust the certificate is not a huge security risk, since a man in the middle attack would generally require a malicious user access to the Tableau Server cluster in a private network. If a malicious user can spoof

the certificate for the controller in CLI scenario then the malicious user already has “the keys to the kingdom.”

However, in the scenario where administrators are connecting to the Controller over TSM Web UI from outside the internal network, the lack of host validation via trusted certificate authority presents more of a security risk.

Until recently, customers running TSM Web UI on a Windows machine could place the Tableau Server CA certificate in a Windows trusted root store. Most browsers would validate the trust of the certificate by virtue of this configuration. Today, Chrome no longer validates (trusts) self-signed certificates that are placed in the OS trust store. Now, Chrome (and most major browsers) will only trust certificates that chain back to a trusted third-party root CA.

Tableau Server v2023.1 SSL custom certificate

The custom SSL TSM certificate feature closes the trust gap by allowing administrators to configure the TSM Controller with an identity certificate that chains back to a trusted third-party root CA.

There are a number of important details to understand:

- Trust for the TSM custom SSL certificate is validated when connecting with TSM Web UI.
- Trust validation is not attempted for TSM CLI scenario. As described previously, a “man-in-the-middle” attack on the CLI scenario does not present a credible risk.
- Certificate chain may be included in the configuration. The chain may present all certificates signed by intermediate CAs. The chain can end at any point, and any certificates missing from the chain are presumed to be installed in the operating system trust store.

Configuration

You must use TSM CLI, to configure (or update) SSL custom certificate for TSM.

See tsm security custom-tsm-ssl enable.

Configure Postgres SSL to Allow Direct Connections from Clients

When Tableau Server is configured to use SSL for internal communication with the postgres repository, you can also require Tableau clients and external postgres clients that connect directly to the repository to verify the identity of the Tableau postgres repository by comparing the SSL certificate presented by the internal postgres instance wit the certificate distributed to the Tableau or external postgres client.

Direct connections include those using the **tableau** user or the **readonly** user. Examples of Tableau clients include Tableau Desktop, Tableau Mobile, REST API, web browsers.

1. Enable internal SSL for the repository by running the following commands:

   ```
   tsm security repository-ssl enable

   tsm pending-changes apply
   ```

   This enables internal SSL support and generates new server certificate and key files, and requires all Tableau clients to use SSL to connect to the repository. For additional repository-ssl commands and options, see tsm security.

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

2. (Optional) If you have configured your client computer to validate Postgres SSL connections, then you must import the certificate that is generated by Tableau Server onto the computers running Tableau Desktop. For each client computer that will connect directly to the repository, do the following:

   - Copy the **server.crt** file to the client computer. You can find this file in the following directory:

`/var/opt/tableau/tableau_server/data/tabsvc/config/pgsql_<version_code>/security`

> **Note**: Do not copy **server.key** to the client computer. This file should reside only on the server.

- Import the certificate into the computer's certificate store.

  For information, use the documentation from the operating system manufacturer.

3. (Optional) Configure any external (non-Tableau) postgres clients (PgAdmin or Dbeaver for example) to verify the identity of the Tableau Server postgres repository. Do this in the postgresql JDBC driver the client is using to connect by setting the "sslmode" directive to "verify-ca" or "verify-full". The options available may be different depending on the version of the postgres driver being used. For more information, see the drive documentation about SSL support.

Configure Mutual SSL Authentication

Using mutual SSL, you can provide users of Tableau Desktop, Tableau Mobile, and other approved Tableau clients a secure, direct-access experience to Tableau Server. With mutual SSL, when a client with a valid SSL certificate connects to Tableau Server, Tableau Server confirms the existence of the client certificate and authenticates the user, based on the user name in the client certificate. If the client does not have a valid SSL certificate, Tableau Server can refuse the connection.

You can also configure Tableau Server to fall back to username/password authentication if mutual SSL fails. Additionally, a user can log in using the REST API with a username and password (if one exists) whether or not fallback authentication is configured.

User authentication session time limits

When users log in with mutual SSL, the authentication session is governed by the same method that governs the Tableau Server global authentication session configuration.

For clients that connect to Tableau Server using a web browser, configuration of the global authentication session is described in the *Security Hardening Checklist*, see 9. Verify session lifetime configuration.

Sessions for connected clients (Tableau Desktop, Tableau Mobile, Tableau Prep Builder, and Bridge) use OAuth tokens to keep users logged in by re-establishing a session. By default, OAuth client tokens reset after a year. If a client token has not been used in 14 days, then it will expire. You can change these values by setting the `refresh_token.absolute_expiry_in_seconds` and `refresh_token.idle_expiry_in_seconds` options. See tsm configuration set Options.

Certificate usage

Before you enable and configure mutual SSL, you must configure external SSL. External SSL authenticates Tableau Server to the client and encrypts the session using the certificate and key that is required when you configure external SSL.

For mutual SSL, an additional certificate file is required. The file is a concatenation of CA certificate files. The file type must be `.crt`. A "CA" is a *certificate authority* that issues certificates to the client computers that will connect to Tableau Server. The action of uploading the CA certificate file establishes a trust, which enables Tableau Server to authenticate the individual certificates that are presented by the client computers.

As part of your disaster recovery plan, we recommend keeping a backup of the certificate and revocation (if applicable) files in a safe location off of the Tableau Server. The certificate and revocation files that you add to Tableau Server will be stored and distributed to other nodes by the Client File Service. However, the files are not stored in a recoverable format. See Tableau Server Client File Service.

RSA key and ECDSA curve sizes

The CA certificate used for mutual SSL must either have an RSA key strength of 2048, or ECDSA curve size of 256.

.You can configure Tableau Server to accept the less-secure sizes by setting the respective configuration keys:

- ssl.client_certificate_login.min_allowed.rsa_key_size
- ssl.client_certificate_login.min_allowed.elliptic_curve_size

See tsm configuration set Options.

Client certificate requirements

Users authenticating to Tableau Server with mutual SSL must present a client certificate that meets minimum security requirements.

Signing algorithm

Client certificates must use a SHA-256 or greater signing algorithm.

Tableau Server configured for mutual SSL authentication will block authentication of users with client certificates that use the SHA-1 signing algorithm.

Users who attempt to log in with SHA-1 client certificates encounter an "Unable to sign in" error, and the following error will be visible in the VizPortal logs:

```
Unsupported client certificate signature detected: [certificate Sig-
nature Algorithm name]
```

You can configure Tableau Server to accept the less secure SHA-1 signing algorithm by setting the ssl.client_certificate_login.blocklisted_signature_algorithms tsm configuration option.

RSA key and ECDSA curve sizes

The client certificate used for mutual SSL must either have an RSA key strength of 2048, or ECDSA curve size of 256.

Tableau Server will fail mutual authentication requests from client certificates that do not meet these requirements. You can configure Tableau Server to accept the less-secure sizes by setting the respective configuration keys:

- ssl.client_certificate_login.min_allowed.rsa_key_size
- ssl.client_certificate_login.min_allowed.elliptic_curve_size

See tsm configuration set Options.

Use the TSM web interface

1. Configure SSL for External HTTP Traffic to and from Tableau Server.
2. Open TSM in a browser:

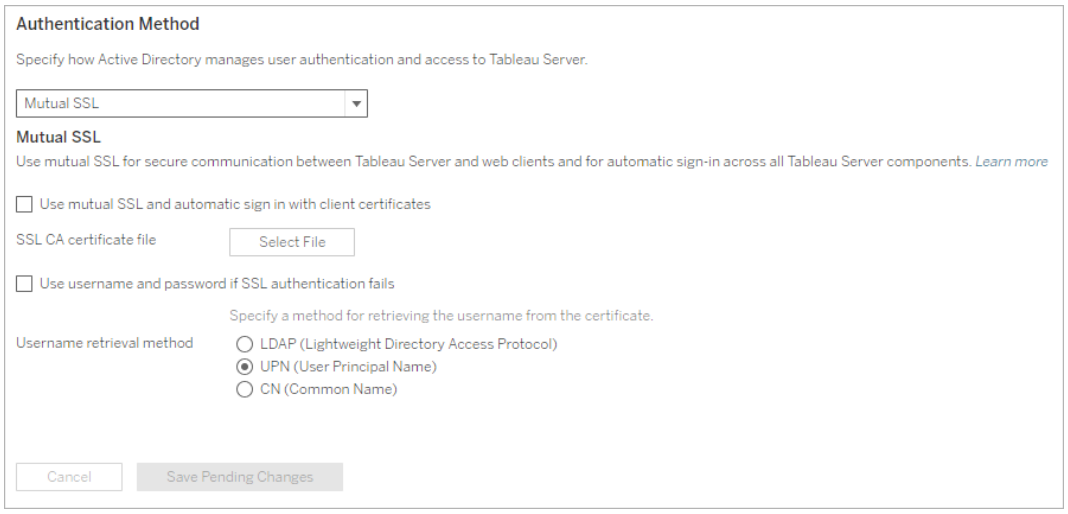
   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.
3. On the **Configuration** tab, select **User Identity & Access** > **Authentication Method**.
4. Under **Authentication Method**, select **Mutual SSL** in the drop-down menu.
5. Under Mutual SSL, select **Use mutual SSL and automatic sign in with client certificates**.
6. Click **Select File** and upload your certificate authority (CA) certificate file to the server.

   The file (.crt) is an all-in-one file that includes certificates of CAs that are used for client authentication. The file you upload must be a concatenation of the various PEM-encoded certificate files, in order of preference.
7. Enter remaining SSL configuration information for your organization.

   Username format: When Tableau Server is configured for mutual SSL, the server gets the user name from the client certificate, so it can establish a direct sign-in for the client user. The name that Tableau Server uses depends on how Tableau Server is configured for user authentication:

- Local Authentication—Tableau Server uses the UPN (User Principal Name) from the certificate.
- Active Directory (AD)—Tableau Server uses LDAP (Lightweight Directory Access Protocol) to get the user name.

Alternatively, you can set Tableau Server to use the CN (Common Name) from the client certificate.



8. Click **Save Pending Changes** after you've entered your configuration information.
9. Click **Pending Changes** at the top of the page:

10. Click **Apply Changes and Restart**.

Use the TSM CLI

Step 1: Require SSL for external server communication

To configure Tableau Server to use SSL for external communication between Tableau Server and web clients, run the `external-ssl enable` command as follows, providing the names for the server certificate's .crt and .key files:

```
tsm security external-ssl enable --cert-file <file.crt> --key-
file <file.key>
```

- For `--cert-file` and `--key-file`, specify the location and file name where you saved the server's CA-issued SSL certificate (.crt) and key (.key) files.
- The above command assumes the you are signed in as a user that has the **Server Administrator** site role on Tableau Server. You can instead use the `-u` and `-p` parameters to specify an administrator user and password.
- If the certificate key file requires a passphrase, include the `--passphrase` parameter and value.

Step 2: Configure and enable mutual SSL

Add mutual authentication between the server and each client, and allow for Tableau client users to be authenticated directly after the first time they provide their credentials.

1. Run the following command:

   ```
   tsm authentication mutual-ssl configure --ca-cert <certificate-
   file.crt>
   ```

   For `--ca-cert`, specify the location and file name of the certificate authority (CA) certificate file.

   The file (.crt) is an all-in-one file that includes certificates of CAs that are used for client authentication. The file you upload must be a concatenation of the various PEM-encoded certificate files, in order of preference.

2. Run the following commands to enable mutal SSL and apply the changes:

   ```
   tsm authentication mutual-ssl enable
   ```

   ```
   tsm pending-changes apply
   ```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Additional options for mutual SSL

You can use `mutual-ssl configure` to configure Tableau Server to support the following options.

For more information, see tsm authentication mutual-ssl <commands>.

Fallback authentication

When Tableau Server is configured for mutual SSL, authentication is automatic and clients must have a valid certificate. You can configure Tableau Server to allow a fallback option, to accept user name and password authentication.

```
tsm authentication mutual-ssl configure -fb true
```

Tableau Serer accepts username and password authentication from REST API clients, even if the above option is set to `false`.

User name mapping

When Tableau Server is configured for mutual SSL, the server authenticates the user directly by getting the user name from their client certificate. The name that Tableau Server uses depends on how the server is configured for user authentication:

- **Local Authentication**—uses the UPN (User Principal Name) from the certificate.
- **Active Directory (AD)**—uses LDAP (Lightweight Directory Access Protocol) to get the user name.

You can override either of these defaults to set Tableau Server to use the common name.

```
tsm authentication mutual-ssl configure -m cn
```

For more information, see Mapping a Client Certificate to a User During Mutual Authentication

Certificate Revocation List (CRL)

You might need to specify a CRL if you suspect that a private key has been compromised, or if a certificate authority (CA) did not issue a certificate properly.

```
tsm authentication mutual-ssl configure -rf <revoke-file.pem>
```

Mapping a Client Certificate to a User During Mutual Authentication

When you use mutual (two-way) SSL authentication, the client presents its certificate to Tableau Server as part of the authentication process. Tableau Server then maps user information in the client certificate to a known user identity. The strategy that Tableau Server uses to perform client mapping depends on the content of your organization's client certificates.

This topic discusses the ways information in a client certificate can map to a user identity and how to change the way Tableau Server performs that mapping. To understand how the mapping happens and whether you need to change it, you must know how client certificates are structured in your organization.

- User-name mapping options
- Change the certificate mapping
- Address user-name ambiguity in multi-domain organizations

User-name mapping options

Tableau Server uses one of the following approaches to map a client certificate to a user identity:

- **Active Directory**. If Tableau Server is configured to use Active Directory for user authentication, when Tableau Server receives a client certificate, it passes the certificate to Active Directory, which maps the certificate to an Active Directory identity. Any

explicit user name information in the certificate is ignored.

> **Note**: This approach requires client certificates to be published for the user accounts in Active Directory.

- **User principal name (UPN)**. A client certificate can be configured to store the user name in the user principal name field. Tableau Server reads the UPN value and maps it to a user in Active Directory or to a local user.
- **Common name (CN)**. A client certificate can be configured to store the user name in the common name field of the certificate. Tableau Server reads the CN value and maps it to a user in Active Directory or to a local user.

If you configure the server for Active Directory authentication and UPN or CN user-name mapping, put the user name in one of the following formats:

`username`, `domain/username`, or `username@domain`.

For example: `jsmith`, `example.org/jsmith`, or `jsmith@example.org`.

If the server uses local authentication, the format of the name in the UPN or CN fields is not predetermined, but the name in the field must match a user name on the server.

Change the certificate mapping

You use the tsm authentication mutual-ssl <commands> commands to map a client certificate to a user identity in Tableau Server:

```
tsm authentication mutual-ssl configure -m <value>
```

Possible values are `ldap` for Active Directory mapping, `upn` for UPN mapping, or `cn` for CN mapping.

When you first install and configure Tableau Server, the server sets the default user-name mapping to match the server's authentication type:

- If the server is configured to use Active Directory, it also uses Active Directory for mapping the certificate to the user identity.
- If the server is configured to use local authentication, the server gets the user-name value from the UPN field in the certificate.

If the default behavior for how Tableau Server maps a user name to an identity is not correct for your server configuration, run the following set of commands to change the mapping to use the CN value:

```
tsm authentication mutual-ssl configure -m cn

tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Address user-name mapping ambiguity in multi-domain organizations

Under some circumstances, the user name in a certificate's UPN or CN field can be ambiguous. This ambiguity can lead to unexpected results when the user name is mapped to a user identity on the server.

For example, if Tableau Server is presented with a user name that does not include a domain, the server maps the user name to an identity using the default domain. This can cause an incorrect user-name mapping, potentially assigning a user a different user's identity and permissions.

This can occur particularly in environments where the following conditions apply:

- Your organization supports multiple Active Directory domains.
- The server is configured to use Active Directory authentication.

- The server is configured to use UPN or CN mapping.
- Some users have the same user name but different domains. For example, `jsmith@example.org` and `jsmith@example.com`.
- The user name in the certificate's UPN or CN fields does not include the domain as part of the user name—for example, it shows `jsmith`.

To avoid incorrect user-name mapping, make sure the client certificates include fully qualified user names with the domain, using the format `jsmith@example.org` or `example.org/jsmith`.

Configure Encrypted Channel to LDAP External Identity Store

Tableau Server that is configured to connect to an external LDAP identity store must query the LDAP directory and establish a session. The process of establishing a session is called *binding*. There are multiple ways to bind. Tableau Server supports two methods of binding to an LDAP directory:

- **Simple bind:** Establishes a session by authenticating with a username and password. By default, Tableau Server will attempt StartTLS to encrypt sessions when connecting to Windows Active Directory. If Tableau Server has a valid TLS certificate, then the session will be encrypted. Otherwise, LDAP with simple bind is not encrypted. If you are configuring LDAP with simple bind, we strongly recommend that you enable LDAP over SSL/TLS.
- **GSSAPI bind:** GSSAPI uses Kerberos to authenticate. When configured with a keytab file, authentication is secure during GSSAPI bind. However, subsequent traffic to the LDAP server is not encrypted. We recommend configuring LDAP over SSL/TLS . **Important:** StartTLS is not supported for GSSAPI bind with Active Directory.

  If you are running Tableau Server on Linux on a computer that is joined to an Active Directory domain, you can configure GSSAPI. See LDAP with GSSAPI (Kerberos) bind.

This topic describes how to encrypt the channel for **simple LDAP bind** for communications between Tableau Server and LDAP directory servers.

Certificate requirements

- You must have a valid PEM-encoded x509 SSL/TLS certificate that can be used for encryption. The certificate file must have an extension .crt.
- Self-signed certificates are not supported.
- The certificate you install must include `Key Encipherment` in the key usage field to be used for SSL/TLS. Tableau Server will only use this certificate for encrypting the channel to the LDAP server. The expiry, trust, and CRL and other attributes are not validated.
- If you are running Tableau Server in a distributed deployment, then you must manually copy the SSL certificate to each node in the cluster. Copy the certificate only to those nodes where the Tableau Server Application Server process is configured. Unlike other shared files in a cluster environment, the SSL certificate used for LDAP will not be automatically distributed by the Client File Service.
- If you are using a PKI or non-3rd party certificate, upload the CA root certificate to the Java trust store.

Import certificate into the Tableau keystore

If you do not have certificates already in place on your computer that are configured for the LDAP server then you must obtain a SSL certificate for the LDAP server and import it into the Tableau system keystore.

Use the "keytool" Java tool to import certificates. In a default installation, this tool is installed with Tableau Server in the following location:

`/opt/tableau/tableau_server/packages/repository.<installer version>/jre/bin/keytool`.

The following command imports the certificate:

```
sudo "<installation_directory>/packages/repository*/jre/bin/keytool
-importcert -file "<cert_directory/<cert_name.crt>" -alias "<cert_
alias>" -keystore /etc/opt/tableau/tableau_server-
/tableauservicesmanagerca.jks -storepass changeit -noprompt
```

The password for the Java keystore is `changeit`. (Do not change the password for the Java keystore).

Encryption methods

Tableau Server 2021.1 and newer supports two methods for encrypting the LDAP channel for simple bind: StartTLS and LDAPS.

- **StartTLS:** This is the default configuration for communicating with Active Directory in Tableau Server 2021.2. Beginning with Tableau Server 2021.2, TLS is enforced for simple bind LDAP connections to Active Directory. This default TLS configuration is enforced for both new installations and for upgrade scenarios.

  > **Note**: StartTLS is only supported on Tableau Server on Linux when communicating with Active Directory and simple bind. StartTLS is not supported for communication with other LDAP server types or with GSSAPI.

  The StartTLS method works by establishing an insecure connection with the Active Directory server. After a client-server negotiation, the connection is upgraded to a TLS encrypted connection. As the default configuration, this scenario only requires a valid TLS certificate on Tableau Server. No other configuration is required.

- **LDAPS:** Secure LDAP, or LDAPS, is a standard encrypted channel that requires more configuration. Specifically, in addition to a TLS certificate on Tableau Server, you must

set the host name and the secure LDAP port for the target LDAP server.

LDAPS is supported on any LDAP server, including Active Directory servers.

Configure encrypted channel for simple bind

This section describes how to configure Tableau Server to use an encrypted channel for LDAP simple bind.

When to configure

You must configure Tableau Server to use an encrypted channel for LDAP simple bind before Tableau Server is initialized or as part of configuring the initial node as mentioned in the “Use the TSM CLI” tab in Configure Initial Node Settings.

For new installations of Tableau Server

If your organization uses an LDAP directory other than Active Directory, then you cannot use the TSM GUI Setup to configure the identity store as part of Tableau Server installation. Instead, you must use JSON entity files to configure the LDAP identity store. See identityStore Entity.

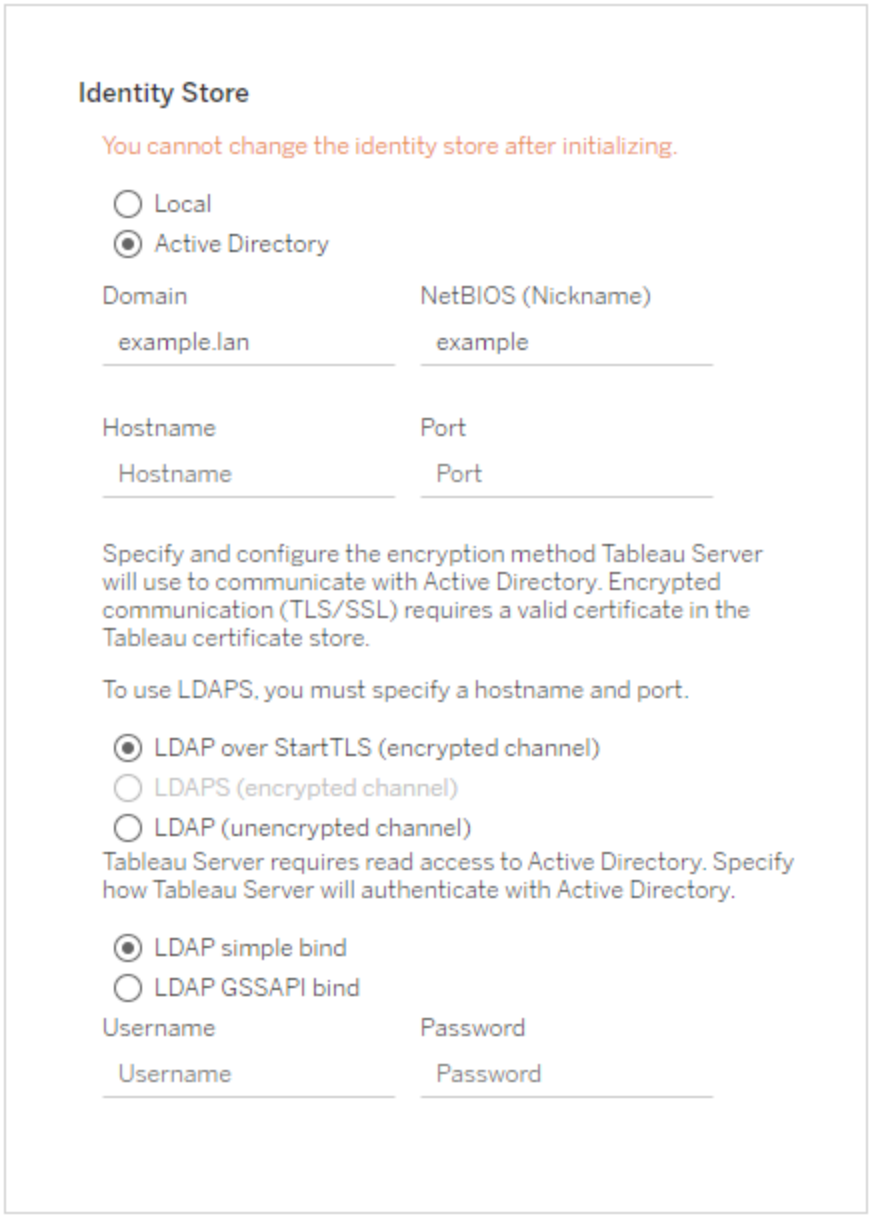
Before you configure the identityStore entity, import a valid SSL/TLS certificate into the Tableau key store as documented earlier in this topic.

Configuring LDAPS requires setting the hostname and sslPort options in the identityStore JSON file.

For new installations in an Active Directory environment

If you are using Active Directory as an external identity store, you must run the GUI version of Tableau Server Setup. Unlike the CLI process for installing Tableau Server, the GUI version of Setup includes logic to simplify and validate Active Directory configuration.

The Tableau Server Setup GUI where you configure Active Directory is shown here.



If you are installing a new instance of Tableau Server on Linux and you have a valid SSL/TLS certificate installed in the Tableau keystore, we recommend that you leave the default option set to StartTLS.

If you want to configure for LDAPS, then enter the hostname and secure port (typically 636) for the LDAP server, before selecting the LDAPS option.

You can make changes to these configurations after you install by signing into TSM Web UI, clicking the **Configuration** tab, **User Identity & Access**, and then **Identity Store**.

Upgrade scenarios

If you are upgrading to a 2021.2 (or newer) version of Tableau Server and using Active Directory as your external identity store, then the encrypted channel is enforced for LDAP simple bind connections. If you do not have an encrypted channel configured, then upgrade will fail.

To successfully upgrade to version 2021.2 or newer, one of the following must be true:

- The existing Tableau Server installation has already been configured for LDAPS and includes a certificate in the Tableau key store.
- A valid SSL/TLS certificate is present in the Tableau key store prior to upgrading. In this scenario, the default StartTLS configuration will enable an encrypted channel.
- The encrypted LDAP channel has been disabled as described in the following section.

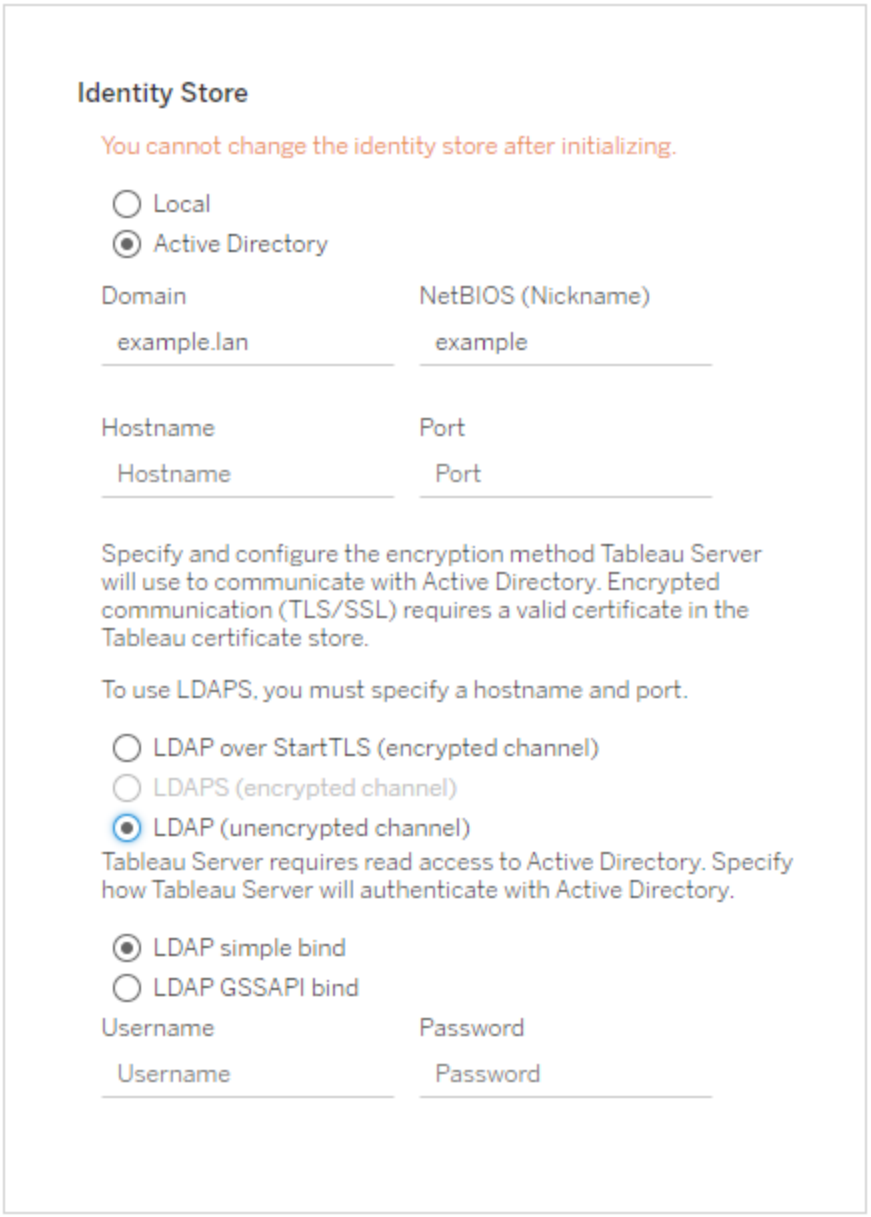
Disable default encrypted LDAP channel

If you are running Tableau Server on Linux and connecting to Active Directory, you can disable the encrypted channel requirement.

When disabled, user credentials that are used to establish the bind session with Active Directory are communicated in plaintext between Tableau Server and the Active Directory server.

Disable new installation

If you will be using Active Directory as your identity store, then you must use the TSM GUI to configure the Active Directory connection. See Configure Initial Node Settings.



Select **LDAP (unencrypted channel)** when running Setup.

Disable before upgrading

If you are upgrading to Tableau Server 2021.2 (or newer) from an earlier version, run the following commands on earlier version of Tableau Server before you upgrade:

```
tsm configuration set -k wgserver.domain.ldap.starttls.enabled -v
false --force-keys
```

```
tsm pending-changes apply
```

To verify that the key has been set, run the following command:

```
tsm configuration get -k wgserver.domain.ldap.starttls.enabled
```

The command should return `false`.

Error messages

The following error messages may be displayed or logged. If you see these errors, do the following:

- Verify that your certificate is valid and imported to the Tableau key store as described earlier in this topic.
- (LDAPS only) - Verify that the host and port name is correct.

In the Setup GUI

The following error will be displayed if you have misconfigured LDAPS or StartTLS when running the Setup or Upgrade GUI.

```
TLS handshake failed. Tableau Server and the Active Directory server
could not negotiate a compatible level of security.
```

Vizportal logs

If you are configuring LDAPS or StartTLS using CLI, the following error message will not be displayed. Rather, the error will be logged in the vizportal logs at `/var/-opt/tableau/tableau_server/data/tabsvc/logs/vizportal`.

```
Authentication with LDAP server failed. The provided credentials or
configuration are either incorrect or do not have the necessary per-
missions to bind.
```

System User, sudo Privileges, and systemd

This topic describes system user, systemd user service, and sudo privilege in the context of Tableau Server.

Privilege separation

Following standard security best practices, Tableau Server for Linux runs processes with the least privilege possible. During installation, the unprivileged user, *tableau*, is created in a server authorized group (*tableau*).

An example user entry in the /etc/passwd file is as follows:

- `tableau:x:993:991:Tableau Server:/var/opt/tableau/tableau_server:/bin/bash`

All processes run as the unprivileged *tableau* user. This means that if one of the Tableau Server processes (such as a process displaying vizzes to users) were compromised in some fashion, it would only be able to impact Tableau Server, not the rest of the Linux system. For this reason, you should not add the `tableau` unprivileged user to the `tsmadmin` group. The `tsmadmin` group should only contain accounts that require authorization to access OS-related Tableau configurations.

The `tableau` user and `tsmadmin` group are created by the Tableau Server initialization process. You can specify a different unprivileged user or TSM authorization group during installation. For more information about system users and groups, in the context of installation and LDAP configuration, see Identity Store.

sudo privileges

The first version (10.5) of Tableau Server on Linux relied on sudo privileges by updating the sudoers file. Updating the sudoers file conflicts with some system management configuration best practices and security policies. Therefore, the 2018.1 version (and later) of Tableau Server no longer creates or uses a privileged user (`tsmagent`). Nor does the current version of Tableau Server update or include a Tableau-specific sudoers file.

All privileged operations now occur during package and software installation.

systemd user service

In the 10.5 version of Tableau Server on Linux, sudo privileges were required to modify or restart the TSM services, which required systemctl commands. All TSM services were run from the normal system-wide systemd process (`process ID 1`, which runs all processes on the operating system). In this scheme, systemd process runs as root. Therefore, the 10.5 version of Tableau Server required sudo privileges.

With the current 2018.1 (and later) releases, we have removed the need for sudo privileges by making use of the systemd capability to run as a user service. The systemd user service runs as a normal user, so it does not need any special privileges once it has been enabled.

In normal use cases, you will not need to issue commands to systemd because TSM takes care of that. However, for troubleshooting scenarios, you may need to interact with the TSM services. As with the previous versions, you will issue the same systemctl commands for these scenarios. However, commands should be run as the `tableau` user, and not as root. If you specified a different unprivileged system user during Tableau Server setup, then run the commands as that user.

Running systemctl commands

Use the following syntax example to issue request to systemd with the systemctl commands.

Start a session as the unprivileged user. The `-l` flag is critical to set environment variables properly.

```
sudo su -l tableau
```

Then issue commands. For example:

```
systemctl --user status tabadmincontroller_0
```

```
systemctl --user restart tabadmincontroller_0
```

Security Hardening Checklist

The following list provides recommendations for improving the security ("hardening") of your Tableau Server installation.

> Looking for Tableau Server on Windows? See Security Hardening Checklist.

Installing security updates

Security updates are included in the latest versions and maintenance releases (MR) of Tableau Server. You cannot install security updates as patches. Rather, you must upgrade to a current version or MR to update Tableau Server with the latest security fixes.

Always reference the most current version of this topic after upgrading. The current version includes `/current/` in the topic URL.

For example, the US version URL is: https://help.tableau.com/current/server/en-us/security_harden.htm.

1. Update to the current version

We recommend that you always run the latest version of Tableau Server. Additionally, Tableau periodically publishes maintenance releases of Tableau Server that include fixes for known security vulnerabilities. (Information regarding known security vulnerabilities can be found on the Tableau Security Bulletins page and the Salesforce Security Advisories page.) We recommend that you review maintenance release notifications to determine whether you should install them.

To get the latest version or maintenance release of Tableau Server, visit the Customer Portal page.

2. Configure SSL/TLS with a valid, trusted certificate

Secure Sockets Layer (SSL/TLS) is essential for helping to protect the security of communications with Tableau Server. Configure Tableau Server with a valid, trusted certificate (not a self-signed certificate) so that Tableau Desktop, mobile devices, and web clients can connect to the server over a secured connection. For more information, see SSL.

3. Disable older versions of TLS

Tableau Server uses TLS to authenticate and encrypt many connections between components and with external clients. External clients, such as browsers, Tableau Desktop, Tableau Mobile connect to Tableau using TLS over HTTPS. Transport layer security (TLS) is an improved version of SSL. In fact, older versions of SSL (SSL v2 and SSL v3) are no longer considered to be adequately secure communication standards. As a result, Tableau Server does not allow external clients to use SSL v2 or SSL v3 protocols to connect.

We recommend that you allow external clients to connect to Tableau Server with TLS v1.3 and TLS v1.2.

TLS v1.2 is still regarded as a secure protocol and many clients (including Tableau Desktop) do not yet support TLS v1.3.

TLS v1.3 capable clients will negotiate TLS v1.3 even if TLS v1.2 is supported by the server.

The following tsm command enables TLS v1.2 and v1.3 (using the "all" parameter) and disables SSL v2, SSL v3, TLS v1, and TLS v1.1 (by prepending the minus [-] character to a given protocol). TLS v1.3 is not yet supported by all components of Tableau Server.

```
tsm configuration set -k ssl.protocols -v 'all -SSLv2 -SSLv3 -TLSv1
-TLSv1.1'

tsm pending-changes apply
```

To modify the protocols that govern SSL for the Tableau Server PostgreSQL repository, see pgsql.ssl.ciphersuite.

You can also modify the default list of cipher suites that Tableau Server uses for SSL/TLS sessions. For more information see the *ssl.ciphersuite* section at tsm configuration set Options.

4. Configure SSL encryption for internal traffic

Configure Tableau Server to use SSL to encrypt all traffic between the Postgres repository and other server components. By default, SSL is disabled for communications between server components and the repository. We recommend enabling internal SSL for all instances of Tableau Server, even single-server installations. Enabling internal SSL is especially important for multi-node deployments. See Configure SSL for Internal Postgres Communication.

5. Enable firewall protection

Tableau Server was designed to operate inside a protected internal network.

> **Important**: Do not run Tableau Server, or any components of Tableau Server on the internet or in a DMZ. Tableau Server must be run within the corporate network protected by an internet firewall. We recommend configuring a reverse proxy solution for internet clients that need to connect to Tableau Server. See Configuring Proxies and Load Balancers for Tableau Server.

A local firewall should be enabled on the operating system to protect Tableau Server in single and multi-node deployments. In a distributed (multi-node) installation of Tableau Server, communication between nodes does not use secure communication. Therefore, you should enable firewalls on the computers that host Tableau Server. See Configure Local Firewall.

To prevent a passive attacker from observing communications between nodes, configure a segregated virtual LAN or other network layer security solution.

See Tableau Services Manager Ports to understand which ports and services Tableau Server requires.

6. Restrict access to the server computer and to important directories

Tableau Server configuration files and log files can contain information that is valuable to an attacker. Therefore, restrict physical access to the machine that is running Tableau Server. In addition, make sure that only authorized and trusted users have access to the Tableau Server files in the `/var/opt/tableau/tableau_server/` directory.

7. Generate fresh secrets and tokens

Any Tableau Server service that communicates with repository or the cache server must first authenticate with a secret token. The secret token is generated during Tableau Server setup. The encryption key that internal SSL uses to encrypt traffic to Postgres repository is also generated at during setup.

We recommend that after you install Tableau Server, you generate new encryption keys for your deployment.

These security assets can be regenerated with the `tsm security regenerate-internal-tokens` command.

Run the following commands:

```
tsm security regenerate-internal-tokens
```

```
tsm pending-changes apply
```

8. Disable services that you're not using

To minimize the attack surface of the Tableau Server, disable any connection points that are not needed.

JMX Service

JMX is disabled by default. If it's enabled but you're not using it, you should disable it by using the following:

```
tsm configuration set -k service.jmx_enabled -v false
```

```
tsm pending-changes apply
```

9. Verify session lifetime configuration

By default, Tableau Server does not have an absolute session timeout. This means that browser-based client (Web authoring) sessions can remain open indefinitely if the Tableau Server inactivity timeout is not exceeded. The default inactivity timeout is 240 minutes.

If your security policy requires it, you can set an absolute session timeout. Be sure to set your absolute session timeout in a range that allows the longest-running extract uploads or workbook publishing operations in your organization. Setting the session timeout too low may result in extract and publishing failures for long-running operations.

To set the session timeout run the following commands:

```
tsm configuration set -k wgserver.session.apply_lifetime_limit -v
true
```

`tsm configuration set -k wgserver.session.lifetime_limit -v` *`value`*, where *value* is the number of minutes. The default is 1440, which is 24 hours.

`tsm configuration set -k wgserver.session.idle_limit -v` *`value`*, where *value* is the number of minutes. The default is 240.

```
tsm pending-changes apply
```

Sessions for connected clients (Tableau Desktop, Tableau Mobile, Tableau Prep Builder, Bridge, and personal access tokens) use OAuth tokens to keep users logged in by re-establishing a session. You can disable this behavior if you want all Tableau client sessions to be solely governed by the browser-based session limits controlled by the commands above. See Disable Automatic Client Authentication.

10. Configure a server allowlist for file-based data sources

As of October 2023 Tableau Server releases, default file-based access behavior has changed. Previously, Tableau Server allowed authorized Tableau Server users to build workbooks that use files on the server as file-based data sources (such as spreadsheets). With the October 2023 releases, access to files stored on Tableau or on remote shares must be specifically configured on Tableau Server using the setting described here.

This setting allows you to limit access by the `tableau` system account only to those directories that you specify.

To configure access to shared files, you must configure allowlist functionality. This lets you limit `tableau` account access to just the directory paths where you host data files.

1. On the computer running Tableau Server, identify the directories where you will host data source files.

   Important Make sure the file paths you specify in this setting exist and are accessible by the system account.

2. Run the following commands:

   `tsm configuration set -k native_api.allowed_paths -v "`*`path`*`"`, where *path* is the directory to add to the allowlist. All subdirectories of the specified path will be added to the allowlist. You must add a trailing backslash to the specified path. If you want to specify multiple paths, separate them with a semicolon, as in this example:

   ```
   tsm configuration set -k native_api.allowed_paths -v "/data-sources;/HR/data/"
   ```

   ```
   tsm pending-changes apply
   ```

11. Enable HTTP Strict Transport Security for web browser clients

HTTP Strict Transport Security (HSTS) is a policy configured on web application services, such as Tableau Server. When a conforming browser encounters a web application running HSTS, then all communications with the service must be over a secured (HTTPS) connection. HSTS is supported by major browsers.

For more information about how HSTS works and the browsers that support it, see The Open Web Application Security Project web page, HTTP Strict Transport Security Cheat Sheet.

To enable HSTS, run the following commands on Tableau Server:

```
tsm configuration set -k gateway.http.hsts -v true
```

By default, HSTS policy is set for one year (31536000 seconds). This time period specifies the amount of time in which the browser will access the server over HTTPS. You should consider setting a short max-age during initial roll-out of HSTS. To change this time period, run `tsm configuration set -k gateway.http.hsts_options -v max-age=<seconds>`. For example, to set HSTS policy time period to 30 days, enter `tsm configuration set -k gateway.http.hsts_options -v max-age=2592000`.

```
tsm pending-changes apply
```

12. Disable Guest access

Core-based licenses of Tableau Server include a Guest user option, which allows any user in your organization to see and interact with Tableau views embedded in web pages.

Guest user access is enabled by default on Tableau Servers deployed with core-based licensing.

Guest access allows users to see embedded views. The Guest user cannot browse the Tableau Server interface or see server interface elements in the view, such as user name, account settings, comments, and so on.

If your organization has deployed Tableau Server with core licensing and Guest access is not required, then disable Guest access.

You can disable Guest access at the server or site level.

You must be a server administrator to disable the Guest account at either the server or the site level.

To disable Guest access at the server level:

1. In the site menu, click **Manage All Sites** and then click **Settings** > **General**.
2. For **Guest Access**, clear the **Enable Guest account** check box.
3. Click **Save**.

To disable Guest access for a site:

1. In the site menu, select a site.
2. Click **Settings**, and on the Settings page, clear the **Enable Guest account** check box.

For more information, see Guest User.

13. Set referrer-policy HTTP header to 'same-origin'

Beginning in 2019.2, Tableau Server includes the ability to configure Referrer-Policy HTTP header behavior. This policy is enabled with a default behavior that will include the origin URL for all "secure as" connections (`no-referrer-when-downgrade`), which sends origin referrer information only to like connections (HTTP to HTTP) or those that are more secure (HTTP to HTTPS).

However, we recommend setting this value to `same-origin`, which only sends referrer information to same-site origins. Requests from outside the site will not receive referrer information.

To update the referrer-policy to `same-origin`, run the following commands:

```
tsm configuration set -k gateway.http.referrer_policy -v same-ori-
gin

tsm pending-changes apply
```

For more information about configuring additional headers to improve security, see HTTP Response Headers.

14. Configure TLS for SMTP connection

Beginning in 2019.4, Tableau Server includes the ability to configure TLS for the SMTP connection. Tableau Server only supports STARTTLS (Opportunistic or Explicit TLS).

Tableau Server can be optionally configured to connect to a mail server. After configuring SMTP, Tableau Server can be configured to email server administrators about system failures, and email server users about subscribed views and data-driven alerts.

To configure TLS for SMTP:

1. Upload a compatible certificate to Tableau Server. See tsm security custom-cert add.
2. Configure TLS connection using TSM CLI.

   Run the following TSM commands to enable and force TLS connections to the SMTP server and to enable certificate verification.

   ```
   tsm configuration set -k svcmonitor.notification.smtp.ssl_
   enabled -v true

   tsm configuration set -k svcmonitor.notification.smtp.ssl_
   required -v true

   tsm configuration set -k svcmonitor.notification.smtp.ssl_
   check_server_identity -v true
   ```

   By default, Tableau Server will support TLS versions 1, 1.1, and 1.2, but we recommend that you specify the highest TLS version that the SMTP server supports.

Run the following command to set the version. Valid values are `SSLv2Hello`, `SSLv3`, `TLSv1`, `TLSv1.1`, and `TLSv1.2`. The following example sets the TLS version to version 1.2.:

```
tsm configuration set -k svcmonitor.notification.smtp.ssl_versions -v "TLSv1.2"
```

For more information about other TLS configuration options, see Configure SMTP Setup.

3. Restart Tableau Server to apply changes. Run the following command:

```
tsm pending-changes apply
```

15. Configure SSL for LDAP

If your Tableau Server deployment is configured to use a generic LDAP external identity store, we recommend configuring SSL to protect authentication between Tableau Server and your LDAP server. See Configure Encrypted Channel to LDAP External Identity Store.

If your Tableau Server deployment is configured to use Active Directory, we recommend enabling Kerberos to protect authentication traffic. See Kerberos.

Change List

Date	Change
May 2018	Added clarification: Do not disable REST API in organizations that are running Tableau Prep.
May 2019	Added recommendation for referrer-policy HTTP header.
June 2019	Removed recommendation to disable Triple-DES. As of version 2019.3, Triple-DES is no longer a default supported cipher for SSL. See What's Changed - Things to Know Before You Upgrade.

January 2020	Added recommendation to configure TLS for SMTP.
February 2020	Added recommendation to configure SSL for LDAP server.
May 2020	Added TLS v1.3 to the disabled list of TLS ciphers. Added clarification to introduction about topic versioning.
October 2020	Added TLS v1.3 as a default supported cipher.
January 2021	Added clarification: All products enabled by the Data Management license require REST API.
February 2021	Removed recommendation to disable REST API. The API is now used internally by Tableau Server and disabling it may limit functionality.

Manage Licenses

You can manage your Tableau Server licenses and view license usage.

Licensing Overview

An important administrative role in a Tableau Server deployment is the Tableau portal administrator. The portal administrator manages licensing and the associated keys for the Tableau deployment. As the portal administrator, your first step is to purchase licenses on the Tableau Customer Portal. When you purchase licenses, the portal will return corresponding product keys. To renew your license, visit the Tableau renewal web page.

Tableau has a number of products (Desktop, Server, Prep Builder, and more). Each of the Tableau products require that you activate licenses by updating the Tableau software with the product keys that are purchased and stored on the Tableau Customer Portal. As the administrator who is tasked with activating Tableau licenses, it important that you understand the relationship between licenses and keys. See Understanding License Models and Product Keys.

Activation

Activation is the process of uploading and saving Tableau product keys to Tableau Server. This operation is done with Tableau Services Manager (TSM). TSM is a tool that makes changes to the local operating system and file system and therefore requires administrative access to the local computer. A TSM administrator requires different permissions and access than a Tableau Server administrator, which is the administrative role for day-to-day operation of Tableau Server tasks, such as adding users, sites, managing projects and permissions, etc. See Administrative roles for more information about various Tableau Server administrative roles.

The following topics describe how to connect to TSM:

- Sign in to Tableau Services Manager Web UI
- tsm Command Line Reference

Online activation

If your Tableau Server installation is able to communicate with the internet, then we recommend using the default online activation method.

- To understand how to activate during the installation process, see Activate and Register Tableau Server.
- To understand how to activate product keys after you have refreshed your subscription, see Refresh Expiration Date and Attributes for the Product Key.
- To understand how to activate product keys after you have added purchased new features or user licenses, see Add Capacity to Tableau Server.

Offline activation

If Tableau Server is running in an offline environment, where it is not possible to access the Tableau license servers on the internet, then you must activate licenses according to the Tableau offline activation process:

- To understand how to activate offline, see Activate Tableau Server Offline.
- To understand how to deactivate a product key that you activated using offline activation, see Deactivate Tableau Server Offline.

Lost activation

In some cases license activations can fail after the license has been activated. These failures can occur due to connection failures with local processes or when a change has occurred with the VM or hardware configuration. For example, proxy changes, port blocking, network changes, or altering a machine hardware can cause the licensing activation to fail. If Tableau Server is unable to verify the license, operation may be interrupted and the server will be in an “unlicensed” state.

To view the product keys and the Tableau Server license state, run `tsm licenses list` and `tsm status -v`.

Depending on the product key that is unverified, Tableau Server may operate in a degraded state until the product key is in a valid state. See Troubleshoot Licensing.

Deactivate

You can activate the same Tableau Server product key on up to three environments. This allows you to test Tableau Server (in a sandbox or QA environments, for example), as well as use Tableau in production. To maximize your activations, you should deactivate your product key when you remove Tableau Server from a computer or close down a VM, unless you will be reinstalling Tableau on the same computer. Doing this gives you the opportunity to use the activation on a different computer. For example, if you move Tableau Server from one computer to another, deactivate the product key, then remove Tableau from the original computer. When you install Tableau on the new computer, you can activate the key there without any conflict. If you are removing Tableau Server to reinstall it on the same computer, you don't need to deactivate the key. Tableau will use the key when reinstalled unless an obliterate command was performed with the “-l” option

See Deactivate Product Key.

Tableau Server licensing and virtual machines (VMs)

If you run Tableau Server on VMs, either locally, or in the cloud, be aware of the potential for complications related to licensing. If you are simply upgrading Tableau Server on the VM, you do not need to take any extra action related to licensing. If you plan to clone the VM to create

either a new production or test environment to upgrade, you need to deactivate any Tableau Server licenses before cloning. If you do not do this, the new VM environment will end up with untrusted licenses, and any attempts to upgrade or start Tableau Server will fail. You may also end up hitting the maximum number of activations for the licenses when trying to activate the product keys on the new VM.

To avoid issues with licensing on VMs, deactivate all Tableau licenses before cloning a VM or allowing it to be permanently shut down.

Login-based License Management

Login-based license management, helps you manage licensing for users with Creator roles on Tableau Server and Tableau Cloud. Users with Explorer or Viewer roles cannot use this feature. If you're using Role Based Subscriptions with Tableau Server or Tableau Cloud, you can simplify your license management using login-based license management to eliminate separate Tableau Desktop and Tableau Prep Builder product keys. You only need to manage one or more product keys for on-premises Tableau Server, or in the case of Tableau Cloud, you don't need to manage any product keys at all.

See Login-based License Management.

Adding users

Each user who accesses resources on Tableau Server must be licensed.

- To understand user roles and licensing, see Understanding License Models and Product Keys.
- To understand how to add users, see Add Users to Tableau Server.
- To understand how to activate product keys after you have added purchased new user licenses, see Add Capacity to Tableau Server.

Understanding License Models and Product Keys

This topic describes the different licensing models and the product keys or subscriptions that may be associated with them. A useful visual of how product keys are represented in Tableau

Server can be found in tsm licenses list. The Tableau Customer Portal will also display product key information including type and seat count.

When viewing product keys using tsm licenses list (Tableau Server), or in the Tableau Customer Portal (Tableau Desktop and Tableau Prep Builder), note the product specific prefixes.

Product Key Prefix	Description
TC	Tableau Creator product key, can be used to activate or deactivate Tableau Desktop and Tableau Prep Builder.
TD	Tableau Desktop product key, can be used to activate or deactivate Tableau Desktop only. This is a legacy product key that is no longer sold or provided.
TS	Tableau Server product key, can be used to activate or deactivate Tableau Server. Tableau Server product keys can be role-based, core-based, or feature-based.

Term licensing models

Tableau's term license model is defined by the metric that permits use of Tableau Server. Term licenses are also called subscription licenses. Tableau currently sells access to Tableau Server with subscription licenses. In the subscription license model, customers pay a yearly subscription fee. If the subscription expires, the software will stop working.

Subscription licenses are either role-based or core-based subscriptions. A single license key can be purchased with all roles and features and this license is called an Updatable Subscription License (USL). Only one key needs to be activated on Tableau Server to represent the entire purchase.

Previous subscription licenses (non-USL) were provided with one role type per key and the licenses were "stacked" and activated together to obtain the purchased configuration on Tableau Server.

- A *role-based license* allows you to deploy Tableau Server on a single computer or on multiple computers in a cluster. Each user that accesses Tableau Server must be licensed and assigned a role. Administrators can add users based on available licenses of each type.

- A *core-based license* imposes no constraints on the number of user accounts in Tableau Server. Instead, the license specifies the maximum number of computer cores on which you can run Tableau Server. You can install Tableau Server on a single computer or across multiple computers as a multi-node cluster, as long as the total number of cores in all the computers does not exceed the total number that the license allows.

  Not all processes installed with Tableau Server impact the calculation of total number of cores used. A subset of processes is considered "licensed processes." Core licensing is calculated only on computers running licensed processes. If a computer has one or more licensed processes installed on it, the cores on that computer count toward the total cores used. For more information about licensed processes, see Licensed processes.

- An *updatable subscription license*, enables you to consolidate licenses and update your Tableau Servers using a single product key. You can add new features, adjust capacity, and apply license renewals to a single existing Tableau Server license. You no longer need to add new licenses or replace existing ones. USL:

  - Simplifies key management by reducing the number of product keys that you must manage for ease of maintenance.

  - Minimizes service interruptions because you don’t have to restart Tableau Server after renewing a license, adjusting capacity, or adding new features.

For Updatable Subscription Licenses (USL), your product key in the Tableau Customer Portal does not change even when updated with new features or changes to role counts. With the non-USL subscription license model, you get a new key in your Customer Portal after each license renewal. A new product key will appear in the TSM web UI after the previous product key expires when viewing a non-USL subscription product key. For Updatable Subscription

Licenses (USL), your product key does not change in the UI or Tableau Customer Portal. In the Tableau Customer Portal, USL licenses display a selected **Is USL Key** checkbox on the **License Detail** page or by **true** in the **Is USL Key** column on the **Licenses** tab.

In the output returned by tsm licenses list, the TYPE field describes the user license metric. In the Tableau Services Manager Web UI, hover over the Type field (or column) to verify if the key is an Updatable Subscription License (USL) product key.

Role-based license model

Tableau offers role-based term licenses that grant a range of capabilities at various price points. Four types of role-based term licenses are available: Display, Viewer, Explorer, and Creator.

- *Display licenses* let users share and display Tableau content with a broad, internal audience of users who consume dashboards via shared displays with no interaction. There is no separate site role for Display licenses; when using a Display license, administrators create a dedicated login account for each licensed Display location, which is not the same as an individual user's login account, and assign the maximum site role of Viewer.
- *Viewer licenses* let users view and interact with workbooks in Tableau Server. Viewer licenses also let users access Tableau Mobile, add comments to workbooks, export visuals in various formats, download workbook summary data, create subscriptions for themselves, and receive data-driven alerts.
- *Explorer licenses* are similar to the user-based licenses available in previous Tableau Server releases, and include the capabilities provided with Viewer licenses, and additional capabilities. An Explorer license allows access to workbook authoring capabilities using a web browser, as well as a full set of collaboration features.
- *Creator licenses* permit a wide range of capabilities when using Tableau Server, and also grant use of Tableau Desktop and Tableau Prep Builder. A Creator license allows all of the capabilities available under the Explorer license, as well as the following capabilities when using Tableau Server:
    - Create and publish new workbooks from a new data source.
    - Edit embedded data sources in the Data pane.
    - Create and publish new data connections.

- Use login-based license management activation on Tableau Desktop and Tableau Prep Builder.

> **Note**: Tableau Server Administrators will always consume the highest role available. If you activate a product key that contains the Creator role, the Tableau Server Administrator(s) will take this role. If the highest role available on Tableau Server is an Explorer, the Server Administrator will take the Explorer role. If a Creator role is added to a server where no Creators are currently activated, any existing Server Administrator accounts using Explorer licenses will automatically convert to use Creator licenses.
>
> TSM administrator accounts do not require licenses.

For non-USL licenses, product keys are used to activate and add licenses to Tableau Server. When an update to capacity or functionality is purchased with a non-USL product key, the Tableau Server Administrator can activate these additional product keys. For USL product key(s), one product key is activated that contains all roles and features. If a USL license has been updated to change functionality, features, or role counts, no additional product keys will be issues or need to be activated.

For Updatable Subscription License (USL) product key(s), when a key has been updated with a new subscription term, feature, or role count change, the Server ATR service automatically obtains these updates and no additional key(s) need to be activated. USL licenses and Server ATR activations do not require refreshes to obtain these changes. Everything will be handled by the Server ATR service. For a USL product key using the non-Server ATR activation method, refreshes need to occur to obtain the update to the USL product key. For non-USL product keys, when an update to capacity or functionality is purchased, additional product keys are provided in the Tableau Customer Portal that you need to activate on Tableau Server.

If you're using non-USL product keys, select a Tableau Server Creator product key from the Tableau Customer Portal to ensure that you can create a Tableau Server Administrator. Explorer and Viewer product keys can then be activated to add additional licenses. Once the

product keys have been activated, administrators can add users and assign them site roles, which automatically consume available licenses.

In the output returned by tsm licenses list, the CREATOR, EXPLORER, VIEWER fields display the number of licenses for each role license type. For non-USL product keys, each role is governed by it's own product key. Therefore, if your organization has purchased licenses for all three roles, then you must activate three product keys. For USL product key(s), one key has all roles and features which is reflected in TSM.

Core-based license model

In a core licensing model, the license defines the number of total computer cores the server can run on, instead of how many users can be added. This means the server can support virtually unlimited users (as Explorers or Viewers; the legacy term for these types of roles was "interactors"). Core-based license also allows a Guest User account, which is not possible with role-based licensing.

However, core licensing on its own does not include Creator seats on the server (those site roles will be grayed out when adding users).

New data sources can only be added to the server by publishing from Tableau Desktop or via a browser by a user with a Creator site role on the server. Therefore, there must either be (a) a user with a Creator site role on the server, or (b) a licensed Tableau Desktop and a user with Explorer (can publish) site role. For route (b), there are two ways to license Tableau Desktop in a core-based server organization:

- At least one role-based Creator license (which includes Tableau Desktop and Tableau Prep Builder), or
- At least one licensed Tableau Desktop that uses a legacy Tableau Desktop product key.

Note that if a Creator license is stacked on a server with core-based licensing, that role-based Creator license will be consumed by an admin. The only way to give a Creator site role to a non-admin user is to first ensure that all admins have Creator licenses. Only then can additional Creator licenses be used to assign a Creator site role to non-admin users.

In the output returned by tsm licenses list, the TYPE field will display the number of cores that are licensed. Additionally, the GUEST ACCESS field will display `true`.

Embedded Analytics usage-based model

Tableau's Embedded Analytics offering is a usage-based license for Tableau Server that's made available to customers who want to embed Tableau Server functionality into an external facing solution to provide Tableau content and insights to clients outside of their organization.

> **Note:** Embedded Analytics licenses cannot be used in the same environment as full-use licenses. To change to an Embedded Analytics license, first deactivate your existing full-use licenses and then activate the Embedded Analytics license.

For example, consider an organization that runs a service where they analyze consumer data and generate reports on behavioral patterns regarding different consumer demographics. In this scenario, Tableau Server acts in support of a specific proprietary application titled 'Demographics Analyzer' and connects with exported TXT files and a SQL database. The organization makes visualizations available to its clients in a secure portal, where clients log in to manage their account and view the results. End users are uniquely identified by the account they use to access the portal; this determines the number of User-Based Embedded Analytics Server licenses.

The Embedded Analytics license is not displayed in the output returned by tsm licenses list. To verify the license contact Customer Success.

Perpetual license model (legacy)

In the past, Tableau sold access to Tableau Server with perpetual licenses. Although these licenses are no longer available, some customers use this licensing arrangement.

In the perpetual license model, customers paid a maintenance subscription that was renewed annually. If maintenance expires, the software continues to work, but the customer loses

access to technical support and software upgrades. Purchasing annual maintenance for perpetual licenses is no longer possible.

Perpetual licenses were either sold for a specific number of users, called interactors, or for a specific number of cores:

- Interactor licensing is a named-user model where customers purchased licenses by the seat, similar to current role-based licensing. However, unlike role-based licensing, where different access roles are priced accordingly, in interactor licensing, licenses were unbound by role. Licensed users could be Server admins, Site admins, Publishers, Interactors, or Viewers. User roles were set by the administrator only as a means to manage access to content and server configuration.

  If you activate a valid Server Interactor key on version 2020.4 or later, the Interactor count is mapped to the Explorer role.

  Interactor perpetual core licenses cannot use login-based license management. You must purchase a Creator term license, also called a Creator subscription license, in order to use login-based license management with Tableau Desktop or Tableau Prep Builder.

- Perpetual core licensing has the same model as subscription core licensing, it specifies the number of computer cores the software can be run on and supports unlimited users and a guest account.

In the output returned by tsm licenses list, the TYPE filed displays `Perpetual`. Note also that the MAINT EXP date is also displayed.

License editions

License editions include a suite of features and functionality to which users are entitled. License editions cannot be mixed within a deployment, meaning all users on a deployment must be on the same license edition. Consider the needs of your entire deployment when selecting your license edition.

Tableau license edition

Tableau license edition is the standard Tableau Server edition. It provides Tableau Server access for each licensed user, governance, collaboration, data prep, and visual analytics functionality.

Tableau Enterprise license edition

Enterprise edition is designed for sophisticated business environments on Tableau Server. It is ideal for organizations that require advanced administration, security, and data management functionality to scale to more users in more complex data environments and meet Enterprise standards.

For more information and a list of features included with Tableau Enterprise, see About Tableau Enterprise.

Note: The Tableau+ license edition is available exclusively for Tableau Cloud. For more information and a list of features included with Tableau+, see About Tableau+ in the Tableau Cloud Help.

Feature licenses

Important: As of September 16, 2024, Advanced Management and Data Management are no longer available as independent add-on options. Advanced Management and Data Management capabilities are only available if you previously purchased these, or if you purchase certain license editions - either Tableau Enterprise (for Tableau Server or Tableau Cloud) or Tableau+ (for Tableau Cloud).

Feature licenses are sold differently than other licenses. Features with independent licenses must be licensed for every user (or all cores) in the deployment. A deployment includes a licensed production Tableau Server installation and licensed non-production Tableau Server installations that support the production installation.

These features are licensed annually, and in the context of licensing, the availability of these features to the user base are “all or none:”

- Data Management
- Advanced Management
- Login-based License Management

Note: Updatable subscription licenses include both features and roles in one license. You no longer need to activate multiple licenses and product key(s) for different features and their associated roles.

Data Management

The Data Management license includes Tableau Catalog and Tableau Prep Conductor for a single Tableau Server deployment, which may be role-based or core-based. For more information, see About Data Management.

Data Management may require resource cores, which specifies the computing power that is used to run flows for Prep Conductor. Servers with core-based licensing are required to purchase at least four Resource Cores. See License Data Management.

In the output returned by tsm licenses list, a single product key for Data Management is indicated by the DATA MANAGEMENT field, which displays `true` under the appropriate license.

Advanced Management

Tableau Server Advanced Management is licensed on a per deployment basis, which may be role-based or core-based. For more information on Advanced Management and the capabilities included, see About Tableau Advanced Management on Tableau Server.

In the output returned by tsm licenses list, a single product key for Advanced Management is indicated by the SERVER MANAGEMENT ADD-ON field, which displays `true` under the appropriate license.

Login-based License Management

Login-based license management (LBLM) simplifies licensing for Tableau Desktop and Tableau Prep Builder. Instead of using product keys, these products are activated when a user with a Creator's license authenticates with Tableau Server. If a customer is using Tableau Desktop or Tableau Prep Builder without Tableau Server, LBLM cannot be used.

Note: As of Tableau Server version 2023.1, LBLM is supported in offline deployments. To deploy Tableau Server with LBLM enabled, you must configure Tableau Server to use Authorization-To-Run (ATR) Service. For more information, see Activate Tableau Server Offline.

For more information, see Login-based License Management.

In the output returned by tsm licenses list, a single product key for login-based licensed management is indicated by the LBLM field, which displays `true` under the appropriate license.

Updatable Subscription Licensing (USL)

Updatable Subscription Licensing (USL) is the latest improvement to Tableau's license activation and server administration experience.

Understanding the Basics of USL

Updatable Subscription Licensing simplifies the requirements of license management for Tableau Server by eliminating the need to manage multiple separate Product Keys for various product types, capacities or subscription periods. Instead, all licensed features and attributes of your Tableau Deployment (its "License Entitlement") are provisioned as a single unified Product Key.

Additionally, USL consolidates the delivery of subsequent license renewals, add-ons, and entitlement changes into updates which are made to that single, updateable Product Key over time, so it consistently reflects the complete License Entitlement for its designated Deployment. You can think of each update like adding a layer of new attributes to your existing

Product Key; these layered updates each have a unique corresponding Activation ID. When you sync your Server installation with Tableau's licensing service (by activating or refreshing your Product Key), the latest Activation ID is pulled down to your local Server installation in order to reflect your Deployment's updated License Entitlement.

The process required to pull down your Product Key's latest Activation ID differs depending on your Deployment's internet connectivity.

Activating USL in Online/Connected Environments

In online/connected environments, the USL Product Key used to activate and unlock the License Entitlement for your Deployment remains constant over time.

- By simply activating or refreshing the original USL Product Key (reflected as the "Key Name" in the Tableau Customer Portal), a connected environment will automatically pull down the latest current Activation ID available for your License Entitlement, facilitated by the live communication with Tableau's hosted licensing services.
- For Server installations configured to use Tableau's Authorization to Run (ATR) licensing service, these updates are downloaded silently in the background during periodic Server check-ins with the ATR service; No additional action needed!

Activating USL in Offline or Disconnected Environments

In offline or disconnected environments, there is no direct communication between Server and Tableau's hosted licensing services, so the USL Product Key cannot be used to automatically retrieve new Activation ID updates.

- Instead, administrators must directly activate the latest Activation ID through Tableau's Offline Activation process to reflect updates to the Deployment's License Entitlement.
- Once a new Activation ID is added to your License Entitlement, no prior Activation ID can be used for activation; they become functionally obsolete.

- For this purpose, the latest current Activation ID is always reflected in the Tableau Customer Portal as the “Offline Activation ID” visible in the License Detail view for your USL Product Key.
- Because new Activation IDs are generated for any update to your License Entitlement, the Offline Activation ID will change repeatedly over time.

> **Important:** Offline activation of Tableau requires that customers use the most current Offline Activation ID to activate any new installation of the Software.

Managing license entitlement updates in offline environments

Given the requirement to leverage the most-current Offline Activation ID when activating your USL Product Key in a disconnected environment, it’s important to get your current Offline Activation ID from the Tableau Customer Portal before attempting to activate any new or existing installation of Tableau Server. This may not match the one you last used.

Salesforce’s provisioning systems periodically synchronize to ensure complete and accurate fulfillment, and this can generate a new Offline Activation ID for your License Entitlement unrelated to any specific purchase or subscription renewal. Be sure to check the Customer Portal *before* any change to the active licensing in your Deployment, including prior to deploying any new installations of Server Software, to confirm the current Offline Activation ID and ensure successful activation.

The following are some key points related to the Offline Activation ID life cycle:

- Successful activation *will always require*
  a. Deactivation of the earlier Offline Activation ID, and
  b. Activation of the current Offline Activation ID.

  For Server installations configured to use ATR (offline), deactivation of the earlier Offline Activation ID is handled automatically.

- The issuance of a new Offline Activation ID *will prevent new* activations using any prior Offline Activation ID.

  For this reason, be sure to consult the Customer Portal before any licensing update in your deployment.

- The issuance of a new Offline Activation ID *will not impact* any Server environment where you have previously activated your USL Product Key.

  - These installations won't reflect your updated License Entitlement until the new Offline Activation ID is directly applied.

USL Offline Activation Instructions

To activate USL in an offline or disconnected environment:

1. Identify the current Offline Activation ID for your Tableau Server Deployment.

   You can find the Offline Activation ID in the Tableau Customer Portal by navigating to the **Licenses** pane and clicking on your USL Product Key to open the License Details view. The 20-digit string is labeled "Offline Activation ID". For further detail, see the Knowledge Base article Find the Correct Key to Offline Activate on Tableau Server

2. Activate the Offline Activation ID on Tableau Server.

   With the Offline Activation ID accessible, follow the steps outlined in the Tableau Help to activate Tableau Server offline (applicable to both USL and pre-USL Product Keys). Activate Tableau Server Offline

USL offline license entitlement updates

The steps required for changes in your license entitlement in an offline environment depend on your method of activation (ATR or legacy FNO) and your Tableau Server version.

- ATR *and* version 2023.1.0 or higher:

  If you activated with ATR and have version 2023.1+ use the new Offline Activation ID from your Customer Portal to update your license entitlement changes. The ATR service handles any other necessary actions.

  > **Note:** After activating the Offline Activation ID, you will see the Product Key listed in the Tableau Customer Portal and not the Offline Activation ID which was activated. This will allow for ease of management between any ongoing Offline Activation ID changes and product key management.

- Legacy FNO activation *or* any version before 2023.1.0:

  1. Obtain Current Offline Activation ID from the Customer Portal
  2. Stop Tableau Server:

     ```
     tsm stop
     ```

  3. Initiate an offline deactivation of the existing, activated key on Server. This Offline Activation ID is now obsolete in your environment. The offline deactivation creates a return request file. Save this file. You need it for a future step.
  4. Initiate an offline activation request for the current Offline Activation ID. This generates an activation request. Save this file for the next step.
  5. Transfer both the return request file and the activation request files to a computer that has internet access.
  6. Navigate to Offline Activation page and initiate two requests in this order:
     a. Upload the return request file and download the response file; save and continue.

b. Upload the activation request file and download the response file; save and continue.

7. Transfer both return response file and the activation response file to the disconnected Tableau Server computer.
8. Apply the return response file.
9. Apply the activate response file.
10. Start Tableau Server:

```
tsm start
```

View Server Licenses

Server administrators can view the license and product key information for Tableau Server.

Viewing licenses from the Tableau Server web UI

How you navigate to the Licenses page in Tableau Server depends on whether you have a single site, or multiple sites.

- On a server with a single site, click **Settings** and **Licenses**:
- On a multi-site server, click **Manage all sites** on the site menu, **Settings**, and **Licenses**:

> **Note:** The **Manage all sites** option only displays when you are signed in as a server administrator.

This page displays information for any licenses that have been activated on your server, including any user-based (term) or core-based licenses.

Use the TSM web interface

1. Open TSM in a browser:

   http://<tsm-computer-name>:8850

2. Click **Configuration** , and then click **Licensing** :

   The table displays the product key, expiration date, and expiration of maintenance.

   > **Note**: The TSM Web UI provides a limited amount of licensing information. Use the TSM CLI or the Tableau Server Web UI to see additional licensing information, including the number of each type of user-based license (Creator, Explorer and Viewer).

Use the TSM CLI

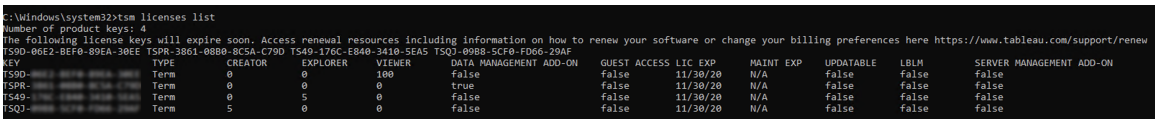
1. Open a command prompt as administrator on the initial node (the node where TSM is installed).

2. Run the following command:

   ```
   tsm licenses list
   ```

The command lists licenses that are activated on the Tableau Server deployment.

For example, a server with five Creator licenses, five Explorer licenses, 100 Viewer licenses, and Data Management would provide command output similar to the following:



The following fields are returned:

- **KEY**: A globally unique 16-character string that identifies the license.
- **TYPE**: Describes the type of license
  - Term: Term licenses map to a subscription schedule and must be renewed. The expiration date is listed under the LIC EXP field.
  - Perpetual: Perpetual licenses are purchased once and do not need to be renewed but must be refreshed to update the MAINT EXP or maintenance expiration date.
  - Cores: Core licenses are licenses that map to the number of cores on the computers running specific Tableau Server services. Core licensing allows for a guest user access to views on the server or embedded on other web servers. Core licenses also allow for unlimited Explorer and Viewer users.
- **CREATOR**: The number of Creator licenses issued to the Tableau Server deployment.
- **EXPLORER**: The number of Explorer licenses issued to the Tableau Server deployment.
- **VIEWER**: The number of Viewer licenses issued to the Tableau Server deployment.
- **DATA MANAGEMENT**:Tableau Server is licensed with Data Management (`True`/`False`). See About Data Management.
- **GUEST ACCESS**: Tableau Server is licensed for a Guest User. See Guest User. The ability to leverage a Guest User requires Core licensing. See TYPE field.
- **LIC EXP**: The date that the license expires and Tableau Server will stop working. Term licenses expire. See TYPE field. Visit the Tableau Customer Portal to refresh licenses.
- **MAINT EXP**: Applies only to legacy perpetual licenses (TYPE = Perpetual). For Term licenses, this field will output, `N/A`. MAINT EXP displays the date that the maintenance contract for the Tableau Server deployment expires. To update the license maintenance key see Refresh Expiration Date and Attributes for the Product Key. Visit the Tableau Customer Portal to view maintenance purchase history and to purchase additional maintenance.
- **UPDATABLE**: Specifies whether the license is an updatable subscription license (`True`/`False`).
- **LBLM**: Specifies if login-based license management (LBLM) is enabled for the Tableau Server deployment (`True`/`False`). When enabled, LBLM allows users to log into Tableau Server to license their instance of Tableau Desktop or Prep, rather than entering a product key. For more information about LBLM, see Login-based License Management.
- **SERVER MANAGEMENT**: Tableau Server is licensed for Advanced Management (formerly Server Management Add-on) (`True`/`False`). For more information about

Advanced Management, see About Tableau Advanced Management on Tableau Server.

Note: The license terms for Creator, Explorer and Viewer users are set according to the terms of the user-based license (term license), if present. So, a server with only a core-based license will have unlimited Explorer and Viewer users and guest access, but no Creator users. To learn more, see Use role-based licenses on a server with core-based licensing.

Refresh Expiration Date and Attributes for the Product Key

When using Server ATR and Updatable Subscription Licensing (USL), you don't need to refresh your product key(s) when you purchase a new subscription term, or add roles and/or features to your deployment. That's because USL product keys can be updated to reflect changes to your Tableau Server license capacity, feature, and subscription term over time, and Server ATR automatically refreshes product key(s) for you as a background process.

Note: If you were recently enabled for USL alongside your subscription renewal, you cannot refresh your licensing as outlined here. Instead, refer to the Tableau Customer Portal for the new USL-compatible product key which must be activated within your deployment; see Activate and Register Tableau Server. Your original product keys have not been renewed, and should no longer be used in your deployment.

If you aren't using Server ATR, you can refresh your product key(s) manually. If you refresh a subscription (term) product key before the expiration date occurs, the product key will not change but the expiration date will. This can create a mismatch between the product keys listed in the Tableau Customer Portal and those listed in the Tableau Server TSM product key list. The product key will change when the expiration date occurs or shortly thereafter. If a subscription (term) product key is not refreshed and has expired, Tableau will stop working and you will have to activate a new product key from the Tableau Customer Portal to relicense your

Tableau Server installation, regardless of whether you are using Server ATR or manual activation methods.

On the other hand, if the product key is perpetual (legacy) and its maintenance has expired, Tableau will continue to operate but you will not have access to upgrades until the maintenance is renewed. After renewing the maintenance, if you aren't using Server ATR, you must refresh the existing product key to update its maintenance expiration date. The product key will never change. For more information about different product key types and associated licenses, see tsm licenses list.

Note: This topic describes how to refresh the expiration date for Tableau Server, Advanced Management, and Data Management. For information about refreshing the maintenance date on Tableau Desktop see Refresh the product key in the Tableau Desktop and Tableau Prep Deployment Guide.

Before you begin

Verify the expiration date of your license(s). You can view the expiration date by following the TSM web interface procedure below, or by running tsm licenses list in the CLI.

- Compare the date with the date displayed in the Tableau Customer Portal.
- If the portal does not display the date that you expect, contact Customer Success.
- To renew your license, visit the Tableau renewal web page.

If the TSM date does not match the date displayed in the Tableau Customer Portal and the following refresh operation fails, contact Tableau Support.

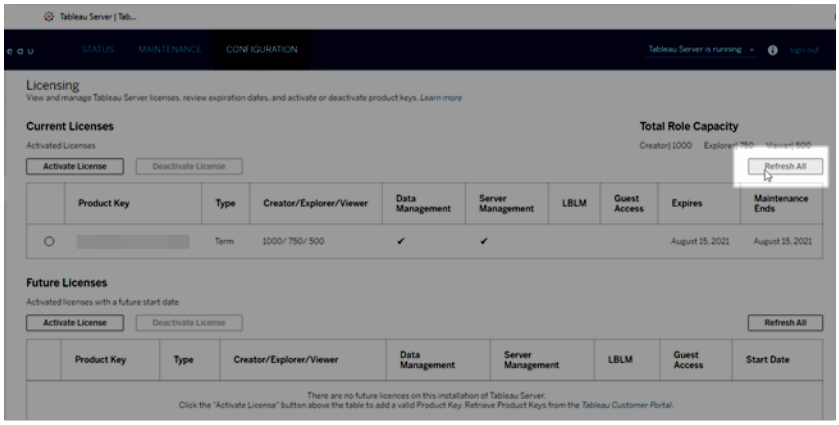
Note: To refresh your product key in an offline environment, visit the Tableau Customer Portal to obtain the Offline Activation ID for your latest renewal, and then activate it. For more information about offline activation, see Activate Tableau Server Offline. Activate all product keys before starting Tableau Server. Otherwise some users might become unlicensed.

Use the TSM web interface

1. Open TSM in a browser:

   ```
   https://<tsm-computer-name>:8850
   ```

2. Click **Configuration** and **Licensing** and click **Refresh All**:



Use the TSM CLI

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).

2. Run the following command:

   ```
   tsm licenses refresh
   ```

Add Capacity to Tableau Server

You may need to add capacity to your Tableau Server installation to allow you to increase the number of users (if you have a user-based license) or the number of cores (if you have a core-based license).

Tableau will provide you with a new product key that adds capacity to your existing Tableau Server installation. You need to activate this key and use it together with your existing product key(s) to get the combined capacity you are licensed for. Each feature (e.g., "Data

Management") and license type (e.g., "Explorer") requires a key. For more information about the relationship between keys and licenses, see tsm licenses list.

Follow the steps below to add a product key to Tableau Server.

If your Tableau Server is not connected to the internet, then you must perform an offline activation. See Activate Tableau Server Offline.

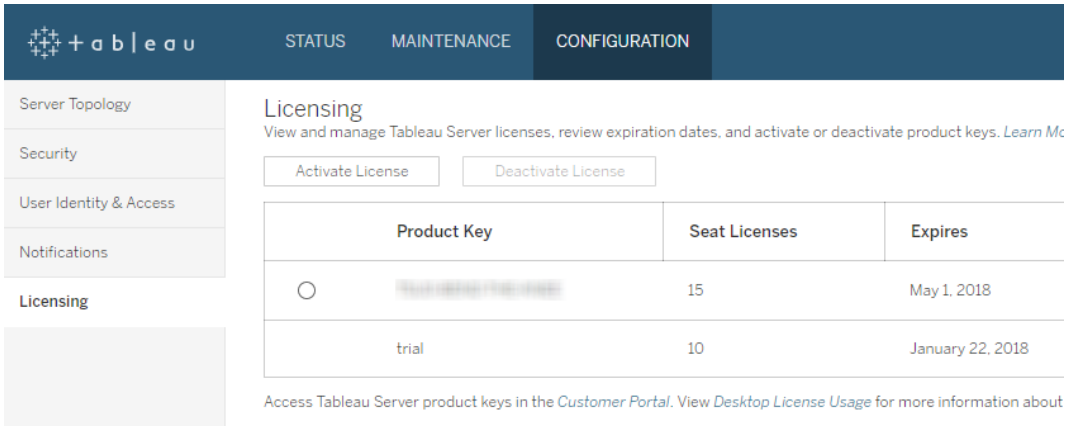
Note: If you have upgraded to Tableau Server version 2021.1 or later, you no longer need to restart Tableau Server when you add capacity. For more information, see Zero Downtime Licensing.

Use the TSM web interface

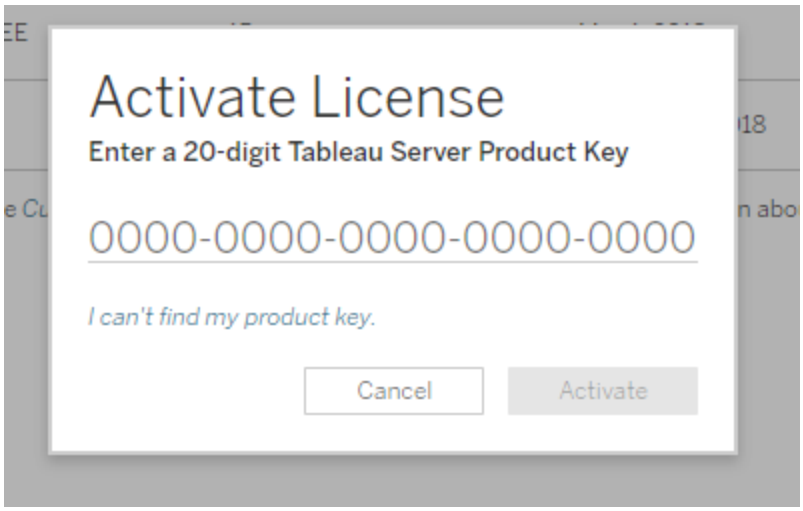
1. Open TSM in a browser:

   ```
   https://<tsm-computer-name>:8850
   ```

2. Click **Licensing** on the **Configuration** tab and click **Activate License**:



3. Enter or paste your new product key and click **Activate**:



4. After activating the license, you may be prompted to register with Tableau. On the Register page, enter your information into the fields and click **Register**.

Register with Tableau. All fields are required.

Contact Information

First Name

Last Name

Phone Number

Email

Company Information

Organization

Industry

Company Size

Department

Job Role

Region Information

City

Postal Code

Country/Region

State/Province

Use the TSM CLI

1. Copy the product key to your computer.
2. Run the following command:

   ```
   tsm licenses activate --license-key <license key>
   ```

3. After activating the license, you may need to register Tableau Server. To do this, create a registration file and then pass it as an option with the `tsm register` command.

   a. Generate a template that you can edit by running the following command:

   ```
   tsm register --template > /path/to/<registration_file>.json
   ```

   b. Edit the template file to create your completed registration file.

   Here is an example of a registration file in the required format:

   ```
   {
    "zip" : "97403",
    "country" : "USA",
    "city" : "Springfield",
    "last_name" : "Simpson",
    "industry" : "Energy",
    "eula" : "yes",
    "title" : "Safety Inspection Engineer",
    "phone" : "5558675309",
    "company" : "Example",
    "state" : "OR",
    "department" : "Engineering",
    "first_name" : "Homer",
    "email" : "homer@example.com"
   }
   ```

   c. After saving changes to the file, pass it with the `--file` option to register Tableau Server:

```
tsm register --file /path/to/<registration_file>.json
```

For example:

```
tsm register --file /usr/share/tableau-reg-file.json
```

Activate Tableau Server Offline

When you install Tableau Server, you have to activate at least one product key, but we recommend that you activate all Tableau Server licenses found in the Tableau Customer Portal. Doing this activates the server, and specifies the number of license levels you can assign to users. For offline activations, you should activate the product key listed in the **Offline Activation Id** field in the Tableau Customer Portal. For information about finding the right key, see the Find the Correct Key to Activate on Tableau Server Knowledge Article.

There are also times you may need to activate licenses after Tableau Server is installed, for example, if you add capacity to your server, or get a new product key. If you don't have your product key, you can get it from the Tableau Customer Account Center.

> **Note:** Activating any product key after Tableau Server has already started will require a Tableau Server restart for the changes to take effect.

In most cases, you can activate your key directly from Tableau Server, either during installation, or later, using the Tableau Services Manager (TSM) Licenses page, but there are some circumstances that don't allow you to do this. If your computer is not connected to the internet for example, or has a firewall that restricts access outside your intranet. In these cases you need to do an offline activation.
Tableau Server in a Container only supports license activation using Server ATR. Offline activation using Server ATR is supported in 2023.1 and later. This functionality is available in Containers but requires extra steps and approval. If you need to run Tableau Server in a Container in an air-gapped or offline environment, contact your Account representative for more information.

Offline activation and login-based license management (LBLM)

Beginning in Tableau Server version 2023.1.0, offline activation is supported for LBLM when your server is configured to use the Authorization-to-Run (ATR) service. You can only configure Tableau Server to use the ATR service during a new install. Upgrading customers with existing server installations need to install a new instance of Tableau Server version 2023.1.0 or later and restore a backup of their existing installation to that new instance. For information on this process, see Using a Blue/Green approach for upgrading Tableau Server. For more information about ATR service, see Activate Tableau Server Using the Authorization-To-Run (ATR) Service.

Offline activation and updateable subscription licenses (USL)

Offline activation of updateable subscription licenses requires special steps. For details, see Activating USL in Offline or Disconnected Environments.

There are two scenarios in which you may need to do an offline activation:

- Offline activation during install—To complete an offline activation when you are installing Tableau Server.
- Offline activation of licenses after install—To complete an offline activation after your server is installed and running.

Offline activation overview

Offline activation of Tableau Server involves the following steps:

1. Generate an offline activation request file.
2. Copy the offline activation request file to a computer with internet access.
3. Upload the offline activation request file to the Tableau activation website.
4. Download the resulting offline activation response file from the website. You'll use this file to activate Tableau Server

Offline activation file name changes

Beginning in Tableau Server version 2023.1, the Tableau licensing system supports two underlying licensing technologies. From an administrative perspective, the only configuration difference between the two systems is the file types that are generated and consumed for offline activation. The licensing technology is determined during the initial installation of Tableau Server, and cannot be changed after install.

We refer to the legacy (and still supported) version of licensing technology as FlexNet. The latest version of the technology is referred to as Server ATR. For more information, see Activate Tableau Server Using the Authorization-To-Run (ATR) Service. The following table describes the file naming nomenclature for each technology. The table also includes the generic reference.

Generic file name	Server ATR file names	FlexNet file names
OfflineActivationRequest	OfflineActivationRequestFile_yyyyMMdd.hhmmss.json	Tableau-OfflineActivationRequest.tlq
OfflineActivationResponse	OfflineActivationLicensingAtrs.zip	activation.tlf

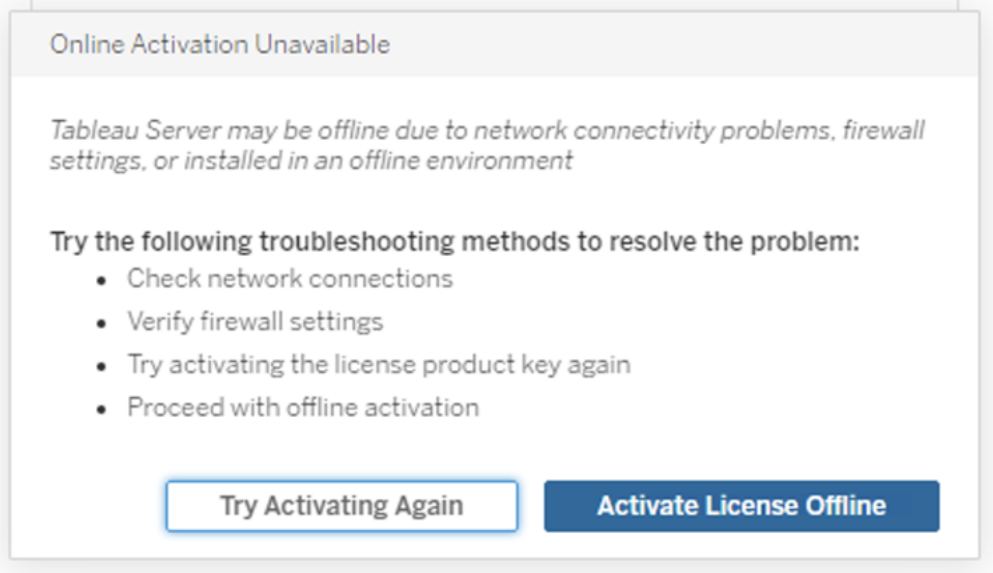
Note: Since this documentation supports multiple versions of Tableau Server, we will use the generic file name references (OfflineActivationRequest and OfflineActivationResponse) for the rest of this topic. You can identify the licensing technology your Tableau Server installation uses according to the file type that generated in the steps that follow.

Use the TSM web interface

If you attempt to activate your product key from the TSM licenses page and see a dialog that says online activation is unavailable, you can activate the key offline. The offline activation

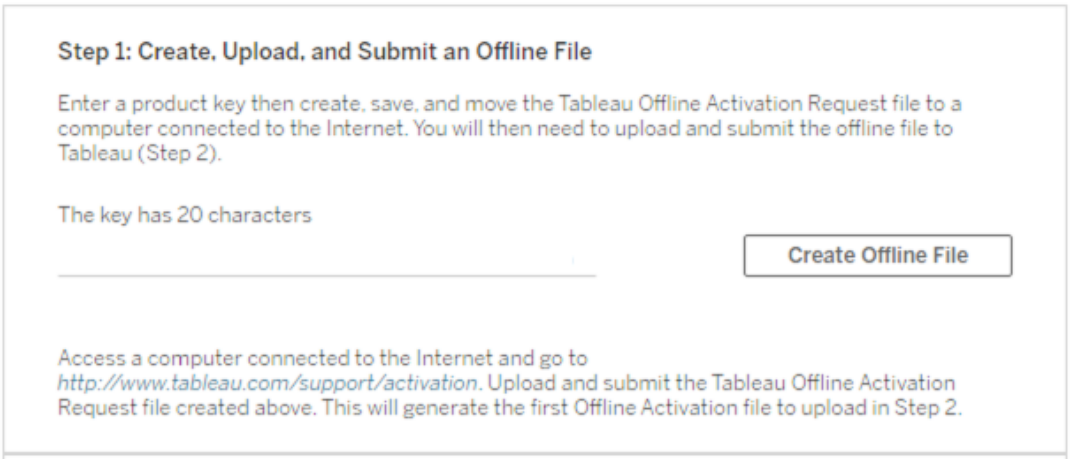
process must be completed once for each product key.

1. Click **Activate License Offline**.



2. Create an offline activation request file (OfflineActivationRequest) for the product key.

   Create an OfflineActivationRequest file you will upload to the Tableau activation web-site. If your product key is not pre-filled in the form, enter your key and click **Create Off-line File** to generate an OfflineActivationRequest file on the local computer.



Copy the OfflineActivationRequest file to a computer with internet access. You need to upload this file to the Tableau activation website to generate an activation response file.

3. Upload and submit the OfflineActivationRequest file.

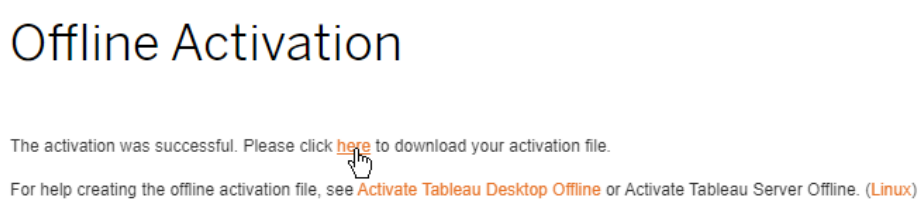
   You will upload and submit the OfflineActivationRequest file to the Tableau activation website. This automatically generates an activation response file (OfflineActivationResponse) that you can download and copy back to the Tableau Server computer.

   a. On the computer where you copied the OfflineActivationRequest file, open a browser and go to http://www.tableau.com/support/activation to open the Tableau Support Activation page.

   b. On the Offline Activation page, click **Choose File** to select the OfflineActivationRequest file.

   c. Click **Upload Activation File** to submit the file to the Tableau activation website.

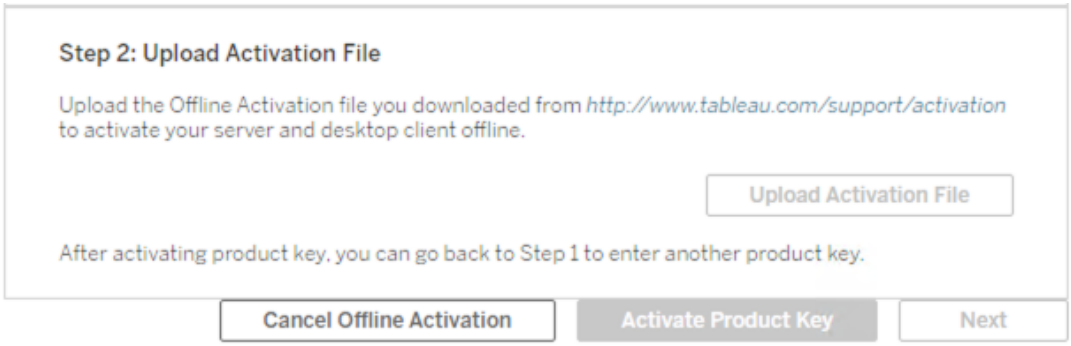
   d. Click the `here` link to download the OfflineActivationResponse file to your computer.



   e. Copy the OfflineActivationResponse file to the computer where Tableau Server is installed.

4. Upload the OfflineActivationResponse file.

On the Tableau Server computer, click **Upload Activation File** to upload the OfflineActivationResponse file to Tableau Server. When you do this successfully, the **Activate Product Key** button is enabled.



5. Click **Activate Product Key** to complete the offline activation.

6. (Skip this step if you are installing Tableau Server for the first time.)

   Restart Tableau Server for licensing changes to take effect.

Use the TSM CLI

Step 1. Log in to Tableau Services Manager

Before you can proceed you must log in to Tableau Services Manager (TSM). To log in to TSM, run the following command:

```
tsm login -u <username>
```

What if I can't log in?

The account that you use to configure the rest of the installation must be a member of the `tsmadmin` group that was created during initialization. To view the user accounts in the `tsmadmin` group, run the following command:

```
grep tsmadmin /etc/group
```

If the user account is not in the group, run the following command to add the user to the `tsmadmin` group:

```
sudo usermod -G tsmadmin -a <username>
```

After you have added the user to the `tsmadmin` group, run the `tsm login` command.

Step 2. Generate an offline activation request file

1. On the initial node, open a terminal session.
2. Type this command to get your offline activation file:

   ```
   tsm licenses get-offline-activation-file -k <product-key> -o
   <target-directory>
   ```

   You can get your product key from the Tableau Customer Portal. The target directory must already exist.
3. Copy the offline activation file (OfflineActivationRequest) from the target directory to a computer that has internet access.

Step 3. Upload the offline activation request to the Tableau activation website

1. On the computer that has internet access, go to the Tableau Product Activations page.
2. Complete the instructions to upload your OfflineActivationRequest file.

   This creates an activation response file (OfflineActivationResponse).
3. Download the OfflineActivationResponse file from the Tableau activation website.

Step 4. Initialize or activate your license

1. Copy the OfflineActivationResponse file to a location accessible from the Tableau Server computer.
2. Run the following command:

```
tsm licenses activate -f <path-and-activation-file>
```

Note: When using ATR to activate Tableau Server, `<path-and-activation-file>` should point to the packaged OfflineActivationResponse `.zip` file. Do not unzip the OfflineActivationResponse file prior to running this command.

3. (Skip this step if you are installing Tableau Server for the first time.)

   Restart Tableau Server for licensing changes to take effect:

   ```
   tsm restart
   ```

4. (Optional) To verify that all licenses are activated, you can run this command:

   ```
   tsm licenses list
   ```

   If you have completed the steps above, you should see a success message:

   ```
   Activation successful.
   ```

   Tableau Server is activated. If you need additional assistance, contact Tableau Technical Support.

Deactivate Product Key

There are some scenarios where you must deactivate a product key:

- Changing a hardware configuration
- Changing product keys
- Moving a product key to a new installation

Before you begin

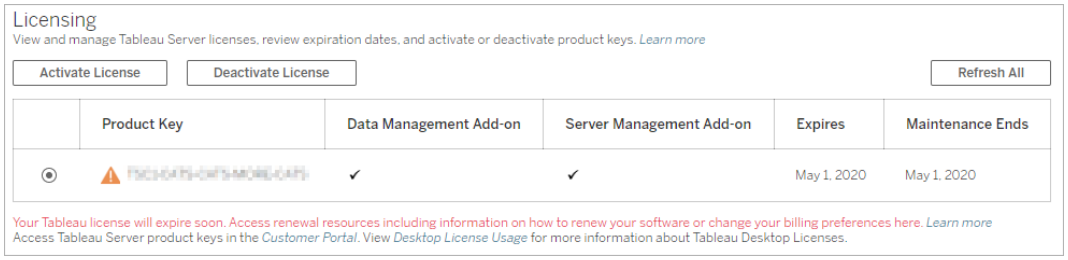
Verify that you are removing the correct product key(s). You can view license details by running tsm licenses list in the CLI.

Use the TSM web interface

1. Open TSM in a browser:

   ```
   https://<tsm-computer-name>:8850
   ```

2. Click **Configuration** and **Licensing** .



3. Select the product key that you want to deactivate, and then click **Deactivate License**.

4. After the key is deactivated, restart Tableau Server.

> **Note:** If no other product keys remain activated before restarting, Tableau Server will not restart in a useable state. If this occurs, you will not be able to use Tableau Server until you activate a new product key in TSM.

Use the TSM CLI

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).

2. Run the following commands:

```
tsm licenses deactivate --license-key <product-key>

tsm restart
```

Deactivate Tableau Server Offline

If Tableau Server does not have Internet access, you can use the instructions in this topic to deactivate Tableau Server. You will need to use a second computer that is able to access the Internet to complete this activation process.

> **Note:** At this time, you cannot deactivate Tableau Server if the ATR Service is enabled on your Tableau Server deployment.

1. On the Tableau Server initial node, create a directory to store the offline deactivation file that is created in the next step.
2. Log in to TSM using a Tableau Administrator account, and then run the following command:

   ```
   tsm licenses get-offline-deactivation-file -k <productkey> -o
   <deactivation-file-directory>
   ```

3. Move the `TableauOfflineDeactivationRequest.tlq` file from the deactivation file directory that you specified in the previous command to a trusted computer that has Internet access.
4. On the trusted computer that has Internet access, open a web browser and visit the Tableau Product Activations page. Follow the instructions on that page to submit your `TableauOfflineDeactivationRequest.tlq` file.

   > **Important:** This process will not work with Microsoft Edge browser.

5. When prompted, save the product key return file (`return.tlr`) from the **Product Activations** page.
6. Move the product key return file (`return.tlr`) from the trusted computer that has Internet access to the Tableau Server initial node that runs the Licensing Server service.
7. Log in to TSM using a Tableau Administrator account, and then run the following command:

```
tsm licenses deactivate -f <path-to-license-key-return-file>

tsm restart
```

Automate Licensing Tasks

You can use `tsm licenses` to perform licensing tasks such as activating or deactivating a Tableau Server product key on- or off-line, and getting associated files for offline activation or deactivation. However, Tableau Server must already be deployed and configured. You can automate these licensing tasks using the Tableau Services Manager API.

```
C:\ProgramData\Tableau\Tableau Server\data\tabsvc\logs\
```

Troubleshoot Licensing

This topic includes instructions for troubleshooting issues related to Tableau Server licensing.

Handle an unlicensed server

Tableau offers two licensing models: role-based and core-based. To learn more about role-based and core-based licensing, see Licensing Overview.

role-based licensing requires each active user account to be covered by a license. role-based licenses have a defined capacity, or number of users that they allow. Each user is assigned a unique user name on the server and is required to identify themselves when connecting to the server.

Core-based licensing has no constraints on the number of user accounts in the system, but it does restrict the maximum number of processor cores that Tableau Server can use. You can install Tableau Server on one or more machines to create a cluster, with the restriction that the total number of cores in all the machines does not exceed the number of cores you have licensed and that all of the cores on a particular machine are covered by the license.

Unlicensed role-based server

The most common reason for a server that has role-based licensing to be unlicensed is an expired product key or an expired maintenance contract.

Unlicensed core-based server

A core-based server can become unlicensed for a variety of reasons, such as an expired product key or when Tableau Server nodes running licensed processes cannot contact the Tableau Server node running the License Manager service. To learn more about licensed processes, see Tableau Server Processes.

When the server is unlicensed you may not be able to start or administer the server. You can, however, manage your licenses using the tsm licenses command.

Unlicensed server administrator

All Tableau Server administrators require a user license. Tableau Server administrators will always consume the highest role available. If a Creator product key is activated, the Tableau Server Administrator(s) will take this role. If the highest role available on Tableau Server is an Explorer, the Server Administrator will take the Explorer role. If Creator licenses are added to the server, any existing Server Administrator accounts using Explorer licenses will automatically convert to use Creator licenses.

TSM administrator accounts do not require licenses.

If the license that the server administrator is using expires, then the account will become unlicensed and will be unable to sign in.

Verify the expiration date of your license(s) for the administrators on the server:

- Run tsm licenses list.
- Compare the date with the date displayed in the Tableau Customer Portal.
- If the portal does not display the date that you expect, contact Customer Success.
- To renew your license, visit the Tableau renewal web page.
- Run the tsm licenses activate command to activate a new license for the administrator account(s).

If the TSM date matches the portal date and the following refresh operation fails, contact Tableau Support.

If the license for your administrator account has expired or will expire soon, you will need to activate a new license for the account. Alternatively, you can unlicense a non-administrator user to free a license for the server administrator account.

If a Tableau Server administrator is using a Creator, Explorer or Viewer license and their license expires, they will use another license of the same type, if available. If no license seats are available the user will become “unlicensed”.

Important: Do not restart Tableau Server until you have activated a new license or transferred a site role for the server administrator account.

Troubleshoot role-based licensing

This section provides information about resolving issues that can occur when adding the role-based Viewer, Explorer and Creator licenses to Tableau Server or Tableau Cloud, or when these licenses expire. The highest available license type is Creator, followed by Explorer, and finally Viewer. To learn more about role-based licensing, see Licensing Overview.

A user or administrator is unlicensed due to license expiration

To avoid having users unexpectedly become unlicensed or move to another site role, you should always do one of the following before the license that they are currently using expires:

- Renew and activate a replacement license. If a user occupies a Creator, Explorer or viewer license and their license expires, they will use another license of the same type, if available.
- Change the site role of those users to allow the use of a license that is not due to expire.

To learn how site roles can be changed to require a different license, see Set Users’ Site Roles.

The reassignment of users to new licenses is governed by the following logic:

- When a Server Administrator user occupies a Creator license and their license expires (with no replacement licenses available), they are reassigned to an Explorer license if any Explorer licenses are available. This license reassignment occurs in order of most recent login. Server Administrators displace other users who might be currently using an Explorer license. If no Creator or Explorer licenses are available a Server Administrator becomes unlicensed.
- When a non-Server Administrator user occupies a Creator license and their license expires (with no replacement licenses available), they become unlicensed. To avoid having these users become unlicensed, change their site role prior to license expiration. This is especially important for users in the Site Administrator Creator site role, who must move to the Site Administrator Explorer site role before their Creator license expires to avoid losing Site Administrator capabilities.
- When a non-Server Administrator user occupies an Explorer or Viewer license and their license expires (with no replacement licenses available), they are upgraded to a higher license type, if licenses of that type are available. Specifically, the following occurs when a license expires:
    - Users who occupy an Explorer license will move to a Creator license, if available (with no change to site role).
    - Users who occupy a Viewer license will move to an Explorer license, if available. If no Explorer licenses are available, these users will move to a Creator license, if available (with no change to site role).
    - If no licenses are available at the higher license types, those users are moved to Unlicensed.

Users are reassigned to a new license as described above in order of most recent login, with lower license types reassigned first (first Viewer, then Explorer, and then Creator).

For example: Two users with a Viewer license, a user with the Creator license, and two Server Administrators with a Creator license all have their licenses expire. Four unexpired Explorer licenses are available for these users. In this situation, the following occurs in the order shown below:

1. The user with a Viewer license who logged in most recently is reassigned to an Explorer license.
2. The second user with a Viewer license is reassigned to an Explorer license.

3. The Server Administrator user with a Creator license who logged in most recently is reassigned to an Explorer license, and then the second Server Administrator with a Creator license is reassigned to the remaining Explorer license.
4. The user with the Creator license becomes unlicensed.

Server Administrator site role is unchanged when using a Creator license

Server Administrators gain Creator capabilities if Creator licenses are available in Tableau Server, with no change to their site role name. All other Tableau Server and Tableau Cloud users gain Creator licenses only if assigned to a site role that includes Creator in its name.

Licenses are not immediately available

When you add a role-based license to Tableau Server, those licenses become available to all users when you restart Tableau Server.

A user with a Viewer license cannot open Tableau Server or Tableau Cloud workbooks from Tableau Desktop

A user with a Viewer license who also has a separate Tableau Desktop license will be unable to open workbooks on Tableau Server or Tableau Cloud using Tableau Desktop. To open workbooks such using Tableau Desktop, that user will need an Explorer or Creator license on Tableau Server or Tableau Cloud.

Migrate from Core-Based to Role-Based Licensing

You can migrate Tableau Server from a core-based license metric (which counts the processor cores on which you have Tableau Server installed) to a role-based license metric (which counts named users). To learn more about licensing metrics, see Licensing Overview.

Prepare for migration to role-based licensing

Core-based licenses allow an unlimited number of users, including view-only guest accounts. Every user has a site role when they are added to Tableau Server, and these users and site roles persist when licensing is changed. Because role-based licenses limit the number of users, you should ensure that your new role-based licenses accommodate the number of

users who are currently connecting to Tableau Server, including the users currently using guest accounts.

Important: If your new role-based licenses don't accommodate the full number of users, some users will move to the **Unlicensed** site role, which can be very CPU intensive. If you have a large number of users, your application server processes may become unavailable until all the users are processed. This operation could take hours to complete, so plan accordingly.

Before you migrate, verify that you have user licenses and corresponding product keys to allow all users to access Tableau Server after the migration is complete:

- Sign in to the Tableau Customer Portal to verify licenses and to copy the corresponding product key(s).
- To learn more about site roles, see Set Users' Site Roles.
- To count the number of users in your Tableau Server installation, export a list of users to count them with a tool such as Microsoft Excel. To learn how to export a list of users, see Export a User List.

Migrate to role-based licensing

To migrate to role-based licensing you must stop Tableau Server, deactivate the core-based product key, activate the role-based product key(s), and then start Tableau Server. Because this process will result in a restart of Tableau Server and cause downtime for Tableau Server users, you should migrate licensing during a period of low usage.

1. Stop Tableau Server:

   Use the `tsm stop` command.

2. Deactivate the core-based product key(s):

   Use the `tsm licenses deactivate` command with the core-based product key(s).

3. Activate the role-based product key(s):

   Use the `tsm licenses activate` command with the role-based product key(s).

4. Start Tableau Server:

   Use the `tsm start` command.

Use role-based licenses on a server with core-based licensing

The 2018.1 release of Tableau Server allows you to add Creator role-based licenses to Tableau Server installations with existing core-based licensing.

> **Note:** If you upgrade a Tableau Server installation to 2018.1 without activating role-based licenses, Tableau Server will continue to operate as it did previously, with no changes to Tableau Server UI or permissions except that the legacy **Viewer** site role is renamed to **Read Only**. In version 2018.2, the **Read Only** site role was deprecated and once again became the **Viewer** site role.

When **Creator** licenses are introduced to Tableau Server, all Server Administrator users are required to have **Creator** license roles, which may require additional licenses. Administrators can activate additional **Creator** licenses using the `tsm licenses activate` command in Tableau Server to increase licensed user capacity. Activating these role-based licenses gives you a combination of the capabilities granted by role-based Creator licenses and the capabilities granted by your core-based license. To learn more about the different types of role-based licenses, see role-based licenses.

> **Note:** The unlimited number of users who have access to Tableau Server under core-based licensing have equivalent capabilities to users with an **Explorer** license under role-based licensing. **Creator** functionality is limited to the defined user license model(s).

For example, if a Tableau Server installation has a 16 core license that includes guest access, and you added an Updatable Subscription License (USL) product key(s) with 10 Creator licenses, that server would have the following capabilities:

- A limit of 16 processor cores on hardware that runs Tableau Server
- Guest access
- Unlimited Explorer/Viewer licenses (from the unlimited user licenses that come with a core license)
- 10 Creator licenses

Example of completing a migration from core-based licensing

To extend the example above: If the core-based license was then deactivated, the following capabilities would be available:

- No limits on server hardware
- No guest access
- 10 Creator licenses

If you then added 50 Explorer licenses and 200 Viewer licenses, after deactivating your core-based license, the following capabilities would be available:

- No limits on server hardware
- No guest access
- 10 Creator licenses
- 50 Explorer licenses
- 200 Viewer licenses

Quick Start: Use Login-based License Management with Tableau Server

You can use the following steps to get up and running quickly with login-based license management.

Step 1: Install Tableau Server

To use login-based license management to activate Tableau, you must install Tableau Server version 2021.1 or later, and activate it with a product key that is enabled for login-based license management. Login-based license management enables Tableau Creator users to sign-in and activate Tableau Desktop or Tableau Prep Builder.

> **Note:** If your Tableau Server product key is not enabled for login-based license management in the Tableau Customer Portal, contact your Tableau account representative.

The following topics provide additional information about installing Tableau Server and using login-based license management to activate Tableau Desktop and Tableau Prep Builder:

- Install and Configure Tableau Server

To change login-based license management configuration settings, or to disable login-based license management on Tableau Server, see Change login-based license management settings.

Step 2: Add authorized users to Tableau Server

After you activate Tableau Server, you can add authorized users. When adding users, you'll need to select the site role for that user, for example Creator, Explorer, or Site Administrator. Users who will activate Tableau Desktop or Tableau Prep Builder must be assigned to a Creator site role (Creator or Site Administrator Creator). For more information on site roles, see Set Users' Site Roles.

Step 3: Activate Tableau Desktop or Tableau Prep Builder

A Creator user must download and install Tableau Desktop and/or Tableau Prep Builder on their computer before they can be activated using login-based license management. For more information, see "Install Tableau Desktop or Tableau Prep Builder from the User Interface" in the Tableau Desktop and Tableau Prep Deployment Guide.

1. Launch Tableau Desktop or Tableau Prep Builder.

   If this is the first time you've launched Tableau Desktop or Tableau Prep Builder, you have the option of starting a 14-day free trial or activating Tableau.

2. Click **Activate Tableau** to skip the free trial.

3. On the **Activate Tableau** screen, click **Activate by signing in to a server**, to sign-in and activate Tableau using login-based license management.

4. When prompted, specify the URL for Tableau Server to sign-in using your credentials.

5. After you are authenticated by Tableau Server, Tableau Desktop and/or Tableau Prep Builder are activated and ready to use.

Login-based License Management

Login-based license management helps you manage licensing for users with Creator roles on Tableau Server and Tableau Cloud. Users with Explorer or Viewer roles cannot use this feature. If you're using Role Based Subscriptions with Tableau Server or Tableau Cloud, you can simplify your license management using login-based license management to eliminate separate Tableau Desktop and Tableau Prep Builder product keys. You only need to manage one or more product keys for on-premises Tableau Server, or in the case of Tableau Cloud, you don't need to manage any product keys at all.

You only need one Tableau Server or Tableau Cloud site to authorize an individual Tableau Desktop or Tableau Prep Builder. Login-based license management is enabled per production instance of your Tableau Server deployment. You can assign Creator roles to users who use Tableau Server, Tableau Desktop, and Tableau Prep Builder for license activation and centralized license management. In addition, login-based license management gives you more visibility into license usage.

You can use administrative views on Tableau Server and Tableau Cloud to see the assignment and use of Creator seats in one place. Administrative views provide information about who is using your Creator licenses, and shows the most recent license lease and version of both Tableau Desktop and Tableau Prep Builder, which helps you to monitor your Tableau deployments.

The following Tableau products support login-based license management:

- Tableau Cloud
- Tableau Server 2020.1+
- Tableau Desktop 2020.1+
- Tableau Prep Builder 2020.1.3+

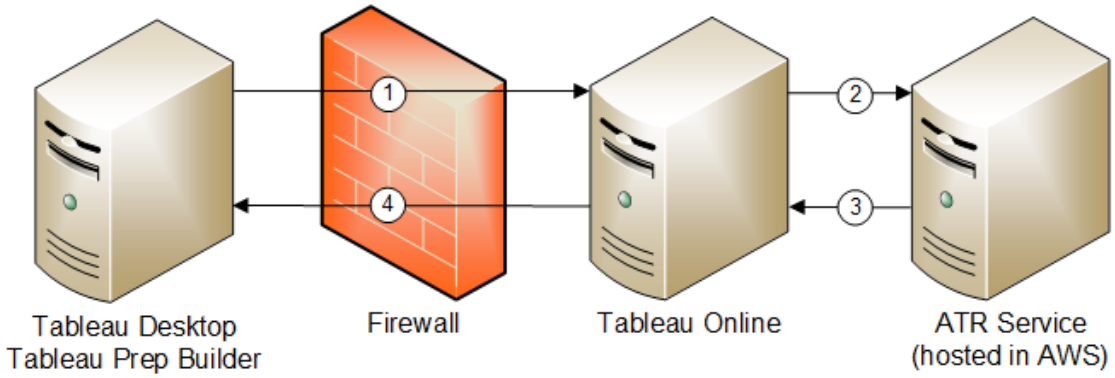
If your Tableau Server is not enabled for login-based license management, contact your Tableau sales representative to obtain a special login-based license management-enabled product key.

How login-based license management works

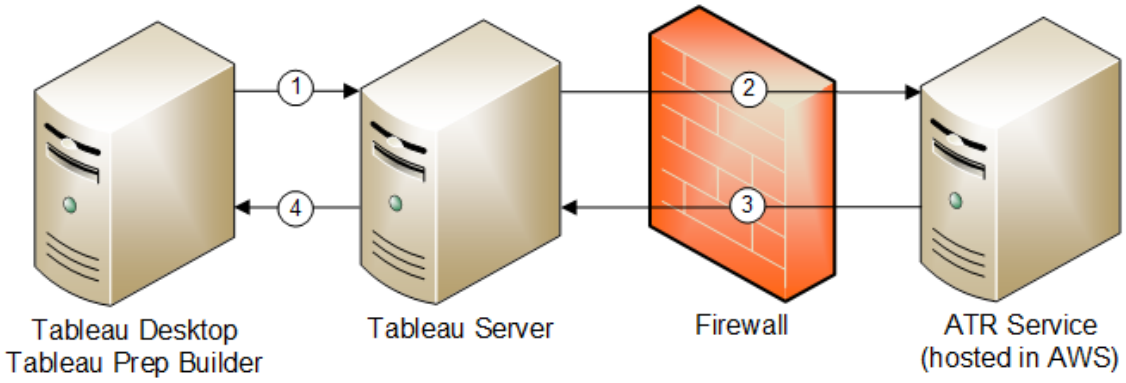
When login-based license management is in use, Tableau Desktop or Tableau Prep Builder periodically contacts Tableau Server or Tableau Cloud in order to verify that the user is a Creator and to obtain an authorization to run. Tableau Server or Tableau Cloud contacts the Tableau-hosted authorization-to-run (ATR) service to verify that the Tableau Desktop client is allowed activation. The ATR service, hosted on AWS, verifies and manages the license and the length of the authorization window. The diagrams below show the communication process between the client, Tableau Server or Tableau Cloud, and the licensing service.

Tableau uses common ports (80 and 443) to make internet requests. In most cases, the computer where Tableau Desktop or Tableau Prep Builder is installed and the network it is running on are already configured to allow the necessary access. For more information about the ports used by Tableau, see Tableau Services Manager Ports.

Login-based license management with Tableau Cloud



Login-based license management with Tableau Server



1. You install Tableau Desktop or Tableau Prep Builder and select **Activate with your credentials**. You sign into your Tableau Cloud or Tableau Server site.
2. Tableau Cloud or Tableau Server verifies that you are a Creator. If not, you get an error. If yes, Tableau Cloud or Tableau Server communicates with the ATR service.
3. The ATR service returns an ATR lease to Tableau Cloud or Tableau Server.
4. Tableau Cloud or Tableau Server provides an ATR lease to Tableau Desktop or Tableau Prep Builder to finalize activation.

Use login-based license management

To use login-based license management, you either need to use Tableau Cloud or you need to install Tableau Server and activate it with a login-based license management enabled product key. The end user needs to be assigned the Creator role on either Tableau Cloud or Tableau Server and that user will need to install Tableau Desktop. Although the user can activate Tableau Desktop on up to two machines, only one Creator seat needs to be allocated.

Note: As of Tableau Server version 2023.1, LBLM is supported in offline deployments. To deploy Tableau Server with LBLM enabled, you must configure Tableau Server to use

Authorization-To-Run (ATR) Service. For more information, see Activate Tableau Server Offline.

Step 1: Install Tableau Server

If you already have Tableau Server installed, skip to (Optional) Step 2: Change login-based license management settings. Otherwise, you'll need to install Tableau Server:

1. Refer to the Deploy chapter to install Tableau Server.
2. During activation, you must enter a Tableau Server product key that supports role-based subscription and login-based license management.
3. Add users to your Tableau Server and set them to one of the three Creator roles: Server Administrator, Site Administrator Creator, or Creator. This assigns a Creator license to these users on Tableau Server. For more information, see Set User's Site Roles.

Important: If you're using connected clients and login-based license management, do not disable automatic client authentication. In addition, do not sign out of Tableau Server using the connected desktop user interface after enabling login-based license management. Otherwise, login-based license management cannot automatically refresh the license activation or provide current data to the LBLM Usage report on Tableau Server or Tableau Cloud. For more information about connected desktop, see Automatically keep Tableau Desktop connected to Tableau Server or Online.

(Optional) Step 2: Change login-based license management settings

Login-based license management is enabled by default for Tableau Cloud, Tableau Server, and Tableau Desktop starting with version 2020.1. You can, however, change some login-based license management settings.

Disable login-based license management on Tableau Server

Although login-based license management is the preferred method of activation for Tableau Desktop and Tableau Prep Builder, you may have Tableau Server installations in your organization used for testing that don't need to use login-based license management. On Tableau Server, you use the Tableau Services Manager (TSM) command line utility to turn off login-based license management.

To turn off login-based license management, at a command prompt, type:

```
tsm configuration set -k licensing.login_based_license_management.enabled -v false
```

```
tsm pending-changes apply
```

Change login-based license management settings on Tableau Desktop or Tableau Prep Builder at install

To change login-based license management settings at the command line, you can run the installer .exe file from your computer's command line as an administrator. If you need to extract the .msi files, follow the instructions to Extract and run the Windows (MSI) installer.

To use a duration length other than the default of 14 days/1209600 seconds, include the `ATRREQUESTEDDURATIONSECONDS` switch. For example:

```
tableauDesktop-64bit-2020-1-0.exe /quiet /norestart ACCEPTEULA=1 ATRREQUESTEDDURATIONSECONDS=43200
```

You must run the command from the directory where the .exe file is located or specify a full path to the location of the .exe file on the computer. Do not run the setup program from a shared directory on your network. Instead, download the .exe file to a directory on the computer where you're installing.

The following example shows the Windows installer command that disables login-based license management:

```
tableauDesktop-64bit-2020-1-0.exe /quiet /norestart ACCEPTEULA=1
LBLM=disabled
```

or

```
tableauPrepBuilder-64bit-2020-1-0.exe /quiet /norestart
ACCEPTEULA=1 LBLM=disabled
```

To set the default URL for the Tableau Server you want users to use for activation when using login-based license management, add the `ACTIVATIONSERVER` or `WorkGroupServer` option.

To update the exe:

```
tableauDesktop-64bit-2021-4-0.exe /quiet /norestart ACCEPTEULA=1
ACTIVATIONSERVER=http://<tableau_server_url>
```

To update the registry:

```
reg.exe add HKEY_LOCAL_MACHINE\SOFTWARE\Tableau\Tableau 2021.4\Set-
tings /f /v WorkGroupServer /d https://<tableau_server_url>
```

Note: The `ACTIVATIONSERVER` option is only intended for first time activation. If you have previously signed-in to this version of Tableau, you use the WorkGroupServer (Windows) or WorkgroupServer (macOS) option. For example, if you are using Tableau Desktop version 2021.1 on Windows and have previously signed-in successfully, you would use the `WorkGroupServer` option to specify an activation server. On Tableau Desktop on macOS, you would use the `WorkgroupServer` option. On macOS, this option is case sensitive and uses a lowercase "g".

Change login-based license management settings on Tableau Desktop by editing the registry

If Tableau Desktop is already installed, you can change login-based license management settings by editing the Windows registry.

To turn off login-based license management:

```
reg.exe add HKEY_LOCAL_MACHINE\SOFTWARE\Tableau\ATR /f /v LBLM /d disabled
```

To make login-based license management the only login option:

```
reg.exe add HKEY_LOCAL_MACHINE\SOFTWARE\Tableau\ATR /f /v LBLM /d required
```

Or, you can enable, disable, or require login-based license management by editing the registry directly:

1. As an administrator on the computer running Tableau Desktop, make a backup of the registry file before you make any changes to it.
2. Edit the registry, and in `HKEY_LOCAL_MACHINE\SOFTWARE\Tableau`, find the hive named ATR and modify the LBLM value to reflect the desired setting:
    a. Name: LBLM.
    b. Data: enabled, disabled, or required.
3. Restart Tableau so the changes take effect.

Change login-based license management settings on Tableau Desktop on macOS

To change login-based license management settings on macOS, run the following commands in a terminal window to update the preferences file, and then install or restart Tableau Desktop.

To turn off login-based license management:

```
sudo defaults write /Library/Preferences/com.tableau.ATR LBLM "disabled"
```

To make login-based license management the only login option:

```
sudo defaults write /Library/Preferences/com.tableau.ATR LBLM "required"
```

To set the default URL for the Tableau Server you want user to use for activation when using login-based license management on macOS.

First time activation:

```
sudo defaults write /Library/Preferences/com.tableau.ATR LBLM "required"
```

Subsequent activations:

```
sudo defaults write /Library/Preferences/com.tableau.Tableau-<version> Settings.WorkgroupServer "https://<tableau_server_url>"
```

Note: On Tableau Desktop on macOS, you would use the WorkgroupServer option. On macOS, this option is case sensitive and uses a lowercase "g".

Login-based license management settings

You use the following settings to change login-based license management, set the ATR duration, and set the activation server URL.

Setting	Value	Description
`LBLM`	enabled, disabled, or required	Set to `enabled` (the default), the licensing screens will present the two options for activation (product key, or credentials). Set to `disabled`, login-based license management will not appear on the licensing screens. Set to `required`, login-based license management is the only way to activate the Tableau Desktop (when the licensing screen appears, it will offer only the credentials option for activation). **Note:** When `LBLM` is set to enabled or required, reporting is also enabled.
`ATRREQUESTEDDURATIONSECONDS`	1209600	Sets the authorization to run (ATR) duration (in seconds), which is the length of time that an instance of Tableau Desktop and Tableau Prep Builder is authorized to run. The default is 1209600 seconds (14 days). Do not use commas as separators in the value.

`ACTIVATIONSERVER`	http://<tableau_server_url>	For first time activation, sets the default URL for the Tableau Server you want users to use for activation.
`WorkGroupServer` (Windows) `WorkgroupServer` (macOS)	http://<tableau_server_url>	For updates to the Windows registry or macOS plist, sets the default URL for the Tableau Server you want users to use for activation. **Note:** On Tableau Desktop on macOS, you would use the WorkgroupServer option. On macOS, this option is case sensitive and uses a lowercase "g".
`REPORTINGFREQUENCYSECONDS`	3600	Sets the default (in seconds) for how often the login-based license management report is sent to Tableau Server or Tableau Cloud. The minimum setting is 60 seconds, and the default is 3600 seconds (one hour). Change this setting to reduce the load on Tableau Server or to reduce network traffic. No matter what the authorization-to-run (ATR) lease duration is set to, `REPORTINGFREQUENCYSECONDS` sets the time interval that the Tableau Desktop or Tableau

		Prep Builder client report login-based license management activations back to Tableau Server or Tableau Cloud.

Additional configuration for virtual deployments

Tableau Desktop and Tableau Prep Builder periodically contact Tableau Cloud or Tableau Server to verify that Tableau is authorized to run, based on its license. Tableau Cloud or Tableau Server then contacts the ATR service to verify the license and the length of the authorization window.

When configuring a virtual (non-persistent) deployment of Tableau Desktop or Tableau Prep Builder, the duration time should be set to one of the lower values such as 4 or 8 hours in order to avoid an over-use activation error message. After the virtual machine (VM) is returned, the ATR service will handle the activation monitoring.

The following flags should be used on a source image prior to publishing Tableau Desktop to end users. Each end user will be activating the software by logging into Tableau Server or Tableau Cloud with each new VM delivered. No product keys need to be entered if the end user is a Tableau Creator on Tableau Server or Tableau Cloud.

If you are using login-based license management for Tableau Desktop or Tableau Prep Builder on a VM, you may get an error message that your license information has changed whenever you launch a new VM for Tableau Desktop or Tableau Prep Builder. This error forces a restart, which then asks you to register Tableau Desktop again. This error occurs because the ATR service sends a new token that doesn't match the license cache.

Microsoft Windows

To prevent the error from occurring on Microsoft Windows, you can use the `SYNCHRONOUSLICENSECHECK` and `SILENTLYREGISTERUSER` options with the Windows installer. For example:

```
tableau-setup-std-tableau-2020 SYNCHRONOUSLICENSECHECK="true"
SILENTLYREGISTERUSER="true" ATRREQUESTEDDURATIONSECONDS=14400
```

or

```
tableauDesktop-64bit-2020-1-0.exe /quiet /norestart ACCEPTEULA=1
ATRREQUESTEDDURATIONSECONDS=14400 ACTIVATIONSERVER=http://<tableau_
server_url> SYNCHRONOUSLICENSECHECK="true" SILENTLYREGISTERUSER-
R="true"
```

If Tableau Prep Builder and Tableau Desktop are being delivered on one Virtual Desktop, `ATRREQUESTEDDURATIONSECONDS` only needs to be set during Tableau Desktop installation. However, if you plan to install Tableau Prep Builder as a stand-alone, you'll need to set `ATRREQUESTEDDURATIONSECONDS` during Tableau Prep Builder installation.

Or, you can edit the following registry keys on the source image:

```
Reg key path: HKLM\SOFTWARE\Tableau\<Tableau version>\Settings\
```

```
Reg key (String value, need to set to true to make that feature
enabled)
```

```
SynchronousLicenseCheck
```

```
SilentlyRegisterUser
```

macOS

To prevent the error from occurring on macOS, run the following command to set the `LicenseCache.Desktop` flag to `'false'`.

```
sudo defaults write ~/Library/Preferences/com.tableau.Tableau-<ver-
sion>.plist LicenseCache.Desktop false
```

(Optional) Step 3: Change the authorization to run (ATR) duration

The login-based license management default settings for the authorization to run (ATR) duration are appropriate for most environments, but you can change these default settings if needed. Login-based license management uses the default authorization to run (ATR) duration of 1209600 seconds (14 days), which is the length of time that an instance of Tableau Desktop and Tableau Prep Builder is authorized to run. This means that after the initial authorization, you could use Tableau without any network connection for 14 days before the activation expired.

The default duration value for login-based license management is not appropriate for delivering a non-persistent VM delivery solution to end users. The ATR duration should be lowered to 4 or 12 hours depending on VM use. When a new VM is delivered to an end user, a new authority to run token will be created. When the VM is returned, this token is also returned and is able to be used on the new VM authorization to run request.

Note: Login-based license management uses the following hierarchy when determining ATR duration.

1. **ATR Service** – Establishes the minimum (4 hours/14400 seconds) and maximum (90 days/7776000 seconds) ATR durations applicable to all users/installations. It specifies the default ATR duration (14 days/1209600 seconds) if nothing is specified by Tableau Server or Tableau Desktop.
2. **Tableau Server** - Can optionally specify a maximum or default ATR duration (`licensing.login_based_license_management.max_requested_duration_seconds` or `licensing.login_based_license_management.default_requested_duration_seconds`) for all Tableau Desktop installations. Using these ATR duration settings, you can globally set the default ATR duration and maximum ATR duration for all Tableau Desktop clients, which eliminates the need to individually sign in to each Tableau Desktop client to set the ATR duration. The ATR Service maximum setting can be between the minimum of 4 hours (14400 seconds) and maximum of 90 days (7776000 seconds).
3. **Tableau Desktop** – Can optionally specify the ATR duration (`ATRREQUESTEDDURATIONSECONDS`) for the computer on which it is installed. If necessary, you can change the default ATR duration (14 days/1209600 seconds) to a setting

within the ATR Service minimum (4 hours/14400 seconds) and maximum (90 days/7776000 seconds). This local ATR duration overrides any durations set by the ATR Service or Tableau Server. However, this default ATR duration cannot be more than the maximum ATR duration set on Tableau Server.

Change the ATR duration for Tableau Desktop or Tableau Prep Builder using Tableau Server

On Tableau Server, you use the Tableau Services Manager (TSM) command line utility to set the ATR duration.

To set the ATR duration, at a command prompt, type:

```
tsm configuration set -k licensing.login_based_license_man-
agement.default_requested_duration_seconds -v <value in seconds>

tsm pending-changes apply
```

To set the ATR maximum duration, at a command prompt, type:

```
tsm configuration set -k licensing.login_based_license_man-
agement.max_requested_duration_seconds -v <value in seconds>

tsm pending-changes apply
```

Change the ATR duration on Tableau Desktop by editing the registry

To use a duration length other than the default of 14 days/1209600 seconds, update the `ATRRequestedDurationSeconds` registry setting. For example:

1. As an administrator on the computer running Tableau Desktop, make a backup of the registry file before you make any changes to it.

2. Edit the registry, and in `HKEY_LOCAL_MACHINE\SOFTWARE\Tableau\ATR`, update the ATRRequestedDurationSeconds as follows (0 uses the default setting):

    a. Name: Find the string value named `ATRRequestedDurationSeconds`.

    b. Data: Update the number of seconds the duration should last. For example, add 43200 to set a duration of 12 hours.

3. Restart Tableau so that the changes take effect.

Change the ATR duration on Tableau Desktop on macOS

To use a duration length other than the default of 14 days/1209600 seconds, include the `ATRRequestedDurationSeconds` preferences setting. For example:

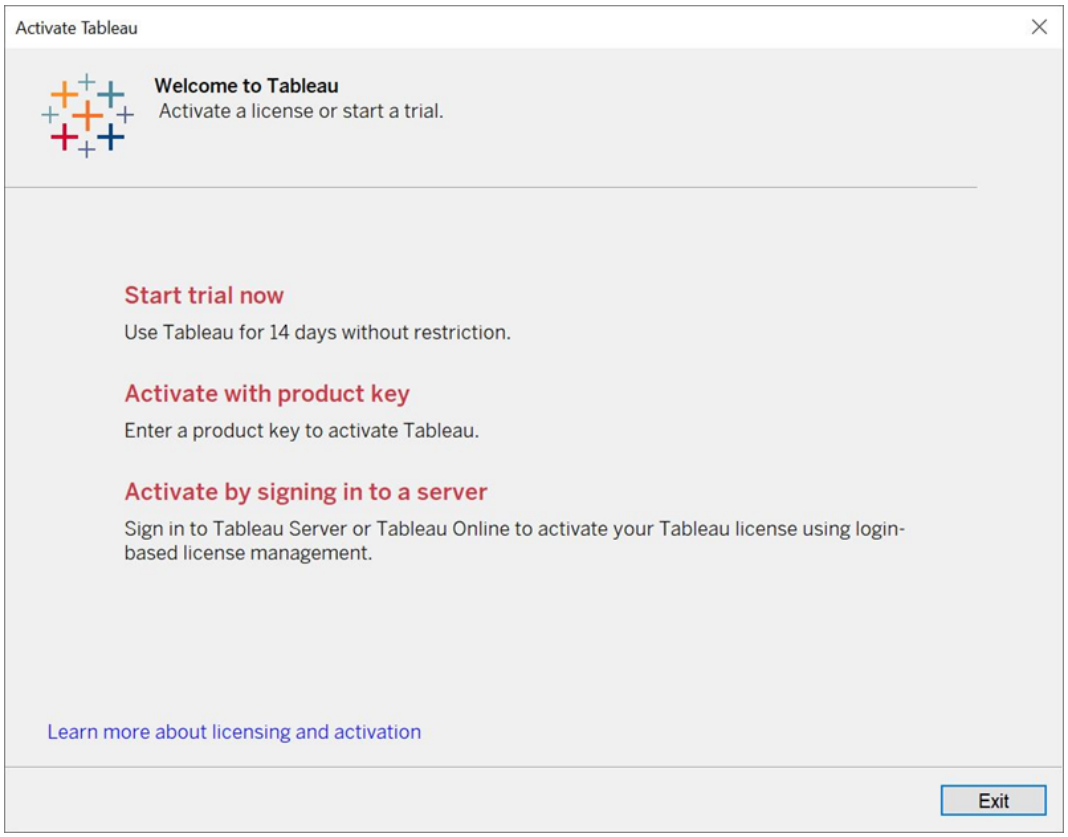
```
sudo defaults write /Library/Preferences/com.tableau.ATR ATRReques-
tedDurationSeconds -string "43200"
```

Step 4: Activate Tableau Desktop

The 2020.1 and later versions of both Tableau Desktop for Windows or macOS support login-based license management.

1. Run Tableau Desktop setup.

2. The Activate Tableau screen will include the **Activate by signing in to a server** option.



3. Click **Activate by signing in to a server** and then do one of the following:

    - If you're using Tableau Cloud, click the Tableau Cloud link.
    - If you're using Tableau Server, enter the Tableau Server URL, and then click **Connect**.

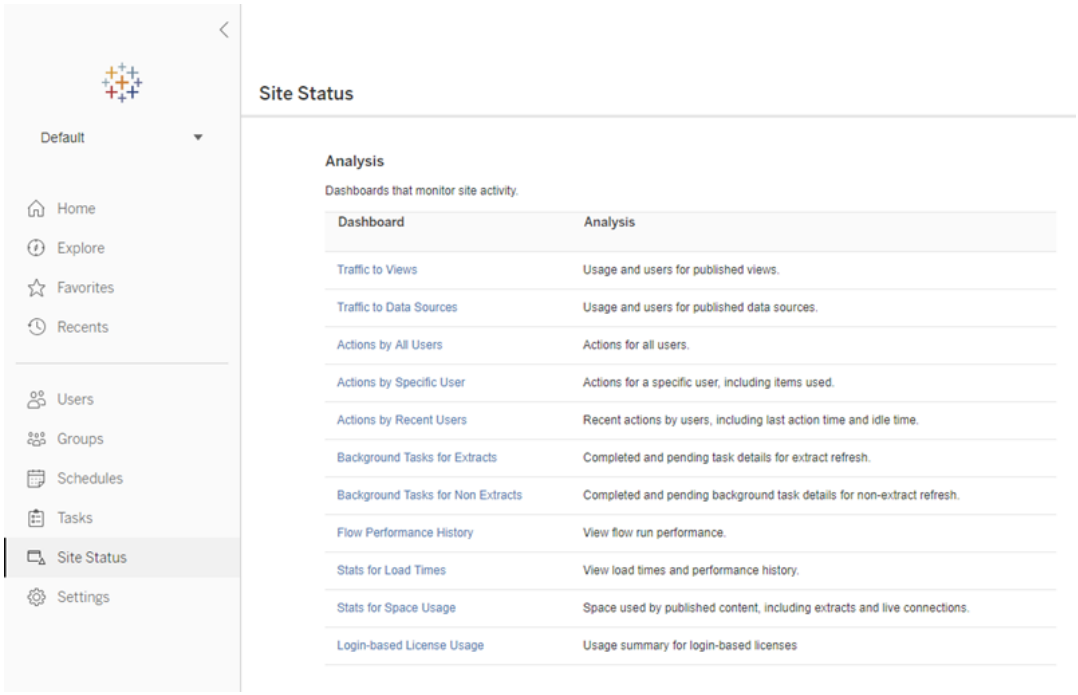
    When prompted, enter valid credentials for a user with a Creator role subscription, and then click **OK**.

View login-based license usage

You can view login-based license usage for Tableau Cloud or Tableau Server. The report shows users, hosts, user role, product, version, activations, Creator seats in use, Creator seats not in use, and when a Creator seat was last used. You can view data for the past 30 days up to a maximum of 183 days.

To view the Login-based license management License Usage administrative view:

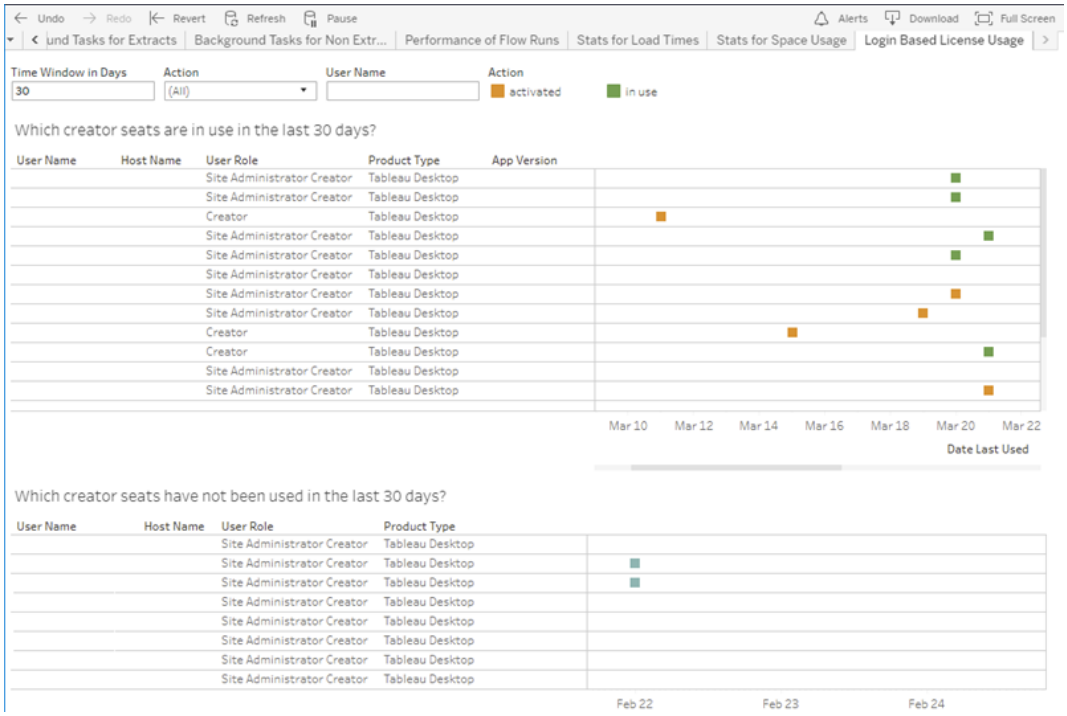
1. In Tableau Cloud or Tableau Server, in the navigation pane, click **Site Status**.



2. On the Site Status page, click **Login-based License Usage**.

3. Optional. On the report screen, you can change the time window to show when seats

were last used, filter on actions, filter on user name, and sort by columns.



Troubleshooting

You may encounter one of the following error messages while using login-based license management. Use the information below to resolve the issue.

Login-based license management is not enabled on Tableau Server

If login-based license management is available as an activation option on Tableau Desktop, but not enabled on Tableau Server, or if Tableau Server does not have a login-based license management capable license, the following error message may be displayed:

Login-based license management is not enabled on the server you connected to. Select a different server to connect to, use a product key to activate your license, or start a trial to begin working immediately.

Contact your administrator to ensure that you're using a Tableau Server product key that's enabled for login-based license management. To verify that you have activated the

appropriate product key, in the TSM web UI click **Configuration**, and then click **Licensing**. For guidance on activating licenses, see tsm licenses activate. Login-based license management should be enabled by default unless it was previously turned off by the server administrator.

Login-based license management is not enabled on Tableau Desktop

If login-based license management is not enabled on Tableau Desktop, you will not have the option to activate Tableau using your credentials. Prior to version 2020.1, the ability to use login-based license management on Tableau Desktop is not turned on as a default. Check the Tableau Desktop version to ensure the correct version is being used.

If the login-based license management option has been turned off during installation or with an update, see Step 2: Enable login-based license management.

Product key expiration date doesn't change after purchasing a year subscription

After purchasing a renewal of Tableau Server, and the new subscription term is reflected in Tableau Server, it may take up to 24 hours before the new expiration date appears in **Manage Product Keys** on Tableau Desktop.

You do not have a Creator license

When attempting to activate from Tableau Desktop, the following error message may be displayed:

You do not have a Creator license. Contact your administrator to obtain one.

This error is displayed if you have not been assigned a Creator role. If you belong to multiple sites on Tableau Cloud or Tableau Server, you need to sign in to the site where you have the Creator role when using login-based license management. Otherwise you'll get this error.

To verify that the product key you have activated on the server includes Creator licenses, open the TSM web UI and click **Configuration**, and then click **Licensing**.

You have activated the maximum number of computers

When attempting to activate from Tableau Desktop, the following error message may be displayed:

You have activated Tableau the maximum number of times allowed under your account. You must wait for the license activation on another computer to expire before you can activate Tableau again.

This error is displayed when you activated Tableau from multiple computers with the same Creator user credentials and exceeded the maximum number of activations. You must wait until the authorization-to-run (ATR) token expires on one of the existing computers before attempting to activate a new computer. If you are using non-persistent virtual machines (VMs), you can shorten the ATR duration to prevent this error from occurring again.

To shorten the ATR token duration for maximum activation

If you encounter this maximum use error when using a non-persistent virtual deployment, it is possible to shorten the ATR duration to 4 hours (14400) seconds to avoid the error in the future. Alternatively, instead of changing the duration on an individual desktop, you can set the default duration on Tableau Server to affect all users.

The following steps shorten the lease on a computer previously activated with login-based license management that will no longer be used, in order to free up a seat to be activated on a new computer:

1. Open a Command Prompt as an administrator on a Tableau Desktop computer that will no longer be used.
2. Navigate to the Tableau binaries (\bin) directory, using the following command.

   Windows: `cd Program Files\Tableau\Tableau <version>\bin`

   Mac: `cd /Applications/Tableau\ Desktop\ <version>.app/Contents/MacOS`

3. Run the following command to set the duration to 4 hours, in seconds (e.g., 14400).

   Windows: `atrdiag.exe -setDuration 14400`

   Mac: `sudo ./atrdiag -setDuration 14400`

4. Delete the previous ATR token using the following command:

   Windows: `atrdiag.exe -deleteAllATRs`

   Mac: `./atrdiag -deleteAllATRs`

5. Next, overwrite the existing ATR token. Open Tableau Desktop. Tableau displays the "License has Changed" message. Click **Exit** to automatically close and reopen Tableau Desktop.

6. In the registration dialog box, click **Activate**, and then reactivate Tableau Desktop through Tableau Server using login-based license management, which will overwrite the existing token.

7. Close Tableau Desktop and wait for the ATR duration to elapse (e.g., 4 hours) so that the ATR token expires and frees-up a user seat. Do not open Tableau Desktop before the ATR duration has elapsed. Check to make sure the ATR duration has elapsed. The ATR token TTL End should show a date and time in the future (e.g., 4 hours from now).

   Windows: `atrdiag.exe`

   Mac: `./atrdiag`

8. After the ATR token expires and you can successfully sign in to Tableau Server on a new computer.

To return your computer to an unlicensed state

1. Open a Command Prompt as an administrator.

2. Navigate to the Tableau binaries (\bin) directory, using the following command:

```
cd Program Files\Tableau\Tableau <version>\bin
```

3. Run the following command:

   Windows: `atrdiag.exe –deleteAllATRs`

   Mac: `./atrdiag.exe –deleteAllATRs`

Note: This removes only the ATR token from the computer. It does not free-up any of the user seats. The user seat is only freed-up after the deleted ATR token expires.

Your Tableau credentials are invalid

When attempting to activate from Tableau Desktop, the following error message may be displayed:

Your Tableau credentials are invalid. Contact your administrator to reset your account.

This error is displayed when your Tableau license is not recognized. Contact your administrator..

Your computer’s clock is not synchronized to the current time

When attempting to activate from Tableau Desktop, the following error message may be displayed:

Your computer’s clock is not synchronized to the current time. Synchronize your computer’s clock to the current time and then try to activate Tableau.

This error is displayed when your computer's clock is not synchronized with the current time. Synchronize your computer’s clock with a time server on the internet or enable automatic time synchronization.

Unable to activate with your credentials

When attempting to activate Tableau, the following error message may be displayed:

Tableau Server cannot verify your licensing information over the internet. Contact your administrator to check your internet connection.

This error is displayed when the port `atr.licensing.tableau.com:443` is not open on all Application Server (VizPortal) nodes, or you have a proxy that is not configured properly to forward traffic to Tableau's licensing server.

To diagnose connectivity to Tableau's licensing server, paste the following URL (`https://atr.licensing.tableau.com/_status/healthz`) into a browser or at a curl command prompt.

Zero Downtime Licensing

With zero downtime licensing, which was introduced in Tableau Server version 2021.1, you can apply most licensing changes to Tableau Server without needing to restart when license end dates, capacity, or installed features are changed.

When should you restart Tableau Server?

When you apply a license update that does not require a restart, Tableau Server displays the following message: **Updated Licensing details are being applied across Server**. However, there are some situations when applying a license update that require you to restart Tableau Server. For example, if you are activating or deactivating a product key that reduces features or changes the allowed data source connections, you'll need to restart Tableau Server after making your changes. When a restart is required, Tableau Server will display the following message: **Restart Server to apply updated Licensing details**. The following table lists the times when license changes require you to restart Tableau Server.

License change	Restart required?
Extending a license term	No
Adding or reducing user license capacity	No
Adding core license capacity	No

Reducing core license capacity	Yes
Adding a Data Management or Advanced Management license	No
Removing a Data Management or Advanced Management license	Yes
Changing allowed data sources	Yes
Expired product key	Yes

About Tableau Enterprise

Tableau Enterprise is our advanced software package to help you explore and manage data faster with Tableau Server. It also makes it easier to purchase the capabilities needed by organizations that require advanced data and deployment management options. It includes Tableau role-based licenses, Data Management, Advanced Management, and eLearning for Creators and Explorers.

Tableau Enterprise Licensing

Tableau Enterprise is sold on a per-User role-based licensing model, including Creators, Explorers, and Viewers. This model is structured around the specific roles within an organization, each requiring different levels of functionality. With Tableau Enterprise, each role-based license includes Data Management, Advanced Management, and eLearning for Creators and Explorers.

Tableau Enterprise Feature Table

The following table lists the features that are included with the Tableau Enterprise license edition. Feature availability below is noted for Creator. There are differences in available functionality between Creator, Explorer, and Viewer.

Feature	Description
Tableau Authoring	Use Tableau Desktop or Tableau web authoring on Tableau Server to create, col-

	laborate, and share insights about your data. Tableau provides you a way to identify and solve problems, or highlight key findings in a visual and easily understandable way.
Tableau Prep	Tableau Prep is a data preparation tool for cleaning, shaping, and combining data for analysis.
eLearning	Role-based training Tableau Learning Paths provide a clear track to proficiency with the most up-to-date Tableau training content. Assessments help you evaluate where you are in your learning path and give you confidence in your new skills. Accelerate the onboarding process for new employees and help more experienced users get the most out of Tableau capabilities.
About Data Management	Data Management is a collection of features and capabilities that helps customers manage Tableau content and data assets in their Tableau Server environment.
About Tableau Advanced Management on Tableau Server	Advanced Management is a collection of features and capabilities designed to provide enhanced security, manageability, and scalability for Tableau Server.

About the Identity Migration

Beginning in version 2022.1, Tableau Server stores and manages identity information using the Identity Service. With the Identity Service, Tableau Server uses a more modern, more

secure, and immutable identity structure for the user provisioning and authentication process. Identity migration is a prerequisite for configuring and using identity pools.

Note: If you do not plan to use the identity pools capability, we recommend you do not run the identity migration. Running the identity migration without plans to use identity pools will not provide benefits to your Tableau Server deployment.

All new deployments of Tableau Server 2022.1 (and later) use the Identity Service by default and require no additional action from you. As you add new users to Tableau Server, the default Identity Service is used.

For existing deployments, if upgrading Tableau Server to version 2022.1 (or later) and restoring a backup of Tableau 2021.4 (or earlier), you can start the identity migration after Tableau Server upgrade completes to populate the new Identity Service. The identity migration populates supplemental Identity Service tables for all Tableau Server users, which are then used to authenticate users through the Identity Service. The migration runs in the background and won't interrupt or interfere with your users' use of Tableau Server.

As an administrator, you can monitor and manage the migration, including changing when the migration runs or resolving any potential migration conflicts, through a dedicated **Identity Migration** page available from Tableau Server's Users page. This page is available for the duration of the migration process.

Summary of steps for existing deployments

For existing deployments, you must configure Tableau Server to use the Identity Service after the migration completes to take advantage of the identity structure improvements and configure identity pools.

Step 1: Before you begin

Step 2: Start the identity migration

Step 3: Complete the identity migration

Step 4: Configure Tableau Server to use the Identity Service

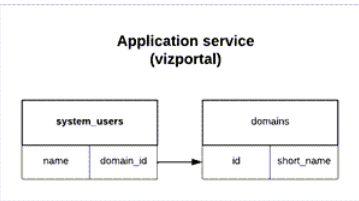
Key terms

- **Identity Service** - a service in Tableau Server 2022.1 (and later) that is responsible for the administration of user identities, including authentication and provisioning. The service uses an identity schema where user identities are represented by Identity Service tables and the legacy "system_users" table.
- **Identity pools** - an identity management tool that uses provisioning and authentication information to enable user access to Tableau Server. Identity pools enable a more centralized and flexible identity management workflow built on the Identity Service for the storage and management of user identities in Tableau Server.
- **Legacy identity store mode** - a limited identity schema used by Tableau Server 2021.4 (and earlier), where user identities are only represented by the legacy "system_users" table.
- **Identity migration** - the auditing process that evaluates existing Tableau Server user identities, queries the upstream external identity stores for additional identity information (as appropriate), and imports that additional identity information to the Identity Service.
- **External identity store** - an identity store type external and upstream to Tableau Server where all identity information is stored and managed by an external directory service (Active Directory (AD) or LDAP). If configured, Tableau Server synchronizes to the external directory so that a copy of the identity information exists in Tableau Server.
- **Local identity store** - an identity store type provided by Tableau Server. If configured, Tableau Server stores and manages identity information in the Tableau Server repository without any configured external directory for this information.
- **System user** - a Tableau Server user. A user corresponds to a signin record ("system_users") in both the Identity Service (through the "system_users_identities" table) and legacy identity store mode. A "system_users" record can potentially have multiple user identities associated with it and enabled to sign in to multiple sites. The link between a "system_users" record and sites is defined in the "users" table.

Purpose of the identity migration

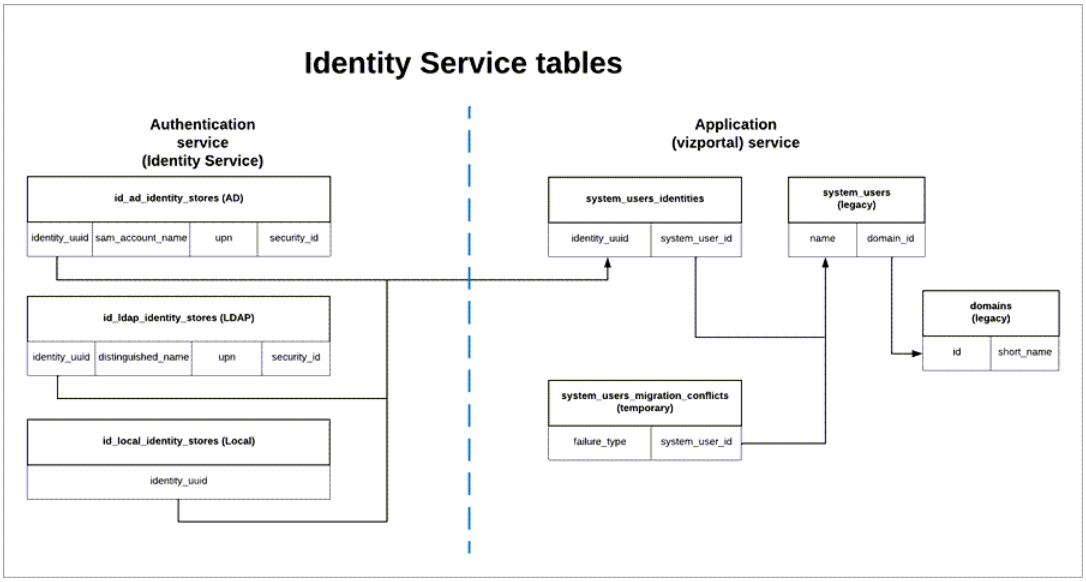
When you create a Tableau Server backup, identity information is saved in the identity schema used by the version of Tableau Server the backup was created for. The migration is necessary to populate identity information from the identity schema used in the backup to the identity schema used by the Identity Service.

Identity schema of Tableau Server 2021.4 and earlier



The identity schema used by the legacy identity store mode consists of two tables, "system_users" and "domains."

Identity schema of Tableau Server 2022.1 and later



The identity schema used by the Identity Service includes the legacy "system_users" tables and supplemental Identity Service tables (*_identity_stores and *identities) that capture more identity information. The additional tables help reduce issues that can be caused by upstream changes in the external identity stores.

What happens during the identity migration

When information about user identities are migrated, identity information stored in the legacy "system_users" table are supplemented with the Identity Service tables.

The type of Identity Service tables that the identity information is supplemented with depends on which identity store type Tableau Server is configured for: local, Active Directory (AD), or Lightweight Directory Access Protocol (LDAP).

- For **AD** identity store types, Identity Service tables only inherit unambiguous attributes or attributes that are not stored in the same database record.

  For example, sAMAccountNAme and userPrincipalName (UPN) can be stored in the same name record of a legacy "systems_users" table, which can occur as a result of a complex series of rules. In most cases, the migration is able to correctly interpret and successfully migrate the user identity. However, if the migration produces ambiguous results, you must either manually acknowledge the ambiguity or manually resolve the conflict using the dedicated Identity Migration page. For more information, see Resolve Identity Migration Conflicts.

- For **LDAP** identity store types, like AD identity store types, Identity Service tables only inherit unambiguous attributes. In most cases, the migration is able to correctly interpret and successfully migrate the user identity. However, if the migration produces ambiguous results, you must either manually acknowledge the ambiguity or manually resolve the conflict using the dedicated Identity Migration page. For more information, see Resolve Identity Migration Conflicts.

- For **Local** identity store types, Identity Service tables inherit the user and domain fields directly. This means, no additional information or manual resolution is required from you. When Tableau Server is configured for this type of identity store, migration of users identities happens after the Tableau Server backup restore process.

Step 1: Before you begin

Before you begin, identify your Tableau Server upgrade method below to determine next steps in the identity migration.

- If you're performing a **Blue/Green upgrade** or **manually upgrading** Tableau Server by 1) installing Tableau Server on a new machine and then 2) **backing up and restoring**

Tableau Server using the tsm maintenance (backup and restore) **commands**, you're required to take some additional steps to initiate the migration.

For next steps, see Troubleshoot Issues with the Identity Migration.

- If you're doing an **"in-place" single-server or multi-node upgrade** of Tableau Server using the method described here, there are no additional steps required from you to initiate the migration. The migration initiates after Tableau Server upgrade to version 2022.1 (or later) is complete.

  Skip to Step 2.

- If you're **manually upgrading** Tableau Server by 1) installing Tableau Server on a new machine and then 2) **exporting and importing configuration and topology information using** tsm settings (export and import) **commands**, there are also no additional steps required from you to initiate the migration. The migration initiates after the import process completes on the new Tableau Server machine.

  Skip to Step 2.

Step 2: Start the identity migration

To start the identity migration, you must enable the identity migration capability by using the tsm command features.IdentityMigrationBackgroundJob.

Note: If you've upgraded to Tableau Server versions 2021.4.21, 2022.1.17, 2022.3.9, and 2023.1.5, the identity migration starts by default and you can skip to Step 3: Complete the identity migration.

1. Open a command prompt as admin on the initial node (where TSM is installed) in the cluster.
2. Run the command the following command:

```
tsm configuration set -k fea-
tures.IdentityMigrationBackgroundJob -v true
```

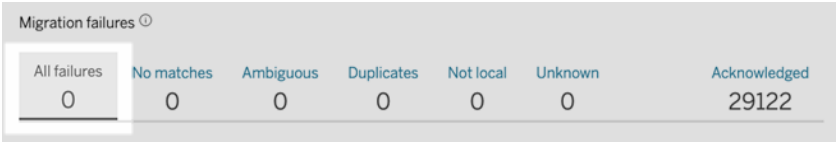
After the identity migration begins, you'll see a notification in Tableau Server that links you to the Identity Migration page. The Identity Migration page is where you can monitor the status of the identity migration and identity conflicts that need to be resolved.

Step 3: Complete the identity migration

To complete the identity migration, all identity conflicts must be resolved or acknowledged before you can enable the Identity Service for Tableau Server.

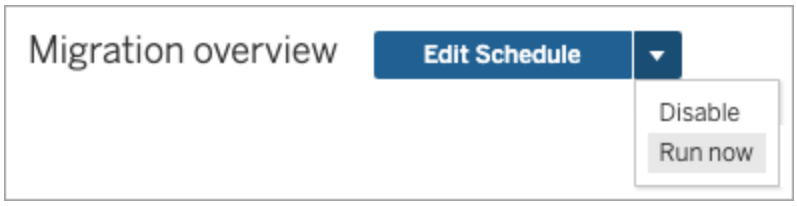
1. Sign in to Tableau Server as an administrator.
2. From the left navigation pane, select **Users** (or **All Sites** > **Users** for a multi-site Tableau Server) and then click the **Identity Migration** page to verify the migration has started.

   You can monitor and manage its progress using the dedicated Identity Migration page available from Tableau Server's Users page. For more information, see Manage the Identity Migration.
3. Resolve or acknowledge all identity conflicts as described in Resolve Identity Migration Conflicts so that **All failures** tab displays "0" like in the image below.



4. Do *one* of the following:

   - To run the identity migration job now, next to the Migration Overview heading, click the Edit Schedule drop-down arrow, and then select **Run Now**.



- Alternatively, you can wait for the migration job to run during the next scheduled time.

5. After the migration completes, from the Identity Migration page, validate that the Migration Overview shows **100% complete**.

Step 4: Configure Tableau Server to use the Identity Service

After the identity migration is complete, configure Tableau Server to use the Identity Service to ensure a more secure and immutable identity structure for the user provisioning and authentication process.

1. Open a command prompt as an administrator on the initial node (where TSM is installed) in the cluster.
2. Run the following commands:

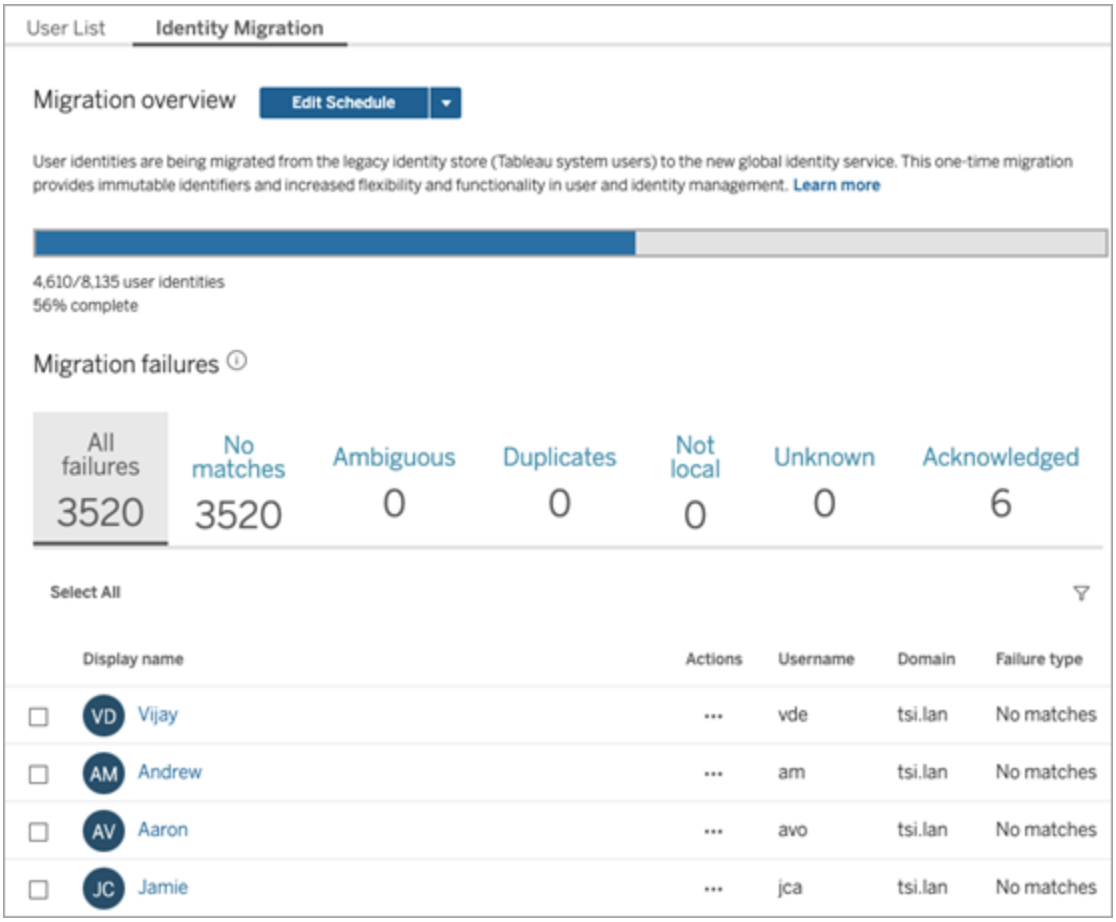
```
tsm authentication legacy-identity-mode disable
tsm pending-changes apply
```

Note: After running the commands above, the dedicated **Identity Migration** page is removed and no longer accessible. The page is accessible only when `tsm authentication legacy-identity-mode` is enabled.

After Tableau Server is configured to use the Identity Service, when users sign in to Tableau Server, Tableau Server searches for their user identities using their identifiers in the configured identity store. From the identifiers, the universal unique identifiers (UUID) are returned and used to match existing Tableau Server user identities. This process then generates sessions for the users and completes the authentication workflow.

Manage the Identity Migration

As an administrator, you can monitor and manage the identity migration, including changing when the migration jobs run, through the dedicated Identity Migration page available from Tableau Server's Users page. This page is available for the duration of the migration process.



The migration jobs are designed to run in the background without interrupting or interfering with the use of Tableau Server. If needed, however, you can make adjustments that affect how frequently migration jobs run, when the migration jobs run, and how long the migration jobs can run.

Generally, the migration can take anywhere from 3 minutes to 10 days, depending on the size of your Tableau Server deployment and any changes to the default settings you make during

the migration. For example, if you have 10,000 users, the migration can take about 30 minutes.

Note: While the migration jobs are running, all authentication and user-related capabilities work normally.

Manage identity migration jobs

You can manage the following aspects of the identity migration.

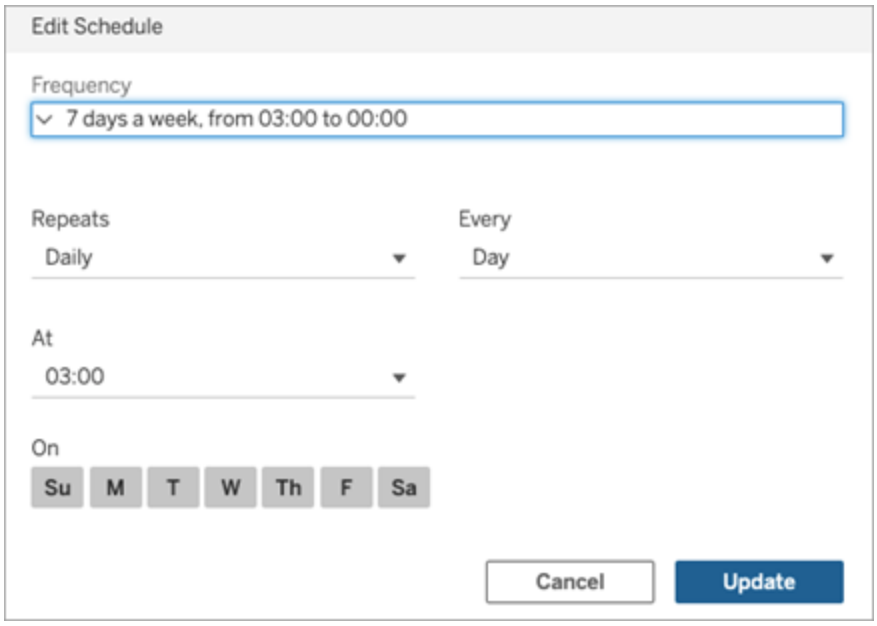
Resolve identity conflicts

To review the identity conflicts you might encounter during the migration, see Resolve Identity Migration Conflicts.

Change daily migration job schedule

1. Sign in to Tableau Server as an administrator.
2. From the left navigation pane, select **Users** (or **All Sites** > **Users** for a multi-site Tableau Server) and then click the **Identity Migration** page.
3. Next to the Migration Overview heading, click the **Edit Schedule** button.
4. In the Edit Schedule dialog box, change when and how frequently jobs can run.

   Note: You can ignore the **Priority** and **Execution** options in this dialog box.



5. When finished, click **Update**.

Initiate a migration job

1. Sign in to Tableau Server as an administrator.
2. From the left navigation pane, select **Users** (or **All Sites** > **Users** for a multi-site Tableau Server) and then click the **Identity Migration** page.
3. From any tab, select a user or multiple users.
4. From the Actions menu, select **Retry Migration** or **Acknowledge**, depending on what you need to do.

5. Next to the Migration Overview heading, click the Edit Schedule drop-down arrow.

6. Select **Run Now**.

Pause the identity migration

1. Sign in to Tableau Server as an administrator.
2. From the left navigation pane, select **Users** (or **All Sites** > **Users** for a multi-site Tableau Server) and then click the **Identity Migration** page.
3. Next to the Migration Overview heading, click the Edit Schedule drop-down arrow.
4. Select **Disable**.

Restart the identity migration

1. Sign in to Tableau Server as an administrator.
2. From the left navigation pane, select **Users** (or **All Sites** > **Users** for a multi-site Tableau Server) and then click the **Identity Migration** page.
3. Next to the Migration Overview heading, click the Edit Schedule drop-down arrow.
4. Select **Enable**.

Change identity migration settings

To reduce any potential strain the identity migration might have on your Tableau Server, the migration is configured to run with the default settings listed below.

Migration settings

Type	tsm command	Default	Procedure
Jobs schedule	N/A	3:00 AM, daily until complete	Configurable from the Identity Migration page, see Change daily migration job schedule above.
User identity requests per second (rate)	`tsm authentication identity-migration configure --rate`	Up to 5	If needed, you can use the `tsm authentication identity-migration` command to change the migration settings listed above using the steps below. 1. Open a command prompt as an admin on the initial node (where TSM is installed) in the cluster. 2. Run one or both of the commands described in `tsm authentication identity-migration`. For example, to change the individual job runtime and rate from their default values, you can run the following command: `tsm authentication identity-migration configure --job-run-`

			`time 180 --rate 3`
Individual job runtime	`tsm authentication identity-migration configure --job-run-time`	120 minutes	
Enable identity migration	`tsm configuration set -k features.IdentityMigrationBackgroundJob`	`false`	By enabling the identity migration, Tableau Server can use the Identity Service to store and manage user identity information. 1. Open a command prompt as admin on the initial node (where TSM is installed) in the cluster. 2. Run the command the following command: `tsm configuration set -k features.IdentityMigrationBackgroundJob -v true` **Note:** The identity migration and the Identity Service is a prerequisite for certain capabilities like identity pools. For more information about the tsm command, see features.Iden-

			tityMigrationBackgroundJob.

Disable identity migration

If you've upgraded to Tableau Server versions 2021.4.21, 2022.1.17, 2022.3.9, and 2023.1.5, you might need to disable the identity migration. By disabling the identity migration, Tableau Server cannot use the Identity Service to store and manage user identity information.

1. Open a command prompt as admin on the initial node (where TSM is installed) in the cluster.
2. Run the following command:

```
tsm configuration set -k fea-
tures.IdentityMigrationBackgroundJob -v false
```

Note: The identity migration and the Identity Service is a prerequisite for certain capabilities like identity pools.

Complete the identity migration and configure the Identity Service

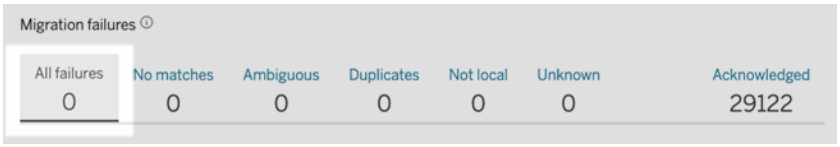
After all user conflicts are resolved or addressed and migration jobs run, you must configure Tableau Server to use the Identity Service to complete the identity migration process.

Step 1: Validate and complete the identity migration

1. Sign in to Tableau Server as an administrator.
2. From the left navigation pane, select **Users** (or **All Sites** > **Users** for a multi-site Tableau Server) and then click the **Identity Migration** page to verify the migration has started.

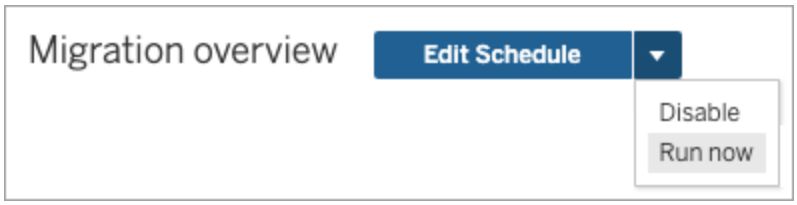
   You can monitor and manage its progress using the dedicated Identity Migration page available from Tableau Server's Users page. For more information, see Manage the Identity Migration.

3. Resolve or acknowledge all identity conflicts as described in Resolve Identity Migration Conflicts so that **All failures** tab displays "0" like in the image below.



4. Do *one* of the following:

   - To run the identity migration job now, next to the Migration Overview heading, click the Edit Schedule drop-down arrow, and then select **Run Now**.



   - Alternatively, you can wait for the migration job to run during the next scheduled time.

5. After the migration completes, from the Identity Migration page, validate that the Migration Overview shows **100% complete**.

Step 2: Configure Tableau Server to use the Identity Service

1. Open a command prompt as an administrator on the initial node (where TSM is installed) in the cluster.

2. Run the following commands:

```
tsm authentication legacy-identity-mode disable
tsm pending-changes apply
```

> **Note:** After running the commands above, the dedicated **Identity Migration** page is removed and no longer accessible. The page is accessible only when `tsm authentication legacy-identity-mode` is enabled.

After Tableau Server is configured to use the Identity Service, when users sign in to Tableau Server, Tableau Server searches for their user identities using their identifiers in the configured identity store. From the identifiers, the universal unique identifiers (UUID) are returned and used to match existing Tableau Server user identities. This process then generates sessions for the users and completes the authentication workflow.

Resolve Identity Migration Conflicts

During the identity migration, Tableau Server might encounter certain user identities that can't be migrated to use the Identity Service. When user identities can't be migrated, they become identity conflicts that require you, the administrator, to manually resolve.

To ensure user identities are migrated correctly, you must resolve or address all identity conflicts before the identity migration can complete using the dedicated **Identity Migration** page.

Step 1: Resolve identity conflicts

You can resolve identity conflicts in a few ways depending on the conflict type. Regardless of the conflict type, all user identities must be resolved or addressed before you can proceed to Step 2 below and before the identity migration process can complete.

When identity conflicts occur, the identity migration will group the conflicts into types. These types help narrow down the reason why the migration is unable to migrate the user identity automatically.

There are a few reasons why identity conflicts can occur. For example, you might see an identity conflict when the migration has identified a Tableau Server user that matches more than one user identity in the external identity store.

When identity conflicts are identified, you can address them using one of the following options:

- **Retry Migration** - This option moves the selected user identities back into the queue to be migrated again. After the migration jobs run again, it's possible the identity conflicts resolve on their own, the original identity conflicts occur again, or new identity conflicts occur.
- **Acknowledge** - This option moves the selected user identities to the **Acknowledged** tab. When you acknowledge user identities, you understand that 1) those users don't have matching user identities in an identity store and will therefore not be migrated, and 2) those users will be unable to sign in to Tableau Server after you enable the Identity Service in Step 3 below.
- **Reevaluate** - When conflicts have already been acknowledged, from the **Acknowledged** tab, this option moves the selected user identities back to their conflict state. This option gives you the chance to see the original conflict, resolve the conflict, or acknowledge the identity conflict again.

Quick reference: Identity conflicts

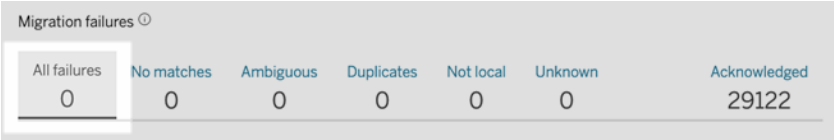
Conflict type	Applies to configuration	Conflict reason	Action
All failures	All	This tab captures all identity conflicts categorized in the No matches, Ambiguous, Duplicate, Not local, and Unknown tabs.	Retry Migration or Acknowledge
No matches	AD, LDAP	The users identities have no matching users in the external identity store.	Retry Migration or Acknowledge
Ambiguous	AD, LDAP	For the specified user identities, there is more than one possible match in the external identity store.	Retry Migration, Acknowledge, or select one of the suggested user identities

Duplicate	AD	Two user identities were created using one AD account. This is an artifact of legacy functionality that is not supported in the Identity Service.	Retry Migration or Acknowledge
Not local	Local	User identities that are associated with an identity store that is not local. This conflict occurs because manual changes were made that are not supported.	Retry Migration or Acknowledge
Unknown	All	This conflict might indicate an internal Tableau Server error or an identity conflict caused by a reason not listed in this table.	Retry Migration or Acknowledge
Acknowledged	All	This tab captures all user identities that will not be migrated. Those users will not be able to sign in to Tableau Server after Tableau Server is configured to use the Identity Service.	Retry Migration or Acknowledge

To resolve a conflict, follow the steps below.

1. Sign in to Tableau Server as an administrator.
2. From the left navigation pane, select **Users** (or **All Sites** > **Users** for a multi-site Tableau Server) and then click the **Identity Migration** page.
3. Select one or more user identities in the **All Failures** tab or in one of the conflict-specific tabs.
4. From the Actions drop-down menu, click **Retry Migration** or **Acknowledge**.

   If you select “Retry Migration,” the user identities might generate different conflict types. In this case, address the conflicts as needed until the **All failures** tab displays "0" like in the image below.



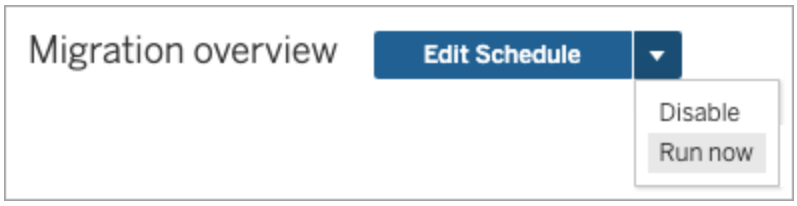
   Note: User identities that are acknowledged are ignored in Active Directory (AD) and LDAP group syncs moving forward. If users associated with these users identities require access to Tableau Server at a later time, see Troubleshoot Issues with the Identity Migration for more information.

Step 2: Complete the identity migration

To complete the identity migration, in addition to all identity conflicts being resolved or addressed, all migration jobs have to run before you can enable the Identity Service for Tableau Server.

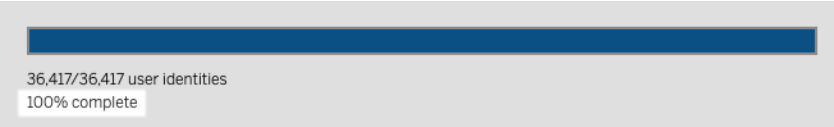
1. Do *one* of the following:

   - To run the identity migration jobs now, next to the Migration Overview heading, click the Edit Schedule drop-down arrow, and then select **Run Now**.



- Alternatively, you can wait for the migration to run during the next scheduled time.

2. From the Identity Migration page, validate that the Migration Overview displays **100% complete**.



Step 3: Configure Tableau Server to use the Identity Service

After the identity migration is complete, configure Tableau Server to use the Identity Service to ensure a more secure and immutable identity structure for the user provisioning and authentication process.

1. Open a command prompt as an administrator on the initial node (where TSM is installed) in the cluster.

2. Run the following commands:

```
tsm authentication legacy-identity-mode disable
tsm pending-changes apply
```

Note: After running the commands above, the dedicated **Identity Migration** page is removed and no longer accessible. The page is accessible only when `tsm authentication legacy-identity-mode` is enabled.

After Tableau Server is configured to use the Identity Service, when users sign in to Tableau Server, Tableau Server searches for their user identities using their identifiers in the configured identity store. From the identifiers, the universal unique identifiers (UUID) are returned

and used to match existing Tableau Server user identities. This process then generates sessions for the users and completes the authentication workflow.

Troubleshoot Issues with the Identity Migration

Unable to restore backup

After upgrading to Tableau Server 2022.1 (or later), restoring a Tableau Server backup might cause the following error:

"The backup cannot be restored because Tableau Server uses the new identity service tables by default."

This issue might occur when Tableau Server needs to run the identity migration, which is a necessary process to populate the Identity Service. The Identity Service is an identity schema that was introduced beginning with Tableau Server 2022.1 and is used to provision and authenticate users. To prevent any potential issues, the restore process can't proceed when Tableau Server detects that the Tableau Server backup uses a different identity schema than the version that it's being restored to.

Note: The Identity Service is the default identity schema in Tableau Server version 2022.1-2022.1.7, 2022.3-2022.3.9, and 2023.1-2023.15.

To resolve this issue, follow the steps described below.

Step 1: Enable `legacy-identity-mode` and restore the backup

1. Open a command prompt as an administrator on the initial node (where TSM is installed) in the cluster.
2. Set Tableau Server 2022.1 (or later) to use the legacy identity store mode by running the following commands:

```
tsm authentication legacy-identity-mode enable
tsm pending-changes apply
```

Tableau Server must use the legacy identity store mode to populate the Identity Service. For more information about the tsm command, see `tsm authentication legacy-identity-mode`.

3. Restore the backup again to enable the migration to initiate by running the following commands:

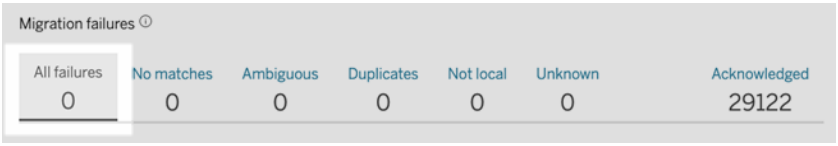
```
tsm maintenance restore --file <file_name>
tsm start
```

Important: After the backup is restored, the migration populates the Identity Service with identity information. For general information about restoring from backup, see Restore from a Backup.

Step 2: Validate and complete the identity migration

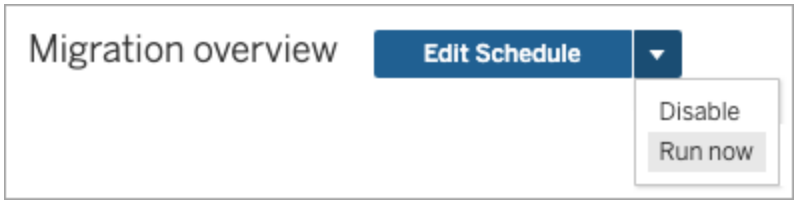
1. Sign in to Tableau Server as an administrator.
2. From the left navigation pane, select **Users** (or **All Sites** > **Users** for a multi-site Tableau Server) and then click the **Identity Migration** page to verify the migration has started.

   You can monitor and manage its progress using the dedicated Identity Migration page available from Tableau Server's Users page. For more information, see Manage the Identity Migration.
3. Resolve or acknowledge all identity conflicts as described in Resolve Identity Migration Conflicts so that **All failures** tab displays "0" like in the image below.



4. Do *one* of the following:

- To run the identity migration job now, next to the Migration Overview heading, click the Edit Schedule drop-down arrow, and then select **Run Now**.



- Alternatively, you can wait for the migration job to run during the next scheduled time.

5. After the migration completes, from the Identity Migration page, validate that the Migration Overview shows **100% complete**.

Step 3: Configure Tableau Server to use the Identity Service

1. Open a command prompt as an administrator on the initial node (where TSM is installed) in the cluster.

2. Run the following commands:

```
tsm authentication legacy-identity-mode disable
tsm pending-changes apply
```

Note: After running the commands above, the dedicated **Identity Migration** page is removed and no longer accessible. The page is accessible only when `tsm authentication legacy-identity-mode` is enabled.

After Tableau Server is configured to use the Identity Service, when users sign in to Tableau Server, Tableau Server searches for their user identities using their identifiers in the configured identity store. From the identifiers, the universal unique identifiers (UUID) are returned

and used to match existing Tableau Server user identities. This process then generates sessions for the users and completes the authentication workflow.

“Unexpected error” on Identity Migration page

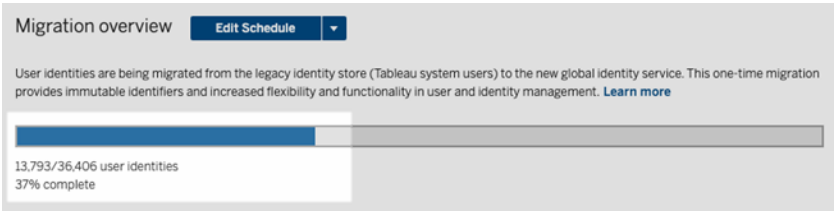
After resolving or acknowledging all user identities from the Identity Migration page, you see an “Unexpected error” message. This message can display when you’ve tried to resolve or acknowledge more than 1000 user identities at one time.

To resolve this issue, select and resolve or acknowledge 1000 or less user identities, and then try again.

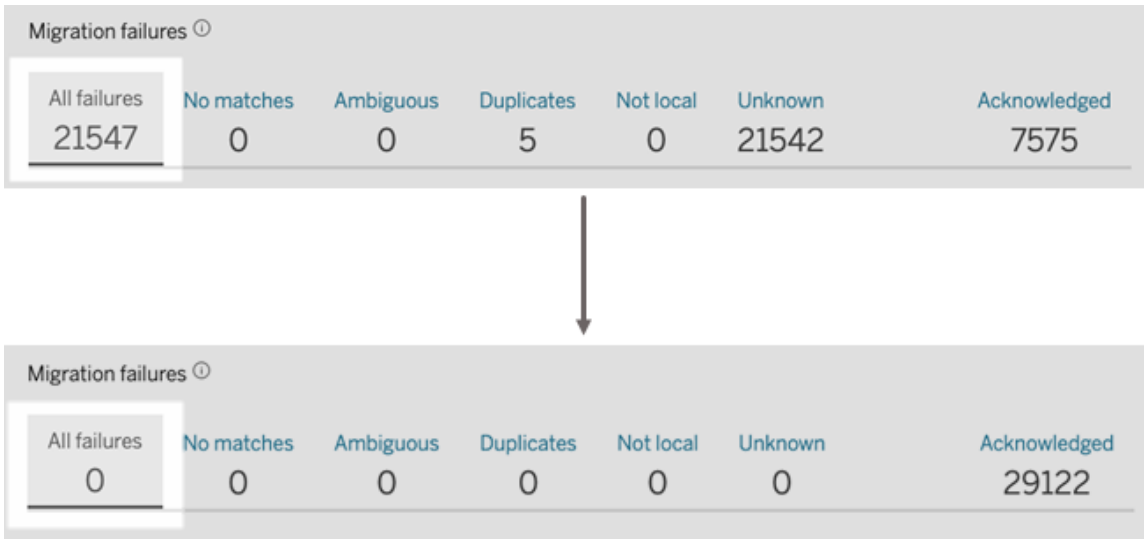
For more information about managing identity conflicts, see Resolve Identity Migration Conflicts.

Migration progress appears unresponsive or stuck

If the migration status or migration progress bar appears unresponsive or stuck, validate that you have resolved and acknowledged all user conflicts under **Migration failures**.



To resolve this issue, ensure all conflicts are resolved and acknowledged by selecting one or more users identities in the **All failures** tab and clicking **Acknowledge** from the Actions drop-down menu. Perform this task until the **All failures** tab displays "0." For more information about managing identity conflicts, see Resolve Identity Migration Conflicts.



Note: After all identity conflicts have been resolved or addressed, all migration jobs have to run before you can enable the Identity Service for Tableau Server. You can run the migration jobs now by clicking the Edit Schedule drop-down arrow next to the Migration overview heading, and then selecting **Run Now**. When the Migration Overview shows **100% complete**, you can configure Tableau Server to use the Identity Service. For more information, see Step 3: Complete the identity migration.

"Identity migration is in progress" pop-up persists

The "identity migration is in progress" notification persists despite completing the identity migration because the Identity Service has not been enabled yet. To complete the identity migration the Identity Service needs to be enabled in Step 4: Configure Tableau Server to use the Identity Service so that Tableau Server can use the identity structure that enables identity pools capability.

Identity Migration page disappears

When the identity migration is complete and Tableau Server is configured to use the Identity Service, the dedicated **Identity Migration** page is removed and no longer accessible. The Identity Migration page is needed only for the identity migration or when `tsm authentication legacy-identity-mode` is enabled.

Users can't sign in

After the identity migration has completed and the Identity Service enabled, some users are unable to sign in to Tableau Server. In most cases, this issue occurs to users whose identities had conflicts and were subsequently **Acknowledged** during the identity migration. Users identities that were acknowledged are not migrated to the Identity Service and subsequently ignored during Active Directory (AD) or LDAP group syncs going forward.

If users associated with those acknowledged user identities require access to Tableau Server again, manually add the users to Tableau Server. After the users are manually added, subsequent AD or LDAP group syncs recognize the user identities and sync as expected.

Revert identity migration

If there are issues, such as certain users not being able to sign in to Tableau Server, that you believe are caused by the Identity Service, you can use the `tsm authentication legacy-identity-mode` command to revert back to use the legacy identity store mode. After reverted, both new users who were added after the identity migration and users who could only sign in to Tableau Server before the migration are able to sign in to Tableau Server without any issues.

After reverting from the Identity Service to the legacy identity store mode, you can use the Identity Migration page to run the migration for the problematic user identities. For more information about managing identity conflicts, see Resolve Identity Migration Conflicts.

Provision and Authenticate Users Using Identity Pools

Introduced in Tableau Server version 2023.1, identity pools are an identity management tool that uses provisioning and authentication information to enable user access to Tableau Server. Identity pools enable a more centralized and flexible identity management workflow built on the Identity Service for the storage and management of user identities in Tableau Server.

Identity pools do not replace the user provisioning and authentication configurations you make using Tableau Services Manager (TSM) during Tableau Server setup. Instead, identity pools are designed to complement and support additional user provisioning and authentication options you might need in your organization, particularly for organizations where TSM is configured with Active Directory (AD) or Lightweight Directory Access Protocol (LDAP). Identity pools add an alternative method, after Tableau Server setup, that supports Tableau Server administrators to add users, who are often external users, partners, or contractors, to your Tableau Server deployment.

Identity pools are optimized for the following use cases:

- **External users:** A large enterprise organization who doesn't want to add external users to their internal AD.

  For example, suppose your organization has two types of employees: regular employees and contract employees. Your regular employees are provisioned through Active Directory (AD) with SAML authentication that is managed through your IdP, Okta. Your contract employees consist of users who are typically assigned temporary group membership or are a part of another organization who provisions users outside of AD and authenticates separately. Identity pools can enable you to add Tableau Server users who are external to your AD.

- **Multiple identity stores:** An organization hosting SaaS applications that sources users from multiple identity stores.

  For example, suppose your organization shares Tableau content to multiple external organizations from a single site. You can separate these users using different identity pools configured with local identity stores to more easily identify and manage the users from each organization.

- **Security boundaries between internal organizations:** An organization of multiple acquired child organizations with distinct security boundaries.

For example, you can add users from the newly added organization to an identity pool configured with a local identity store to workaround the complexities associated with combining identity stores.

What are identity pools?

An identity pool has three main components: an identity store to provision users, OpenID Connect (OIDC) authentication, and assigned users.

- **Identity store:** The identity store that you source or provision your users can be a local identity store or an external identity store.
    - If a local identity store, an identity pool can be configured to use a new local identity store or an existing local identity store. **Note:** Local authentication is not supported.
    - If an external identity store, an identity pool can only use the same external identity store (AD or LDAP) that you configured in TSM during Tableau Server setup. You can't configure an identity pool to use a different external identity store.

    The provisioning and authentication configurations you make in TSM during Tableau Server setup is referred to as the default or “initial pool (TSM configured).”
- **Authentication:** The only supported authentication method for an identity pool is OIDC.
- **Users:** In order for users to sign in to Tableau Server, they must either be sourced from the initial pool (TSM configured) or be a member of at least one identity pool.

When to use identity pools

As a Tableau Server administrator, you can use an identity pool to segment your users into identity cohorts based on where your users are provisioned from and how those users authenticate into Tableau Server. Though the identity store and authentication configurations you make in TSM during Tableau Server setup, also referred to as the initial pool (TSM configured), remains unchanged, identity pools are configurable from Tableau Server.

Note: Identity pools are currently available for server-level configuration only. Identity pools can't be scoped to a site.

More about identity pools

Initial pool (TSM configured) versus identity pools

As noted above, the combination of provisioning and authentication configurations you make in TSM during Tableau Server setup is referred to as the "initial pool (TSM configured)". The initial pool (TSM configured) is a required component of the Tableau Server setup process and cannot be modified.

An identity pool, however, is optional and you can create as many identity pools as needed from Tableau Server directly.

Identity pools impact on users' sign-in experience

By default, when no identity pools are created for Tableau Server, there is no change to how your users navigate to the Tableau Server landing page and sign in to Tableau Server.

When one or more identity pools are created, the Tableau Server landing page displays multiple sign-in options. The primary sign-in option is displayed at the top of the page and is the way your users that belong to the initial pool (TSM configured) can sign in.

Below the primary sign-in option are the secondary sign-in options. Each option represents an identity pool, displayed in the order they were created. Users assigned to those pools must sign in using the option for the identity pool they belong to. To help guide your users to the correct sign-in option, consider adding a description to the identity pool when creating one.

Note: All users will see all pools that are configured for your Tableau Server, regardless of their pool membership.

Usernames and identifiers in Tableau

A username is the information that represents the system user. An identifier is used to supplement the username information and can be used by external identity stores as alternatives

to usernames.

In Tableau, a username is an immutable value that is used to sign in to Tableau and identifiers are mutable values used in Tableau's identity structure as a way to match users to their usernames. Identifiers enable Tableau to be more flexible because they can deviate from the username. If there are changes to the username in the external identity store, Tableau Server administrators can update the identifier to ensure users are matched to the correct usernames.

When you add an existing user to an identity pool, you might expect the ability to set an identifier. For example, if an existing user belongs to an identity pool configured with a local identity store and you want to add them to an identity pool configured with an AD identity store, we ask you to provide the username to search for identifiers associated with that user. On the other hand, if an existing user belongs to an identity pool configured with an AD identity store and you want to add them to an identity pool configured with a local identity store, we ask you to provide an optional identifier. An exception to this is if you want to add a user to the initial pool (TSM configured) that's configured with a local identity store and local authentication. You will be unable to set an identifier for that user.

Set Up and Manage Identity Pools

To create and manage identity pools, you are required to programmatically make calls against the Identity Pools Methods using the Tableau REST OpenAPI. To add or manage users in an identity pool, you can use Tableau Server user interface (UI) directly or through the Tableau REST API.

The identity pools setup process is summarized in the following steps.

1. **Configure Tableau Server and establish a session**.
2. **Provision users** by setting up a new local identity store instance. **Note:** You can skip this step to use an existing local identity store or the external identity store you configured in TSM during Tableau Server setup.
3. **Set up authentication** to authenticate your users to Tableau Server using OpenID Connect (OIDC).
4. **Create an identity pool** that uses the identity store and OIDC authentication you configured.

5. **Add users to an identity pool** by using Tableau Server UI or REST API to enable users to sign in to Tableau Server.

After setup, you can test, manage, and troubleshoot your identity pools.

Note: You can use the Identity Pools Postman collection in the Salesforce Developer's Postman workspace to learn about, develop, and test the methods described in this topic.

Prerequisites

Before getting started with identity pools, the following requirements must be met:

- Integration with an OIDC identity provider (IdP), such as Okta, is already configured
- You are running Tableau Server 2023.1 or later
- You have performed the identity migration if you are running Tableau Server after upgrading from version 2021.4 or earlier

Get started

Step 1: Configure Tableau Server and establish a session

Enabling changes associated with setting up identity pools require a one-time TSM configuration and a declaration of session and host variables.

1. Open a command prompt as an administrator on the initial node (where TSM is installed) in the cluster.
2. Run the following command:
   a. `tsm configuration set -k gateway.external_url -v http://<host>`
   b. `tsm pending-changes apply`

   For example, you can run the following commands to configure your Tableau Server, http://myco:

   ```
   tsm configuration set -k gateway.external_url -v http://myco
   tsm pending-changes apply
   ```

For more information, see gateway.external_url.

3. (Optional) Run the following commands to add a description for the initial pool (TSM configured):

   a. `tsm configuration set -k wgserver.authentication.identity_pools.default_pool_description -v "<description>"`
   b. `tsm pending-changes apply`

   For example, you can run the following commands to add a "Sign-in for MyCo employees" description:

   ```
   tsm configuration set -k wgserver.authentication.identity_
   pools.default_pool_description -v "Sign-in for MyCo employees"
   tsm pending-changes apply
   ```

   For more information, see wgserver.authentication.identity_pools.default_pool_description.

4. Sign in to Tableau Server as an administrator and do the following:

   a. Go to the browser's developer tools and navigate to the application's cookies.
   b. Note the **workgroup_session_id** value.

   For example, if working in Chrome, right-click anywhere on the Tableau Server homepage, right-click and select **Inspect**. Click **Application** from the top navigation pane and click **Cookies** from the left navigation pane. Under Cookies, click your Tableau Server name, like http://myco.com, and note the **workgroup_session_id** value in the center pane.

5. In the script or API developer tool you're using to make identity pools requests using the Tableau REST OpenAPI, do the following:

   a. Add the workgroup_session_id value as a global variable.
   b. In addition, add port 80, host (your Tableau Server URL), and protocol (HTTP or HTTPS) to your global variables.

For example, the following table shows the global variables necessary for your Tableau Server, http://myco.

Global Variable	**Value**
Work-group ses-sion ID	AbC_2ab-cDe-fDwGVzPu1hCQ|FJk5Z6OroPCLEDTKk-wDxaeA0YzrIY04f|ca608d3c-fc01-4e40-ae5e-9b2131e4e7mm
Port	80
Host	http://myco
Protocol	HTTP

Step 2: Set up an identity store

Tableau Server requires you to configure an identity store to source or provision your Tableau Server users.

When setting up an identity pool, you can use a new or existing local identity store, or you can use an external identity store, either Active Directory (AD) or Lightweight Directory Access Protocol (LDAP) if that external identity store was configured during Tableau Server setup.

Note: New AD or LDAP instances that are not the AD or LDAP instance you configured in TSM during Tableau Server setup (also referred to as the initial pool (TSM configured)), are not configurable with identity pools.

To set up a new local identity store, use the procedure below. Skip to Step 3: Set up authentication if you want to use an existing local identity store or the identity store you configured during Tableau Server setup.

1. Make a Sign In request to the Tableau REST API to generate a credentials token.

   Example

   URI

   ```
   POST https://myco/api/3.19/auth/signin
   ```

2. After the credentials token is generated, add the credentials token to the header of all subsequent API requests.

3. Configure the identity store by calling the Configure Identity Store endpoint using the Tableau REST API OpenAPI.

4. In the request, specify the following:

   a. Type. The type value is always 0 for a local identity store type. If you want to use an existing local identity store or the identity store you configured in TSM during Tableau Server setup, you don't need to set up a new local identity store instance. Instead, skip to Step 3: Set up authentication, below.
   b. Name. The name must be unique.
   c. Display name. This is optional.

   Example

   URI

   https://myco/api/services/authn-service/identity-stores/

   Request body (JSON)

   ```
   {
   "type": "0",
   "name": "Local identity store #1",
   "display_name": "Local identity store #1"
   }
   ```

Response body

None

Step 3: Set up authentication

You can configure OpenID Connect (OIDC) authentication method to authenticate your users.

Note: OIDC is currently the only authentication method configurable with identity pools, regardless of the identity store type you use with the identity pool.

1. After setting up an identity store, call the Create Authentication Configuration endpoint using the Tableau REST API OpenAPI.
2. In the request, specify the following:
   a. Authentication type. The authentication type values is "`OIDC`".
   b. iFrame. The iFrame's default value is "`false`".
   c. The required OIDC client ID, client secret, configuration URL, ID claim, client authentication, and username claim.
      - The client ID, client secret is provided by your OIDC IdP.
      - The configuration URL is also provided by your IdP. The URL and can typically use the following format: `https://<idp_url>/.well-known/openid-configuration`.
      - The ID claim's default value is "`sub`". For more information, see Changing the sub claim.
      - The client authentication's default value is "`CLIENT_SECRET_BASIC`".
      - The username claim's default value is "`email`". For more information, see Default: using email claim to map users.

      About username claim

      Tableau uses the username claim for identity matching purposes. If you provide identifiers when adding users to Tableau Server, the identifier is used to match

the value provided in the username claim. If no identifiers are provided, Tableau defaults to the username set in Tableau Server.

Notes:

- If you intend to use this authentication configuration with an identity pool that uses AD as its identity store, make sure that the assigned user has the AD sAMAccountName value in the username claim.
- If you intend to use this authentication configuration with an identity pool that uses LDAP as its identity store, make sure that the assigned user has the LDAP username value in the username claim.

Example

URI

https://myco/api/services/authn-service/auth-configurations/

Request body (JSON)

```
{
"auth_type": "OIDC",
"iframed_idp_enabled": true,
"oidc": {
       "client_id": "0oa1hotzhjv4tyCd08",
       "client_secret": "EsKd2NCxY-BiLu_zcIwr2lJZLziT_7sw9Fi6HV3",
       "config_url": "https://dev-532601-admin.oktapreview.com/.well-
known/openid-configuration",
       "custom_scope": "",
       "id_claim": "sub",
       "username_claim": "email",
       "client_authentication": "CLIENT_SECRET_BASIC",
       "essential_acr_values": "",
       "voluntary_acr_values": "",
       "prompt": "login,consent",
       "connection_timeout": 100,
       "read_timeout": 100,
       "ignore_domain": false,
```

```
        "ignore_jwk": false
        }
}
```

Response body

None

Step 4: Create an identity pool

Depending on the identity store that you configured during Tableau Server setup, the identity pool you create can have only one of the following identity store and authentication method combinations:

- AD identity store + OIDC authentication
- LDAP identity store + OIDC authentication
- Local identity store + OIDC authentication

The first two combinations require that the initial pool (TSM configured) is configured to use AD or LDAP.

The procedure described below creates an identity pool with last combination, "local identity store + OIDC authentication".

1. After configuring OIDC authentication, call the Create Identity Pool endpoint using the Tableau REST API OpenAPI.
2. In the request, specify the following:
    a. Name and description for the identity pool. Both the identity pool name and description are visible to all users on the Tableau Server landing page.
    b. Identity store instance ID and authentication type instance ID.

       Notes:

       - To get the identity store instance ID and authentication type instance ID, you can call the List Identity Stores and List Authentication Configurations

endpoints.

- If you want to create an identity pool that uses the identity store you configured in TSM during Tableau Sever setup, the identity store instance value is always `'1'`.

Example

URI

https://myco/api/services/authn-service/identity-pools/

Request body (JSON)

```
{
"name": "MyCo contractors",
"identity_store_instance": "2",
"auth_type_instance": "0",
"is_enabled": true,
"description": "Sign-in for MyCo contractors"
}
```

Example response body

None

3. After creating the identity pool, go to your IdP configurations and set the sign-in redirect URI to `http://<host>/authn-service/authenticate/oidc/<identity_pool_id>/login`.

   For example, `http://myco/authn-service/authenticate/oidc/57tgfe21-74d2-3h78-bdg6-g2g6h4734564/login`

   Note: To get the identity pool ID, you can call the List Identity Pools endpoint.

Notes:

- You can create as many identity pools as your organization needs.
- Other identity store types and authentication methods are supported by the initial pool (TSM) configured. For more information, see Authentication.

Step 5: Add users to identity pool

You can use Tableau Server directly to add users to an identity pool. Users must belong to the initial pool (TSM configured) or be added to an identity pool to sign in to Tableau Server. When adding users to an identity pool, your workflow can change depending on the identity store that was configured with the identity pool.

The procedure below describes how to add users to an identity through the Tableau Server UI. However, you can add users to an identity pool using the Tableau REST API by calling the Add User to Identity Pool endpoint.

1. Sign in to Tableau Server UI as an administrator.
2. From the left navigation pane, select **Users** (or **All Sites** > **Users** for a multi-site Tableau Server).
3. Click the **Add Users** button, and select **Create new user** or **Import users from file**.

   For **Create new user**:

   a. Select the identity pool you want to add the new user to, and then click Next.

      i. If you selected an identity pool that's configured with an AD or LDAP identity store, enter usernames, assign site membership, and site roles. When finished, click the Import Users button.

      ii. If you selected an identity pool that's configured with a local identity store, enter the username. The dialog box expands so that you can add a display name, an identifier (in most cases), email address, and set site and site roles. When finished, click the **Create User** button.

      For more information about usernames and how to assign site membership and site roles, see Set Users' Site Roles.

      About usernames and identifiers in Tableau

A username is the information that represents the system user. An identifier is used to supplement the username information and can be used by external identity stores as alternatives to usernames.

In Tableau, a username is an immutable value that is used to sign in to Tableau and identifiers are mutable values used in Tableau's identity structure as a way to match users to their usernames. Identifiers enable Tableau to be more flexible because they can deviate from the username. If there are changes to the username in the external identity store, Tableau Server administrators can update the identifier to ensure users are matched to the correct usernames.

When you add an existing user to an identity pool, you might expect the ability to set an identifier. For example, if an existing user belongs to an identity pool configured with a local identity store and you want to add them to an identity pool configured with an AD identity store, we ask you to provide the username to search for identifiers associated with that user. On the other hand, if an existing user belongs to an identity pool configured with an AD identity store and you want to add them to an identity pool configured with a local identity store, we ask you to provide an optional identifier. An exception to this is if you want to add a user to the initial pool (TSM configured) that's configured with a local identity store and local authentication. You will be unable to set an identifier for that user.

For **Import users from file**:

a. Upload a .csv file that contains the following columns in the order listed:

```
username, password, display name, license level, admin
level, publishing capability, email address, identity pool
name, identifier
```

Note: Username and password are the only required columns. However, if you don't specify the identity pool name, the user will be added to the initial pool (TSM configured). For more information, see CSV Import File Guidelines.

For example, suppose you want to add Henry Wilson and Fred Suzuki to the General Contractors identity pool. Your .csv might contain the following values:

```
henryw,henrypassword,Henry Wilson,View-
er,None,yes,hwilson@myco.com,General Contractors,hwilson
freds,fredpassword,Fred Suzuki,Creat-
or,None,no,fsuzuki@myco.com,General Contractors,fsuzuki
```

Note: When one or more identity pools are created, the Tableau Server landing page updates to include sign-in options for users that are members of those identity pools. For more information, see Provision and Authenticate Users Using Identity Pools.

Test identity pools

After setting up an identity pool, we recommend you test it by signing out of Tableau Server and signing in again as a user that belongs to the identity pool. Make sure you complete the sign in process to ensure that OIDC authentication was configured correctly.

Note: If you've configured an optional description for the initial pool (TSM configured) in Step 1: Configure Tableau Server and establish a session or have a Server Settings (General and Customization) note for Tableau Server, we suggest that the description is specific to users who sign in using the initial pool (TSM configured) and the Sign In Customization note applies to all users who sign in to Tableau Server.

Manage identity pools

You can manage the users in identity pools from both the server-level and site-level Users page. On the Users page, you can see which identity pools users belong to and summary details about the identity pool.

For all other identity pools management tasks, including updating an authentication configuration or identity pool and deleting a local identity store or identity pool, use the Tableau REST API OpenAPI described in the Identity Pools Methods.

Troubleshoot identity pools

Limitations of identity pools

Identity pools are available with Tableau Server only.

Note: Identity pools are currently available for server-level configuration only. Identity pools can't be scoped to a site.

Tableau Server landing page shows IdP errors

On the Tableau Server landing page, below the primary sign-in option, an IdP-related error message might display next to an identity pool sign-in option. This OIDC authentication-related issue can occur when one or both of the following are true: 1) Tableau Server hasn't been configured to send an external URL to the IdP and 2) the global variables haven't been declared.

To resolve this issue, make sure you complete the procedure described in Step 1: Configure Tableau Server and establish a session above.

Tableau Server landing page is not showing identity pools

If the identity pools capability is disabled, you can enable it again using the following TSM commands:

```
tsm configuration set -k features.IdentityPools -v true

tsm configuration set -k features.NewIdentityMode -v true

tsm configuration set -k wgserver.authentication.legacy_identity_mode.enabled -v false

tsm pending-changes apply
```

Note: Running these commands causes Tableau Server to restart.

Add Users to Tableau Server

You can add users to Tableau Server one at a time or in batches. You can add them to the server as unlicensed users, and then add them to sites and assign site roles as you onboard them to Tableau Server. Or you can add users to sites and specify their site roles at the same time, at which point they are ready to sign in.

Before you begin

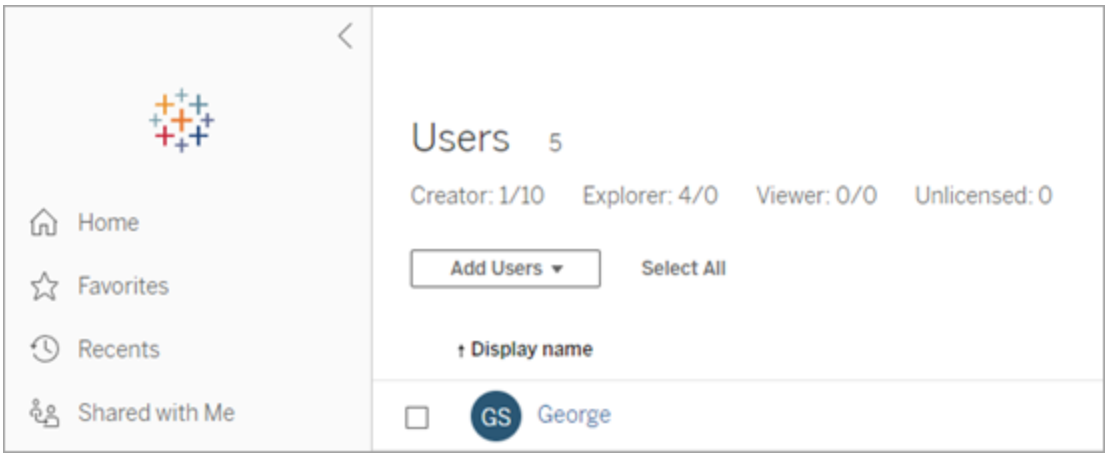
- If your Tableau Server is configured with an Active Directory external identity store, review User Management in Deployments with External Identity Stores to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

  Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

- Verify that you have enough user licenses and role licenses for your user base. If you attempt to add a user for a site role that you do not have a license for, then you will receive an error.
- To add user licenses to your Tableau Server deployment, visit the Tableau Customer Portal to purchase licenses and the corresponding product key(s). After you have purchased licenses, see Add Capacity to Tableau Server to update the server with the new key(s).
- The steps in this topic describe how to add an individual user and assign their site role. To add users in batches, see Import Users.

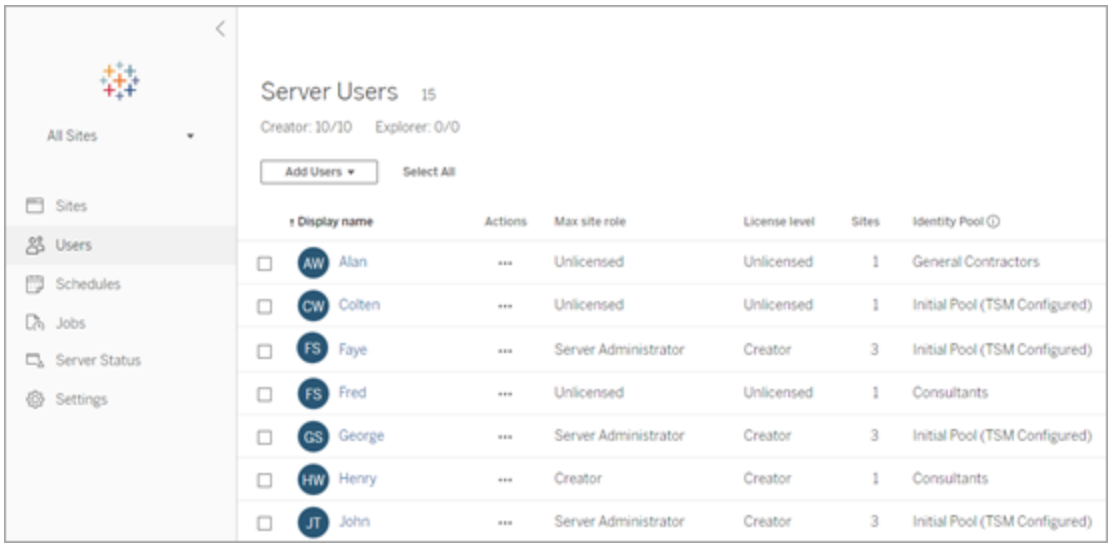
When adding users at the server level versus the site level

In a single-site environment, server administrators can add users on the **Users** page.



After you add a site to Tableau Server, it becomes a multi-site server with a **Server Users** page (all server users from every site appear here) and a **Site Users** page. Only server administrators can access the **Server Users** page, and both site administrators and server administrators can access the **Site Users** page.

The **Server Users** page is the only place where you can assign users to multiple sites, delete users from the server, and if the server is using local authentication, reset user passwords.



Add a user to the server

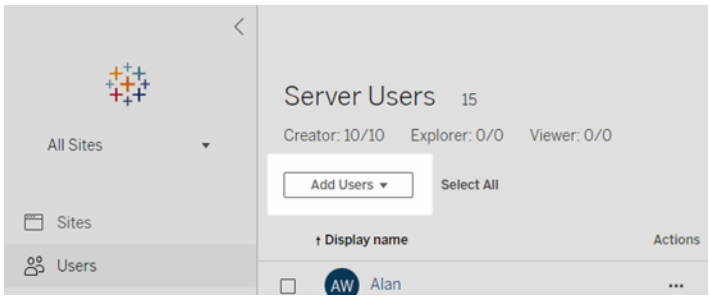
When adding a user, the workflow changes depending on whether at least one identity pool is configured for Tableau Server.

No identity pools configured

If no identity pools are configured, i.e., Tableau Server is only using the user provisioning and authentication configured during Tableau Server setup (also referred to as the initial pool (TSM configured)), follow the steps described below to add a user.

1. In the site menu, select **Manage All Sites** > **Users**, and then click **Add Users**.

   To add a user to a site, you select the site and go to the **Users** page.



2. Do one of the following:

    - If the server is configured for **local authentication**, click **New User**, and enter a user name. With local authentication, the best way to avoid user name collisions is to provide an email address for the user name. For example, *jsmith@example.-com* instead of *jsmith*.

      User names are not case sensitive. Characters not allowed in user names include the semi-colon (;) and colon (,).

    - If the server is configured for **Active Directory authentication**, click **Active Directory User**. If you are adding a user from the same Active Directory domain that Tableau Server runs on, the server domain will be assumed, and you can type the AD user name without the domain.

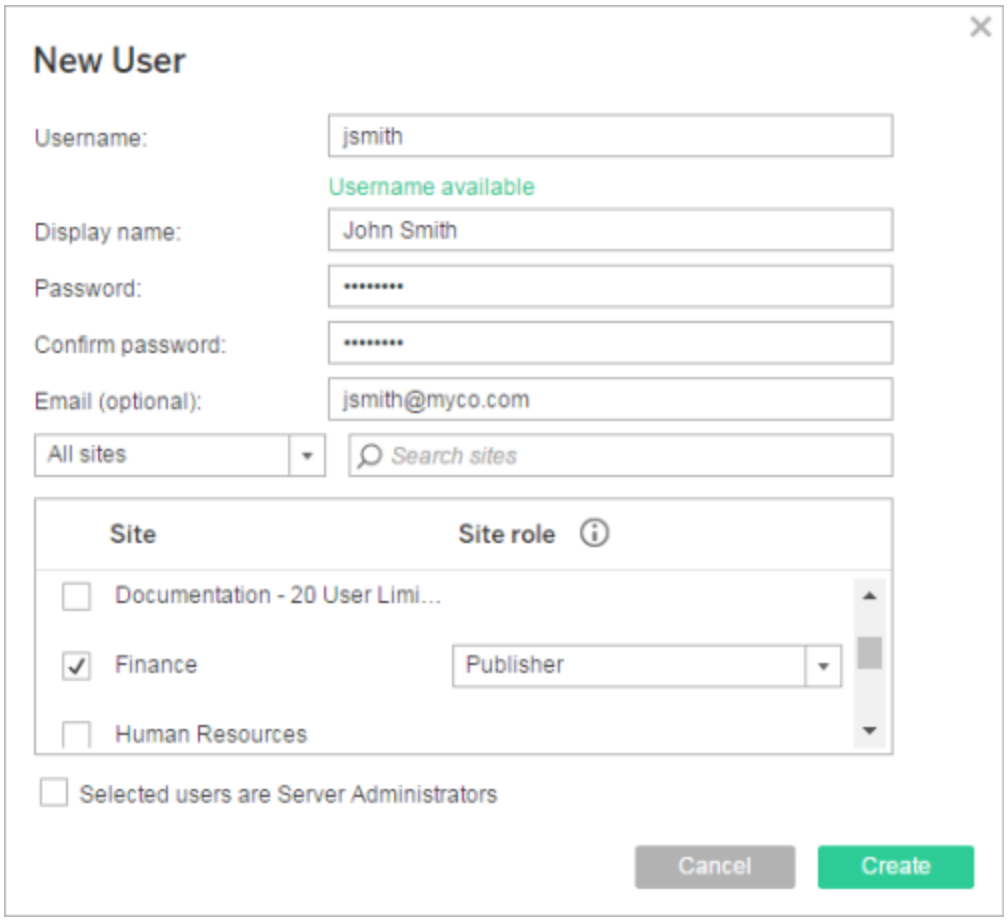
      Note: Do not enter the user's full name; this can cause errors during the importing process.

3. If the server is using local authentication, provide the following:

    - **Display Name**—Type a display name for the user (e.g., *John Smith*).
    - **Password**—Type a password for the user.
    - **Confirm password**—Retype the password.
    - **Email**—This is optional and can be added at a later time in the user profile settings.
    - **Selected users are Server Administrators**: Specify whether the user should be a server administrator.
    - **Name (Site Membership) / Site Role**: If the user is not a server administrator, you can assign a user to zero or more sites, along with a site role for each site. You do not have to choose site membership and site role at this time. If you don't

specify site membership and site role for a new server user, the user will be added as a server user only, with a site role of Unlicensed. For details, see Set Users' Site Roles.

4. Click **Create**.

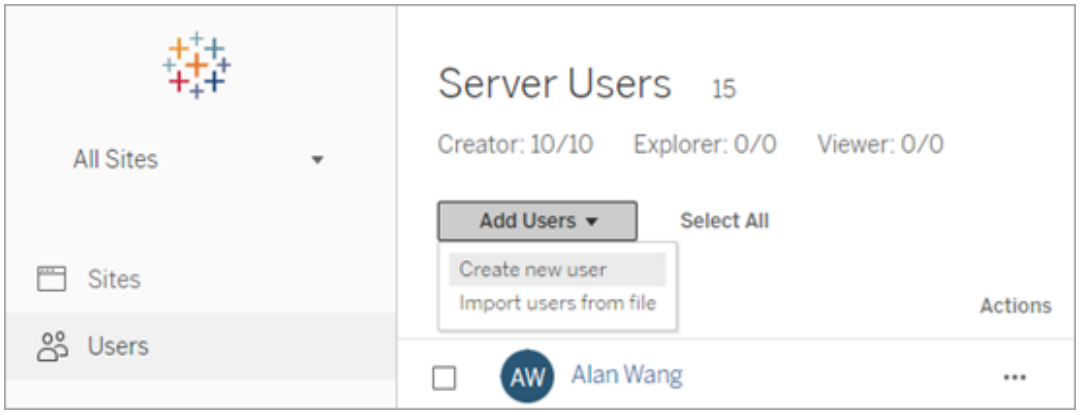


One or more identity pools configured

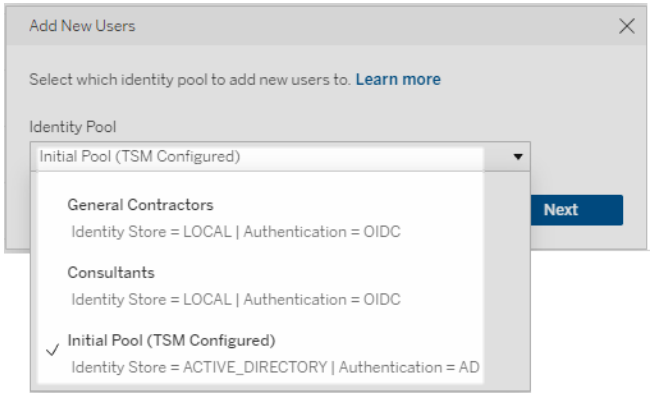
When adding a user, if one or more identity pools are configured, you must first select an identity pool or the **initial pool (TSM configured)**, which is the user provisioning and authentication configured in TSM during Tableau Server setup.

1. In the site menu, select **Manage All Sites** > **Users**, and then click **Add Users** > **Create new user**.

   To add a user to a site, you select the site and go to the **Users** page.



2. In the Add New Users dialog box, select the identity pool you want to add the user to.



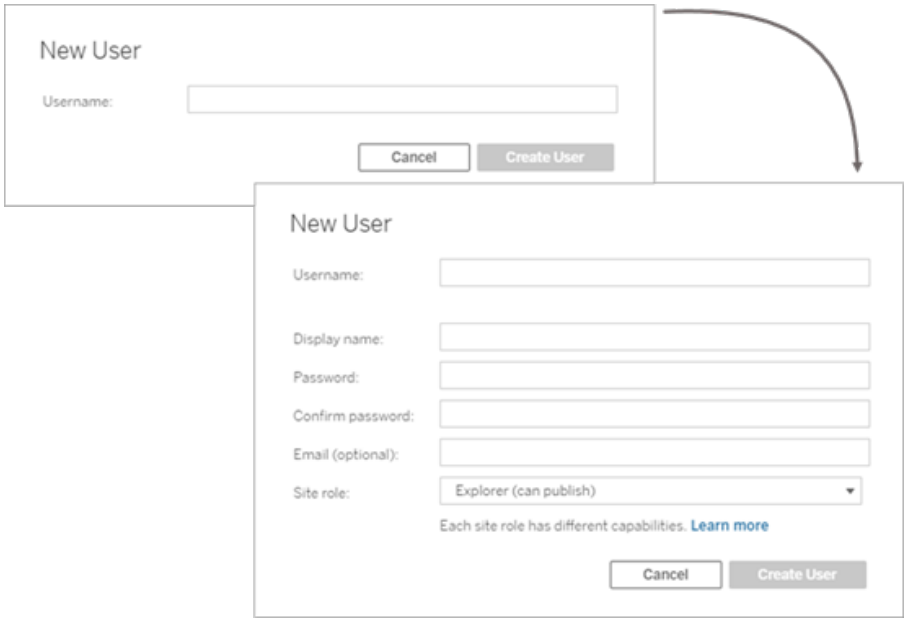
3. Do one of the following:

   - If the identity pool you selected in step 2 is configured for **LOCAL** identity store and **LOCAL** authentication:

     a. In the New User dialog box, in the **Username** text box, enter a user name. With local authentication, the best way to avoid user name collisions is to provide an email address for the user name. For example,

jsmith@example.com instead of *jsmith*.

User names are not case sensitive. Characters not allowed in user names include the semi-colon (;) and colon (,).



b. Enter the following information:

- **Display name**—Type a display name for the user (e.g., *John Smith*).
- **Password**—Type a password for the user.
- **Confirm password**—Retype the password.
- **Email (optional)**—This is optional and can be added at a later time in the user profile settings.
- **Selected users are Server Administrators**: Specify whether the user should be a server administrator.

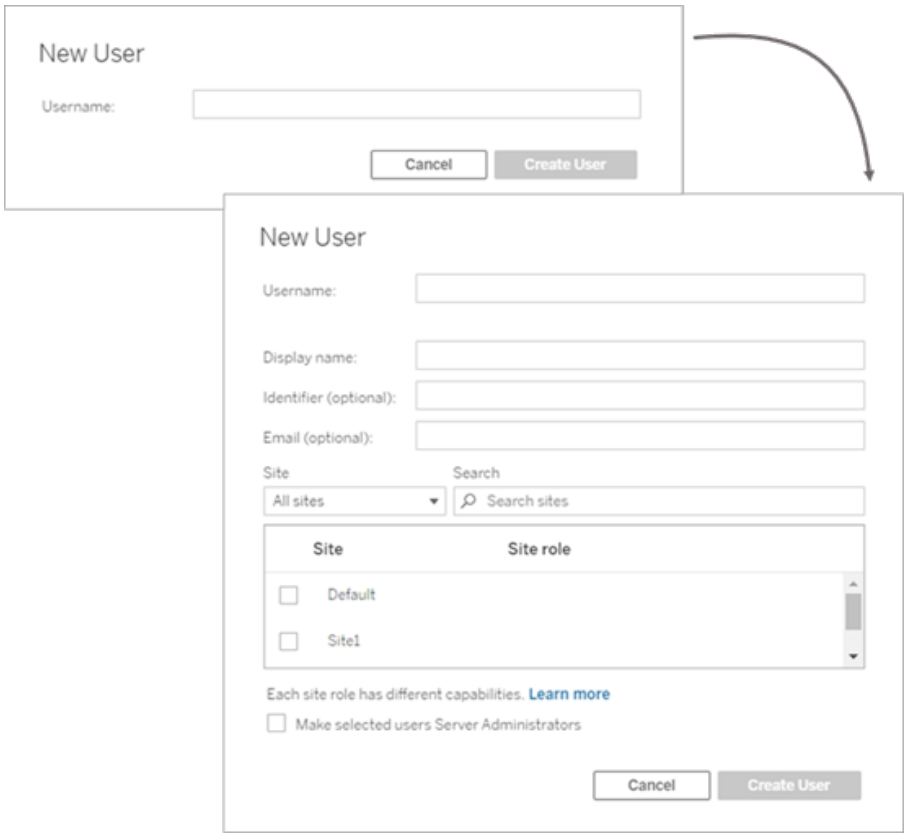
- **Site Role**: If the user is not a server administrator, you can assign a site role. You do not have to choose site role at this time. If you don't specify site role for a new user, the user will be added as a server user only, with a site role of Unlicensed. For details, see Set Users' Site Roles.

c. When finished, click **Create User**.

- If the identity pool you selected in step 2 is configured for **LOCAL** identity store with **OIDC** authentication:

  a. In the New User dialog box, in the **Username** text box, enter a user name. With local authentication, the best way to avoid user name collisions is to provide an email address for the user name. For example, *jsmith@example.com* instead of *jsmith*.

     User names are not case sensitive. Characters not allowed in user names include the semi-colon (;) and colon (,).



b. Enter the following information:

- **Display name**—Type a display name for the user (e.g., *John Smith*).
- **Password**—Type a password for the user.
- **Identifier (optional)**—Type the identifier you want to associate with the user. Identifiers are for identity matching purposes. For more information, see Usernames and identifiers in Tableau.
- **Email (optional)**—This is optional and can be added at a later time in the user profile settings.

- **Site** and **Site role**—If the user is not a server administrator, you can assign a user to zero or more sites, along with a site role for each site. You do not have to choose site membership and site role at this time. If you don't specify site membership and site role for a new server user, the user will be added as a server user only, with a site role of Unlicensed. For details, see Set Users' Site Roles.

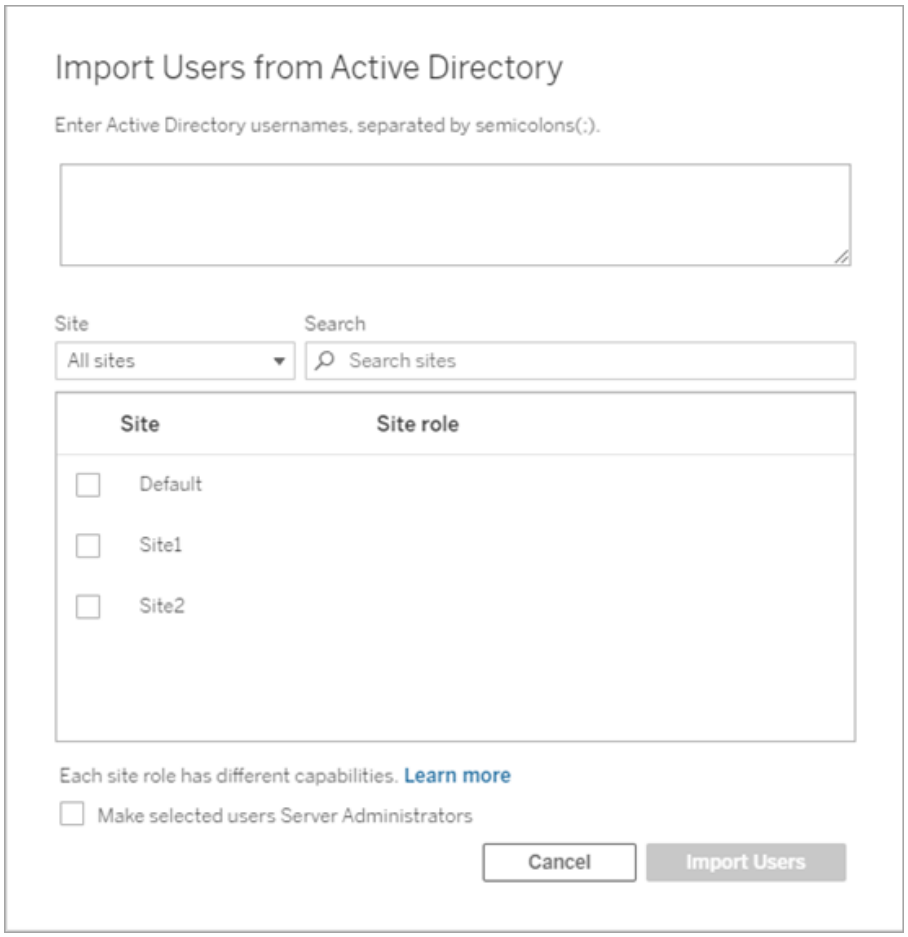
- **Make selected users Server Administrators**—Specify whether the user should be a server administrator.

c. When finished, click **Create User**.

- If the identity pool you selected in step 2 is configured for **ACTIVE_DIRECTORY** or **LDAP** identity store:

  a. Type the AD or LDAP user name without the domain. In this workflow, you're adding a user from the same Active Directory domain that Tableau Server was configured with in TSM during Tableau Server setup. Therefore, the server domain will be assumed, and you can type the AD or LDAP user name without the domain.

  Note: Do not enter the user's full name; this can cause errors during the importing process.



b. When finished, click **Import Users**.

Sign in to the Tableau Server Admin Area

As a server administrator on Tableau Server, you can access admin settings to configure sites, users, projects, and to do other content-related tasks.

If you want to change server settings such as processor, caching, authentication, distributed deployment, and other related configurations, see Sign in to Tableau Services Manager Web UI.

If you are running Tableau Desktop and want to sign in to Tableau Server to publish or access content and data sources, see Sign in to Tableau Server in Tableau Desktop.

Here's how to sign in to the Tableau Server admin pages:

1. Open your browser and enter the server URL. Here are some examples of what the URL might look like:

   `http://localhost/` (if you're working directly on the server computer)

   `http://MarketingServer/` (if you know the server's name)

   `http://10.0.0.2/` (if you know the server's IP address)

   If the server is *not* using port 80, you need to include the port number in the URL, as in these examples:

   `http://localhost:8000/`

   `http://MarketingServer:8080/`

   `http://10.0.0.2:8888/`

   ... where `8000` or `8080` or `8888` is the port that you configured.

   Tableau Server displays one of the following pages depending on whether identity pools are configured:

   - When no identity pools are configured, a page where you can enter a user name and password.

   - When identity pools are configured, the Tableau Server landing page with primary (initial pool (TSM configured)) and secondary (identity pools) sign-in

options.

2. Enter the credentials for the server administrator that you created when you finished installing Tableau Server.

   You're then taken to the main page of the **Default** site, and you're ready to create users, sites, and manage content.

Reset the server administrator account and password

If you have lost the password for the initial server administrator account run the following commands:

1. tsm reset.
2. `tabcmd initialuser`. See initialuser.

Navigate the Admin Areas of the Tableau Web Environment

As an administrator on Tableau Server or Tableau Cloud, you can access admin settings that aren’t available to other users to configure sites, users, projects, and to do other content-related tasks.

The settings in this article refer to the Tableau web environment. Tableau Server administrators with appropriate credentials can also change server settings such as processor, caching, authentication, distributed deployment, and related configurations using the TSM web environment. For information, see Sign in to Tableau Services Manager Web UI.

Access based on site role and number of sites

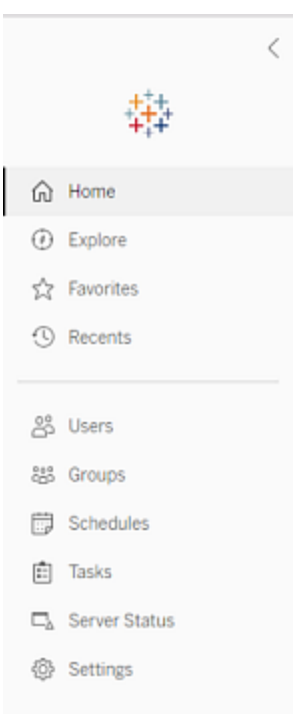
The menus you get when you sign in to Tableau Server or Tableau Cloud depend on the following conditions:

- Whether you're a site or server administrator.

  Site administrator access is available on Tableau Cloud and Tableau Server. Server administrator access is only on Tableau Server.

- Whether you have access to only one site or to multiple sites.

Server administrator

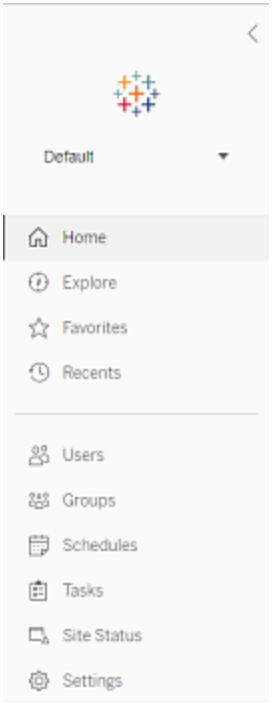
On a **single-site** server, the site selector does not appear, and all other menus are the same.	

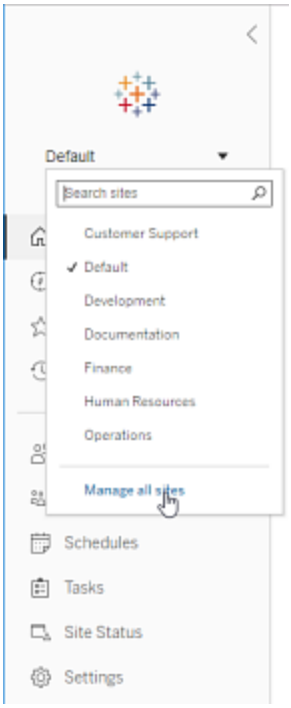
In a **multi-site** environment, menus along the left enable you to modify a specific site or all sites, and to configure users, groups, schedules, tasks, and server settings.

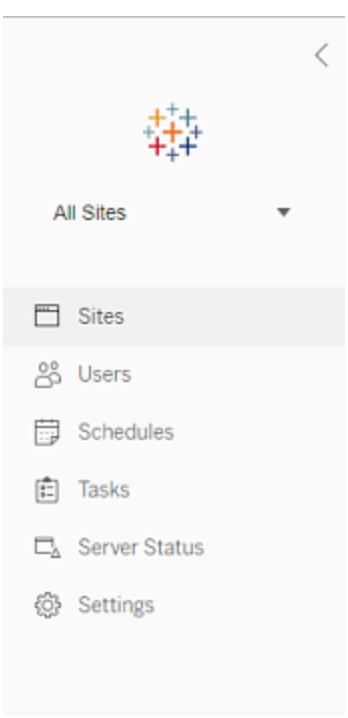
To access server administrator settings that affect all sites, open the site menu by clicking the arrow next to the current site name, and then select **Manage all sites**.

The **Content** and **Group** tabs go away, and the site menu text changes to **All Sites** to let you know you are managing server-wide settings, and options like **Server Status** reflect the server-wide view.

To return to the site administration menus, select **All Sites**, and the select the site you want to manage.





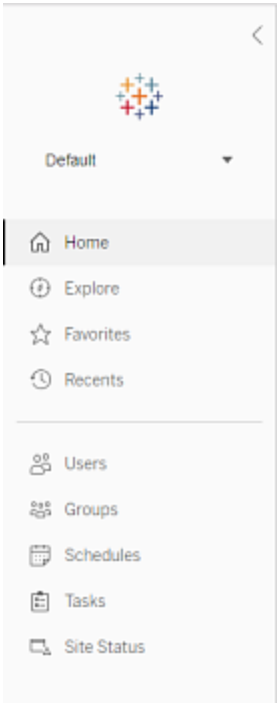


Site administrator

If you are a site administrator for Tableau Cloud or Tableau Server, and you have access to multiple sites, you'll get menus for selecting which site to manage, and for managing that site's content, users, groups, schedules, and tasks, and for monitoring its status.

The site selector displays the name of the current site. To go to another site, select the site menu, and then select the site name.

If you have access to only one site, the site selector does not appear, but all other menus are the same.



Server administrator tasks

Server administrators (available with Tableau Server Enterprise only) can do the following:

- Monitor server status and activity.
- Generate log files.
- Add sites and edit site settings. Only server administrators can add sites to the server.
- Add users to the server, and assign users to sites.
- Add and manage site groups.
- Add users to or remove users from identity pools.

To manage settings only for a specific site, you must first navigate to the site. Within each site, you can do the following:

- Administer content: Create projects, move content from one project to another, assign permissions, change ownership of a content resource, and so on.
- Manage schedules for extract refreshes and subscriptions.
- Monitor site activity and record workbook performance metrics.
- Manage storage space limits for content published by users.
- Allow web authoring.
- Enable revision history.
- Allow site administrators to add and remove users.
- Set the maximum number of licenses that site can consume for each license type (Creator, Explorer, Viewer).
- Allow users to subscribe to workbooks and views, and allow content owners to sub-

scribe others to workbooks and views.

- Enable offline snapshots for favorites (iOS only).

Site administrator tasks

A site administrator on Tableau Cloud or Tableau Server can do the following tasks:

- Administer content: Create projects, move content from one project to another, assign permissions, change ownership of a content resource, and so on.
- View, manage, and manually run schedules for extract refreshes and subscriptions.
- Add and manage site users (if allowed by the server administrator; see Site Settings Reference).
- Add and manage site groups.
- Monitor site activity.

Sign in to Tableau Services Manager Web UI

This topic explains how to sign in to the Tableau Services Manager (TSM) web UI. The TSM web pages are used to configure Tableau Server settings such as user authentication, server processes, caching, and other server-related settings. You can also configure TSM from a command line shell. See tsm Command Line Reference.

Important: Signing into TSM is *not* the same as signing into Tableau Server. TSM is used to configure the server, and requires an account with administrative privileges on the computer running TSM. Depending on how your enterprise is organized, the TSM administrator could be a user who does not have a Tableau Server account. A Tableau Server administrator has access to administrative pages for creating and editing sites, user, product, and other content-related tasks. For information about signing into Tableau Server as a Tableau Server administrator, see Sign in to the Tableau Server Admin Area.

If you are running Tableau Desktop and want to sign in to Tableau Server to publish or access content and data sources, see Sign in to Tableau Server in Desktop.

Requirements

- The account that you use to sign in to TSM must have authorization to make changes on the local computer where Tableau Server is installed.

  Specifically, the account that you use to sign in to TSM must be a member of the TSM authorization group that was created during initialization. The default name of the TSM authorization group is `tsmadmin`.

  To view the user accounts in the TSM authorization group, run the following command in the Bash shell. This example uses the default group name, `tsmadmin`:

  ```
  grep tsmadmin /etc/group
  ```

  If the user account is not in the group, run the following command to add the user to the `tsmadmin` group:

  ```
  sudo usermod -G tsmadmin -a <username>
  ```

  Tableau Server on Linux relies on PAM for core authentication scenarios. For more information about PAM integration for TSM administration, see TSM Authentication.

- If you are running a distributed deployment of Tableau Server, then enter the host name, or IP address of the computer running the initial node. The credentials you enter must have administrative access to the computer running the initial node, as specified above.

- Specify an HTTPS protocol in the server URL. Tableau Server installs a self-signed certificate as part of the installation process. Therefore, the protocol must be specified as `https`. For more information about the self-signed certificate and certificate trust for TSM connections, see Connecting TSM clients.

- Specify the port for TSM web UI (`8850`) in the URL.

- If you are running a local firewall, open port 8850. See Configure Local Firewall.
- You must specify the hostname or IP address of the computer running TSM. If you have set up a load balancing or proxy solution in front of Tableau Server, do not specify the load balancer or proxy address.
- As a security best practice, do not expose the TSM port (by default, `8850`) to the internet.

Sign in to the TSM web UI

1. Open a browser and enter the Tableau Server URL, and append the dedicated TSM web UI port.

   Here are some examples of what the URL might look like:

   `https://localhost:8850/` (if you're working directly on the server computer)

   `https://MarketingServer:8850/` (if you know the server's name)

   `https://10.0.0.2:8850/` (if you know the server's IP address)

2. In the sign-in page that appears, enter your administrator user name and password.

   Note: Tableau Server creates and configures a self-signed certificate during the installation process. This certificate is used to encrypt traffic to the TSM Web UI. Because it's a self-signed certificate, your browser will not trust it by default. Therefore, your browser will display a warning about the trustworthiness of the certificate

before allowing you to connect.

Customize Your Server

You can customize the Tableau Server web pages to personalize it for your company or group. You can perform these customizations:

- Change the server name that appears in the browser tab, tooltips, and messages.
- Change the logos that appear in the web environment.

  For more information, see tsm customize.

- Set the language used for the web environment and the locale used for views. See Language and Locale for Tableau Server.
- Install custom fonts on Tableau Server and client computers that connect to Tableau Server. See Use Custom Fonts in Tableau Server.
- Add a custom note to the server sign in page. The Sign In setting lets you add text. You can optionally add a URL to make the text a link. This note will also appear if a user receives a sign in error.

  Custom notes do not display on Tableau Mobile. If Tableau Server is configured with identity pools, the Sign In Customization note appears on both the Tableau Server landing page below all sign-in options and on the page where your initial pool (TSM configured) users enter their username and password.

  To set a custom note, sign in to a site on Tableau Server. On the left-side navigation pane, select **Manage all sites** from the drop-down site list. Select **Settings** and add a message to **Sign In Customization**.

  For more information, see Server Settings (General and Customization).
- Add a custom message to the welcome banner on the home page for all server users to see. The custom message can contain up to 240 characters of text and hyperlinks as well as one paragraph break. Administrators can also disable the default Tableau welcome banner for the server.

  To set a custom welcome banner, sign in to a site on Tableau Server. On the left-side navigation pane, select **Manage all sites** from the drop-down site list. Select **Settings**, then navigate to the **Customization** page.

Administrators and project leaders can also add images for projects in thumbnail view.

Language and Locale for Tableau Server

Tableau Server is localized into multiple languages. Server language and locale settings impact how this affects users. The **Language** setting controls user interface (UI) items such as

menus and messages. The **Locale** setting controls items in views such as number formatting and currency.

Administrators can configure language and locale on a server-wide basis and individual users can configure their own settings (search for "Your Account Settings" in the Tableau Server Help). If a user configures their own language and locale, their settings override the server settings.

Supported Languages

Tableau Server is localized into multiple languages. See the "Internationalization" section of the Tableau Server Technical Specification page for more information.

Default Settings

The default language for Tableau Server is determined during Setup. If the host computer is configured for a language Tableau Server supports, Tableau Server installs with that language as its default. If computer is configured for a language that is not supported, Tableau Server installs with English as its default language.

How Language and Locale are Determined

Another influence on which language and locale display when a user clicks a view is the user's web browser. If a server user has not specified a **Language** setting on their User Account page, and their web browser is set to a language that Tableau Server supports, the browser's language will be used—even if Tableau Server itself is set to a different language.

Here's an example: Assume that Tableau Server has a system-wide setting of English as the **Language** for all users. Server user Claude does not have a language specified on his Tableau Server User Account page. Claude's browser uses German (Germany) for its language/locale.

When Claude signs in to Tableau Server, the server UI displays in German and when he clicks a view, the view uses the Germany locale for numbers and currency. If Claude had set his user account **Language** and **Locale** to French (France), the UI and view would have been

displayed in French. His user account setting supersedes those of his web browser, and both of those have precedence over the Tableau Server system-wide setting.

Another setting to be aware of is the **Locale** setting in Tableau Desktop (**File** > **Workbook Locale**). This setting determines the locale of the data in the view, such as which currency is listed or how numbers are formatted. By default, **Locale** in Tableau Desktop is set to **Automatic**. However, an author can override that by selecting a specific locale. Using the above example, if the author of View A set **Locale** to **Greek (Greece)**, certain aspects of the data in View A would display using the Greek (Greece) locale.

Tableau Server uses these settings, in this order of precedence, to determine language and locale:

1. Workbook locale (set in Tableau Desktop)
2. Tableau Server User Account language/locale settings
3. Web browser language/locale
4. Tableau Server Maintenance page language/locale settings
5. Host computer's language/locale settings

Use Custom Fonts in Tableau Server

You can use custom fonts with Tableau Server. When you do this the safest way to guarantee that users have the experience you intend is to keep the following in mind:

-

- The fonts need to be installed on the computer where Tableau Server is running. After installing the fonts, restart Tableau Server to use the new fonts.
- The fonts need to be installed on any client computers that will connect to Tableau Server. You need to have the fonts installed locally in order for your browser to properly display them.

- As a best practice, use "web safe" fonts that are installed by default on all major browsers. This increases the likelihood that the fonts will display properly on client machines.
- Different browsers render the same fonts differently, so even when a client browser has the custom font installed, it may look different when viewed in different browsers. This can be especially noticeable with comments or titles where specific spacing is used for an intentional effect.

Note: For more information about installing fonts on Linux, refer to your Linux distribution's documentation and support.

Manage Sites Across a Server

You can plan and manage your sites in Tableau Server. You can manage users and groups for your sites, manage projects and content access, manage data, and create and interact with views on the web.

Sites Overview

The topics in this section describe the Tableau Server concept of a site and aspects of working with multiple sites. Topics include authentication type each site uses, and what to know about user licenses and administrator-level access to sites.

What is a site

You might be used to using the term *site* to mean “a collection of connected computers,” or perhaps as the short form of “website.” In Tableau-speak, we use site to mean a collection of users, groups, and content (workbooks, data sources) that’s walled off from any other groups and content on the same instance of Tableau Server. Another way to say this is that Tableau Server supports multi-tenancy by allowing server administrators to create sites on the server for multiple sets of users and content.

All server content is published, accessed, and managed on a per-site basis. Each site has its own URL and its own set of users (although each server user can be added to multiple sites). Each site's content (projects, workbooks, and data sources) is completely segregated from content on other sites.

For site administrator recommendations for how to set up users on a site, how to structure a site for publishers and other content users, how to give users permissions to share and manage their content, and so on, see the Manage Individual Sites section.

For information about how users can get their content to Tableau Server, see Publish Data Sources and Workbooks in the Tableau user help.

Authentication and sign-in credentials

By default, all sites on a server use the same identity store type. You configure these settings when you install Tableau Server. For information, Configure Initial Node Settings.

Users who have access to more than one site on the same Tableau Server instance use the same credentials for each site. For example, if Jane Smith has a user name of *jsmith* and a password of *MyPassword* on Site A, she uses those same credentials on Site B. When she signs in to Tableau Server, she'll be able to choose which site she wants to access.

The Default site

Tableau Server installs with a site named Default. If you maintain a single-site environment on Tableau Server, this becomes the site you work with, and on which your users share their Tableau analysis. If you add sites, Default becomes one of the sites you can select when you sign in to Tableau Server. Default differs from sites that you add to the system in the following ways:

- It can never be deleted but, just like sites that you add, it can be renamed.
- It stores the samples and data connections that ship with Tableau Server.

- The URL used for Default does not specify a site. For example, the URL for a view named Profits on a site named Sales is `http://-localhost/#/site/sales/views/profits`. The URL for this same view on the Default site would be `http://localhost/#/views/profits`.

Why or why not add sites

On Tableau Server, users, projects, groups, data sources, and workbooks are managed per site. You can add users to multiple sites.

Each environment and its needs is unique. However, as a baseline, Tableau Visionaries and Product Managers tend to recommend using sites for true multi-tenancy needs. In other words, create a new site only when you need to manage a unique set of users and their content completely separately from all other Tableau users and content.

For site administrator recommendations for how to set up users on a site, how to structure a site for publishers and other content users, how to give users permissions to share and manage their content, and so on, see the Manage Individual Sites section.

Examples for which it makes sense to use sites

- You are a consultant who manages Tableau analysis for multiple clients, and you want to create a site for each client, to ensure that data from one client is not exposed to another.
- You want to allow Guest user access to a small and contained area of the server.

Examples for which projects can work better than sites

- A content-development process in which data sources and reports evolve from sandbox to production phases.

  Migrating users and content from one site to another is a laborious process. Although you might have good reasons to use sites for this and similar processes, by creating sites, you as the site administrator compound your ongoing maintenance burden. For each configuration update you make to one site (for example creating new projects and

setting permissions), you usually would need to duplicate the same work on each additional site.

- You want to separate areas of the server by functional area.

  Among a group of Tableau users, it's common that some users need to access content in multiple areas. Using sites would encourage publishing the same data sources and reports to multiple sites. This leads to data source proliferation and can negatively impact server performance. Using projects is a simpler way to work with this scenario.

For additional ideas, see the following resources:

- Why use projects in the topic User Projects to Manage Content Access.
- Discussions about sites on the Tableau Community forums. Here's a link to get you started.

Administrator-level access to sites

Tableau Server includes three administrator-level site roles: Server Administrator, Site Administrator Creator, and Site Administrator Explorer.

The **Server Administrator** site role always takes the highest license available, and it allows full access to Tableau Server, including all content access. You can find more information about this role in Server Administrator Overview. Server administrators also create sites as needed. (Site administrators don't have permissions to do this.)

A server administrator can assign one of the **Site Administrator** site roles to users to delegate creating and maintaining a specific site's user and content framework. The content framework enables Tableau users to share, manage, and connect to data sources and workbooks.

- Assign **Site Administrator Creator** to administrators who also connect to data, and create and publish data sources or workbooks. This site role takes a **Creator** license.
- Assign **Site Administrator Explorer** if the user manages the content framework but

doesn't need to edit the content itself. This site role takes an **Explorer** license, and it allows viewing and interacting access.

By default, the Site Administrator site roles allow creating and managing the site's users and groups, creating projects to organize content on the site, assigning permissions to allow users (groups) to access the content they need, scheduling extract refreshes, and a few other tasks.

A server administrator can deny site administrators' user management tasks. For example, you might do this if you use the Site Administrator Creator role for the data experts. In other words, you want to allow these users to manage connections to underlying data, create and publish "single source of truth" data sources, create top-level projects, and organize content across projects without restriction; but not necessarily add and remove site users.

For each site the server administrator can also limit site administrator access, so that site administrators can manage groups and content, but not add or remove users or set users' site roles.

In some organizations, the same person might be both a server administrator and site administrator for one or more sites. Even so, the tasks performed by a site administrator and a server administrator are distinct.

Licensing and user limits

You can add server users to multiple sites, and set their site roles and permissions on each site. A user who belongs to several sites does not need a license for each site. Each server user needs only one license. The license that user will use corresponds to the highest site role they have on the server. To learn more about how licenses and site roles intersect, see Set Users' Site Roles.

Server administrators can use the **Limit number of users** setting (select **Site <*name*> > Settings**) to specify a user limit for the site, or set a site role limit, which limits the number of Creators, Explorers, and Viewers allocated at the site level. To learn more, see Manage Site Role Limits.

Only licensed users are counted. For example, if a site has 90 licensed users (including administrators), 20 unlicensed users, the user count is 90. For information about how to view the number of licenses and site roles across the server, see View Server Licenses.

Export or Import a Site

Note: For detailed information about migrating sites from Tableau Server to Tableau Cloud, see our Tableau Cloud Manual Migration Guide.

You can provision a new Tableau Server site by importing (migrating) information from another site. You do this by exporting the existing site's (the *source* site) information to a file. Then you complete steps to verify and import that information to the *target* site.

Site Migration Options

You can migrate a site in any of these ways:

- To another site on the same Tableau Server instance.
- To a site on a separate Tableau Server instance.
- From Tableau Server on Windows to Tableau Server on Linux or vice-versa.

Note: When migrating sites between instances of Tableau Server, the target site must be on a version of Tableau Server that is the equal to or later than the version of Tableau Server for the source site. Both the source and target sites must be from supported versions of Tableau Server.

Site Migration Limitations

What information is preserved in a site export

- The export file you create preserves workbooks, projects, data sources, and users. This includes permissions set on content, user favorites lists, and site quotas.
- Users' custom views are preserved; however, depending on the type of site migration, custom view URLs might change in a way that breaks users' bookmarks to their views.
- When you export a site on Tableau Server to import to another Tableau Server site, subscription and extract refreshes schedules are preserved.

What information isn't preserved in a site export

- Usage data, which appears in the site's administrative views, is not preserved. For example, view and data source counts, user actions, and performance data.
- Backgrounder jobs that are in-progress while a site is being exported, will not be exported and will not show up on the new site once the import is complete.
- OAuth access tokens embedded in data connections are reset. For those data sources, you will need to edit the connections and re-authenticate to the underlying data.
- Prep flows and flow schedules are not included. These will need to be manually migrated.
- Content saved to users' Personal Space is not included.

Prepare the Source and Target Sites

Before you export a site, complete the following checklist to prepare both environments. Some of these instructions depend on whether both sites are on the same server instance or on separate ones.

Delete stale content

Make sure the source site contains only what you want to import to the new site. As a best practice, remove anything from your source site that you do not want to include in your new site, whether these things will be included in an export/import, or will need to be manually migrated. Delete unused workbooks, data sources, or projects. If you have Prep flows or flow schedules you no longer use, delete these as well.

Remove obsolete users

Confirm that all server users are licensed, and remove accounts that are no longer in use. You can't remove users during the import process, so if the two sites are on the same server instance, all users you export from the source site are imported to the target site.

Create or identify the target site

You must import a site file to a site that already exists on the target Tableau Server instance. Because the import process removes anything from the target site that is not included in the import file, we recommend that you import to an empty site. For more information, see Add or Edit Sites.

Locate site IDs

The tsm command you use to export or import a site requires a parameter that takes the site ID. You can get the site ID from the URL when you are signed in to the site from a web browser.

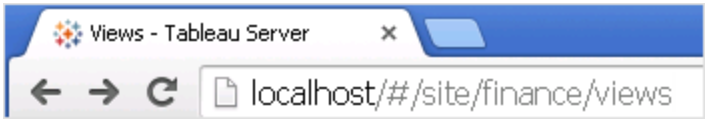
If only one site exists on the server, that site is named **Default**. When you are signed in to the Default site, the browser URL looks something like this:

```
https://server-name/#/projects
```

In the URL, the absence of the `/site` parameter indicates that it's the Default site. The site ID for the Default site is "Default" without the quotation marks.

On a multi-site Tableau Server deployment, the browser URL includes `#/site/` followed by the site ID. The following URL would appear if you navigate to the Views page on a site whose site ID is `finance`:

```
https://localhost/#/site/finance/views
```



Check the identity store

You can export from and import to sites that do not use the same user identity store type, but you will need to modify the mapping files used for the import. This step is built into the import process and described in Step 3: Verify that site settings are mapped correctly.

Create users on the target server if necessary

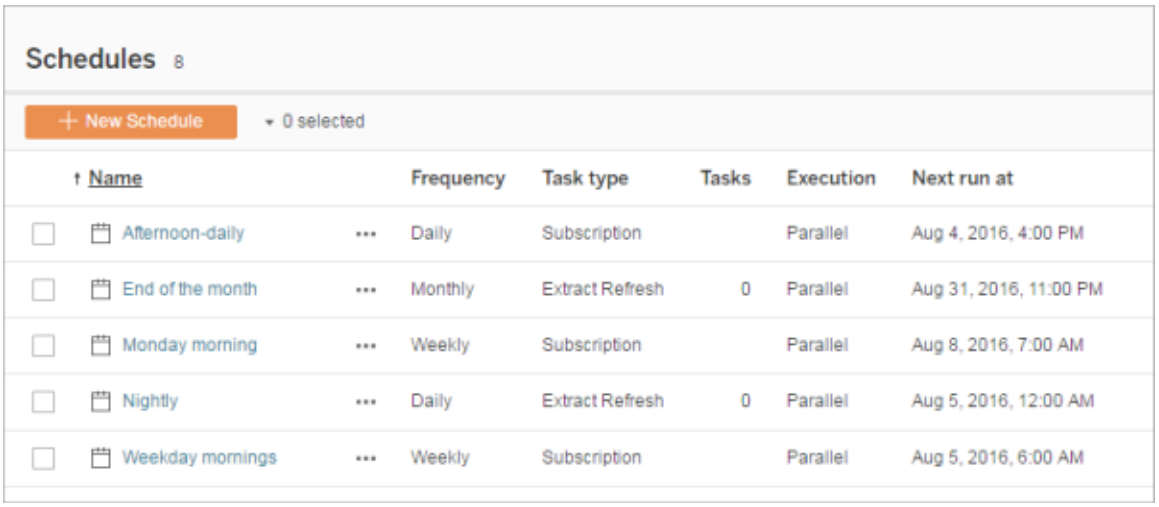
The site import process assigns users to the target site. If the source site is on a Tableau Server instance other than the target site, you must create users on the target server before you can perform the import. If the two sites are on the same Tableau Server instance, the target site has access to the existing users, and you can skip this step.

Configure the target server to deliver subscriptions

Subscriptions are imported, but you must configure the server to deliver them. For more information, see Set Up a Site for Subscriptions.

Check schedules

The **Schedules** page lists the existing schedules for extract refreshes and subscriptions.



For migrations **from one Tableau Server site to another**, refreshes and subscriptions assigned to default schedules on the source site are mapped to the same schedules on the target site. If the source site has schedules that do not exist on the target site, and the target site is on another Tableau Server instance, you must create the schedules on the target site that you want the source schedules to map to. You can edit the mapping files to make sure this is done as you expect.

Tips for importing to a target with fewer users or schedules

When a target site has fewer users or schedules than the source site, many-to-one importing is not supported. How you can address this depends on whether the source and target sites are on the same Tableau Server instance.

You can take any of the following approaches that apply to your site migration use case:

- Remove extra users or schedules from the source site before you export. This is the preferred option if the two sites are on the same server instance.
- Add missing users or schedules to the target site before beginning the import. This is required if the target site is on another server instance.

- Add the missing users or schedules to the target site in the middle of the import process and manually update the mapping files. This is an option only if the sites are on the same instance.
- Manually map the users or schedules to different users and schedules in the target site during the import process. This is required if a user name differs between servers—for example, the exported user named *adavis@company.com* is defined on the target site as *davisa*.

Migrating a Site

You must use the `tsm sites` commands to complete the site migration process. The steps below walk you through exporting information from the source site, mapping site settings, and importing the mapped files to the target site.

Step 1: Export a site

On the source Tableau Server machine, type the following command:

```
tsm sites export --site-id <source-siteID> --file <filename>
```

Tableau Server must be running when you use the `export` command. During the export process, Tableau Server locks the site you are exporting.

For example, to export a site with site ID **weather-data** to the file **export-file.zip**, type the following:

```
tsm sites export --site-id weather-data --file export-file
```

By default, Tableau Server saves `<export-file>.zip` to `/var/-opt/tableau/tableau_server/data/tabsvc/files/siteexports`. For more information, see tsm File Paths.

Step 2: Generate the import mapping files

To generate import files for the target site, you need the .zip file you created when you completed the steps in Step 1: Export a site.

Note: The exported file does not give "Others" read permission by default. Depending on who is importing the file, you may need to adjust the permissions on the file to allow a non-owner to read it.

1. On the target Tableau Server machine, copy the exported .zip file to the directory Tableau Server expects to find the files for importing. For example:

   ```
   /var/opt/tableau/tableau_server/data/tabsvc/files/siteimports
   ```

2. Verify that the target site already exists on Tableau Server, as the import process will not create a new site. For more information, see Prepare the Source and Target Sites.

3. Run the following command on the target Tableau Server machine (Tableau Server must be running):

   ```
   tsm sites import --site-id <target-siteID> --file <export-file.zip>
   ```

   This command generates a set of .csv files that show how source site settings will map to the target site. In the steps described in the next section of this article, you confirm these mappings and adjust them where needed.

   By default, these .csv files are generated to a `mappings` directory created under `siteimports`. For example:

   ```
   /var/opt/tableau/tableau_server-
   /data/tabsvc/files/siteimports/working/import_<id>_<date-time>/mappings
   ```

   For more information, see tsm File Paths.

Step 3: Verify that site settings are mapped correctly

The .csv files you generated in the previous section describe how the source site's resources will be assigned to the target site when the import is complete. Items in the files that Tableau Server was unable to map, and that you need to edit, are indicated by a series of question

marks (`???`). Before you can complete the import process, you must replace the question marks with valid assignments on the target site.

Important: Some requirements apply to mapping users, schedules, and published content resources, particularly when the source and target sites are on separate Tableau Server instances. For more information, see Prepare the Source and Target Sites earlier in this article.

To verify mapping files

1. Navigate to the directory that contains the .csv map files generated by the `tsm sites import` command. By default:

   `/var/opt/tableau/tableau_server-/data/tabsvc/files/siteimports/working/import_<id>_<date-time>/mappings`

2. Use your preferred text editor to open one of the .csv files in the `mappings` directory, and do the following.

   a. Confirm that the mappings are correct.

   b. If an entry shows a series of question marks (`???`), replace them with a valid value.

      For descriptions of the settings in each of these files, use the tables in Mapping File Content Reference later in this article.

   c. Save the changes and preserve the CSV file formatting.

   Repeat this process for the remaining .csv files.

Step 4: Import the correctly mapped files to the target site

After you verify the site mappings in the .csv files, you can import the settings to the new site to complete the migration process.

1. Run the following command on the target Tableau Server machine:

   ```
   tsm sites import-verified --import-job-dir <import-id-dir-
   ectory> --site-id <target-siteID>
   ```

   For example:

   ```
   tsm sites import-verified --import-job-dir /var/-
   opt/tableau/tableau_server-
   /data/tabsvc/files/siteimports/working/import_ff00_
   20180102022014457
   --site-id new-site
   ```

2. When the success message appears, sign in to the new site and confirm that everything was imported as you expected.

Note: The `tsm sites import` and `tsm sites export` commands can leave a site in a locked state if an error occurs. To unlock a site, use the `tsm sites unlock` command.

Mapping File Content Reference

The following tables list the columns in each of the mapping files created when you run the `tsm site import` command.

CSV file name: mappingsDomainMapperForGroups

Column title	Can it be edited?	Description
source_name	No	A user group name on the source site.
source_domain_name	No	The identity store type on the source site: either **local** (for local identity store) or a domain name (for Active Directory or LDAP

		external identity store).
target_domain_name	Yes*	The identity store type on the target site: either **local** for local identity store, or a domain name (such as example.com or example.lan) for Active Directory or LDAP external identity store. *For the **All Users** group, keep the **target_domain_name** value set to **local**, even if your target server is configured for Active Directory identity store. The **All Users** group is a special default user group that must exist on every Tableau Server.

CSV file name: mappingsScheduleMapper

Column title	Can it be edited?	Description
source_name	No	The names of custom and default extract or subscription schedules on the source site.
source_scheduled_action_type	No	The type of schedule, either **Refresh Extract**, for extract refreshes, or **Subscriptions**, for subscription deliveries on the source site.
target_name	Yes	The names of custom schedules on the target site. You can edit this value. For example, if the schedule is named **Friday Update** on the source site you can rename it **Friday Refresh** on the target site.

target_scheduled_action_type	No*	The type of schedule, either **Refresh Extract**, for extract refreshes, or **Subscriptions**, for subscription deliveries on the target site. *In the rare case that you see question marks (???) in this column, replace them with either **Refresh Extract** or **Subscriptions**, to match the entry you see under **source_scheduled_action_type**.

CSV file name: mappingsSiteMapper

Column title	Can it be edited?	Description
source_url_namespace	No	The site ID of the source site.
target_url_namespace	No	The site ID of the target site.

CSV file name: mappingsSystemUserNameMapper

Column title	Can it be edited?	Description
source_name	No	The user name attribute of a user on the source site.
source_domain_name	No	The identity store type on the source site: either **local** (for local identity store) or a domain name (for Active Directory or LDAP identity store), or **external**.
target_name	Yes	The user name attribute for users who will be assigned to the target site upon import.

		Confirm that all the user names in the list exist on the target server, and replace question marks (`???`) with user names that exist on the target server. You cannot create user names by adding rows to the CSV file. Similarly, you cannot remove user names by deleting rows. You can edit a user name in the **target_name** column to be different from its source user name, as long as the user already exists on the target server with that name. For example, a user can have a **source_name** value of **agarcia@company.com** and a **target_name** value of **ashleygarcia@company.com**. You can map a user on the source site to only one user name on the target site.
target_domain_name	Yes	The identity store type on the target site: either **local** (for local identity store) or a domain name (for Active Directory or LDAP external identity store).

CSV file name: MappingsScheduleRecurrenceMapperWithAutoCreation

This file does not require updates.

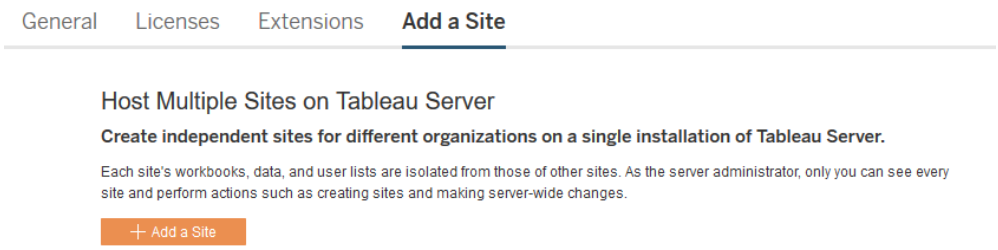
Add or Delete Sites

Tableau Server comes with one site named Default. Server administrators can add or delete sites as the needs of an organization change.

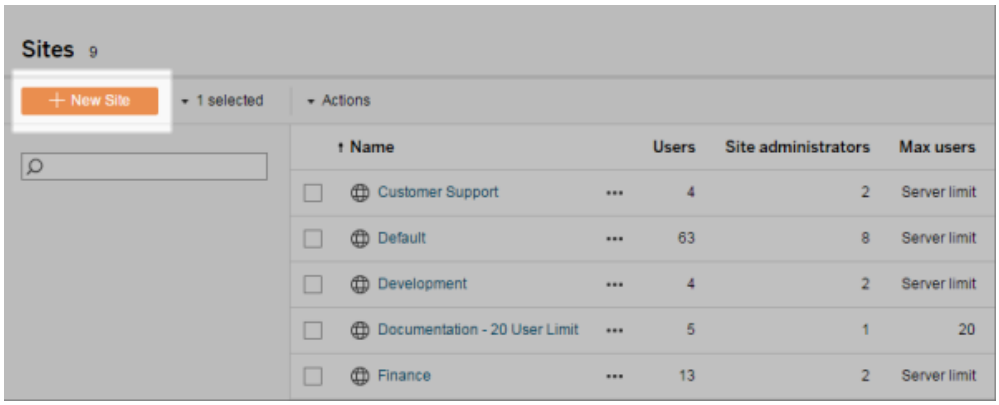
Add a site

1. Do one of the following:

   - If you're adding a site to the server for the first time, select **Settings** > **Add a Site**, and then click **Add a Site**.



   - If you've added sites before, in the site menu, click **Manage All Sites**, and then click **New Site**.



2. Edit the site's settings to customize it for your organization.

Delete sites

Server administrators can delete sites that have been added to Tableau Server. Deleting a site also removes workbooks and data sources that were published to the site, as well as users. If a user belongs to additional sites, they will not be removed. To permanently delete a user, go to the Server Users page.

Note: The Default site cannot be deleted.

1. On the site menu, click **Manage all sites**, and then click **Sites**.
2. Select the site you want to remove, and then on the **Actions** menu, click **Delete**.
3. Click **Delete** in the confirmation dialog box that appears.

Site Availability

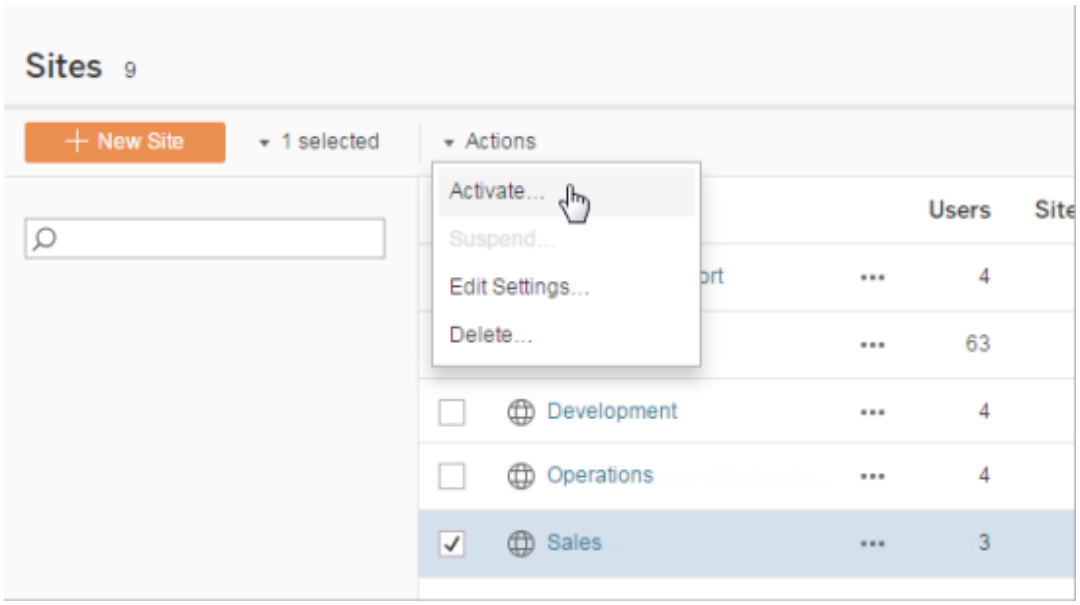
A site can become suspended or locked due to a site import failure, or because a server administrator chooses to suspend the site for a period of time.

When a site is suspended, only the server administrator can activate the site to make it available again.

Note: If a site becomes locked and you cannot access the Sites page through the Server interface, use the tsm sites unlock command to change the state to active.

To activate or suspend a site

1. In the site menu, click **Manage All Sites**, and then click **Sites**.
2. Select the site, and then select **Actions** > **Activate** or **Suspend**.



Manage Site Role Limits

Server administrators can create site role limits to set a maximum number of licenses of each type (Creator, Explorer, or Viewer) that can be consumed on a given site. After a server administrator sets a site role limit, site administrators can add users (thereby automatically consuming licenses) up to that site role limit. If a user is a member of multiple sites across the server, then that user will count against each site's role limit, but will only consume one license (which corresponds to the highest site role they have on the server). Server administrators do not count against site role limits.

To learn more about the capabilities of each site role, see User-based licenses in the Tableau Server help Licensing Overview. For information for server administrators on how to set up site role limits, see Site Settings Reference.

Create role limits on a site

Before you can enable site administrators to manage their users, you can View Server Licenses to determine how to allocate licenses across the Server, or Add Capacity for new users to the Server.

Server administrators can configure site role limits through the Settings page in the web UI, or through the REST API.

To set a site role limit on a site:

1. Go to the General tab of the Settings page for your site.
    - If you have a single site, on the side navigation, click Settings and General.
    - If you have multiple sites, select the site you want to configure and click Settings and General.
2. Under Managing Users, specify that Server and site administrators can add and remove users.
3. Under Limit the number of users to: select Site Role Limit
4. Set a limit for Creators, Explorers, and Viewers.
5. Click Save

Server administrators can set site role limits within the following restrictions:

- A site role limit cannot exceed the number of licenses of that type that have been activated on the server
- A site role limit cannot be less than the number of users of that site role already present for that site
- If a limit is set for any license type, a limit must be set for every license type.
- If the limit for a role is left blank, the server license limit is used.

When site role limits are met

If a user is added to a site as an Explorer, they will consume an Explorer license, unless the site has met its role limit set for Explorer. When this happens, a few different things can occur:

- If there are available licenses at a higher tier, the user will be added to the site as an Explorer, but will consume a Creator license.
- If no higher licenses are available, the user is added to the site as an Unlicensed user.

Site role limits can be affected by users consuming licenses that are different than their given site roles (Viewers consuming Explorer licenses, for example). In this case, administrators can unlicense those users, then update the site role quotas. For more information on how to troubleshoot licensing issues, see Troubleshoot Licensing.

Allow Users to Save Revision History

Revision history enables your users to see how workbooks and data sources (content resources) have changed over time. Each time someone saves (publishes) a content resource, Tableau Server creates a new version, which becomes the current version. It makes the previous version the most recent revision in the revision history list. Revision history gives users confidence to experiment with their content, knowing that their older versions are available.

Notes

- This information applies to Tableau Server, and is for server administrators who want to allow publishers to work with revisions.
- On Tableau Cloud, workbook and data source revision history are enabled on all sites. Users can save up to 10 revisions.
- For information about working with the content revisions themselves, including potential issues, see Work with Content Revisions in the User/Analyst section of the Tableau help.

Permissions users need to work with revision history

To access revision history, a user must have a site role of **Creator** or **Explorer (Can Publish)**, plus the following permissions, depending on the content type:

- Project: **View** and **Save**
- Workbooks in the project: **View**, **Save**, and **Download Workbook/Save As**
- Flows in the project: **View**, **Save**, and **Download Flow/Publish As**
- Data sources in the project: **View**, **Save**, and **Download Data Source**

For virtual connections in the project, you must have a **Creator** site role and the **View** and **Overwrite** permissions. (Virtual connections require Data Management. See About Data Management for details.)

Enable revision history and set the number of revisions allowed

Revision history is set at the site level, and is enabled by default, with a limit of 25 revisions for each content resource.

1. Sign in to a site as a Server Administrator, and click **Settings**.
2. Under **Revision History**, select **Save a history of revisions**, and enter the maximum number of revisions you want to allow for each content resource.
3. Click **Save**.

When you lower the number of revisions, the most recent revisions are saved. For example, if you set the limit to 15, the 15 most recent versions of the workbook or data source are saved.

Note: A content resource's revision history list might not show the changes to the limit until a background cleanup process runs on the server.

Clear all revisions

Server administrators can delete all previous revisions of published workbooks and data sources from a site. The most recent version of each published workbook and data source is always retained.

1. Sign in to a site as a Server Administrator, and click **Settings**.
2. Under **Revision History**, click **Clear Revision History**.
3. Click **Save**.

Security for previewing and restoring workbooks

When users select **Restore** or **Preview** for workbook revisions, user passwords are exchanged between the user's browser and the server. Tableau Server encrypts these passwords using public/private key encryption. To ensure these public keys are provided by

Tableau Server, you must configure the server to use SSL (HTTPS). For more information, see SSL.

See also

Potential revision history issues in the User/Analyst section of the Tableau help.

Tableau Mobile App Security Settings

Starting in , you can adjust Tableau Server security policies for the Tableau Mobile app. These policies help keep your data secure by checking to see if mobile devices are compromised, and by limiting certain interactions with the Tableau Mobile app.

The policies apply to only the standard version of Tableau Mobile, not the MAM versions of the app. If you have deployed an MAM app, use the specific to Tableau Mobile, in addition to your MAM system's settings, to secure the app.

Security settings

Configure security settings for Tableau Mobile either on the site settings page for Tableau Server or using the REST API. For more information about the REST API, see Mobile Settings Reference in the Tableau REST API Help.

To access the site settings page:

1. Sign into your Tableau Server site as administrator.
2. From the navigation pane, select **Settings**.
3. Select the **Mobile** tab.

Starting in Server 2023.1 settings related to mobile device security are available.

These settings include detecting the following conditions:

- **Jailbreak Detection**

  This setting is enabled by default at the **Critical** level and detects whether the app is running on a device that has been jailbroken or rooted.

- **Malware Detection** (Android devices only)

  This setting is enabled by default at the **Critical** level and detects whether the device has malware on it.

- **Maximum Days Offline Without a Policy Refresh**

  This setting is enabled by default at the **Critical** level with a default maximum of 14 days. It determines if the app can be used on a device that has been offline (and thus without a policy refresh) longer than the configured maximum.

- **Prevent Debugging**

  This setting is on by default and cannot be disabled. It detects whether the device has a debugger attached to it.

- **Screen Sharing and Screenshots** (Android devices only)

  This setting is enabled by default and determines whether a Tableau Mobile app user can share screenshots or use screen sharing with the app.

You can change the severity level for the Jailbreak Detection and Malware Detection settings:

- **Warn**: Enforce the policy and if it fails, show a dismissible blocking message.
- **Error**: Enforce the policy and if it fails, show a blocking message until the issue is resolved.
- **Critical**: Enforce the policy and if it fails, show a blocking message and the app decides how to handle the logout/clear session through the providers. This is the default.

Extract Refresh Schedules

Tableau Desktop authors and data stewards can create and publish *extracts*. Extracts are copies or subsets of the original data. Because extracts are imported into the data engine, workbooks that connect to extracts generally perform faster than those that connect to live data. Extracts can also increase functionality.

Before refreshing extracts

When an extract refresh is performed on extracts created in Tableau 10.4 and earlier (that is, a .tde extract), the extract is upgraded to .hyper extract automatically. While there are many benefits of upgrading to a .hyper extract, you will be unable to open the extract with previous versions of Tableau Desktop. Tableau 2024.2 is the last version where any .tde-based content can be opened. For more information, see Extract Upgrade to .hyper Format.

Setting up refresh schedules

As a server administrator, you can enable scheduling for extract refresh tasks, and then create, change, and reassign schedules. General scheduling options you change on the server are available as part of the publishing process when a Tableau Desktop user publishes an extract.

Schedules that you create have the following options:

Priority

The priority determines the order in which refresh tasks are run, where 0 is the highest priority and 100 is the lowest priority. The priority is set to 50 by default.

Execution mode

The execution mode indicates to the Tableau Server backgrounder processes whether to run refreshes in parallel or serially. Schedules that run in parallel use all available backgrounder processes and serial schedules run on only one backgrounder process. However, a schedule can contain one or more refresh tasks, and each task will only use one backgrounder process, whether in parallel or serial mode. This means that a schedule in parallel execution mode will use all **available** backgrounder processes to run the tasks under it in parallel, but each task will only use one backgrounder process. A serial schedule uses only one backgrounder process to run one task at a time.

By default, the execution mode is set to parallel, so that refresh tasks finish as quickly as possible. You might want to set the execution mode to serial (and set a lower priority) if you have a very large schedule that prevents other schedules from running.

Frequency

You can set the frequency to hourly, daily, weekly, or monthly.

For information, see Create or Modify a Schedule.

Refreshing extracts manually

In the Tableau Server web environment, both server and site administrators can run extract refreshes on-demand on the **Schedules** page:

- Select the schedule and click **Actions** > **Run Now**.

You can also refresh extracts from the command line using the tabcmd `refreshextracts` command. For more information, see tabcmd Commands.

Refreshing extracts from Tableau Desktop

Tableau Desktop users can refresh extracts they publish and own. They can do this the following ways:

- **At publish time:** When an author publishes a workbook or data source that uses an extract, that author can add it to server refresh schedule. The refresh can be a full or an incremental refresh.

  Incremental refreshes reference a column in the extract that has a data type of date, date/time, or integer; such as a timestamp. Tableau uses this column to identify new rows that need to be added to the extract. For more information, see Refreshing Extracts and Schedule Extract Refreshes as You Publish a Workbook in the Tableau Help.

- **User interface:** In Tableau Desktop, you can use the **Refresh from Source**, **Add Data From File**, and **Add Data From Data Source** commands to upload an addition to or refresh an extract on Tableau Server. A user might want to do this if Tableau Server doesn't have sufficient credentials to access the underlying data. For more information, see Updating Extracts on Tableau Server in the Tableau Help.

- **Data Extract command line utility:** The Data Extract command line utility installs with Tableau Desktop. You can use it to append to or refresh a published extract. For more information, see Tableau Data Extract Command Line Utility in the Tableau Help.

Enable Extract Refresh Scheduling and Failure Notification

Your publishers can schedule extract refreshes when two conditions are met:

- Tableau Server is configured to send email messages when extract refreshes fail. This is configured by a Tableau Server Manager (TSM) administrator and is on by default. For details, see Configure Server Event Notification.
- The site or sites in which you want to allow publishers to schedule extract refreshes is configured to send email when the refresh fails. This is configured by a server administrator in Tableau Server and is on by default. The instructions below explain how to do this if it is not enabled.

While you're enabling scheduling, you can decide whether also to enable sending email to owners of data sources or workbooks that are refreshed when those extract refreshes do not complete successfully. You can read more about these emails below. When you enable refresh failure notification, the owners of the content that has scheduled refreshes can opt out individually by changing their account settings.

1. Sign in to Tableau Server as a server administrator.

2. Go to the General tab of the Settings page for the site you want to configure for subscriptions:

    - If you have a single site, at the top the browser window, click **Settings** and **General**.

- If you have multiple sites, select the site you want to configure and click **Settings** and **General**.

3. On the **General** page, do the following:

   - Scroll to the **Manage Notifications** settings and check **Extract jobs**.

     If a scheduled refresh for a particular data source fails, the email goes only to the owner of that data source, not to owners of workbooks that connect to that data source.

   - Under **Embedded Credentials**, select both options to let publishers embed credentials and schedule extract refreshes. (Automatic refresh schedules require embedded credentials so Tableau Server can directly access data.)

     > **Note:** On a multi-site server, failure notifications are a site setting, and embedded credentials are a server setting. To configure embedded credentials on a multi-site server, select **Manage All Sites** and then **Settings**.

Managing schedules from the server

In your organization it might be more appropriate to manage embedded credentials and refresh schedules centrally from the server. If you do that, you might clear the check boxes in the **Embedded Credentials** section described in the steps above, so that Tableau Desktop publishers do not see schedule options during publishing.

Managing schedules centrally enables you to distribute extract refresh and subscription tasks, so you can run them when most people are offline. It also enables you to oversee which credentials are embedded in connections.

How refresh failure emails work

The email notification for a failed extract refresh lists the extract name and location on the server, gives the time of last successful refresh, the number of consecutive times the refresh has failed, and suggests the reason for the failure and possible solution.

After five consecutive failures, the refresh schedule is suspended until you or the data owner takes an action to address the cause of the failure, such as updating database credentials or a path to the original data file.

How the last successful refresh date is determined

The last successful refresh date and time are shown when that last refresh occurred within a number of days. By default it is 14 days, and this value is set in `wgserver.alerts.observed_days`. If the number of days since the last successful refresh exceeds the number specified in this setting, the message in the email shows “not in the last *N* days.”

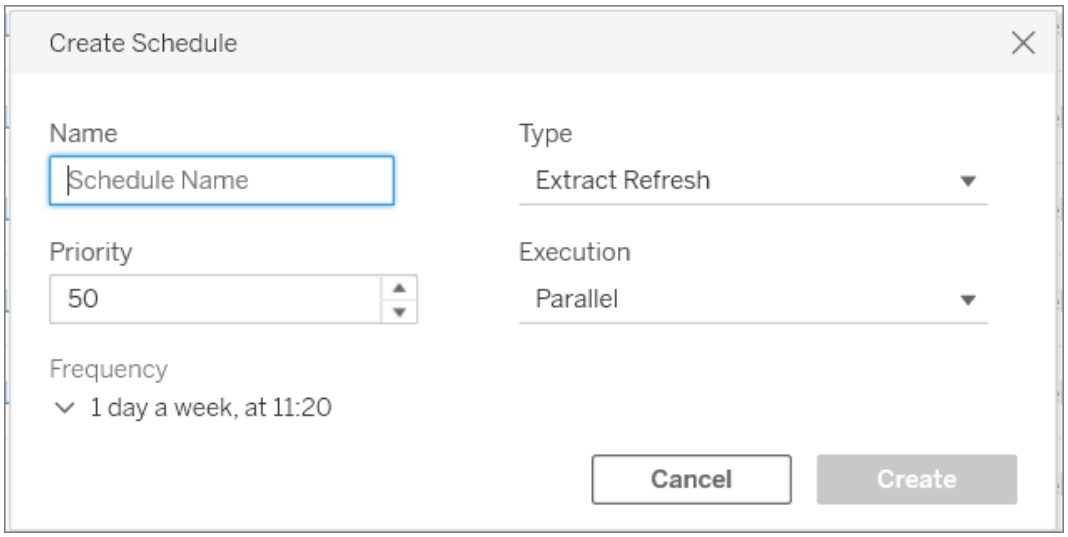
Create or Modify a Schedule

The Schedules page is accessible only by Tableau Server Administrators. It shows a list of schedules, including their name, type, what they’re for (scope), the number of tasks, behavior (concurrent or serial processing), and when they are scheduled to run.

Note: If you enable custom schedules for subscriptions on one or more sites, users will control schedules for their subscriptions on those sites. To learn more, see Enable Custom Schedules for Subscriptions.

To create a new schedule

1. In a site, click **Schedules**.
2. Click **New Schedule**.



3. Specify a descriptive **Name** for the schedule.

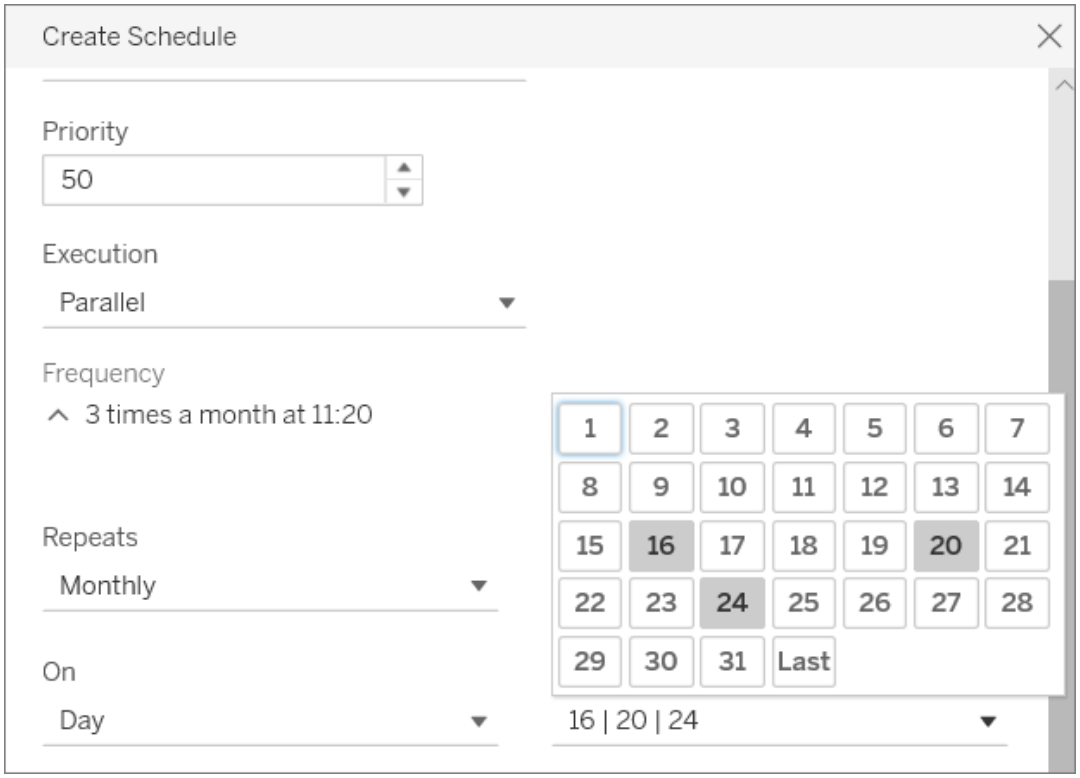
4. Select a **Task type** the schedule will handle—refreshing extracts, running flows, or delivering subscriptions.

5. You must define a priority from 1 to 100, where 1 is the highest priority. This is the priority that will be assigned to the tasks by default. If two tasks are pending in the queue, the backgrounder will evaluate the task priority for extract refreshes and flows, and the schedule priority for subscriptions to determine which one runs first. For more information, see How Scheduled Server Jobs are Prioritized.

6. **Execution**: choose whether a schedule will run in parallel or serially. Schedules that run in parallel run on all available backgrounder processes so that they can complete faster.

   Note: Schedules for the same workbook will always run serially, even if you set this option to parallel.

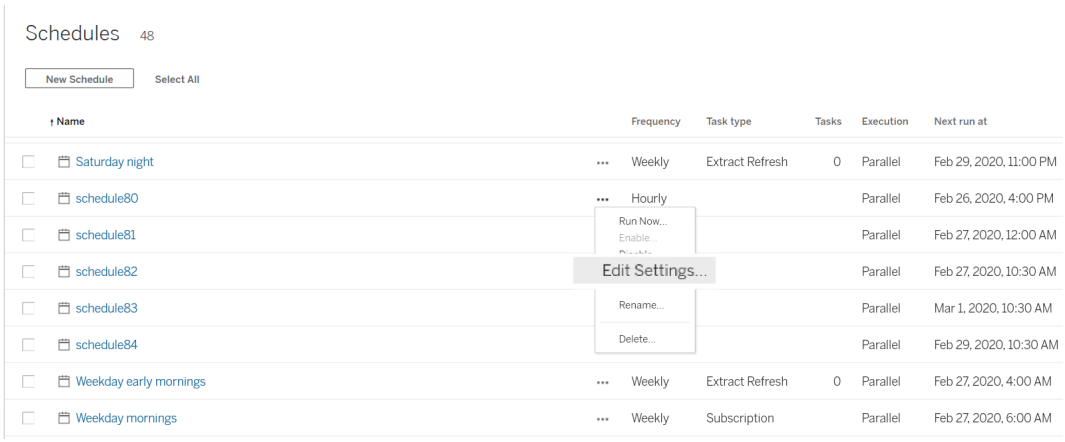
7. Finish defining the schedule. You can define an hourly, daily, weekly, or monthly schedule. The **Frequency** is populated automatically based on the selections you make.



8. Click **Create**.

To modify an existing schedule

1. Navigate to the Schedules page.
2. Select an existing schedule, click the Actions drop-down arrow, and then select **Edit Settings**.



3. Finish editing the schedule, and click **Save**.

Rules for Creating or Modifying Schedules

Following are rules you must follow when creating new or modifying existing schedules.

> **Note:** If you have existing schedules that do not follow these rules, you must modify them accordingly. Not doing so, may result in unexpected behavior and the tasks may not run at the scheduled time.

- Schedules that run every 15 or 30 minutes must have start and end times that are on the hour. Examples of on the hour: 5:00 AM to 6:00 AM.
- Daily schedules on any recurrence must have the same start and end minute. For example, 10:35 am to 4:35 pm. The hour can be different. However, if the daily schedule is set to only happen once a day, it needs only a start time and not an end time.

See also

Manage Refresh Tasks

Extract Refresh Schedules

Schedule Flow Tasks

Enable Custom Schedules for Subscriptions

Custom schedules for subscriptions allow users to receive email messages on a schedule that they define, rather than using a fixed schedule defined by a server administrator. Custom schedules for subscriptions have been available to users of Tableau Cloud since March 2017, and these schedules can now be enabled on a per-site basis in Tableau Server version 2018.2. Enabling custom schedules for subscriptions is a permanent change on any sites where you make this change. Any sites where you don't enable custom schedules remain on fixed schedules defined by a server administrator.

Enable custom schedules

Before you can enable custom schedules on one or more sites, you must first enable custom schedules on Tableau Server, and then enable custom schedules on one or more sites on that server. To learn more about enabling subscriptions on Tableau Server, see Set Up a Site for Subscriptions.

Step 1: Enable custom schedules on Tableau Server

From a command prompt with Tableau administrator permissions, run the following commands:

```
tsm configuration set -k features.SelfServiceSchedules -v true
tsm pending-changes apply
```

This operation will restart Tableau Server.

Step 2: Enable custom schedules on a site

1. Log in to Tableau Server as a server administrator using a web browser:

   ```
   https://<hostname>/#/login
   ```

2. Browse to the **Site Settings** page for a site, and then enable custom schedules:

   1. Click **All Sites**, and then choose one of the sites from the drop-down list.
   2. Click **Settings**.

3. On the **General** tab, under **Subscriptions**, select the following checkbox: **Permanently convert from fixed schedules created by administrators to custom schedules created by users. (You can't undo this)**.
4. Click **Save**.

When custom schedules are enabled on a site, all fixed schedules with a subscription are converted to an equivalent custom schedule.

How Scheduled Server Jobs are Prioritized

Jobs and Tasks

In Tableau Server, users can schedule extract refreshes, subscriptions, or flows to run periodically. These **scheduled items** are referred to as **tasks**. The Backgrounder process initiates unique instances of these tasks to run them at the scheduled time. The **unique instances of the tasks** that are initiated as a result are referred to as **jobs**. **Jobs** are also created for runs that are initiated **manually**, by clicking the **Run now** option.

For example, an extract refresh task is created to run daily at 9 AM. This is an extract refresh task, and every day at 9 AM, a job will be created for the Backgrounder to run.

You can assign a priority number to Tasks and Schedules using values from 1 to 100. Lower the number, higher the priority, 1 is the highest priority, and 100 is the lowest.

Priority Rules for Jobs

When processing scheduled extract refreshes, subscriptions and flow runs, Tableau Server prioritizes background jobs in this order:

1. Any job already in process is completed first.
2. Any task or schedule that you initiate manually using **Run now** starts when the next backgrounder process becomes available. Exception to this are the flow tasks and schedules. Flow runs use the assigned task priority to determine the order in when they should run. If there is no task priority assigned it defaults to 0 which is the highest

priority.

> Note: Learn how to restrict Run now settings- Server Settings (General and Customization).

3. Jobs with the highest priority (the lowest number) start next, independent of how long they have been in the queue.

   For extract refreshes and flows, this is the task priority. The task priority is inherited from the schedule priority when the task is first created. The task priority can be subsequently changed but the task priority returns to the default value when the data source is republished.

   For subscriptions, this is the schedule priority. If you have enabled custom schedules for subscriptions, then the priority of those jobs is set to 50.

   For example, a job with a priority of 20 will run before a job with a priority of 50, even if the second job has been waiting longer. To change task priority, see Create or Modify a Schedule.

4. Jobs with the same priority are executed in the order they were added to the queue. The first job added to the queue starts first; then the second job starts.

5. When multiple jobs with the same priority are scheduled to run at the same time, they start in the order they were created or enabled. Jobs scheduled for the same time are executed by task type with the fastest category of jobs starting first: flow runs, followed by data driven alerts, followed by system jobs, followed by subscriptions, followed by extract creation, followed by incremental extracts, and then full extracts.

   Flows that are scheduled to run as part of a linked task are all assigned the same priority and run in the order they are defined in the linked task. For more information about linked tasks, see Schedule linked tasks.

6. As the last tie breaking measure, the Backgrounder uses the historical run time. Jobs that have run faster in the previous runs, will be prioritized over jobs that have taken longer historically.

Note: Setting backgrounder resource limits on a specific site will have an additional queue as these jobs are picked up after other higher priority jobs. For details, see Tableau Server Backgrounder Resource Limits.

The following limitations also impact when the jobs are run:

- The number of concurrent jobs is limited to the number of backgrounder processes you have configured for Tableau Server.
- Separate refreshes for the same extract or data source cannot run at the same time.
- Jobs associated with a schedule that is set to run serially run one at a time.

Configure Workbook Performance after a Scheduled Refresh

To improve the load times for workbooks, Tableau Server caches the results of queries included in workbooks. For most workbooks, query results are computed and cached when they are first viewed by a user on Tableau Server. However, for workbooks that connect to data extracts, Tableau Server can recompute query results when the corresponding extract refresh tasks run. This reduces the load time for these workbooks when they are first viewed, so this option is turned on by default for workbooks that have been viewed recently.

Important! External query cache warmup will be deprecated in version 2023.1. To improve view load times for workbooks, you should allow View Acceleration on your site instead. For more information, see View Acceleration.

Determine the performance impact

Although this option reduces the initial load time for workbooks, recomputing query results also increases the load on Tableau Server. If your Tableau Server installation is already performance-constrained, you might want to turn this option off or lower the threshold for workbook caching.

Here are some possible reasons why you might want to turn this option off or lower the threshold:

- The Background Tasks for Non Extracts administrative view displays many long-running jobs in the **Warming up external query cache on data change** category.
- The Background Task Delay administrative view displays long delays.
- CPU and memory consumption for the backgrounder processes is consistently high.

However, note that this is only one of the options that impacts the performance of background tasks. For more information about performance, see Performance.

Turn off workbook caching for the server

To decrease the load on Tableau Server, you can turn off workbook caching after a scheduled refresh at the server-level. If you turn this option off, Tableau Server caches query results for workbooks the first time the workbooks are viewed.

Use the following tsm configuration set option to turn off workbook caching after a scheduled refresh:

```
backgrounder.externalquerycachewarmup.enabled
```

For more information on how to use and apply tsm set options, see tsm configuration set Options.

Turn off workbook caching for a site

You can also turn off workbook caching after a scheduled refresh for an individual site. For example, you might do this if there is one site in particular that contains many slow workbooks which increase load on the server.

1. Select the site for which you want to turn off workbook caching in the sites drop-down.
2. Click **Settings**.
3. In the **Workbook Performance after a Scheduled Refresh** section, clear the check box.

Note: Although this option is available in the settings for an individual site, you must have server administrator permissions to view it.

Configure the workbook caching threshold

Tableau Server only recomputes query results for workbooks that both have scheduled refresh tasks and have been viewed recently.

You can increase or decrease the number of workbooks that are cached after a scheduled refresh with the following tsm configuration set option:

```
backgrounder.externalquerycachewarmup.view_threshold
```

By default, the threshold is set to 2.0. The threshold is equal to the number of views that a workbook has received in the past seven days divided by the number of refreshes scheduled in the next seven days. (If a workbook has not been viewed in the past seven days, it is unlikely that it will be viewed soon, so Tableau Server does not spend resources recomputing queries for the workbook.)

Ensure Access to Subscriptions and Data-Driven Alerts

To ensure that users see the Subscribe and Alert buttons in the Tableau Server toolbar and can receive related emails, do the following:

- **Configure SMTP and event notifications on Tableau Server**: See Set Up a Site for Subscriptions.
- **Ensure that users have an email address in Tableau Server**: Users can update their email address on their account settings page.
- **Embed database credentials or don't require them**: To email data in a view, Tableau Server needs to access the data without user involvement. This can be accomplished by using a workbook with embedded database credentials, a Tableau Server data source, or data that doesn't require credentials (such as a file that's included with the workbook at publish time).
- **Ensure that users can access needed workbooks and views**: Access to workbooks and views on the server is controlled by the **View** permission. To receive images of content in email messages, users also need the **Download Image/PDF** permission. For more information, see Permissions.
- **Avoid trusted authentication for embedded views**: If you use restricted tickets (the default) to render an embedded view, the Subscribe and Alert buttons don't appear.

  (Alerts only) Ensure that users can access published data sources with View and Connect capabilities.

Set Up a Site for Subscriptions

When users subscribe to a workbook or view, a snapshot of the view is emailed to them on a scheduled basis, so they can see the latest updates without having to sign into Tableau Server. Administrators, project leaders with appropriate site roles, and content owners have the option to subscribe other users to workbooks and views. For more information, see Subscribe to Views.

Note: To create and receive subscriptions, users need access to related databases and views. See this list of requirements for details.

Looking for Tableau Server on Windows? See Setup a Server for Subscriptions.

Prerequisite: Configure the server to send subscription emails

Before you can enable subscriptions for a site, you need to complete the steps to enable subscriptions on the server. Follow the steps in these topics to configure subscriptions on the server.

1. Configure SMTP Setup
2. Configure Server Event Notification

Enable subscriptions

After you have configured SMTP and server event notifications, you can enable subscriptions.

To enable subscriptions:

1. Sign into Tableau Server as a server administrator.
2. Go to the General tab of the Settings page for the site you want to configure for subscriptions:
    - If you have a single site, on the side navigation, click **Settings** and **General**.
    - If you have multiple sites, select the site you want to configure and click **Settings** and **General**.
3. Scroll to **Subscriptions** and select the subscription options for your users.

> **Note:** Subscription options are only visible after the TSM administrator has enabled the server-wide configuration option, **Allow users to receive email for views that they have subscribed to**. For details, see Configure Server Event Notification.

a. Select **Let users subscribe to workbooks and views**

b. (Optional) To allow content owners to subscribe other users to their content, select **Let content owners to subscribe other users**.

c. (Optional) To allow users to include attachments with their subscriptions, select **Let users add attachments to subscribed workbooks and views**. This option will not be available if the TSM administrator has not enabled attachments in TSM. For details, see Configure Server Event Notification.

4. (Optional) Scroll to **Email Settings**.

   a. Enter an **Email From Address** that will show as the "From" address in email messages.

   b. Enter an **Email Footer** for email messages.

      A site's "From" address and message footer are also used in emails for data-driven alerts.

5. (Optional) Scroll to **High-Visibility Data Labels in View and Workbook Subscriptions** and select **Include high-visibility quality warnings and high-visibility sensitivity labels in view and workbook subscription emails**. (In earlier versions, scroll to **Data Quality Warnings in Subscriptions** and select **Include data quality warnings in subscription emails**.)

> **Note:** Data quality warnings in subscription emails are only visible when Tableau Catalog is enabled. For more information, see Enable Tableau Catalog.

6. Click **Save**.

To specify the subscription schedules available to users, see Create or Modify a Schedule.

Test subscriptions in a site

1. Subscribe to a view.
2. In the site with the subscription you want to test, on the side navigation, click **Schedules**.
3. Select the schedule you chose for the subscription, and then click **Actions** > **Run Now**.

   A snapshot of the view should be emailed to you within 10 minutes. If you experience an issue, see Troubleshoot Subscriptions.

Manage all user subscriptions

1. In the side navigation, click **Tasks**, and then click **Subscriptions**.

   All user subscriptions for the current site appear, including information like subscriber name, view name, and delivery schedule.

2. Select any subscription you want to update. From the **Actions** menu, select **Change Schedule**, **Change Subject**, **Change Empty View Mode**, or **Unsubscribe**.

   (The empty-view option sends subscription emails only when data exists in a view. It's a good choice for high-priority alerts.)

Suspended Subscriptions

By default, a subscription is suspended after 5 consecutive subscription failures and result in the subscription emails not sent. To change the threshold number of subscription failures that can occur before they are suspended, use the tsm configuration set option, backgrounder.subscription_failure_threshold_for_run_prevention. This sets the threshold for the number of consecutive failed subscriptions necessary before suspending the subscription. This is a server-wide setting.

Only Server administrators can configure the threshold number of subscription failures before a subscription is suspended.

Server administrators can opt in to receive email notifications when a subscription is suspended. You can do this by navigating to **My account settings -> Subscription Notifications**. This setting is at a site-level so has to be configured for site separately.

Resume suspended subscriptions

If a subscription fails more than five times, you'll receive a notification email that your subscription has been suspended. There are a few ways to resume a suspended subscription if you're a subscription owner or administrator:

- From the **My Content** area of Tableau web pages, an icon appears in the Last update column to indicate that the subscription is suspended. **Select ... > Resume Subscription** to resume.
- From the **Subscriptions** tab of the affected workbook, an icon appears in the last update column to indicate that the subscription is suspended. **Select ... > Resume Subscription** to resume.
- From the **Subscriptions** tab under **Tasks**, an icon appears in the last update column to indicate that the subscription is suspended. **Select ... > Resume Subscription** to resume (Server administrators only).

When a subscription is resumed, the alert failing count goes back to zero. The next evaluation of the subscription will occur at the next scheduled evaluation time.

See also

Subscribe to Views in the Tableau Desktop and web authoring Help.

Project-level administration to learn which site roles allow full Project Leader capabilities.

Set Up for Data-Driven Alerts

When data reaches important thresholds for your business, data-driven alerts automatically send email notifications to key people users specify. As a Tableau Server administrator, you set up data-driven alerts much like you do subscriptions. For information about how users create and manage these alerts, see Send Data-Driven Alerts in Tableau User Help.

> **Note:** To create and receive data-driven alerts, users need access to related databases and views. See this list of requirements for details. If alerts are enabled for a site, any user on that site can create them except users with the Viewer role.

Configure email for data-driven alerts

1. Complete the steps in Configure SMTP Setup so the server can send the email.
2. While viewing a site, click **Settings** at the top of the browser window.
3. Under Email Settings, enter a site-specific "From" address or message footer.

   A site's "From" address and message footer are also used in emails for subscriptions.
4. Click **Save**.

Manage all data-driven alerts in a site

1. At the top of the browser window, click **Tasks**, and then click **Alerts**.
2. Select any alerts you want to update.
3. From the **Actions** menu, do any of the following:
   - Add or remove yourself as a recipient.
   - Edit alerts to change data thresholds, delivery schedules, and the full list of recipients.
   - Change alert ownership to different users, or delete alerts.

Disable data-driven alerts for a site

Data-driven alerts are supported for all sites by default, but administrators can disable them for specific ones.

1. While viewing a site, click **Settings** on the left-side navigation pane.
2. Under Data-Driven Alerts, uncheck **Let users create alerts and receive alert emails**.
3. Click **Save**.

Suspend data-driven alerts

By default, an alert is suspended after 350 consecutive alert failures. Server administrators can configure the threshold number of alert failures before an alert is suspended. To change the threshold number of data-driven alert failures that can occur before alerts are suspended, use the tsm configuration set option, `dataAlerts.SuspendFailureThreshold`.

This sets the threshold for the number of consecutive failed alerts necessary before suspending the alert. This is a server-wide setting. The threshold value applies to every configured data-driven alert on the server.

Resume suspended alerts

If an alert fails enough times, you'll receive a notification email that your alert has been suspended. There are a few ways that administrators or alert owners can resume a suspended alert:

- From the Tasks > Alerts area of Tableau web pages, an  icon appears in the Last checked column to indicate that the alert is suspended. Select **...** > **Resume Alert** to resume the alert.
- Click **Resume Alert** in the notification email to resume the alert. A notification will either allow you to resume the alert, or indicate that the view has changed and the alert should be deleted.
- From the Alerts panel of the affected view or workbook. To resume the alert from a view or workbook, select **Alert** to open the Alerts panel. An icon appears next to the suspended alert. Select **Actions** > **Resume Alert** on the affected alert to resume.

Alert owners will receive an email notification when the alert is working again.

Control how often the server checks data-driven alerts

By default, Tableau Server checks every 60 minutes to confirm whether data conditions for alerts are true. If you notice performance impacts, you can customize this time interval with

the tsm configuration set option, `dataAlerts.checkIntervalInMinutes`.

Regardless of the `dataAlerts.checkIntervalInMinute` setting, the server also checks alerts whenever extracts in the related workbook are refreshed. To check an alert more frequently than the setting specifies, change the extract-refresh schedule.

Track the server's alert-checking process

In the Background Tasks for Non Extracts view, you can track the server's alert-checking process by looking for these tasks:

- Find Data Alerts to Check
- Check If Data Alert Condition Is True

The "Find" task limits "Check" tasks to alerts that can currently send related emails. For example, if a user has chosen an email frequency of "Daily at most", after the alert condition becomes true, the server waits 24 hours before checking the alert again.

Each "Check" task uses one server backgrounder process, loading the related view to evaluate the alert condition. If all users see the same version of a view, it loads only one time. But if users have applied filters to a view, or the data they see is limited by user-level security, the view loads once for each recipient.

Identify and fix failing alerts

As an administrator, you can proactively identify failing alerts that users may be unaware of. To check:

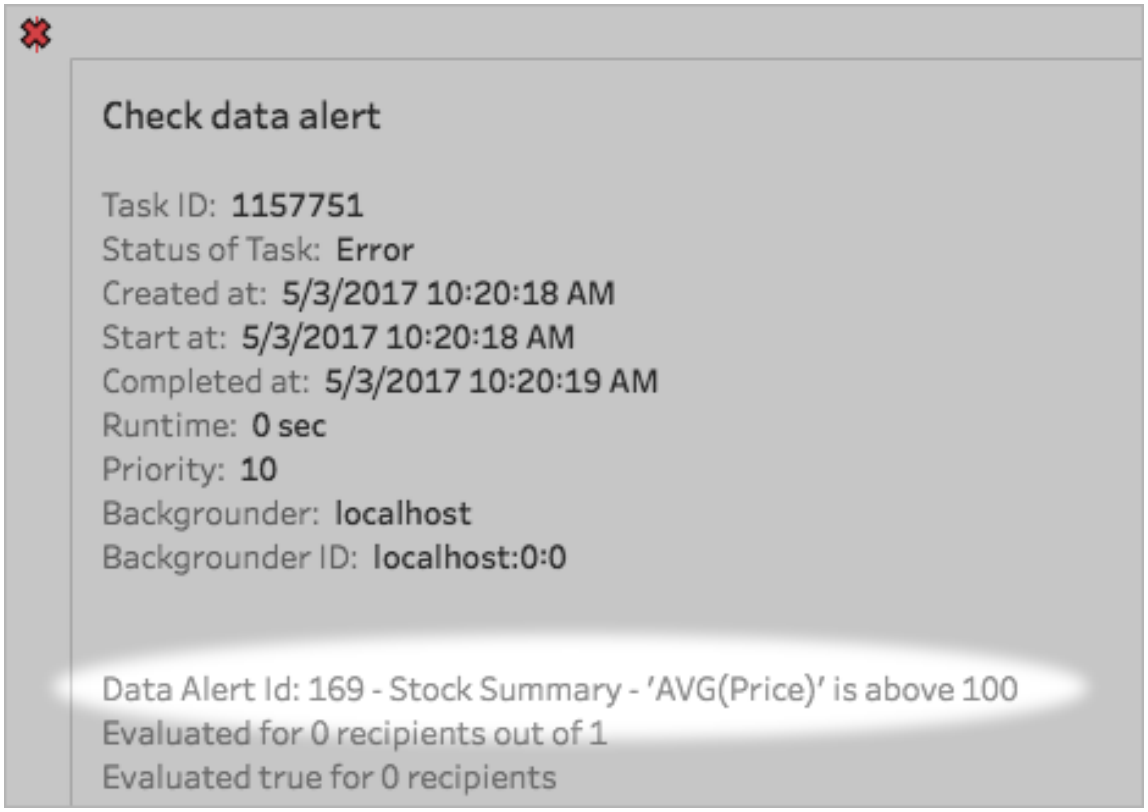
1. Select Status in your site menu.
2. Select Background tasks for non-extracts.
3. From the Task dropdown menu, select Check if Data Alert is True.

4. In the far right, click Error to see a list of failing alerts.

5. Hover over the red failure icon to display a tooltip with alert details.

To determine the alert owner, look for the alert ID number in the data_alerts table of the Tableau Server Repository. (In the alert management area of a site, you can also look for the alert name following the number, but be aware that multiple different alerts may use the same name.)

Note: Alert owners are automatically notified when an alert fails ten times. Administrators can customize when alert owners receive notifications. Users won't be notified for alerts that failed prior to upgrading to Tableau Server 2018.1.



Failing alerts are often caused by content changes on Tableau Server. Encourage users to recreate alerts if changes like the following occur:

- A workbook, view, or data field is removed or renamed.
- Database credentials embedded in workbooks expire.
- Data Driven Alerts require embedded credentials for Live Connections, the use of OAuth isn't currently supported with Alerts.
- A data source becomes inaccessible.

Tip: To automatically get emailed when alerts fail, follow the steps in Collect Data with the Tableau Server Repository, and connect to the "background_jobs" table. From that table, create a custom view that includes the "Check If Data Alert Condition Is True" job name and its finish code. Then set up a data-driven alert to email you whenever a finish code equals 1 (failure).

Set Up for Metrics

Retirement of the legacy metrics feature

Tableau's legacy metrics feature was retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2. In October 2023, Tableau retired the ability to embed legacy metrics in Tableau Cloud and in Tableau Server version 2023.3. With Tableau Pulse, we've developed an improved experience to track metrics and ask questions of your data. For more information, see Create Metrics with Tableau Pulse to learn about the new experience and Create and Troubleshoot Metrics (Retired) for the retired feature.

Metrics are a type of Tableau content that tracks the value of an aggregate measure, such as sum of sales. Because metrics refresh frequently and display their current value in an easy-to-glance format, they are useful for monitoring data. To learn more about how users work with metrics, see Create and Troubleshoot Metrics (Retired).

As a Tableau Server administrator, you have the ability to control how often metrics refresh and how failing refreshes are handled. You also can make sure that users are able to create metrics or disable metrics for particular sites or for the entire server.

Ensure that users can create metrics

When metrics are enabled for a site, all users with a Creator or Explorer (can publish) site role can create metrics, if they have the correct permissions.

Metrics are created from existing views on a Tableau site. To ensure that users can create metrics on a view, verify that:

- Users have the Create/Refresh Metrics permission capability for the workbook that the view belongs to. For more information, see Permissions.
- The password for the data source is embedded, if it is required. For more information, see Edit Connections on Tableau Server.

Disable metrics for a site

Metrics are enabled on all sites by default. You can disable metrics on a per-site basis.

1. On the site where you want to disable metrics, from the navigation panel, click **Settings**.
2. Under **Metrics Content Type**, uncheck **Enable metrics**.
3. Click **Save**.

When you disable the metrics content type, metrics no longer appear on the site. The data for any existing metrics is retained, but these metrics will no longer refresh. If you re-enable metrics, these metrics will reappear and resume refreshing.

You can also disable metrics on a specific workbook by denying the Create/Refresh Metrics permission capability. For more information, see Permissions.

Disable metrics for a server

In addition to disabling metrics for specific sites, you can disable metrics server-wide. When disabled at the server level, metrics don't refresh or appear on any sites, metrics processes

don't run, and the site settings for metrics aren't available. Existing metric data is retained, so that if you re-enable metrics, those metrics will be restored.

Metrics are enabled by default. To disable metrics, use the tsm configuration set option `metricsservices.enabled`.

Configure how often metrics refresh

When a metric refreshes, it checks for new data via the view it was created from, known as the connected view. You might want to increase the time between refreshes if you notice a performance impact on your server—or decrease it if your users require more up-to-date data.

Metrics that rely on live data refresh every 60 minutes, by default. To adjust the refresh interval for live data, use the tsm configuration set option `metricsservices.checkIntervalInMinutes`. This is a server-wide setting.

Metrics that rely on extract-based data refresh when the extracts refresh. To control how often these metrics refresh, change the extract refresh frequency. For more information, see Extract Refresh Schedules.

Configure failure notifications for metric refreshes

If a metric is not able to connect to the data it needs to refresh, the refresh will fail. When a metric refresh fails 10 times in a row, the metric owner receives an email notification.

To adjust the number of consecutive failures before a warning email is sent, use the tsm configuration set option `metricsservices.failureCountToWarnUser`. This is a server-wide setting.

Configure when metric refreshes are suspended

If a metric refresh fails 175 times in a row, the refresh is suspended. Once a metric refresh is suspended, the server will no longer attempt to check for new data, until the refresh is manually resumed.

To adjust the number of consecutive failures before a refresh is suspended, use the tsm configuration set option `metricsservices.maxFailedRefreshAttempts`. This is a server-wide setting.

Manage metrics

Though metrics are created from a view, they are not tied to the view like alerts or subscriptions. This means you can manage metrics similar to how you manage workbooks, by renaming, moving, tagging, deleting, or setting permissions on a metric.

Find metrics to manage either by navigating the project hierarchy or via the following paths.

- To see all metrics on a site: Navigate to the Explore section, then select **All Metrics**.
- To see metrics created from all the views in a workbook: Navigate to the workbook, then select the **Connected Metrics** tab.
- To see metrics created from a single view: Open the view, then select **Watch** > **Metrics** from the toolbar.

Address failing and suspended metric refreshes

Metric refreshes may fail for one of the following reasons.

- The connected view was deleted or modified.
- Permissions changed for the connected view.
- The password for the data source is no longer embedded or is no longer valid.
- The metric owner doesn't have the required site role to refresh the metric. A site role of Creator or Explorer (can publish) is required.
- There was a temporary connectivity issue, which will resolve itself.

> **Note:** If the metric refresh is suspended because the owner doesn't have the required site role for it to refresh, you won't be able to resume the refresh unless you change the owner.

For more information on why metric refreshes fail and what users can do to fix them, see Fix failing refreshes.

Encourage users to overwrite a metric if the connected view was modified in a way that caused the refresh to fail, but the view is still available. Users can overwrite a metric by creating a metric with the same name in the same project as the existing metric.

Resume suspended refreshes

If the cause of the failure is fixed, for example by embedding the correct password for the data source, you can resume the metric refresh.

1. Locate the affected metric. Metrics with suspended refreshes display the text **Refresh Suspended**, instead of the time of last refresh, in grid and list view.
2. On the warning message, click **Resume refresh.**

Tableau attempts to perform the refresh. If this attempt succeeds, you'll receive a confirmation, and the refresh will resume on schedule. If the attempt doesn't succeed, the refresh remains suspended. You or the metric owner can delete or overwrite the metric, or keep it to reference historical data.

Monitor metric activity with administrative views

Use the administrative views for Tableau Server to monitor metric refreshes and see which users are creating and viewing metrics.

1. Navigate to the site you want to monitor, or monitor server-wide activity by selecting **All Sites** from the site picker.
2. From the navigation panel, click **Site Status** or **Server Status**.
3. Select the dashboard you want to inspect.

    - To monitor metric refresh activity, open the Background Tasks for Non Extracts dashboard.

      Filter for the tasks **Find Metrics to Update** or **Update All Metrics on a View**.

    - To see which users are creating metrics, open the Actions by All Users or Actions by Specific User dashboard.

      Filter for the action **Create Metric**.

- To see recent activity involving metrics, open the Actions by Recent Users dashboard.

  Look at the list of actions under **What Actions Were Recently Performed?**

Edit a Published Data Source

Imagine that you've published a data source, and your team is using the data source across a number of workbooks. This is a good start, but you have some changes in mind that will make your data source great. Before you implement these changes, you want to see how your proposed changes look in Tableau. And most importantly, you need to test your changes to ensure they won't negatively impact any existing workbooks that use the data source.

Editing a published data source allows you to test changes and make improvements to your data source while maintaining it as a single source of data.

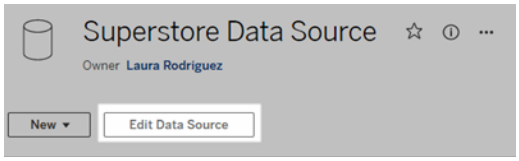
> **Note**: Only users with a site role of Creator can edit publish data sources in the browser.

Edit and test changes

Whether you're creating a new published data source or editing an existing published data source, you can create joins and edit the schema from the Data Source page without leaving your browser. Then use the Scratchpad to test your changes, create folders, organize hierarchies, and rename fields and aliases before publishing your data source. While editing your data source, you'll have all the same features and functionality that you have when authoring in Tableau Cloud. For more information, see Web Authoring and Tableau Desktop Feature Comparison.

To edit a published data source:

1. From the Start or Explore page, navigate to the data source you want to edit.
2. Click **Edit Data Source**.



3. Click the **Data Source** page to make joins or edit the schema.
4. Click the **Scratchpad** sheet.
5. From the **Data** pane, create folders, organize hierarchies, rename fields and their aliases, or update metadata that are saved with the published data source.
6. Drag and drop fields onto the scratchpad to make sure your changes are working as expected.
7. Click **Publish**.

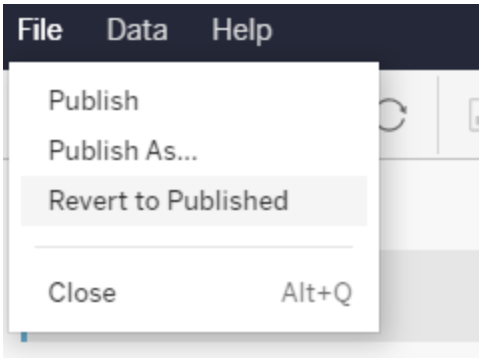
Just like you can with workbooks, you can also **Publish As** if you want to make a copy of the data source.

Note: Personal Spaces don't support published data sources.

Roll back changes

To revert to the last version of the published data source:

1. Navigate to the data source that you want to revert.
2. Click **File**.
3. Choose **Revert to Published**.



This reverts to the latest published version of that data source.

Understand supported connections

Editing published data sources doesn't support:

- Tableau Bridge connectors in Tableau Cloud.
- Data sources that use embedded passwords in Tableau Cloud and Tableau Server.

Also, the Data Source page isn't available for published data source connection types that aren't supported, including but not limited to .hyper file types. To see which connection types are supported, see Creators: Connect to Data on the Web.

Learn about permissions

To edit a published data source, you'll need a Creator license that has Save or Save As permissions for data sources in the respective folder. For more information, see Permissions.

Edit data sources published by a flow

If you make edits to a data source that was published by a flow, the changes will be overwritten during the next scheduled flow. Instead, edit the data source in the flow. For more information, see Publish a Flow to Tableau Server or Tableau Cloud.

Managing Background Jobs in Tableau Server

In Tableau Server, users can schedule extract refreshes, subscriptions, or flows to run periodically. These scheduled items are referred to as **Tasks**. The Backgrounder process initiates unique instances of these tasks to run them at the scheduled time. The unique instances of the tasks that are initiated as a result are referred to as **Jobs**. Jobs are also created for runs that are initiated manually, by clicking the **Run Now** option in the web interface, programmatically through REST API, or tabcmd commands.

For example, an extract refresh task is created to run daily at 9 AM. This is an extract refresh task, and every day at 9 AM, a job will be created for the Backgrounder to run. In addition to

user-generated jobs, the Backgrounder also does a number of System jobs on behalf of the user to support general Tableau work flows, such as thumbnail generation.

Running all these jobs can mean that Backgrounder uses a lot of resources at various times during the day. Using the Job Management feature, Server and Site administrators can get more details on these jobs that happen in their Server or Site, and take action on those jobs to better manage server resource usage. System jobs are only viewable by Server administrators, and by default are filtered out..

The **Run Now** settings on the **General** settings page also allows you to manage your resources by either allowing or blocking users from running jobs manually. By default, this option is selected to allow users to run jobs manually. Clear the check box to prevent users from running jobs manually. To learn more about managing Backgrounder resources, see Tableau Server Backgrounder Process.

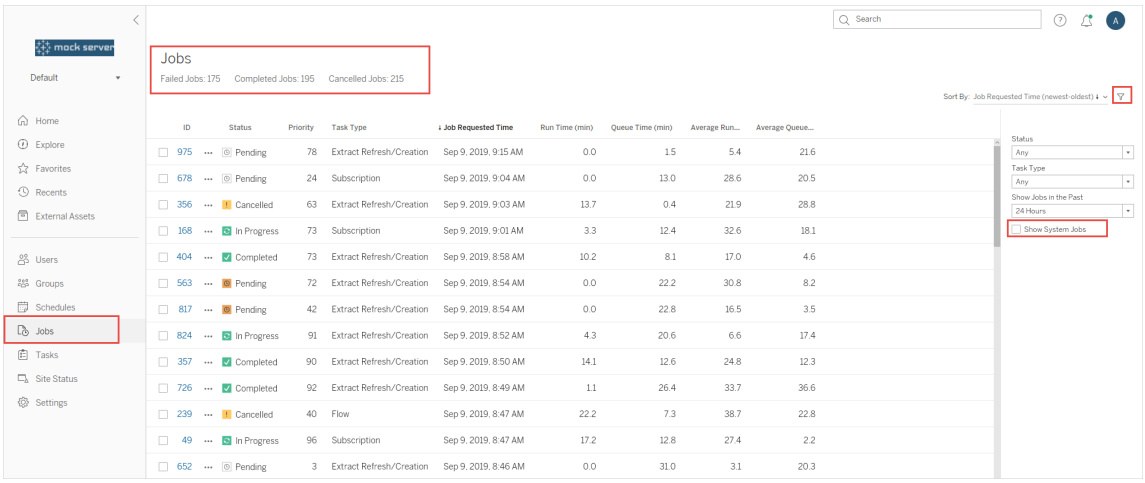
The Jobs page which contains the information about jobs can be accessed by navigating to the **Existing Tasks** menu of the left navigation menu.

Note: Information about jobs can only be viewed by Server and site administrators.

Overview

This topic describes how to view and understand the information displayed in the Jobs page.

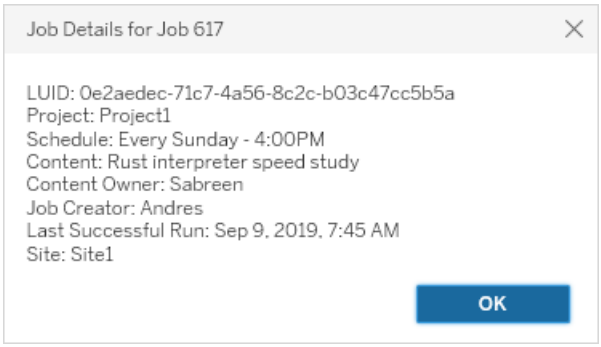
At the top of the page there are high level statistics for the number of **Failed**, **Completed**, and **Canceled** jobs within the past 24 hours. For Server administrators, this also includes System jobs. Applying filters do not change these values.



For each job generated, there is a Job ID, the status of that job, the priority, the type of task that the job was generated from, the current run time - if the job is in-progress, current queue time - if queued, as well as the average run time, and average queue time.

Tableau records historical run times and queue times to compute the average run times and average queue times. Both average run times and average queue times are calculated as weighted averages using the following formula: ((current run time or queue time average x 4) + most recent run time or queue time)/ 5.

The Job ID can be useful when viewing jobs on **Admin views** and can also be used to query the Workgroups Database. When you click on the Job ID, you will see more detailed information about the job, such as the Job LUID, the project name, the schedule , the content name, content owner, job creator, the last time the job ran successfully, and the site name. The site name is displayed if you navigate to the Jobs page using the **Manage All Sites** menu.



Note: Doing a **Refresh Now** from the **Data Sources** page will only show the LUID information in the **Job Details** dialog box.

Important: Jobs that existed 24 hours or newer before an upgrade to Tableau Server 2019.4 will not have data for **Average Queue Time**, **Average Run Time**, **Last Successful Run Time**, and **Job Creator** on the **Jobs** page.

Task Types

There are several types of tasks:

- Extracts: This includes extract creation, incremental extract refreshes, and full extract refreshes. For more information on extract refreshes, see Quick Start: Refresh Extracts on a Schedule.
- Subscriptions: Includes subscriptions for workbooks and views. For more information, see Set Up a Site for Subscriptions.
- Flow: This includes scheduled flows and manual flow runs. See Job runtime capacity for information about the maximum runtime for flows and Concurrent jobs capacity for capacity limits when running concurrent flow jobs. Scheduling more flows than the number of resource blocks you have can result in an error. See the Knowledge Base article Flow Job Pending for more information.
- Encryption: Includes the following:
  - Extract encryption and decryption
  - Flow encryption and decryption
  - Re-key extracts and flows
- System: This is all system Jobs that the Backgrounder handles behind the scenes to support Tableau Server.

Filters

You can filter to see only certain jobs. The available filters are by Job Status type, Task Type, and Time Range. For the Time Range filter, you can choose from past one to 24 hours, in four

hour increments. The option to filter in System Jobs is available if you are a Server Administrator.

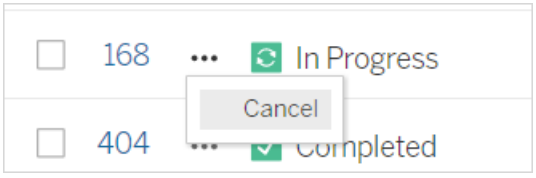
Canceling Jobs

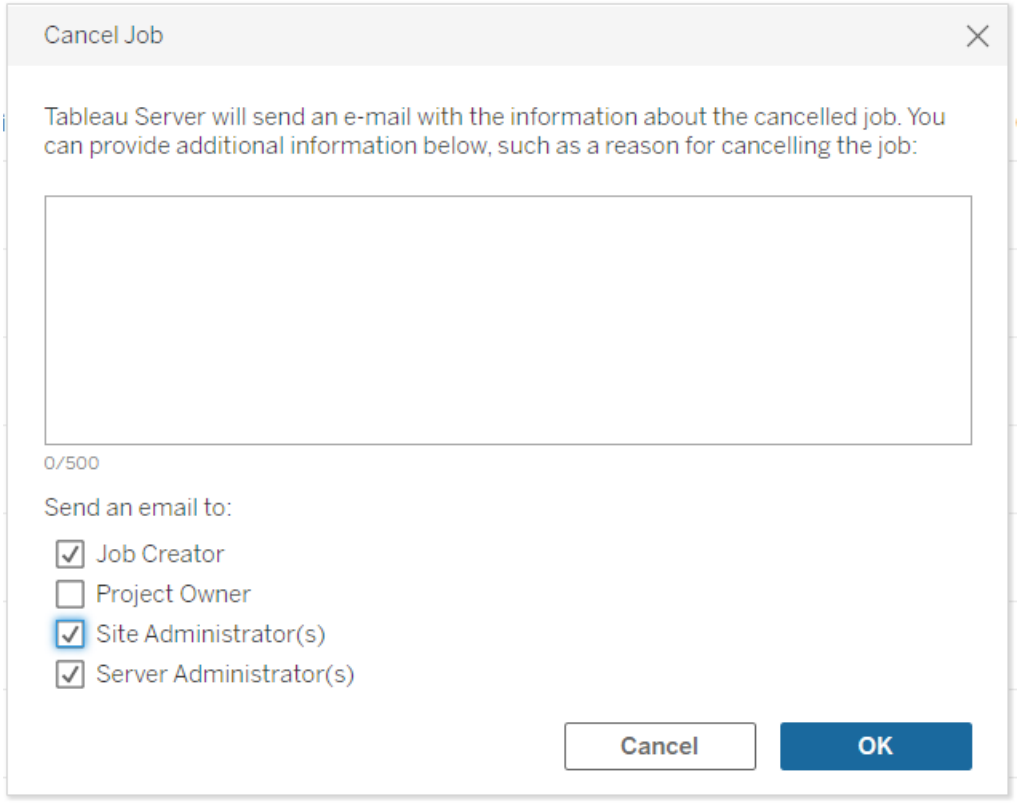
Extract refreshes, subscriptions and flow run jobs can be canceled. You can only cancel one job at a time, and selecting multiple jobs at one time for cancellation is not supported.

When you cancel a job, an email with the time the job was canceled, the affected content, and the time the job ran before being canceled is sent to the recipients that you select in the **Cancel Job** dialog box . In addition you can add your customized notes to be included in the email.

If you do not select any recipients, the job will be canceled, but no email will be sent.

To cancel a job, click on the ellipses next to the Job ID and use the dialog to cancel the job:

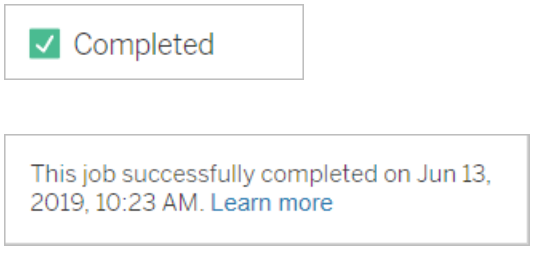




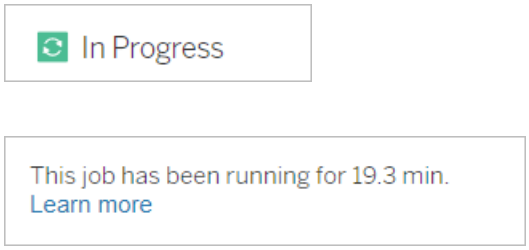
Status

There are seven types of status that jobs can be in, and hovering over each status will display more relevant information.

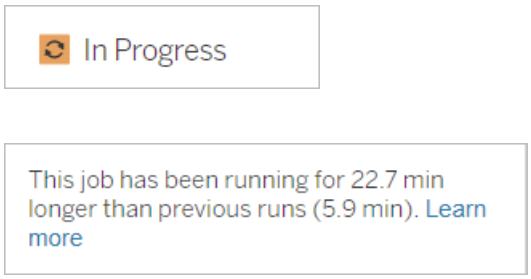
- **Completed:** This job shows as **Completed successfully** and you can see the time when the job completed in the tooltip that is displayed when you hover over the status.



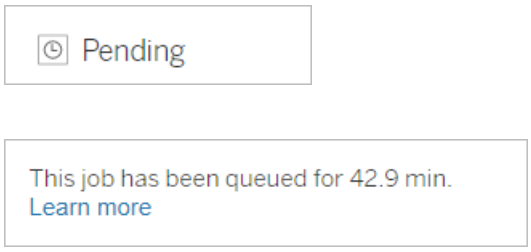
- **In Progress:** This job shows as **In Progress**. A time for how long the job has been running for is displayed in the tooltip when you hover over the status.



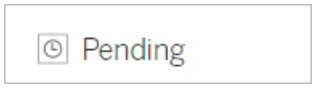
- **In Progress:** This job is **In Progress**, but is **running late**. Tableau keeps track of the average run times for the same job, and if the current run time is longer than the average run time, then it is considered running late. Times for how much longer than average the job has been running and its average run time is provided in the tooltip that is displayed when you hover over the status.

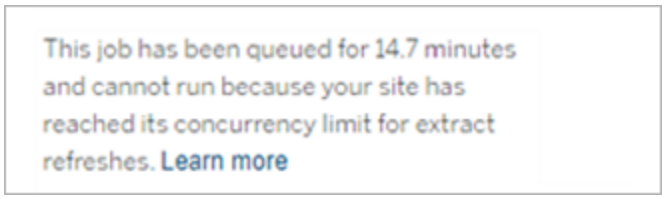


- **Pending:** This job is currently **Pending**, waiting to be run when there is available Backgrounder capacity. A time for how long the job has been in the queue for is provided in the tooltip that is displayed when you hover over the status.

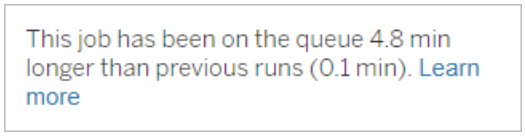


- **Pending:** The flow cannot be run because the number of Resource Blocks is less than the number of flow you have scheduled and the site has reached its concurrency limit.

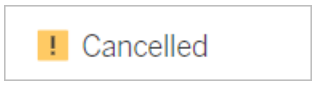


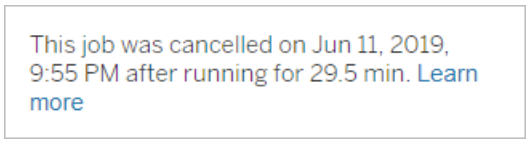


- **Pending:** This job is currently **Pending**, but is **running late**. Tableau keeps track of the average queue times for the same job, and if the current queue time is longer than the average queue time then it is considered running late. Times for how much longer than average the job has been queued is provided in the tooltip that is displayed when you hover over the status.

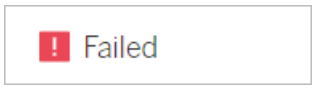


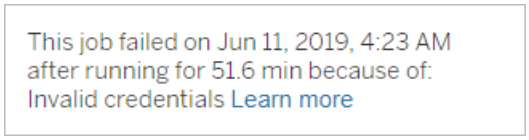
- **Canceled:** This job was **Canceled** by a Server or Site administrator. The time the job was canceled and how long it ran for before cancellation is provided in the tooltip that is displayed when you hover over the status.



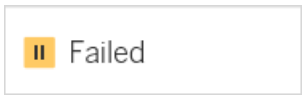


- **Failed:** This job is showing as **Failed**. The time when the job failed, how long it ran for before it failed, and why the job has failed is provided in the tooltip that is displayed when you hover over the status.





- **Suspended:** This job is showing as **Failed** with a pause icon. If the job fails 5 times consecutively, then the job is suspended. Suspended tasks are still available but Backgrounder will not create jobs for these tasks until they are resumed by the user.



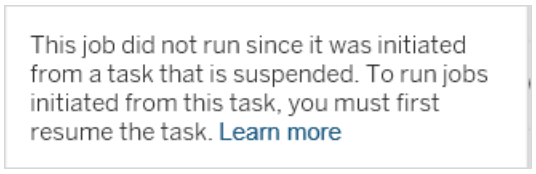


Tableau Service Manager Jobs

TSM jobs are administrative tasks that help configure and maintain Tableau Server. These jobs run by Tableau Services Manager.

Here are some key TSM jobs:

- **Cleanup**: This job is created when a cleanup command is issued to Tableau Server. The cleanup command deletes old log files and temporary files. For more information on the TSM CLI cleanup command, see tsm maintenance.

- **Deployments**: This job is created to apply any configuration updates that you make to Tableau Server. This can be initiated either through the TSM web interface or TSM CLI. For more information on the TSM CLI command, see tsm pending-changes. Here are a few examples of configuration updates: enabling SSL, enabling Run as User, and server topology changes.

  Depending on how many topology changes are being made, and the complexity, this job can take longer than the previous time it ran successfully. For example, if the previous change was a hot topology change, and the current one is not, the current job can take longer to complete than the previous one.

- **Generate backup**: This job is created when a backup command is issued to Tableau Server. The backup command creates a backup file of Tableau data (data in the File Store and repository). For more information on the TSM CLI backup command, see tsm maintenance backup.

  The time taken for the backup job to complete depends on the amount of data that needs to be backed up. If the amount of data to be backed up has increased from the last time this job was run, it will take longer for the job to complete compared to the previous time.

- **Initialize Tableau Server**: This job is initiated to initialize Tableau Server during the installation process. For more information on the TSM CLI command, see tsm initialize
- **Restore**: This job is created when a restore command is issued to Tableau Server. The restore command restores a Tableau Server data backup file. For more information on the TSM CLI restore command, see tsm maintenance. The time taken for the restore job to complete depends on the size of the backup file. If the backup file is larger than the previous times, the restore job will take more time to complete.

- **Start Server**: The job is created to start all the stopped Tableau Server processes. This can be initiated either through the TSM web interface or TSM CLI. For more information on the TSM CLI command see, tsm start.
- **Stop Server**: This job is created to stop all the running Tableau Server processes. This can be initiated either through the TSM web interface or TSM CLI. For more information on the TSM CLI command, see tsm stop.

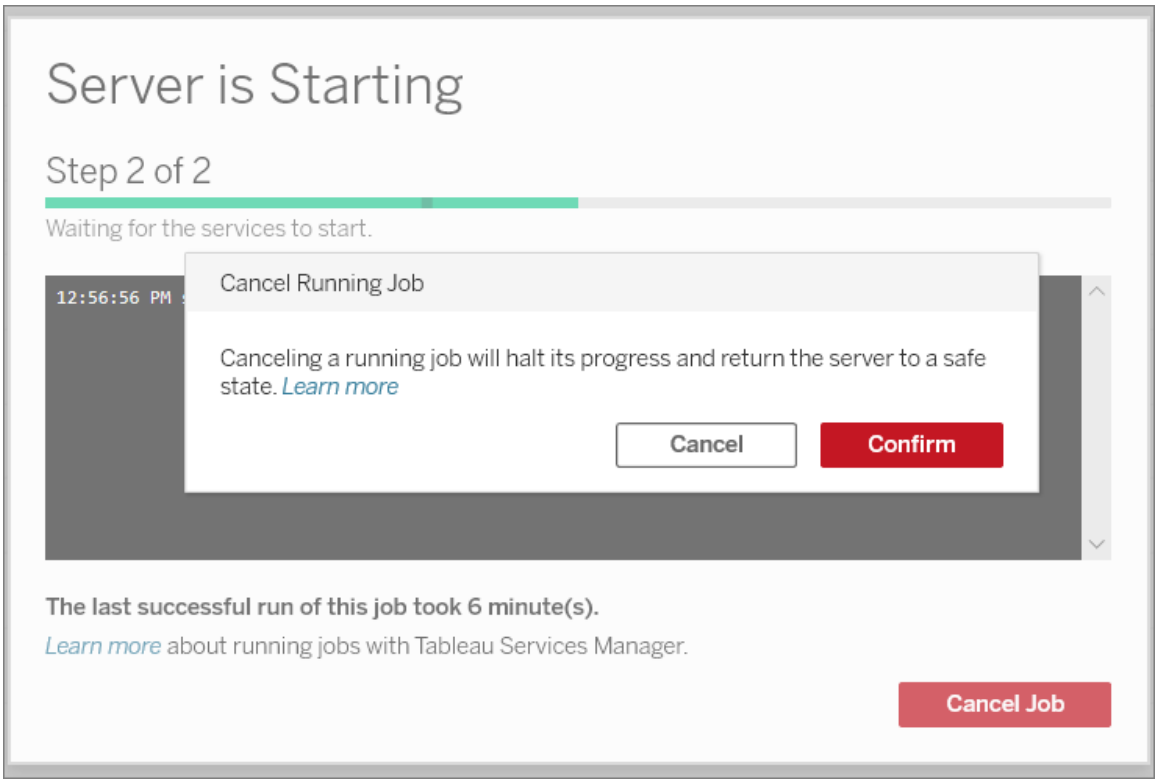
Canceling tsm Jobs

You can cancel jobs either through the TSM web interface or using the TSM CLI. For more information, see Cancel TSM Jobs.

Cancel TSM Jobs

TSM Jobs can be canceled through TSM web interface or TSM CLI. There are many TSM jobs, but only certain jobs can be canceled once they are in progress. Any job that hasn't yet started can be canceled using TSM CLI.

TSM web interface: For jobs that can be canceled while they are running, the Cancel option is available in the Job dialog box as shown below:



TSM CLI: To cancel jobs using TSM CLI, see tsm jobs.

To see more information about TSM jobs in general, see Tableau Service Manager Jobs.

Canceling Jobs that are in progress

Only certain jobs can be canceled while they are already running: Cleanup, Decommission File Store, Generate Backup, Restart Server, Start Server. The cancel behavior can be different depending on the job and the state of the job at the time it was canceled. This is explained in detail below:

- **Cleanup:** If you cancel a cleanup job, it will stop any services that were started in order to do the cleanup. Depending on when it was canceled, some files may be deleted and some may not have been deleted yet resulting in partial cleanup.
- **Decommission File Store:** If you cancel this job, it returns the Tableau Server File Store topology to the state that it was prior to starting the decommissioning process.

- **Generate Backup:** If you cancel this job, any services used for backup are stopped and Tableau Server will try to delete any files that it created as part of the backup process.
- **Restart Server**:
  - Job is canceled when Tableau Server processes are stopping: The job is canceled, but the services will try to get to a stopped state.
  - Job is canceled while Tableau Server processes are restarting: The job is canceled, but the services will try to restart.
- **Start Server**: The job will be canceled, but the processes will still try to start.
- **Stop Server**: The job will be canceled, but the services will try to stop.

Here are some of the main reasons why you may want to cancel a job:

1. Since tsm jobs can only be run one at a time, you might need to cancel a current job if you need to run another job instead.
2. If the running job includes changes to the Tableau Server that you did not intend to make.

Administrative Views

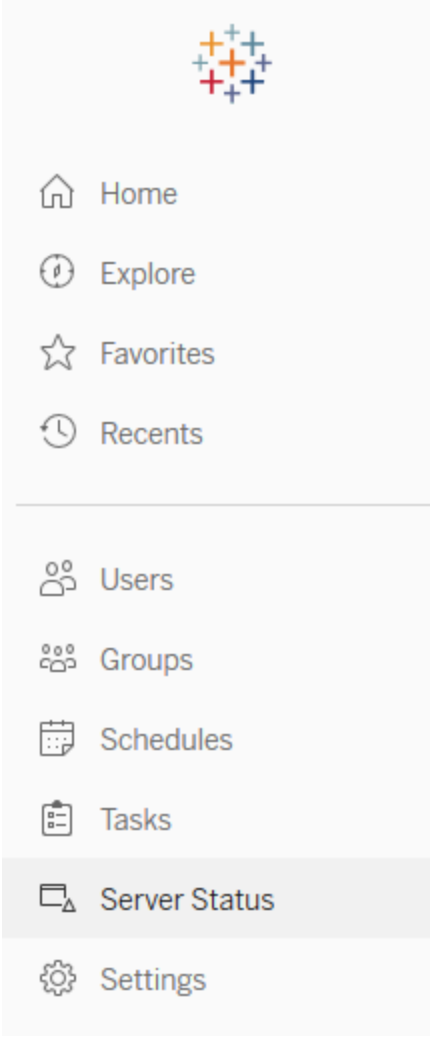
The Status page contains an embedded Tableau workbook with various administrative views. These views help you to monitor different types of server or site activity.

> **Note:** You must install PostgreSQL drivers before you can see Administrative views. For more information, see Database Drivers.

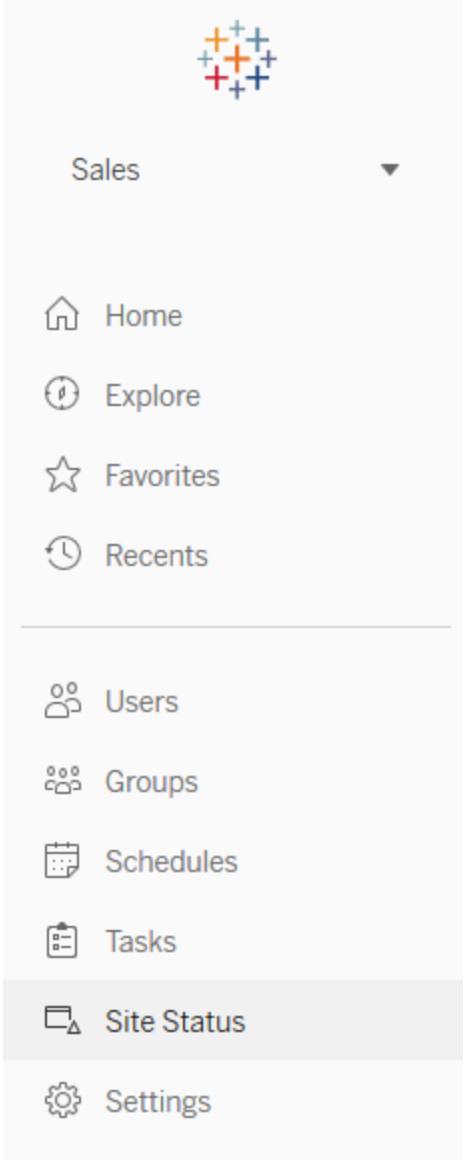
Navigating to administrative views

To see administrative views, click **Status**. Site administrators can see administrative views for their site. Administrators of multiple sites can see views for the current site.

On a multi-site server, server administrators can see views for the entire server. Click the site menu, and then click **Manage All Sites** to access the server menus.



To see views for individual sites on a multi-site server, click the site menu, select the site name, and then click **Site Status**.



Pre-built Administrative Views

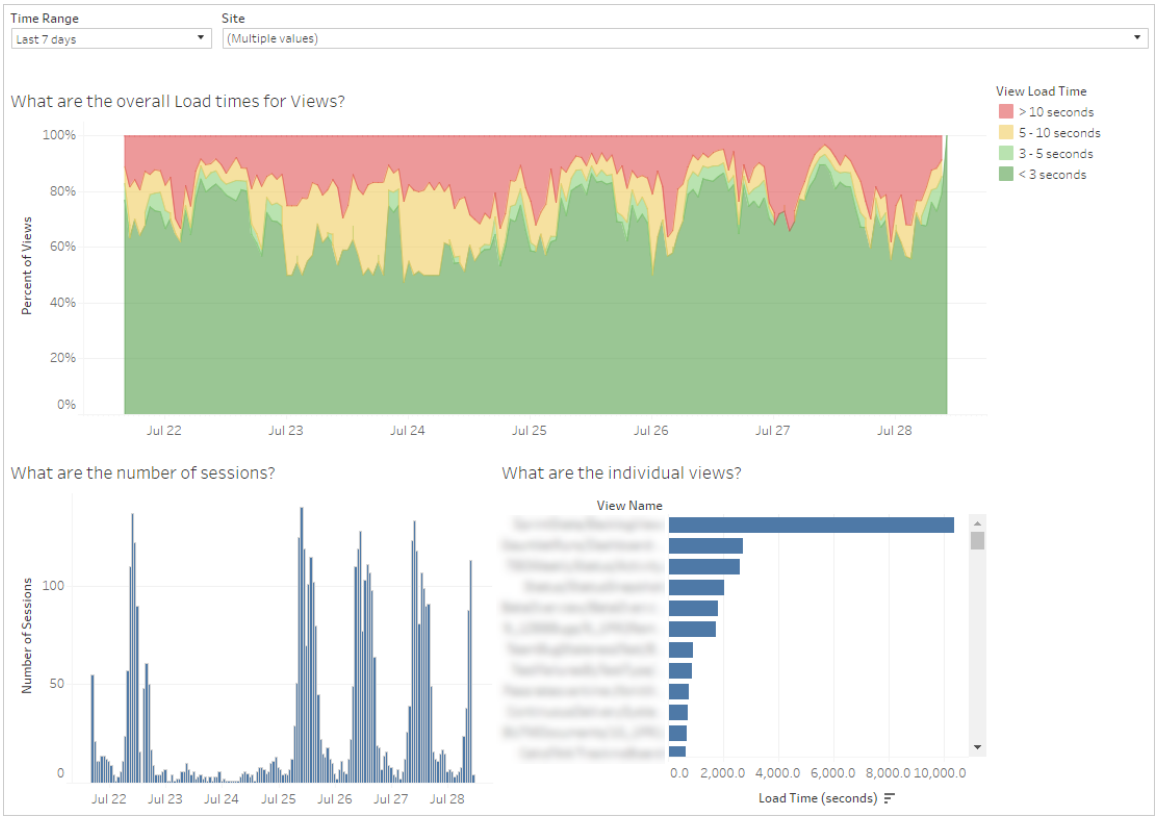
Administrative views are powerful monitoring tools that can help you optimize Tableau Server and better understand how your users are interacting with Tableau content. The administrative views listed to the right are included with Tableau Server. Click on the link for a view to learn more about how to interpret and act on the information the view provides.

To create your own administrative view, see Create Custom Administrative Views.

Performance of Views

> **Note:** This view is only available to server administrators. To access server views on multi-site deployments, click the site menu and select **Manage All Sites**. For information about how to navigate to administrative views, see Administrative Views .

The Performance of Views administrative view displays how long it takes for views to load and how many sessions are running at a time on the server.



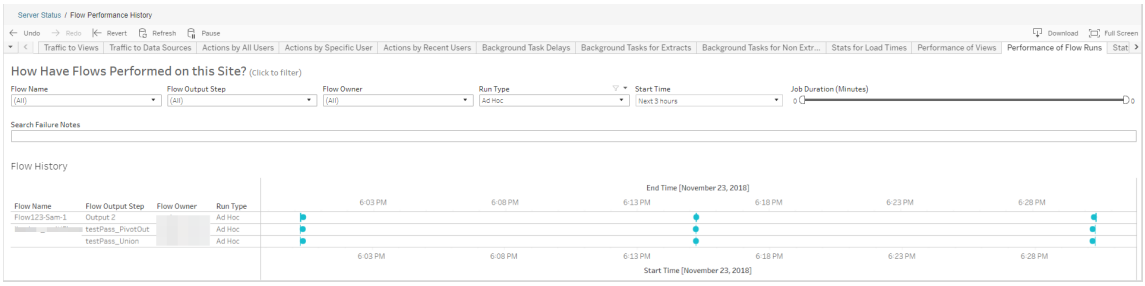
You can compare spikes in the number of sessions with spikes in slow load times to identify the times of day when high user traffic is slowing down the server. You can also look at the individual views by load time to understand which views take the longest to load.

Some views might take a long time to load regardless of when they are viewed. You can identify which workbooks need to be optimized with the **Stats for Load Times** administrative view. Some simple ways to optimize workbooks includes the following:

- Display less information in each view.
- Break up views.
- Reduce the number of filters.
- Use data extracts.

Performance of Flow Runs

Use this view to see the performance history for all the flows on a site. You can filter by Flow Name, the Output Step Name, Flow Owner, Run Type (Scheduled or Ad Hoc), and the time the flow runs were started. For information about other administrative views available for flows, see Monitor Flow Health and Performance.



Here are some questions you can answer using this view:

- **What flow tasks are currently scheduled?** – To do this, use the Start Time filter and select the time frame you want to look at. For example, to see flow tasks that are scheduled in the next 3 hours, select **Hours -> Next ->** and enter **3**.
- **What is the duration of flow tasks? -** To answer this, click on a mark in the view and you should see details including the task duration.

How many flows were run ad hoc, and how many were scheduled runs? - To answer this, use the **Run Type** filter and select **Ad hoc** or **Scheduled**.

> **Note**: This is not functional in this release and will not actually filter the data.

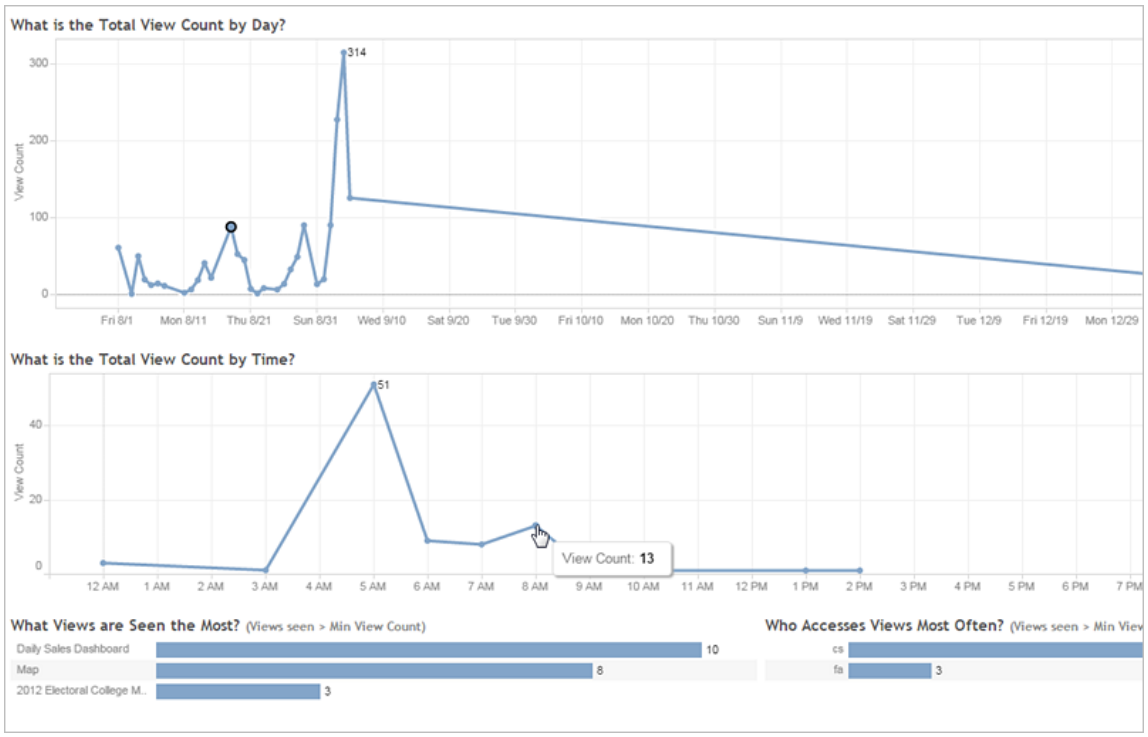
In addition to the questions described above, here are some examples of insights you might be able to gather:

- The flows that are running most frequently will have the most marks.
- To see how many flows are running at the same time currently, hover over a mark that shows "**In Progress**" or "**Pending and select** "**Keep Only**" to filter all flow runs that are currently running.
- To see how many flows are running at the same time during a specific time range, select a range for the **Start Time** filter. For example, you can choose "**Next three hours**" to see which flows will be running in the next three hours.

Traffic to Views

The Traffic to Views view gives you the ability to see how much of your user traffic goes to views.

You can filter what information is displayed and the time frame it comes from by selecting the view, the workbook, and the time range. Server administrators can specify the site.



Two time lines at the top of the view show you how views are being used over a time range you specify (the default is the last 7 days):

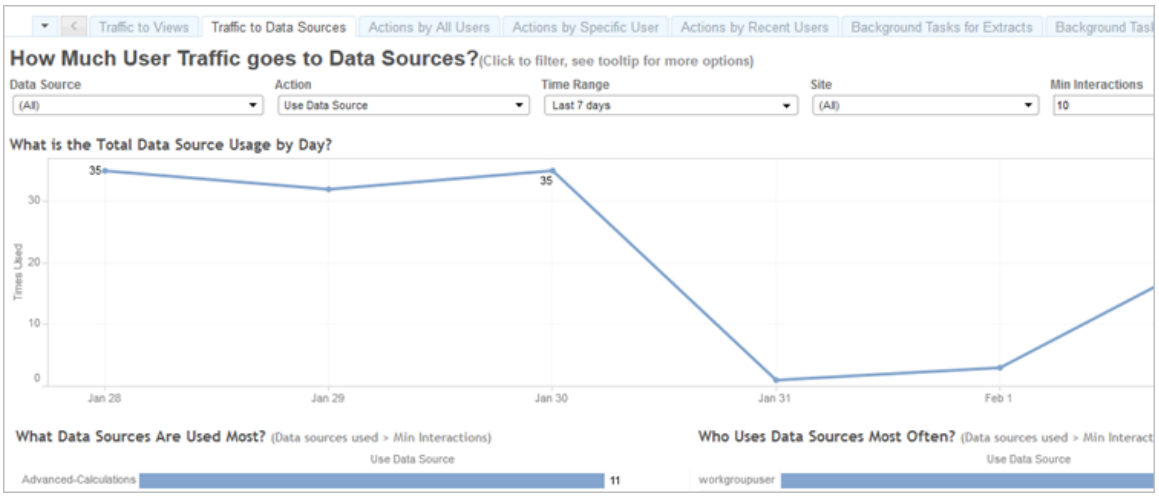
- **What is the Total View Count by Day**—This shows total view count by day, based on the filters you set. Hover your mouse pointer over a point on the line to see the count of views. Select the point to update the other sections of the view based on your selection.
- **What is the Total View Count by Time**—This shows the view count by time of day. The filters and any selection impact this graph.

Two bar graphs at the bottom of the view show results that are filtered by the **Min View Count** filter at the top of the view. These show you the views that are most often accessed, and the users who most frequently access views Only those views and users with counts greater than or equal to the minimum view count value are displayed:

- **What Views are Seen the Most**—This is a list of the most visited views. Like the other sections of the view, the information is limited by filters and any selection you make.
- **Who Accesses Views Most Often**—This shows the users who most often access the views and is limited by filters and any selection you make.

Traffic to Data Sources

The Traffic to Data Sources view gives you the ability to see usage of data sources on your Tableau Server installation. This can help you determine which data sources are most heavily used and those that are less often used. You can filter the information you see by selecting the data source, the action taken on that data source, and the time range. Server administrators can specify the site.



A time line at the top of the view shows you how data sources are being used over a time range you specify (the default is the last 7 days):

- **What is the Total Data Source Usage by Day**—This shows total data source usage by day, based on the filters you set. Hover your mouse pointer over a point on the line to see the count. Select the point to update the other sections of the view based on your selection.

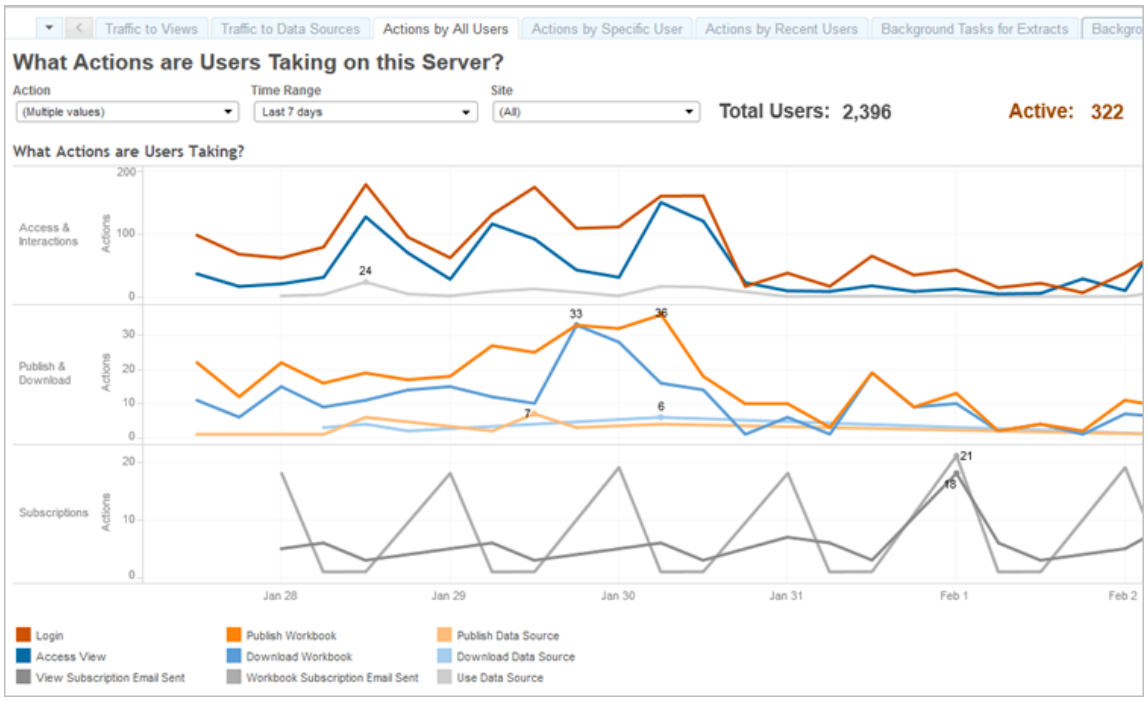
Two bar graphs at the bottom of the view show results that are filtered by the **Min Interactions** filter at the top of the view. These show you which data sources are most used, and who uses data sources most often. Only those data sources and users with interaction counts greater than or equal to the minimum interactions value are displayed:

- **What Data Sources are Used Most**—This is a list of the most used data sources. Like the other sections of the view, the information is limited by filters and any selection you make.

- **Who Uses Data Sources Most Often**—This shows the users who most often use the data sources. This is impacted by filters and any selection you make.

Actions by All Users

The Actions by All Users view gives you insight into how your Tableau Server installation is being used. You can filter the view by actions and by time range. Server administrators can filter by site. The Total Users count shows the number of users who have performed an action. This value is not affected by any filtering. The Active user count shows the number of active users who have performed one of the selected actions.



Up to three separate groups of time lines show you how users are using Tableau Server over a time range you specify (the default is the last 7 days). If no actions are selected for a particular group, that group does not display. Possible groups are:

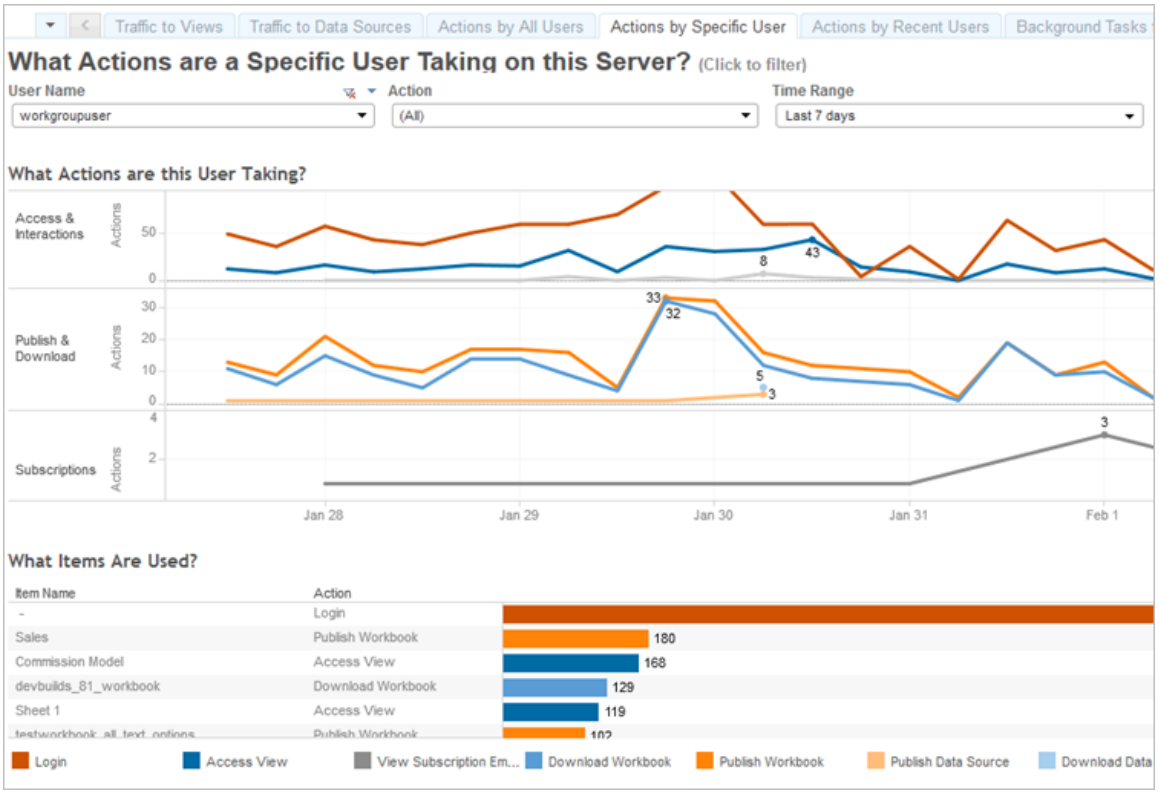
- **Access & Interactions**—This shows you sign in (log on) activity, view access and data source use.
- **Publish & Download**—This shows publishing and downloading of flows, workbooks and data sources.

- **Subscriptions**—This shows counts of subscription email sent for workbooks and views. It also shows the counts of flow runs.

Use the legend at the bottom to view a subset of the displayed actions. Click a single action to highlight the line for the action, or **Ctrl + Click** on multiple actions to highlight more than one. To clear the selection and display all the selected actions, click on any action in the legend.

Actions by Specific User

The Actions by Specific User view gives you insight into how individual users are working in your Tableau Server installation. You can filter the view by user name, actions, and time range. Server administrators on multi-site installations can filter by site.



Up to three separate groups of time lines show you how a selected user is using Tableau Server over a time range you specify (the default is the last 7 days). If no actions are selected for a particular group, or if no actions were taken, that group does not display. Possible groups are:

- **Access & Interactions**—This shows you sign in (log on) activity, view access and data source use. This means any interaction with a Data Source by the selected user, including extract refreshes scheduled by that user, or the user accessing a workbook that is associated with that data source.
- **Publish & Download**—This shows publishing and downloading of flows, workbooks and data sources.
- **Subscriptions**—This shows counts of subscription email sent for workbooks and views. It also shows the counts of flow runs.

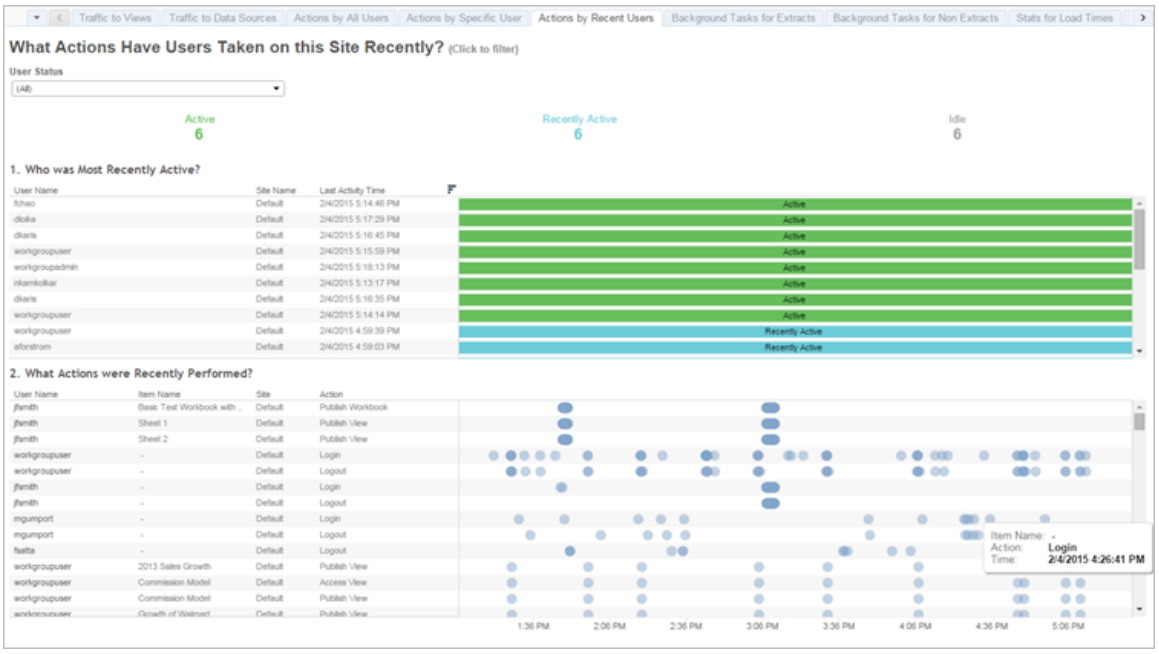
A bar graph at the bottom of the view shows which items the selected user is using.

Use the legend at the bottom to view a subset of the displayed actions. Click a single action to highlight the line for the action, or **Ctrl + Click** on multiple actions to highlight more than one. To clear the selection and display all the selected actions, click on any action in the legend.

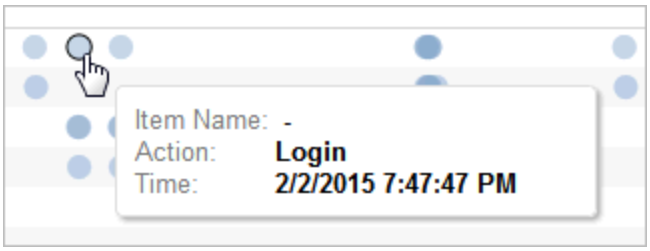
Actions by Recent Users

The Actions by Recent Users view shows you which signed-in users have been active on Tableau Server over the past 24 hours. This can be useful if you need to perform some maintenance activity and want to know how many and which users this will affect, and what they are doing on Tableau Server.

The view **Active**, **Recently Active**, and **Idle** users that are currently signed in to Tableau Server. For this view, an active user is one who took an action in the last 5 minutes, a recently active user is one who last took an action within 30 minutes, and an idle user is one who last took an action more than 30 minutes ago. The actions are displayed in the lower section of the view.

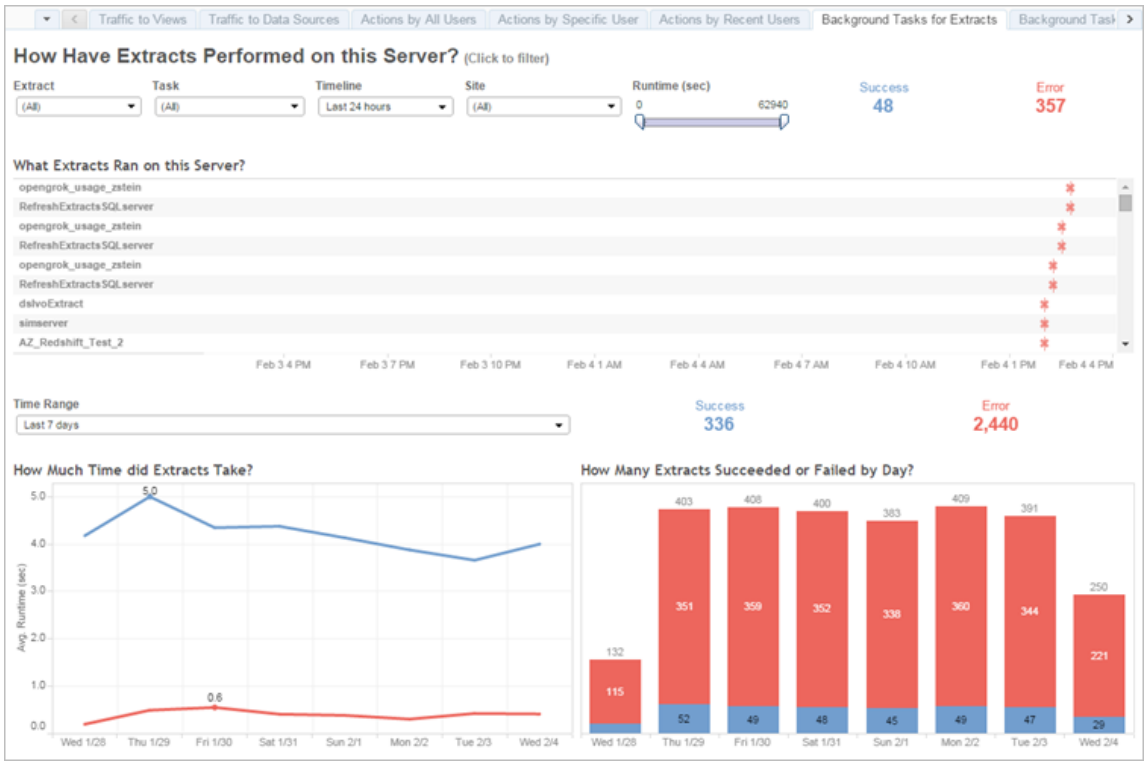


Select a user to see only the actions that user performed recently. Hover over an action to see details of the action.



Background Tasks for Extracts

The Background Tasks for Extracts view displays extract-specific tasks that run on the server.



Understand this view

To better understand this pre-built admininstrative view, make note of the following:

- The table, "What Extracts Ran on this Server," lists the extracts that ran in the time period specified in **Timeline**.
- You can click **Success** or **Error** to filter the table based on status.
- You can also click a specific task to update the "How Much Time did Extracts Take" graph for the selected task.
- The table, "How Many Extracts Succeeded or Failed," updates for the status (success or failure) of the task, but the count of extracts that succeeded or failed does not change.

Status

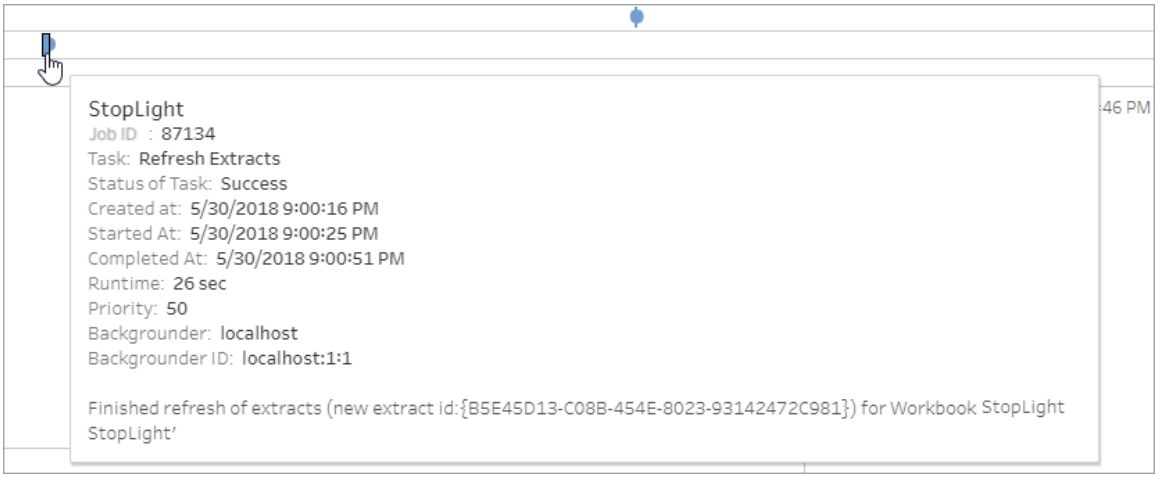
Tasks can have a status of success or error.

Icon	Description
*	**Error**—Server was unable to complete the task.

Icon	Description
	Success—Server completed the task.

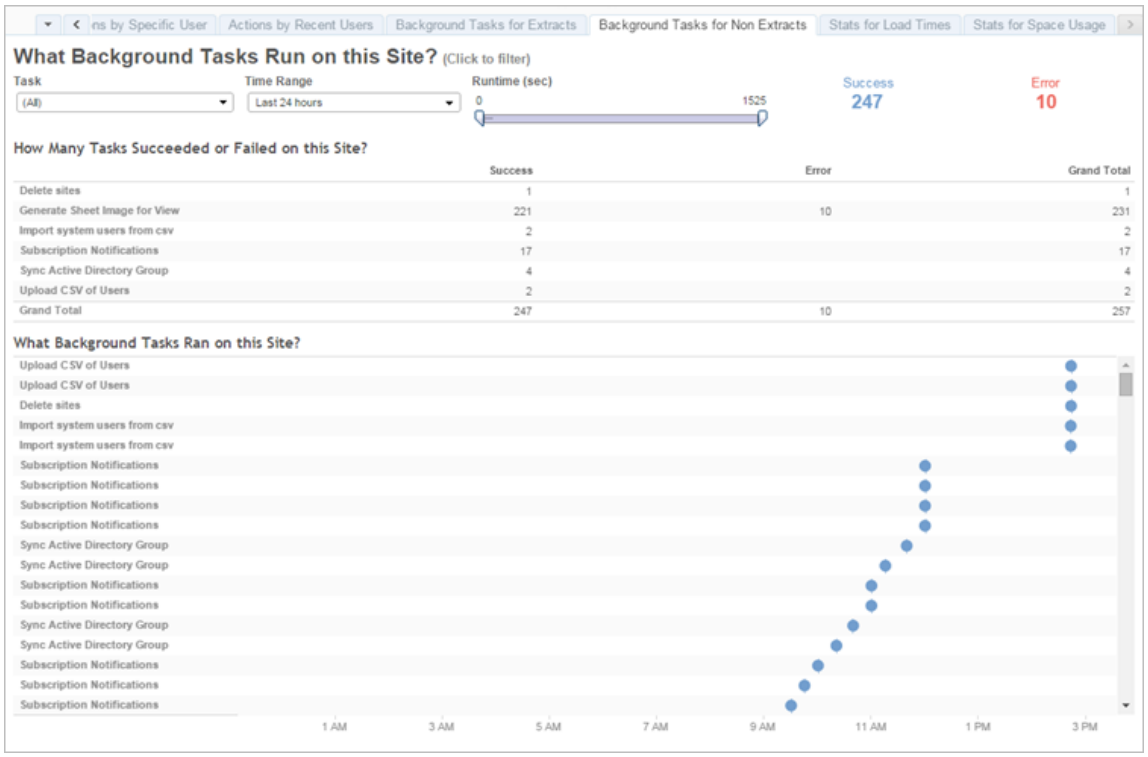
See details about a task

For details on about the task, use your mouse to hover over the success or error icon.



Background Tasks for Non Extracts

The Background Tasks for Non Extracts view displays tasks that the server runs that are not related to extract refreshes. For example, edited OAuth connections, subscription notifications, and so on.



A table lists the tasks that ran in the time range specified. Click **Success** or **Error** to filter the table based on status. Select a specific task in the **How Many Tasks Succeeded or Failed on this Site** table to update the **What Background Tasks Ran on this Site** graph for the selected task.

Tasks can have a status of success or error. For details about the task, use your mouse to hover over the success or error icon.

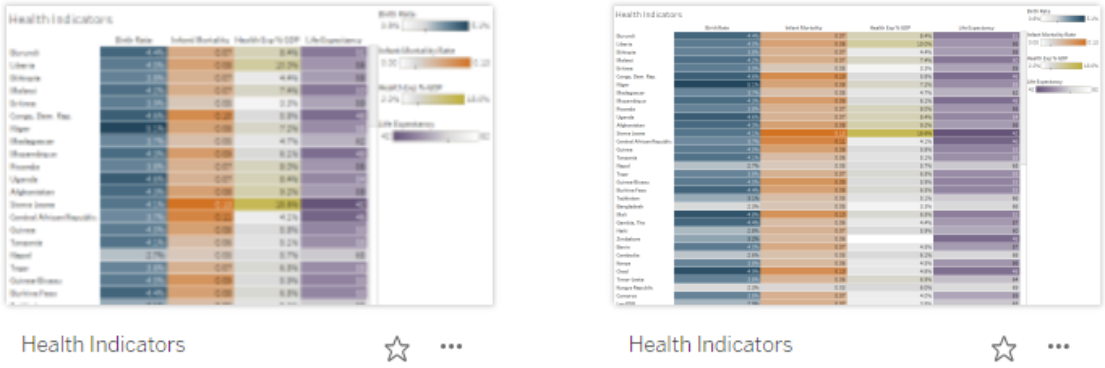
Icon	Description
	**Error**—Server was unable to complete the task.
	**Success**—Server completed the task.

Details that you can see about the task are its ID, status, priority, when it was created, started and completed. You can also see its runtime: the total run time of the background job, which includes the run time of the job plus background job overhead such as initialization and cleanup. You can also see which backgrounder the job is running on.

Upgrade Thumbnails Job

In Tableau Server version 2019.1 and later, workbooks and views use higher resolution thumbnails. When you upgrade to Tableau Server version 2019.1 or later, the Backgrounder runs the Enqueue Thumbnails Upgrade job and the Upgrade Thumbnails job, which refreshes thumbnails for any views that have not been edited or published since Tableau version 2018.2 and converts them from low resolution thumbnail images (192 x192 pixels) to high resolution thumbnail images (300 x 300 pixels), as shown in the figure below.

> **Note:** Starting in Tableau Server version 2020.2 and later, the Upgrade Thumbnails job is turned off by default. Server Administrators can use tabcmd to run the Upgrade Thumbnails job. For more information, see upgradethumbnails.



When the Upgrade Thumbnails job refreshes the thumbnail images, the modified date of the view is changed to match the thumbnail update date. The Upgrade Thumbnails job does not change the modified date of workbooks when it updates the contained view's thumbnails.

Depending on the number of workbooks you have, the Upgrade Thumbnails job can run for a few minutes to a few hours. You may notice that CPU usage on Backgrounder nodes is higher than normal during the time that the Upgrade Thumbnails job is running. You may also notice that thumbnail images may appear in low resolution for up to a week after upgrading to Tableau Server 2019.1 or later, while the Upgrade Thumbnails job processes all of the thumb-

nail images used in your workbooks. We recommend that you wait for the Upgrade Thumbnails job to complete before you back up Tableau Server.

After the first run of the Upgrade Thumbnails job, it runs on a predefined weekly schedule. The Upgrade Thumbnails job runs at lowest priority and creates one task per workbook in the background_jobs table to upgrade any low resolution thumbnails. Low resolution thumbnails published to Tableau Server by Tableau Desktop version 2018.3 and earlier are automatically cleaned up each week when the Upgrade Thumbnails job runs.

Troubleshooting

You can check the status of the Upgrade Thumbnails job using the Background Tasks for Non Extracts administrative view. The Upgrade Thumbnails job can display either the success or error status.

Upgrade Thumbnails job failed, or it completed but some thumbnails are still low-resolution.

The Upgrade Thumbnails job might show error status if your credentials are wrong. In that case, the workbook thumbnails will still appear in fuzzy, low-resolution (192 x192 pixels). Update your credentials, and the Upgrade Thumbnails job will update the workbook thumbnails the next time it runs.

Background Task Delay

> **Note:** This view is only available to server administrators. To access server views on multi-site deployments, click the site menu and select **Manage All Sites**. For information about how to navigate to administrative views, see Administrative Views .

The Background Task Delay view displays the delay for flow tasks, extract refresh tasks and for subscription tasks—that is, the amount of time between when they are scheduled to run and when they actually run. You can use the view to help you identify places you can improve server performance by distributing your task schedules and by optimizing tasks.

Here are possible reasons for the delays, and ways that you might reduce the delays:

- Many tasks are scheduled for the same time. In the example view, tasks that show long delays are clustered at the same time every day, which creates spikes in the wait time. Note that you can set the **Timeline** filter to a single day to view task delays by hour and identify the hours of the day which have many tasks scheduled at the same time. A solution to this issue can be to distribute the tasks to off-peak hours to reduce load on the server.

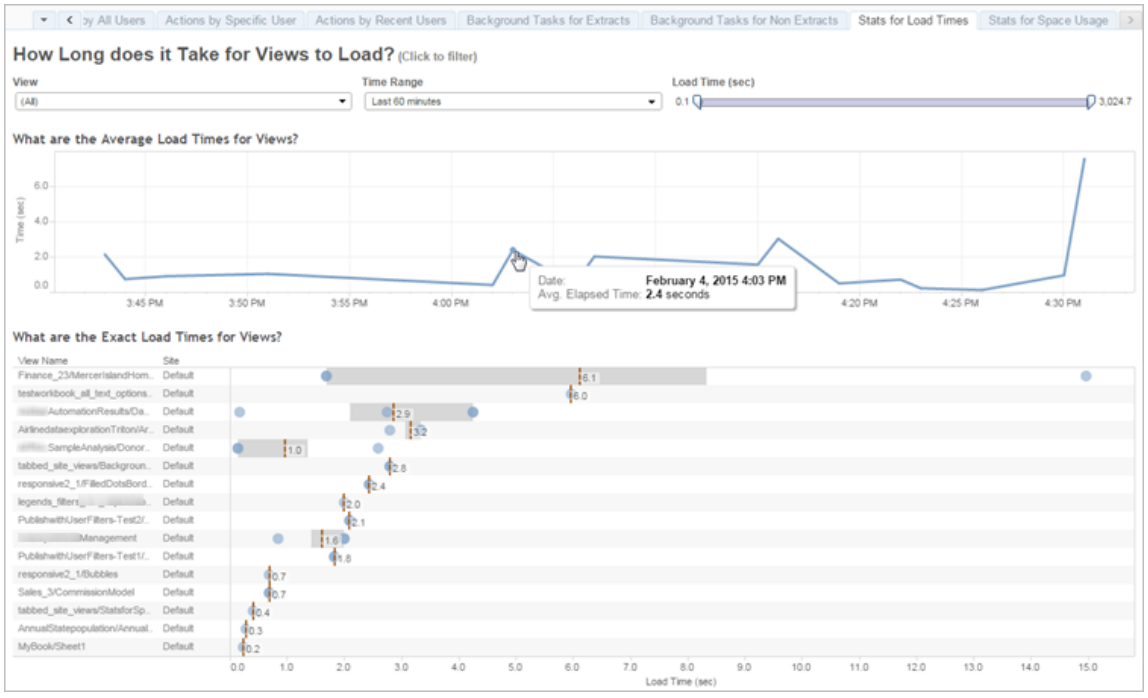
- Specific tasks take a long time to run and are preventing other tasks from running. For example, there might be an extract refresh job that is connecting to a slow data source or that is processing a large amount of data. Use the **Background Tasks for Extracts** administrative view to identify which extract refresh tasks are running slowly. You can then optimize the extract refresh task by filtering the data, aggregating the data, or creating multiple data sources for individual tables in a data source.

- Other server processes are running at the same time and are consuming server resources and slowing down performance. Monitor the CPU and memory usage of server processes to see which processes are consuming the most resources and then adjust the configuration of processes on your server.

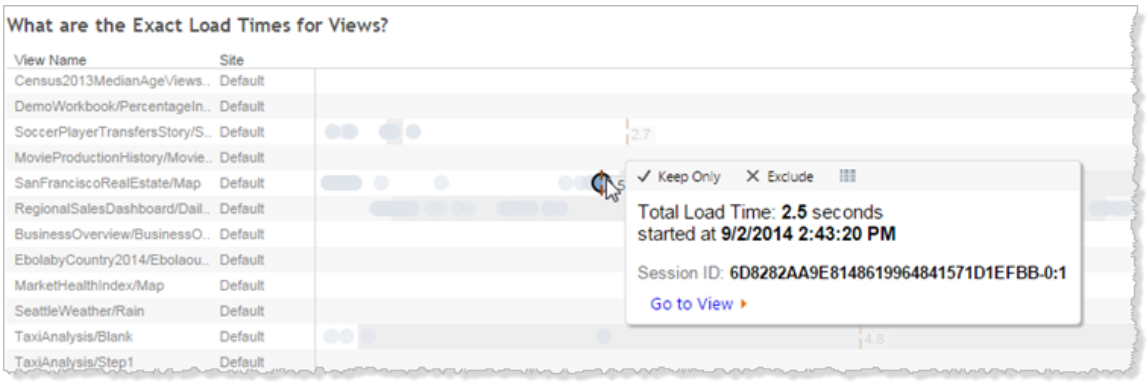
Stats for Load Times

The Stats for Load Times view shows you which views are the most expensive in terms of server performance. You can filter by view and time range. Server administrators can filter by site. You can also limit the view based on load time in seconds, using the sliding Load Time filter. Load times are for the server. Depending on your client browser and networking, actual load time may vary slightly.

The **Average Load Times** graph shows average load times for views based on the filters you set. Hover over a point to see details. Select a point on the line to update the rest of the view for the selection:



The **Exact Load Times** view shows exact time to load the listed views. A vertical line shows the average load time for each view. Select a mark to see details of a specific instance of the view loading:

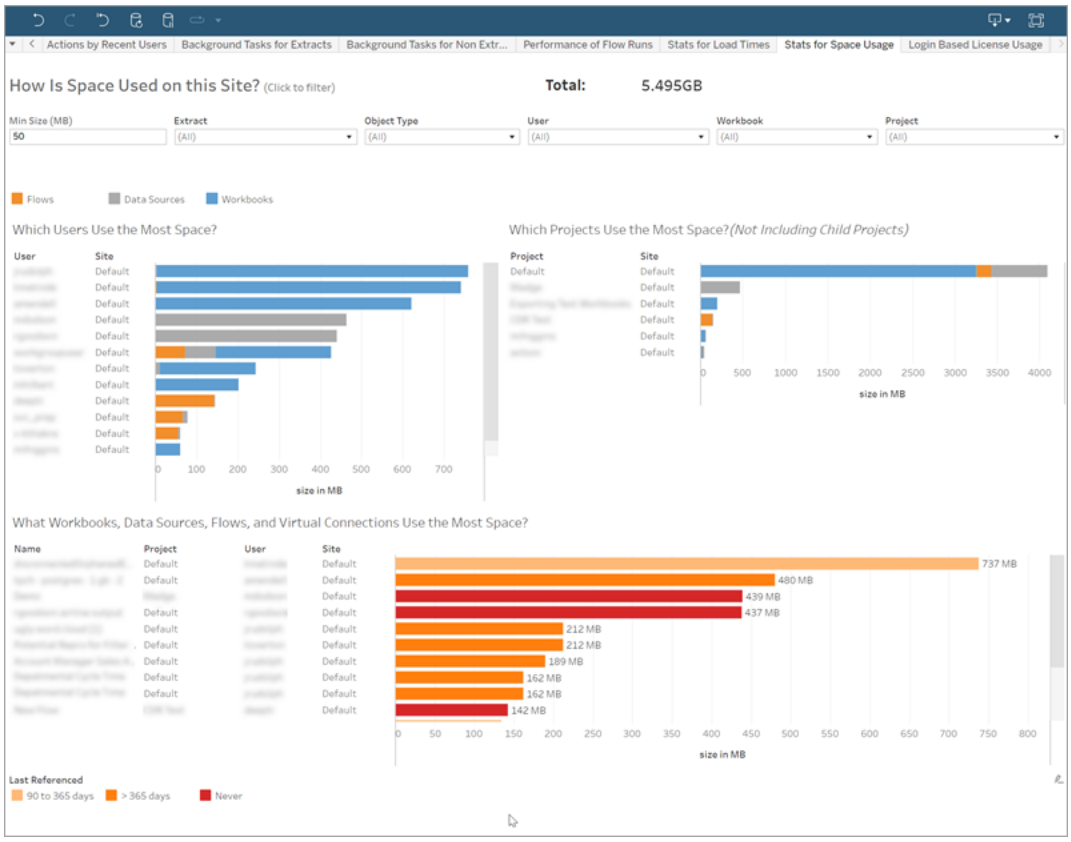


Stats for Space Usage

The Stats for Space Usage view can help you identify which Tableau content uses the most disk space on the server. Disk space usage displays by user, by project, and by the size of the Tableau content (workbook, data source, flow output, or virtual connection) and is rounded down to the nearest number.

Note that virtual connections require Data Management. See About Data Management for details.

To open this view, click **Site Status** on the left nav, and then under Dashboard, click **Stats for Space Usage**.



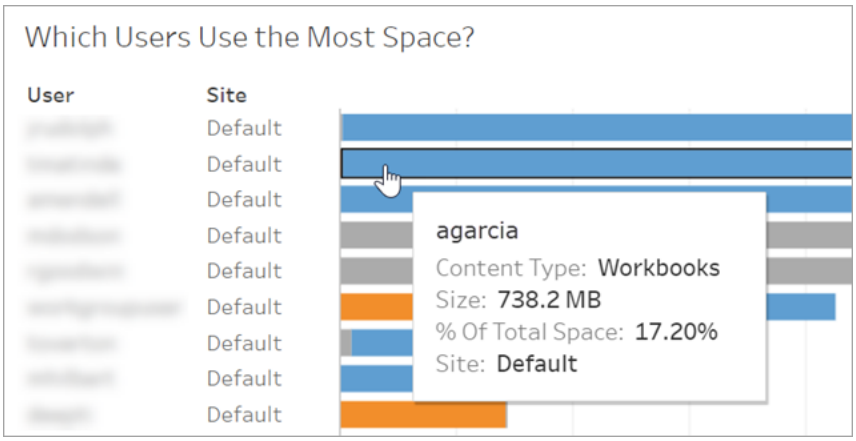
Use the **Min Size** filter to control which Tableau content displays, based on the amount of space they use.

Three bar graphs give you information about space usage on your Tableau Server:

- **Which Users Use the Most Space**—This shows the users who own data sources and workbooks that use the most space. Click a user name to filter the next two graphs for that user. Click the data source bar or the workbook bar for a user to filter the next two graphs for that type of object for that user. Click the selected user or bar to clear the selection.
- **Which Projects Use the Most Space**—This shows the projects with the data sources and workbooks that use the most space. If a user or object type is selected in the Which Users Use the Most Space graph, this displays information specific to the selection.

- **Which Workbooks, Data Sources, Flows, and Virtual Connections Use the Most Space**—This shows which Tableau content uses the most space. The bars are color-coded based on the length of time since the last refresh.

Move your cursor over any bar to display usage details:



Click on a bar to select it and update the other areas of the view based on that selection.

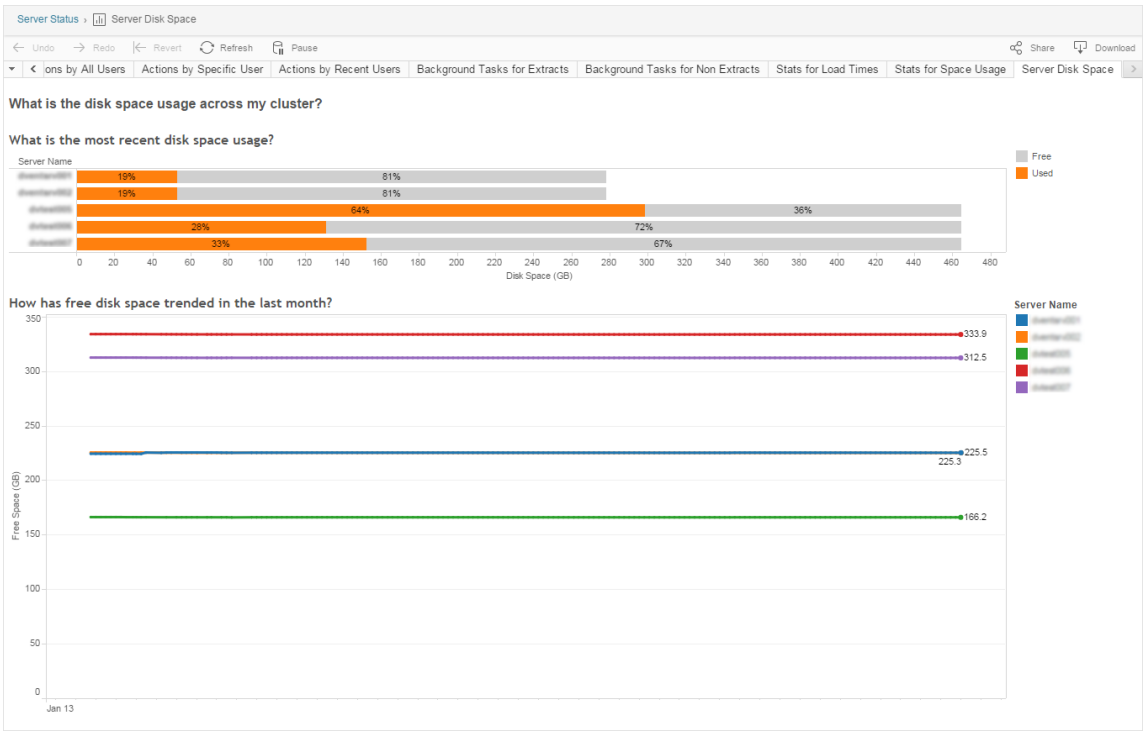
Server Disk Space

Note: This view is only available to server administrators. To access server views on multi-site deployments, click the site menu and select **Manage All Sites**. For information about how to navigate to administrative views, see Administrative Views .

Use the Server Disk Space view to see how much disk space is in use on the computer or computers that run Tableau Server, where disk space refers only to the partition where Tableau Server is installed. You can also use this view to identify sudden changes in disk space usage.

This view reports disk space usage as a decimal GB value. If the operating system of your Tableau Server computers report the value using binary GB, the amounts can differ.

For a distributed installation, the view displays information about each computer in the cluster.



The Server Disk Space view includes two graphs:

- **What is the most recent disk space usage?**—This graph shows disk space usage for the last 30 days both in gigabytes and as a percentage. Disk space refers only to the partition where Tableau Server is installed.
- **How has free disk space trended in the last month?**—This graph shows changes to disk space usage over the last month. Rest your pointer on a line to view the exact amount of free disk space for a point in time.

When Tableau Server is low on disk space, you can remove files to free space.

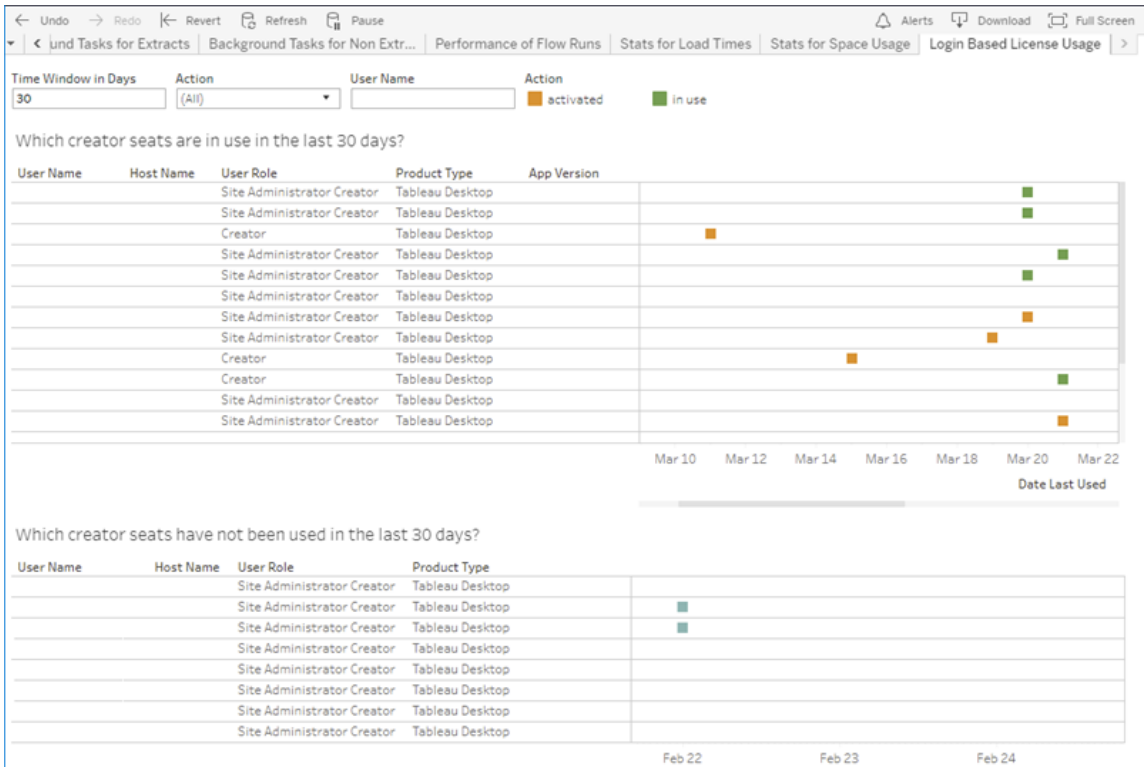
> **Tip:** You can have Tableau Server notification you when free disk space falls below a threshold that you specify. For more information, see Configure Server Event Notification.

Login-based License Usage

Note: This view is only available to site administrators and server administrators when login-based license management is enabled on Tableau Server. For information about how to navigate to administrative views, see Administrative Views .

The Login-based License Usage view lets server administrators view login-based license activation usage for Tableau Cloud or Tableau Server. The Login-based License Usage view can help you manage licenses efficiently and determine if you need more or fewer licenses. This view can help you answer the following questions:

- Who is using a Tableau Desktop or Tableau Prep Builder license in my enterprise?
- Has a Creator role been shared or transferred?
- Has any activation activity occurred on a computer where it should not be?
- On which host is the actiation being used?
- Which role is assigned to the user?
- On which Tableau product is the license in use?
- On which Tableau version is the license in use?
- Did the Creator role activate through Tableau Desktop or Tableau Prep Builder?
- Has the Creator seat been activated?
- How many Creator seats are in use?
- How many Creator seats are not in use?
- When was a Creator seat was last used?



In addition to using the login-based license usage administrative view, you can also access login-based license usage data (`identity_ based_activation_reporting`, `identity_based_activation_user_role_change`, and `identity_based_activation_admin_view`) in the "workgroup" PostgreSQL database of the Tableau Server repository. Before you can access this data, you must enable access to the Tableau Server repository.

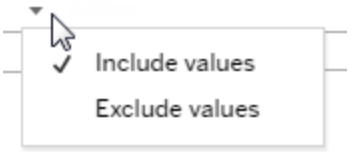
Filters

On the report screen, you can change the time window to show when seats were last used, filter on actions, filter on user name, and sort by columns.

- **Time Window in Days**. Enter the number of days for which to view login-based license management activated client usage data. You can view data for the past 30 days up to a maximum of 183 days.
- **(All)**. Apply all filters to the view.

- **Activated**. Show Creator users that have activated using login-based license management.
- **in use**. Show Creator users who activated using login-based license management whose seats are in use.
- **last used**. Show when the login-based license management client was last used.
- **unassigned**. Show which login-based license management activated Creator seats are currently unassigned.
- **user name**. Show login-based license management activations in use by the specified user.

When you hover over the filter card, a drop-down icon appears. Click the icon to specify whether the view should include data that matches the filter (the default) or exclude data that matches the filter:



Which creator seats are in use in the last <nn> days?

This area of the dashboard shows a list of three types licenses (activated, in use, and unassigned). Hovering over an activated, in use, or unassigned mark gives you information including the registered user of the copy of Tableau. Click a column head to sort the list.

Which creator seats have not been used in the last <nn> days

This area of the dashboard shows a list of licenses that have not been used during the specified time period. A timeline shows the last use date. Hovering over a last use mark gives you information including the registered user of the copy of Tableau.

Desktop License Usage

Note: This view is only available to server administrators. To access server views on multi-site deployments, click the site menu and select **Manage All Sites**. For information about how to navigate to administrative views, see Administrative Views .

The Desktop License Usage view lets server administrators see usage data for Tableau Desktop licenses in your organization. This can help you manage licenses efficiently and determine if you need more or fewer licenses. This view can help you answer the following questions:

- Who is using a Tableau Desktop license in my enterprise?
- Have any licenses been shared or transferred?
- Is any license being used on a computer where it should not be?
- Does a specific user use their license?
- What types of licenses are being used in my enterprise?
- Do I need to convert any trial licenses?

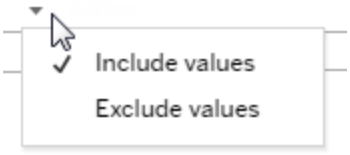
Note: To get data about licenses, each copy of Tableau Desktop version 10.0 or later needs to be configured to send data to Tableau Server. This configuration can be done at installation time, using scripting or third-party software to install and configure Tableau, or after installation, by modifying the registry or property list file. For more information, see Configure Desktop License Reporting.

In order to view license data, Desktop License Reporting must be enabled on Tableau Server. See Enable and configure Desktop license reporting.

Filters

- **Product Keys**. Type a string to filter the dashboard to only those licenses that include the string anywhere in the product key. For example, to see only licenses that begin with TDTD, type TDTD and press Return to filter the view. Click the **X** after the string to reset the filter.
- **Action**. Use this filter to control what the dashboard displays, based on the action taken. Actions are **Activate**, **Use**, and **Return** (deactivate). If the **Use** action is not selected, nothing is displayed in the top bar graph.
- **Department**. Use this filter to control what departments the dashboard displays licenses for. The filter is populated based on the **Department** values specified when Tableau Desktop is registered.
- **Select time duration in days**. Use this slider to specify the time length in days that the dashboard displays information for. The default value is 183 days.

When you hover over the filter card in the first three filters, a drop-down icon appears. Click the icon to specify whether the view should include data that matches the filter (the default) or exclude data that matches the filter:



Who has used Tableau in the last <nn> days?

This area of the dashboard shows a bar graph of three types of Tableau Desktop licenses (Perpetual, Trial, and Term) and the number of users who have used each license type during the specified time period. Hover over a license type segment to see an explanation of the license type. Click a segment to filter the rest of the dashboard for only that license type. This action filters both the tables that show licenses that have been used and those that have not been. For example, to see a list of term licenses that have been used during the time period, click the Term bar. The "used" and "not been used" lists are filtered to just show term licenses.

A table of detailed information shows under the bar graph. For each row in the table, action icons display on the right, above a timeline that shows you when the action last took place.

To see a list of the underlying data in a format that allows you to select and copy values like email or product key, click a row in the list of licenses and click the View Data icon:

The data displays in summary form. Click **Full data** to see all the data. From this view you can select and copy individual values, or download the data as a text file.

What licenses have not been used in the last <nn> days

This area of the dashboard shows a list of licenses that have not been used during the specified time period. A timeline shows the last use date. Hovering over a last use mark gives you information including the registered user of the copy of Tableau.

Desktop License Expiration

> **Note:** This view is only available to server administrators. To access server views on multi-site deployments, click the site menu and select **Manage All Sites**. For information about how to navigate to administrative views, see Administrative Views .

The Desktop License Expiration view gives server administrators information about which Tableau Desktop licenses in your organization have expired or need maintenance renewal. This can help you manage licenses efficiently. This view can help you answer the following questions:

- What trial or term licenses have expired?
- What perpetual licenses have expired maintenance?
- What perpetual licenses have maintenance renewals coming up?

To renew a license or get additional renewal information, see How to renew your Tableau licenses.

Note: In order to get data about licenses, each copy of Tableau Desktop version 10.0 or later needs to be configured to send data to Tableau Server. This configuration can be done at installation time, using scripting or third-party software to install and configure Tableau. For more information, see Configure Desktop License Reporting.

In order to view license data, Desktop License Reporting must be enabled on Tableau Server. See Enable and configure Desktop license reporting.

Filters:

- **Product Keys**—Type a string to filter the dashboard to only those licenses that include the string. For example, to only see licenses that begin with TDTD, type TDTD and press return to filter the view. Click the "x" after the string to reset the filter.
- **Department**—Use this filter to control what department(s) the dashboard displays licenses for. The filter is populated based on the Department values used when registering copies of Tableau Desktop.
- **Time Duration**—Use this filter to control the length of time for which the dashboard displays information.

The view includes the following tables, which are affected by the filters you set at the top of the view:

- **What keys have expired maintenance**—This table shows the product keys for which maintenance has expired, with a vertical line indicating the point at which the six month window for renewing maintenance closes. If maintenance for a key is expired for more than six months you need to purchase a new key in order to qualify for support or

upgrades.

- **What trial and term licenses have expired**—This shows the trial or term product keys that have expired.
- **What is the maintenance schedule for my keys**—This shows the keys and their maintenance status.

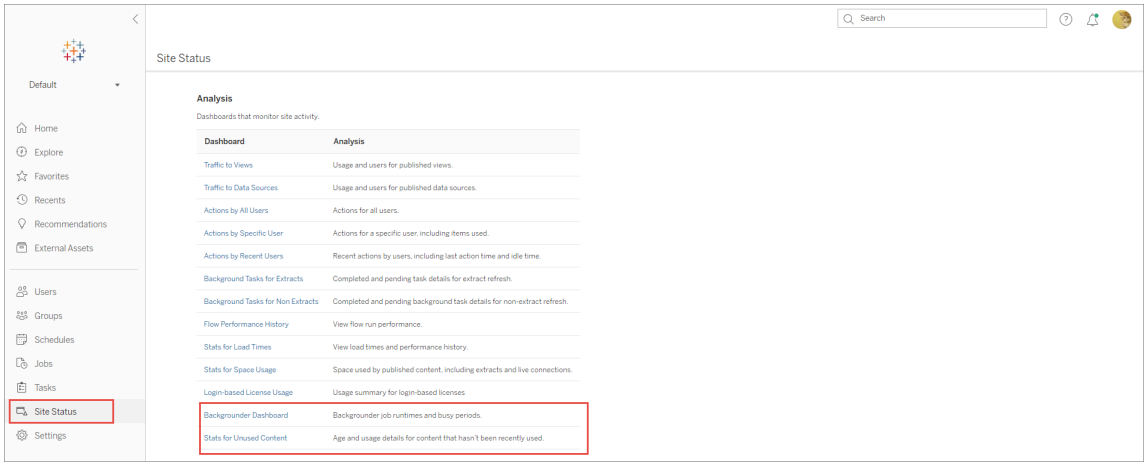
Backgrounder Dashboard

The **Backgrounder Dashboard** view is an overview of the background jobs. Using this view, you can find more information about:

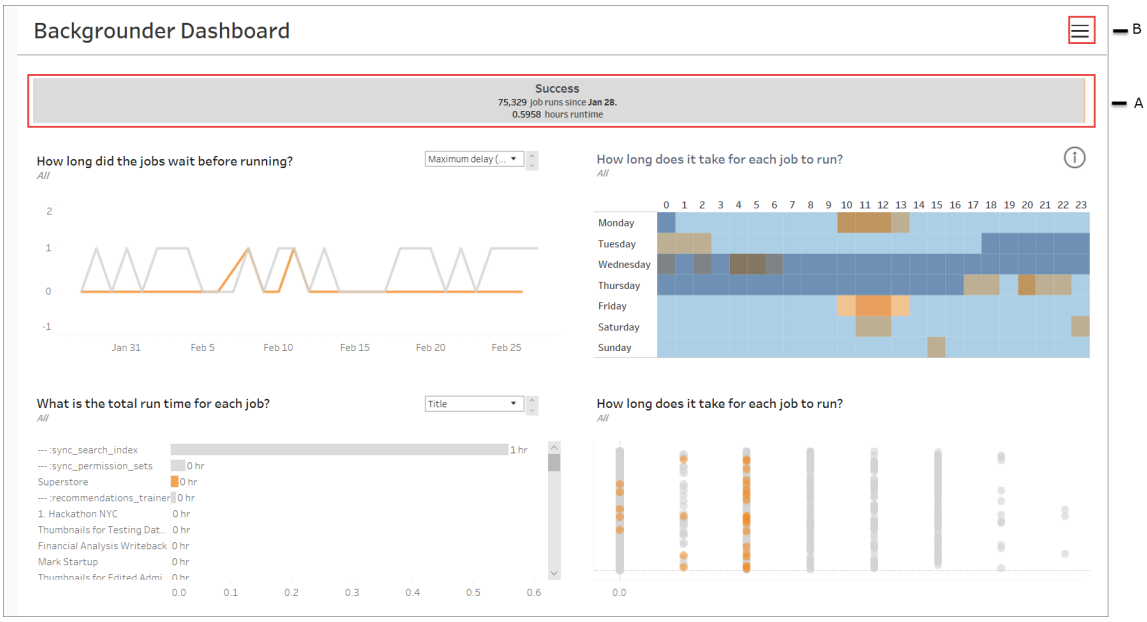
- The time it takes for jobs to run.
- When backgrounder is busy or overloaded.
- Jobs that completed successfully, failed, or canceled.

> Note: This view does not include flow run jobs.

In previous versions of Tableau Server, admin views were all displayed in the same workbook, in separate tabs. However, the two new admin views are displayed as separate workbooks and not part of the existing admin view workbook. You can navigate to the new admin views from the Server or Site Status page:

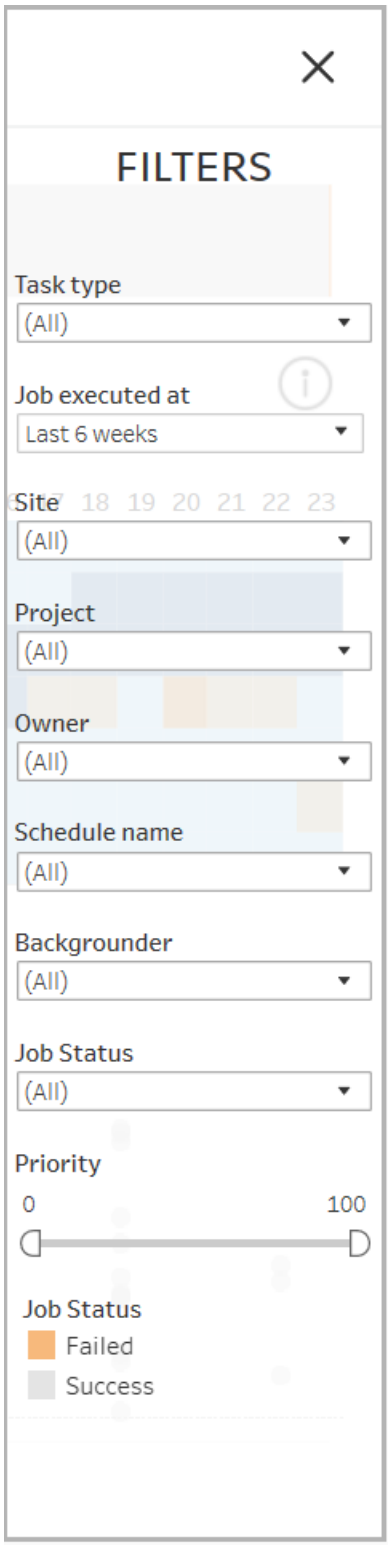


Summary and Filters



A - At the top, is a summary chart that tells you the number of jobs that have succeeded, failed or canceled. You can click on the sections of the bar chart to filter the information by job status. This filter is applied to the entire view and the information displayed includes only jobs with the job status that you selected.

B - You can find additional filter options by clicking on the filter icon. A Filter pane is displayed that allows you to filter by **Task type**, **Job executed at**, **Site**, **Project**, **Content owner**, **Schedule**, and **Backgrounder ID** . The Backgrounder ID is unique to a Backgrounder process. You can use this to see the information about the work done by each Backgrounder process. When you select one or more of these filters, they are applied to the entire view.


FILTERS
Task type
(All)
Job executed at
Last 6 weeks
Site
(All)
Project
(All)
Owner
(All)
Schedule name
(All)
Backgrounder
(All)
Job Status
(All)
Priority
0
100
Job Status
Failed
Success

Details

The Backgrounder Dashboard has four sections each showing different information about jobs. Each of these sections have more filters in the drop down menu that you can apply to that specific section.

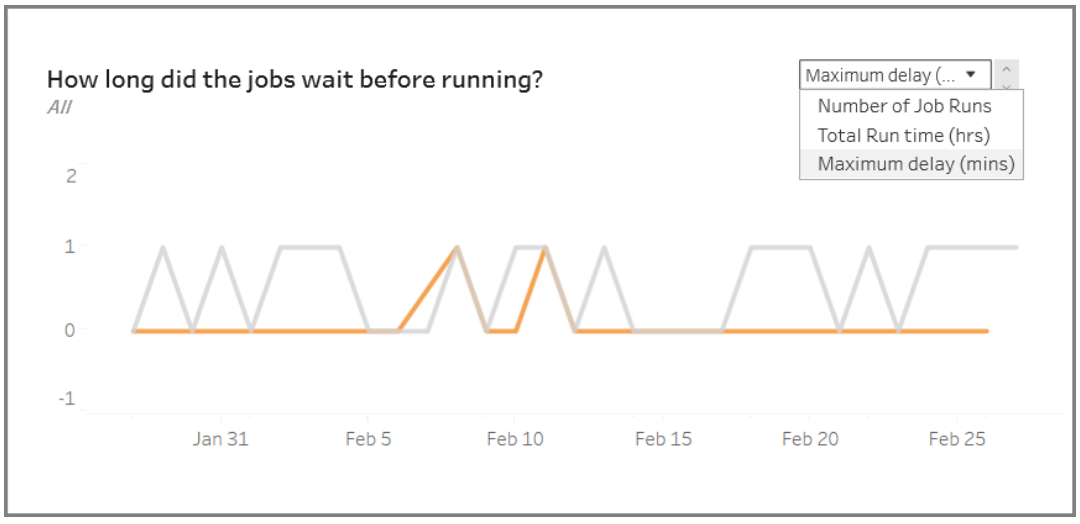
1. **The section at the top left** gives you information about the jobs with the selected job status or any filters you selected using the Filter pane. If no selection is made, all jobs are included.

   The information displayed also varies based on the selection you make using the drop down:

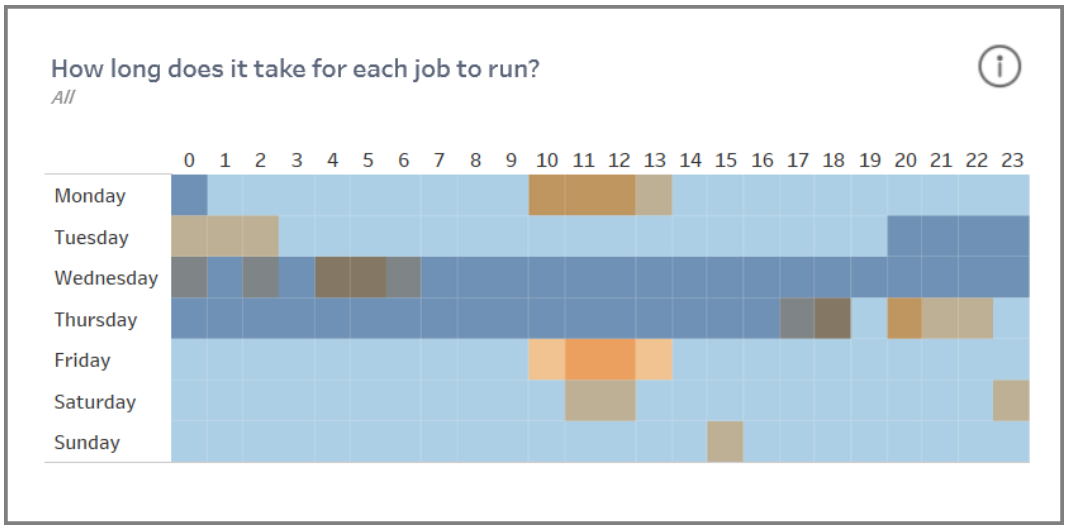
   1. **Total run time** : This shows you the total run time for all jobs with the selected job status, and any other filters you selected.

      For example, if you selected, Job Status: Failed, and Task type : Subscriptions, the graph shows all subscription jobs that failed.

   2. **Number of Jobs**: This shows you the number of jobs that ran for the selected job status and any other selected filters.
   3. **Maximum delay**: This shows you the how long the jobs were queued before running.

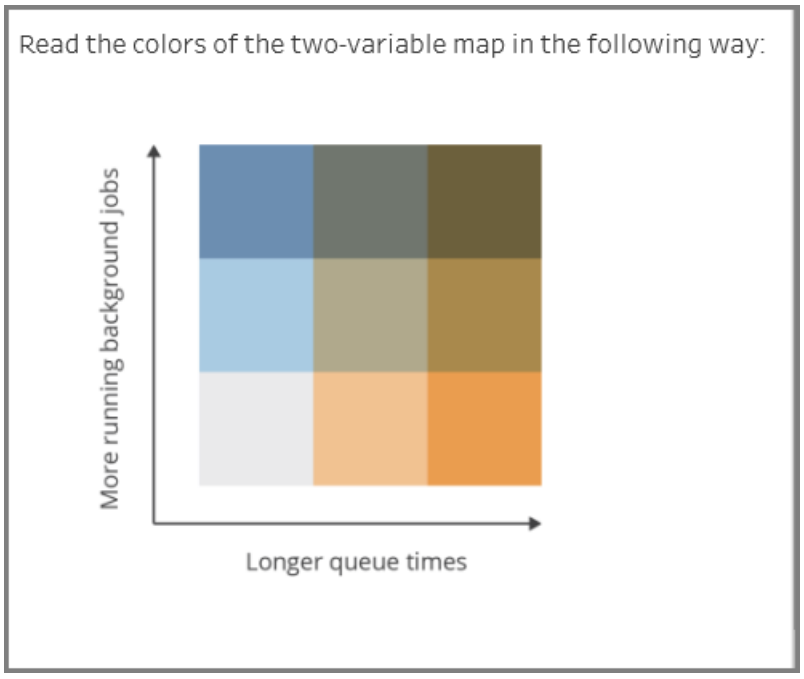


2. **The section at the top right** gives you information on how busy or overloaded the Backgrounder is on a given day and time. The information displayed however depends on the filter selections you made for the job status and other options in the Filter pane.

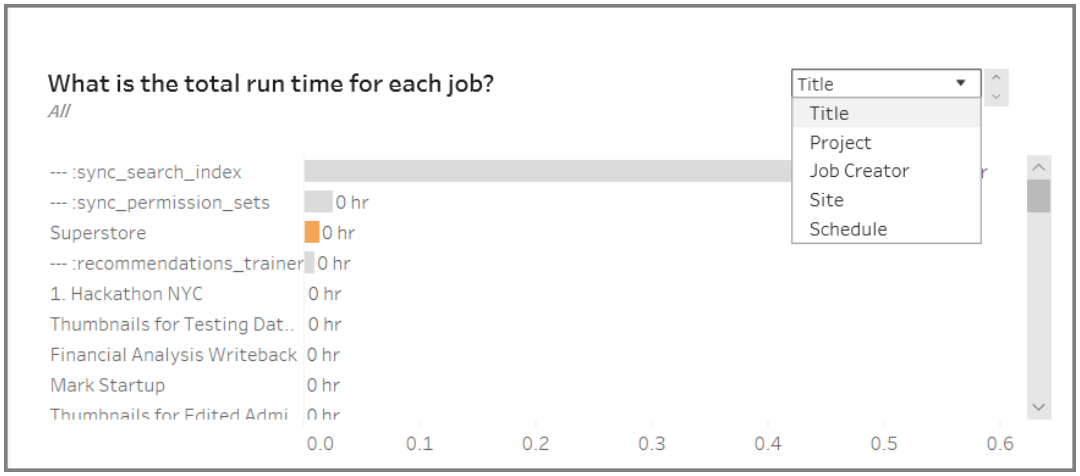


If you click on the information icon at the top right part of this section it, a color legend is displayed.

- The variations in the blue color correlates to the number of jobs running in that time period. The darker the blue, the more jobs that are running.
- The variations in the orange color correlates to the queue times . The darker the orange, the longer the queue time.
- Using both of these metrics, you can identify when the backgrounder is busy (dark blue) or is overloaded (dark orange).



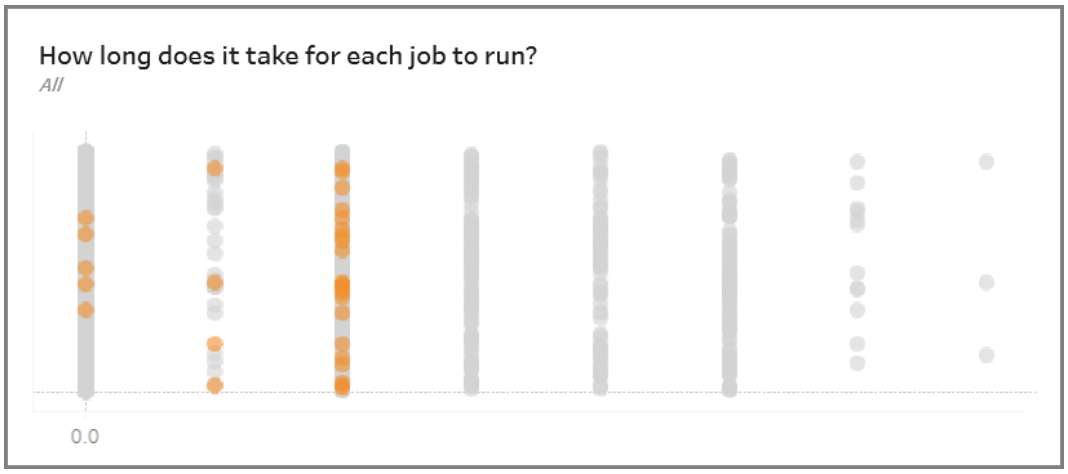
3. **The section at the bottom left** gives you run time information about the jobs with the selected status or any other filters you selected using the Filter pane. If no selection is made, all jobs are included. Select marks in the top sections to populate this section with details for the selected content.



The drop down selections gives you more options and the information displayed changes accordingly:

- When **Title** is selected, the run time information for each individual job is displayed.
- When **Project** is selected, the total run time for each project is displayed.
- When **Job Creator** is selected, the total run time for jobs created by a specific user is displayed.
- When **Site** is selected, the total run time for jobs on that specific site is displayed.
- When **Schedule** is selected, the total run time for jobs using that specific schedule is displayed.

4. **The section at the bottom right** tells you how long it took for each job to run.



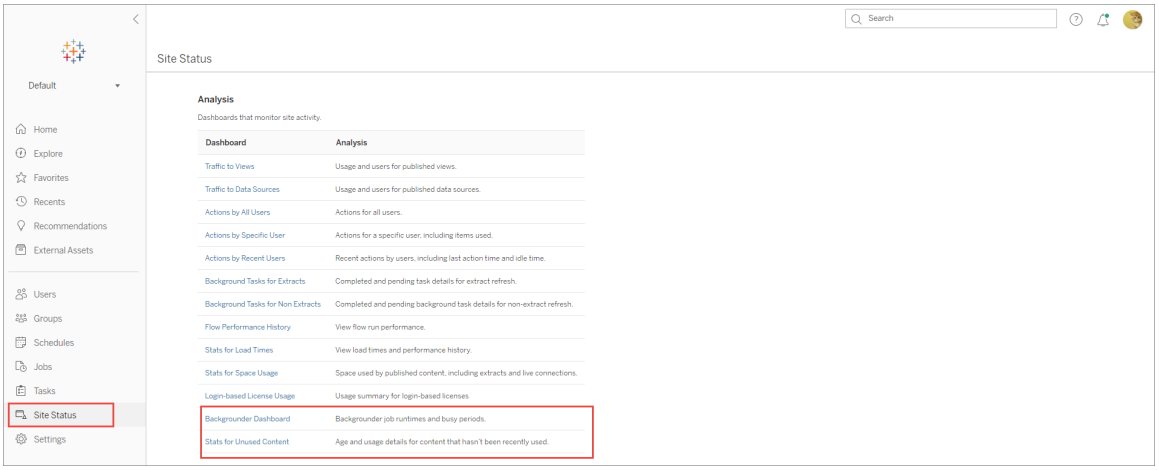
Stale Content

The Stale Content view can be used to identify content that hasn't been used or accessed in the specified time period (displayed as Stale Access Threshold). You can set that time period in days. The minimum value for the time period is 1 day and the maximum is 120 days.

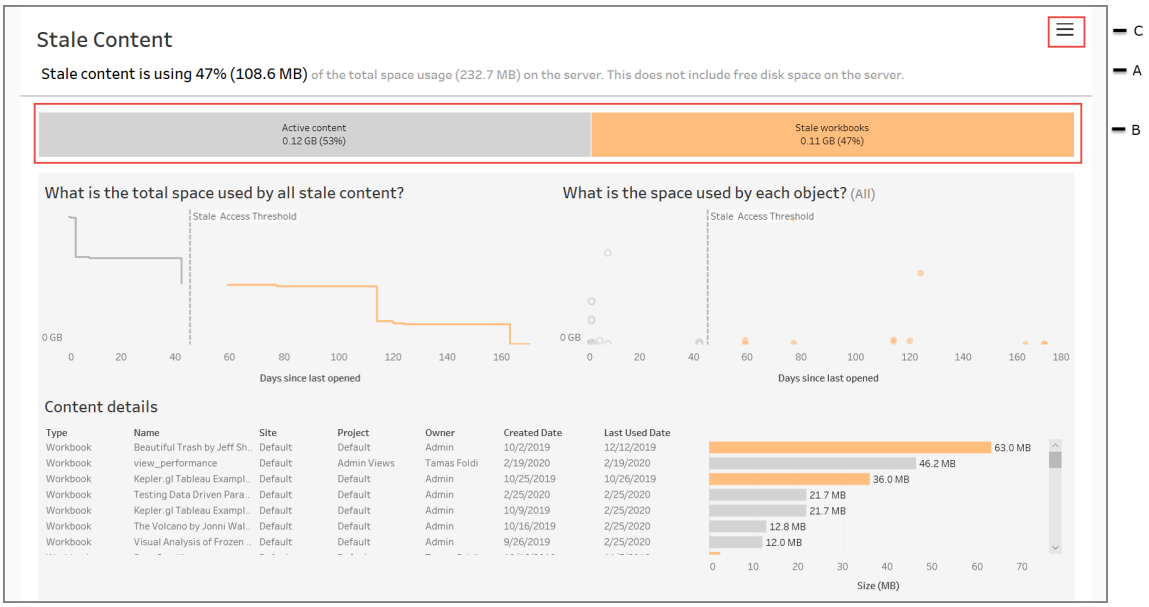
This view also provides the information about the disk space used by stale and active content.

In previous versions of Tableau Server, admin views were all displayed in the same workbook, in separate tabs. However, the two new admin views are displayed as separate workbooks and not part of the existing admin view workbook. You can navigate to the new admin

views from the Server or Site Status page:



Summary and Filters



A - At the top of the view, you will see a statement that summarizes the amount of space that is used by stale content compared to the total space used. The total space used is defined as the sum total of disk space used by active and stale content.

B - This summary is followed by a chart that gives you a further breakdown of the types of stale content and content that is considered active - meaning content that has been accessed in the

time period below the stale threshold. You can click on the bar chart and apply it to filter the data displayed in the view.

C - You can see and apply additional filter options by clicking the filter icon. This filter pane includes:

• Stale Threshold

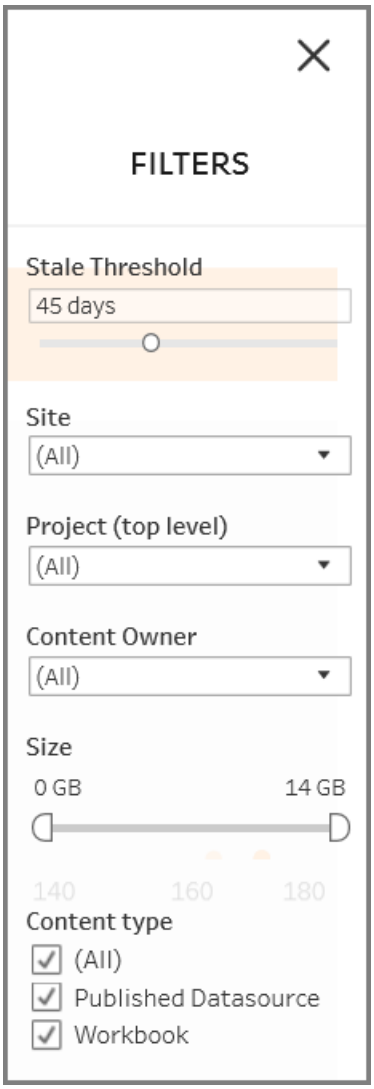
• Site

• Project

• Content Owner

• Size

• Content type



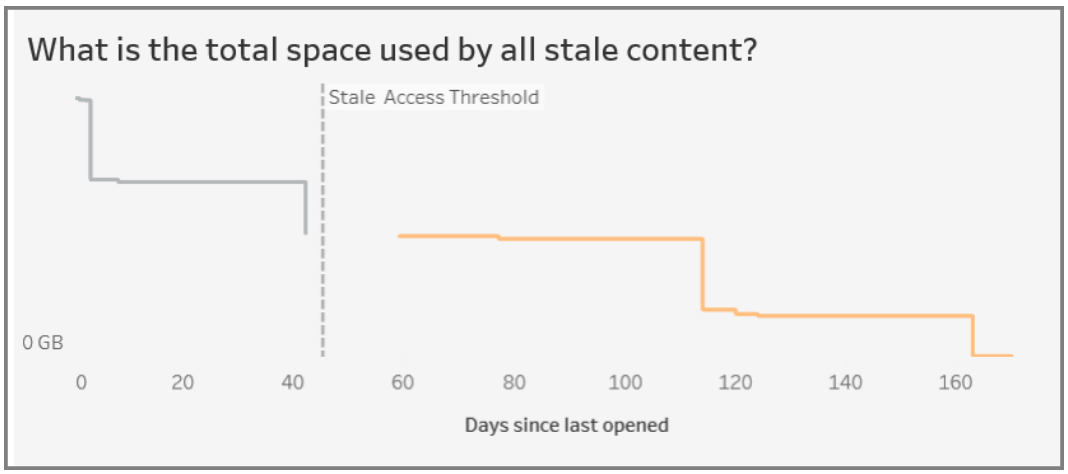
These filters are applied to the entire view.

Details

The Stale Content view has three sections as described below that provides details:

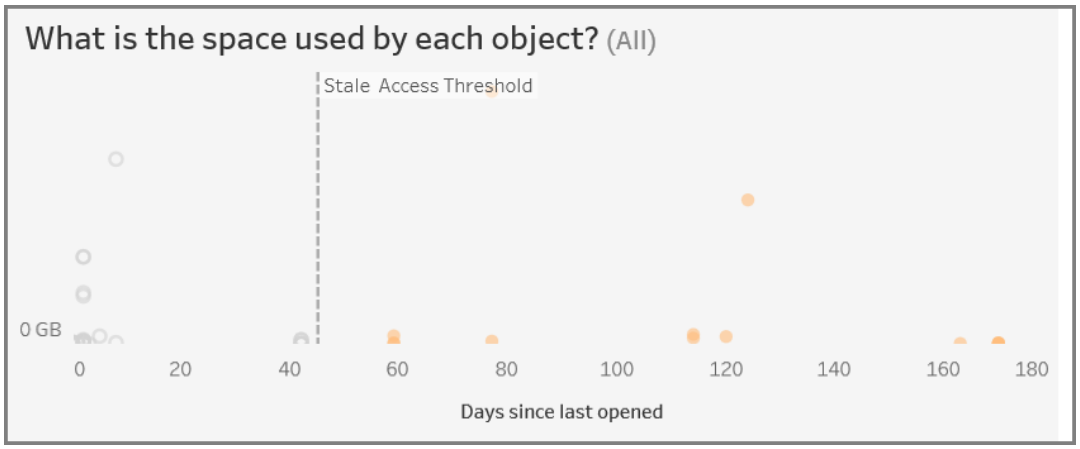
1. **The top left section** shows you the total space used for the selected content. The x-axis shows the number of days that have passed since the content was last opened, and the y-axis shows you the size. The graph also shows the stale threshold.

Set your desired staleness threshold, then use this view to identify content that is the most stale. Click the **Stale Workbooks** or **Stale Datasources** in the bar at the top to filter to the content of interest. Select the marks to the right of the **Stale Access Threshold** to see more details about content. The details are displayed in the bottom left section.



2. **The top right section** shows you the amount of space that is used by each selected content. For example, if you select Stale workbooks, the space used by each stale workbook is displayed. You can use this section to find out which content is the most stale, or is taking the most space.

   This section helps you identify content that hasn't been used in a long time. Click the **Stale Workbooks** or **Stale Data Sources** on the bar at the top. Select the oldest set of unused content (marks further to the right) to see more details. The details are displayed in the bottom left section. This can be your next set of content to consider for archiving or deleting as these are contents that nobody has been using, regardless of size.

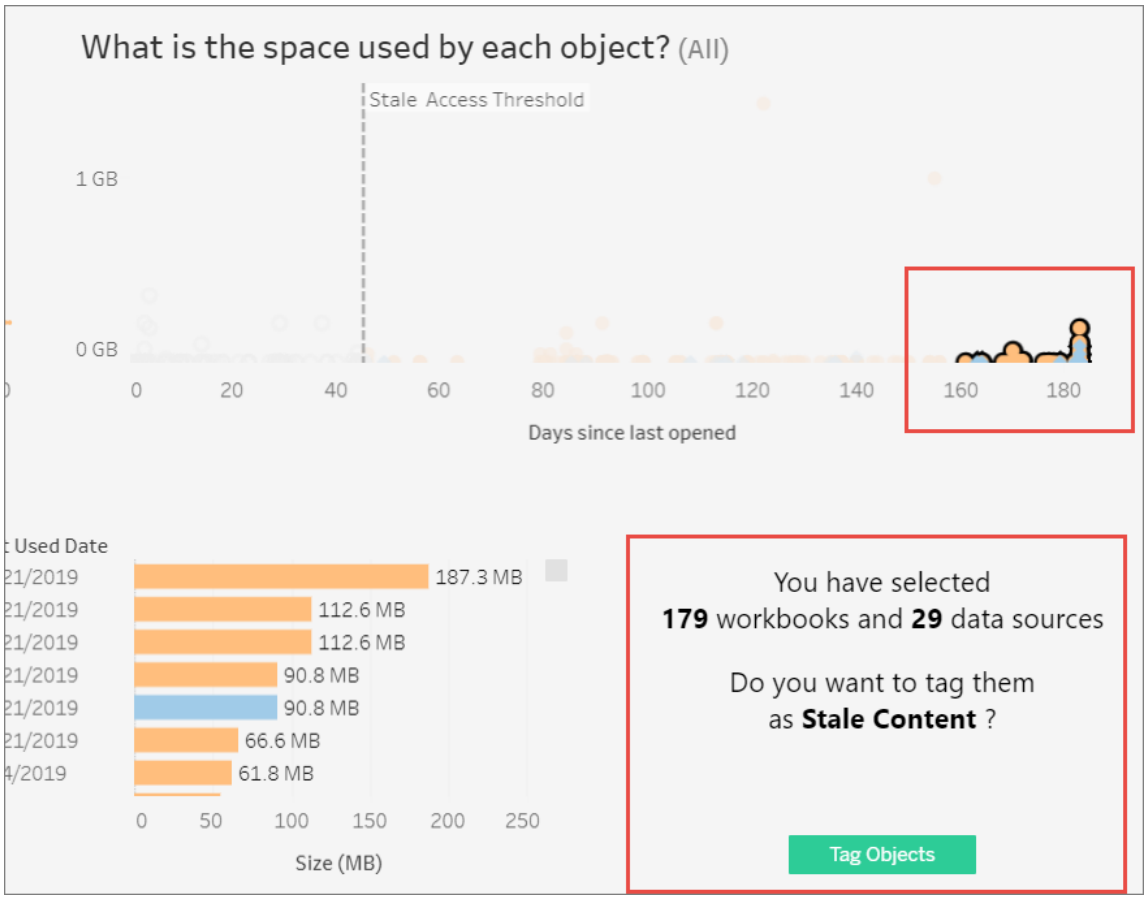


3. **The section at the bottom** shows detailed information about the selected content as shown below:

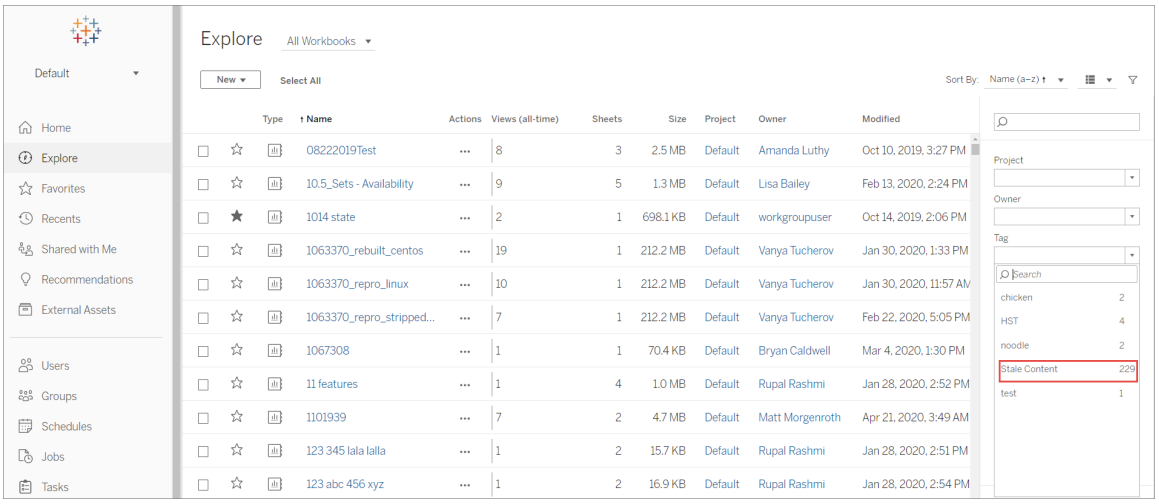
Archive or Delete Stale Content

Starting in 2020.3, the Stale Content admin view includes a feature that allows you to select and tag content as stale. You can select content from either the section at the bottom or the top right section. When you make a selection, you will see the number of objects and the type of content that are selected, as seen in the screen shot below. Click the **Tag Objects** button to tag the selected content.

In the screen shot shown below, content that has not been opened in the last 160 days or more are selected to be tagged as stale content.



To find all the tagged content, on Tableau Server web interface, navigate to **Explore**. Select the **Stale Content** filter to see all the content that have the stale content tag. You can now select the content and either move it to a project that you use for archiving or delete the content.



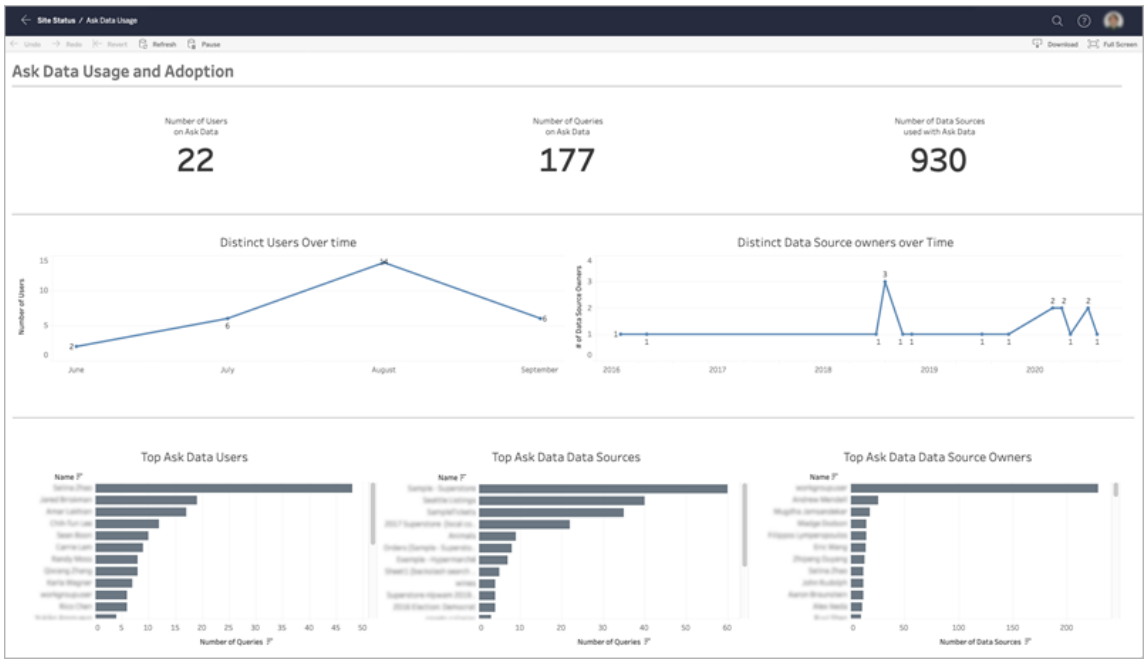
If you have Advanced Management on your Tableau Server, you can use the Tableau Content Migration Tool to manage archiving stale content on a regular schedule. For example, you can build a plan that runs on a regular schedule that can automatically pick up content tagged as Stale Content and move it to an Archive project. After a certain amount of time, the content in this project can be purged from the system. For more information see, Migration Plans: Workbooks.

Ask Data Usage

Important changes for Ask Data and Metrics

Tableau's Ask Data and Metrics features were retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2. With advances in natural language technologies, we're developing an improved interface that will make it easier to ask questions of your data and stay on top of changes. For more information, see *How Tableau AI and Tableau Pulse are reimagining the data experience.*

The Ask Data Usage view is a pre-built dashboard that allows site or server admins to see and understand the usage patterns and value of Ask Data for a site. Admins can see the growth of engagement with Ask Data and monitor the results of internal training or roll-outs. The dashboard highlights the top Ask Data users, data sources, and data source owners, along with some headline value metrics.



To enable Ask Data, see Disable or Enable Ask Data for a Site.

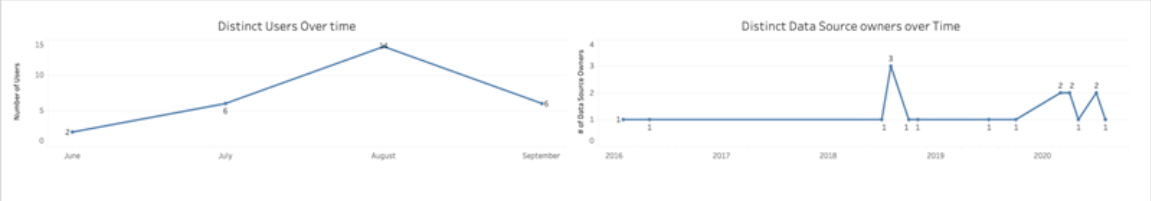
Explore the dashboard

The Ask Data Usage view provides information about Ask Data across the entire site. You can use the following metrics to understand user engagement and help drive self-service analytics adoption in your organization.

At the top of the dashboard, three headline metrics provide an overview of Ask Data usage on the site.

- **Number of Users on Ask Data** - This shows the total number of Ask Data users on the site.
- **Number of Queries on Ask Data** - This shows the total number of Ask Data queries issued on the site.

- **Number of Data Sources Used with Ask Data** - This shows the total number of data sources used with Ask Data.



In the middle of the dashboard, two line charts show you how Ask Data is used over time.

- **Distinct Users Over Time** - This shows the distinct number of Ask Data users over time.
- **Distinct Data Source Owners Over Time** - This shows the distinct number of data source owners over time.

At the bottom of the dashboard, three bar charts list the top Ask Data users, data sources, and data source owners.

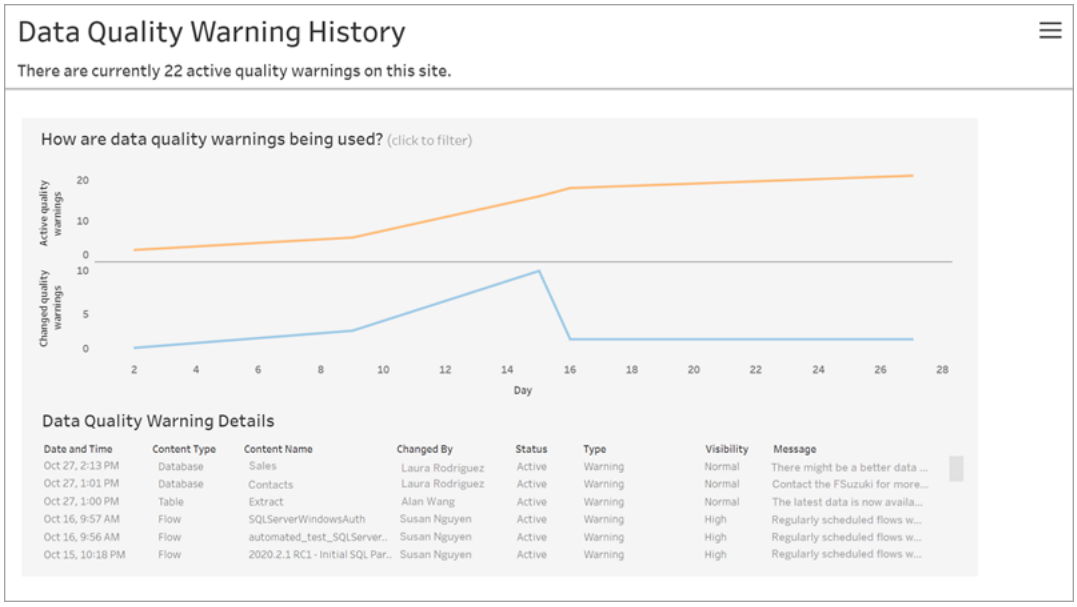
- **Top Ask Data Users** - This lists the top Ask Data users and the total number of queries issued by each user.
- **Top Ask Data Data Sources** - This lists the top Ask Data data sources and the total number of queries issued for each data source.
- **Top Ask Data Data Source Owners** - This lists the top Ask Data data source owners and the total number of data sources owned by each user.

Data Quality Warning History

When Tableau Catalog is enabled in your environment, site administrators can see how data quality warnings are being used on the site using the pre-built admin view, Data Quality Warning History.

For more information about Tableau Catalog, part of Data Management, see "About Tableau Catalog" in the Tableau Server or Tableau Cloud Help.

From the Site Status page, select the Data Quality Warning History dashboard:



The dashboard shows how many data quality warnings are active over a period of time. It also shows how many warnings have been changed (created, updated, and deleted) over that same time period.

See warning details

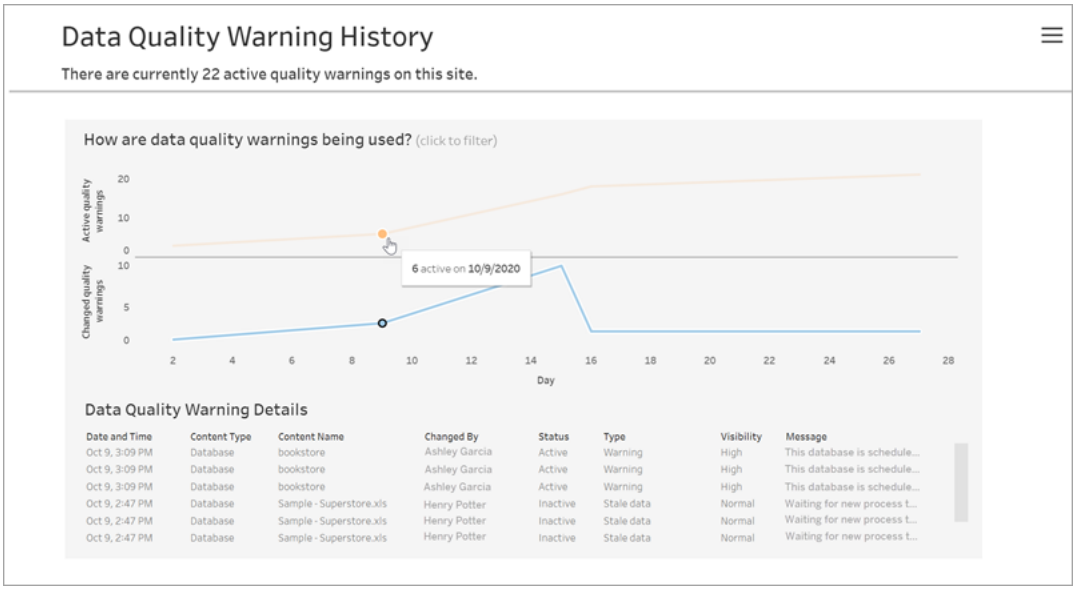
Under the line charts are the details about the data quality warnings, including:

- **Date and Time** - When the warning was created or last changed.
- **Content Type** - The type of asset the warning is set on, such as a database, table, or data source.

- **Content Name** - Name of the asset the warning is set on.
- **Changed By** - Name of the person who created or last changed the warning.
- **Status** - If the warning is active or inactive.
- **Type** - Warning type can be Stale data, Warning, Deprecated, Sensitive data, or Under maintenance.
- **Visibility** - The warning can be configured to have normal (the default) or high visibility.
- **Message** - The message the warning creator wrote to display to users when they see the details of the warning.

Filter warning history

When you review data quality warning history, you can click a mark on the view to filter the details shown below the view.



The numbers on the Day axis represent the date within the time range. For example, if today is November 18, and you filter for the last 7 days, the Day axis shows 12-18.

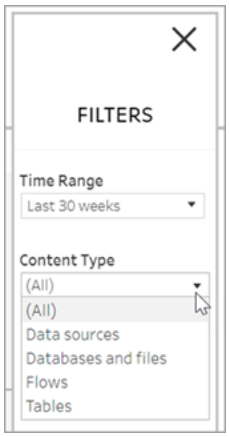
More filters are available when you click the filter icon in the upper right corner: filter by time range and by content type.

Filter by time range

You can configure the time range from years to minutes.

Filter by content type

You can see all the data quality warnings on your site, or you can filter to see warnings for specific types of assets, like data source or table:



Access data quality warning history data

In addition to using the Data Quality Warning History admin view, you can also access data quality warning history data in the "workgroup" PostgreSQL database of the Tableau Server repository. Before you can access this data, you must enable access to the Tableau Server repository.

See About the Tableau Data Dictionary for a link to open the latest data dictionary where you can search for these tables with quality warning history data:

- historical_events
- historical_event_types
- hist_data_quality_indicators
- data_quality_indicators

Who can do this

To set a data quality warning, you must be a server or site administrator.

Create Custom Administrative Views

In addition to the pre-built administrative views available on the Maintenance page on the Server, you can use Tableau Desktop to query and build your own analyses of server activity. To do this, you can connect to and query views in the Tableau Server repository using one of two built-in users: the "tableau" or "readonly" user.

To connect to the Tableau Server repository, see Collect Data with the Tableau Server Repository.

- The **tableau** user—The tableau user has access to special views and a subset of tables in repository database. These views and tables are provided so that administrators can create custom administrative views. Tableau makes an effort to limit changes to these tables and views so that custom views built with them do not break.
- The **readonly** user—The readonly user has access to a large number of the repository tables, providing more data about server usage. Administrators can use these to create custom administrative views too, but many of the tables are intended primarily to support the functioning of Tableau Server and may be changed or removed without warning. This means that views created from these tables can break when the database structure is changed.

For examples of custom administrative views, see the Tableau Community. You can also use the temporary workbook that is generated when you view the built-in Administrative views.

Before you can connect using one of the built-in users, you must enable access to the Tableau Server database. After doing this you can use Tableau Desktop to connect to and query the database as the tableau user or the readonly user.

The `tsm configuration set` option auditing.enabled controls whether Tableau Server collects historical user activity and other information in the repository. It is enabled by default. Be aware that collecting historical events impacts the size of Tableau Server's backup file (.tsbak).

- All `hist_` tables are controlled by the `tsm configuration set` option wgserver.audit_history_expiration_days, which controls how many days of event history are kept in the repository and has default value of 183 days.
- The `_http_requests` table is cleaned of all data older than 7 days when you run tsm maintenance cleanup with the `--http-requests-table` option. For more information, see Remove Unneeded Files.
- The `_background_tasks` table is cleaned automatically and keeps data for the last 30 days.
- All other tables with names that begin with a "_" prefix contain current data.

For more information about the tables in the Tableau Server repository, see Workgroup Database Data Dictionary .

Performance

You can monitor and tune the performance of Tableau Server.

Tableau Server Performance Overview

When you take the time to understand the performance of Tableau Server, you make it easier to serve your users by improving the efficiency of Tableau Server. Although every server environment is unique, and there are many variables that can impact performance, the

general steps that you take to understand and act on performance data in Tableau Server are the same.

- **Notifications**. Configure email notifications for important server events. For example, you can receive notifications when server processes become unavailable and when the server is running out of disk space.
- **Monitoring**. Collect and analyze data about Tableau Server to understand how well the server is performing.
- **Tuning**. Make adjustments to tasks, process configurations, and more to improve the performance of Tableau Server.
- **Troubleshooting**. Identify bottlenecks in resources, workbooks, and more to improve the performance of Tableau Server.

General Performance Guidelines

Hardware and Software

Add more cores and memory: Regardless of whether you're running Tableau Server on one computer or several, the general rule is that more CPU cores and more RAM will give you better performance. Make sure you meet the Tableau Server recommended hardware and software requirements.

If you are running Tableau Server in a virtual environment, use your VM host's best practices for vCPU allocation in relation to the number of physical CPU cores on the VM host.

External repository

For optimal performance for Tableau Server we recommend isolating the repository on a dedicated node in your deployment. If you have an Advanced Management license, consider running the repository as an external database.

If your organization has a peak load of more than 1000 VizQL sessions per hour, we also recommend running Tableau Server on Linux. In this scenario, VizQL sessions refer to any user actions that display or generate visualizations from Tableau Server.

For more information, see Tableau Server External Repository.

Configuration

Schedule refreshes for off-peak hours: Backup tasks tend to stall other background tasks until the backup is completed. Use the Background Tasks for Extracts administrative view to see your refresh and backup task schedules. Your refresh tasks should be scheduled for off-peak hours that don't overlap with your backup window.

Look at caching: Caching helps Tableau Server respond to client requests quickly, especially for views that connect to live databases. Use the tsm data-access caching list command to confirm the caching frequency is set to `low` (this is the default).

Tableau Server uses a query cache to store query results. The size of the query cache is automatically set based on the amount of available system memory, as long as you have not set it manually. The query cache consists of the logical query cache, the metadata cache, and the native query cache. The default settings are suitable for most situations but it is possible to manually configure them using the TSM command line interface. The TSM settings are: `native_api.InitializeQueryCacheSizeBasedOnWeights`, `native_api.QueryCacheMaxAllowedMB`, `native_api.LogicalQueryCacheMaxAllowedWeight`, `native_api.MetadataQueryCachMaxAllowedWeight`, `native_api.NativeQueryCacheMaxAllowedWeight`, and `native_api.QueryCacheEntryMaxAllowedInPercent`. For more information, see native_api.InitializeQueryCacheSizeBasedOnWeights.

Consider changing two session memory settings:

- **VizQL session timeout limit**: The default VizQL session timeout limit is 30 minutes. Even if a VizQL session is idle, it is still consuming memory and CPU cycles. If you can make do with a lower limit, use tsm configuration set Options to change the

`vizqlserver.session.expiry.timeout` setting.

- **VizQL clear session**: By default, VizQL sessions are kept in memory even when a user navigates away from a view. This reduces the need to rebuild views but consumes more session memory. To free up memory, you can end sessions when users leave views by changing the value of the `vizqlserver.clear_session_on_unload` setting to `true`. (Regardless of this setting, sessions for the Tableau Mobile app are always kept in memory, improving mobile performance.)

Assess your process configuration: Tableau Server is divided into six different components called server processes. While their default configuration is designed to work for a broad range of scenarios, you can also reconfigure them to achieve different performance goals. Specifically, you can control on which computers the processes run and how many are run. See Performance Tuning for general guidelines for one-, two-, and three-node deployments.

Server Resource Manager (SRM)

The Server Resource Manager (SRM) monitors the system resources each Tableau process is using as well as tracking the total usage of Tableau Server on the system. If either a specific process or the product as a whole takes up too much system resources, SRM can notify the processes to free the resources or restart those processes.

The thresholds that determine when SRM will notify or restart a process are set in the SRM configuration options. The Tableau development team has set the default settings based on internal testing and don’t recommend you change these settings directly.

If you are seeing excessive system resource usage, we recommend that you contact Tableau Support to help determine if these configuration options need to be modified to solve the problem or issue that you are seeing.

Performance Monitoring Overview

When you monitor a server, you collect and analyze data that signals whether the server is performing badly or running into problems. For example, if you notice that your server is using 100% of its processing capacity for long periods of time, you know that there's a problem.

The data that you need to collect and analyze can be broken down into the following broad categories:

- Resource usage data—how Tableau Server uses hardware resources like diskspace, memory, and processors.
- Session and load time data—how users interact with Tableau Server, including how long it takes for views to load and how many concurrent users there are.
- Background task data—how Tableau Server runs tasks that are not directly tied to a user action. For example, background tasks include extract refresh tasks, subscription tasks, and more.

Some of this data, including load time data and extract refresh data, is already accessible from the administrative views that are built into Tableau Server. However, to collect resource usage data you need to use a external performance monitoring tools. To collect additional load time data and background task data, you can connect to the Tableau Server repository.

For more information on the built-in administrative views, see Administrative Views .

Note: To use the sample workbook and to publish views to Tableau Server, you must have Tableau Desktop.

Collect Data with the Tableau Server Repository

The Tableau Server repository is a PostgreSQL database that stores data about all user interactions, extract refreshes, and more. You can enable access to the repository and use the data in it to help analyze and understand Tableau Server performance.

Looking for Tableau Server on Windows? See Collect Data with the Tableau Server Repository.

After you enable access to the Tableau Server repository, you can create views with data from the repository. The views that you create with this data are sometimes called custom administrative views. In addition to being used for performance monitoring, custom admin views can be used for tracking user activity, workbook activity, and more. For more information on the type of data that you can use for these views, see Create Custom Administrative Views and About the Tableau Server Data Dictionary. Alternatively, if you are only interested in performance data, you can use the preselected database tables in the sample performance workbook.

Enable access to the Tableau Server repository

You can use Tableau Desktop to connect to and query the Tableau Server repository using two built-in users. The user named `tableau` has access to several database views you can use as part of building your own analyses of Tableau Server activity. The user named `readonly` has access to additional database tables that you can use to create views for even more in-depth analysis and this is the user we recommend you use.

Before you can connect to the repository, you need to enable access for the `readonly` user to the database. Use the `tsm data-access repository-access enable` command to enable repository access. When you enable repository access, you also create a password for the `readonly` user. You will use this password to access to the repository. You may also need to have port 8060 opened on the repository node so you can connect to the database.

1. Verify that port 8060 is opened on the computer where the repository is installed. This is a requirement if you are connecting remotely.
2. Enable repository access and create a readonly user password:

   ```
   tsm data-access repository-access enable --repository-username
   readonly --repository-password <PASSWORD>
   ```

   If your password includes special characters, you may need to escape the characters or enclose the password in quotes. Refer to the documentation for the Linux distro that you are running for information about passing special characters in bash shell.

This command will restart Tableau Server.

> **Note:** If you later decide that you want to disable remote access to the Tableau Server repository, use the `tsm data-access repository-access disable` command. The command disables external access to the repository. This will not disable access from localhost. For more information, see tsm data-access repository-access disable.

Connect to the Tableau Server repository

This section describes how to connect to a custom set of tables from Tableau Server repository. For more information on the tables that you can connect to, see About the Tableau Server Data Dictionary.

1. In Tableau Desktop select **Data > Connect to Data**, and then select **PostgreSQL** as the database to connect to.

   > **Note**: You might need to install the PostgreSQL database drivers. You can download drivers from www.tableau.com/support/drivers.

2. In the PostgreSQL connection dialog box, enter the name or URL for Tableau Server in the **Server** box. If you have a distributed server installation, enter the name or IP address of the node where the repository is hosted.

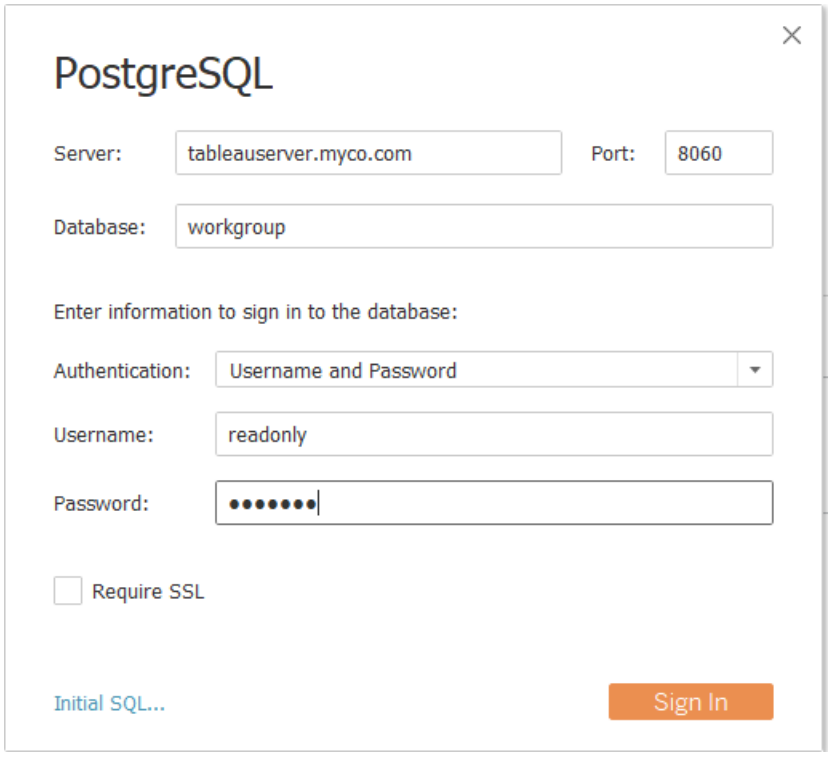
   Connect using the port you have set up for the pgsql.port, which is 8060 by default.

3. Specify `workgroup` as the database to connect to.

4. Connect using the user and the password you specified.

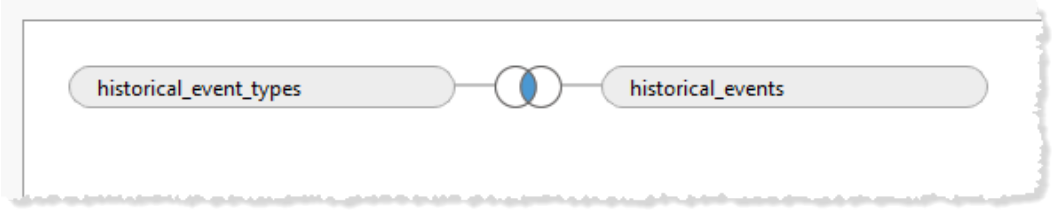
5. Click the **Require SSL** option if you have configured Tableau Server to use SSL for connecting to the repository. For more information, see Configure Postgres SSL to Allow Direct Connections from Clients.

6. Click **Connect**.



7. Select one or more tables to connect to.

   The `tableau` user has access to all of the tables that start with an underscore or with `hist_`. For example, you can connect to `_background_tasks` and `_datasources`. The `hist_` tables include information about server users that isn't currently presented in the Actions by Specific User view. The `readonly` user has access to additional tables that can be used to query other information about server usage.



8. Click **Go to Worksheet**.

PostgreSQL Version

Use the following steps to find the version of PostgreSQL used by Tableau Server:

1. Log into Tableau Server as a user with sudo access.
2. Use the following command to view the version of PostgreSQL installed on the machine:

   $ postgres --version psql --version

If the above command results in an error, you may need to locate the directory. Use the following steps to locate the directory:

1. $ locate find /opt/tableau –name psql
2. Navigate to the path and issue the version command to find the PostgreSQL version:

   $ postgres psql --version

You can also connect to the workgroup database and issue the following query to get the version: `select version()`

About the Tableau Server Data Dictionary

The Tableau Server Data Dictionary includes information about the tables and views in the "workgroup" PostgreSQL database of the Tableau Server repository. This database provides persistent storage for Tableau Server and is primarily intended to support that application. The Data Dictionary is not a complete description of all tables and fields in the database, and is provided for customers who want to query the database for information about usage on

Tableau Server. Because the database and its contents are intended to support Tableau Server, the structure and contents may change without warning. This means any custom views you build from directly querying the database could break.

Important: The Data Dictionary is provided with an **As-Is** level of support. For assistance creating reporting based on the dictionary, including writing queries to the database or updating broken workbooks, please engage with the Tableau Community forums.

Open the Data Dictionary (new window).

Performance Tuning

This section describes how to use the performance data that you collect to identify ways to improve the performance of Tableau Server. Because no two server environments are identical, we can't provide hard and fast rules for tuning server performance. However, you can draw conclusions about performance from patterns in the data that you collected.

For example, are there recurring spikes? Do any of the patterns that you notice in the administrative views correspond to similar patterns in a monitoring tool? Observing patterns like this can guide you in testing and incremental tuning.

Most performance tuning for Tableau Server boils down to these general approaches:

- **Optimize for User Traffic:** This tunes the server to respond to user requests and to display views quickly.
- **Optimize for Extracts:** This tunes the server to refresh extracts for published data sources. You might want to optimize for extract refreshes if your organization has a lot of data and the data needs to be as up to date as possible.
- **Optimize for Extract Query-Heavy Environments:** This is a specialized server configuration to optimize for query performance of workbooks that use extracts as their data source.

Rendering views and refreshing extracts generate the most load on the server, so you should optimize for the task that your organization is most interested in.

As a best practice, optimize your workbooks for performance. For more information and resources on how to optimize your workbooks, see Optimize Workbook Performance.

Optimize for User Traffic

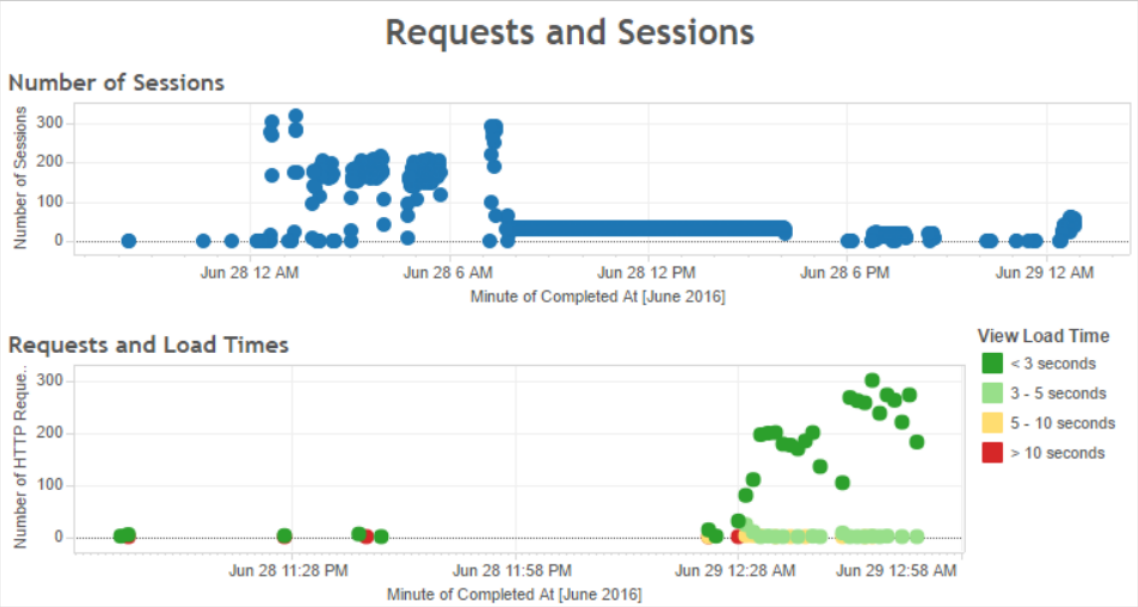
You can optimize for traffic if you have many active Tableau Server users and few published data sources that need extract refreshes.

- When to optimize for user traffic
- Ways to optimize for user traffic

When to optimize for user traffic

Slow load times for views

Use the **Requests and Sessions** dashboard of the sample performance workbook to analyze how long views take to load.

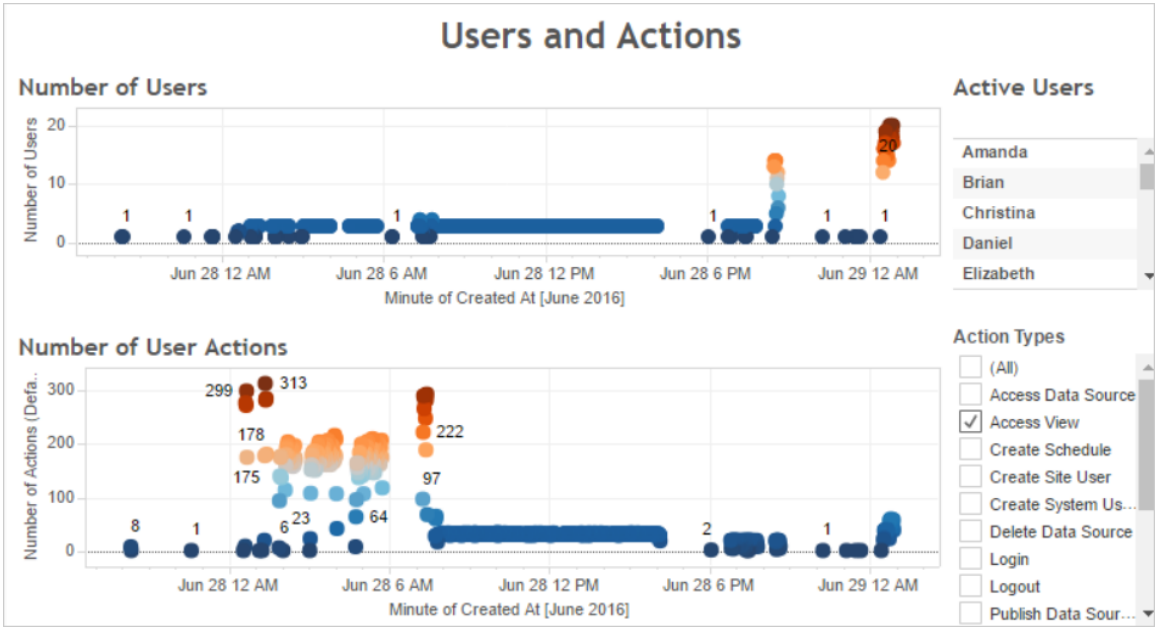


If multiple views take longer than 10 seconds to load, and if the slow load times correspond to a large number of sessions, that can indicate that user traffic is slowing down the server.

However, if a particular view takes a long time to load regardless of when it is viewed, it's a sign that the workbook for the view needs to be optimized. You can identify which workbooks need to be optimized with the Stats for Load Times administrative view. Some simple ways of optimizing workbooks includes displaying less information in each view or breaking up views, reducing the number of filters, and using data extracts.

High resource usage corresponding to user traffic

If your server displays high CPU and memory usage during peak traffic hours, you should optimize for user traffic. To determine peak traffic hours and analyze how many concurrent users are on your server, use the **Users and Actions** dashboard. In addition, you can use the Traffic to Views administrative view to see how much user traffic involves accessing views (as opposed to performing administrative functions, publishing, or other tasks).

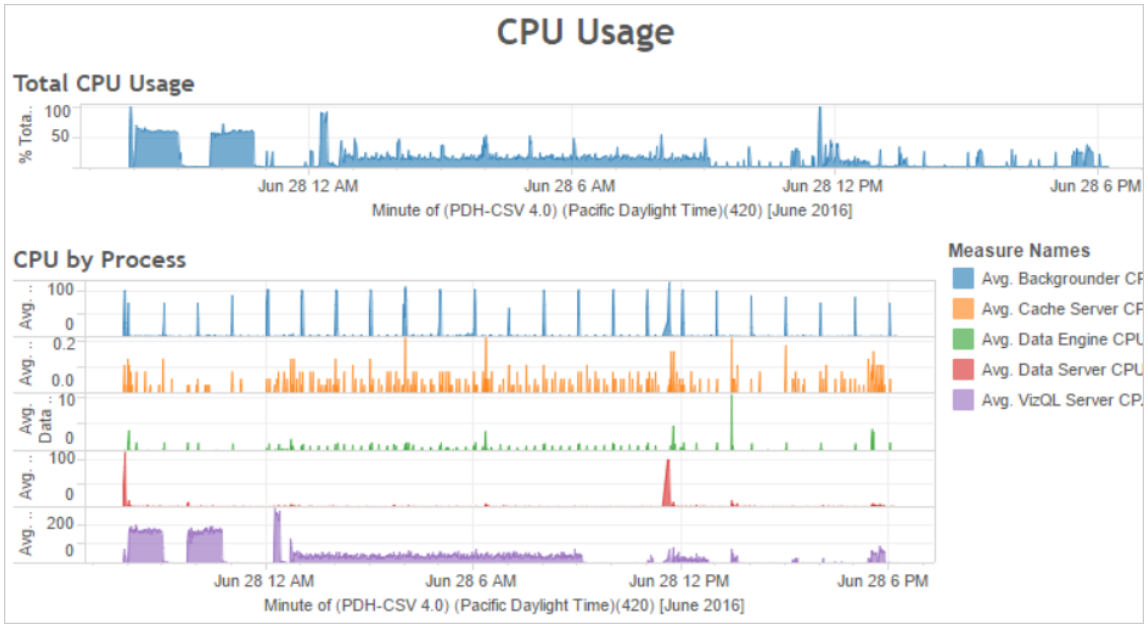


If you click a point in the **Number of Users** view, the dashboard displays the users that were active at the time and the number of user actions that those users performed. By default, the only user actions displayed are user views, but you can use the **Action Types** filter to display additional user actions.

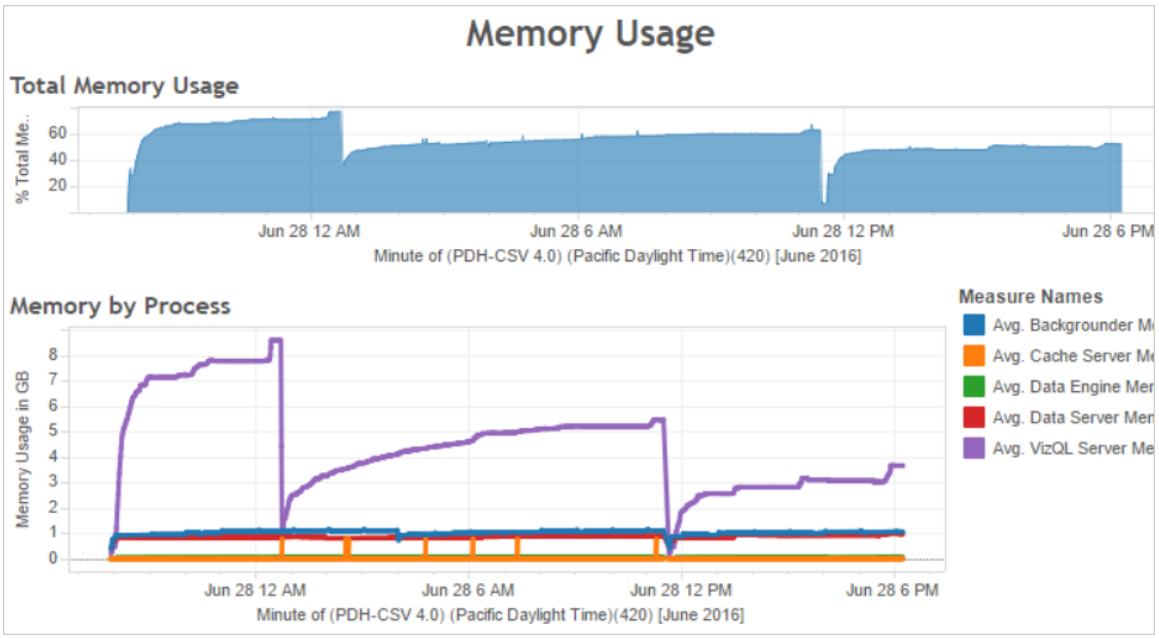
Make a note of the times of day when there are many concurrent users and views so that you can compare this to resource usage. As a rule of thumb, the number of users should correspond to a high number of user actions. However, the view in this example displays an artificially high number of actions for a single user as part of a load generation test. As an example, you can compare the high number of views at 12 AM on June 28th with the resource usage in the dashboard illustrated later.

Use the **CPU Usage** dashboard to display the percent of total CPU usage and the percent of CPU usage for each process. In the following example, note the large spike in total CPU usage and in the VizQL server process at 12 AM on June 28th. Because the VizQL server process loads and render views, the VizQL server process is often the first process to show strain under high user traffic.

> **Note**: The percent of CPU usage for individual processes may add up to more than 100 percent. This is because processor utilization for individual processes is measured for a given processor core. By contrast, the total CPU usage is measured for all processor cores.



Use the **Memory Usage** dashboard to display the percent of total memory usage and the average memory usage in gigabytes. As a general rule, memory usage increases steadily with user traffic. Here again the VizQL server process is the first to show strain under high traffic.



Ways to optimize for user traffic

When high user traffic corresponds to high resource usage as it does in the example shown previously, you should optimize for user traffic.

Adjust the number of VizQL server processes

The most effective way of optimizing for user traffic is to adjust the number of VizQL server processes. Add one VizQL server process at a time and measure the effect with more performance monitoring. Because VizQL server processes can consume a lot of CPU and memory, adding too many processes can slow down the server instead. If you see consistently high memory usage, try to reduce the number of VizQL server processes to reduce the amount of memory reserved.

For more information about configuring processes, see Configure Nodes.

Adjust the number of other processes

Although the most effective way of improving performance for user traffic is to adjust the number of VizQL server processes, you can also tune other processes that support the VizQL server process or that prevent the VizQL server process from accessing resources. For example, the VizQL server process makes frequent requests to the cache server process, so you might also want to increase the number of cache server processes. On the other hand, the Backgrounder processes might contend for CPU resources with the VizQL server process. As a result, if you do not need to run frequent extract refreshes, you might reduce the number of processes for the backgrounder. If you do need additional instances of the backgrounder, and if you're running Tableau Server on a cluster, you can move the Backgrounder process to a dedicated node.

Adjust the VizQL session timeout limit

In the example shown previously, the amount of memory used by the VizQL server process increases with user traffic, and it remains reserved by Tableau Server for some time after the traffic finished. This is because the VizQL server process reserves memory for each session for a specified amount of time. If the VizQL server process uses a high percentage of the available memory, try reducing the timeout for each session to make memory available more quickly.

To do this, use the `tsm configuration set` command to reduce the `vizqlserver.session.expiry.timeout` setting. The default is 30 minutes.

Refresh the cache less often

If your users do not always need the most up-to-date data, you can optimize for user traffic by configuring Tableau Server to cache and reuse data as much as possible.

To do this, use the `tsm data-access caching list` command to confirm the refresh frequency. The default is `Low`. Use the `tsm data-access caching set` command to change the refresh frequency.

Assess view responsiveness

When a user opens a view, the components of the view are first retrieved and interpreted, then displayed in the user's web browser. For most views, the display rendering phase occurs in the user's web browser and in most cases, this yields the fastest results and highest level of interactive responsiveness. Handling most interactions in the client web browser reduces bandwidth and eliminates round-trip request latencies. If a view is very complex, Tableau Server handles the rendering phase on the server instead of in the client web browser, because that generally results in the best performance. If you find that views aren't as responsive as you'd like, you can test and change the threshold that causes views to be rendered by the server instead of in the client web browser. For more information, see Configure Client-Side Rendering.

Configure Client-Side Rendering

When you navigate to a view in Tableau Server, the processing required to display the view, called *rendering*, can be performed by either your client device or Tableau Server. The choice depends on the complexity of the view, which is determined by the number of marks, rows, columns, and more. If a view is less complex, it's faster for a client device to render the view. If a view is more complex, it's faster to send a request to Tableau Server and take advantage of the server's greater computing power.

> **Note:** If a view uses the polygon mark type or the page history feature, server-side rendering is always performed, even if client-side rendering is enabled.

Supported browsers

Client-side rendering is supported in Internet Explorer version 9.0 or higher, Firefox, Chrome, and Safari. All of these web browsers include the HTML 5 `<canvas>` element, which client-side rendering requires.

Client-side rendering is also supported by the Tableau Mobile app.

Configure the complexity threshold for computers and mobile devices

Because computers have more processing power than mobile devices, Tableau Server performs more client-side rendering on computers than on mobile devices.

As a server administrator, you can configure when client-side rendering happens on computers and mobile devices by adjusting the complexity threshold for each. For example, you might lower the threshold for mobile devices if you notice that views display slowly on them. Or, you might increase the threshold to reduce the number of requests to Tableau Server.

By default, the complexity threshold for computer web browsers is 100. To adjust the complexity threshold for computers, use the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold -v
[new value]
```

By default, the complexity threshold for mobile devices is 60. To adjust the complexity threshold for mobile devices, use the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold_
mobile -v [new value]
```

For example, to change the mobile threshold to 40, you might enter the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold_
mobile -v 40
```

For more information on how to use tsm option set, see tsm configuration set Options.

Disable client-side rendering

Client-side rendering is enabled by default and is recommended to improve the performance of views. However, you might want to disable client-side rendering temporarily for testing or if your server is being accessed primarily by computers or mobile devices with very little processing power.

Use the following command to disable client-side rendering:

```
tsm configuration set -k vizqlserver.browser.render -v false
```

For more information on how to use tsm option set, see tsm configuration set Options.

Testing with the URL Parameter

To test server-side rendering on a session basis, type `?:render=false` at the end of the view's URL. For example:

```
http://localhost/views/Supplies/MyView?:render=false
```

If client-side rendering is disabled on Tableau Server, enter `?:render=true` to enable it for the session:

```
http://localhost/views/Supplies/MyView?:render=true
```

You can also test particular complexity thresholds on individual views to see if it's appropriate to adjust the server-wide threshold for your server and network conditions. For example, you may find that lower complexity (such as 80) or higher complexity (such as 120) tipping points result in more responsiveness to user interactions. To test a threshold, you can keep the server's default configuration (client-side-rendering enabled) and enter the test threshold number at the end of the view's URL. For example:

```
http://localhost/views/Supplies/MyView?:render=80
```

Optimize for Extracts

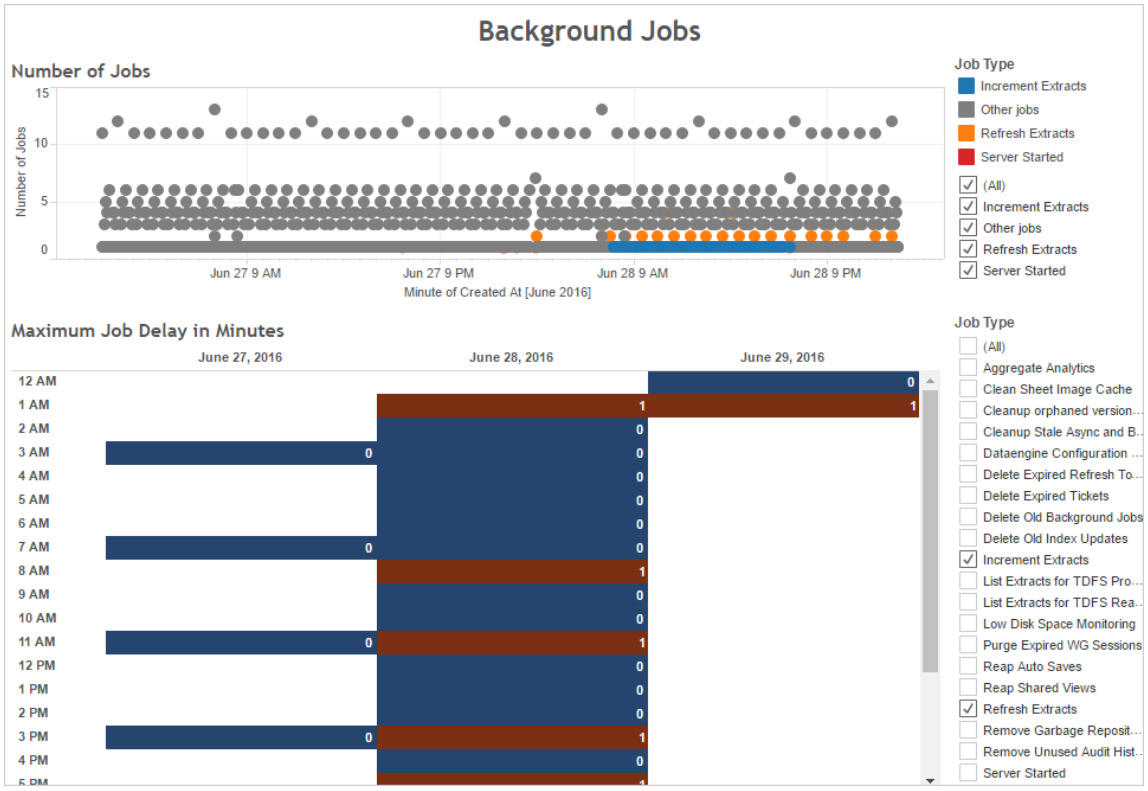
Try to optimize for extracts if the extract schedules correspond to high resource usage or if extracts take a long time to finish.

When to optimize for extracts

High CPU usage corresponds to extract schedules

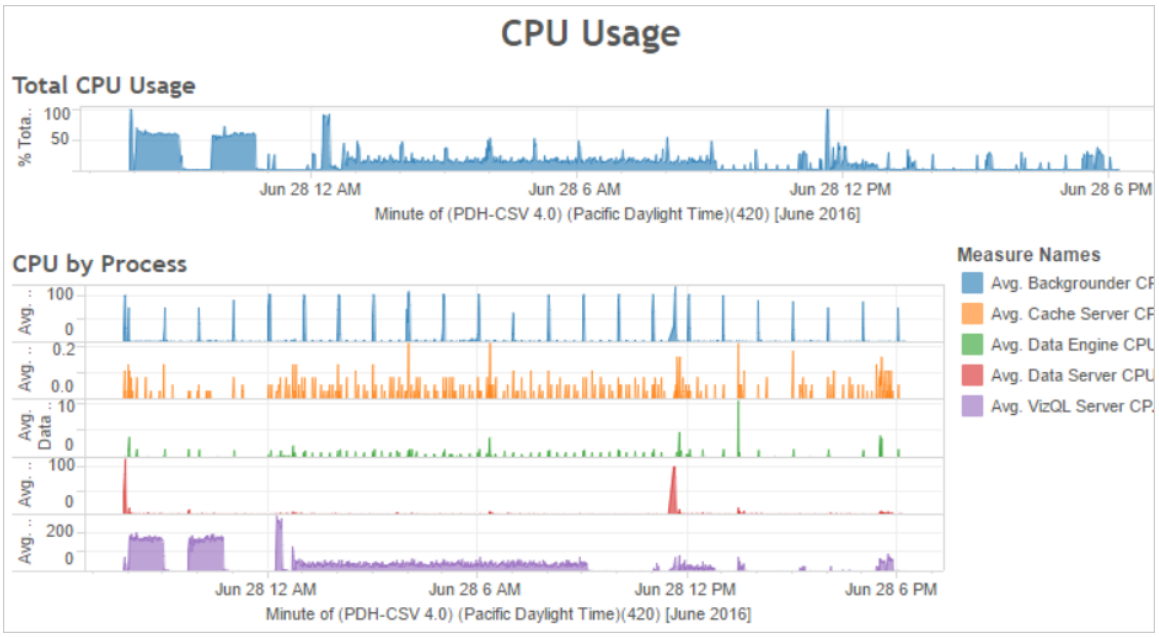
Use the **Background Jobs** dashboard of the sample performance workbook to view the number of background jobs run by Tableau Server, including extract refresh jobs. The dashboard also displays how long background jobs are delayed—that is, the amount of time between when a background job is scheduled and when it actually runs. If you see long delays at par-

ticular times of the day or if many jobs are running at the same time, try distributing the job schedules across different times of the day to reduce the load on the server.



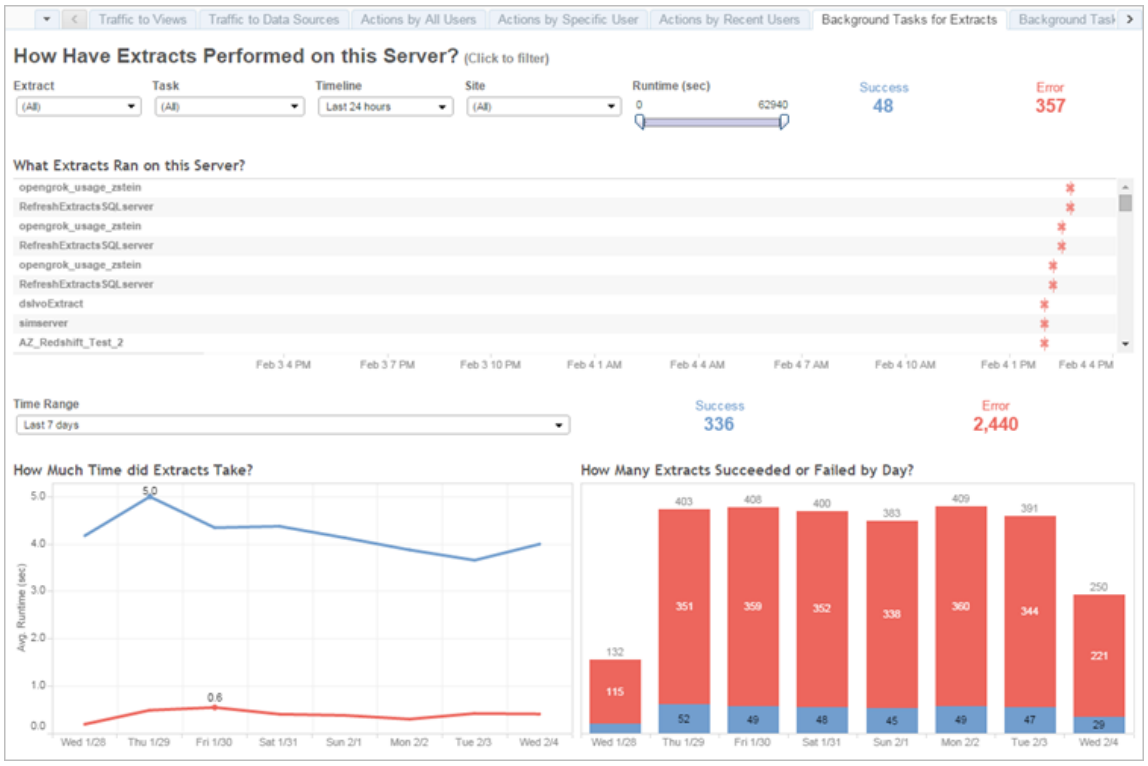
Also compare the times when there are many background jobs or long delays with the CPU usage of the server. Use the **CPU Usage** dashboard to display the percent of total CPU usage and the percent of CPU usage for each process. Because the backgrounder process runs background jobs, it is the first process to show strain when there are many extract refresh jobs or when there are slow extract refresh jobs. Note that the CPU usage of the backgrounder process periodically but briefly reaches 100 percent. This indicates that there are intensive refresh jobs on a recurring schedule.

Note: The percent of CPU usage for individual processes may add up to more than 100 percent because processor utilization for individual processes is measured for a given processor core. By contrast, the total CPU usage is measured for all processor cores.



Extracts fail or run slowly

Use the Background Tasks for Extracts administrative view to determine how many extracts fail and how long extracts take to complete. Frequent failures can indicate a problem with a particular data source.



Ways to optimize for extracts

When high CPU usage corresponds to extract refresh schedules like it does in the example shown previously, you should optimize for extracts.

Adjust the extract refresh schedule

Use the **Background Jobs** dashboard of the sample performance workbook to identify optimal times for running extracts. In addition to running extracts in off-peak hours, you can distribute extract refreshes to minimize concurrent server load. If extract refreshes continue to cause problems, reduce the frequency of extract refreshes as much as possible in these ways:

- Schedule extracts for times when the server isn't busy.
- Reduce the frequency of refreshes.

Speed up specific extracts

Use the Background Tasks for Extracts administrative view to identify failing extracts and long-running extracts.

- Reduce the size of extracts. You can help improve server performance by keeping the extract's data set short, through filtering or aggregating, and narrow, by hiding unused fields. To make these changes, use the Tableau Desktop options **Hide All Unused Fields** and **Aggregate data for visible dimensions**. For more information, see Creating an Extract in the Tableau Help.

  For general tips on building well-performing workbooks, search for "performance" in the Tableau Help. To see how workbooks perform after they've been published to Tableau Server, you can create a performance recording. For more information, see Create a Performance Recording.

- Use incremental refresh jobs. Incremental refresh jobs append new rows to an existing extract instead of creating the extract from scratch. This type of extract refresh runs quickly because it processes only the data that has been added since the last time the extract refresh job ran. However, it does not account for data that has been updated rather than appended to a data source. As a result, if you run incremental refresh jobs, you should still occasionally run full refresh jobs. For example, you might run a full refresh job once or twice a week for a data source instead of every day.

Configure the execution mode for extract refreshes

When you create extract refresh schedules, ensure that they run in parallel execution mode. When you run a schedule in parallel, it runs on all available backgrounder processes, even if the schedule contains only one refresh task. When you run a schedule serially, it only runs on one backgrounder process. By default, the execution mode is set to parallel so that refresh tasks finish as quickly as possible.

However, in some circumstances, it can make sense to set the execution mode to serial. For example, you might set the execution mode to serial if a very large job is preventing other schedules from running because it uses all available backgrounder processes.

Increase the number of backgrounder processes

A single background process can consume 100 percent of a single CPU core for certain tasks. As a result, the total number of instances you should run depends on the computer's available cores. If you have Tableau Server installed in a cluster and you run backgrounder processes on a separate node, a good rule of thumb is to set the number of backgrounder process to between half the number of cores and the full number of cores of the computer running the backgrounder processes.

For more information about configuring processes, see Configure Nodes.

Isolate processes

If you have Tableau Server installed in a cluster, you see the largest benefit from moving the backgrounder processes to a separate node to avoid resource contention. This is because the backgrounder process is very CPU-intensive and running it on the same node where other CPU-intensive processes are running can slow down the server. For example, both the VizQL server process and the data engine process can be CPU-intensive. Read the two-node configuration in the Recommended Baseline Configurations topic for more details.

Optimize for Extract Query-Heavy Environments

The topic provides guidance on setting up a specific Tableau Server topology and configurations to help optimize and improve performance in an extract query-heavy environment.

What is an extract query-heavy environment? Extracts and federated data sources are queried while loading workbooks, views, and dashboards creating a lot of query workload. Therefore, if you have a lot of extracts and federated data sources, you can be said to have a 'extract query-heavy environment'.

If your environment is extract query-heavy as defined above, the next couple of sections can help you decide if this configuration is right for you.

When to use this configuration

Key reasoning behind this configuration: Hyper is Tableau's memory-optimized Data Engine technology suited for fast data ingests and analytical processing, making it key to optimizing query-heavy workloads. As your extract use grows, we recommend **configuring Data Engine on dedicated nodes of the Tableau Server cluster**. This configuration allows Tableau Server to scale-out the infrastructure to optimize performance when querying extracts.

There are several factors that affect Tableau Server performance when viewing content using extracts and federated data sources. The goal here is to **achieve consistent and reliable query performance** when viewing content on the Server. Use this configuration if one of the following conditions apply to your environment:

- You are seeing wide variability in workbook load times and the workbook uses extracts or federated data sources.
- Your Tableau Server deployment is growing in the number of Creators, Explorers, Viewers, and extract-based content, so you want to scale out efficiently.
- You are seeing resource contention between Data Engine and VizQL Server when File Store is present on the machine.
- You analyze large amounts of data. This configuration helps in optimizing performance in big data scenarios, in both data ingestion and analysis. To learn more about Tableau and big data, see Hyper-charge big data analytics using Tableau.

Note: Use Server-side performance recording to determine query execution times. To determine resource usage of Tableau use **Performance monitor** for Windows installations, and *sysstat* or *vmstat* tools for Linux installations.

Benefits of using this configuration

These are the key benefits to configuring dedicated nodes for Data Engine:

- Dedicated Data Engine nodes will reduce resource contention between extract queries and other resource-intensive workloads such as those processed by VizQL Server.
- Extract queries are load balanced dynamically on the dedicated nodes, taking into account the current state of the system to ensure that no one node is over or under-utilized.
- More consistent performance in user experience when loading extract-dependent workbooks. The focus here is to establish a consistent and reliable performance rather than making individual queries better.
- You have more control over scaling out Tableau Server processes that need more resources. If VizQL Server, Data Engine, and Backgrounder are all running on the same node and slow extract queries are the problem, it will be difficult to see performance improvements by adding a second node with all three processes. With this configuration, you can add more nodes that will specifically improve extract query workloads.
- Helps improve availability and up-time. In the event of a failure and one of the dedicated Data Engine nodes is unavailable, VizQL Server will attempt to route the pending requests on the problem node to other dedicated Data Engine nodes.
- Data Engine leverages as many cores as available on the machine. Given this, you have the flexibility to add more resources to the dedicated Data Engine nodes to reduce query response time and variability on expensive extract queries or add more dedicated Data Engine nodes to get more extract query throughput in your Server.
- Data Engine has a default configuration limiting it to an average 75% of CPU per hour. This is intended to help avoid contention with other Tableau Server processes. If you are running Data Engine on a dedicated node, you can increase this average to 95%. For information on doing this, see hyper.srm_cpu_limit_percentage.

When not to use this configuration

- If you are not experiencing issues with extract-based query load, hardware resources may be better allocated to other portions of Tableau Server.
- On nodes where File Store, Data Engine, and VizQL Server co-exist, you are not seeing resource contention between Data Engine and VizQL Server.
- Before implementing this configuration, it is highly recommended that you evaluate your CPU usage for VizQL Server and for the node where Data Engine that installed with the File Store.

Configuration

The main goal of this configuration is to have Data Engine on one or more dedicated nodes.

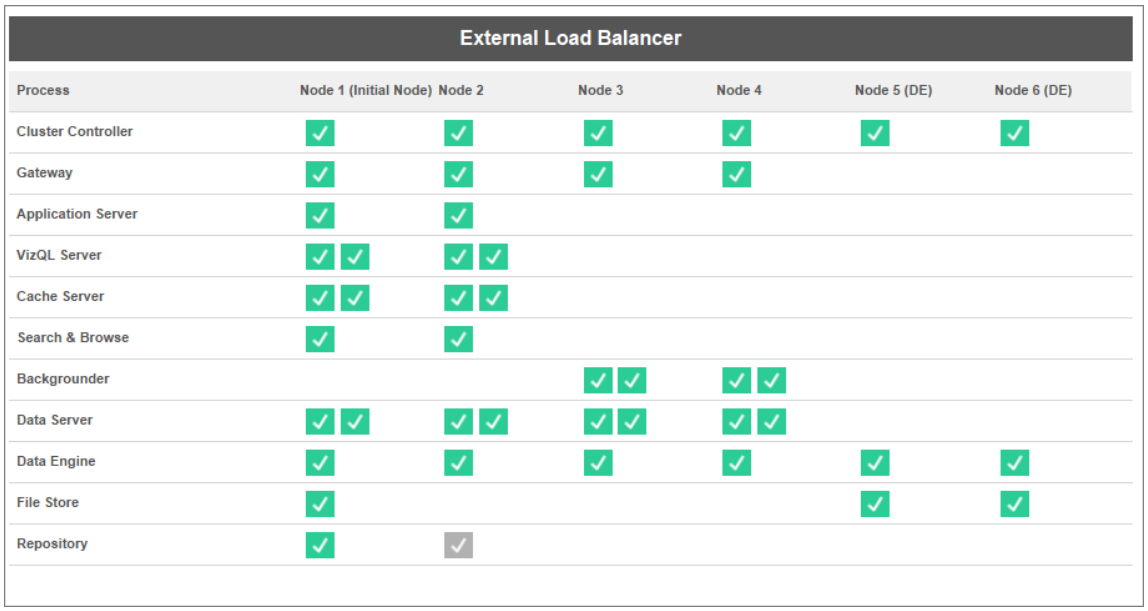
- In deployments where File Store is installed locally, this means configuring File Store on one or more dedicated nodes. Data Engine is automatically installed on the same node as the File Store.
- In deployments where you are configuring External File store, you can still configure Data Engine on dedicated nodes on Tableau Server.

By separating VizQL Server and File Store processes, the load between querying extracts and viewing or interacting with views can be balanced and better managed. This configuration is targeted at consistent performance when querying extracts.

Below is a visual representation of the configuration where the Data Engine/File Store processes have two dedicated nodes, nodes 5 and 6. This is an example where File Store is configured locally which is why the Data Engine and File Store processes are co-located.

The same configuration works for deployments with External File Store, but Node 5 and 6 will have only Data Engine configured in that case.

Additionally, since Node 1 also has the Repository and File Store processes, all of the data needed to perform a backup exist on Node 1 which can improve backup performance.



Hardware Guidance

To get the most out of this configuration, you will need to experiment with various hardware sizes and configurations to see what best fits your peak load performance objectives. Hyper is a high-performance database technology and the key resources that impact performance are memory, cores, and storage I/O. Understanding how Hyper uses resources to process queries will help you make your hardware selection and understand the reason between different configurations.

- **Memory:** When an extract-based query is processed for a user or background process, Tableau Server selects a dedicated Data Engine node to process the query. That dedicated Data Engine node will then copy the extract from local storage, most often the server hard disk, into memory. Having more available system memory allows the operating system better manage memory usage for Tableau. Dedicated Data Engine nodes uses system memory to store the result set of executed queries. If the result set is still valid and the operating system has not cleared it from memory, the result set in memory can be reused.

  Tableau Server's minimum hardware recommendation is 32 GB of memory but if you are expecting a high volume of extract-based workbook loads, you should consider 64

GB or 128 GB. If you are hitting other resource limits in addition to memory (like cores), instead of scaling up to 128 GB of memory, it might be better to scale out to an additional 64 GB dedicated Data Engine node.

The process of copying the extract from local storage into memory can take time and optimizing disk performance may be necessary. Optimizing disk performance is covered in the **Storage I/O** section.

- **Cores:** When processing an extract-based query, the number of cores is an important hardware resource that can impact performance and scalability. CPU cores are responsible for executing a query and having more available cores will result in faster execution time. Generally speaking, doubling the number of cores will reduce the query execution time in half. For example, a 10 second query currently utilizing 4 physical cores or 8 vCPUs, will take 5 seconds if you upgrade to 8 physical cores or 16 vCPUs.

  The current Tableau Server minimum hardware recommendation is 8 cores, but if your deployment utilizes extracts, consider 16 or 32 core machines. An important thing to note is that if memory and I/O are your bottlenecks, then increasing available cores will not improve your query performance.
- **Storage I/O:** Hyper is designed to leverage the available performance of your extract storage device to speed up query processing. We recommend picking fast disk storage like Solid State Drives (SSD) with high read/write speeds. Currently, SSDs that utilize NVMe storage protocol offers the fastest available speeds.

Note: Sizing resources for dedicated Data Engine nodes only impacts the extract query performance. When loading a workbook, there are many other processes involved that make up total VizQL load request time. The VizQL Server process, for example, is responsible for taking the data from the Data Engine and rendering the visualization.

Other Performance Tuning and Optimizations:

There are additional features you can use to optimize performance beyond the basic configuration described above. The optimizations described below are applicable to both local File

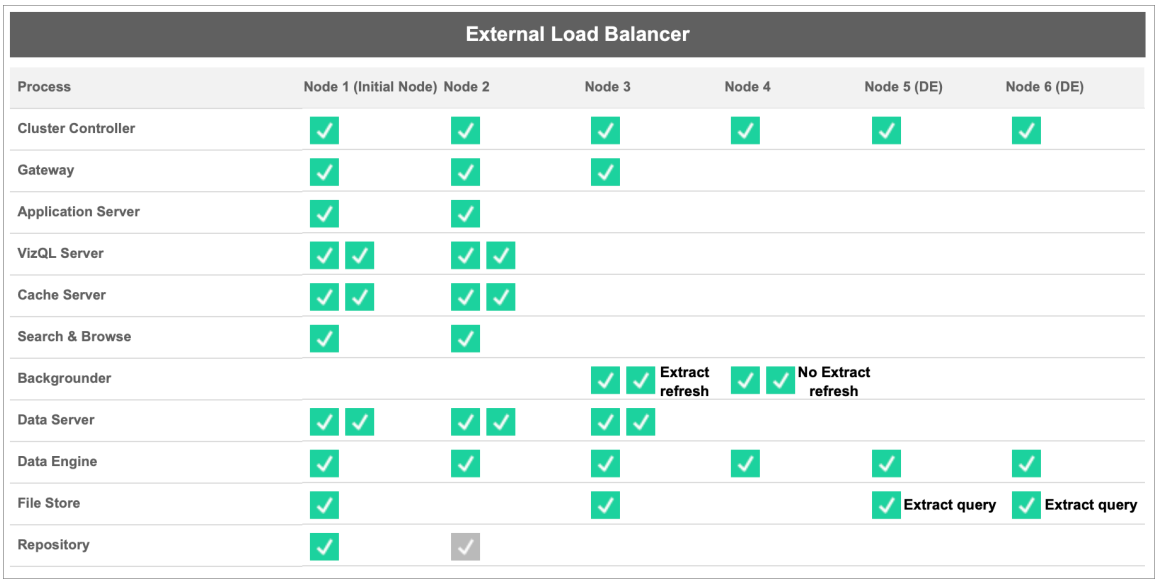
Store and External File Store deployments.

- **Extract Query Load Balancing:** To determine where to route the extract query, Data Engine uses a server health metric- the amount of resources Data Engine is consuming and the load from other Tableau processes that may be running on the same node. In addition to evaluating system resources, whether an extract already exists in memory on the node is also taken into account to make sure an extract query is sent to the node that has the most available resources to process the query. This results in more efficient memory and disk utilization and extracts are not duplicated in memory across nodes. See the Extract Query Load Balancing help article for more details.

  The extract query load balancing feature is enabled by default in Tableau Server version 2020.2 and later..

- **Workload optimizations using node roles:** With Backgrounder and File Store node roles, server administrators have more flexibility and control over which nodes should be dedicated for running extract queries and extract refreshes. As mentioned in the topology diagram above, certain Data Engine nodes are dedicated to processing extract queries and run only the File Store and Data Engine processes. Node Roles are available with Advanced Management. For more information on node roles, see Workload Management through Node Roles.

The diagram below uses the same topology as the basic configuration described above but with the node roles.



- **Extract Refreshes Backgrounder node role:** By setting Node 3 to extract-refreshes Backgrounder node role, only incremental refreshes, full refreshes, and encryption/decryption jobs will run on this node. By setting Node 4 to no-extract-refreshes Backgrounder node role, all background jobs other than extract refreshes will run on this node. Data Server and Gateway help the extract refresh jobs when using federated and shadow extracts. For more information on Backgrounder node roles, see File Store node roles.

  Additionally, since Node 1 also has the Repository and File Store processes, all of the data needed to perform a backup exist on Node 1 which can improve backup performance.

  The Backgrounder node roles are available with Advanced Management in Tableau Server version 2019.3 and later.

- **Extract Queries File Store node role**: Node 5 and 6 which are the **dedicated** Data Engine nodes have the extract-queries File Store node role to ensure they only process queries for viz loads, subscriptions, and data-driven alerts.

- **Extract Queries Interactive File Store node role**: For dedicated Data Engine nodes which have extract-queries File Store node role, server administrators can further

isolate the interactive and scheduled workloads to run on specific **dedicated** Data Engine nodes. This is useful for times when there are a lot of users interacting and loading workbooks during high volume subscription times. For example, let's say there are 1000 subscriptions scheduled for the 8 AM Monday mornings. At the same time, many users are also loading dashboards at the beginning of their day. The combined volume of subscription and user queries can result in users experiencing slower, more variable workbook load times. With the extract-queries-interactive File Store node role, you can designate dedicated Data Engine nodes to only accept queries for interactive users (the ones who are looking at their screens waiting). These dedicated Data Engine nodes that are prioritized for interactive workloads would be protected from the high volume of competing subscription jobs and provide more consistent query times. Additionally, Server Admins can use this node role to better plan for growth since they can add dedicated Data Engine nodes for interactive and scheduled workloads independently. For more information, see File Store node roles.

The File Store node roles are available with Advanced Management in Tableau Server version 2020.4 and later.

- **Optimizations using External File Store**: This feature allows you to use a network share as the storage for File Store instead of using the local disk on a Tableau Server node. By having the storage on a centralized location, you can significantly reduce the amount of network traffic spent on replicating data between the File Store nodes. For example, in the case when File Store is using a local disk, when a 1 GB extract is refreshed using local File Store, the 1 GB of data is replicated across the network to all nodes that are running the File Store process. In the case where Tableau Server is configured with External File Store, the 1 GB extract only needs to be copied to the network share once and all File Store nodes can access that single copy. The centralization of storage also reduces the total amount of local storage needed on File Store nodes.

  Additionally, Tableau Server backups leverage snapshot technology to significantly reduce the time to complete a backup.

While you don't need a dedicated Data Engine node configuration to gain the benefits of External File Store, the additional workload management features with File Store node role and the Extract Query Interactive node role can be used together. See the Tableau Server External File Store topic for more details.

External File Store is available with Advanced Management in Tableau Server version 2020.1 and later.

When to Add Nodes and Reconfigure

Tableau Server can scale up and out as your needs and requirements evolve. Here are some guidelines to help you figure out whether it's time to add more nodes to your system, reconfigure the server, or both:

- **More than 100 concurrent users**: If your deployment is user-intensive (>100 simultaneous viewers), it's important to have enough VizQL processes—but not so many that they exceed your hardware's capacity to handle them. Also, enabling the Tableau Server Guest User account can increase the number of potential simultaneous viewers beyond the user list you may think you have. The administrative view can help you gauge this. For more information, see Actions by Specific User.
- **Heavy use of extracts** and **frequent extract refreshes**: Extracts can consume a lot of memory and CPU resources. There's no one measurement that qualifies a site as extract-intensive. Having just a few, extremely large extracts could put your site in this category, as would having very many small extracts. Sites where extracts are frequently refreshed (for example, several times a day) are often helped by more emphasis on the background process, which handles refresh tasks. Use the Background Tasks for Extracts administrative view to see your current refresh rate. Extract heavy sites benefit from isolating the Backgrounder process on its own machine. For more information, see the two-node configuration in the Recommended Baseline Configurations topic.

- **Query heavy environments**: If you are experiencing slow query performance for Workbooks that use extracts, isolating nodes that handle queries on extracts from VizQL processes can improve and stabilize performance. For more information, see Optimize for Extract Query-Heavy Environments.
- **Downtime potential**: If your server system is considered mission critical and requires a high level of availability, you can configure it so there's redundancy for the server processes that handle extracts, the repository, and the gateway.

Performance Recording

This section describes how to create performance recordings and use the results to improve workbook performance. With performance recordings, you can view how long workbook events take. For example, you can see how long it takes to connect to a data source, run a query, render data, and more.

Create a Performance Recording

The Performance Recording feature in Tableau records performance information about key events as you interact with a workbook. You can then view performance metrics in a workbook that Tableau creates to analyze and troubleshoot different events that are known to affect performance:

- Query execution
- Compiling query
- Geocoding
- Connections to data sources
- Layout computations
- Extract generation

- Blending data
- Server rendering (Tableau Server only)

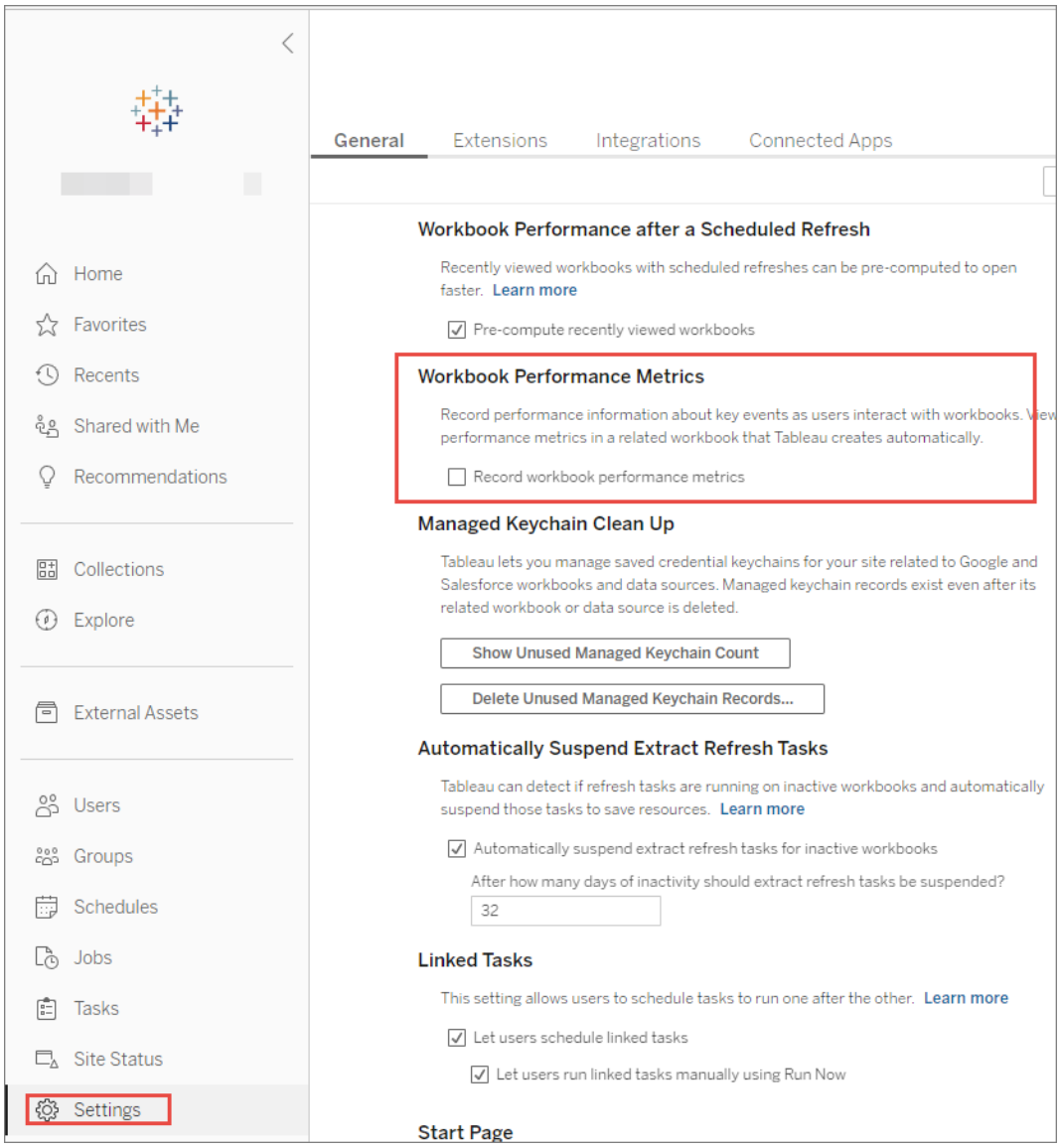
Tableau support may ask that you create a performance workbook as they work with you to diagnose performance issues.

> Looking for Tableau Server on Windows? See Create a Performance Recording.

Enable Performance Recording for a Site

By default, performance recording is not enabled for a site. A server administrator can enable performance recording site by site.

1. Navigate to the site for which you want to enable performance recording.
2. Click **Settings**:



3. Under Workbook Performance Metrics, select **Record workbook performance metrics**.

4. Click **Save**.

Start a Performance Recording for a View

1. Open the view for which you want to record performance.

   When you open a view, Tableau Server appends ":iid=<*n*>" after the URL. This is a session ID. For example:

   ```
   http://10.32.139.22/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?:iid=1
   ```

2. Type `:record_performance=yes&` at the end of the view URL, immediately before the session ID. For example:

   ```
   http://10.32.139.22/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?:record_performance=yes&:iid=1
   ```

3. Click the **Refresh** button in the toolbar.

4. Load the view.

View a Performance Recording

1. Click **Performance** to open a performance workbook. This is an up-to-the-minute snapshot of performance data. You can continue taking additional snapshots as you continue working with the view; the performance data is cumulative.

2. Move to a different page or remove `:record_performance=yes` from the URL to stop recording.

Interpret a Performance Recording

Create a recording to evaluate the performance of your workbook. After you have completed the recording, you can download the resulting workbook and open it in Tableau Desktop for analysis.

A performance recording workbook contains two main dashboards: Performance Summary and Detailed Views. The Performance Summary dashboard provides a high-level overview of the most time-consuming events. The Detailed Views dashboard provides a lot more detail

and is intended to be used by advanced users when building workbooks. The **Detailed Views** dashboard is only visible when the performance recording workbook is opened using Tableau Desktop.

For information on how to create a performance recording in Tableau Server, see Create a Performance Recording.

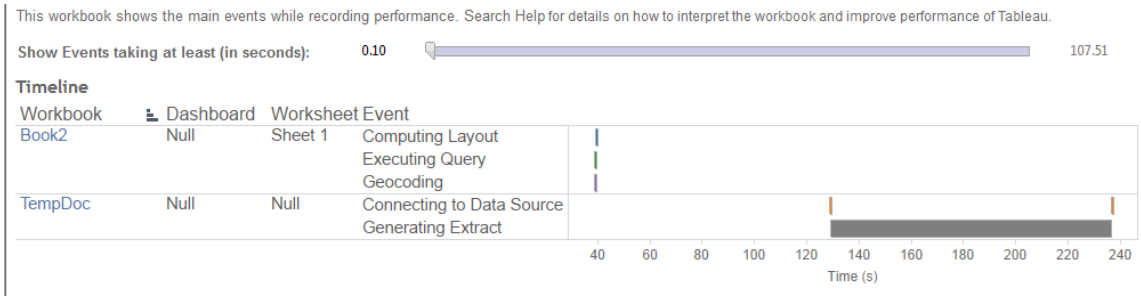
Performance Summary

The **Performance Summary** dashboard contains three views: **Timeline**, **Events**, and **Query**.

Timeline

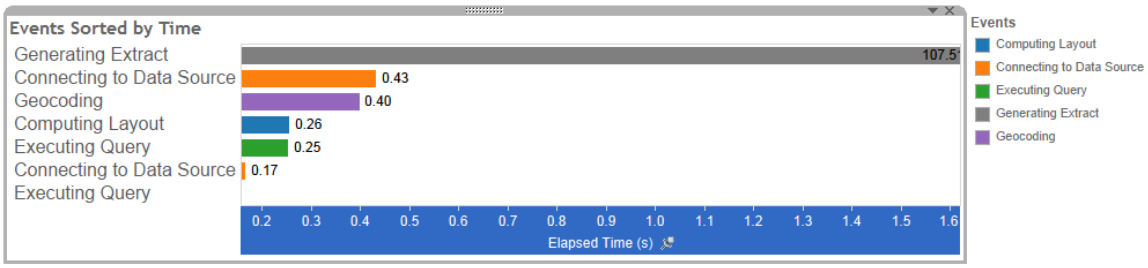
The uppermost view in the performance summary recording dashboard shows the events that occurred during recording, arranged chronologically from left to right. The bottom axis shows elapsed time since Tableau started, in seconds.

In the Timeline view, the **Workbook**, **Dashboard**, and **Worksheet** columns identify the context for events. The **Event** column identifies the nature of the event, and the final column shows each event's duration and how it compares chronologically to other recorded events:



Events

The middle view in a performance summary dashboard shows the events, sorted by duration (greatest to least). Events with longer durations can help you identify where to look first if you want to speed up your workbook.



Different colors indicate different types of events. The range of events that can be recorded is:

- Computing layouts

  If layouts are taking too long, consider simplifying your workbook.

- Connecting to data source

  Slow connections could be due to network issues or issues with the database server.

- Compiling query

  This event captures the amount of time spent by Tableau in generating the queries. Long compile query times indicate that the queries generated are complex. The complexity may be due to too many filters, complex calculations, or generally due to a complex workbook. Examples of complex calculations include, lengthy calculations, LOD calculations, or nested calculations. Try simplifying the workbook, using action filters or moving calculations to the underlying database.

- Executing query

  - For live connections, if queries are taking too long, it could be because the underlying data structure isn't optimized for Tableau. Consult your database server's documentation. As an alternative, consider using an extract to speed performance.

  - For extracts, if queries are taking too long, review your use of filters. If you have a

lot of filters, would a context filter make more sense? If you have a dashboard that uses filters, consider using action filters, which can help with performance.

- Generating extract

  To speed up extract generation, consider only importing some data from the original data source. For example, you can filter on specific data fields, or create a sample based on a specified number of rows or percentage of the data.

- Geocoding

  To speed up geocoding performance, try using less data or filtering out data.

- Blending data

  To speed up data blending, try using less data or filtering out data.

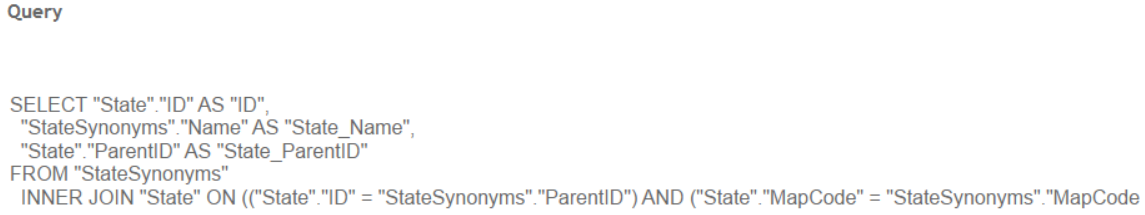
- Server rendering

  You can speed up server rendering by running additional VizQL Server processes on additional machines.

Query

If you click on an **Executing Query** event in either the **Timeline** or **Events** section of a performance summary dashboard , the text for that query is displayed in the Query section.

If you are connected to a published data source, the query text is displayed in XML. If you are connected to the data source directly, the query is displayed in SQL like shown below:



If it makes sense, you can use the query text to work with your database team on optimizing at the database level. Sometimes the query is truncated and you'll need to look in the Tableau

log to find the full query. Most database servers can give you advice about how to optimize a query by adding indexes or other techniques. See your database server documentation for details.

Sometimes for efficiency, Tableau combines multiple queries into a single query against the data. In this case, you may see an **Executing Query** event for the Null worksheet and zero queries being executed for your named worksheets.

Detailed Timeline

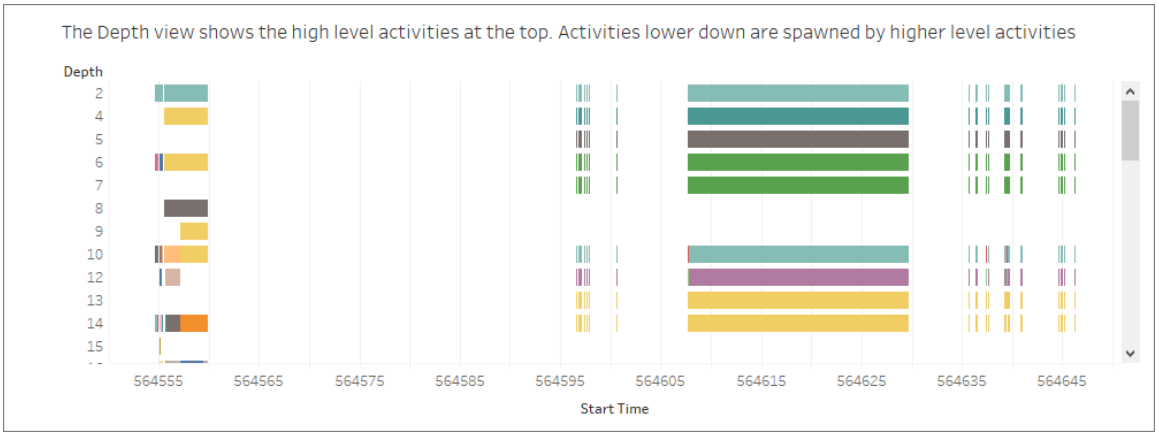
This view is the detailed version of the **Timeline** view that shows all events and separating individual items that were grouped in the **Timeline** view. It is intended to be used by advanced users during workbook designs.

Detailed Views

The **Detailed Views** dashboard contains **Depth**, **Exclusive CPU**, **Inclusive CPU**, and **Elapsed Time** views.

Depth

The **Depth** view is the uppermost view in the **Detailed Views** dashboard and provides insight into what happens when a request is made. This view is the most useful when filtered to a single user request. Examples of user requests are: loading a view, selecting a mark, or changing a filter.



Each bar on the depth view represents a single activity. An activity is a unit of work that is done as part of processing a user request. A single user request results in multiple activities. The length of each bar on the depth view is proportional to the elapsed time for the activity that the bar represents.

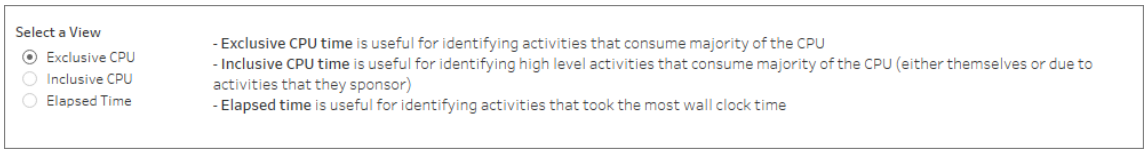
High level activities appear at the top of the view. Activities lower down are child activities generated by high level activities.

Hovering over each bar provides additional details about the activity and also highlights the corresponding row in the **CPU** and **Elapsed Time** view described in the next section.

In order to narrow down on what parts of the request took the most time, investigate long running activities at the highest levels.

CPU and Elapsed Time

The **CPU** and **Elapsed Time** views appear lowermost in the **Detailed Views** dashboard. You can toggle between **Exclusive CPU**, **Inclusive CPU** and **Elapsed Time** views by clicking on the radio buttons.



While the **Depth** view can help in quick visual identification of long running activities, it may not necessarily highlight activities that happen multiple times with each instance taking a small amount of time. The **Exclusive CPU**, **Inclusive CPU**, and the **Elapsed Time** views provide aggregate statistics for each activity. The number of times an activity took place is shown in the **Count** column and the total amount of time taken by a single activity is shown using the bar chart.

Sometimes for efficiency, Tableau combines multiple queries into a single query against the data. In this case, you may see an **Executing Query** event for the Null worksheet and zero queries being executed for your named worksheets.

Performance Monitoring Tools

This topic describes external resources that you can use to monitor and tune performance.

Tableau Server includes several tools that you can use to monitor server performance and health. For more information about these tools, see Performance Monitoring Overview.

Disclaimer: This topic includes information about third-party and community supported products. Please note that while we make every effort to keep references to third-party and community content accurate, the information we provide here might change without notice. For the most up-to-date information, please consult the documentation for products referenced here. To learn more about community supported tools, see Support levels for IT and developer tools.

- TabJolt. A load and performance testing tool that you can use to understand how Tableau Server scales with your workloads, in your environment, and to inform your scalability and capacity needs. Here are some key use cases for when you would use TabJolt:
    - To establish a baseline for server performance and test deployments before pushing them to production environments.
    - In a new Tableau Server, to help understand how the new server scales in your environment, specifically to your hardware and workload?
    - Before upgrading to understand the new version will scale in your environment.
    - To find the best server deployment configuration, given your hardware, workbooks and environments.
- Replayer. A tool that can replay log-based real user traffic from a Tableau Server against any other server or configuration. It replays Tableau Server single- or multi-user sessions. Here are some ways that Replay can be used:

    - Playback specific Tableau Server sessions, and filter the session based upon start time or RequestID.
    - Use it to simulate load conditions so that you can test how to scale and balance your Tableau Server installations.
    - Perform regression testing by running and comparing end-to-end user scenarios for Tableau Server upgrades.
    - Capture and report HTTP exceptions that occur in a single-user session.
    - Replay a defect, so that you can troubleshoot and verify that it is fixed.
- Scout. An exploratory tool that captures performance metrics across any workbooks on both Tableau Desktop and Tableau Server Here are some ways that Scout can be used:
    - Find slow workbooks on Server.
    - Validate performance improvements or regressions after making server configuration or topology changes.
    - Validate that workbooks and dashboards are loading properly after upgrading to new Tableau Server.
    - Validate that workbooks are still working properly after data source changes.
- Sitescope. An agentless application monitoring tool.
- Zabbix. An open-source, real-time monitoring tool.
- Splunk. A tool for monitoring and analyzing machine data, including logs.
- Graylog. An open-source log management tool.

Configure Client-Side Rendering

When you navigate to a view in Tableau Server, the processing required to display the view, called *rendering*, can be performed by either your client device or Tableau Server. The choice depends on the complexity of the view, which is determined by the number of marks, rows, columns, and more. If a view is less complex, it's faster for a client device to render the view. If a view is more complex, it's faster to send a request to Tableau Server and take advantage of the server's greater computing power.

Note: If a view uses the polygon mark type or the page history feature, server-side rendering is always performed, even if client-side rendering is enabled.

Supported browsers

Client-side rendering is supported in Internet Explorer version 9.0 or higher, Firefox, Chrome, and Safari. All of these web browsers include the HTML 5 `<canvas>` element, which client-side rendering requires.

Client-side rendering is also supported by the Tableau Mobile app.

Configure the complexity threshold for computers and mobile devices

Because computers have more processing power than mobile devices, Tableau Server performs more client-side rendering on computers than on mobile devices.

As a server administrator, you can configure when client-side rendering happens on computers and mobile devices by adjusting the complexity threshold for each. For example, you might lower the threshold for mobile devices if you notice that views display slowly on them. Or, you might increase the threshold to reduce the number of requests to Tableau Server.

By default, the complexity threshold for computer web browsers is 100. To adjust the complexity threshold for computers, use the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold -v
[new value]
```

By default, the complexity threshold for mobile devices is 60. To adjust the complexity threshold for mobile devices, use the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold_
mobile -v [new value]
```

For example, to change the mobile threshold to 40, you might enter the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold_
mobile -v 40
```

For more information on how to use tsm option set, see tsm configuration set Options.

Disable client-side rendering

Client-side rendering is enabled by default and is recommended to improve the performance of views. However, you might want to disable client-side rendering temporarily for testing or if your server is being accessed primarily by computers or mobile devices with very little processing power.

Use the following command to disable client-side rendering:

```
tsm configuration set -k vizqlserver.browser.render -v false
```

For more information on how to use tsm option set, see tsm configuration set Options.

Testing with the URL Parameter

To test server-side rendering on a session basis, type `?:render=false` at the end of the view's URL. For example:

```
http://localhost/views/Supplies/MyView?:render=false
```

If client-side rendering is disabled on Tableau Server, enter `?:render=true` to enable it for the session:

```
http://localhost/views/Supplies/MyView?:render=true
```

You can also test particular complexity thresholds on individual views to see if it’s appropriate to adjust the server-wide threshold for your server and network conditions. For example, you may find that lower complexity (such as 80) or higher complexity (such as 120) tipping points result in more responsiveness to user interactions. To test a threshold, you can keep the server’s default configuration (client-side-rendering enabled) and enter the test threshold number at the end of the view's URL. For example:

```
http://localhost/views/Supplies/MyView?:render=80
```

View Acceleration

Administrators and workbook owners who have Creator or Explorer licenses can accelerate workbooks. Administrators can suspend individual views or turn off acceleration for their site. View Acceleration loads views faster by precomputing and fetching the workbook's data in a background process. There are two potential bottlenecks when loading a view:

1. Querying (fetching data from the data source).
2. Rendering (creating the visuals, such as drawing shapes or rendering a map).

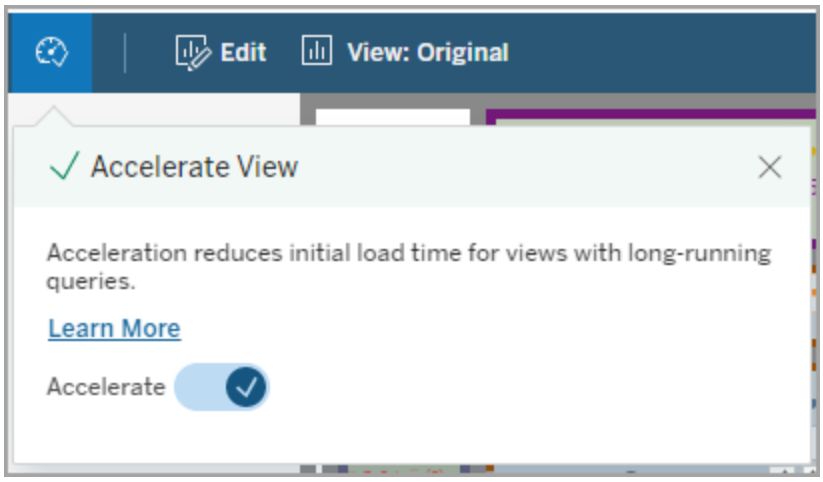
The time it takes to load a workbook depends on the combined time it takes to do these two steps. However, not all views can be accelerated. View Acceleration improves the performance of the first step (querying). If the view is loading slowly for reasons other than querying, View Acceleration won't improve the workbook's performance.

When users create custom views on top of an accelerated view, the ten most used custom views are precomputed automatically. These accelerated custom views don’t count against the view limit. Custom views that haven’t been accessed in the last 14 days won’t be accelerated. If you directly accelerate a custom view, both the original view and the custom view are accelerated.

Accelerate your view

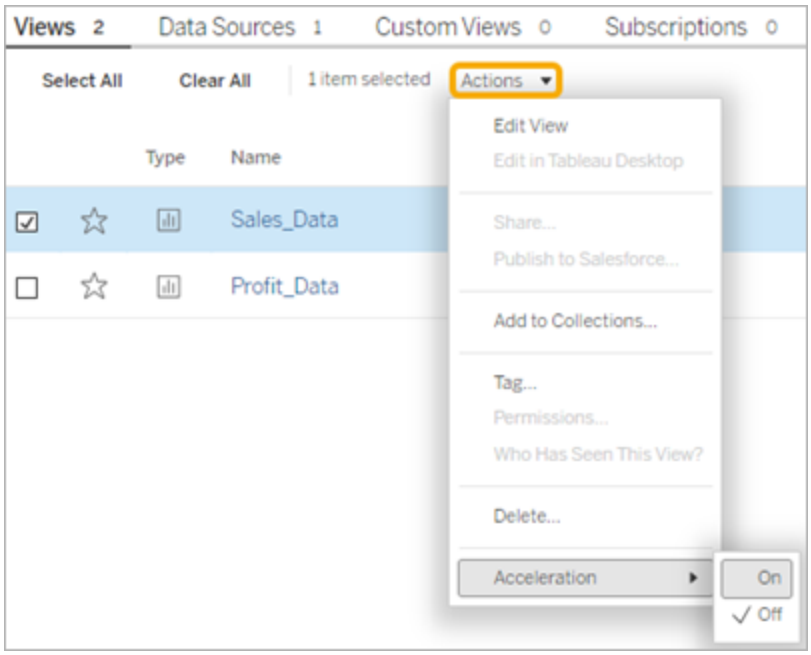
1. Sign in to a site on Tableau Cloud or Tableau Server.
2. From the Home or Explore page, navigate to the view you want to accelerate.

3. Choose the **Accelerate** icon, and select the switch to **Accelerate**.



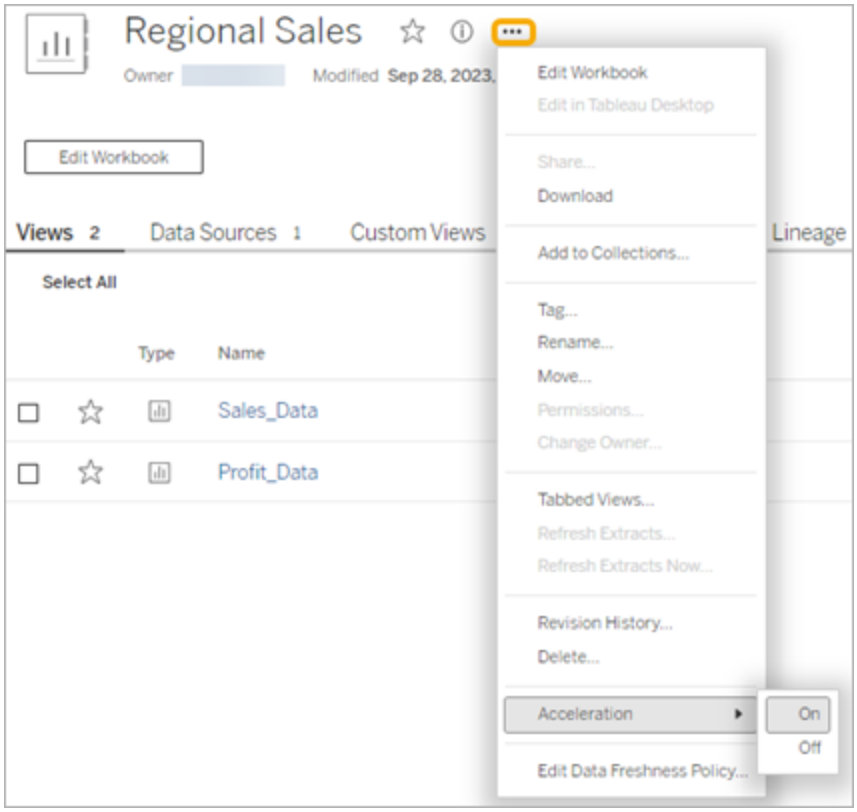
You can also accelerate views from the workbook page in one of three ways:

1. Select the desired view and choose **Acceleration** > **On** from the **Actions** menu.



2. Select the **More options**(...) menu for the desired view and choose **Acceleration** > **On**.

3. To accelerate all views in the workbook, choose **Acceleration** > **On** from the **More options (...)** menu.



Note: View Acceleration isn't available in Tableau Desktop.

Understand why View Acceleration is unavailable, suspended, or ineffective

View Acceleration is unavailable

There are a few scenarios in which acceleration isn't available for a view.

Issue	Resolution
The view doesn't have historical data for the time required to execute queries.	Every time a view loads, it takes a few minutes for the viz data to be available. For a newly created view, load the view and wait a few minutes before trying to accelerate it.
The historical time taken to execute queries for the view is less than 2 seconds.	Acceleration isn't supported for such views because acceleration won't significantly improve the performance of the view.
The view doesn't have embedded credentials.	To precompute the data, Tableau needs to automatically connect to the data source in the background without any user interaction. As a result, View Acceleration is only supported for workbooks with embedded connection credentials.
The view has user-based functions, or the view has a data source with user-based functions.	Currently, Tableau doesn't support accelerating such views. Examples of user-based functions are USERDOMAIN() and USERNAME().
The view's owner is inactive.	To precompute the data, the owner must be an active user. Tableau doesn't support accelerating a view if its owner is inactive. Change the ownership to an active user.
The view's data freshness policy is less than 2 hours.	Cost can be high for accelerating views that are refreshed so frequently, and Tableau doesn't want to overload your site performance. For more information, see Set a Data Freshness Policy.
The site has reached	Update site settings to increase the maximum number of views

the limit for the number of views that can be accelerated.	that can be accelerated, or choose No limit.

View Acceleration is suspended

There are a few scenarios in which acceleration is suspended.

Issue	Resolution
The view's acceleration was suspended.	Re-enable acceleration for the view.
There are background acceleration jobs running to precompute the data for the view.	If the jobs fail regularly, the view is auto suspended. The jobs may fail if: • A view's credentials have expired. Update the view's credentials. • The owner of the view becomes inactive. Change the ownership to an active user. • The view's data source was deleted. Contact the view's owner to update the data source. • The job to precompute the data takes too long and times out. View Acceleration has a maximum runtime of 30 minutes. Contact the view's owner to optimize the workbook.

View Acceleration is ineffective

View Acceleration reduces the time it takes to execute queries of a view. If the time taken to execute queries isn't the bottleneck for the viz load time, you won't notice a significant improvement in performance during a viz load. Likewise, a view usually has many queries. You can't accelerate queries with transient functions, such as now() or using relative date filters. If a view has a long-running query with transient functions, you won't notice an improvement in performance during a viz load.

Refresh accelerated views

Event based refresh of accelerated views

In workbooks that have an extract, all accelerated views are refreshed when the extract refresh completes. When a workbook is republished or renamed, all accelerated views in the workbook are refreshed.

Schedule based refresh of accelerated views

Schedules for refreshing accelerated views can only be configured if a workbook has at least one live data source.

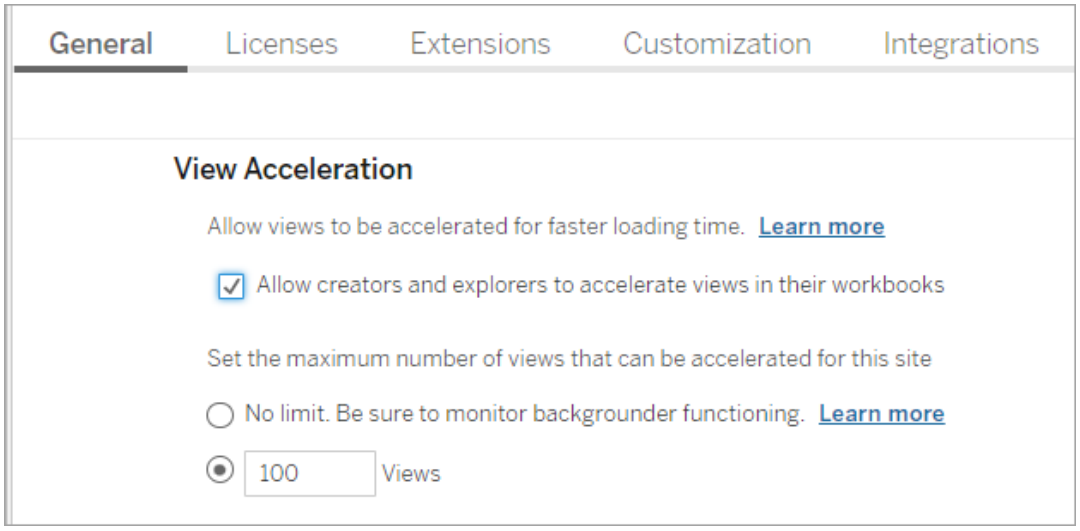
Accelerated views for workbooks utilizing live data sources are refreshed based on the workbook's data freshness policies. If there isn't a schedule set for refreshing the data, the default data freshness policy is used. For more information, see Edit a workbook data freshness policy.

Manage View Acceleration on your site

By default, View Acceleration is allowed.

1. Sign in to your site on Tableau Server.
2. From the left pane, choose **Settings**.
3. From the **General** tab, scroll to the **View Acceleration** section.
4. Select the check box to allow creators and explorers to accelerate views in their workbooks. Clear the check box to turn off View Acceleration for the site.

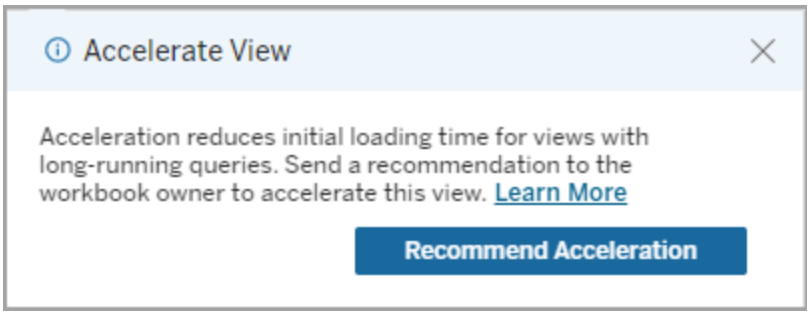
5. Enter the maximum number of views that can be accelerated for your site, or choose **No limit**.



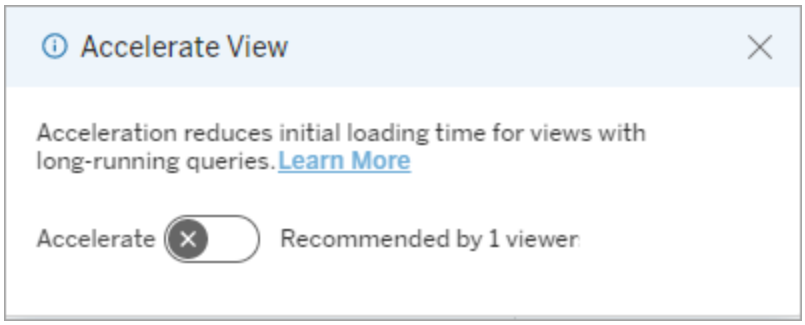
Accelerate recommended views

Based on a workbook's query time and usage, Tableau sometimes recommends acceleration to improve the performance of slower and more popular views and dashboards. If acceleration is available for a view, users can recommend acceleration for a view once every 30 days.

When a user visits a view, they'll see the option to Recommend Acceleration to the site admin or workbook owner.



When the site admin or workbook owner visits the same view, they'll see the option to accelerate the view, and they'll see how many users have recommended acceleration.



Manage Views recommended for acceleration

As a site admin, you can see when Tableau has recommended acceleration for a view:

1. Sign in to your Tableau site.
2. From the left pane, choose **Tasks**.
3. From the **Acceleration status** column, check for views with a **Recommended** status. You can also use the **Filter** in the right-side pane to filter for views with a **Recommended** status.

Personalized recommendations for acceleration as a workbook owner or admin:

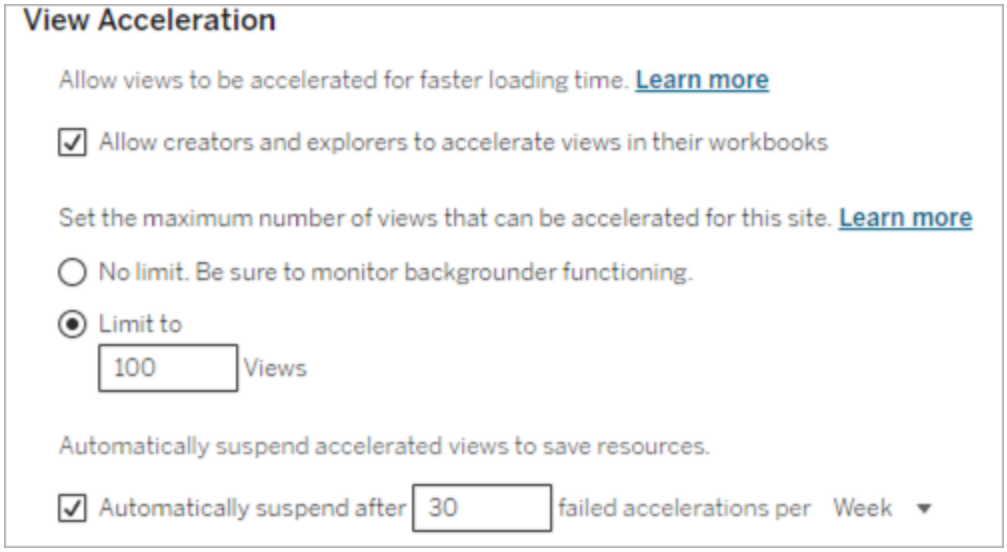
1. Sign in to your Tableau site.
2. From the top-right of the page, choose the account menu icon.
3. Select **My Content**.
4. Select the **Performance** tab.
5. From the **Actions** column, choose **Accelerate**.

Automatically suspend acceleration to save resources

To conserve resources, administrators can automatically suspend acceleration for views that are consistently failing. Administrators can set a threshold for the number of times an acceleration task can fail per day, week, or month before the acceleration is automatically suspended.

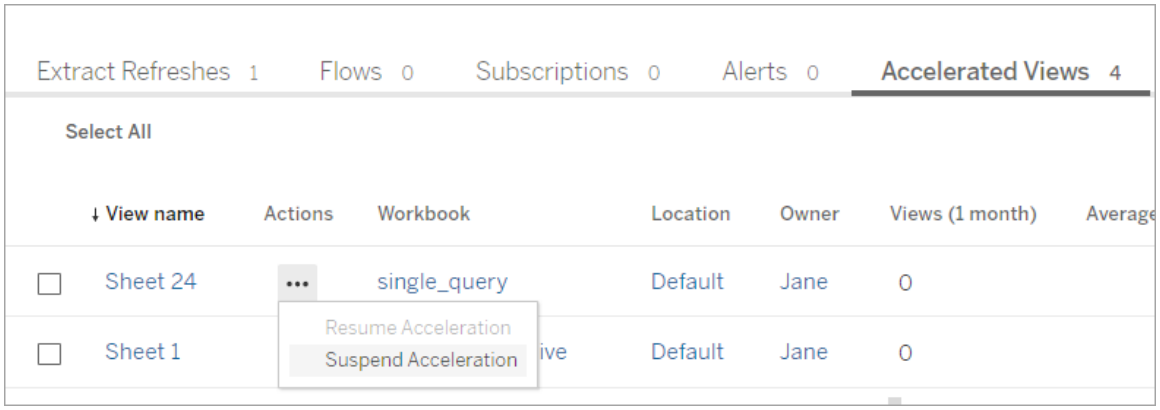
1. Sign in to your Tableau site.
2. From the left pane, choose **Settings**.
3. From the **General** tab, scroll to the **View Acceleration** section.
4. Set the maximum number of failures allowed per day, week, or month.

5. Choose **Save**.



View and manage accelerated workbooks

1. Sign in to your Tableau site.
2. From the left pane, choose **Tasks**.
3. Select the **Accelerated Views** tab.
4. Choose the **Actions** menu (...) to Resume or Suspend Acceleration for the selected view or views.



Manage View Acceleration notifications

Administrators can manage whether to receive notifications for views that are automatically suspended.

1. Sign in to your Tableau site.
2. From the left pane, choose **Settings**.
3. From the **General** tab, scroll to the **Manage Notifications** section.
4. To receive notifications for views that are automatically suspended, check the box for **View Acceleration**.
5. Choose **Save**.

When views are automatically suspended, notifications are sent to site and server administrators. The notification includes information about why the view was suspended and the time that the view was suspended. Select the notification to go to the **Accelerated Views** tab of the **Tasks** page. From this page, administrators can filter the Acceleration Status to find views that were automatically suspended.

Understand user context for precomputation

Precomputation for accelerated workbooks is performed with the user context of only one user. This user is either:

- The owner of the workbook (if there are no user filters in the workbook or data source, or if there are user filters on the data source but the data source is a published data source).
  -or-
- The user that was selected for thumbnail generation the last time the workbook was published (if there are user filters on the workbook and the data source isn't a published data source).

Understand the cost of View Acceleration

Enabling this feature increases the computation load and number of jobs on Tableau Server backgrounder processes because View Acceleration fetches the required data from data sources in a background process. A background job to precompute the data of an enabled workbook is run if any of the following happen:

- The workbook and published data source are republished (this includes the web-authoring save).
- An extract used by the workbook is refreshed.

Administrators should consider those costs before enabling View Acceleration for many workbooks, or scheduling acceleration jobs too frequently.

- Workbooks that are being heavily edited and republished might not be suitable because each republish triggers a precomputation. We recommend acceleration for workbooks that are published for consumption.
- If a workbook uses multiple extracts, then their refresh triggers precomputation of the data. Thus, frequent extract refreshes for enabled workbooks could cause a spike in backgrounder job load, especially given that, by design, View Acceleration jobs are run after the successful extract refresh.
- The precomputed data for workbooks is stored as materialized views in Hyper.

Extract Query Load Balancing

In Tableau Server version 2020.2 and later, load balancing for extract-based queries has improved and may result in faster load times for extract-based dashboards. Large deployments that are extract heavy and have a high volume of dashboard loads will probably see the most improvement, particularly if they have Hyper running on standalone nodes. Smaller deployments where Hyper is running on nodes that are shared with other server processes may also see performance improvements.

Hyper logs a server health metric about the amount of resources Hyper is consuming and also takes into account load from other Tableau processes that may be running on the same server node. Based on this information, extract queries will be sent to a node that has available resources to process the query. In addition to evaluating system resources, the load balancer improves the chance that the node you are routed to has the extract already cached. This reduces the number of duplicate extracts across nodes and improves memory and I/O usage.

To use this feature, the Cache Server process must be active. For more information, see Tableau Server Cache Server. If Cache Server process is not enabled, load balancing will automatically revert to the previous functionality.

This feature is turned on by default. To disable it, use the following tsm commands:

```
tsm configuration set -k hyper_standalone.consistent_hashing.enabled
-v false
```

```
tsm configuration set -k hyper_standalone.health.enabled -v false
```

Apply the changes using the following tsm command: `tsm pending-changes apply`

For more information, see tsm configuration set Options.

Monitoring Tableau Server

You can configure SMTP and alerts and subscriptions to aid in monitoring Tableau Server.

Configure SMTP Setup

Tableau Server can email server administrators about system failures, and email server users about subscribed views and data-driven alerts. First, however, you need to configure the SMTP server that Tableau Server uses to send email. After configuring SMTP, complete the steps to configure notifications (Configure Server Event Notification), then when you start or restart the server, it will trigger an email notification, which confirms that you have set up notifications correctly.

Configuring SMTP requires that you restart Tableau Server services.

Secure SMTP

To enable and configure TLS for SMTP, you must use the TSM CLI as described in this topic. Tableau Server only supports STARTTLS (Opportunistic or Explicit TLS).

If your organization does not use public certificates for verifying TLS connections, then you can upload a private certificate to Tableau Server to verify trusted connections. For more information, see the tsm security custom-cert add command.

You may also configure SMTP TLS for encryption-only by disabling the certificate validation process. For more information, see the section, *Configuration file reference*,in the *Use the TSM CLI* tab below.

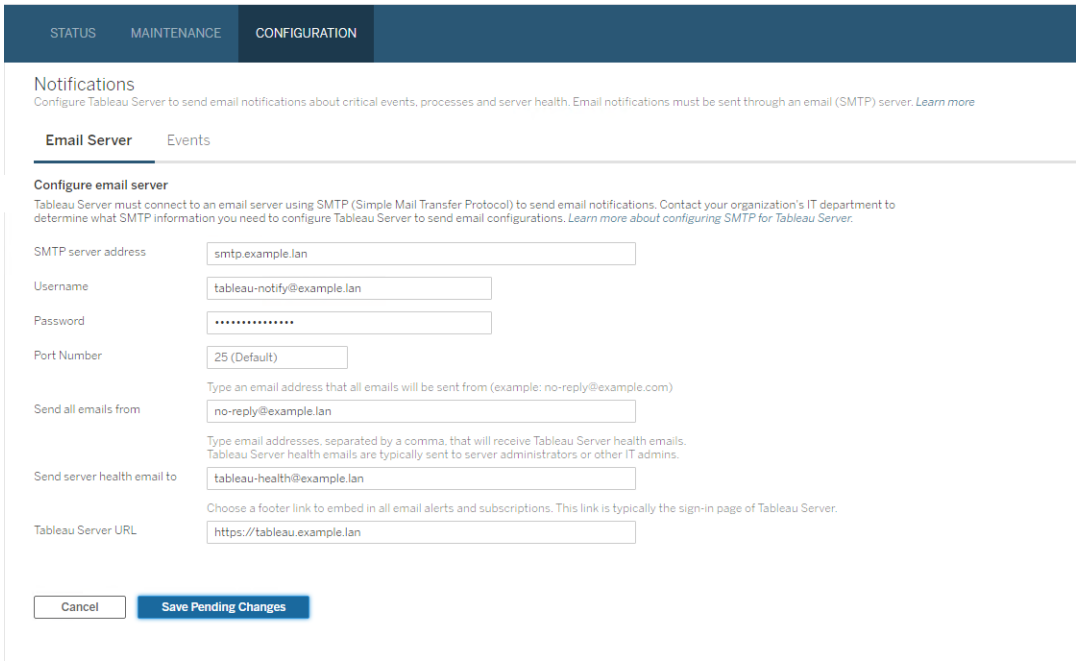
Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click **Notifications**on the **Configuration** tab and click **Email Server**.

3. Enter the SMTP configuration information for your organization:



4. Click **Save Pending Changes** after you've entered your configuration information.

5. Click **Pending Changes** at the top of the page:

6. Click **Apply Changes and Restart**.

7. Run the `tsm email test-smtp-connection` to view and verify the connection configuration. See tsm email test-smtp-connection.

Use the TSM CLI

For the initial configuration of SMTP, we recommend that you use the configuration file template below to create a json file. You can also set any single configuration key listed below with the syntax described in tsm configuration set.

1. Copy the following json template to a file.

   Important: The template below includes common options for most deployments. After you copy the template to a text file, you must edit the option values for your SMTP server requirements. You may need to remove or add options. See the reference section that follows for more information about all supported SMTP key options.

```
{
“configKeys”: {
       "svcmonitor.notification.smtp.server": "SMTP server host
name",
       "svcmonitor.notification.smtp.send_account": "SMTP user name",
       "svcmonitor.notification.smtp.port": 443,
       "svcmonitor.notification.smtp.password": "SMTP user account
password",
       "svcmonitor.notification.smtp.ssl_enabled": true,
       "svcmonitor.notification.smtp.from_address": "From email
address",
       "svcmonitor.notification.smtp.target_addresses": "To email
address1,address2",
       "svcmonitor.notification.smtp.canonical_url": "Tableau Server
URL"
       }
}
```

2. Run the `tsm settings import -f file.json` to pass the json file with the appropriate values to Tableau Services Manager to configure Tableau Server for SMTP.

Tableau Services Manager will validate the entity values.

3. Run the `tsm pending-changes apply` command to apply the changes. See tsm pending-changes apply.

4. Run the `tsm email test-smtp-connection` to view and verify the connection configuration. See tsm email test-smtp-connection.

SMTP CLI configuration reference

This table lists all of the options that can be used to configure SMTP with TSM CLI.

Option	Description
`svc-mon-itor.notification.smtp.server`	Address of SMTP server. Example: `"svc-monitor.notification.smtp.server": "mail.example.com"`
`svc-mon-itor.notification.smtp.send_account`	User name for SMTP account.
`svc-mon-itor.notification.smtp.port`	Port number for SMTP server. The default is 25.
`svc-mon-itor.notification.smtp.password`	Password for SMTP server account. Example: `"svc-mon-`

Option	Description
	`itor.no-tification.smtp.password":"password"`
`svc-mon-itor.notification.smtp.ssl_enabled`	Specifies whether the connection to the SMTP server is encrypted. The default is false.
`svc-mon-itor.notification.smtp.ssl_required`	If enabled, Tableau Server will refuse to connect to SMTP servers without using TLS. The `svc-monitor.notification.smtp.ssl_enabled` option must also be set to true. The default is false.
`svc-mon-itor.notification.smtp.ssl_check_server_identity`	If set to true, Tableau Server will check the SMTP server identity as specified by RFC 2595. These additional checks based on the content of the server's certificate are intended to prevent man-in-the-middle attacks. The default is false.
`svc-mon-itor.notification.smtp.ssl_trust_all_hosts`	When using TLS, trust certificates from all mail servers, ignoring the validity of the certificate's chain of trust. By setting this key to true, TLS will be used only to encrypt the traffic to the SMTP host. The default is false.
`svc-mon-itor.notification.smtp.ssl_`	The default and supported sets of cipher suites is defined by the version of JDK that is installed with

Option	Description
`ciphers`	Tableau Server. See the section below, TLS ciphers, for a list of supported and default ciphers. To update the cipher suites used by Tableau Server for SMTP TLS connections, enter a white space-separated list of cipher suites for this value. For example, "`TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384`".
`svc-monitor.notification.smtp.ssl_versions`	The default TLS versions enabled on this version of Tableau Server are TLSv1, TLSv1.1, TLSv1.2 and TLSv1.3. TLS version support is defined by the version of JDK that is installed with Tableau Server. Supported versions of TLS are `SSLv2Hello, SSLv3, TLSv1, TLSv1.1, TLSv1.2, TLSv1.3`. To update the versions used by Tableau Server for SMTP TLS connections, enter a white space-separated list of versions for this value. For example, "`TLSv1.2 TLSv1.3`".
`svc-monitor.notification.smtp.from_address`	Email address that will send an notification if there's a system failure. The email address must have valid syntax (for example, ITalerts@bigco.-com or noreply@mycompany), but it does not have to be an actual email account on Tableau Server. (Some SMTP servers may require an

Option	Description
	actual email account, however.) **Note**: You can override the system-wide email address on a per-site basis. For more information, see What is a site. Example: `"svcmonitor.notification.smtp.from_address": "donot-reply@example.com"`
`svc-monitor.notification.smtp.target_addresses`	Email address to receive notifications. If email notifications are enabled, you need to include at least one address. Separate multiple addresses with commas. Example: `"svc-monitor.notification.smtp.target_addresses": "iluvdata@example.com"`
`svc-monitor.notification.smtp.canonical_url`	URL of the Tableau Server. Enter `http://` or `https://`, followed by the name or IP address of the Tableau server. Used in the footer of subscription email. Example: `"svc-monitor.notification.smtp.canonical_url": "http://myserver.example.com"`

TLS ciphers

The following is a list of TLS ciphers that are supported by the JDK that is included with Tableau Server. In this version of Tableau Server, all of these ciphers are enabled by default. You can specify a custom list of ciphers for your SMTP configuration by entering a white-space separated list with the option, `svcmonitor.notification.smtp.ssl_ciphers`, as described in the table above.

TLS_RSA_WITH_AES_128_CBC_SHA256	TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384	TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA	TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256	TLS_DHE_DSS_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS_EMPTY_RENEGOTIATION_INFO_SCSV
TLS_ECDH_ECDSA_WITH_AES_256_	TLS_ECDHE_ECDSA_WITH_AES_256_

CBC_SHA384	CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256	TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA	TLS_DHE_DSS_WITH_AES_128_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_RSA_WITH_AES_128_CBC_SHA	TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	TLS_AES_256_GCM_SHA384

TLS_AES_128_GCM_SHA256	

Configure Server Event Notification

A Tableau Services Manager (TSM) administrator can configure Tableau Server to allow notifications for the following events:

- Content updates
  - Extract failures (enabled by default)
  - Subscription views for users (disabled by default)
- Server health monitoring
  - Server status changes (disabled by default)
  - Desktop License reporting (disabled by default)
- Drive space
  - Email alerts when disk space crosses or remains below pre-configured thresholds (disabled by default)
  - Recording usage history (enabled by default)

Note: You need to configure SMTP before you can configure subscriptions or notifications. For more information, see Configure SMTP Setup.

Use the TSM web interface

1. Open TSM in a browser:

   `https://<tsm-computer-name>:8850`.

   For more information, see Sign in to Tableau Services Manager Web UI.

2. Click **Notifications**on the **Configuration** tab and click **Events**.

3. Configure notification settings for your organization:

- Content updates
    - **Send emails for extract refresh failures**

      When this option is enabled (the default), a server administrator can configure email notifications to be sent when extract refreshes fail. These messages are configured at the site level, so even if this option is enabled, messages are not sent unless the **Send email to data source and workbook owners when scheduled refreshes fail** option is enabled for a site (this is enabled by default). For details, see Enable Extract Refresh Scheduling and Failure Notification.

    - **Allow users to receive email for views that they have subscribed to**

      When this option is enabled (by default is it disabled), a server administrator can configure a site to send subscription email. These email messages are configured at the site level and can only be configured when this option is enabled. For details, see Set Up a Site for Subscriptions.

      When users subscribe to a workbook or view, a snapshot of the view is emailed to them on a scheduled basis, so they can see the latest updates without having to sign into Tableau Server.

      To allow users to attach PDF renderings on subscription emails, select **Let users add attachments to subscribed views**.
- Server health monitoring
    - **Send emails for Tableau Server process events (up, down, and failover)**

      Tableau Server sends an email message when the data engine, file store, gateway, or repository server processes stop or restart, or when the initial Tableau Server node stops or restarts.

If you are running a single-server installation (all processes on the same computer), health alerts are only sent when Tableau Server is up. No "down" alerts are sent. If you are running a distributed installation that's configured for failover, a DOWN alert means that the active repository or a data engine instance has failed and the subsequent UP alert means that the passive instance (repository) or second instance (data engine) of that process has taken over.

> **Note:** Tableau Server is designed to be self-correcting. If a service or process stops responding or goes down, Tableau Server attempts to restart it. This can take 15 to 30 minutes to complete. Because of this, reacting immediately to service or process alerts can be counter-productive, especially in an installation with redundant services that can handle requests while one restarts.

- **Enable Tableau Desktop License reporting**

  License reporting data originates in Tableau Desktop and is sent to Tableau Server. When this option is enabled, Tableau Server will generate and display the administrative report for Desktop License reporting. For information on the report, see Desktop License Usage.

- Drive space

  Enable notifications (alerts) for remaining disk space on your Tableau Server.

  - **Send emails when unused drive space drops below thresholds**

    You can configure Tableau Server to send email notifications when disk space usage on any node crosses a threshold, or remains

below the threshold. And you can configure how often threshold notifications are sent.

There are two thresholds you must set, **Warning threshold** and **Critical threshold**. Thresholds are expressed in percentage of disk space remaining. The critical threshold must be less than the warning threshold.

You also specify the **Send threshold alert every** option. This determines how often, in minutes, warning and critical notifications should be sent. The default value is 60 minutes.

- **Record disk space usage information and threshold violations for use in custom administrative views**

  When you configure Tableau Server to record disk space usage, information about free disk space is saved in the repository and you can view the usage history using the Administrative Views.

4. Click **Save Pending Changes** after you've entered your configuration information.
5. Click **Pending Changes** at the top of the page:

6. Click **Apply Changes and Restart**.

Use the TSM CLI

The various notification values described above can be set individually with the tsm configuration set command. Alternatively, you can construct a json file and pass all configuration values in one operation. Both methods are described in this section.

Set notification values individually

The following table shows the key/value pairs that map to the notification events described earlier in this topic. Use the tsm configuration set command with the following syntax to set a single key/value pair:

```
tsm configuration set -k <config.key> -v <config_value>
```

For example, to enable job failure notifications, run the following command:

```
tsm configuration set -k backgrounder.notifications_enabled -v true
```

Notification option	**Key**	**Value**
Extract failures or Flow run failures	`backgrounder.notifications_enabled`	`true` | `false`
Enable subscription views for user	`subscriptions.enabled`	`true` | `false`
Enable PDF attachments for subscriptions	`subscriptions.attachments_enabled`	`true` | `false`
Maximum attachment size (MB) for subscription notifications	`subscriptions.max_attachment_size_megabytes`	integer value, default is `150`
Server status changes	`svcmonitor.notification.smtp.enabled`	`true` | `false`
License reporting	`features.DesktopReporting`	`true` | `false`
Remaining	`storage.monitoring.email_enabled`	`true` | `false`

space thresholds: enable email notifications		
Remaining space thresholds: warning percentage	`storage.monitoring.warning_percent`	integer value, for example, `20`
Remaining space thresholds: critical percentage	`storage.monitoring.critical_percent`	integer value, for example, `15`
Set email interval	`storage.monitoring.email_interval_min`	integer value, in minutes, for example, `25`
Record usage history	`storage.monitoring.record_history_enabled`	`true` | `false`

After you are done setting values, you must run the following command:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Set all notification values with a single json file

To make all notifications settings with a single configuration, you can pass a json file.

Copy and edit the following template to create a file for your configuration.

```
{
  "configKeys": {
    "backgrounder.notifications_enabled": true,
    "subscriptions.enabled": true,
    "subscriptions.attachments_enabled": true,
    "subscriptions.max_attachment_size_megabytes": 150,
    "svcmonitor.notification.smtp.enabled": true,
    "features.DesktopReporting": true,
    "storage.monitoring.email_enabled": true,
    "storage.monitoring.warning_percent": 20,
    "storage.monitoring.critical_percent": 15,
    "storage.monitoring.email_interval_min": 25,
    "storage.monitoring.record_history_enabled": true
  }
}
```

After you have saved the file, pass it with the following command:

```
tsm settings import -f <path-to-file.json>
```

To apply changes, run the following command:

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Maintenance

You should perform regular maintenance on Tableau Server, such as creating backups, synchronizing Active Directory groups, and removing unneeded files.

Backup and Restore

A Tableau Server administrator should perform regular database maintenance, monitor disk usage on the server, clean up unnecessary files to free up space on the server, and back up Tableau Server and its data. Taking these steps can help ensure that Tableau Server runs with maximum efficiency.

You can use the Tableau Services Manager (TSM) command line tool to back up and restore your Tableau data. Tableau data includes data extract files, as well as Tableau Server's own PostgreSQL database, which stores workbook and user metadata, and server configuration data. Tableau Server log files capture activity and can help you diagnose problems. Logs are written to folders on the server and you can archive and remove them to save disk space.

Note: You can use the `tsm maintenance restore` command to restore Tableau Server backups created using `tabadmin backup` and `tsm maintenance backup`. Database backups made in other ways, and virtual machine snapshots are not valid sources for restoring Tableau Server.

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

Platform compatibility

You can use a backup created with Tableau Server on Linux to restore Tableau Server on Windows version 2018.2 and later.

You cannot use a backup created with Tableau Server on Linux to restore earlier versions of Tableau Server on Windows (version 2018.1 and earlier).

You can use a backup created with Tableau Server on Windows (version 2018.2 and earlier) to restore Tableau Server on Linux. For more information about restoring a Windows backup on Tableau Server on Linux, see Migrate Tableau Server from Windows to Linux.

Disk Space Usage for Backup and Restore

The free disk space required to create a backup varies depending on the amount of data in the Tableau Server repository and file store services, and their collocation with the tabadmincontroller service. During backups, the background tasks for cleaning up old extracts are temporarily paused. This means that, for the duration of the backup, extract refreshes will leave extra files in place, adding to disk space usage. If your backup takes a long time, or if your organization uses many extracts that are regularly updated, this can result in a significant amount of temporary disk space usage. These temporary files will be removed after the backup is complete.

The following table lists the disk space requirements for backup based on whether the node hosts the repository, file store, controller, or some combination of them. In multi-node Tableau Server environments you need to estimate the required disk space on each node.

Repository	File Store	Controller	Disk Space Required
✓			3x repository data + 250 MB To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base` directory. To obtain the exact size of the repository data, open the backup file and use the size of the workgroup.pg_dump file.

	✓		1.5x file store data To obtain an estimate of file store data (extracts, flows, etc.), check the size of `<data dir-ectory>/dataengine` directory.
		✓	3x repository data + 250 MB + 2.5x file store data
✓	✓		3x repository data + 250 MB + 1.5x file store data
	✓	✓	3x repository data + 250 MB + 1.5x file store data
✓		✓	3x repository data + 250 MB + 2.5x file store data
✓	✓	✓	3x repository data + 250 MB + 1.5x file store data

Restore disk space requirements

You must have adequate disk space for the database restore process to run successfully.

To restore Tableau Server:

- On controller nodes, you need free space equal to at least the size of the backup archive.
- On repository nodes, you need free space equal to at least three times the size of the repository data in the backup archive, plus 250 MB, plus the size of the pgsql data dir-ectory.
- On file store nodes, you need free space equal to at least twice the size of the datae-ngine folder in the backup archive.

Best Practices for Backing Up Tableau Server

We recommend following these security and performance best practices.

Protect backup file

While configuration secrets are encrypted when stored on disk internally, when these configurations are exported to a backup file, some secrets are written into the file in plain text. It is up to the administrator to take measures to protect the backup file. There are a variety of options available:

- Write the file to an encrypted file system.
- Write the file to a disk that is physically protected and restricted to specific users.
- Encrypt the backup file.

Maximize backup efficiency

There are several ways you can maximize backup efficiency. Your environment can impact how effective each of these is, so test with your data to see what works best.

Optimizing with topology configurations:

- Co-locating File Store on the same node as the Administration Controller can reduce the length of time it takes to back up Tableau Server by reducing or eliminating the need to transfer data between nodes during the backup process. This is especially true if your organization uses many extracts.
- Co-locating the repository (pgsql) with the Administration Controller node can also help to reduce back up time, but the time savings is less significant than that of the File Store.

The Administration Controller is usually on the initial node, unless you have had an initial node failure and moved the controller to another node.

Optimizing with backup strategies:

Backup is a resource intensive process. If possible, doing your backups during off peak hours is a generally a good strategy. But this however, depends on your requirements and how often Tableau Server data is updated and what your restore requirements. For a detailed explan-

ation of backup and disaster recovery, see Tableau Server Disaster Recovery. Here are some backup strategies and adopt them to your requirements

- **Type of storage:** Solid State disks are recommended in general for backups. SSD helps make your backups faster and complete sooner compared to traditional spinning disks.
- **Backup compression:** You have the option of running your backups with or without compression. When you do your backup with compressions, your backup size will be comparatively smaller, but you may see a slower performance. So if your goal is more focused on speed, choose the `--skip-compression` option:

  Use the `--skip-compression` option when backing up Tableau Server. This creates the backup without using compression, and results in a larger backup file but can reduce the amount of time it takes for the backup to complete. For more information, see tsm maintenance backup.
- **Snapshot backup:** This option is only available if you have configured your Tableau Server with External File Store. Although the performance of snapshot backups depend on the type of network attached storage, in general snapshot backups are faster than the traditional Tableau Server backups. For more information see, Tableau Server External File Store.

Perform a Full Backup and Restore of Tableau Server

You can use the following steps to back up your Tableau Server deployment. Specifically, these steps describe how to recover a clone of a server from a collection of backup data and assets.

> **Note:** The backup process can take a long time to run. Since no other jobs can be run while backup is running, we recommend that you run backup during non-business hours.

Backup data types

There are two types of backup data that Tableau Server can generate. We recommend performing regular backups of each type in case you must restore a server in a recovery scenario:

- **Data managed by Tableau Server:** consists of the Tableau PostgreSQL database or repository, and File Store, which contains workbook and user metadata, data extract files, and site configuration data. When you use TSM to create a backup, all of this data is saved in a single file with a .tsbak extension. This data is backed up with the `tsm maintenance backup` command.

  Note: When an external File Store is configured you cannot use the `tsm maintenance backup` command to back up Tableau Server Data. For information on how to back up this data, see Backup and Restore with External File Store.

  - You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.
  - You can only restore a backup file to a version of Tableau Server version that is the same or newer than the version the backup was created on. You cannot restore to an older version of Tableau.
  - **Important:** If you perform Blue/Green upgrades or manually upgrade Tableau Server 2021.4 (or earlier) using the tsm maintenance (backup and restore) method, you must enable `legacy-identity-mode` before you can restore to Tableau Server 2022.1 (or later). For more information, see Troubleshoot Issues with the Identity Migration.
  - Beginning with version 2022.3, backups created using tabadmin ("pre-TSM backups") are not supported. You cannot restore a pre-TSM backup to Tableau Server version 2022.3 or later.

- **Configuration and Topology data:** includes most of the server configuration information required to fully recover a server. SMTP, alerting, some authentication assets, are all examples of configuration data that are exportable for backup. Topology data defines how your Tableau Server processes are configured in both single-server and multiple node deployments. Configuration and topology data is backed up with the `tsm settings export` command.

Note: You can change the file path used by the `tsm maintenance backup` command from the default value. For more information, see tsm File Paths.

Backup assets that require a manual process

Some configuration data is not included in the `tsm settings export` command and must therefore be documented and restored manually. The following configuration data is excluded from the `tsm settings export` operation. Your backup maintenance process should include documenting the following Tableau Server configuration data:

- System user accounts. Tableau Server setup creates an unprivileged user account, `tableau`. This account is used to access Tableau Server resources. This account can be changed during setup. If you have not changed this account, then you do not need to document it.

- TSM group membership. There are two groups created by Tableau Server: `tableau` and `tsmadmin`. If you configured alternative groups when you installed Tableau Server, then you'll need to document the group names.

  In all cases you should document the user accounts that are in these groups. To view membership in a group, run the following command `grep <group_name> /etc/-group`.

- Coordination Service deployment configuration. If you are running a multinode cluster, document which nodes are running the Coordination Services process. To view process configuration on your nodes, run `tsm topology list-nodes -v`.

- Customization settings. If your organization uses custom header or sign-in logos for Tableau Server web pages, you should include a copy of those assets with your back up portfolio. See tsm customize.

- Most authentication assets. While the locations for files may be included in an exported `settings.json` file, most certificate files, key files, keytab files or other authentication-related assets are not backed up by TSM. Verify that any of these assets you are trying to move won't need to be recreated.

  There are three exceptions:
  - The public certificate and private key for the internal PostgreSQL database (if enabled) are backed up.
  - The certificate and key for external SSL are backed up and included in the configuration data.
  - The custom certificate installed by tsm security custom-cert add (if added) is backed up.

  However, all other authentication-related assets are not backed up. For example, if you have enabled access to the PostgreSQL database with the tsm data-access repository-access enable command, be sure to document the name/password pairs for each account you've configured. These credentials are not backed up. The certificate and key for mutual SSL are not included in the back up.

- LDAP assets. Keytab files, configuration files, and or other LDAP-related assets are not backed up by TSM.

Internal server secrets and repository passwords are crypto-related configurations that are not exported. However, you do not need to document configuration values. New secrets will be created as part of the restoration process when you initialize the new instance.

Backing up Tableau Server for recovery

Tableau Server includes commands that you run to generate backup data for Tableau Server.

Note: When backing up Tableau Server on Linux, the unprivileged user must have write access to the network share where the backup files are written. Otherwise, backup will fail.

To back up server topology and configuration data, use the tsm settings command.

1. Topology and configuration data are included when you run the `tsm settings export` command. The data is exported as a json file. Specify the name and location of the json file by running the following command:

   ```
   tsm settings export -f <filename>.json
   ```

   Note: Because the backup contains secrets, we recommend that you encrypt the backup and store it in a secure place. For more information about Tableau Server secrets, see Manage Server Secrets.

2. Back up repository and File Store data. Repository data is backed up with the `tsm maintenance backup` command. Specify the name and location of the backup file by running the following command:

   ```
   tsm maintenance backup -f <filename>.tsbak -d
   ```

   The backup file is assembled in a temporary location in the data directory and then written to the directory defined in the TSM `basefilepath.backuprestore` variable:

   ```
   /var/opt/tableau/tableau_server/data/tabsvc/files/backups/<filename>.tsbak
   ```

   For more information about where backup files are written, and how to change that location, see tsm File Paths. **Note:** Even when you change the backup location, the

backup process uses a temporary location in the data directory to assemble the backup file.

Note: When File Store is configured external to Tableau Server you cannot use the tsm maintenance backup command to backup Tableau Server Data. For more information on how to backup this data, see Backup and Restore with External File Store.

Restoring core Tableau Server functionality

The procedure below uses the assets from the previous two sections to rebuild a Tableau Server in a recovery scenario.

Note: If you need to restore only the repository on an otherwise functional Tableau Server, see Restore from a Backup. If you are running a distributed deployment, and your initial node has failed, see Recover from an Initial Node Failure.

Topology and configuration backup data must be from Tableau Server on Linux. You cannot restore configuration data from a backup file that was generated on Tableau Server on Windows. To restore a backup made from Tableau Server on Windows to Tableau Server on Linux, see Migrate Tableau Server from Windows to Linux.

You must have the following assets ready:

- Topology and configuration data: This is the json file that is generated by the `tsm settings export` command.
- Repository backup file: This is the file with a .tsbak extension that is generated by the `tsm maintenance backup` command.

  You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a

server using Active Directory authentication cannot be restored to a server initialized with local authentication.

When you use `tsm maintenance restore` to restore your Tableau data, data extract files and the contents of the PostgreSQL database are overwritten with the content in the backup file (`.tsbak`). If you are running a distributed installation of Tableau Server, perform the restore on the node running the TSM Controller (this is usually the initial node).

- Backup assets: These assets include the list of documented configurations as noted in the previous section.

To restore a standalone Tableau Server

1. On the computer where you want to restore Tableau Server, Install and Initialize TSM. If your organization used non-default system user accounts, as described in an early section of this topic, then you must specify the users during this step.
2. Activate and Register Tableau Server.
3. (Optional). Configure Local Firewall.
4. (Optional). Verify LDAP.
5. Initialize Tableau Server. See Configure Initial Node Settings.
6. Import topology and configuration data. Copy the topology and configuration json backup file to the computer. Import the json file by running the following command:

   ```
   tsm settings import -f <filename>.json
   ```

7. (Optional). Apply pending changes. At a command prompt, run:

   ```
   tsm pending-changes apply
   ```

8. Restart Tableau Server. At a command prompt, run:

   ```
   tsm restart
   ```

9. Restore repository data. See Restore from a Backup.

10. (Optional). Repopulate TSM group membership. Add users to groups with this command:

    ```
    sudo usermod -G <group_name> -a <username>
    ```

To restore a Tableau Server cluster

1. On the initial node, Install and Initialize TSM. If your organization used non-default system user accounts, as described in an early section of this topic, then you must specify the users during this step.

2. On the initial node, Activate and Register Tableau Server.

3. (Optional). On the initial node, Configure Local Firewall.

4. On the initial node verify LDAP (optional), and initialize Tableau Server. See Configure Initial Node Settings.

5. On the initial node, run `tsm topology nodes get-bootstrap-file --file <path\file>.json`.

6. Copy the bootstrap.json file to all additional nodes in the cluster.

7. On each additional node in the cluster:

   a. Install the Tableau Server package.

   b. Navigate to the scripts directory.

   c. Initiate communication between initial node and the additional node:

```
sudo ./initialize-tsm -b <path-to-bootstrap>.json -u
<admin-user-on-first-node> --accepteula
```

8. On the initial node, run `tsm topology list-nodes -v` and ensure that the node names have not changed from exported topology settings. *If the node names have changed, topology settings should be manually updated with new names, or the processes should be manually configured*.

9. The Cluster Controller process is required on every node and needs to be added explicitly. From the initial node, add an instance of the Cluster Controller to each additional node, where `<nodeID>` is the ID of one of the additional nodes. Add the process to each node separately. In this example we are adding the Cluster Controller to nodes 2 and 3:

   ```
   tsm topology set-process -n node2 -pr clustercontroller -c 1
   tsm topology set-process -n node3 -pr clustercontroller -c 1
   tsm pending-changes apply
   ```

10. From the initial node, Deploy a Coordination Service Ensemble . The ensemble configuration must match your previous configuration.

11. On the initial node, import topology and configuration data. Copy the topology and configuration json backup file to the computer. Import the json file by running the following command:

    ```
    tsm settings import -f <filename>.json
    ```

12. On the initial node, apply pending changes. At a command prompt, run:

    ```
    tsm pending-changes apply
    ```

13. On the initial node, restart Tableau Server. At a command prompt, run:

    ```
    tsm restart
    ```

14. On the initial node, restore repository data. See Restore from a Backup.

15. On the initial node, repopulate TSM group membership. Add users to groups with this command:

    ```
    sudo usermod -G <group_name> -a <username>
    ```

Restore other functionality

If the previous server was configured with the following features, then you will need to re-enable and reconfigure them on the restored server:

- Authentication solutions: OpenID, external SSL, and trusted authentication. See Authentication.
- Site customizations: See tsm customize.
- Enable access to PostgreSQL repository: See tsm data-access repository-access enable.

Reencrypt Extracts After Restore

Optionally, if you are using the extract encryption at rest feature, after the backup is restored, you can reencrypt the extracts using different encryption keys. See Extract Encryption at Rest.

Run `tabcmd reencryptextracts <site-name>` to reencrypt extracts on a given site. For more information, see reencryptextracts. Run this command on every site where you are storing encrypted extracts. Depending on the number of encrypted extracts on the site, this operation could consume significant server processing load. Consider running this operation outside of business hours.

Back up Tableau Server Data

Regularly backing up Tableau Server is an important step in proper administration and maintenance of your server. You can use the `tsm maintenance restore` command to restore Tableau Server backups created using the `tsm maintenance backup` command. Database backups made in other ways, and virtual machine snapshots are not valid sources for restoring Tableau Server, so it is critical that you have an up-to-date backup.

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

Looking for Tableau Server on Windows? See Back Up Tableau Server Data.

Tableau Server data includes data extract files, as well as the Tableau PostgreSQL database, which contains workbook and user metadata, and configuration data. When you use TSM to create a backup, all of this data is saved in a single file with a .tsbak extension. If you are running a distributed installation of Tableau Server data from all the nodes is backed up.

The frequency of your backups depends on your environment, including how much use your server gets and how much and frequently the content and users change. Any changes or updates that happen after your backup will be lost if there is a system failure and you need to restore Tableau Server. The more activity there is, the more often you need to back the server up.

In addition to regularly scheduled backups, you should *always* create a backup before upgrading to a new version of Tableau Server. The upgrade process does not create a backup except when the version of PostGRES is being updated, and then the upgrade process creates a PostGRES-only backup to be used internally.

To help protect against data loss, after you create the backup, you should store the .tsbak file on a computer that is not a part of your Tableau Server installation.

Disk Space Usage for Backup

The free disk space required to create a backup varies depending on the amount of data in the Tableau Server repository and file store services, and their collocation with the tabadmincontroller service. During backups, the background tasks for cleaning up old extracts are

temporarily paused. This means that, for the duration of the backup, extract refreshes will leave extra files in place, adding to disk space usage. If your backup takes a long time, or if your organization uses many extracts that are regularly updated, this can result in a significant amount of temporary disk space usage. These temporary files will be removed after the backup is complete.

The following table lists the disk space requirements for backup based on whether the node hosts the repository, file store, controller, or some combination of them. In multi-node Tableau Server environments you need to estimate the required disk space on each node.

Repository	File Store	Controller	Disk Space Required
✓			3x repository data + 250 MB To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base` directory. To obtain the exact size of the repository data, open the backup file and use the size of the workgroup.pg_dump file.
	✓		1.5x file store data To obtain an estimate of file store data (extracts, flows, etc.), check the size of `<data directory>/dataengine` directory.
		✓	3x repository data + 250 MB + 2.5x file store data
✓	✓		3x repository data + 250 MB + 1.5x file store data

	✓	✓	3x repository data + 250 MB + 1.5x file store data
✓		✓	3x repository data + 250 MB + 2.5x file store data
✓	✓	✓	3x repository data + 250 MB + 1.5x file store data

Optimizing Tableau Server Backup

There are several ways you can maximize backup efficiency. Your environment can impact how effective each of these is, so test with your data to see what works best.

Optimizing with topology configurations:

- Co-locating File Store on the same node as the Administration Controller can reduce the length of time it takes to back up Tableau Server by reducing or eliminating the need to transfer data between nodes during the backup process. This is especially true if your organization uses many extracts.
- Co-locating the repository (pgsql) with the Administration Controller node can also help to reduce back up time, but the time savings is less significant than that of the File Store.

The Administration Controller is usually on the initial node, unless you have had an initial node failure and moved the controller to another node.

Optimizing with backup strategies:

Backup is a resource intensive process. If possible, doing your backups during off peak hours is a generally a good strategy. But this however, depends on your requirements and how often Tableau Server data is updated and what your restore requirements. For a detailed explanation of backup and disaster recovery, see Tableau Server Disaster Recovery. Here are some backup strategies and adopt them to your requirements

- **Type of storage:** Solid State disks are recommended in general for backups. SSD helps make your backups faster and complete sooner compared to traditional spinning

disks.

- **Backup compression:** You have the option of running your backups with or without compression. When you do your backup with compressions, your backup size will be comparatively smaller, but you may see a slower performance. So if your goal is more focused on speed, choose the `--skip-compression` option:

  Use the `--skip-compression` option when backing up Tableau Server. This creates the backup without using compression, and results in a larger backup file but can reduce the amount of time it takes for the backup to complete. For more information, see tsm maintenance backup.

- **Snapshot backup:** This option is only available if you have configured your Tableau Server with External File Store. Although the performance of snapshot backups depend on the type of network attached storage, in general snapshot backups are faster than the traditional Tableau Server backups. For more information see, Tableau Server External File Store.

Create a backup using the TSM command line interface (CLI)

Use the `tsm maintenance backup` command to create a backup of the data managed by Tableau Server. This data includes data extract files and the Tableau PostgreSQL database, which contains workbook and user metadata.

Important: Do not use the `pg-only` option when generating a backup unless instructed by Tableau Support. This option will only back up the repository and *cannot* be used to restore your Server. Its primary use is for troubleshooting, and Tableau Support will ask you to create a `--pg-only` back up if this is necessary.

Note: When backing up Tableau Server on Linux to a network location, the unprivileged user must have write access to the network share where the backup files are written or the backup will fail.

To back up server configuration data, use the tsm settings command. When you use the `tsm maintenance backup` command, the current date is appended to the backup file:

```
tsm maintenance backup -f <backup_file> -d
```

For more information, see tsm maintenance backup.

Create a pre-upgrade backup

You should always create a backup before upgrading Tableau Server. You can create a backup while Tableau Server is running and minimize the amount of time the server is unavailable during upgrade. The process for creating a pre-upgrade backup is the same as for creating regular backups, with one additional consideration for distributed installations.

Note: Uninstall Tableau Server from any nodes that you are not including in your new installation to avoid conflicts between the older nodes and the new installation.

Backups during upgrades

During a Tableau Server upgrade, when necessary, a temporary backup of the database may be created to allow for migrations that occur as part of upgrades. This is done during the upgrade and in most cases has no noticeable impact to the upgrade process. In certain special cases there can be additional impacts:

- Upgrades to Tableau Server 2022.1 (or later) from version 2021.4 (or earlier)—If you perform Blue/Green upgrades or manually upgrade Tableau Server 2021.4 (or earlier) using the tsm maintenance (backup and restore) method, you must enable `legacy-identity-mode` before you can restore to Tableau Server 2022.1 (or later). For more

information, see Troubleshoot Issues with the Identity Migration.

- Major version pgsql updates—If an upgrade includes a major version update to the database used for the Tableau repository, the internal upgrade backup is done without compression to save time. This requires additional temporary disk space during the upgrade process.

  Tableau Server versions that include a major version database update; 2020.4.

Scheduling and Managing Backups

Beginning in 2020.4.0 you can use tsm commands to schedule a backup. You need to do this from the command line (there is no TSM UI to schedule backups). The `tsm maintenance backup` command allows you to create and update backup schedules. The `tsm schedules` commands give you the ability to view, delete, pause, resume, and update schedules.

To schedule a backup:

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run the following command:

   ```
   tsm maintenance backup -f <backup-file> -sr <recurrence> -st
   <time-to-run> -sd <days-to-run> -sn <schedule-name>
   ```

   For example, to create a backup schedule named "monthly-backup" that runs on the 15th of each month at 2 am and generates a file called `<yyyy.mm.dd.hh.mm>-ts-mid_month_backup.tsbak`:

   ```
   tsm maintenance backup -f ts-mid_month_backup -sr monthly -st
   02:00 -sd 15 -sn monthly-backup
   ```

To view a scheduled backup:

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run the following command:

   ```
   tsm schedules list
   ```

   You can sort the schedules by scheduled run time, earliest to latest, or by name using the `--next-run` or `--schedule-name` options. You can alternately display details for a single schedule using the `--schedule-id` option. When you view a single schedule you see additional details about it, including when i twas created, how many times it has run, and specific options used when it is run. Job options are shown in JSON format as "Job args".

To update a scheduled backup:

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run the following command:

   ```
   tsm schedules update --schedule-id <ID> --schedule-time <time-to-run> --schedule-recurrence <frequency> --schedule-days <day-to-run>
   ```

   Note: To add or change a name, use the `tsm-maintenance-backup` command.

To suspend or resume a backup schedule:

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run one of the following commands:

   - To suspend a schedule:

     ```
     tsm schedules suspend --schedule-id <scheduleID>
     ```

- To resume a suspended schedule:

```
tsm schedules resume --schedule-id <scheduleID>
```

Script the backup process

If you back up often, you might want to create a script that performs the backup and related tasks for you. These tasks include:

- Clean up files and folders before running the backup.
- Running the backup itself.
- Copying the backup file to a separate computer for safekeeping.

This section discusses `tsm` commands you can use together to perform a backup and related tasks.

Remove log files and clear temporary folders

You can clean old Tableau Server log file and temporary files to reduce the time it takes to create a backup, and to ensure the backup file is as small as possible.

To clean log files older than a few days, run the following command:

```
tsm maintenance cleanup
```

Run the backup

> **Note:** When backing up Tableau Server on Linux to a network location, the unprivileged user must have write access to the network share where the backup files are written or the backup will fail.

To create the backup, use the `tsm maintenance backup` command:

```
tsm maintenance backup --file <backup_file> --append-date
```

Note the following about the command:

- Add `--append-date` to the command to include the date in the backup file name.
- The backup file is assembled in a temporary location in the data directory and then written to the directory defined in the TSM `basefilepath.backuprestore` variable:

  `/var/opt/tableau/tableau_server/data/tabsvc/files/backups/<file-name>.tsbak`

  For more information about where backup files are written, and how to change that location, see tsm File Paths. **Note:** Even when you change the backup location, the backup process uses a temporary location in the data directory to assemble the backup file.

Copy the backup file to another computer

As a best practice, after the backup is created, copy the backup file to another location that is separate from Tableau Server.

Restore from a Backup

Use the `tsm maintenance restore` command to restore your Tableau Server data. You might do this if you had a system failure and need to restore your data, if you need to switch back to a previous version of Tableau Server (for example, if there is a problem with an upgrade), or if you are moving Tableau Server to new hardware. You can use the `tsm maintenance restore` command to restore Tableau Server backups created using `tabadmin backup` and `tsm maintenance backup`.

Limitations when restoring Tableau Server

- If you perform Blue/Green upgrades or manually upgrade Tableau Server 2021.4 (or earlier) using the tsm maintenance (backup and restore) method, you must enable `legacy-identity-mode` before you can restore to Tableau Server 2022.1 (or later). For more information, see Troubleshoot Issues with the Identity Migration.

- Database backups made in other ways, and virtual machine snapshots are not valid sources for restoring Tableau Server.
- When you use `tsm maintenance restore` to restore your Tableau data, data extract files and the contents of the PostgreSQL database are overwritten with the content in the backup file (`.tsbak`). If you are running a distributed installation of Tableau Server, perform the restore on the node running the TSM Controller (this is usually the initial node).
- You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.
- You can only restore a backup file to a version of Tableau Server version that is the same or newer than the version the backup was created on. You cannot restore to an older version of Tableau.
- Beginning with version 2022.3, backups created using tabadmin ("pre-TSM backups") are not supported. You cannot restore a pre-TSM backup to Tableau Server version 2022.3 or later.
- During restore, the restore process will initiate a full re-indexing of the content and external assets managed by Tableau Server. This process consumes CPU resources which may be noticeable during backup and restore.

Restore Tableau Server from a backup file

Note: This operation includes steps that you may need to perform using the TSM command line.

1. (Optional) Copy the `.tsbak` file to the default file location.

   The `restore` command expects a backup file in the directory defined in the TSM `basefilepath.backuprestore` variable. By default:

   ```
   /var/opt/tableau/tableau_server/data/tabsvc/files/backups/
   ```

   For more information about file paths and how to change them, see tsm File Paths.

   > **Note:** If you are restoring a backup that was copied into the backups folder, ensure that the unprivileged user (by default, `tableau`), has at least Read access to the backup file. Otherwise, the restore process may not be able to unzip the backup file and the restore will fail.

2. Stop the server. At a command prompt, type:

   ```
   tsm stop
   ```

3. Restore from a backup file. At a command prompt, type:

   ```
   tsm maintenance restore --file <file_name>
   ```

   In the above line, replace `<file_name>` with the name of the backup file you want to restore from.

   > **Note:** If you encounter errors when trying to restore from backup, see Troubleshoot Tableau Server on Linux.

4. Restart the server:

   ```
   tsm start
   ```

Server Maintenance

As an administrator, you will want to check the status of the server, analyze and monitor the activity on the server, manage scheduled tasks, or perform certain maintenance activities such as clearing saved data connection passwords. In addition, there are several settings that you may want to specify to customize the user experience for people using the server. You can do some of these tasks from the General page of the Status page and others from the Settings page.

View Server Process Status

You can view server process status can be by running a TSM CLI command or by accessing TSM Web UI or Admin pages on Tableau Server.

Viewing process status with TSM CLI

Run the following command:

```
tsm status -v
```

This command outputs all of the processes that are configured on the instance and their corresponding status.

Viewing process status in web UI

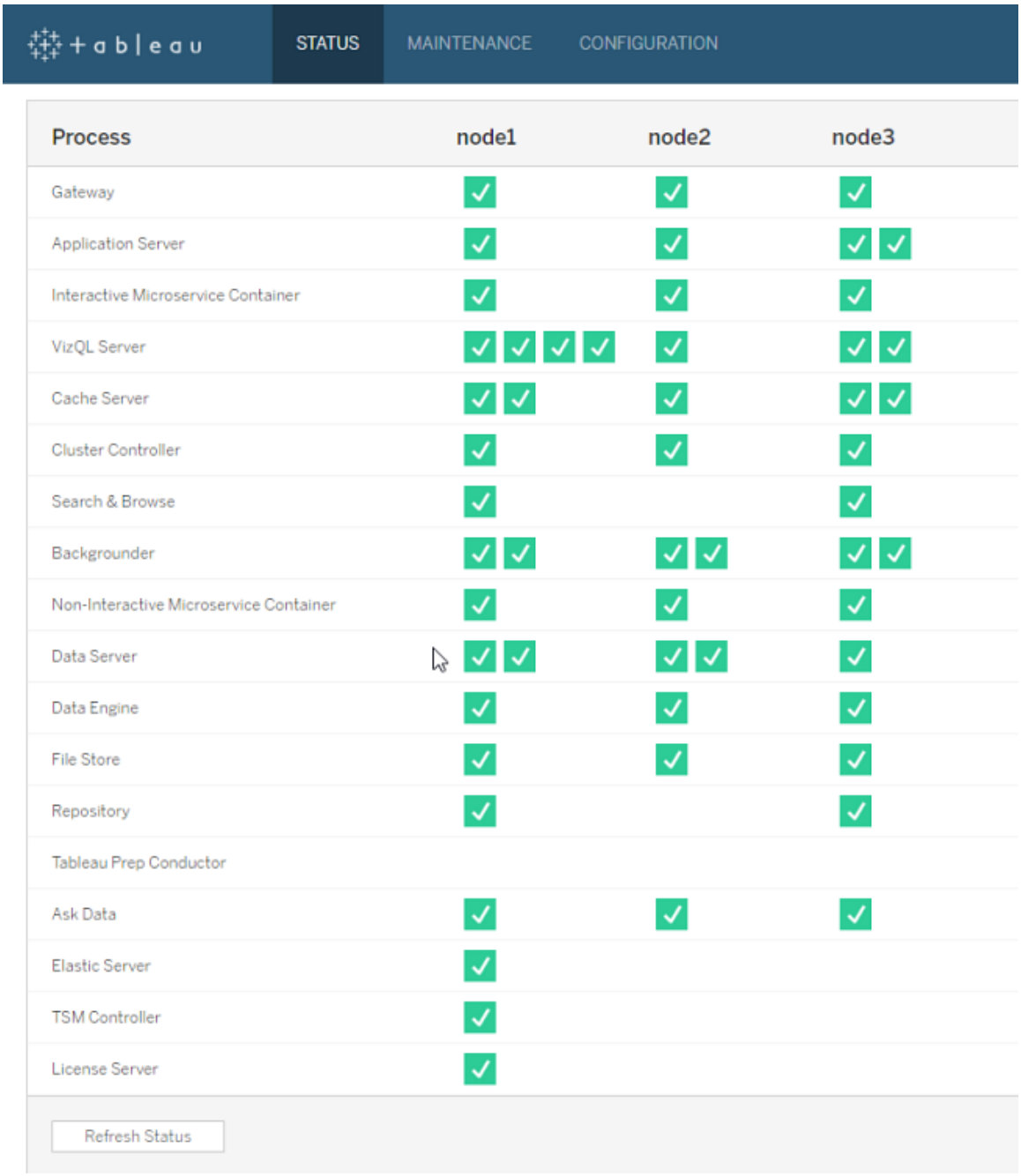
There are two locations in Tableau Server or Tableau Services Manager (TSM) where administrators can view the state of Tableau processes. You may be able to access one or both of these locations, depending on how your account and server are set up. Most of the process status information that displays is duplicated on both Status pages. This section explains each page, and identifies what is unique for each one.

Note The status pages displayed in both web UI locations show a subset of the total processes configured on a given node. To view all processes, you must run the TSM CLI command, `tsm status -v`.

- The Tableau Services Manager (TSM) status page is accessible in TSM and can be viewed by TSM administrators. You must be able to log into TSM to see this page. For information about signing into TSM, see Sign in to Tableau Services Manager Web UI.
- The Tableau Server status page appears in the Tableau Server web UI and is accessible by Tableau Server administrators. This page includes Tableau Server processes, along with links to troubleshooting documentation if a process is not running as expected. If you hover your mouse pointer over the status indicator for a process, a tooltip shows the node name and the port the process is running on. The Tableau Server status page does not show TSM processes. For information about signing into Tableau Server as an administrator, see Sign in to the Tableau Server Admin Area.

Tableau Services Manager (TSM) Status page

The TSM Status page shows you the state of server processes, including TSM Controller and License Server. These two processes do not display on the Tableau Server Status page.



Possible status indicators are listed at the bottom of the table:

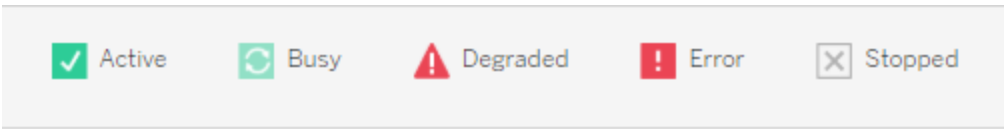
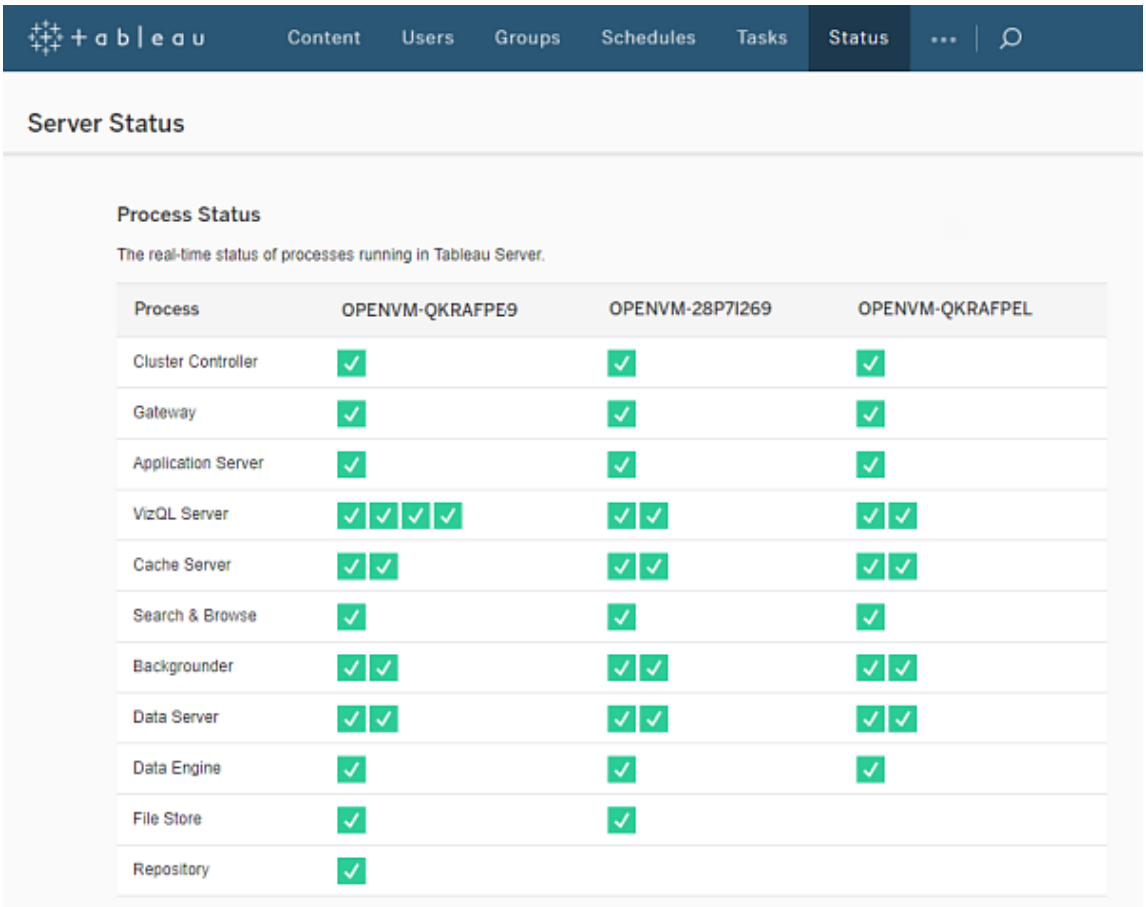
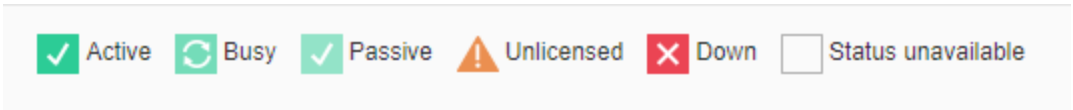


Tableau Server Status page



Possible status indicators are listed at the bottom of the table:

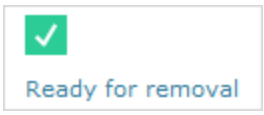


When Tableau Server is functioning properly, most processes will show as Active, Busy or Passive (Repository):

- **Active**—The process is functioning as intended. See File Store in Troubleshoot Server Processes for details on possible active states.
- **Busy**—The process is completing some task. See File Store and Repository in Troubleshoot Server Processes for more information.

- **Passive**—The repository is in passive mode, or a process cannot receive traffic. See Repository and Index and Search Server in Troubleshoot Server Processes for more information.
- **Unlicensed**—The process is unlicensed.
- **Down**—The process is down. The implications of this differ depending on the process.
- **Status unavailable**—Tableau Server is unable to determine the status of the process.

If there is additional information, a message displays below the status icon and links to appropriate documentation:



Note: Tableau Server is designed to be self-correcting. If a service or process stops responding or goes down, Tableau Server attempts to restart it. This can take 15 to 30 minutes to complete. Because of this, reacting immediately to service or process alerts can be counter-productive, especially in an installation with redundant services that can handle requests while one restarts.

For more information about troubleshooting process status, see Troubleshoot Server Processes.

External Node

Some processes can be configured external to Tableau Server. For example, File Store can be configured on a SAN or NAS, the repository can be deployed to an AWS RDS instance. In such cases, the Tableau Server Status page will show these processes on **External Node** with status **E** , and the Tableau Services Manager (TSM) status page will show these processes on **external** with a check mark to indicate that the process is configured externally.

Tableau Server Manager (TSM) status page showing File Store as configured external to Tableau Server:

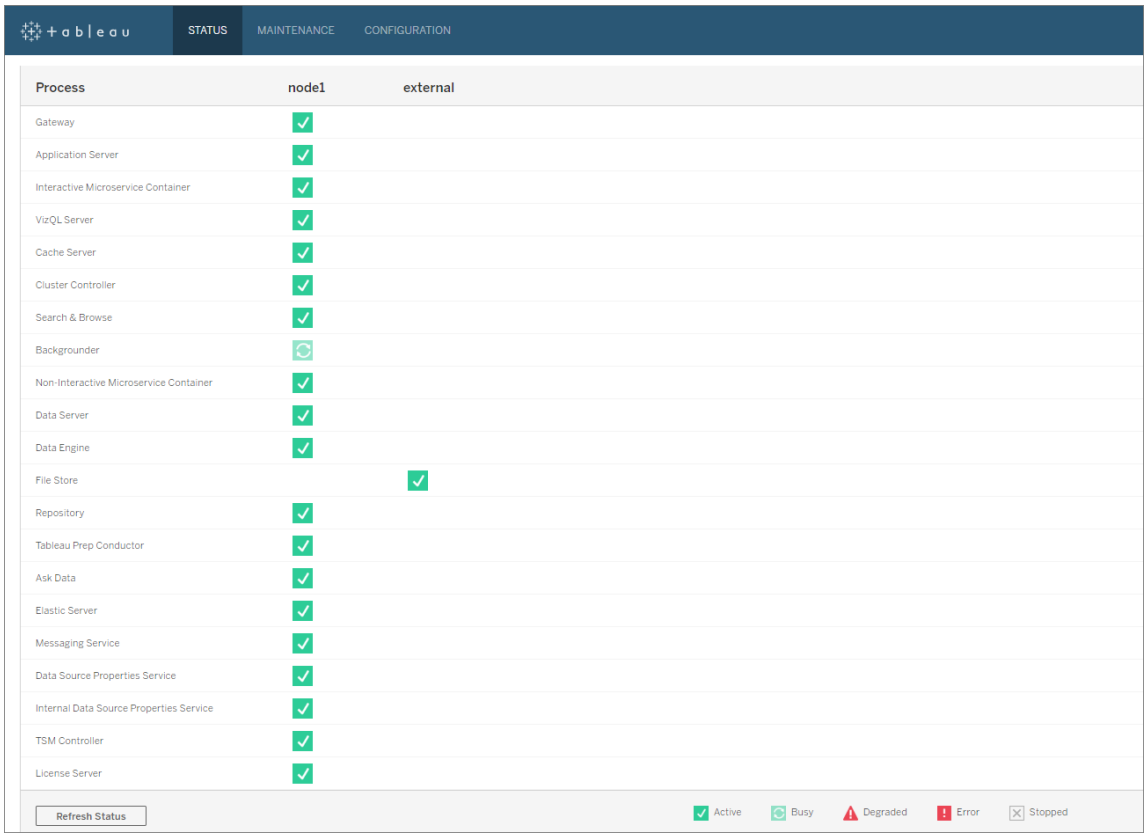
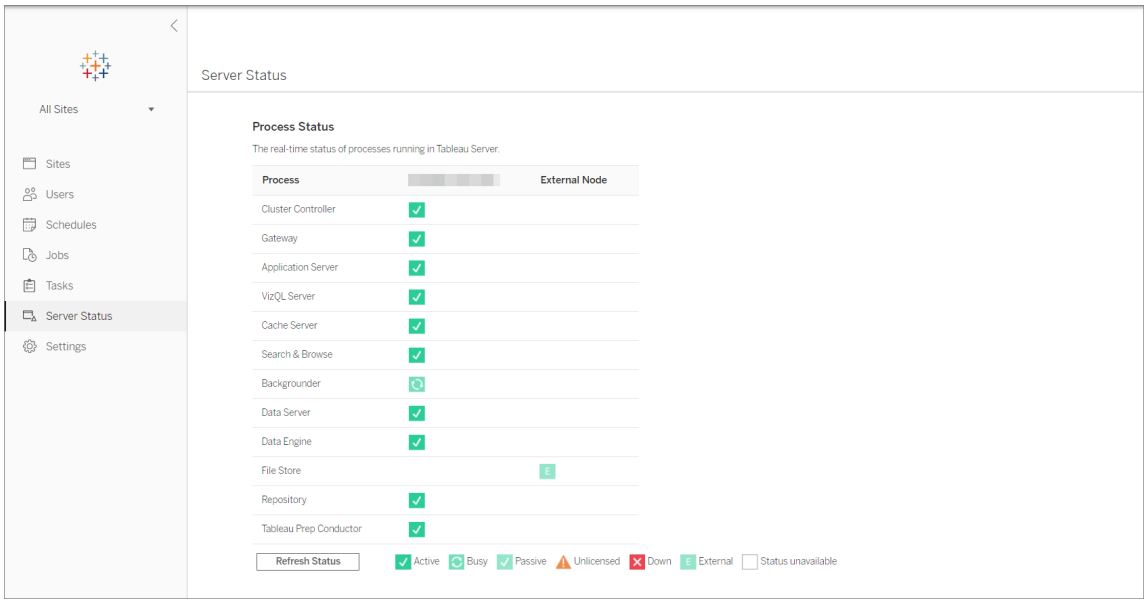


Tableau Server status page showing File Store as configured external to Tableau Server:



Access Status Remotely

> **Note**: The information in this article refers to the Tableau Server status page. For information about the Tableau Server status page and the TSM status page, see View Server Process Status.

You must be a Tableau Server administrator to see the Server Status page, but you can grant remote access to other computers to allow access to a machine-readable (XML) version of the Status table by non-admin users and by computers other than the initial Tableau Server node. One reason you might do this is as part of a remote monitoring process.

To grant remote access to Tableau Server status:

1. Open a command prompt as an administrator and type the following:

   ```
   tsm configuration set -k wgserver.systeminfo.allow_referrer_ips
   -v <ip address>
   ```

   In the above command, `<ip address>` is the IPv4 address of the computer for which you want to enable remote access to the Tableau Server status XML.

   For example:

   ```
   tsm configuration set -k wgserver.systeminfo.allow_referrer_ips
   -v 10.32.139.31
   ```

   If you are enabling remote access for more than one computer, use commas to separate each IP address.

   ```
   tsm configuration set -k wgserver.systeminfo.allow_referrer_ips
   -v 10.32.139.31,10.32.139.35
   ```

2. Commit the configuration change:

   ```
   tsm pending-changes apply
   ```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Now, users of computers with the IP addresses that have been added can view Tableau process status by entering the URL `http://<server->/admin/systeminfo.xml` in a browser or from a command line (for example, `curl http://jsmith/admin/systeminfo.xml`).

If Tableau Server has been configured to work with a load balancer or proxy server, use the hostname or IP address of the initial Tableau Server node to access the XML version of the status page.

For details on the XML that is returned, see Get Process Status as XML.

Get Process Status as XML

To get a machine-readable version of the server process status, that is, a version of the status formatted in XML, use the following URL:

```
http://my_tableau_server/admin/systeminfo.xml
```

You must be signed in to Tableau Server to view the machine-readable process status, or have enabled remote access.

The server returns a status report similar to the following:

```
<systeminfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <machines>
    <machine name="my_tableau_server">
    <repository worker="my_tableau_server:8060" status="Active" pre-
ferred="false"/>
    <dataengine worker="my_tableau_server:27042" status="Active"/>
    <applicationserver worker="my_tableau_server:8600"
```

```
status="Active"/>
    <apiserver worker="my_tableau_server:8000" status="Active"/>
    <vizqlserver worker="my_tableau_server:9100" status="Active"/>
    <dataserver worker="my_tableau_server:9700" status="Active"/>
    <backgrounder worker="my_tableau_server:8250" status="Active"/>
    <gateway worker="my_tableau_server:80" status="Active"/>
    <searchandbrowse worker="my_tableau_server:11000" status-
s="Active"/>
    <cacheserver worker="my_tableau_server:6379" status="Active"/>
    <filestore worker="my_tableau_server:9345" status="Active"
pendingTransfers="0" failedTransfers="0" syncTimestamp="2015-02-
27T20:30:48.564Z"/>
    <clustercontroller worker="my_tableau_server:12012" status-
s="Active"/>
    <coordination worker="my_tableau_server:12000" status="Active"/>
    </machine>
  </machines>
  <service status="Active"/>
</systeminfo>
```

Status values in the XML

- <process> **worker** - The name of the node running the process and the port the process is using.
- **status** - The status of the process on the node. Possible values are: `Active`, `Passive`, `Unlicensed`, `Busy`, `Down`, `ReadOnly`, `ActiveSyncing`, `StatusNotAvailable`, `StatusNotAvailableSyncing`, `NotAvailable`, `DecommisionedReadOnly`, `DecomisioningReadOnly`, and `DecommissionFailedReadOnly`
- **pendingTransfers** - A count of the workbook or data source extracts the node needs to get to be fully synced. These represent items that were published to this file store node, and items that were published to other file store nodes and need to be copied to this node.

- **failedTransfers** - A count of the workbooks or data sources that did not transfer successfully to this file store node during the last automated job. The automated job normally runs about every 15 to 30 minutes, but may take longer when transferring a large number of extracts or large extracts.

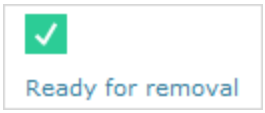
  Failed transfers do not necessarily indicate a problem with Tableau Server. The recurring automated job will normally transfer files that failed during the previous sync. Reasons for failed file transfers are listed in the logs.

- **syncTimestamp** - The time in UTC of the last automated job that ran and synchronized files.

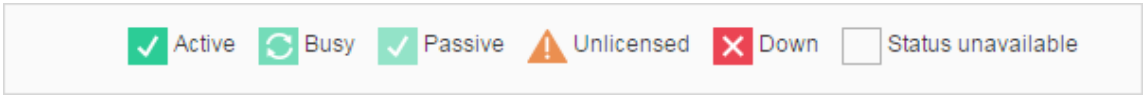
Troubleshoot Server Processes

> **Note**: The information in this article refers to the Tableau Server status page. For information about the Tableau Server status page and the TSM status page, see View Server Process Status.

When Tableau Server is functioning properly, processes will show as Active, Busy or Passive (Repository). If there is additional information, a message appears below the status icon:



Possible status indicators are:



> **Note:** Tableau Server is designed to be self-correcting. If a service or process stops responding or goes down, Tableau Server attempts to restart it. This can take 15 to 30 minutes to complete. Because of this, reacting immediately to service or process alerts can be counter-productive, especially in an installation with redundant services that can handle requests while one restarts.

The following sections provide troubleshooting recommendations for status messages that you may see.

Cluster Controller

This message will only display if you have more than two nodes.

Status: Down; **Message: "Node degraded"**

One or more of the following are true:

- Repository on the node is stopped.
- Node cannot respond to failover elsewhere on the cluster.
- If Tableau Server is configured for high availability and this is the active repository, fail-over to the second repository occurs.
- No status available for repository or file store on this node.

No action is necessary unless the cluster controller is regularly down or is down for an extended period of time.

If that occurs, take the following actions, in order, until the problem is resolved:

1. Check disk space. If disk space is limited, save the log files (use `tsm maintenance ziplogs`) in case you need them for Support, then remove unnecessary files.
2. Restart Tableau Server.
3. If Cluster Controller continues to show as down, save the log files (`tsm maintenance ziplogs`) and contact Support.

File Store

File Store status only reflects the state of the File Store when the page was loaded.

An active status () with no message indicates that no extracts were being synchronized when the page was loaded. It is possible that the recurring "catch-all" job is running and synchronizing extracts.

 **Status: Busy**; **Message: "Synchronizing"**

"Synchronizing," usually indicates that extracts were being synchronized across File Store nodes when the page was loaded.

However, the "synchronizing" message is also returned following installation (both single-node and multi-node). After Tableau initializes the status should disappear within 15 or 20 minutes.

 **Status: Down**; **"Data Extracts unavailable"**

On a single-node installation: "Data Extracts unavailable" indicates that existing extracts may be available but publish/refresh will fail. On multi-node installations, this message indicates that extract synchronization will fail for this node.

No action is necessary unless the File Store is regularly down or is down for an extended period of time.

If that occurs, take the following actions, in order, until the problem is resolved:

1. Check disk space. If disk space is limited, save the log files (use `tsm maintenance ziplogs`) in case you need them for Support, then remove unnecessary files.
2. Restart Tableau Server.
3. If File Store continues to show as down, save the log files (`tsm maintenance ziplogs`) and contact Support.

 **Status: Busy**; **"Decommissioning"**

This message indicates that this File Store is in read-only mode and that any unique files on this node are being replicated to other File Store nodes.

To remove this node, wait until the status message changes to "Ready for removal".

 **Status: Active**; **"Ready for removal"**

This message indicates that the File Store is in read-only mode.

You can safely stop (`tsm stop`) the cluster and remove File Store processes, or remove entire node.

 **Status: Active**; **"Decommission failed"**

This message indicates that the File Store is in read-only mode, and that at least one unique file failed to replicate to another File Store node.

To resolve a failed decommissioning:

1. Run the `tsm topology filestore decommission` command again.
2. Check disk space on other File Store nodes. Decommissioning will fail if another File Store node does not have enough space to store all the extracts.
3. Check the `tsm.log` file on the initial node and additional nodes for errors.
4. Stop Tableau Server (`tsm stop`) and then try running the`tsm topology filestore decommission` command again.
5. Put the File Store node back into read/write mode (`tsm topology filestore recommission`), collect logs, and then contact Support.
6. With Support: copy and merge `extracts` directory from this File Store node to the same directory on another File Store node.

Index and Search Server

 Status: Passive; **Message: n/a**

In multi-node environments, a passive status indicates that the node is working as intended, but cannot join the cluster and receive traffic.

To get the Index and Search Server process to active status:

1. Use the tsm topology set-process command to remove passive Index and Search Server processes from the nodes.

```
tsm topology set-process -n <Node> -pr indexandsearchserver -c
0
```

2. Apply the changes (tsm pending-changes apply).
3. Restart Tableau Server (tsm restart).
4. Use the tsm topology set-process command to add the Index and Search Server process to nodes one at a time.

   ```
   tsm topology set-process -n <Node> -pr indexandsearchserver -c
   1
   ```

5. Apply the changes (tsm pending-changes apply --ignore-warnings).
6. Restart Tableau Server (tsm restart).
7. Use the tsm status command to check the status of `indexandsearchserver` on affected nodes.

Repository

Status: Busy; **Message: "Setting up"**

The "Setting up" message indicates one or more of the following states:

- Passive repository is being synchronized with active repository.
- Repository is not ready to handle failover.
- Repository may have gotten more than two minutes behind active repository and is being setup again (this is faster than waiting for a sync).
- Failover occured and this former active repository is rejoining the cluster.

Wait until the repository status message changes to "Passive".

If this message does not appear, or if it is taking a long time:

1. Check disk space and free space if possible.
2. Check cluster controller logs for errors.
3. Restart node.

Status: Busy; **Message: "Synchronizing"**

Repository is synchronizing, for example after a failover.

Status: Down; **Message: n/a**

When the Repository shows a status of down and there is no message, then the Repository is in one of the following states:

- If the installation is configured for high availability, failover of the repository occurred.
- Processes are restarting with updated database connection configurations after fail-over.
- If another active repository is not available, Tableau Server is down.

Take these actions in order until a step resolves the problem:

1. Wait several minutes for cluster controller to attempt to restart.
2. Restart Tableau Server (`tsm restart`).
3. Check disk space. If disk space is limited, save the log files (use `tsm maintenance ziplogs`) in case you need them for Support, then remove unnecessary files.
4. Restart Tableau Server.
5. If repository continues to show as down, save the log files (`tsm maintenance ziplogs`) and contact Support.

Status: Passive; **Message: n/a**

A passive status with no message indicates that the node is working as intended and that it is ready for failover if needed.

VizQL Server

Status: Unlicensed; **Message: n/a**

For information about unlicensed status for a VizQL Server process, see Handle an Unlicensed Server Process.

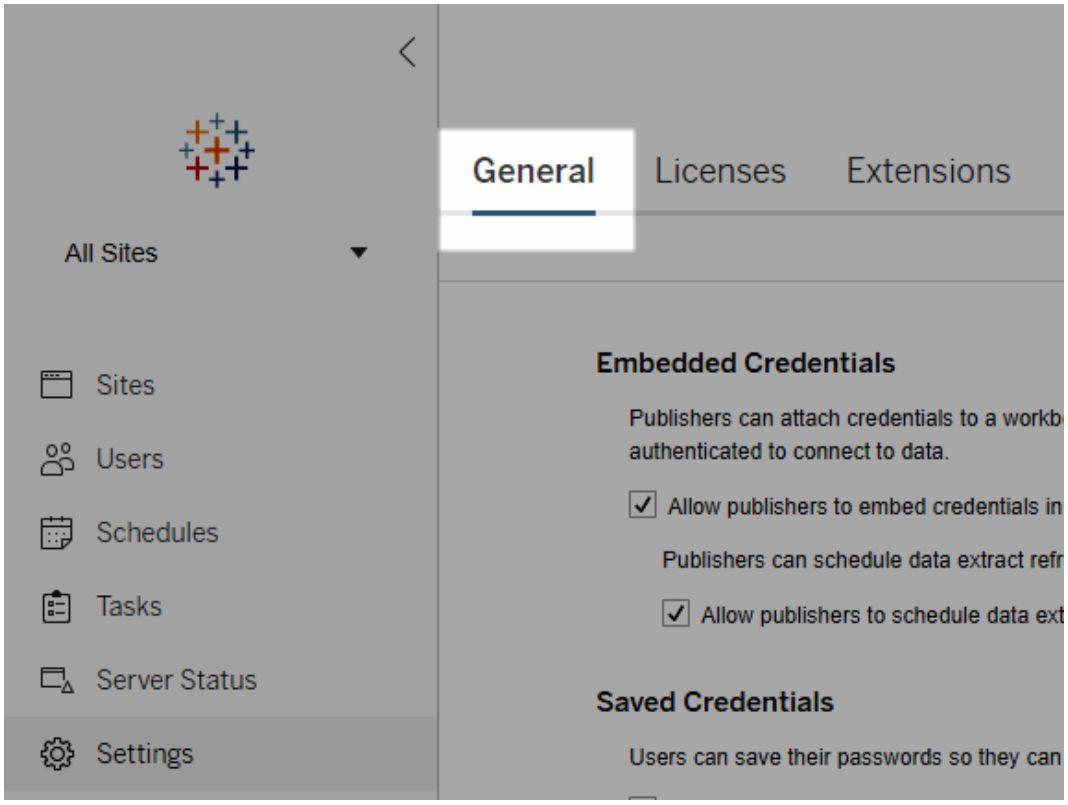
Clear Saved Data Connection Passwords

As the administrator, if you enable users to save data source passwords, server users can save data source passwords across multiple visits and browsers so they are no prompted for their credentials each time they connect to a data source.

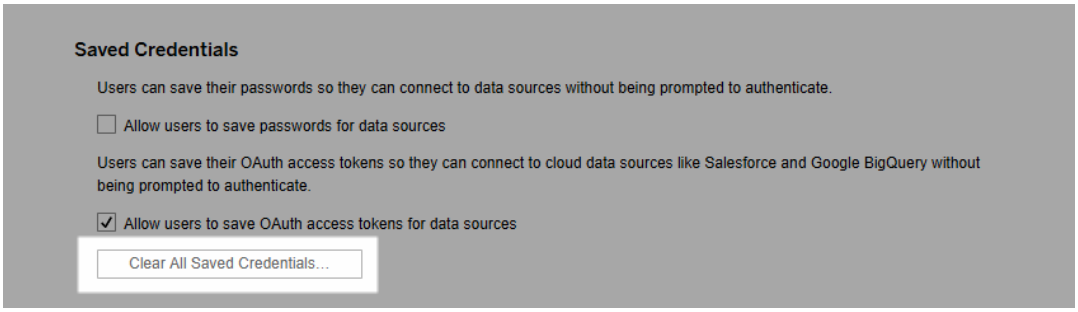
You can reset the data source passwords for all Tableau Server users. Doing this forces them to sign in to the data sources the next time they visit a view that requires database authentication. Server users can also clear their saved data connection passwords on an individual basis using their User Preferences page.

To clear saved data connection passwords for all server users:

1. In a single-site server, click **Settings** > **General**. On a multi-site server, select **Manage all sites**, then click **Settings** > **General**.

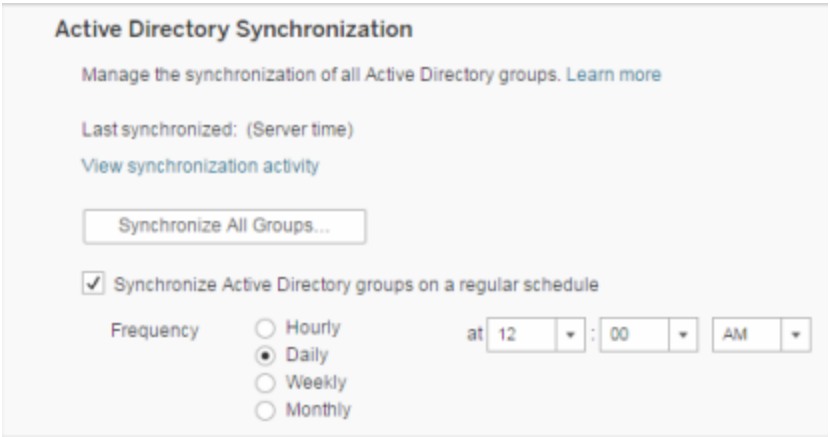


2. Under Saved Credentials, click **Clear All Saved Credentials**.



Synchronize External Directory Groups on the Server

As a server administrator, you can synchronize all external directory (such as Active Directory) groups (that have been configured on Tableau Server) on a regular schedule or on-demand on the **General** tab of the **Settings** page for the server.



Note: In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

Before you begin

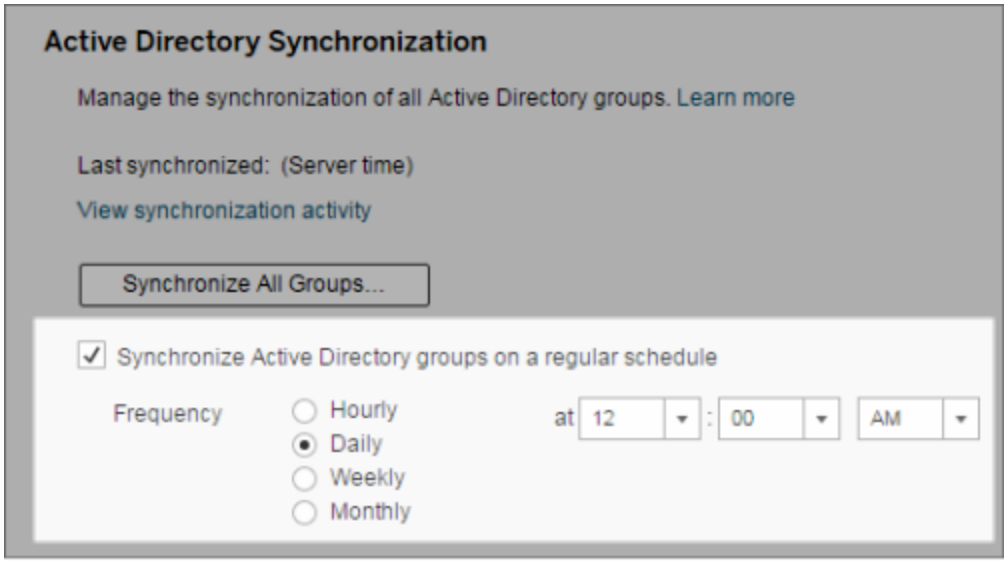
Before synchronizing groups as described in this topic, you must first import the external directory group into Tableau Server. See Create Groups via Active Directory.

Synchronize external directory groups on a schedule

1. **Single-site**: Click **Settings**> **General**.

   Multisite: In the site menu, click **Manage All Sites** and then click **Settings**> **General**.

2. Scroll down the page to **Active Directory Synchronization**, and then select **Synchronize Active Directory groups on a regular schedule**.



3. Select the frequency and time of synchronization.

4. Click **Save**.

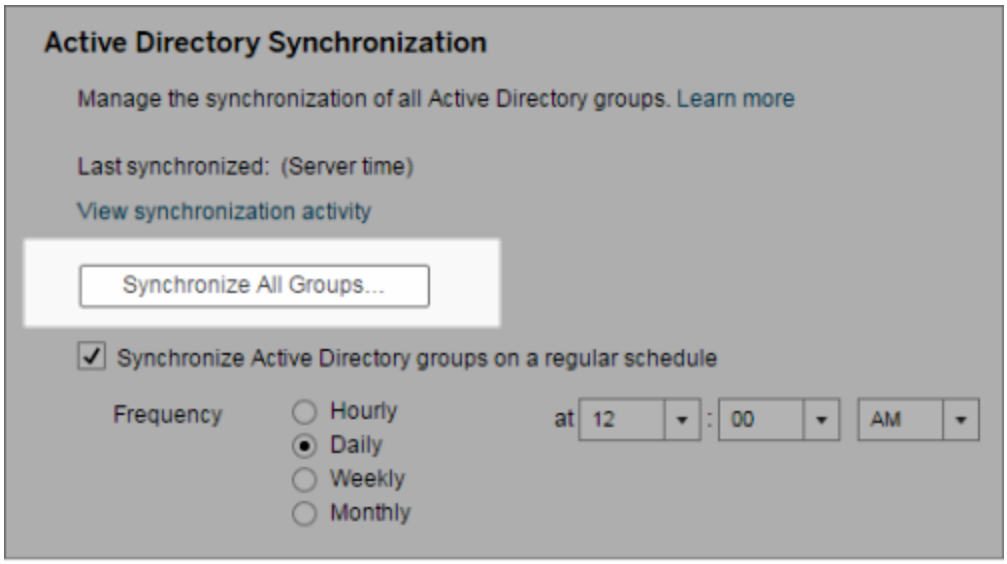
Note: Beginning in versions 2021.1.23, 2021.2.21, 2021.3.20, 2021.4.15, 2022.1.11, 2022.3.3, 2023.1, a default time limit of 4 hours limits how long a scheduled group synchronization can take before it is canceled. A server administrator can change this time limit if your scheduled synchronization is of very large groups, or taking longer than the default. For more information, see Synchronize All Active Directory Groups on a Schedule and backgrounder.timeout.sync_ad_group.

Synchronize all external directory groups on demand

At any time, you can synchronize external directory (such as Active Directory) groups with Tableau Server to ensure that new users and changes in the external directory are reflected in all external directory groups on Tableau Server.

1. **Single-site**: Click **Settings**> **General**.

   Multisite: In the site menu, click **Manage All Sites**, and then click **Settings**> **General**.



2. Under **Active Directory Synchronization**, click **Synchronize All Groups**.

View synchronization activity

You can view the results of synchronization jobs in the **Background Tasks for Non Extracts** administrative view. **Queue Active Directory Groups Sync** is the task that queues and indicates the number of **Sync Active Directory Group** tasks to be run.

1. **Single-site**: Click **Status**.

   Multisite: In the site menu, click **Manage All Sites** and then click **Status**.

2. Click the **Background Tasks for Non Extracts** link.

3. Set the **Task** filter to include **Queue Active Directory Groups Sync** and **Sync Active Directory Group**.

You can quickly navigate to this administrative view by clicking the **View synchronization activity** link in the **Settings** page for the server.

Set the minimum site role for users in an external directory group

In the **Groups - Details** page, you can set the minimum site role for group users to be applied during Active Directory synchronization.

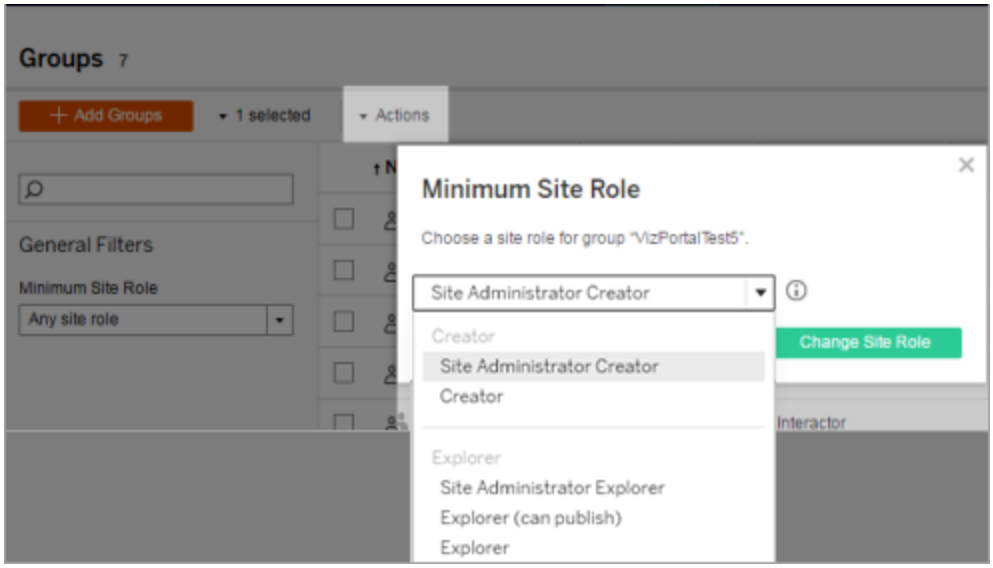
This setting does not run synchronization; instead, it sets the minimum site role to applied to the group every time synchronization runs. The result is that when you synchronize external directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role is applied if it gives the user more access in a site. If you don't set a minimum site role, new users are added as **Unlicensed** by default.

Note: A user's site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see Site roles and Active Directory import and synchronization.

1. In a site, click **Groups**.
2. On the Groups page, select a group.

   Click **Actions** > **Minimum Site Role**.

3. Select the minimum site role, and then click **Change Site Role**.



What happens when users are removed in the source external directory?

Users cannot be automatically removed from the Tableau Server through an external directory sync operation. Users that are disabled, deleted, or removed from groups in the external directory remain on Tableau Server so that administrators can audit and reassign the user's content before removing the user's account completely. For more information, see Sync behavior when removing users from Active Directory.

Improving group synchronization performance

External directory synchronization is performed by the backgrounder process. The Backgrounder process is the same process that is used for managing and creating extracts, and is also used to generate subscription content. In large organizations with dynamic group membership and heavy extract usage, the external directory group synchronization process may be disruptive. We recommend running group synchronization during non-business hours.

By default, the Backgrounder process performs synchronization in a serial operation. This means that each group is synchronized, one after the other, in a single Backgrounder process. If you are running multiple instances of Backgrounder processes either on a single Tableau Server or across a distributed deployment, consider enabling parallel processing for external

directory synchronization. When parallel Backgrounder processing is enabled, the group synchronization is distributed across multiple Backgrounder processes for better performance.

To enable parallel backgrounder processing for group synchronization, open TSM CLI and enter the following commands:

```
tsm configuration set -k backgrounder.enable_parallel_adsync -v
true

tsm pending-changes apply
```

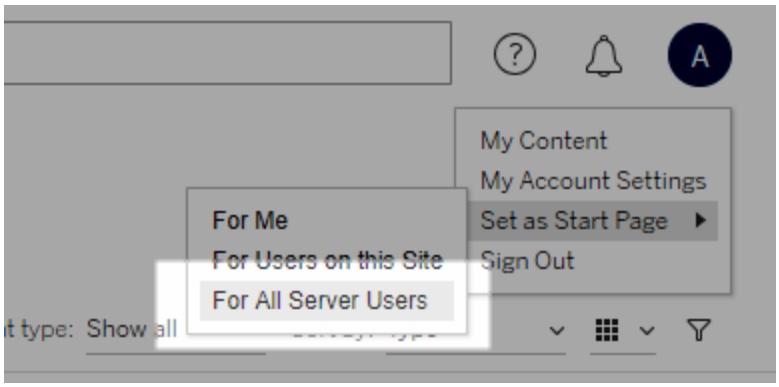
Set the Default Start Page for All Users

In a new deployment, when users sign in to the Tableau Server web authoring environment, they are taken to the Home screen, which displays a role-based welcome banner. Home also displays recent views, favorites, and the site's most-viewed content. As the server administrator, you can change users' default landing page at the server and site level. For example, you can show all workbooks, and when the user signs in, they see the workbooks they have access to.

To set the default start page for all users

1. Display the page or filtered view you want to be the default page users see when they sign in to the site.
2. Select your profile icon in the upper right area of the page.
3. To set the start page for:
    - All users on a site: click **Set as Start Page** and **For Users on this Site**.

- All users on the server: click **Set as Start Page** and **For All Server Users**.



User-set start pages and hierarchy

Users can set their own start pages from their profile icon, and can reset their start pages in their account settings (for information, search for “Access Your Profile and Account Settings” in the online Tableau Server Help for your operating system).

If a user sets their own start page, it will override any start page set by a server administrator. The next time that user signs in, they will land on the start page they've set. If a server administrator sets start pages for both a server and a site, users will default to the start page set for themselves (if any), then the site start page, then the server start page. If neither a user or an administrator has set a start page, users will default to Home.

Access Sites from Connected Clients

By default, Tableau Server allows users to access a site directly from a Tableau *client*. It allows this access after the user provides credentials the first time they sign-in from the client. A client in this case is a Tableau application or service that can exchange information with Tableau Server. Examples of Tableau clients include Tableau Desktop, Tableau Prep Builder, and Tableau Mobile.

Tableau Server establishes a *connected client* by creating a secure refresh token that uniquely identifies a user when the user signs in from the client.

Disable Automatic Client Authentication

After Tableau connected clients (e.g., Tableau Desktop, Tableau Mobile, Tableau Prep Builder, etc.) and personal access tokens (PATs) successfully sign in to Tableau Server, they are automatically authenticated in the future. Both connected client sessions and PATs are managed by refresh tokens.

By default, refresh tokens reset after a year. If a refresh token has not been used in 14 days, then it will expire. As a server administrator, you can change these values by setting the `refresh_token.absolute_expiry_in_seconds` and `refresh_token.idle_expiry_in_seconds` options. See tsm configuration set Options.

As a Tableau Server administrator you can also disable automatic authentication for connected clients. In this case, session expiration is solely governed by Tableau Server session behavior, which manages web authoring sessions. See 9. Verify session lifetime configuration. Web authoring sessions are not considered a "connected client," and they do not use refresh tokens.

To immediately disconnect connected clients from Tableau Server and require users to sign in every time they connect:

1. Sign in to Tableau Server as a server administrator.
2. In the site menu, click **Manage All Sites**, and then click **Settings** > **General**.
3. Under **Connected Clients**, clear the **Let clients to automatically connect to Tableau Server** check box.
4. Click **Save** button at the top or bottom of the page.

Note: This setting described above only applies to connected clients and does not affect the creation and redemption of PATs.

Remove Unneeded Files

As a best practice, you should regularly monitor disk space usage on your server. If the Tableau Server computer runs low on disk space, the impact to can be significant, including ultimately causing a failure. If you determine that space is getting low, you can archive any you want to save, and purge unneeded files, freeing up space for Tableau.

Monitoring disk space usage

There are several things you can do to monitor disk space usage:

- Notifications: You can configure Tableau Server to send notifications when disk space reaches predetermined levels. For more information, see Configure Server Event Notification.
- Administrative views: You can use a pre-built administrative view to help monitor disk space usage. For more information, Server Disk Space.

Reducing disk space usage

To make more disk space available, you can take the following steps:

- Archive log files: Tableau Server generates log files when running. These can be helpful in troubleshooting issues, and when you are working with Tableau Support, but you do not need to leave them in place indefinitely. To save disk space without losing logs, you can archive them with the `tsm maintenance ziplogs` command, and then copy the ziplogs archive to a computer that is not part of the Tableau installation for safe keeping. For more information, see Log File Snapshots (Archive Logs).

- Clean up unwanted files: After archiving any logs you want to save, use the `tsm maintenance cleanup` command to remove log files older than seven days, temporary files, and optionally, rows from the `http_requests` table in the Tableau Server repository. You should run the cleanup command regularly. For more information about which files are removed, see tsm maintenance cleanup.

- Remove other files: Over time Tableau Server can generate files that do not need to be left in place. In addition to the files mentioned above, be aware of files like old backups

from previous versions. Tableau Server backup files have a `.tsbak` extension. We strongly recommend regularly backing up Tableau, and saving the backup files to a computer that is not part of the Tableau Server installation for safe keeping, but once you save the file in another location, you can delete it from the Tableau computer. For more information on backup files, including how to create them and where they are saved, see Back up Tableau Server Data.

Server Settings (General and Customization)

The following settings are available on the **General** and **Customization** pages in **Server - Settings**.

Many of these settings move from the Server Settings page to the Site Settings page when there is more than one site on the server. These are marked with "Moves to Site Settings on multi-site servers."

General

Setting	Description
Site Name and ID Moves to Site Settings on multi-site servers.	Specifies the site name seen in the user interface and the ID seen in the site URL. (If you are editing the Default site, you cannot change the ID.) You can't change the “#/site” portion of the URL (for example, http://localhost/#/site/sales). In multi-site server environments, these segments appear in the URL for sites other than the Default site.
Storage Moves to Site Settings on multi-site servers.	Select either **Server Limit** or **GB**, and for the latter enter the number of gigabytes you want as a limit for storage space for published workbooks, extracts, and other data sources. If you set a server limit and the site exceeds it, publishers will be prevented from uploading new content until the site is under the limit again. Server admin-

	istrators can track where the site is relative to its limit using the Max Storage and Storage Used columns on the Sites page.
Revision History Moves to Site Settings on multi-site servers.	Specifies the number of previous versions of workbooks, flows, and data sources that are stored on the server.
Managing Users Moves to Site Settings on multi-site servers.	Determines whether only server administrators can add and remove users and change their site roles, or whether site administrators can too. If you allow site administrators to manage users, specify how many users they can add to the site by selecting one of the following: • Server Limit adds the number of available server seat licenses. For a server with core-based licensing, there is no limit. • Site Limit lets site administrators add users up to a limit you specify. • Site Role Limit lets site administrators add users of each site role up to the license limit you specify for the site. For more information, see View Server Licenses.
Web Authoring Moves to Site Settings on multi-site servers.	Controls whether browser-based authoring is enabled for the site. When web authoring for workbooks is disabled, users can't create or edit published workbooks from the server web environment but instead must use Tableau Desktop to republish the workbook. When web authoring for flows is disabled, users can't create or edit published flows from the server web environment but instead must use Tableau Prep Builder to re-publish the flow.

	For more information, see Set a Site's Web Authoring Access and Functions in Tableau Cloud Help.
Tableau Prep Conductor Moves to Site Settings on multi-site servers.	Controls whether users with appropriate permissions can schedule and monitor flows. Tableau Prep Conductor is part of Tableau Data Management. For more information, see About Tableau Prep Conductor.
Tableau Catalog Moves to Site Settings on multi-site servers.	Turns off Catalog capabilities when Tableau Server or a Tableau Cloud site is licensed with Data Management. For more information, see Disable Catalog.
Email Settings Moves to Site Settings on multi-site servers.	Specifies the From address and message footer seen in automatic emails for alerts and subscriptions.
Workbook Performance after a Scheduled Refresh Moves to Site Settings on multi-site servers.	Pre-computes recently viewed workbooks with scheduled refreshes to open them faster. For more information, see Configure Workbook Performance after a Scheduled Refresh.
Workbook Performance Metrics Moves to Site Settings on multi-site servers.	Lets site users collect metrics on how workbooks perform, such as how quickly they load To initiate recording, users must add a parameter to the workbook's URL. For more information, see Create a Performance Recording.
Managed Keychain Clean Up Moves to Site Settings on multi-site servers.	Lets site administrators manage saved credential keychains for OAuth connections on the site. For more information, see OAuth Connections.
Automatically Suspend Extract Refresh Tasks Moves to Site Settings on multi-	To save resources, Tableau can automatically suspend extract refresh tasks for inactive workbooks. This feature applies only to refresh schedules that run weekly or more often. For more information, see Automatically

site servers.	Suspend Extract Refreshes for Inactive Workbooks in Tableau Cloud Help.
User Visibility Moves to Site Settings on multi-site servers.	Controls what user and group names are visible to other users. For more information, see Manage User Visibility in Tableau Cloud Help.
Availability of Ask Data Moves to Site Settings on multi-site servers.	Controls whether Ask Data is enabled or disabled by default for data sources. Ask Data lets users query data using conversational language and automatically see visualizations. For more information, see Automatically Build Views with Ask Data in Tableau user Help.
Availability of Explain Data Moves to Site Settings on multi-site servers.	Controls whether site users with the appropriate permissions can run Explain Data and authors can access Explain Data Settings. For more information, see Control Access to Explain Data. To learn more about Explain Data, see Discover Insights Faster with Explain Data.
Automatic Access to Metadata about Databases and Tables Moves to Site Settings on multi-site servers.	Automatically grants users certain capabilities to external assets using derived permissions. For more information, see Turn off derived permissions in Tableau Cloud Help.
Sensitive Lineage Data Moves to Site Settings on multi-site servers.	Specifies whether sensitive lineage data should be obfuscated or filtered when users don't have the appropriate permissions to related metadata. For more information, see Sensitive lineage data.
Extract Encryption at Rest Moves to Site Settings on multi-site servers.	Lets you encrypt .hyper extracts while they are stored on Tableau Server. Server administrators can enforce encryption of all extracts on their site or allow users to encrypt all extracts associated with particular published workbooks or data sources. For more information, see

	Extract Encryption at Rest.
Tableau Mobile Moves to Site Settings on multi-site servers.	• Enable offline previews Controls whether offline previews are generated for display when users access the site on Tableau Mobile. For more information, see Manage Tableau Mobile Data on Devices in the Tableau Mobile Deployment Guide. • Enable app lock Requires a biometric method or device passcode for users to open the site on Tableau Mobile. For more information, see Enable App Lock for Added Security in the Tableau Mobile Deployment Guide.
Sharing Moves to Site Settings on multi-site servers.	Allows users to share items directly with other users. When an item is shared, the recipients get a notification and the item is added to their Shared with Me page. If this is not enabled, users can only copy a link to share. For more information, see Share Web Content in Tableau user Help.
Comments Moves to Site Settings on multi-site servers.	Controls whether users can add remarks in a Comments side pane for each view and @mention other Tableau users to notify them via email. For more information, see Comment on Views in Tableau user Help.
Data-Driven Alerts Moves to Site Settings on multi-site servers.	Lets users automatically receive emails when data reaches key thresholds. For more information, see Send Data-Driven Alerts in Tableau user Help.
Tagging	Specifies the number of tags that users can add to items. The default limit is 50 tags, and the maximum is

Moves to Site Settings on multi-site servers.	200. For more information, see Use Tags.
Recommendations for Views Moves to Site Settings on multi-site servers.	Controls whether recommendations show on the site and whether the names of users who have looked at recommended items show on recommendation tooltips. **Note:** If you use Tableau Server, your administrator can disable Recommendations.
Request Access Moves to Site Settings on multi-site servers.	Lets users send access requests to content or project owners. For more information, see Let Site Users Request Access to Content in Tableau Cloud Help.
Cross-Database Joins Moves to Site Settings on multi-site servers.	Determines where the join process happens when joining data from multiple sources. For more information, see Combine Tables from Different Databases in Tableau user Help.
Metrics Content Type Moves to Site Settings on multi-site servers.	Controls whether metrics are available on the site. When enabled, users can create metrics from views and metrics appear as a content type. When disabled, metrics won't appear on the site or continue to sync; however, you can re-enable the feature to bring back previously created metrics. For more information, see "Set Up for Metrics" in Tableau Cloud Help or Tableau Server Help. **Retirement of the legacy metrics feature** Tableau's legacy metrics feature was retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2. In October 2023, Tableau retired the ability to

	embed legacy metrics in Tableau Cloud and in Tableau Server version 2023.3. With Tableau Pulse, we've developed an improved experience to track metrics and ask questions of your data. For more information, see Create Metrics with Tableau Pulse to learn about the new experience and Create and Troubleshoot Metrics (Retired) for the retired feature.
Site Time Zone for Extracts Moves to Site Settings on multi-site servers.	The default time zone for extract-based data sources in a site is Coordinated Universal Time (UTC). Site administrators can set a different time zone. For more information, see Set the Site Time Zone for Extracts in Tableau Cloud Help.
Run Now On multi-site servers this appears on both Server Settings (at bottom of page) and Site Settings.	Server administrators can use this setting to allow or block users from manually running extract refreshes, flow runs, and subscriptions. This setting can be applied at the server level to include all the sites on Tableau Server or applied at the site level to include only specific sites. • By default, this option is set to allow users to run jobs manually. Clear the check box to prevent users from running jobs manually. • This applies only to jobs that are manually initiated by a user from the web interface, REST API calls, or tabcmd commands. Jobs initiated from scheduled tasks will continue to run at the schedule time and will not be affected. • Select **Run Now** to allow users to change the connection type (Live/Extract) of data sources on the web.
Manage Notifications Moves to Site Settings on multi-	Controls how site users can receive notifications for events such as extract jobs, flow runs, or when another

site servers.	user shares content with them or mentions them in a comment. Notifications can be seen in their Tableau site via the notification center or sent by email. When a notification is enabled, users can configure their notification preferences on their Account Settings page.
Flow Subscriptions Moves to Site Settings on multi-site servers.	Controls whether flow owners can schedule and send emails with flow output data to themselves and others. When you allow flow subscriptions, you can control whether flow output data is included in the subscription email and whether flow output files are attached to the email. For more information, see Notify Users of Successful Flow Runs.
Web Page Objects Moves to Site Settings on multi-site servers.	Controls whether these dashboard objects can display target URLs. For more information, see Security for Web Page objects in Tableau user Help.
Personal Space Moves to Site Settings on multi-site servers.	Allows Creator and Explorer site users to create and save content to a private Personal Space. When Personal Space is turned on, you can set user storage limits. For more information, see Create and Edit Private Content in Personal Space.
Collections Moves to Site Settings on multi-site servers.	Controls whether collections are available. When you turn on collections, users can create collections to organize content and browse collections made available by other users. For more information, see Organize Items in a Collection.
OAuth Clients Registry	For a subset of connectors, you can register a custom OAuth client for the site to override an OAuth client that has been configured for the server. By registering a custom OAuth client, you enable new and existing connections to use the site-level OAuth client instead of the server-wide OAuth client. For more information, see Configure a custom OAuth for a site.

View Acceleration	Controls whether Creator and Explorer site users can accelerate the views in their workbooks for faster loading times. When you allow view acceleration, you can set a maximum number of views to be accelerated, and you can choose to automatically suspend acceleration for views that repeatedly fail the acceleration task. For more information, see View Acceleration.
Start Page	Links to the server's current default start page for all users. For more information on how to change the default start page, see Set the Default Start Page for All Users. Individual users will be able to override this setting (search for "Access Your Profile and Account Settings" in the Tableau Server Help for details).
Guest Access	Allows users to view and interact with embedded views without having to sign in to a Tableau Server account. Permission can be assigned to the Guest User account to control the interactivity allowed for each view. This option is only available if you have a core-based server license.
Embedded Credentials in Content	• Let publishers embed credentials in a data source, flow, or workbook Allows publishers to attach passwords to published workbooks or flows that will automatically authenticate web users to connect to data sources. The passwords are attached to workbooks or flow inputs and are only accessible on the server. For example, when a workbook is opened in Tableau Desktop, users still need to enter a user name and password to connect to the data source. When this setting is turned off, all existing embedded passwords are saved but

	are not used for authentication. If you turn the setting back on, users don't have to re-embed the passwords. • Let publishers schedule flow runs and extract refreshes Allows publishers to assign workbooks or flows to schedules. This option is only available if **Let publishers embed credentials in a data source, flow, or workbook** is enabled. When this setting is enabled, Tableau Desktop users will see scheduling options in the Publish dialog box.
Sign In Customization	Add a custom note to the server sign in page. The Sign In setting lets you add text. You can optionally add a URL to make the text a link. This note will also appear if a user receives a sign in error. Custom notes do not display on Tableau Mobile. If Tableau Server is configured with identity pools, the Sign In Customization note appears on both the Tableau Server landing page below all sign-in options and on the page where your initial pool (TSM configured) users enter their username and password. To set a custom note, sign in to a site on Tableau Server. On the left-side navigation pane, select **Manage all sites** from the drop-down site list. Select **Settings** and add a message to **Sign In Customization**. For more information, see Customize Your Server.
Recommendations Training Schedule	Recommendations take two forms: recommendations

	for data sources and tables (for Tableau Desktop) and recommendations for views (for Tableau Server). Recommendations are based on the popularity of content and on content used by other users determined to be similar to the current user. The training schedule controls how often the server checks for new content and new usage information to keep the recommendations up to date. New content includes new and updated data sources and workbooks. New usage information includes information such as "Laura Rodriquez used the Food Catering data source" and "Henry Wilson accessed the Monthly Sales view." If you notice an impact on server performance, schedule this process to occur at a time when the server load is low. To track performance impact, look for the "Recommendations Trainer" or "View Recommendations Trainer" tasks in the Background Tasks for Non Extracts view. If you want to disable Recommendations, change the tsm configuration set option "recommendations.enabled" to false. If you want to disable only recommendations for views, change the option “recommendations.vizrecs.enabled” to false.
Connected Clients	Controls whether mobile users must sign in and provide their credentials every time they connect to Tableau Server, or if users can connect with their devices to Tableau Server without providing credentials after they authenticate their device successfully the first time. For more information, see Disable Automatic Client

	Authentication.
Language and Locale	Controls the language used for the server user interface and the locale used for views. Individual users can override this setting on their Account Settings page. Also, web browser settings are evaluated first to determine which language and locale should be used. For more information, see Language and Locale for Tableau Server.
Saved Credentials for Data Sources	• Let users save passwords for data sources Allows users to choose "Remember my password" and save data source passwords across multiple visits, browsers, and devices. (By default, users can choose to "Remember my password until I sign out," which lets them save their password only for a single browser session.) As an administrator, you can clear all saved passwords at any time. In addition, users can clear their own saved passwords. • Let users save OAuth access tokens for data sources Allows users to store access tokens with their user preferences. Access tokens are provided by cloud data sources that support OAuth connections, and they are used instead of user names and passwords to grant access to the data.
Linked Tasks On multi-site servers this	Server administrators can use this setting to enable users to schedule flow tasks to run one after the other.

appears on both Server Settings and Site Settings.	They can also enable users to trigger the scheduled flow tasks to run using **Run Now**. This setting can be applied at the server level to include all the sites on Tableau Server. The setting can be disabled at the site level to include only specific sites. If the setting is turned off after linked tasks are scheduled, any tasks that are running will complete and the scheduled linked tasks are hidden and no longer show on the **Scheduled Tasks** tab. For more information, see Schedule linked tasks.
Flow Parameters On multi-site servers this appears on both Server Settings and Site Settings.	Enables users to schedule and run flows that include parameters. Administrators can also enable flow parameters to accept any value. If this option is enabled, any flow user can enter any value in a parameter, potentially exposing data that the user should not have access to. Parameters can be entered in an input step for file name and path, table name, or when using custom SQL queries, in an output step for file name and path and table name, and in any step type for filters or calculated values. Flow parameter settings can be applied at the server level to include all sites on Tableau Server. The settings can be disabled at the site level to include only specific sites. For more information about using parameters, see

	Create and Use Parameters in Flows in the Tableau Prep help.
Active Directory Synchronization Only appears when server is configurefd for AD identity store.	Controls the synchronization of all Active Directory groups in Tableau Server based on a schedule that you specify after you select the option **Synchronize Active Directory groups on a regular schedule**. For more information, see Synchronize External Directory Groups on the Server.
Reset to Default Settings	Returns any server settings described here that have been changed since setup back to their original state.
Assertions for Group Membership On multi-site servers this appears on both Server Settings and Site Settings pages.	Enables local group membership to be controlled and managed by the IdP or through the connected app by dynamically asserting group membership when a user authenticates to Tableau Server. Requires additional configuration in the SAML assertion or JSON web token (JWT). For more information, see Dynamic group membership using assertions. **Note:** This server-wide setting must be enabled to allow the site-level setting to be enabled.
Group Sets Moves to site Settings on multi-site servers.	Enables the **Group Sets** page and the ability to create group sets. Group sets can be used by certain users (server admins, site admins, project owners, and content owners) to apply permission rules that require users to be members of all groups in the group set to access content whose permissions are dependent on the group set. For more information, see Work with Group Sets

Customization

Setting	**Description**
Welcome Banner	Add a custom message to the welcome banner

	on the home page for all server users to see. The custom message can contain up to 240 characters of text and hyperlinks as well as one paragraph break. Administrators can also disable the default Tableau welcome banner for the server. For more information, see Customize Your Server.

Mobile

Setting	Description
Tableau Mobile	
App Lock	Requires a biometric method or device passcode for users to open this site on Tableau Mobile. For more information, see Enable App Lock for Added Security in the Tableau Mobile Deployment Guide.
Offline Previews	Controls whether offline previews are generated for display when users access the site on Tableau Mobile. For more information, see Manage Tableau Mobile Data on Devices in the Tableau Mobile Deployment Guide.
Mobile Security Policies Added in version 2023.1.0	Some security policies are enabled automatically and cannot be disabled. Mobile security policies are not available for MAM versions of Tableau Mobile. **Note:** Mobile Security policies configured at the site level override Server-level Mobile Security policies.
Jailbreak Detection	Controls whether a Tableau Mobile app user with a device that has been "jailbroken" or "rooted" is allowed to access content on Tableau, and what level of response occurs when a jailbroken or rooted device is detected. For more information, see Tableau Mobile App Security Settings.

Malware Detection (Android only)	Controls whether malware detection is enabled for mobile devices, and what level of response occurs when malware is detected. For more information, see Tableau Mobile App Security Settings.
Maximum Days Offline Without Policy Refresh	Controls whether there is a maximum number of days a mobile device can be offline and still use the app. For more information, see Tableau Mobile App Security Settings.
Prevent Debugging	Controls whether debuggers are prevented on mobile devices. For more information, see Tableau Mobile App Security Settings.
Screen Sharing and Screenshots (Android only)	Controls whether a Tableau Mobile user is able to take screenshots or use screen sharing while in the app. For more information, see Tableau Mobile App Security Settings.

Stop or Restart the Tableau Server Computer

As a best practice, you should *always* stop Tableau Server before you stop or restart the computer it is running on. This is true whether you are running Tableau on virtual machines (VMs), or on dedicated hardware. You should never turn off a computer without first stopping Tableau Server. Shutting down the computer while Tableau is running can cause problems restarting Tableau Server, and may result in unexpected results.

To be safe, follow these steps, whether you have a scripted process to shut down your systems, or manually shut down your computers:

1. Stop Tableau Server.

   You can do this either from the command line, using the `tsm stop` command, or from the TSM Web UI, by clicking **Tableau Server is running**, and selecting **Stop Tableau Server**.

> **Note:** Some TSM processes will continue to run, even after you stop Tableau Server. This is normal, and you can go ahead and stop your computer. The running services are designed to shut themselves off when the computer is stopped.

2. Once Tableau is stopped, stop your computer.
3. When you are ready, restart your computer. This might be after you have completed planned maintenance, or after leaving the computer off for some extended period of down time.
4. Start Tableau Server.

   You can do this either from the command line, using the `tsm start` command, or from the TSM Web UI, by clicking **Tableau Server is stopped**, and selecting **Start Tableau Server**.

   TSM will start automatically when the computer starts, so you can run tsm commands even though Tableau Server is stopped.

tsm Command Line Reference

The topics in this section include reference content for Tableau Services Manager (TSM) command line interface (CLI) to support Tableau Server.

TSM is used to manage installation and configuration of Tableau Server. To learn more about TSM, see Tableau Services Manager Overview.

You can automate the installation and configuration tasks supported by the TSM CLI using the TSM API. To learn more about the prerelease (Alpha) TSM API, see Tableau Services Manager API.

> Looking for tsm commands for Tableau Server on Windows? See tsm Commands.

Using the tsm CLI

You can run tsm commands on the initial node (the node where TSM is installed), or on any additional node in the cluster.

To run tsm commands, you need to open a command prompt.

1. Open a command prompt with an account that is a member of the `tsmadmin` group on a node in the cluster.
2. Run the command you want. If you are running the command from a node other than the initial node, include the `-s` option to specify the URL of the initial node by name (not IP address), and include the TSM port, 8850.

   To see the version of TSM and Tableau Server from the initial node:

   ```
   tsm version
   ```

   To see the version of TSM and Tableau Server from an additional node:

   ```
   tsm version -s https://<inital_node_name>:8850
   ```

   For example:

   ```
   tsm version -s https://myTableauHost:8850
   ```

Authenticating with tsm CLI

Beginning in the 2019.2 release of Tableau Server, running tsm commands will not require you to enter a password if the following are true:

- The account you are running commands with is a member of the TSM-authorized group, by default, the `tsmadmin` group. The Tableau unprivileged user (by default, the `tableau` user) and root account may also run TSM commands.
- You are running commands locally on the Tableau Server that is running the Tableau Server Administration Controller service. By default, the Tableau Server Administration Controller service is installed and configured on the initial node in a distributed deployment.

Logging into tsm CLI locally

If you are running tsm commands on the local computer with user account that is a member of a TSM-authorized group, then you will not need to specify a password. In this case, just run the command, for example:

```
tsm version
```

Logging into tsm CLI remotely

If you are running TSM commands from a node in a cluster where the Tableau Server Administration Controller service is not running, then you must authenticate a session with the Tableau Server Administration Controller service on the remote computer before you can run commands. For example, run the following command:

```
tsm login -s <server_name> -u <account_name>
```

Where `<server_name>` is the name of the node where the Tableau Server Administration Controller service is running and `<account_name>` is an account that is a member of a TSM-authorized group.

After running this command, you will be prompted for a password. After the account has been authenticated, you can run TSM commands.

As a security best practice, do not expose the TSM port (by default, `8850`) to the internet.

Viewing and adding accounts to the TSM-authorized group

The TSM-authorized group is created during server installation. By default, the TSM-authorized group that is named `tsmadmin`. If you created an alternative TSM-authorized group during installation, then substitute your group name for `tsmadmin` in the following code examples.

To view the user accounts in the `tsmadmin` group, run the following command:

```
grep tsmadmin /etc/group
```

To add a user account to the `tsmadmin` group:

```
sudo usermod -G tsmadmin -a <username>
```

Scripting and automating with tsm CLI

To run automation on a Tableau Server without a password in the script file, run the script on the initial node and with an account in the proper TSM-authorized group. See the "Authenticating" section above for more details.

Viewing help content in the shell

To view minimal help content from a command line, use the `tsm help` category.

Synopsis

```
tsm help [category] [command]
```

Commands

`tsm help`

Help for all tsm commands

`tsm help <category>`

Show help for a specific command category. For example, `tsm help authentication`.

`tsm help <category> <command>`

Show help for a specific command. For example, `tsm help authentication open-id`.

`tsm help commands`

List all top-level commands or categories.

Categories

tsm authentication

You can use the `tsm authentication` commands to enable, disable, and configure user authentication options for Tableau Server.

- identity-migration
- kerberos
  - configure
  - disable
  - enable
- legacy-identity-mode
  - enable
  - disable
- list
- mutual-ssl
  - configure
  - disable
  - enable
- openid
  - configure
  - disable
  - enable
  - get-redirect-url
  - map-claims
- pat-impersonation
  - disable
  - enable
- saml
  - configure
  - disable
  - enable

    - export-metadata
    - map-assertions
- sitesaml
    - disable
    - enable
- sspi
    - disable
    - enable
- trusted

tsm authentication identity-migration configure

Change the default job settings for the identity migration. For more information, see Manage the Identity Migration.

Synopsis

```
tsm authentication identity-migration configure –job-run-time

tsm authentication identity-migration configure –rate
```

Options

`-j,--job-run-time <number>`

Optional.

Determines the longest allowable time, in minutes, the scheduled migration job can run. Default value is 120 minutes.

`-r,--rate <number>`

Optional.

Determines the number of requests during a migration job that can run per second. Default value is 5 requests per second.

tsm authentication kerberos <commands>

Enable, disable, and configure Kerberos user authentication on Tableau Server. See Configure Kerberos.

Synopsis

```
tsm authentication kerberos configure --keytab-file <keytab_
file.keytab> [global options]

tsm authentication kerberos enable [global options]

tsm authentication kerberos disable [global options]
```

Options for kerberos configure

`-kt, --keytab-file <keytab_file.keytab>`

Required.

Specifies the service .keytab file used for requests to the KDC.

tsm authentication legacy-identity-mode <commands>

Enable or disable legacy identity store mode that might be required during the identity migration. Review the Unable to restore backup section to determine when to use this command.

For more information, see About the Identity Migration.

Synopsis

```
tsm authentication legacy-identity-mode enable

tsm authentication legacy-identity-mode disable
```

tsm authentication list

List the server's existing authentication-related configuration settings.

Synopsis

```
tsm authentication list [--verbose][global options]
```

Options

```
v, --verbose
```

Optional.

Show all configured parameters.

tsm authentication mutual-ssl <commands>

Enable, disable, and configure mutual SSL for user authentication on Tableau Server. To learn more about mutual SSL, see Configure Mutual SSL Authentication.

Before you enable mutual SSL, you must enable and configure SSL for external communication. For information, see Configure SSL for External HTTP Traffic to and from Tableau Server.

Synopsis

```
tsm authentication mutual-ssl configure [options] [global options]

tsm authentication mutual-ssl disable [global options]

tsm authentication mutual-ssl enable [global options]
```

Options

```
-cf, --ca-cert <certificate-file.crt>
```

Optional.

Specifies the location and file name for the certificate file. The file must be a valid, trusted certificate from a Certificate Authority (for example, Verisign).

```
-fb, --fallback-to-basic <true | false>
```

Optional.

Specifies whether Tableau Server should accept user name and password for authentication if SSL authentication fails.

Default value is false, to indicate that when configured for mutual SSL, Tableau Server does not allow a connection when SSL authentication fails. However, Tableau Serer accepts username and password authentication from REST API clients, even if this option is set to `false`.

`-m, --user-name-mapping <upn | ldap | cn>`

Optional.

Specifies the user name syntax (UPN, LDAP or CN) to retrieve from identity store or directory. The syntax must match the format for Subject or Subject Alternative Name on the user certificate.

`-rf, --revocation-file <revoke-file.pem>`

Optional.

Specifies the location and file name for the certificate revocation list file. This file can be a .pem or .der file.

tsm authentication openid <commands>

Enable, disable, and configure OpenID Connect (OIDC) user authentication on Tableau Server.

Synopsis

```
tsm authentication openid configure [options] [global options]

tsm authentication openid disable [global options]

tsm authentication openid enable [global options]

tsm authentication openid get-redirect-url [global options]
```

```
tsm authentication openid map-claims [options] [global options]
```

Options for openid configure

> **Note:** Options must be set during initial configure, or during reconfigure. Individual options cannot be set separately.

`-a, --client-authentication <string>`

Required.

Specifies custom client authentication method for OpenID Connect.

To configure Tableau Server to use the Salesforce IdP, set this value to `client_secret_post`.

`-cs, --client-secret <string>`

Required.

Specifies the provider client secret. This is a token that is used by Tableau to verify the authenticity of the response from the IdP. This value is a secret and should be kept securely.

`-cu, --config-url <url>`

Required.

Specifies location of the provider configuration discovery document that contains OpenID provider metadata. If the provider hosts a public JSON file, use --config-url. Otherwise, specify a path on the local computer and file name using --metadata-file instead.

`-mf, --metadata-file <file-path>, --config-file <config-file.json>`

Optional.

Specifies the local path to the static OIDC discovery JSON document.

`-i, --client-id <client-id>`

Required.

Specifies the provider client ID that your IdP has assigned to your application.

`-id, --ignore-domain <true | false>`

Optional. Default: `false`

Set this to `true` if the following are true:

- You are using email addresses as usernames in Tableau Server
- You have provisioned users in the IdP with multiple domain names
- You want to ignore the domain name portion of the `email` claim from the IdP

Before you proceed, review the user names that will be used as a result of setting this option to `true`. User name conflicts may occur. In the case of a user name conflict, the risk of information disclosure is high. See Requirements for Using OpenID Connect.

`-if, --iframed-idp-enabled <true | false>`

Optional. Default: `false`

Specifies if the IdP is allowed inside of an iFrame. The IdP must disable clickjack protection to allow iFrame presentation.

`-ij, --ignore-jwk <true | false>`

Optional. Default: `false`

Set this to `true` if your IdP does not support JWK validation. In this case, we recommend authenticating communication with your IdP using mutual TLS or another network layer security protocol.

`-r, --return-url <return-url>`

The URL of your server. This is typically is the public name of your server, such as `"http://example.tableau.com"`.

`-sn, --custom-scope-name <string>`

Optional.

Specifies a custom scope user-related value that you can use to query the IdP. See Requirements for Using OpenID Connect.

Options for openid map-claims

Use these options to change the default OIDC claims Tableau Server will use when communicating with your IdP. See Requirements for Using OpenID Connect.

`-i, --id <string>`

Optional. Default: `sub`

Change this value if your IdP does not use the `sub`claim to uniquely identify users in the ID token. The IdP claim that you specify should contain a single, unique string.

`-un, --user-name <string>`

Optional. Default: `email`

Change this value to the IdP claim that your organization will use to match user names as stored in Tableau Server.

tsm authentication pat-impersonation <commands>

Enable and disable personal access token impersonation for administrators on Tableau Server.

For more information about personal access token impersonation, see Personal Access Tokens.

Synopsis

```
tsm authentication pat-impersonation disable [global options]

tsm authentication pat-impersonation enable [global options]
```

To view whether personal access token impersonation is enabled, run the following command:

```
tsm authentication list
```

tsm authentication saml <commands>

Configure Tableau Server to support single-sign on using the SAML 2.0 standard, enable or disable SAML for a site, map assertion attribute names between Tableau Server and the identity provider (IdP).

Available commands

```
tsm authentication saml configure [options] [global options]

tsm authentication saml disable [options] [global options]

tsm authentication saml enable [options] [global options]

tsm authentication saml export-metadata [options] [global options]

tsm authentication saml map-assertions [options]
```

tsm authentication saml configure

Configure the SAML settings for the server. Specify the SAML certificate and metadata files, provide additional required information, set additional options.

If you are configuring SAML for the first time or have previously disabled it, you must run this command with `tsm authentication saml enable`. For more information, see Configure Server-Wide SAML.

Synopsis

```
tsm authentication saml configure [options] [global options]
```

Options

`-e, --idp-entity-id <id>`

Required for initial SAML configuration; otherwise optional. IdP entity ID value.

Typically this is the same as the Tableau Server return URL (specified in the `--idp-return-url` parameter). The entity ID that you enter is used as a base for generating site-specific entity IDs. For example, if you enter the following:

http://tableau-server

A site configured for SAML might display the following entity ID:

http://tableau-server/samlservice/public/sp/metadata?alias=48957410-9396-430a-967c-75bdb6e002a0

To find a site's entity ID, go to the site's **Settings** page, and select the **Authentication** tab. When SAML is enabled, the entity ID is shown under the first step for configuring site-specific SAML, exporting metadata.

`-r, --idp-return-url <idp-return-url>`

Required for initial SAML configuration; otherwise optional. The SAML return URL configured in the IdP. This is typically the Tableau Server external URL; for example, https://tableau-server.

Notes

- http://localhost does not work for an external server.
- Adding a trailing slash to the URL (https://tableau-server/) is not supported.

`-i, --idp-metadata <idp-metadata.xml>`

Required for initial SAML configuration; otherwise optional. Provide the location and name of the XML metadata file you exported from the IdP's settings.

For example, `/var/opt/tableau/tableau_server/data/saml/<metadata-file.xml>`

`-cf, --cert-file <certificate.crt>`

Required for initial SAML configuration; otherwise optional. The location and file name for the x509 certificate file for SAML. For requirements for the certificate file, see SAML Requirements.

For example, `/var/opt/tableau/tableau_server/data/saml/<file.crt>`

`-kf, --key-file <certificate.key>`

Required for initial SAML configuration; otherwise optional. Location and name of the key file that goes along with certificate.

For example, `/var/opt/tableau/tableau_server/data/saml/<file.key>`

`-a, --max-auth-age <max-auth-age>`

Optional. Default value is -1 (beginning in Tableau Server version 2020.4.15, 2021.1.12, 2021.2.9,2021.3.8, 20214.4, 2022 and later). Previous default value was 7200 (2 hours).

The maximum number of seconds allowed between a user's authentication and processing of the AuthNResponse message.

We recommend you keep this default value, which disables the check for maximum authentication age. If this value is different from your IdP, users signing in to Tableau Server can see a sign-in error. For more information about this error, see the Intermittent Error "Unable to Sign In" with SAML SSO on Tableau Server article in the Tableau Knowledge Base.

`-d, --desktop-access <enable | disable>`

Optional. Default value is enable.

This option only applies to server-wide SAML. Use SAML to sign in to the server from Tableau Desktop. If single sign-on from Tableau client applications does not work with your IdP, you can set this to `disable`.

`-m, --mobile-access <enable | disable>`

Optional. Default value is enable.

Allow using SAML to sign in from older versions of Tableau Mobile app. Devices running Tableau Mobile app version 19.225.1731 and higher ignore this option. To disable devices running Tableau Mobile app version 19.225.1731 and higher, disable SAML as a client login option on Tableau Server.

`-so, --signout <enable | disable>`

Optional. Enabled by default.

Enable or disable SAML sign out for Tableau Server.

`-su, --signout-url <url>`

Optional. Enter the URL to redirect to after users sign out of the server. By default this is the Tableau Server sign-in page. You can specify an absolute or a relative URL.

Example

```
tsm authentication saml configure --idp-entity-id https://tableau-
server --idp-metadata /var/opt/tableau/tableau_server-
/data/saml/<metadata.xml> --idp-return-url https://tableau-server --
cert-file /var/opt/tableau/tableau_server/data/saml/<file.crt> --
key-file /var/opt/tableau/tableau_server/data/saml/<file.key>
```

tsm authentication saml enable and saml disable

Enable or disable server-wide SAML authentication. In this context, all sites and users that you enable for SAML go through a single identity provider.

Synopsis

```
tsm authentication saml enable [global options]

tsm authentication saml disable [global options]
```

tsm authentication saml export-metadata

Export the Tableau Server .xml metadata file that you will use to configure the SAML IdP.

Synopsis

```
tsm authentication saml export-metadata [options] [global options]
```

Options

`-f, --file [/path/to/file.xml]`

Optional.

Specifies the location and file name in which the metadata will be written. If you don't include this option, `export-metadata` saves the file to the current directory, and names it `samlmetadata.xml`.

`-o, --overwrite`

Optional.

Overwrites an existing file of the same name specified in `-f`, or of the default name if `-f` is not included. If a file specified in `-f` exists, and `-o` is not included, the command does not overwrite the existing file.

tsm authentication saml map-assertions

Maps attributes between the IdP and Tableau Server. Provide the name that the IdP uses for the attribute specified in each argument.

Synopsis

```
tsm authentication saml map-assertions --user-name <user-name>
[global options]
```

Options

`-r, --user-name <user-name-attribute>`

Optional. The attribute in which the IdP stores the user name. On Tableau Server, this is the same as the display name.

`-e, --email <email-name-attribute>`

Do not use. This option is not supported by Tableau.

`-o, --domain <domain-name-attribute>`

Optional. The attribute in which the IdP stores the domain name. Use this option if you add users from a domain that's different from the domain of the Tableau Server computer. For more information, see When running multiple domains.

`-d --display-name <display-name-attribute>`

Do not use. This option is not supported by Tableau.

Example for saml map-assertions

`tsm authentication saml map-assertions --user-name=<sAMAccountName> --domain=<FQDM>` or `tsm authentication saml map-assertions --user-name-e=jnguyen --domain=example.myco.com`

tsm authentication sitesaml enable and sitesaml disable

Set the server to allow or disallow SAML authentication at the site level. Enabling site-specific SAML gives you access to the **Settings** > **Authentication** tab in the Tableau Server web UI. The **Authentication** tab contains the site-specific SAML configuration settings.

Use the `sitesaml enable` command with `saml configure` if you haven't yet configured the server to allow site-specific SAML. For more information, see Configure Site-Specific SAML.

Synopsis

```
tsm authentication sitesaml enable [global options]

tsm authentication sitesaml disable [global options]
```

tsm authentication sspi <commands>

This command will only work on Tableau Server on Windows. If you attempt to enable SSPI on Tableau Server on Linux, an error will be returned.

Enable or disable automatic sign-in using Microsoft SSPI.

If you use Active Directory for authentication, you can optionally enable automatic logon, which uses Microsoft SSPI to automatically sign in your users based on their Windows user-name and password. This creates an experience similar to single sign-on (SSO).

Do not enable SSPI if you plan to configure Tableau Server for SAML, trusted authentication, a load balancer, or for a proxy server. SSPI is not supported in these scenarios.

Synopsis

```
tsm authentication sspi disable [global options]

tsm authentication sspi enable [global options]
```

As with all authentication commands, you must run `tsm pending-changes apply` after running this command.

tsm authentication trusted <commands>

Configure trusted authentication (trusted tickets) for user authentication on Tableau Server.

Synopsis

```
tsm authentication trusted configure [options] [global options]
```

Options

`-th, --hosts <string>`

Optional.

Specifies the trusted host names (or IPv4 addresses) of the web servers that will be hosting pages with Tableau content.

For multiple values, enter the names in a comma-separated list where each value is encapsulated in double-quotes.

For example:

```
tsm authentication trusted configure -th "192.168.1.101",
"192.168.1.102", "192.168.1.103"
```

or

```
tsm authentication trusted configure -th "webserv1", "web-
serv2", "webserv3"
```

`-t, --token-length <integer>`

Optional.

Determines the number of characters in each trusted ticket. The default setting of 24 characters provides 144 bits of randomness. The value can be set to any integer between 9 and 255, inclusive.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm configuration

You can use the `tsm configuration` commands to get, set, and update configuration key values.

"Unknown key" responses

Certain configuration keys will return an "Unknown key" response when you attempt to get their current value, or set a new value. If this happens, verify that you have the key spelled correctly, including proper capitalization. To change the value, use the `--force-keys` option on the `tsm configuration set` command. For a list of configuration keys you can change, see tsm configuration set Options.

"Null" value responses

Certain configuration keys have a specific default value but will return a "Null" response when you attempt to get their current value. These keys use a default that is derived from the Tableau Server code. If a key is listed as having a specific default in tsm configuration set Options and the `tsm configuration get` command returns "Null" for the current value, the default value is determined by code running Tableau Server. You can set the key value using `tsm configuration set`, but this is not necessary unless you want to change the value.

- tsm configuration get
- tsm configuration list-dynamic-keys
- tsm configuration set

tsm configuration get

View the current server configuration and topology.

Synopsis

```
tsm configuration get --key <config.key> [global options]
```

Option

`-k, --key`

Required.

Get the current value of the specified configuration key.

tsm configuration list-dynamic-keys

View all the configuration keys that can be configured dynamically (without restarting Tableau Server).

Synopsis

```
tsm configuration list-dynamic-keys [global options]
```

tsm configuration set

Set or import server configuration or topology.

Quotes around the `<config.key>` and the `<config_value>` are optional unless there are spaces, in which case you must use quotes around the key or value.

Note: After setting a configuration key value you must apply the pending configuration changes using `tsm pending-changes apply`. Until you do, the new value will not be used by Tableau or show up in the results of a `tsm configuration get` command. You can view pending changes using `tsm pending-changes list`. For more information, see tsm pending-changes.

Synopsis

```
tsm configuration set --key <config.key> --value <config_value>
[global options]
```

Options

`-k,--key <config.key>`

Required.

Configuration key.

`-v,--value <config_value>`

Required. Beginning in the March maintenance releases (versions 2021.2.10, 2021.3.9, 2021.4.5), if you do not include this option, you will be prompted for the value.

Configuration value.

`-d`

Optional.

Reset the configuration value to its default.

`-frc, --force-keys`

Optional.

Force a key to be added to configuration even if it did not previously exist.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm configuration set Options

Below is a list of configuration options or keys that you can set with the `tsm configuration set` command. In many cases you can find out the current value of a configuration key with the `tsm configuration get` command.

This list is not intended to be an exhaustive list of Tableau Server configuration settings. It represents a subset of configuration keys that can be set by server administrators. Finally, some keys used internally by Tableau Server do not appear in this list.

Note: Configuration keys are case-sensitive.

Using the tsm CLI

You can run tsm commands on the initial node (the node where TSM is installed), or on any additional node in the cluster.

To run tsm commands, you need to open a command prompt.

1. Open a command prompt with an account that is a member of the `tsmadmin` group on a node in the cluster.
2. Run the command you want. If you are running the command from a node other than the initial node, include the `-s` option to specify the URL of the initial node by name (not IP address), and include the TSM port, 8850.

   To see the version of TSM and Tableau Server from the initial node:

   ```
   tsm version
   ```

   To see the version of TSM and Tableau Server from an additional node:

   ```
   tsm version -s https://<inital_node_name>:8850
   ```

For example:

```
tsm version -s https://myTableauHost:8850
```

Basic Use of tsm configuration keys

Setting a configuration key

```
tsm configuration set -k <config.key> -v <config_value>
```

In some cases, you must include the `--force-keys` option to set a configuration value for a key that has not been set before. For more information, see "Unknown key" responses.

After setting a configuration key value you must apply the pending configuration changes using `tsm pending-changes apply`. Until you do, the new value will not be used by Tableau or show up in the results of a `tsm configuration get` command. You can view pending changes using `tsm pending-changes list`. For more information, see tsm pending-changes.

Resetting a configuration key to default

To reset a configuration key back to its default value, use the `-d` option:

```
tsm configuration set -k <config.key> -d
```

Viewing the current value of a configuration key

To see what a configuration key is currently set to, use the `configuration get` command:

```
tsm configuration get -k <config.key>
```

There are two special cases that will not return a useful current value for a key:

- In certain cases you cannot get a configuration value for a key that has not been explicitly set. Instead the `tsm configuration get` command will return an "Unknown key" response. For more information, see "Unknown key" responses.

- For certain keys with predefined default values, the `tsm configuration get` command will return a "Null" response. For more information, see "Null" value responses.

Configuration Keys

adminviews.disabled

Default value: `false`

Disables access to the Tableau Administrative views. By default, access to views is enabled (this option is set to "false").

api.server.enabled

Version: Deprecated in version 2023.1. You cannot disable the REST API in version 2023.1 and later.

Default value: `true`

Allows access to the Tableau Server REST API.

By default, this functionality is enabled. We strongly recommend that you maintain this setting. Disabling the REST API will disrupt the functionality of a broad range of Tableau features. It will not improve performance or enhance security. If you choose to disable the REST API on your Tableau Server installation, test the functionality you require carefully.

Functionality impacted by disabling the REST API includes:

- Search
- Favorites
- Collections
- Content Management Tool (CMT)
- Resource Monitoring Tool (RMT)
- Personal Spaces

auditing.enabled

Default value: `true`

Allows access to the PostgreSQL (Tableau Server's own database) historical auditing tables.

backgrounder.default_run_now_priority

Default value (integer): 0

This setting controls what priority is assigned to run now jobs, with 0 being the highest priority. Values should be specified should be in the range of 0 – 100.

backgrounder.enable_parallel_adsync

Version: Added in version 2018.3.6

Default value: `false`

Controls whether parallel processing of external directory group synchronization jobs is allowed when there are multiple backgrounders. By default a scheduled synchronization of external directory groups is handled serially, by a single backgrounder. Set this to `true` to enable parallel processing on multiple backgrounder instances.

backgrounder.externalquerycachewarmup.enabled

Version: Deprecated in version 2023.1. To improve view load times for workbooks, allow View Acceleration on your site instead.

Default value: `false`

Controls the caching of workbook query results after scheduled extract refresh tasks.

backgrounder.externalquerycachewarmup.view_threshold

Version: Deprecated in version 2023.1. To improve view load times for workbooks, allow View Acceleration on your site instead.

Default vaule: `2.0`

The threshold for caching workbook query results after scheduled extract refresh tasks. The threshold is equal to the number of views that a workbook has received in the past seven days divided by the number of refreshes scheduled in the next seven days.

The following two *backgrounder* command options determine how long a flow task can run before the flow background task is canceled. These two commands together determine the total timeout value for flow tasks.

backgrounder.extra_timeout_in_seconds

Default value: `1800`

The number of seconds beyond the setting in `backgrounder.querylimit` before a background job is canceled. This setting makes sure that a stalled job does not hold up subsequent jobs. The setting applies to processes listed in `backgrounder.timeout_tasks`. 1800 seconds is 30 minutes.

backgrounder.default_timeout.run_flow

Default value: `14400`

The number of seconds before a flow run task is canceled. 14,400 seconds is 4 hours.

backgrounder.failure_threshold_for_run_prevention

Default value: `5`

The number of consecutive failures of a subscription, extract, or flow run job before that job is suspended. Suspending continuously failing jobs helps preserver backgrounder resources for other jobs. To disable suspension of failing background tasks, set this to `-1`.

backgrounder.log.level

Version: Added in version 2020.3.0.

Default value: `info`

The logging level for the backgrounder process. This is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see Change Logging Levels.

backgrounder.querylimit

Default value: `7200`

Longest allowable time, in seconds, for completing a single extract refresh job. 7200 seconds = 2 hours.

> **Note:** If a background job reaches this time limit, it may continue to run for an additional several minutes while being canceled.

backgrounder.restrict_serial_collections_to_site_level

Default value: `false`

In Tableau Server, you can schedule extract refreshes, subscriptions, or flows to run periodically. These scheduled items are referred to as tasks. The Backgrounder process initiates unique instances of these tasks to run them at the scheduled time. The unique instances of the tasks that are initiated as a result are referred to as jobs.

This setting affects schedules that are configured to run serially. By default, when a schedule is configured to run serially, all jobs using that schedule will run serially. When this setting is set to `true`, jobs running on different sites can run in parallel. Jobs for scheduled tasks on the same site will continue to run serially.

The example below illustrate this scenario:

Tableau Server includes a schedule named "Daily" to run jobs every day at 7 am. The "Daily" schedule is configured to run serially. Site "HR" and site "Payroll" each have multiple scheduled tasks that use the schedule, "Daily". When this setting is set to `true`, jobs for these scheduled tasks on Site "HR" can run in parallel with jobs on site "Payroll", whereas jobs on the same site will still only run serially.

backgrounder.notifications_enabled

Default value: `true`

Controls whether extract refresh and flow run alerts are enabled for all sites on the server. By default alerts are enabled. To disable the alerts for all sites on a server, set this to `false`.

Extract alerts can be enabled or disabled on a site basis by site administrators in site settings, or at the user level in user settings.

backgrounder.sort_jobs_by_type_schedule_boundary_heuristics_milliSeconds

Default value: `60000`

Controls the time window that identifies backgrounder jobs which are determined to have the same scheduled start time.

The backgrounder process orders work that is scheduled at the same time to be executed by job type, running the fastest category of jobs first: Subscriptions, then Incremental Extracts, then Full Extracts.

Jobs are batched to determine which jobs are scheduled at the “same time”. A value 60,000 milliseconds (the default) indicates jobs for schedules starting within a 1 minute window should be classified in the same batch and so are ordered by type within that batch.

backgrounder.subscription_failure_threshold_for_run_prevention

Default value: `5`

Determines the number of consecutive subscription failures that must occur before alerting for a condition is suspended. When set to the default of `5`, alerting is suspended after 5 consecutive subscription failures. A value of `-1` will allow notification email to continue indefinitely. This threshold is server-wide, so applies to all subscriptions defined on the server.

backgrounder.subscription_image_caching

Default value: `true`

Controls whether backgrounder will cache images that are generated for subscriptions. Cached images do not have to be regenerated each time so caching improves subscription

performance. By default image caching is enabled. To disable image caching for all sites on a server, set this to `false`.

backgrounder.timeout_tasks

Default value: The default value may be different, depending on your version of Tableau Server. To see the default value list for your version of Tableau, run the `tsm configuration get` command:

```
tsm configuration get -k backgrounder.timeout_tasks
```

The list of tasks that can be canceled if they run longer than the combined values in `backgrounder.querylimit` and `backgrounder.extra_timeout_in_seconds`. The list of tasks is delimited with commas. The default list represents all the possible values for this setting.

backgrounder.timeout.single_subscription_notify

Version: Added in version 2021.2.

Default Value: `1800` seconds (30 minutes)

This is the maximum allowable time specified in seconds for completing a single subscription job.

backgrounder.timeout.sync_ad_group

Version: Added in version 2021.1.23, 2021.2.21, 2021.3.20, 2021.4.15, 2022.1.11, 2022.3.3, 2023.1.

Default Value: `14400` seconds (4 hours)

This is the maximum allowable time, specified in seconds, for completing an Active Directory group sync. This applies to *scheduled* group synchonizations done by the backgrounder service and prevents long-running syncs from running indefinitely. This does not impact group synchronizations done using the Tableau Server UI or the REST API.

backgrounder.vInstances_max_overflow_queue_size

Version: Added in version 20221.2.

Default Value: `1000`

The maximum number of jobs that can be in the secondary queue. A secondary queue is created when the number of jobs running is at the set concurrency limit. The default maximum is set to 1000 jobs - meaning if there are more than 1000 jobs when the concurrency limit is hit, anything more than 1000 jobs will not be queued. Use the backgrounder.vInstance_max_overflow_queue_size tsm command to make changes to the overflow maximum queue size.

The values should be specified in whole numbers.

backup.zstd.thread_count

Version: Added in version 2021.1.0. This key is dynamically configurable. For more information, see Tableau Server Dynamic Topology Changes

Default value: 2

The number of threads that should be used when creating a backup.

Increasing this number can improve backup performance, but we recommend thread count not exceed the number of logical processors on the Tableau Server computer, up to four.

basefilepath.backuprestore

Default value: `/var/opt/tableau/tableau_server/data/tabsvc/files/backups/`

The location in which the `tsm maintenance backup` command creates the backup. This is also the location where the backup file must be when restored using the `tsm maintenance restore` command or the `tsm maintenance send-logs` command. After setting this, you should run the tsm maintenance validate-backup-basefilepath command (available in version 2022.1 and later) to verify that permissions are set properly for the location. For more information, see tsm File Paths.

basefilepath.log_archive

Default value: `/var/opt/tableau/tableau_server/data/tabsvc/files/log-archives/`

The location in which the `tsm maintenance ziplogs` command creates the zipped archive. For more information, see tsm File Paths.

basefilepath.site_export.exports

Default value: `/var/opt/tableau/tableau_server-/data/tabsvc/files/siteexports/`

The location in which the `tsm sites export` command creates the export file. For more information, see tsm File Paths.

basefilepath.site_import.exports

Default value: `/var/opt/tableau/tableau_server-/data/tabsvc/files/siteimports/`

The location in which the `tsm sites import` command expects the import file to be located. For more information, see tsm File Paths.

clustercontroller.log.level

Version: Added in version 2020.3.0.

Default value: `info`

The logging level for Cluster Controller. This is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see Change Logging Levels.

clustercontroller.zk_session_timeout_ms

Default value: `300000`

The length of time, in milliseconds, that Cluster Controller will wait for the Coordination Service (ZooKeeper), before determining that failover is required.

dataAlerts.checkIntervalInMinutes

Default value: `60`

The frequency, in minutes, at which Tableau Server checks to determine if data-alert conditions are true.

(The server also checks whenever extracts related to data alerts are refreshed.)

dataAlerts.retryFailedAlertsAfterCheckInterval

Default value: `true`

Determines how often Tableau Server rechecks failing data alerts. When set to `true`, the server rechecks failing alerts at the frequency defined by `dataAlerts.checkIntervalInMinutes`. When set to `false`, the server rechecks failing alerts every five minutes, more quickly notifying alert recipients if data conditions have changed, but reducing server performance.

(The server also checks whenever extracts related to data alerts are refreshed.)

dataAlerts.SuspendFailureThreshold

Default value: `350`

Determines the number of consecutive data alert failures that must occur before alerting for a condition is suspended. When set to the default of 350, alerting is suspended after roughly two weeks of alerts. This threshold is server-wide, so applies to any data alert defined on the server.

databaseservice.max_database_deletes_per_run

Version: Added in version 2021.2.

Default value: null

Use this option to adjust the maximum number of embedded external assets (databases and tables) that can be deleted each time the backgrounder process, controlled by `features.DeleteOrphanedEmbeddedDatabaseAsset`, runs. If this option is left empty, the default maximum number of embedded external assets that can be deleted is 100.

For more information, see features.DeleteOrphanedEmbeddedDatabaseAsset.

dataserver.log.level

Version: Added in version 2020.3.0.

Default value: `info`

The logging level for Data Server. This is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see Change Logging Levels.

elasticserver.vmopts

Version: Added in version: 2019.1. Removed: 2022.1

This configuration option is not valid for Tableau Server versions 2022.1 and later. For Tableau Server versions 2022.1 and later, use `indexandsearchserver.vmopts` configuration option

Default value: "`-Xmx<default_value> -Xms<default_value>`"

The default value varies based on the amount of system memory. The JVM maximum heap size is scaled to be 3.125% of the total system RAM.

Controls the Elastic Server heap size. Because the default value scales automatically, use this option to override the default value only when absolutely necessary. Append the letter 'k' to the value to indicate kilobytes, 'm' for megabytes, or 'g' to indicate gigabytes. As a general rule, set initial heap size (`-Xms`) equal to the maximum heap size (`-Xmx`) to minimize garbage collections.

excel.shadow_copy_all_remote.enabled

Version: Added in versions 2019.1.5, 2019.2.1.

Default value: `false`

Controls whether Tableau Server creates a "shadow copy" of a shared Excel spreadsheet (`.xlxs` or `.xlxm`) that is being used as a live data source. When enabled, this option prevents Excel users from seeing a "Sharing Violation Error" and a message that the file is "currently in use." This option can have a performance impact with large Excel files. If Excel users do not need to edit the shared file, you do not need to enable this option.

> **Note:** Tableau Server always attempts to create a shadow copy of a `.xls` file. This option does not change that behavior.

extractservice.command.execution.timeout

Version: Added in version 2021.4.

Default value: `7200` seconds

Sets the timeout value for VConn extract refresh run time.

Example: `tsm configuration set -k extract-service.command.execution.timeout -v <timeout_in_seconds> --force-keys`

> **Note:** You must use the `--force-keys` option to change this value.

features.ActiveMQ

Version: Added in version 2021.4.

Default value: `true`

Controls whether Tableau Server uses the Apache ActiveMQ service (Tableau Server Messaging Service) for the internal messaging mechanism.

features.DeleteOrphanedEmbeddedDatabaseAsset

Version: Added in version 2021.2.

Default value: `true`

Controls a backgrounder process, for Tableau Catalog (or Tableau Metadata API), that deletes embedded external assets (databases and tables) that are no longer associated with downstream Tableau content. This process runs everyday at 22:00:00 UTC (coordinated universal time) and can delete a maximum of 100 external assets each day until there are no remaining external assets without connections to downstream Tableau content. You can set this option to `false` to stop this process from running. Alternatively, you can also adjust the maximum number of external embedded assets that can be deleted using data-baseservice.max_database_deletes_per_run.

For more information see, Troubleshoot missing content.

features.DesktopReporting

Default value: `false`

Controls whether Desktop License Reporting is enabled on the server. When set to `false` (the default), no Administrative Views related to desktop licenses are available. Set this to `true` to enable license reporting and to make license usage and expiration Administrative Views visible on the Server Status page. **Note:** Desktop License Reporting must be enabled on the client (Tableau Desktop) in order for information to be reported to Tableau Server.

features.IdentityMigrationBackgroundJob

Version: Added in version 2022.1. Default value was changed to `false` in versions 2021.4.22, 2022.1.18, 2022.3.10, 2023.1.6, and 2023.3.

Default value: `false`

Controls the process that performs the identity migration. When set to `true`, the identity migration runs in existing deployments immediately after upgrading Tableau Server to version 2022.1 (or later) and restoring a backup of Tableau Server version 2021.4 (or earlier). Set to `false` (default) to disable the identity migration.

For example, to start the identity migration, run the following:

```
tsm configuration set -k features.IdentityMigrationBackgroundJob -v
true
```

For more information, see About the Identity Migration.

Note: If the identity migration is disabled, Tableau Server cannot use the Identity Service to store and manage user identity information. Using the Identity Service is a prerequisite for certain capabilities like identity pools.

features.IdentityPools

Version: Added in version 2023.1.

Default value: `false`

A component of the identity pools capability that needs to be enabled if you perform a new Tableau Server installation. Requires `feature.NewIdentityMode` and `wgserver.authentication.legacy_identity_mode.enabled`. Set to `true` to enable identity pools. Set to `false` (default) to disable identity pools.

For example, to enable identity pools, run the following:

```
tsm configuration set -k features.IdentityPools -v true
tsm configuration set -k features.NewIdentityMode -v true
tsm configuration set -k wgserver.authentication.legacy_identity_mod-
e.enabled -v false
tsm pending-changes apply
```

For more information, see Troubleshoot identity pools.

features.MessageBusEnabled

Version: Added in version 2019.4.

Default value: `true`

Controls whether Tableau Server uses the new internal messaging mechanism.

features.NewIdentityMode

Version: Added in version 2022.1.

Default value: `false`. The default value was changed from `true` to `false` in 2023.1.6.

A prerequisite of the identity pools capability. Requires `wgserver.authentication.legacy_identity_mode.enabled` to be set to `false` to enable identity pools. Set to `true` to disable identity pools.

```
tsm configuration set -k features.IdentityPools -v true
tsm configuration set -k features.NewIdentityMode -v true
tsm configuration set -k wgserver.authentication.legacy_identity_
mode.enabled -v false
tsm pending-changes apply
```

For more information, see Troubleshoot identity pools.

features.PasswordlessBootstrapInit

Default value: `true`

Controls whether Tableau Server allows embedded credentials in bootstrap files. When enabled (the default), embedded credentials are included in the bootstrap file unless you specify that they should not be included. Set this to `false` if credentials should never be included in any bootstrap file you generate. For more information on generating bootstrap files, see tsm topology nodes get-bootstrap-file.

This option was added beginning with Tableau Server version 2019.3.

features.PasswordReset

Version: Retired in version 2024.2. For versions 2024.2 and later, use `viz-portal.password_reset`.

Default value: `false`

Applies only to servers that use local authentication. Set to `true`to let users reset their passwords with a "Forgot password" option on the sign-in page.

filestore.empty_folders_reaper.enabled

Version: Added in 2020.x (2020.1.14, 2020.2.11, 2020.3.6, 2020.4.2) and 2021.1.x. The default value was changed to `true` in 2021.2.

Default value: `true`

Enables the job that "reaps" (removes) empty Filestore folders.

filestore_empty_folders_reap.frequency_s

Version: Added in 2020.x (2020.1.14, 2020.2.11, 2020.3.6, 2020.4.2).

Default value: `86400` (24 hours)

Specifies in minutes, how often to run the job that removes empty Filestore folders.

features.Hyper_DisallowTDEPublishing

Version: Defaults to `true` beginning in version 2023.1.0. Deprecated in Tableau Server 2024.2.

Default value: `true`

Specifies if users can upload .tde format files. This format was replaced by .hyper format beginning in version of 10.5 of Tableau Server, but were not blocked from upload. Starting with Version 2024.3, .tde format files are no longer usable. The files were automatically converted to

.hyper format if one of several actions were performed. For more information, see Extract Upgrade to .hyper Format.

filestore.log.level

Version: Added in version 2020.3.0.

Default value: `info`

The logging level for File Store. This is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see Change Logging Levels.

filestore.reapemptyfoldersholdoffms

Version: Added in 2020.x (2020.1.14, 2020.2.11, 2020.3.6, 2020.4.2). This is not yet available in 2021.1.

Default value: `300000` (5 minutes)

Specifies in milliseconds, the amount of time to wait before removing empty Filestore folders.

floweditor.max_datafile_upload_size_in_kb

Version: Added in version 2020.4

Default value: `1048576`

For Tableau Prep flow web authoring, the maximum size of delimited text files (for example, CSV or TXT) that can be uploaded to Tableau Server.

gateway.external_url

Version: Added in version 2023.1

Default value: Null

Required when OpenID Connect (OIDC) authentication is configured in TSM during Tableau Server setup or with identity pools. Specifies the Tableau Server URL used by the identity

provider (IdP) to redirect users who authenticate into Tableau. The gateway external URL is the same URL that you specified as the redirect URL with your IdP, which is used for matching purposes.

For example, to redirect the IdP associated with OIDC authentication configuration to your Tableau Server, http://myco, run the following command:

```
tsm configuration set -k gateway.external_url -v http://myco
```

gateway.http.cachecontrol.updated

Default value: `false`

The Cache-Control HTTP header specifies whether the client browser should cache content sent from Tableau Server. To disable caching of Tableau Server data on the client, set this option to `true`.

gateway.http.hsts

Default value: `false`

The HTTP Strict Transport Security (HSTS) header forces browsers to use HTTPS on the domain where it is enabled.

gateway.http.hsts_options

Default value: `"max-age=31536000"`

By default, HSTS policy is set for one year (31536000 seconds). This time period specifies the amount of time in which the browser will access the server over HTTPS.

gateway.httpd.loglevel

Version: Added in 2021.3.0.

Default value: `notice`

Specifies the logging level for the Gateway (Apache HTTPD server). By default this is set to `notice`. Other options include `debug`, `info`, `warning`, `error`. If you change the logging level, be aware of potential impact to disk space usage and performance. As a best practice, return the logging level to the default after you have gathered the information you need. For detailed information on Apache logging, see the Apache HTTP documentation.

gateway.httpd.shmcb.size

Version: Added in 2021.4

Default value: `2048000`

Specifies the amount of memory in bytes for the circular buffer when using the `shmcb` storage type. This configuration key doesn't apply when using the `dbm` storage type.

gateway.httpd.socache

Version: Added in 2021.4

Default value: `shmcb`

Specifies the storage type of the global/inter-process SSL Session Cache. By default, this is set to `shmcb`, with another configurable option `dbm`. For more information about `shmcb` and `dbm` storage types, see SSLSessionCache Directive on the Apache website.

gateway.http.request_size_limit

Default value: `16380`

The maximum size (bytes) of header content that is allowed to pass through the Apache gateway on HTTP requests. Headers that exceed the value set on this option will result in browser errors, such as HTTP Error 413 (Request Entity Too Large) or authentication failures.

A low value for `gateway.http.request_size_limit` can result in authentication errors. Single sign-on solutions that integrate with Active Directory (SAML and Kerberos) often require large authentication tokens in HTTP headers. Be sure to test HTTP authentication scenarios before deploying into production.

We recommend setting `tomcat.http.maxrequestsize` option to the same value that you set for this option.

gateway.http.x_content_type_nosniff

Default value: `true`

The X-Content-Type-Options response HTTP header specifies that the MIME type in the Content-Type header should not be changed by the browser. In some cases, where MIME type is not specified, a browser may attempt to determine the MIME type by evaluating the characteristics of the payload. The browser will then display the content accordingly. This process is referred to as "sniffing." Misinterpreting the MIME type can lead to security vulnerabilities. The X-Content-Type-Options HTTP header is set to 'nosniff' by default with this option.

gateway.http.x_xss_protection

Default value: `true`

The HTTP X-XSS-Protection response header is sent to the browser to enable cross-site scripting (XSS) protection. The X-XSS-Protection response header overrides configurations in cases where users have disabled XXS protection in the browser. The X-XSS-Protection response header is enabled by default with this option.

gateway.log.level

Version: Added in version 2020.3.0.

Default value: `info`

The logging level for Gateway. This is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see Change Logging Levels.

gateway.public.host

Default value: <hostname>

The name (URL) of the server, used for external access to Tableau Server. If Tableau Server is configured to work with a proxy server or external load balancer, it is the name entered in a browser address bar to reach Tableau Server. For example, if Tableau Server is reached by entering `tableau.example.com`, the name for gateway.public.host is `tableau-.example.com`.

gateway.public.port

Default value: `80` (`443` if SSL)

Applies to proxy server environments only. The external port the proxy server listens on.

gateway.slow_post_protection.enabled

Default value: `true`

When enabled, this can provide some help in protecting against slow POST (Denial-of-Service) attacks by timing out POST requests that transfer data at extremely slow rates.

> **Note:** This will not eliminate the threat of such attacks, and could have the unintended impact of terminating slow connections.

gateway.slow_post_protection.request_read_timeout

Default value: `header=10-30,MinRate=500 body=30,MinRate=500`

When enabled by the preceding option, `gateway.slow_post_protection.enabled`, this option sets the Apache httpd ReadRequestTimeout. The httpd directive is documented at Apache Module mod_reqtimeout. The primary use of this option is as a defense the Slowloris attack. See the Wikipedia entry, Slowloris (computer security).

Note: Older versions use a default value: `header=15-20,MinRate=500 body-y=10,MinRate=500`

gateway.timeout

Default value: `7200`

Longest amount of time, in seconds, that the gateway will wait for certain events before failing a request (7200 seconds = 2 hours).

gateway.trusted

Default value: IP address of proxy server machine

Applies to proxy server environments only. The IP address(es) or host name(s) of the proxy server.

gateway.trusted_hosts

Default value: Alternate names of proxy server

Applies to proxy server environments only. Any alternate host name(s) for the proxy server.

hyper.file_partition_size_limit

Default value: 0

When set to 0, the size is set to unlimited and will use all the disk space that is available.

This option is used to set the disk space limit for a query that spools to disk. If your disk space usage by the spool.<id>.tmp file is higher than where you need it to be for your environment, it means that queries are spooling and taking up disk space. Use this option to limit the amount of disk space that any one query can use. The spool.<id>.tmp file can be found in the temp folder of the user account running Tableau Server. You can specify this value in K(KB), M(MB), G(GB), or T(TB) units. For example, you can specify the size limit as 100G when you want to limit the disk space usage to 100 GB.

For more information about spooling see the Memory and CPU Usage section in Tableau Server Data Engine.

hyper.global_file_partition_size_limit

Default value: 0

When set to 0, the size is set to unlimited and will use all the disk space that is available.

This option is used to set the disk space limit for all queries that spool to disk. If your disk space usage by the spool.<id>.tmp file is higher than where you need it to be for your environment, it means that queries are spooling and taking up disk space. The spool.<id>.tmp file can be found in the temp folder of the user account running Tableau Server. Use this option to limit the amount of disk space in sum total that all queries use when spooling to disk . You can specify this value in K(KB), M(MB), G(GB), or T(TB) units. For example, you can specify the size limit as 100G when you want to limit the disk space usage to 100 GB. Tableau recommends that you start with this configuration when fine tuning your spooling limits.

For more information about spooling see the Memory and CPU Usage section in Tableau Server Data Engine.

hyper.enable_accesspaths_symbolic_canonicalization

Default value: `false`

On Windows OS systems, in order to resolve symlinks, Hyper needs to have access to the directory where extracts are stored and all its parent directories. If this is not the case, you may see an error message in the Hyper log that says: **Unable to obtain canonical path for** `//dirA/subdir/myextract.hyper ... Access is denied`.

In such cases, you can set this to `true` so Data Engine (Hyper) will not try to resolve symlinks when using canonical paths.

> **Note:** Setting the value to `true` also implies that Hyper can no longer guarantee to guard against a potential attacker who manages to place a symlink to escape the allowed set of directories which Hyper is configured to allow access to.

hyper.log_queries

Default value: `true`

When set to true, query information is logged.

By default query information is logged. If however you find that the log files are too large for the amount of disk space available, you can set it to `false` to disable logging query information. Tableau recommends leaving this configuration set to `true`.

hyper.log_query_cpu

Default value: `false`

Use this setting to log how much time each query takes and the CPU usage.

hyper.log_timing

Default value: `false`

This setting is useful to find out more information about the queries, like compilation and parsing times. By default this setting is disabled. You can turn this by setting the value to `true` to collect more details about your queries. Note, however that this will increase the size of your data engine log files (\logs\hyper).

hyper.log_troublesome_query_plans

Default value: `true`

When set to `true`, logs query plans of query that are identified as problematic. Queries that are either canceled, running slower than 10 seconds, or if the queries are spooling to disk fall into this category. The information in the logs can be useful to troubleshoot problematic queries. You can change the setting to `false` if you are concerned about the size of the logs.

hyper.memory_limit

Default value: `80%`

Controls the maximum amount of memory used by Hyper. Specify the number of bytes. Append the letter 'k' to the value to indicate kilobytes, 'm' to indicate megabytes, 'g' to indicate gigabytes, or 't' to indicate terabytes. For example, `hyper.memory_limit="7g"`. Alternatively, specify the memory limit as a percentage of the overall available system memory. For example, `hyper.memory_limit="90%"`.

hyper.memtracker_hard_reclaim_threshold

Default value: `80%`

This setting only applies to Windows. Hyper keeps decompressed and decrypted parts of the extract in memory to make subsequent accesses faster. This setting controls when worker threads will start writing this data out to a disk cache to reduce memory pressure. If given as a percentage, the value is interpreted as a percentage of the overall `hyper.memory_limit` setting. For example, `hyper.memtracker_hard_reclaim_threshold="60%"`. Absolute values can be specified as 'k' (kilobytes), 'm' (megabytes), 'g' (gigabytes), or 't' (terabytes). For example, `hyper.memtracker_hard_reclaim_threshold="10g"`. The value should be larger than the `hyper.memtracker_soft_reclaim` threshold.

hyper.memtracker_soft_reclaim_threshold

Default value: `50%`

This setting only applies to Windows. When interacting with a Hyper file, Hyper will write out some data for caching or persisting the data. Windows has the special behavior that it locks freshly written data into memory. To avoid swapping, we force out the data when Hyper reaches the configured limit for the reclaim threshold. When the soft reclaim threshold is reached, Hyper will try to reclaim cached data in the background to attempt to stay below the reclaim threshold. In situations where swapping would happen otherwise, triggering reclamation in Hyper can lead to a better outcome. Therefore, if your Tableau Server installation experiences a lot of swapping, this setting can be used to attempt to reduce the memory pressure.

Specify the number of bytes. Append the letter 'k' to the value to indicate kilobytes, 'm' to indicate megabytes, 'g' to indicate gigabytes, or 't' to indicate terabytes. Alternatively, specify the value as a percentage of the overall configured memory for Hyper. For example, `hyper.memtracker_soft_reclaim_threshold="20%"`.

hyper.network_threads

Default value: `150%`

Controls the number of network threads used by Hyper. Specify either the number of network threads (for example, `hyper.network_threads=4`) or specify the percentage of threads in relation to the logical core count (for example, `hyper.network_threads="300%"`).

Network threads are used for accepting new connections and sending or receiving data and queries. Hyper uses asynchronous networking, so many connections can be served by a single thread. Normally, the amount of work that is done on network threads is very low. The one exception is opening databases on slow file systems, which can take a long time and block the network thread. If connection times are slow when you try to view or edit dashboards that use extracts and have not been used in a while and you frequently see “asio-continuation-slow” messages in the Hyper log and long “construct-protocol” times to Hyper in the Tableau log, try to increase this value.

hyper.objectstore_validate_checksums

Default value: `false`

A boolean setting that controls file integrity checks in Hyper. When set to `true`, Hyper will check the data in an extract file when it is first accessed. This allows silent corruption and corruption that would crash Hyper to be detected. In general, it is advisable to turn this setting on except for installations with very slow disks where it could cause performance regressions.

hyper.query_total_time_limit

Default value: `0` (which means unlimited)

Sets an upper bound on the total thread time that can be used by individual queries in Hyper. Append 's' to the value to indicate seconds, 'min' to indicate minutes, or 'h' to indicate hours.

For example to restrict all queries to a total time usage of 1500 seconds of total thread time, run the following command:

```
tsm configuration set -k hyper.query_total_time_limit -v 1500s
```

If a query runs longer then the specified limit, the query will fail and an error will be returned. This setting allows you to automatically control runaway queries that would otherwise use too many resources.

Hyper executes queries in parallel. For example, if a query executes for 100 seconds and during this time is running on 30 threads, the total thread time would be 3000 seconds. The thread time of each query is reported in the Hyper log in the “query-end” log entries in the “total-time” field.

hyper.session_memory_limit

Default value: `0` (which means unlimited)

Controls the maximum memory consumption that an individual query can have. Specify the number of bytes. Append the letter 'k' to the value to indicate kilobytes, 'm' to indicate megabytes, 'g' to indicate gigabytes, or 't' to indicate terabytes.

For example, to set the memory limit to 900 megabytes, run the following command:

```
tsm configuration set -k hyper.session_memory_limit -v 900m.
```

Alternatively, to specify the session memory limit as a percentage of the overall available system memory run the following command:

```
tsm configuration set -k hyper.session_memory_limit -v 90%.
```

Lowering this value can help when a query is using excessive amounts of memory and making other queries fail over a long period of time. By lowering the limit, the single big query would fail (or resort to spooling if spooling isn’t turned off) and not have a negative impact on other queries.

hyper.srm_cpu_limit_percentage

Default value (in percent): `75`

Specifies the maximum hourly average CPU usage permitted by Hyper. If exceeded, Data Engine will restart itself to minimize impact to other processes on the computer.

By default Data Engine will restart itself if it averages more than 75% usage of CPU over an hour. This value should not be changed except when working with Tableau Support, or if you are running Data Engine on a dedicated server node. If Data Engine is running on a dedicated node, you can safely increase this value to 95 percent to take full advantage of available computer hardware. For details on running Data Engine on a dedicated node, see Optimize for Extract Query-Heavy Environments.

To increase this to 95%:

```
tsm configuration set -k hyper.srm_cpu_limit_percentage -v 95 --
force-keys

tsm pending-changes apply
```

To reset this to the default of 75%:

```
tsm configuration set -k hyper.srm_cpu_limit_percentage -v 75 --
force-keys

tsm pending-changes apply
```

hyper_standalone.consistent_hashing.enabled

Default value: `true`

Improves the chance that the extract for a query is already cached. If the node with the extract cached cannot support additional load, you will be routed to a new node and the extract will be loaded into cache on the new node. This results in better system utilization because extracts are only loaded into memory if there is load that justifies the need.

hyper_standalone.health.enabled

Default value: `true`

Switches the load balancing metric from random selection to picking the Data Engine (Hyper) node based on a health score that is made of up of a combination of current Hyper activity and

system resource usage. Based on these values, the load balancer will pick the node that is most capable of handling an extract query.

hyper.temp_disk_space_limit

Default value: `100%`

Sets the upper limit of disk space at which Hyper will stop allocating space for temporary files. This setting can help to stop the hard disk from filling up with temporary files from Hyper and running out of disk space. If disk space reaches this threshold, Hyper will attempt to recover automatically without administrator intervention.

Specify it as percentage of the overall available disk space to be used. For example, `hyper-.temp_disk_space_limit="96%"`. When set to 100%, all of the disk space that is available can be used.

For Data Engine to start, the configured amount of disk space must be available. If not enough disk space is available, you will see a Data Engine log entry that says, “Disk limit for temporary files has been reached. Please free up disk space on the device. See the Hyper log for more information: No space left on device”.

hyper.hard_concurrent_query_thread_limit

Default value: 150%

Use this option to set the maximum number of threads Hyper should use for running queries. Use this when you want to set a hard limit on the CPU usage. Specify either the number of threads or specify the percentage of threads in relation to the logical core count. Hyper will most likely not use more resources than are configured by this setting but Hyper background and network threads are not affected by this setting (though they tend to not be CPU intensive).

It is important to consider that this setting controls the number of concurrent queries that can be executed. So, if you decrease this setting, the chance of queries needing to wait for currently running queries to complete increases, which may affect workbook load times.

hyper.soft_concurrent_query_thread_limit

Default value: 100%

Use this option to specify the number of threads that a single query can be parallelized across if sufficiently many threads are available given the `hard_concurrent_query_thread_limit` setting. Specify either the number of threads or specify the percentage of threads in relation to the logical core count.

To illustrate this, here is a simplified example:

Let's say you set this value to 10 threads, this means queries can be parallelized up to 10 threads. If only 2 queries are running, the remaining 8 threads are used to parallelize the 2 queries.

The *hyper. hard_concurrent_query_thread_limit*, and *hyper.soft_concurrent_query_thread_limit* options work together to give you some options to manage your CPU usage while maximizing available CPU resources to complete queries faster. If you don't want the Data Engine to use all the available CPU on the machine, change it to less than 100% to a percentage that is optimal for your environment. The soft limit is a way for you to limit CPU usage but allow it to go beyond the soft limit up to the hard limit if necessary.

Note: The *hyper.hard_concurrent_query_thread_limit* and *hyper.soft_concurrent_query_thread_limit* options replace hyper.num_job_worker_threads and hyper.num_task_worker_threads options available in Tableau Server versions 2018.3 and earlier, and are retired and no longer available. For information on the hyper.num_job_worker_threads and hyper.num_task_worker_threads, see tsm configuration set Options.

hyper.use_spooling_fallback

Default value: `true`

When set to `true`, it allows spooling to disk when querying extracts exceeds set RAM usage (80% of installed RAM). In other words, it allows Hyper to execute a query using the disk if it exceeds RAM usage.

Tableau recommends that you use the default setting. You can turn this off by setting the value to `false` if you are concerned about disk usage. If you turn this setting off, queries that use more than 80% of installed RAM will be canceled. Spooling queries usually take substantially longer to finish.

For more information about spooling see the Memory and CPU Usage section in Tableau Server Data Engine.

indexandsearchserver.vmopts

Version: Added in version: 2022.1.

Default value: "`-Xmx<default_value> -Xms<default_value>`"

The default value is based on the amount of system memory and is 3.125% of the total system RAM.

Controls the Index and Search Server heap size. Because the default value scales automatically, use this option to override the default value only when absolutely necessary. Append the letter 'k' to the value to indicate kilobytes, 'm' for megabytes, or 'g' to indicate gigabytes. As a general rule, set initial heap size (`-Xms`) equal to the maximum heap size (`-Xmx`) to minimize garbage collections.

jmx.security.enabled

Version: Added in version: 2022.1.

Default value: `false`

JMX is disabled by default, so secure JMX is also disabled. If you are enabling JMX we strongly recommend you enable secure JMX.

This is set to `true` and turns secure JMX on with SSL and basic username/password authentication for readonly access when you run the `tsm maintenance jmx enable` command and answer `y` when prompted to enable security features for JMX:

```
tsm maintenance jmx enable
We do not recommend you enable JMX unsecured on a production environment. Would you like to enable security features for JMX?
(y/n): y
```

jmx.ssl.enabled

Version: Added in version: 2022.1.

Default value: `true`

Enforces SSL for JMX. This option defaults to `true` but has no effect unless `jmx.security.enabled` is also set to `true`. To enable JMX security, run the `tsm maintenance jmx enable` command. Answer `y` when prompted to leave SSL enabled, or `n` to disable SSL:

```
tsm maintenance jmx enable
...
Would you like to enable SSL?
(y/n): n
```

jmx.ssl.require_client_auth

Version: Added in version: 2022.1.

Default value: `false`

This is set to `true` when you run the `tsm maintenance jmx enable` command and answer `y` when prompted to require client authentication (mTLS):

```
tsm maintenance jmx enable
...
Would you like to require client authentication (mTLS)?
(y/n): y
```

To complete configuration you must have a client cert and place this in the correct location on your client computer.

jmx.ssl.user.name

Version: Added in version: 2022.1.

Default value: `tsmjmxuser`

This is set when you install or upgrade Tableau Server.

jmx.ssl.user.password

Version: Added in version: 2022.1.

Default value: `<generated>`

This is set when you install or upgrade Tableau Server.

jmx.user.access

Version: Added in version: 2022.1.

Default value: `readonly`

You can change this to `readwrite` when you run the `tsm maintenance jmx enable` command and answer `y` when prompted to add `readwrite` access:

```
tsm maintenance jmx enable
...
JMX access is readonly by default. Would you like to add readwrite
access?
(y/n): y
```

licensing.login_based_license_management.default_requested_duration_seconds

Default value: `0`

Set to the duration (in seconds) that a user's login-based license can be offline with no connection to Tableau Server before they are prompted to activate again. This duration is always refreshed when Tableau Desktop is in use and can connect to Tableau Server.

licensing.login_based_license_management.enabled

Default value: `true`

Set to true to enable login-based license management. Set to false to disable login-based license management.

> **Note:** In order to use login-based license management, you must activate a product key that is enabled for login-based license management. You can use the `tsm licenses list` to see which product keys have login-based license management enabled.

licensing.login_based_license_management.max_requested_duration_seconds

Default value: `7776000`

Set to the maximum duration (in seconds) that a user's login-based license can be offline with no connection to Tableau Server before they are prompted to activate Tableau again. The maximum value is 7776000 seconds (90 days). This duration is always refreshed when Tableau Desktop is in use and can connect to Tableau Server.

maestro.app_settings.sampling_max_row_limit

Default value: `1000000`

Sets the maximum number of rows for sampling data from large data sets with Tableau Prep on the web.

maestro.input.allowed_paths

Default value: ""

By default, access to any directory will be denied, and only publishing to Tableau Server with content that is included in the tflx file is allowed.

A list of allowed network directories for flow input connections. You must enable Tableau Prep Conductor to schedule flows on your Tableau Server. For more information, see Tableau Prep Conductor.

The following rules apply and must be considered when configuring this setting:

- Paths should be accessible by Tableau Server. These paths are verified during server startup and at flow run time.
- Network directory paths have to be absolute and cannot contain wildcards or other path traversing symbols. For example `\\myhost\myShare\*` or `\\myhost\myShare*` are invalid paths and would result in all the paths as disallowed. The correct way to safelist any folder under *myShare* would be `\\myhost\myShare or \\myhost\\myShare\`.

  > **Note:** The `\\myhost\myShare` configuration will not allow `\\myhost\myShare1`. In order to safe list both of these folders one would have safe list them as \\myhost\myShare; \\myhost\myShare1.

- The value can be either `*` meaning that any path, including local (with the exception of some system paths configured using “native_api.internal_disallowed_paths”), or a list of paths, delimited by “;”.

  > **Note:** If a path is both on the flows allowed list and internal_disasslowed list, internal_disallowed takes precedence.

Important:
This command overwrites existing information and replaces it with the new information you provided. If you want to add a new location to an existing list, you must provide a list of all the

locations, existing and the new one you want to add. Use the following commands to see the current list of input and output locations:

```
tsm configuration get -k maestro.input.allowed_paths
tsm configuration get -k maestro.output.allowed_paths
```

For more information and details about configuring allowed directories for flow input and output connections, see Step 4: Safe list Input and Output locations.

maestro.output.allowed_paths

Default value: ""

By default, access to any directories will be denied.

A list of allowed network directories for flow output connections. You must enable Tableau Prep Conductor to schedule flows on your Tableau Server. For more information, see Tableau Prep Conductor.

The following rules apply and must be considered when configuring this setting:

- Paths should be accessible by Tableau Server. These paths are verified during server startup and at flow run time.
- Network directory paths have to be absolute and cannot contain wildcards or other path traversing symbols. For example `\\myhost\myShare\*` or `\\myhost\myShare*` are invalid paths and would result in all the paths as disallowed. The correct way to safelist any folder under *myShare* would be `\\myhost\myShare or \\myhost\\myShare\`.

  Note: The `\\myhost\myShare` configuration will not allow `\\myhost\myShare1`. In order to safe list both of these folders one would have safe list them as \\myhost\myShare; \\myhost\myShare1.

- The value can be either `*` meaning that any path, including local (with the exception of some system paths configured using “native_api.internal_disallowed_paths”), or a list of paths, delimited by “;”.

  > **Note:** If a path is both on the flows allowed list and internal_disasslowed list, internal_disallowed takes precedence.

For more information and details about configuring allowed directories for flow input and output connections, see Step 4: Safe list Input and Output locations.

maestro.output.write_to_mssql_using_runas

Version: Added in version: 2022.3.1

Default value: `false`

When enabled, flow outputs published to Tableau Server are allowed write access to a Microsoft SQL Server database using Run As credentials. The credentials used by the Run As service account must have write permission to the database. Evaluate your security and deployment requirements before enabling the `maestro.output.write_to_mssql_using_runas` setting. For more information, see Run As Service Account.

> **Note**: This command requires the `--force-keys` option. For example: `tsm configuration set -k maestro.output.write_to_mssql_using_runas -v true --force-keys`.

maestro.sessionmanagement.maxConcurrentSessionPerUser

Default value: `4`

Sets the maximum number of flow web editing sessions that a user can have open at one time.

metadata.ingestor.blocklist

Default value: null

When configured, Tableau Catalog blocks specified content from being ingested. To specify which content to block, you must identify the blocklist values, which is a combination of both the site ID, content type, and content ID of the content you want to block, from the server "non-interactive" log files. Blocklist values must be separated by a comma.

Important: You should only use this option when directed to do so by Tableau Support.

For example, you can use the `tsm configuration set --force-keys -k metadata.ingestor.blocklist` to block ingestion of a combination of data sources, workbooks, and flows using the following command:

```
tsm configuration set --force-keys -k metadata.ingestor.blocklist -v
"sites/1/datasources/289, sites/2/datasources/111, sites/1/-
workbooks/32, sites/3/workbooks/15, sites/1/flows/13, sites/1/-
flows/18"
```

To validate blocked content, review the server "noninteractive" log files for the following events:

- `Skipping ingestion for`
- `Successfully updated blocklist to`

For example:

```
Skipping ingestion for contentType [Workbook], contentId
[sites/1/datasources/289], siteDisabled [false], swallowEvent
[false], contentBlocked [true]
```

```
Skipping ingestion for contentType [Workbook], contentId [sites/3/-
workbooks/15], siteDisabled [false], swallowEvent [false], con-
tentBlocked [true]
```

and

```
Successfully updated blocklist to: [sites/1/datasources/289,
sites/1/workbooks/32, sites/2/datasources/111]
```

metadata.ingestor.pipeline.throttleEventsEnable

Default value: `false`

Controls whether indexing of new and updated content, also called eventing, is regulated across all sites on the server. By default, event throttling is turned off. To turn on event throttling, change this setting to `true` using the following command:

```
tsm configuration set -k metadata.in-
gestor.pipeline.throttleEventsEnable -v true --force-keys
```

For more information about event throttling, see Enable Tableau Catalog.

metadata.ingestor.pipeline.throttleLimit

Default value: `20`

When event throttling is enabled, this is the maximum number of new and updated content items that can be indexed during a specified period of time. Once the specified limit is reached for a specific item, indexing is deferred.

By default, the limit is set to `20` and can't be set to lower than `2`. You can use the following command to change the limit:

```
tsm configuration set -k metadata.ingestor.pipeline.throttleLimit -
v 25 --force-keys
```

Throttled events can be identified in the server "noninteractive" log files as `ingestor event flagged for removal by throttle filter`.

metadata.ingestor.pipeline.throttlePeriodLength

Default value: `20`

When event throttling is enabled, this is the period of time, in minutes, a specified maximum number of new and updated content items can be indexed. Once the specified time is reached, indexing of any additional new and updated content is deferred.

By default, the time is set to 30 minutes. You can use the following command to change the time:

```
tsm configuration set -k metadata.in-
gestor.pipeline.throttlePeriodLength -v PT45M --force-keys
```

metadata.query.limits.time

Default value: `20`

This is the longest allowable time, in seconds, for a Catalog or Metadata API query to run before a timeout occurs and the query is canceled. Tableau recommends incrementally increasing the timeout limit to *no more than* 60 seconds using the following command:

```
tsm configuration set -k metadata.query.limits.time –v PT30S --
force-keys
```

Important: This option should be changed only if you see the error described here, Timeout limit and node limit exceeded messages. Increasing the timeout limit can utilize more CPU for longer, which can impact the performance of tasks across Tableau Server. Increasing the timeout limit can also cause higher memory usage, which can cause issues with the interactive microservices container when queries run in parallel.

metadata.query.limits.count

Default value: `20000`

This is the number of objects (which can loosely map to the number of query results) that Catalog can return before the node limit is exceeded and the query is canceled. Tableau recommends incrementally increasing the timeout limit, to *no more than* 100,000 using the following command:

```
tsm configuration set -k metadata.query.limits.count -v 3000 --
force-keys
```

Important: This option should be changed only if you see the error described here, Timeout limit and node limit exceeded messages. Increasing the node limit can cause higher memory usage, which can cause issues with the interactive microservices container when queries run in parallel.

metadata.query.throttling.enabled

Version: Added in version 2023.3

Default value: `true`

Controls whether Metadata API query throttling is enabled. Metadata API query throttling is a feature designed to prevent a server's API responses from negatively impacting overall performance. When set to `true` (the default), if a request to the Metadata API exceeds the defined threshold, a `RATE_EXCEEDED` error is returned.

If Metadata API users are seeing frequent `RATE_EXCEEDED` errors, an administrator can try to adjust throttling using the `metadata.query.throttling.tokenRefilledPerSecond` and `metadata.query.throttling.queryCostCapacity` settings. Alternatively, the administrator can disable throttling entirely by setting `metadata.query.throttling.enabled` value to `false`. Doing so would prevent the performance-protecting benefits of the feature however.

metadata.query.throttling.queryCostCapacity

Version: Added in version 2023.3

Default value: `20000000`

A number representing the capacity that the Metadata API has for answering queries. Each request to the Metadata API has a calculated cost that is subtracted from this number when it's executed. (Using a token bucket model, this is the maximum amount of tokens that can be in the bucket.)

If Metadata API users are seeing frequent `RATE_EXCEEDED` errors, an administrator can adjust throttling settings. They should adjust `metadata.query.throttling.tokenRefilledPerSecond` and test the results before trying to adjust `metadata.query.throttling.queryCostCapacity`. Alternatively, the administrator can disable throttling entirely by setting `metadata.query.throttling.enabled` to `false`. Doing so would prevent the performance-protecting benefits of the feature however.

metadata.query.throttling.tokenRefilledPerSecond

Version: Added in version 2023.3

Default value: `5555`

A number representing the amount of Metadata API query capacity that's regenerated every second. (Using a token bucket model, this is the number of tokens that are put into the bucket every second.)

If Metadata API users are seeing frequent `RATE_EXCEEDED` errors, an administrator can adjust throttling settings. They should adjust `metadata.query.throttling.tokenRefilledPerSecond` and test the results before trying to adjust `metadata.query.throttling.queryCostCapacity`. Alternatively, the administrator can disable throttling entirely by setting `metadata.query.throttling.enabled` to `false`. Doing so would prevent the performance-protecting benefits of the feature however.

metricsservices.checkIntervalInMinutes

Version: Retired in version 2024.2.

Default value: `60`

Controls the interval, in minutes, between refreshes for metrics that rely on live data sources. A metric refreshes when the server checks for new data via the metric's connected view.

metricsservices.enabled

Version: Added in version: 2022.3. Retired in version 2024.2.

Default value: `true`

When set to `false`, the metrics content type is disabled for all sites on a server. For more information, see Disable metrics for a server.

Retirement of the legacy metrics feature

Tableau's legacy metrics feature was retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2. In October 2023, Tableau retired the ability to embed legacy metrics in Tableau Cloud and in Tableau Server version 2023.3. With Tableau Pulse, we've developed an improved experience to track metrics and ask questions of your data. For more information, see Create Metrics with Tableau Pulse to learn about the new experience and Create and Troubleshoot Metrics (Retired) for the retired feature.

metricsservices.failureCountToWarnUser

Version: Retired in version 2024.2.

Default value: `10`

Controls the number of consecutive refresh failures that must occur before the metric owner is warned. When set to the default of 10, a metric refresh must fail 10 times in a row before the owner is sent a notification about the failure.

metricsservices.maxFailedRefreshAttempts

Version: Retired in version 2024.2.

Default value: `175`

Controls the number of consecutive refresh failures that must occur before a metric refresh is suspended.

mobile.deep_linking.on_prem.enabled

Default value: `true`

Controls whether links to Tableau Server are treated as deep links by the Tableau Mobile app. When set to `true`, links to supported content types open in the app. When set to `false`,

links open in the mobile browser. For more information see, Control deep linking for Tableau Mobile.

monitoring.dataengine.connection_timeout

Default value: `30000`

The length of time, in milliseconds, that Cluster Controller will wait for the data engine, before determining that a connection timeout occurred. The default is 30,000 milliseconds (30 seconds).

native_api.allowed_paths

Default value: `""`

> **Note:** In Tableau Server releases (including maintenance releases) before October 2023, this setting was configured to allowed access to all paths by default.

Use this setting to specify an allowlist for access to files stored on Tableau or on remote shares. This scenario allows authorized Tableau Server users to build workbooks that use files on the server as file-based data sources (such as spreadsheets).

This setting allows you to limit access only to those directories that you specify. The `tableau` system account access is therefore limited to the directory paths where you host data files.

`tsm configuration set -k native_api.allowed_paths -v "`*`path`*`"` , where *path* is the directory to add to the allowlist. All subdirectories of the specified path will be added to the allowlist. If you want to specify multiple paths, separate them with a semicolon, as in this example:

```
tsm configuration set -k native_api.allowed_paths -v "/data-
sources;/HR/data"
```

Important: Make sure the file paths you specify in this setting exist and are accessible by the system account.

native_api.connection.limit.<connection class>

Set parallel query limit for the specified data source (connection class). This overrides the global limit for the data source.

native_api.connection.globallimit

Default value: `16`

Global limit for parallel queries. Default is 16 except for Amazon Redshift which has a default of 8.

native_api.ExplainDataEnabled

Default value: `true`

This option controls whether Explain Data is enabled or disabled for the server. For more information about Explain Data, see Get Started with Explain Data in the Tableau Help.

This option was added beginning with Tableau Server version: 2019.3.

native_api.force_alternative_federation_engine

Default value: `false`

Override the operation restrictions when joining data from a single file connection and a single SQL database connection. Set this option to `True` to force Tableau to process the join using the live database connection.

native_api.ProtocolTransitionLegacyFormat

Default value: `false`

Use the legacy name format for constrained delegation.

The name format was changed in version 10.1 to allow cross-domain protocol transition (S4U). If this causes problems with existing configurations and you don't need cross-domain protocol transition, configure Tableau Server to use the old behavior by setting this to `true`.

native_api.unc_mountpoints

Default value: none

Specifies UNC and FQDN path for shared Windows directories that are accessed by Tableau Server on Linux. Each path must also be referenced in a corresponding auto.cifs file. Separate each path by a semicolon, for example:

```
'//files-
rv01/devel-
opment;/m-
nt/-
files-
rv01/devel-
opment;//-
filesrv01.example.lan/development;/mnt/filesrv01/development'
```

Subsequent updates to the `native_api.unc_mountpoints` value will overwrite the existing value. Therefore, each time you add a Windows share, you must include all shares in the updated value.

For more information, see the Community wiki topic, Connecting to a Windows Shared Directory.

native_api.InitializeQueryCacheSizeBasedOnWeights

Default value: `True`

Controls whether the query cache size is initialized automatically based on the amount of available system memory. The query cache consists of the logical query cache, metadata cache, and native query cache. By default, this functionality is enabled.

native_api.QueryCacheMaxAllowedMB

The maximum size of the query cache in megabytes. This value varies based on the amount of system memory. The query cache consists of the logical query cache, metadata cache, and native query cache. Use the table below to determine your default value:

System Memory	Default Value for Tableau Server	Default Value for Tableau Desktop
64 GB and more	`3200 MB`	`1600 MB`
From 32 GB to 64 GB	`2400 MB`	`1200 MB`
From 16 GB to 32 GB	`1600 MB`	`800 MB`
16 GB and less	`800 MB`	`400 MB`

native_api.LogicalQueryCacheMaxAllowedWeight

Default value: `70`

The weight of logical query cache size limit in the total query cache size.

native_api.MetadataQueryCachMaxAllowedWeight

Default value: `4`

The weight of metadata query cache size limit in the total query cache size.

native_api.NativeQueryCacheMaxAllowedWeight

Default value: `26`

The weight of native query cache size limit in the total query cache size.

native_api.QueryCacheEntryMaxAllowedInPercent

Default value: `60`

Specifies the maximum size of query results that can be put into the query cache. It is set as the percentage of the total query cache size. For example, if the logical query cache size is 100 MB and `native_api.QueryCacheEntryMaxAllowedInPercent` is set to 60 percent, then only query results that are smaller than 60 MB can be put into the logical query cache.

native_api.UserInfoInGeneratedSQLEnabled

Default value: `false`

Determines if query tagging is enabled for all content on a Tableau Server. When true, queries sent from Tableau to customer SQL databases will include metadata about the source of the query. The resulting content in customer database logs can be used for troubleshooting performance or other issues.

nlp.concepts_shards_count

Default value: 1

> **Note:** The default shard count value is sufficient for most Tableau Server installations.

Controls the number of data shards for the Concepts index of Ask Data, field names, field synonyms, and analytical terms stored in shards in:

- The Index and Search Server for 2022.1 and later versions.
- Elastic Server for 2019.1 - 2021. 4

The shard count partitions the search index to reduce total index size, which may improve the performance of Ask Data's semantic parser. Adjusting the shard count is another performance enhancement measure that you can take along with increasing the heap size through `elasticserver.vmopts` or `indexandsearchserver.vmopts`, depending on the version of Tableau Server that you are running.

Tableau recommends increasing the shard count by 1 for every 50 GB. To reduce the number of times you need to adjust the shard count, calculate the total index size by adding 50% to the current index. For example, if the total index size is less than 50 GB, then 1 shard is sufficient.

Actual performance will vary depending on the server, the rate at which the index size grows, and other factors.

- 0 to 50 GB: 1
- 50 GB to 100 GB: 2
- 100 GB to 150 GB: 3

You can use the following command to increase the Concepts index shard count from default to 2:

```
tsm configuration set -k nlp.concepts_shards_count -v 2
```

nlp.values_shards_count

Default value: 1

Controls the number of data shards for the Concepts index of Ask Data, field names, field synonyms, and analytical terms stored in shards in:

- The Index and Search Server for 2022.1 and later versions.
- Elastic Server for 2019.1 - 2021. 4

The shard count partitions the search index to reduce total index size, which may improve the performance of Ask Data's semantic parser. Adjusting the shard count is another performance enhancement measure that you can take along with increasing the heap size through `elasticserver.vmopts` or `indexandsearchserver.vmopts`, depending on the version of Tableau Server that you are running.

Tableau recommends increasing the shard count by 1 for every 50 GB. To reduce the number of times you need to adjust the shard count, calculate the total index size by adding 50% to the current index. For example, if the total index size is less than 50 GB, then 1 shard is sufficient. Actual performance will vary depending on the server, the rate at which the index size grows, and other factors.

- 0 to 50 GB: 1
- 50 GB to 100 GB: 2
- 100 GB to 150 GB: 3

You can use the following command to increase the Values index shard count from default to 2:

```
tsm configuration set -k nlp.values_shards_count -v 2
```

nlp.defaultNewSiteAskDataMode

Default value: `disabled_by_default`

Use this option to set the initial value of the Ask Data Mode when a site is created. For more information see Disable or Enable Ask Data for a Site.

Valid options are `disabled_by_default` and `disabled_always`.

This option was added beginning with Tableau Server versions: 2019.4.5, 2020.1.3.

noninteractive.vmopts

Default value: `"-XX:+UseConcMarkSweepGC -Xmx<default_value>g -XX:+ExitOnOutOfMemoryError"`

The default value varies based on the amount of system memory. The JVM maximum heap size is scaled to be 6.25% of the total system RAM.

This option controls the JVM maximum heap size for Tableau Catalog ingestion. Because the default value scales automatically, use this option to override the default value only when absolutely necessary by modifying the `-Xmx<default_value>g` argument. For example, you can use the following command to increase the max heap size to 2 GB:

```
tsm configuration set -k noninteractive.vmopts -v "-XX:+UseConcMarkSweepGC -Xmx2g -XX:+ExitOnOutOfMemoryError"
```

For more information, see Memory for non-interactive microservice containers.

pgsql.port

Default value: `8060`

Port that PostgreSQL listens on.

pgsql.preferred_host

Specifies the computer name of the node with the preferred repository installed. This value is used if the `--preferred` or `-r` option is specified with the tsm topology failover-repository command.

Example:

```
tsm configuration set -k pgsql.preferred_host -v "<host_name>"
```

Note: The `host_name` is case-sensitive and must match the node name shown in the output of `tsm status -v`.

pgsql.ssl.ciphersuite

Default value: `HIGH:MEDIUM:!aNULL:!MD5:!RC4`

Specifies the cipher algorithms that are allowed for SSL for the Repository.

For acceptable values and formatting requirements, see ssl_ciphers on the Postgres website.

pgsql.ssl.max_protocol_version

Default value:`TLSv1.3`

Sets the maximum SSL/TLS protocol version to use when connecting to the repository over SSL.

Valid values: `TLSv1`, `TLSv1.1`, `TLSv1.2`, `TLSv1.3`

pgsql.ssl.min_protocol_version

Default value:`TLSv1.2`

Sets the minimum SSL/TLS protocol version to use when connecting to the repository over SSL.

Valid values: `TLSv1`, `TLSv1.1`, `TLSv1.2`, `TLSv1.3`

pgsql.verify_restore.port

Default value: `8061`

Port used to verify the integrity of the PostgreSQL database. See tsm maintenance backup for more information.

ports.blocklist

Version: Added in version 2021.1

Default value: no ports blocked in the range used for automatic port assignment.

Used to specify ports within the port assignment range that should not be used by Tableau when dynamically assigning ports. This is useful when you know that another application is using a port within the range. Separate multiple ports with commas, for example:

```
tsm configuration set -k ports.blocklist -v 8000,8089, 8090
```

For more information on using the `ports.blocklist` key, see Blocking specific ports within the range

recommendations.enabled

Default value: `true`

Controls the recommendations feature, which powers recommendations for data sources and tables (for Tableau Desktop) and recommendations for views (for Tableau Server). Recommendations are based on the popularity of content and on content used by other users determined to be similar to the current user.

recommendations.vizrecs.enabled

Default value: `true`

Controls recommendations for views for Tableau Server users. This option is a child of `recommendations.enabled` and will have no effect if the parent option is set to false. When the parent option is set to true, and this option is set to false, data sources and tables will still be

recommended to Tableau Desktop users, but recommendations for views on Tableau Server will be disabled.

redis.max_memory_in_mb

Default value: `1024`

Specifies the size in megabytes of the cache server external query cache.

refresh_token.absolute_expiry_in_seconds

Default value: `31536000`

Specifies the number of seconds for absolute expiration of refresh tokens and personal access tokens (PATs).

Refresh tokens are used by connected clients (Tableau Desktop, Tableau Prep Builder, Tableau Mobile, etc.) for authentication to Tableau Server after initial sign-in.

To remove limits set the value to `-1`. To disable refresh tokens and PATs, see Disable Automatic Client Authentication.

refresh_token.idle_expiry_in_seconds

Default value: `1209600`

Specifies the number of seconds when idle refresh tokens expire. The refresh tokens are used by connected clients (Tableau Desktop, Tableau Prep Builder, Tableau Mobile, etc.) for authentication to Tableau Server after initial sign-in. To remove limits set the value to `-1`.

refresh_token.max_count_per_user

Default value: `24`

Specifies the maximum number of refresh tokens that can be issued for each user. If the maximum number of users sessions is not enough, increase this value or set it to `-1` to entirely remove this refresh token limit.

rsync.timeout

Default value: `600`

Longest allowable time, in seconds, for completing file synchronization (600 seconds = 10 minutes). File synchronization occurs as part of configuring high availability, or moving the data engine and repository processes.

schedules.display_schedule_description_as_name

Default value: `false`

Controls whether a schedule name displays when creating a subscription or extract refresh (the default), or the "schedule frequency description" name describing the time and frequency of the schedule displays. To configure Tableau Server to display timezone-sensitive names for schedules, set this value to `true`.

When true, the "schedule frequency description" is also displayed after the schedule name on the schedule list page.

schedules.display_schedules_in_client_timezone

Default value: `true`

Shows the "schedule frequency description" in the timezone of the user when true (uses the client browser timezone to calculate the "schedule frequency description").

schedules.ignore_extract_task_priority

Default value (boolean): `False`

This setting controls whether or not task priority is considered for determining the job rank which determines when to pull jobs off the queue. Setting this to `true` disables editing the task priority on tasks, and only schedule priority will be considered for determining the job rank.

searchserver.connection_timeout_milliseconds

Version: Added in version 2019.1. Deprecated in version 2022.3. Retired in version 2023.3.

Default value, in milliseconds: `100000`

Specifies, in milliseconds, the amount of time Search & Browse clients will wait to establish a connection to the Search & Browse server.

On especially busy Tableau Server computers, or if you see log errors "Failed zookeeper health check. Refusing to start SOLR." increase this value.

For more information, see Client session timeouts.

searchserver.index.bulk_query_user_groups

Version: Retired in version 2022.3.

Default value: `true`

Specifies whether querying of site users is done in bulk when importing or deleting users with a CSV file. When set to `true`(the default) indexing is done as in bulk.

searchserver.javamemopts

Version: Added in version 2019.1. Retired in 2023.3

Default value: `-Xmx512m -Xms512m -XX:+ExitOnOutOfMemoryError -XX:-UsePerfData`

Determines JVM options for SOLR.

Of all configurable options, the maximum heap memory, configured by the `-Xmx` parameter, is the most important when tuning the searchserver. In most cases this should be set as high as is possible, up to 24 GB, based on available physical memory on the Tableau Server computer. To change only the max heap memory, specify the entire default string but only change the value for `-Xmx`.

Valid values for `-Xmx` depend on available memory on the Tableau Server computer, but cannot be greater than 24 GB. For more information, see Search & Browse Max Heap Memory.

searchserver.startup.zookeeper_healthcheck_timeout_ms

Version: Added in version 2020.1. Retired in version 2023.3.

Default value, in milliseconds: `300000`

Specifies, in milliseconds, the amount of time Tableau Server should wait for a successful Zookeeper health check on startup.

On especially busy Tableau Server computers, or if you see log errors "Failed zookeeper health check. Refusing to start SOLR." increase this value.

For more information, see Zookeeper connection health check timeout at startup.

searchserver.zookeeper_session_timeout_milliseconds

Version: Retired in version 2022.3.

Default value, in milliseconds: `100000`

Specifies, in milliseconds, the amount of time Search & Browse clients will wait to establish a connection to the Coordination Service (Zookeeper).

For more information, see Client session timeouts.

ServerExportCSVMaxRowsByCols

Version: Added in version 2020.3.

Default value: `0` (no limit)

Specifies the maximum number of cells of data that can be downloaded from View Data into a CSV file. By default, there is no limit. Specify the number of cells. For example to set a limit of 3 million:

```
tsm configuration set -k ServerExportCSVMaxRowsByCols -v 3000000
tsm pending-changes apply
```

service.jmx_enabled

Default value: `false`

Setting to `true` enables JMX ports for optional monitoring and troubleshooting.

service.max_procs

Default value: <number>

Maximum number of server processes.

service.port_remapping.enabled

Default value: `true`

Determines whether or not Tableau Server will attempt to dynamically remap ports when the default or configured ports are unavailable. Setting to `false` disables dynamic port remapping.

sheet_image.enabled

Default value: `true`

Controls whether you can get images for views with the REST API. For more information, see REST API Reference.

ssl.ciphersuite

Default value: `HIGH:MEDIUM:!EXP:!aNULL:!MD5:!RC4:!3DES:!CAMELLIA:!IDEA:!SEED`

Specifies the cipher algorithms that are allowed for SSL for Gateway.

For acceptable values and formatting requirements, see SSLCipherSuite on the Apache website.

ssl.client_certificate_login.blocklisted_signature_algorithms

Default value:

- Version 2020.4.0:

```
sha1withrsaencryption,
sha1withrsa
```

- Version 2020.4.1 and later:

```
sha1withrsaencryption,
sha1withrsa,
sha1withrsaandmgf1,
sha1withdsa,
sha1withecdsa
```

The default value blocks certificates with the SHA-1 signing algorithm. Specifies the client signing algorithms that are blocked for SSL. To disable blocking of all signature algorithms, run this key with an empty set of quotes.

For more information about this key, see the Knowledge Base article, Mutual SSL Fails After Upgrading if Certificates Signed with SHA-1.

ssl.client_certificate_login.min_allowed.elliptic_curve_size

Default value: `256`

Specifies the minimum elliptic curve size required for ECDSA client certificates that are authenticating with Tableau Server over mutual SSL If a client presents an ECDSA client certificate that does not satisfy this minimum curve size, the authentication request will fail.

This option was introduced in Tableau Server version 2021.1.

ssl.client_certificate_login.min_allowed.rsa_key_size

Default value: `2048`

Specifies the minimum key size for RSA client certificates that are authenticating with Tableau Server over mutual SSL If a client presents an RSA client certificate that does not satisfy this minimum key size, the authentication request will fail.

This option was introduced in Tableau Server version 2021.1.

ssl.protocols

Default value: `all +TLSv1.2 -SSLv2 -SSLv3 -TLSv1.3`

Specifies the SSL protocols that Tableau Server supports for TLS connections for Gateway. Acceptable values derive from the Apache SSLPrtocol Directive. We recommend following SSL protocol configuration as described in Security Hardening Checklist.

storage.monitoring.email_enabled

Default value: `false`

Controls whether email notifications are enabled for server disk space monitoring. By default, email notifications are enabled. To enable notifications for disk space monitoring, set this to `true`.

SMTP must be configured for notifications to be sent. For details, see Configure SMTP Setup.

storage.monitoring.warning_percent

Default value: `20`

Warning threshold of remaining disk space, in percentage of total disk space. If disk space falls below this threshold, a warning notification is sent.

storage.monitoring.critical_percent

Default value: `10`

Critical threshold of remaining disk space, in percentage of total disk space. If disk space falls below this threshold, a critical notification is sent.

storage.monitoring.email_interval_min

Default value: `60`

How often, in minutes, that email notifications should be sent when disk space monitoring is enabled and a threshold is crossed.

storage.monitoring.record_history_enabled

Default value: `true`

Determines whether free disk space history is saved and available to view in Administrative Views. To disable history storage for monitoring, set `storage.monitoring.record_history_enabled` to `false`.

subscriptions.enabled

Default value: `false`

Controls whether subscriptions are configurable system-wide. See Set Up a Site for Subscriptions.

subscriptions.timeout

Default value: `1800`

Length of time, in seconds, for a view in a workbook subscription task to be rendered before the task times out. If this time limit is reached while a view is being rendered, the rendering continues, *but any subsequent view in the workbook is not rendered*, and the job ends in error. In the case of a single-view workbook, this value will never result in the rendering being halted due to a timeout.

svcmonitor.notification.smtp.enabled

Default value: `false`

Controls whether email notifications are enabled for server process events. By default notifications are sent when processes go down, fail over, or restart. To enable server process notifications, set this to `true`.

SMTP must be configured for notifications to be sent. For details, see Configure SMTP Setup.

svcmonitor.notification.smtp.mime_use_multipart_mixed

Version: Added in version: 2020.1.8, 2020.2.5, 2020.3.1

Default value: `false`

Controls whether subscription HTML MIME attachments are sent as *multipart/related* (the default) or *multipart/mixed*.

In rare cases, email clients may not properly parse emails sent by Tableau Server. Many times this can be fixed by setting this property to `true`. Known clients include iOS Mail and Microsoft Outlook (when paired with Exchange S/MIME encryption).

tabadmincontroller.auth.expiration.minutes

Default value: `120`

Controls how long session cookies are valid. By default this is set to 120 minutes. This value also determines how long the embedded credentials in a node bootstrap file are valid. For more information, see tsm topology nodes get-bootstrap-file.

tdsservice.log.level

Version: Added in version 2020.3.0

Default value: `info`

The logging level for the Data Source Properties service. This is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see Change Logging Levels.

tomcat.http.maxrequestsize

Default value: `16380`

The maximum size (bytes) of header content that is allowed to pass through the Apache gateway on HTTP requests. Headers that exceed the value set on this option will result in browser errors, such as HTTP Error 413 (Request Entity Too Large) or authentication failures.

A low value for `tomcat.http.maxrequestsize` may result in authentication errors. Single sign-on solutions that integrate with Active Directory (SAML and Kerberos) often require large authentication tokens in HTTP headers. Be sure to test HTTP authentication scenarios before deploying into production.

We recommend setting `gateway.http.request_size_limit` option to the same value that you set for this option.

tomcat.http.proxyHost

Specifies forward proxy host name for OpenID requests to the IdP. See Configure Tableau Server for OpenID Connect.

tomcat.http.ProxyPort

Specifies forward proxy port for OpenID requests to the IdP. See Configure Tableau Server for OpenID Connect.

tomcat.https.proxyHost

Specifies forward proxy host name for OpenID requests to the IdP. See Configure Tableau Server for OpenID Connect.

tomcat.https.ProxyPort

Specifies forward proxy port for OpenID requests to the IdP. See Configure Tableau Server for OpenID Connect.

tomcat.https.port

Default value: `8443`

SSL port for Tomcat (unused).

tomcat.server.port

Default value: `8085`

Port that tomcat listens on for shutdown messages.

tomcat.useSystemProxies

Default value: `false`

Specifies whether tomcat components (OpenID) require access to the forward proxy configuration on the local Windows operating system. See Configure Tableau Server for OpenID Connect.

tomcatcontainer.log.level

Default value: `info`

The logging level for microservices in the Interactive Microservice Container and Non-Interactive Microservice Container. This is dynamically configurable starting in version 2020.4, so if you are only changing this you do not have to restart Tableau Server. For more information, see Change Logging Levels.

tsm.log.level

Default value: `info`

Logging level for TSM services. These logs include information that can be useful if you have problems with TSM services: Administration Agent, Administration Controller, Client File Service, Cluster Controller, Service Manager, and License Service. This configuration key does not change the logging level for Coordination Service or for maintenance processes. For more information, see Change Logging Levels and Tableau Server Processes.

tsm.controlapp.log.level

Default value: `info`

Logging level for `control_<app>` services. These logs include information that can be useful if you are running into problems starting or reconfiguring a TSM or Tableau Server process. For more information, see Change Logging Levels.

usernotifications.reap_after_days

Default value: `30`

Number of days after which a user notification will be deleted from the server.

vizportal.adsync.update_system_user

Default value: `false`

Specifies whether email addresses and display names of users are changed (even when changed in Active Directory) when an Active Directory group is synchronized in Tableau Server. To ensure that user email addresses and display names are updated during synchronization, set `vizportal.adsync.update_system_user` to `true`, and then restart the server.

vizportal.alwaysUseEmbeddedShareLinks

Version: Added in version 2021.3.0

Default value: `false`

Specifies whether the **Copy Link** option should include the "embed=y" parameter. Starting in version 2019.4, by default it does not include this parameter. Setting this configuration key to true changes the behavior so that the "embed=y" parameter is included. For details about using the **Copy Link** option to share links for embedding in web pages, see Embed Views into Webpages in the Tableau Desktop and Web Authoring Help.

vizportal.art_skip_list

Version: Added in version 2024.2.

Default value: `null`

Use this configuration key to specify aspects of Tableau Server functionality that does not use Activity and Resource Tracing (ART) and will generate large amounts of unnecessary data while ART is enabled.

This key is used together with `vizportal.log_art_java` and `vizportal.enable_art` for troubleshooting issues with Application Server (VizPortal). When set to [need info here

about what it gets set to.] To learn how to use this configuration setting, see Troubleshooting problems with Application Server.

vizportal.commenting.delete_enabled

Default value: `true`

When set to `true`, lets users delete comments on views. You can delete a comment if you created it, are the content owner, a project leader with an appropriate site role, or are an administrator. To learn which site roles are required for full project leader access, see Project-level administration.

vizportal.csv_user_mgmt.index_site_users

Version: Deprecated in version 2022.3. Retired (removed entirely) in version 2023.3.

Default value: `true`

Specifies whether indexing of site users is done user by user when importing or deleting users with a CSV file. When set to `true`(the default) indexing is done as each user is added or deleted. To delay the indexing of the site users until after the entire CSV file has been processed, set this to `false`.

vizportal.csv_user_mgmt.bulk_index_users

Version: Deprecated in version 2022.3. Retired (removed entirely) in version 2023.3.

Default value: `false`

Specifies whether indexing of site users is done in bulk when importing or deleting users with a CSV file. When set to `false`(the default) indexing is done individually. To have the indexing done in bulk after the CSV file has been processed, set this to `true`.

vizportal.enable_art

Version: Added in version 2024.2.

Default value: `false`

This configuration key is used together with `vizportal.log_art_java` and `vizportal.art_skip_list` for troubleshooting issues with Application Server (VizPortal). When set to true, this enables Activity and Resource Tracing in Application Server. To learn how to use this configuration setting, see Troubleshooting problems with Application Server.

vizportal.log_art_java

Version: Added in version 2024.2.

Default value: `false`

This configuration key is used together with `vizportal.enable_art` and `vizportal.art_skip_list` for troubleshooting issues with Application Server (VizPortal). When set to true, this enables Activity and Resource Tracing in Application Server. To learn how to use this configuration setting, see Troubleshooting problems with Application Server.

vizportal.log.level

Default value: `info`

The logging level for vizportal Java components. Logs are written to `/var/opt/tableau/tableau_server/data/tabsvc/logs/vizportal/*.log`.

Set to `debug` for more information. Using the debug setting can significantly impact performance, so you should only use this setting when directed to do so by Tableau Support.

Beginning with version 2020.4.0, this is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see Change Logging Levels.

vizportal.oauth.connected_apps.max_expiration_period_in_minutes

Version: Added in version 2021.4.

Default value: `10`

The maximum period of time, in minutes, the JSON web token (JWT) is valid. At the time the JWT is verified, Tableau Server checks that the time period specified in the JWT doesn't exceed this default value. This setting is used when a Tableau connected app has been configured on Tableau Server using the Tableau REST API.

For example, to change maximum period to 5 minutes, run the following command:

```
tsm configuration set -k vizportal.oauth.external_authorization_
server.max_expiration_period_in_minutes -v 5
```

vizportal.oauth.external_authorization.enabled

Version: Added in version 2021.4.

Default value: `false`

In Tableau Server 2024.2 and later, **Enable connected apps** option is enabled for Tableau Server. In Tableau Server 2023.2 and earlier, specifies whether the **Enable OAuth Access for Embedding Content** option is enabled for Tableau Server.

Use this option to register an external authorization server (EAS) with Tableau Server so that you can enable application integration. For more information, see Configure Connected Apps with OAuth 2.0 Trust.

To enable this option, run the following command:

```
tsm configuration set -k vizportal.oauth.external_author-
ization.enabled -v true
```

vizportal.oauth.external_authorization_server.blocklisted_jws_algorithms

Version: Added in version 2021.4.

Default value: `ES256K`

When an external authorization server (EAS) is registered or connected app is configured, you can use this command to specify the signing algorithm used in JSON web token (JWT)

header. For more information, see Configure Connected Apps with OAuth 2.0 Trust or Use Tableau Connected Apps for Application Integration.

For example, if needed, you might run the following command to remove the algorithm:

```
tsm configuration set - k vizportal.oauth.external_authorization_
server.blocklisted_jws_algorithms -v
```

Important: The example command above allows unsafe signing algorithms and should only be used to troubleshoot errors.

vizportal.oauth.external_authorization_server.issuer

Version: Added in version 2021.4.

Default value: `null`

Required. Use this command to specify the issuer URL. The issuer URL is required to register the external authorization server (EAS) with Tableau Server. For more information, see Configure Connected Apps with OAuth 2.0 Trust.

For example, if your EAS is Okta, you might run a command similar to the following:

```
tsm configuration set -k vizportal.oauth.external_authorization_serv-
er.issuer -v "https://dev-12345678.okta.-
com/oauth2/abcdefg9abc8eFghi76j5"
```

vizportal.oauth.external_authorization_server.jwks

Version: Added in version 2021.4.

Default value: `null`

When an external authorization server (EAS) is registered, you can use this command to specify the JSON web key set (JWKS) URL. The JWKS URL is required if the identity provider (IdP) doesn't expose the external authorization server metadata endpoint.

For example, if your IdP is Amazon Cognito, you might run a command similar to the following:

```
tsm configuration set -k vizportal.oauth.external_authorization_
server.jwks -v "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_Ab129faBb/.well-known/jwks.json"
```

vizportal.oauth.external_authorization_server.max_expiration_period_in_minutes

Version: Added in version 2021.4.

Default value: `10`

The maximum period of time, in minutes, the JSON web token (JWT) is valid. At the time the JWT is verified, Tableau Server checks that the time period specified in the JWT doesn't exceed this default value. This setting is used when an EAS has been registered with Tableau Server. For more information, see Configure Connected Apps with OAuth 2.0 Trust.

For example, to change maximum period to 5 minutes, run the following command:

```
tsm configuration set -k vizportal.oauth.external_authorization_
server.max_expiration_period_in_minutes -v 5
```

vizportal.openid.client_authentication

Specifies custom client authentication method for OpenID Connect.

To configure Tableau Server to use the IdPs that require the `client_secret_post`, set this value to `client_secret_post`.

An example would be when connecting to the Salesforce IDP, which requires this.

vizportal.openid.essential_acr_values

Version: Added in version 2020.4.

Specifies a list of authentication context class reference (ACR) values to provide the OpenID Connect IdP as an essential claim request. The IdP is responsible for ensuring that authentication meets the expected criteria. If the `vizportal.openid.essential_acr_values` configuration key is populated, Tableau Server acts as the relying party and will inspect the

ACR claim in the token response. Tableau Server will only warn if the ACR claim doesn't match the expected configuration key value.

To set this option, enter the ACR values in order of preference, enclosed by double-quotes. You must separate multiple values by a comma and space, as in this example:

```
tsm configuration set -k vizportal.openid.essential_acr_values -v
"value1, value2"
```

vizportal.openid.full_server_request_logging_enabled

Default value: `false`

Specifies whether to do full logging of OpenID activity.

Set this to `true` when troubleshooting OpenID Connect issues to gather more detailed logs and allow you to better troubleshoot.

As with all logging-related configurations, we recommend that after you are finished troubleshooting and collecting logs, you reset this key to its default (`false`). This limits the amount of information logged, and keeps the log file sizes to a minimum.

vizportal.openid.voluntary_acr_values

Version: Added in version 2020.4.

Specifies a list of authentication context class reference (ACR) values to provide the OpenID Connect IdP as a voluntary claim request. The IdP is responsible for ensuring that authentication meets the expected criteria. If the `vizportal.openid.voluntary_acr_values` configuration key is populated, Tableau Server acts as the relying party and will inspect the ACR claim in the token response. The authentication request will fail if the ACR claim is missing or the provided claim value doesn't match the expected configuration key value.

To set this option, enter the ACR values in order of preference, enclosed by double-quotes. You must separate multiple values by a comma and space, as in this example:

```
tsm configuration set -k vizportal.openid.voluntary_acr_values -v
"value1, value2"
```

vizportal.password_reset

Version: Replaces `features.PasswordReset` in version 2024.2.

Default value: `false`

Applies only to servers that use local authentication. Set to `true`to let users reset their passwords with a "Forgot password" option on the sign-in page.

vizportal.rest_api.cors.allow_origin

Specifies the origins (sites) that are allowed access to the REST API endpoints on Tableau Server when `vizportal.rest_api.cors.enabled` is set to `true`. You can specify more than one origin by separating each entry with a comma (,).

```
tsm configuration set -k vizportal.rest_api.cors.allow_origin -v
https://mysite, https://yoursite
```

If `vizportal.rest_api.cors.enabled` is `false`, the origins listed by this option are ignored. For more information, see Enabling CORS on Tableau Server.

> **Note:** You can use an asterisk (*) as a wild card to match all sites. This is not recommended as it allows access from any origin that has access to the server and can present a security risk. Do not use an asterisk (*) unless you fully understand the implications and risks for your site.

vizportal.rest_api.cors.enabled

Default value: `false`

Controls whether Tableau Server allows Cross Origin Resource Sharing (CORS). When set to `true`, the server allows web browsers to access the Tableau REST API endpoints. You

can use this option and the REST API to create custom portals. By default, this functionality is not enabled. To specify which origins (sites) have access, use the `vizportal.rest_api.cors.allow_origin` option. Only the origins specified with this option are allowed to make requests to the Tableau Server REST API. For more information, see Enabling CORS on Tableau Server.

vizportal.site_user_group_count_enabled

Version: Added in version 2022.3.5 and later, 2023.1.0 and later.

Default value: `false`

Controls whether Site Users page will include a column showing the group count for each user.

vizqlserver.allow_insecure_scripts

Default value: `false`

Allows a workbook to be published to the server from Tableau Desktop, and to be opened from the server, even if the workbook contains SQL or R expressions that are potentially unsafe (for example, a SQL expression that could potentially allow SQL injection). When this setting is `false` (the default), publishing a workbook or opening it from the server results in an error message, and the workbook is blocked. Before you set this value to `true` review the Knowledge Base article, Blocking or Allowing Insecure Scripts in Tableau Server.

vizqlserver.browser.render

Default value: `true`

Views under the threshold set by `vizqlserver.browser.render_threshold` or `vizqlserver.browser.render_threshold_mobile` are rendered by the client web browser instead of by the server. See Configure Client-Side Rendering for details.

vizqlserver.browser.render_threshold

Default value: `100`

The default value represents a high level of complexity for a view displayed on a PC. Complexity factors include number of marks, headers, reference lines, and annotations. Views that exceed this level of complexity are rendered by the server instead of in the PC's web browser.

vizqlserver.browser.render_threshold_mobile

Default value: `60`

The default value represents a high level of complexity for a view displayed on a tablet. Complexity factors include number of marks, headers, reference lines, and annotations. Views that exceed this level of complexity are rendered by the server instead of in the tablet's web browser.

vizqlserver.clear_session_on_unload

Default value: `false`

Determines whether or not VizQL sessions are kept in memory when a user navigates away from a view or closes their browser. The default value (false) keeps sessions in memory. To close VizQL sessions on leaving a view or closing a browser, set this to `true`.

vizqlserver.force_maps_to_offline

Version: Added in version 2020.4.0.

Default value: `false`

Determines whether Tableau Server runs in offline mode for maps. This is useful in disconnected environments where access to the internet and the map server is restricted. To enable offline mode for maps, set this value to `true`. For more information about installing and configuring Tableau Server in an environment without internet access, see Install Tableau Server in a Disconnected (Air-Gapped) Environment.

vizqlserver.geosearch_cache_size

Default value: `5`

Sets the maximum number of different geographic search locale/language data sets that can be loaded into server memory at the same time. When the server receives a geographic search request for locale/language data set that is not in memory, it will load the set into memory. If loading the data set will exceed the specified limit, the least recently used locale/language data set is cleared from memory so the requested one can be loaded. The minimum value is 1. Each cache takes approximately 60 MB in memory (so if you set this to 10, the memory usage would be 600 MB (60 * 10).

vizqlserver.initialsql.disabled

Default value: `false`

Specify whether to ignore initial SQL statements for all data sources. Set this to true to ignore initial SQL:

```
tsm configuration set -k vizqlserver.initialsql.disabled -v true
```

vizqlserver.log.level

Default value: `info`

The logging level for VizQL Server Java components. Logs are written to `/var/-opt/tableau/tableau_server/data/tabsvc/logs/vizqlserver/*.log`.

Set to `debug` for more information. Using the debug setting can significantly impact performance, so you should only use it when directed to do so by Tableau Support.

Beginning with version 2020.3.0, this is dynamically configurable, so if you are only changing this you do not have to restart Tableau Server. For more information, see Change Logging Levels.

vizqlserver.NumberOfWorkbookChangesBetweenAutoSaves

Default value: `5`

Auto recover configuration for web authoring. Specifies the number of changes that a user must make to trigger auto save. Take care when changing this value. Auto recover functionality may impact the performance of web authoring and other viz-related operations on

Tableau Server. We recommend tuning this value by making incremental adjustments over time.

vizqlserver_<n>.port

The port a VizQL server instance (specified by "<n>") is running on.

vizqlserver.protect_sessions

Version: Retired in 2024.2.0. Beginning in 2024.2.0, Tableau Server always prevents VizQL sessions from being reused after the original user signs out.

Default value: `true`

When set to `true`, prevents VizQL sessions from being reused after the original user signs out.

vizqlserver.querylimit

Default value: `1800`

Longest allowable time for updating a view, in seconds. 1800 seconds = 30 minutes. This configuration option impacts VizQL Server and Data Server.

vizqlserver.RecoveryAttemptLimitPerSession

Default value: `3`

Auto recover configuration for web authoring. The maximum number of attempts to recover the same session. Take care when changing this value. Auto recover functionality may impact the performance of web authoring and other viz-related operations on Tableau Server. We recommend tuning this value by making incremental adjustments over time.

vizqlserver.session.expiry.minimum

Default value: `5`

Number of minutes of idle time after which a VizQL session is eligible to be discarded if the VizQL process starts to run out of memory.

vizqlserver.session.expiry.timeout

Default value: `30`

Number of minutes of idle time after which a VizQL session is discarded.

vizqlserver.sheet_image_api.max_age_floor

Default value: `1`

The amount of time, in minutes, to cache images that are generated by the Query View Image method of the REST API. For more information, see the REST API Reference in the REST API help.

vizqlserver.showdownload

Default value: `true`

Controls the display of the **Tableau Workbook** option of the Download menu in views. When set to `false`, the Tableau Workbook option is unavailable.

> **Note:** This setting does not remove the option for users in Web Edit mode.

vizqlserver.showshare

Default value: `true`

Controls the display of Share options in views. To hide these options, set to false.

> **Note:** Users can override the server default by setting the "showShareOptions" JavaScript or URL parameter.

vizqlserver.url_scheme_whitelist

Specifies one or more URL schemes to allow (safe list) when using URL actions on views and dashboards. The schemes `http`, `https`, `gopher`, `mailto`, `news`, `sms`, `tel`, `tsc`, and `tsl`

are allowed (safe listed) by default. This command can contain multiple comma and space-separated values, as in this example:

```
tsm configuration set -k vizqlserver.url_scheme_whitelist -v
scheme1, scheme2
```

The values you specify overwrite previous settings. Therefore, you must include the full list of schemes in the `set` command. (You cannot amend the list of schemes by running the `set` command repeatedly.)

vizqlserver.web_page_objects_enabled

Default value: `true`

Controls whether Web Page objects in dashboards can display target URLs. To prevent web pages from appearing, set to false.

vizqlserver.WorkbookTooLargeToCheckpointSizeKiB

Default value: `5120`

Auto recover configuration for web authoring. Size limit (KB) for a workbook that will auto save. Workbooks larger than this value will not be auto-saved. Take care when changing this value. Auto recover functionality may impact the performance of web authoring and other viz-related operations on Tableau Server. We recommend tuning this value by making incremental adjustments over time.

Note: Older versions of Server use a default value: `1024`

vizqlserver.workflow_objects_enabled

Default value: `true`

Determines whether the Tableau External Actions Workflow object can be added to dashboards.

webdataconnector.refresh.enabled

Deprecated. Use `tsm data-access web-data-connectors allow` instead.

Determines whether extract refreshes for web data connectors (WDCs) are enabled in Tableau Server. To disable refresh for all WDCs, set the value for this key to `false`, as shown below:

```
tsm configuration set --key webdataconnector.refresh.enabled --value
false
```

To learn more, see Web Data Connectors in Tableau Server.

webdataconnector.whitelist.fixed

Deprecated. Use `tsm data-access web-data-connectors add` instead.

Specifies one or more web data connectors (WDCs) that can be used by to access data connections that are accessible over HTTP or HTTPS. This command is formatted as JSON data on a single line, with all double-quotes (") escaped using a backslash (\).

For example to add a San Francisco Film Locations WDC to the safe list:

```
tsm configuration set --key webdataconnector.whitelist.fixed --value
"'{\"https://tableau.data.world:443\": {\"properties\": { \"sec-
ondary_whitelist\": [\"(https://data.world/)(.*)\"] } } }'"
```

To learn more, see Web Data Connectors in Tableau Server.

webdataconnector.enabled

Deprecated. Use `tsm data-access web-data-connectors allow` instead.

Default value: `true`

When set to `true`, you can use `tsm` commands to manage web data connectors on the server.

webdataconnector.whitelist.mode

Default value: `mixed`

Determines how Tableau Server can run web data connectors. Supported modes are:

- `mixed`. Users can run connectors that are on an allowlist (safe list) of URLs. This mode originally also allowed users to run WDCs that had been imported. Importing WDCs is no longer supported.
- `fixed`. Users can run connectors that are on an allowlist (safe list) of URLs.
- `insecure`. Users can run any connector.

> **Important:** Use the `insecure` option *only* for development and testing. Because connectors run custom code, running connectors that have not been vetted can pose a security threat.

wgserver.audit_history_expiration_days

Default value: `183`

Specifies the number of days after which historical events records are removed from the PostgreSQL database (the Tableau Server database).

wgserver.authentication.legacy_identity_mode.enabled

Version: Added in version 2022.1

Default value: `false` for Tableau Server 2022.1 and later. For pre-2022.1 Tableau Server deployments upgraded to 2022.1 or later, default value is `true`.

Set to `false` to use identity pools.

For more information, see Troubleshoot identity pools.

wgserver.authentication.identity_pools.default_pool_description

Version: Added in version 2023.1

Default value: Null

Optionally, you can add a description for the initial pool (TSM configured) to the Tableau Server landing page and is visible to all users. When one or more identity pools are created,

this description is added below the primary sign-in option and can be used to help guide users that belong to the initial pool (TSM configured) to the correct sign-in option.

For example, to add a “Regular employees sign in here” description, you can use the following command:

```
tsm configuration set -k wgserver.authentication.identity_pools.de-
fault_pool_description -v “Regular employees sign in here"
```

Note: The initial pool (TSM configured) description is different from the Sign In Customization note. The Sign In Customization note is displayed on the Tableau Server landing page below all sign-in options and on the page where your initial pool (TSM configured) users enter their username and password.

wgserver.change_owner.enabled

Default value: `true`

Controls whether the ownership of a workbook, data source or project can be changed. Other options include `false` and `adminonly`.

wgserver.clickjack_defense.enabled

Default value: `true`

When set to `true`, helps prevents a malicious person from "clickjacking" a Tableau Server user. In a clickjack attack, the target page is displayed transparently over a second page, and the attacker gets the user to click or enter information in the target page while the user thinks he or she is interacting with the second page.

For more information, see Clickjack Protection.

wgserver.domain.accept_list

Version: This was added in version 2020.4.0 and replaces `wgserver.domain.whitelist`.

Default value: null

Allows connection from Tableau Server to secondary Active Directory domains. A secondary domain is one that Tableau Server connects to for user synchronization, but is a domain where Tableau Server is not installed. Tableau Server will attempt to connect to secondary domains for user and group synchronization. In some cases, Tableau Server may be unable to connect to the secondary domain, which will result in the error, "Domain not in accept list (errorCode=101015)."

Setting the `wgserver.domain.accept_list` option is required by a fix for the security vulnerability, [Important] ADV-2020-003: Tableau Server Forced Authentication. As of February 2020, the fix for this vulnerability is included in all latest versions and maintenance releases of Tableau Server.

To set this option, enter the secondary domain enclosed by double-quotes. Multiple domains must be separated by a comma and a space. For example, `tsm configuration set -k wgserver.domain.accept_list -v "example.org, domain.com"`.

Wildcard functionality is not supported. For example, if Tableau connects to `sub-1.example.org` and `sub2.example.org`, then both domains must be added.

Updating the `wgserver.domain.accept_list` option overwrites the existing value. Therefore, if you are adding a new domain to an existing set of domains stored in the value, include all existing domains with the new domain when you set the option. You can retrieve the full list of existing domains by running `tsm configuration get -k wgserver.domain.accept_list`.

wgserver.domain.ldap.domain_custom_ports

Default value: null

Allows you to map child domains and their LDAP ports. Domain and port are separated by a colon (:) and each domain:port pair is separated by a comma (,) using this format: `FQDN1:-port,FQDN2:port`

Example: `tsm configuration set -k wgserver.domain.ldap.domain_custom_ports -v child-domain1.lan:3269,childdomain2.lan:3269,childdomain3.lan:389`

wgserver.domain.password

Default value: null

Specifies password for the user account that is used for LDAP connection. See External Identity Store Configuration Reference.

wgserver.domain.username

Default value: null

Specifies name for the user account that is used for LDAP connection. See External Identity Store Configuration Reference.

wgserver.domain.whitelist

> **Important:** This key has been deprecated as of version 2020.4.0. Use `wgserver.domain.accept_list` instead.

Default value: null

Allows connection from Tableau Server to secondary Active Directory domains. A secondary domain is one that Tableau Server connects to for user synchronization, but is a domain where Tableau Server is not installed. Tableau Server will attempt to connect to secondary domains for user and group synchronization. In some cases, Tableau Server may be unable to connect to the secondary domain, which will result in the error, "Domain not in whitelist (errorCode=101015)."

wgserver.extended_trusted_ip_checking

Default value: `false`

Enforces IP client matching for trusted ticket requests.

wgserver.ignore_domain_in_username_for_matching

Version: Added in versions 2021.4.21, 2022.1.17, 2022.3.9, and 2023.1.5

Default value: `false`

When you enable SAML, you can configure Tableau Server to ignore the domain portion of the SAML username attribute when matching the identity provider (IdP) user name to a user account on Tableau Server. You might ignore the domain portion of the username attribute when you already have users defined in Tableau Server that match the prefix portion of a user-name attribute but not the domain portion of the username attribute. For more information, see the Ignore domain when matching SAML username attribute section in the SAML Requirements topic.

For example, to ignore the domain name in the SAML username attribute, run the following command:

```
tsm configuration set -k wgserver.ignore_domain_in_username_for_
matching -v true
```

Important:

- We do not recommend ignoring the domain name without taking precautions. Specifically, verify that user names are unique across the configured domains that you've created in your IdP.
- This command only works in Tableau Server deployments that are in `legacy-identity-mode` or deployments that have not been updated through the identity migration to use the Identity Service.

wgserver.restrict_options_method

Default value: `true`

Controls whether Tableau Server accepts HTTP OPTIONS requests. If this option is set to `true`, the server returns HTTP 405 (Method Not Allowed) for HTTP OPTIONS requests.

wgserver.saml.blocklisted_digest_algorithms

Version: Added in version 2021.1.

Default value: `SHA1`

Specifies the hashing algorithms that are not allowed for any relevant SAML certificate signatures or SAML assertion digest method or signature methods . When set, certificates or assertions that are signed & hashed with a blocklisted algorithm will be rejected and fail.

There are multiple places where SHA-1 could be used on both the Tableau and IdP side. For example:

- Certificates uploaded with TSM that are used by Tableau Server to sign the request that is sent to the IdP.
- Certificates in the IdP metadata used to verify the AuthnResponse (signature) received from the IdP using the public key in the Certificate.
- Incoming assertions signed and hashed with SHA-1 (DigestMethod set to SHA-1 and SignatureMethod set to SHA-1).

The default value was changed to (`SHA1` in Tableau Server 2021.2. For more information about upgrading to 2021.2 with SAML configured, see the Knowledge Base article, Tableau Server Using SAML Authentication Fails to Start or Rejects Login After Upgrade to Tableau Server 2021.2.

wgserver.saml.forceauthn

Version: Added in version 2019.3.

Default value: `false`

When set to `true`, if the Tableau user session expires, Tableau Server will re-authenticate the user with the IdP. This option can also be used to ask the IdP to prompt the user for re-authentication, even if the user has an active IdP session.

wgserver.saml.idpattribute.username

Specifies the name of the attribute in which your SAML IdP stores user names. By default, this is set to `username`. If the attribute name that your IdP uses contains spaces, enclose it in quotation marks. For more information, see Configure Server-Wide SAML or Configure Site-Specific SAML.

wgserver.saml.iframed_idp.enabled

Default value: `false`

Default of false means that when users select the sign-in button on an embedded view, the IdP's sign-in form opens in a pop-up window.

When you set it to true, and a server SAML user who is already signed in navigates to a web page with an embedded view, the user will not need to sign in to see the view.

You can set this to true only if the IdP supports signing in within an iframe. The iframe option is less secure than using a pop-up, so not all IdPs support it. If the IdP sign-in page implements clickjack protection, as most do, the sign-in page cannot display in an iframe, and the user cannot sign in.

If your IdP does support signing in via an iframe, you might need to enable it explicitly. However, even if you can use this option, it disables Tableau Server clickjack protection for SAML, so it still presents a security risk.

wgserver.saml.maxassertiontime

Default value: `3000`

Specifies the maximum number of seconds, from creation, that a SAML assertion is usable.

wgserver.saml.min_allowed.elliptic_curve_size

Default value: `256`

Version: Added in version 2021.1 but did not include a default value. In 2021.2, the default value was set to `256`.

This option specifies the minimum allowed ECDSA curve size for the certificate used for SAML authentication. If you upload a certificate that has an ECDSA curve size less than 256, TSM will log an error when you apply changes.

If you are upgrading to Tableau Server 2021.2 or later and your SAML certificate uses an ECDSA curve size less than 256, Tableau Server will not start after upgrading. We recommend uploading a new certificate with 256 (or larger) ECDSA curve size before upgrading. Alternatively, you can run this command to set a lower ECDSA curve size on older versions (pre-2021.1) of Tableau Server before you upgrade. If you are running this command on a version prior to 2021.1, you must include the `--force-keys` option with the command. For more information about upgrading to 2021.2 with SAML configured, see the Knowledge Base article, Tableau Server Using SAML Authentication Fails to Start or Rejects Login After Upgrade to Tableau Server 2021.2.

wgserver.saml.min_allowed.rsa_key_size

Default value: `2048`

Version: Added in version 2021.1 but did not include a default value. In 2021.2, the default value was set to `2048`.

This option specifies the minimum allowed RSA key length for the certificate used for SAML authentication. If you upload a certificate that has an RSA key length less than 2048, TSM will log an error when you apply changes.

To run SAML authentication with a 1024 RSA key length (not recommended), set this value to `1024`.

If you are upgrading to Tableau Server 2021.2 or later and your SAML certificate uses a key length less than 2048, Tableau Server will not start after upgrading. We recommend uploading a new certificate with 2048 (or larger) key length before upgrading. Alternatively, you can run this command to set a lower key strength on older versions (pre-2021.1) of Tableau Server before you upgrade. If you are running this command on a version prior to 2021.1, you must include the `--force-keys` option with the command. For more information about upgrading

to 2021.2 with SAML configured, see the Knowledge Base article, Tableau Server Using SAML Authentication Fails to Start or Rejects Login After Upgrade to Tableau Server 2021.2.

wgserver.saml.responseskew

Default value: `180`

Sets the maximum number of seconds difference between Tableau Server time and the time of the assertion creation (based on the IdP server time) that still allows the message to be processed.

wgserver.saml.sha256

Default value: `true`

When set to `true`, Tableau Server will hash message signatures and digests with SHA-256 in SAML assertions to the IdP. Set this option to `false` only if your IdP rejects assertions containing SHA-256 hashed content.

wgserver.session.apply_lifetime_limit

Default value: `false`

Controls whether there is a session lifetime for server sessions. Set this to `true`to configure a server session lifetime.

wgserver.session.idle_limit

Default value: `240`

The number of minutes of idle time before a sign-in to the web application times out.

wgserver.session.lifetime_limit

Default value: `1440`

The number of minutes a server session lasts if a session lifetime is set. The default is 1440 minutes (24 hours). If `wgserver.session.apply_lifetime_limit` is `false` (the default) this is ignored.

wgserver.unrestricted_ticket

Default value: `false`

Specifies whether to extend access to server resources for users authenticated by trusted tickets. Default behavior allows users to access views only. Setting this to `true` allows users with valid trusted tickets to access server resources (projects, workbooks, and so on) as if they had signed in using their credentials.

workerX.gateway.port

Default value: `80` (`443` if SSL)

External port that Apache listens on for workerX (where a “worker” is the term used for subsequent server nodes in the cluster). worker0.gateway.port is Tableau Server’s external port. In a distributed environment, worker0 is the initial Tableau Server node.

workerX.vizqlserver.procs

Default value: <number>

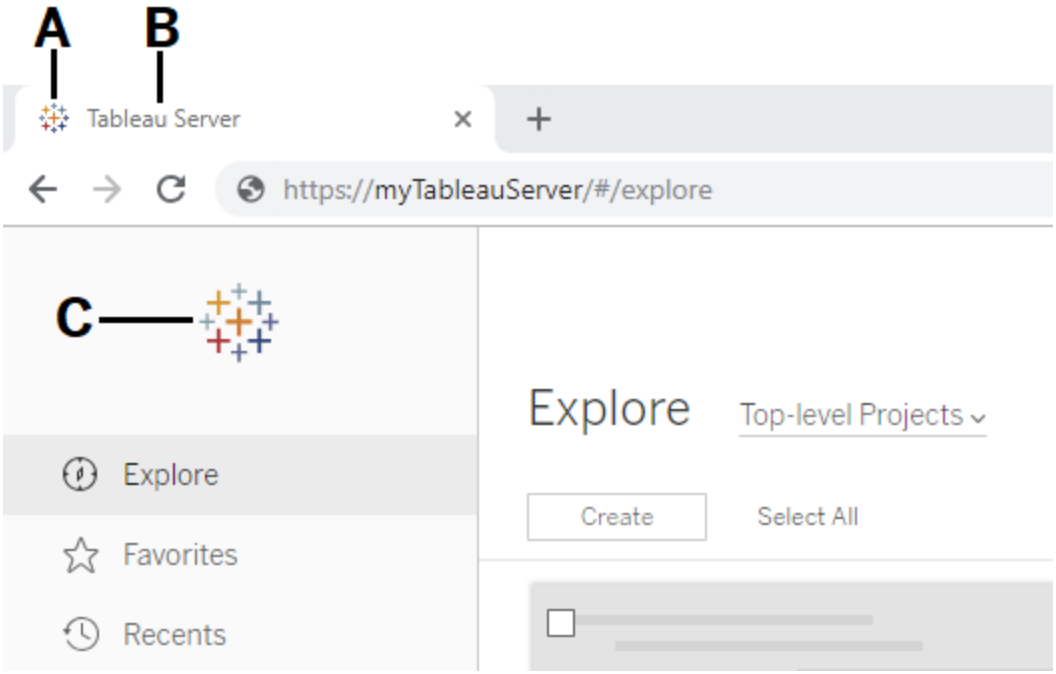
Number of VizQL servers.

zookeeper.config.snapCount

Specifies the number of transactions necessary to cause the Coordination Service to create a snapshot of the logs. By default this value is 100,000 transactions. If your Coordination Service is not writing enough transactions to result in snapshots, the automatic cleanup of snapshots older than five days will not take place, and you may lose disk space to the transaction logs. By default transaction logs and snapshots are created in the Tableau data directory.

tsm customize

You can use the `tsm customize` command to customize the look and feel of the Tableau Server client browser experience.



Logo	Option flag	Minimum size/Maximum size, in pixels	Recommended size, in pixels
A - Window tab logo	cannot be changed	cannot be changed	cannot be changed
B - Server name	`--server-name`	does not apply	does not apply
C - Header logo	`--header-logo`	32 by 32 min, 160 by 160 max	48 by 48
Not shown - Sign in logo	`--signin-logo`	3000 by 3000 max	
Header logo/Sign in logo	`--logo`	32 by 32 min, 160 by 160 max	48 by 48
Not shown - Logo shown when navigation pane is minimized	`--compact-logo`	32 by 32 max	32 by 32

The image files you use should be in GIF, JPEG, or PNG format.

The background colors on the header and sign in page are not the same. If you use the same image for both locations (if you use the `--logo` option, for example) your logo might look different depending on where it appears in the server interface.

As part of your disaster recovery plan, we recommend keeping a backup of the customization image files in a safe location off of the Tableau Server. The image files that you add to Tableau Server will be stored and distributed to other nodes by the Client File Service. However, the files are not stored in a recoverable format. See Tableau Server Client File Service.

Synopsis

```
tsm customize [options] [global options]
```

After you run the `customize` command, you must run the following command to apply changes:

```
tsm pending-changes apply
```

Options

Note: Use the path and image file name cannot include any spaces.

`--compact-logo "<path-to-logo>"`

Optional.

Specify a path to the image file that will be displayed when the navigation pane size is minimized. The maximum (and optimal) size is 32 by 32 pixels.

`--header-logo "<path-to-logo>"`

Optional.

Specify a path to the image file that will be displayed in the header only.

`--logo "<path-to-logo>"`

Optional.

Path to a single image file that will display for both the header and the sign-in window.

`--restore-defaults`

Optional.

Reset all customization options to default install state.

`--server-name <server_name>`

Optional.

Server name that appears in the browser tab, tooltips, and messages.

`--signin-logo "<path-to-logo>"`

Optional.

Specify a path to the image file that will be displayed for sign-in window only.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm data-access

You can use the `tsm data-access` commands to configure data caching, enable or disable data repository access, enable SAML for single sign-on, and configure settings for Web Data Connectors (WDCs).

- caching
  - data-access caching list
  - data-access caching set

- repository
  - repository-access disable
  - repository-access enable
  - repository-access list
- set-saml-delegation
  - set-saml-delegation configure
  - set-saml-delegation disable
  - set-saml-delegation enable
- web-data-connectors
  - web-data-connectors add
  - web-data-connectors allow
  - web-data-connectors delete
  - web-data-connectors list

tsm data-access caching list

Displays data connection caching settings. To learn more about data connection caching settings, see Configure Data Cache.

Synopsis

```
tsm data-access caching list [global options]
```

tsm data-access caching set

Sets data connection caching settings. To learn more about data connection caching settings, see Configure Data Cache.

Synopsis

```
tsm data-access caching set [options] [global options]
```

Options

```
-r, --refresh-frequency
```

Optional.

Sets the frequency to refresh cached data with a new query to the underlying data source. You can specify a number to define the maximum number of minutes that data should be cached. You can also specify **low** to cache and reuse data for as long as possible, or **always** (equivalent to **0**) to refresh data each time that a page is loaded. If this option is not specified, it defaults to **low**.

tsm data-access repository-access disable

Disable external access to the Tableau PostgreSQL database for the default remote user. This will not disable access from localhost.

Synopsis

```
tsm data-access repository-access disable [options] [global options]
```

Options

`--repository-username <username>`

Required.

The username, either **tableau** or **readonly**, with access to the data repository.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1500 (25 minutes).

`--ignore-prompt`

Optional.

Suppress the prompt for restart and restart Tableau Server.

tsm data-access repository-access enable

Enables access to the Tableau PostgreSQL database.

By default, PostgreSQL traffic uses port 8060 (TCP). If you are running a local firewall, be sure to allow traffic for this port. To change the PostgreSQL port, see Ports that are not dynamically mapped.

Synopsis

```
tsm data-access repository-access enable [options] [global options]
```

Options

`--repository-password <password>`

Required.

Sets (or changes) the password to access the data repository for the specified user-name.

`--repository-username <username>`

Required.

The username, either **tableau** or **readonly**, with access to the data repository.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1500 (25 minutes).

`--ignore-prompt`

Optional.

Suppress the prompt for restart and restart Tableau Server.

tsm data-access repository-access list

Lists users who have access to the Tableau PostgreSQL database.

Synopsis

```
tsm data-access repository-access list [global options]
```

tsm data-access set-saml-delegation configure

Setup single sign-on for SAML SAP HANA so that Tableau Server functions as an Identity Provider (IdP) that provides single sign-on for users making SAP HANA data connections.

Synopsis

```
tsm data-access set-saml-delegation configure [options]
[global options]
```

Options

`-kf, --cert-key <cert-key>`

Optional.

The SAML certificate key file.

`-cf, --cert-file <file-path>`

Optional.

The location of the SAML certificate file.

`-uf, --username-format <username-format>`

Optional.

Username format. Valid format keys are: 'username', 'domain_and_username', and 'email'.

`-uc,--username-case <username-case>`

Optional.

Username case. Valid case keys are: 'lower', 'upper', and 'preserve'.

.

tsm data-access set-saml-delegation disable

Disable single sign-on for SAML SAP HANA.

Synopsis

```
tsm data-access set-saml-delegation disable [global options]
```

tsm data-access set-saml-delegation enable

Enable single sign-on for SAML SAP HANA.

Synopsis

```
tsm data-access set-saml-delegation enable [global options]
```

tsm data-access web-data-connectors add

Add a web data connector (WDC) to the WDC safe list.

Synopsis

```
tsm data-access web-data-connectors add [options] [global options]
```

Options

`-n, --name <name>`

Required.

The name for the WDC that will be displayed in the Tableau Server data source list. This name must be enclosed in single quotes (') or double quotes ("). Use double quotes (") if the name includes a space.

`-sec, --secondary <secondary-URL-1>, <secondary-URL-2>`

Required if the WDC uses secondary domains.

A comma-delimited list of URLs that indicates which domains the connector can make requests to or receive data from, for example, external JavaScript libraries, REST APIs, or local files. Do not enclose the URLs in quotes. To add an entire domain to this secondary safe list, you can use a wildcard expression `.*` at the end of the URL, as shown in the following example: `https://www.example.com/.*`

Note that on Windows, you could include parentheses `(.*)` around the wildcard, but it isn't necessary. On Linux, the parentheses will cause an error. Use `.*` as the wildcard.

If you don't know whether the WDC uses secondary domains, or what the secondary domains are, you might need to contact the WDC's developer. You can also choose to use `http://.*` and `https://.*` wildcard URLs to allow all domains. However, to increase security, we strongly recommend that you use more specific URLs.

`--url <URL>`

Required.

The URL for the WDC (formatted as `<scheme>://<host>:<port>/<path>`, for example `https://www.tableau.com:443/example/`). For many WDCs the <port> value is 443, which is the default port used for HTTPS, but you can check the value for your connector by looking at the data source details on Tableau Server or Tableau Cloud. Note that you can't use a wildcard (`.*`) as part of the URL for the WDC.

tsm data-access web-data-connectors allow

Enable or disable WDC refreshes. Also, enable or disable the use of WDCs on Tableau Server.

Synopsis

```
tsm data-access web-data-connectors allow [options] [global options]
```

Options

Use one or both options. At least one of `--refreshes` or `--type` is required.

`-r, --refreshes <refreshes-allowed>`

Optional if `--type` is specified.

Set to `false` to disallow WDC refreshes or `true` to allow WDC refreshes.

`-t, --type <WDC-allowed>`

Optional if `--refreshes` it specified.

Set to `none` to disallow the use of WDCs on Tableau Server (and omit WDCs from backups) or `all` to allow the use of WDCs.

tsm data-access web-data-connectors delete

Delete a specified WDC, or all WDCs, from the Tableau Server safe list.

Synopsis

```
tsm data-access web-data-connectors delete [options]
[global options]
```

Options

`--all`

Optional.

This option will delete all WDCs.

`--url <URL>`

Optional.

The URL for the WDC to delete.

tsm data-access web-data-connectors list

List all WDCs currently on the safe list.

Synopsis

```
tsm data-access web-data-connectors list [options] [global options]
```

Options

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm email

Use the `tsm email` command to view and test your SMTP configuration.

For more information about configuring SMTP, see Configure SMTP Setup.

tsm email test-smtp-connection

Run this command to test the STMP connection. When run, TSM will attempt to establish a connection with the SMTP server that you have configured for Tableau Server. TSM will also return a connection status and the details of the SMTP configuration.

In some cases, the command will return a false-positive status. For example, if your Postfix SMTP server is set to require TLS, but Tableau Server is not configured for TLS, the connection is established and TSM will report a succesful connection. However, in this scenario, Postfix actually rejects the email message after TSM has connected.

Synopsis

```
tsm email test-smtp-connection [global options]
```

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm initialize

You can use the `tsm initialize` command to initialize Tableau Server.

> **Note:** You must apply or discard pending changes before running `tsm initialize` or the initialize will fail. Apply pending changes using the `tsm pending-changes apply` command. Discard any pending changes you do not want to apply using `tsm pending-changes discard`.

Synopsis

```
tsm initialize [options] [global options]
```

Options

`-r,--start-server`

Optional. Leave the server running after initialization is complete.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm jobs

You can use the `tsm jobs` commands to list, reconnect to, and cancel jobs.

tsm jobs cancel

Cancel a job on the server. Any job can be canceled before it starts running (when queued). Only certain jobs can be canceled when they are already running: Cleanup, Decommission

File Store, Generate Backup, Restart Server, Start Server. For more information about canceling jobs, see Cancel TSM Jobs.

Synopsis

```
tsm jobs cancel --id <jobID> [global options]
```

Options

```
-i,--id <jobID>
```

Required.

Id of the job to cancel.

tsm jobs list

List asynchronous jobs on the server.

Synopsis

```
tsm jobs list [--status <status>] [global options]
```

Options

```
-t,--status <status>
```

Optional.

Filter for jobs that match the given status.

tsm jobs reconnect

Reconnect to an asynchronous job to display its progress. If no job id is specified, it reconnects to the latest job.

Synopsis

```
tsm jobs reconnect[--id <jobID>] [global options]
```

Options

`-i,--id <jobID>`

Optional.

Specifies the id of the job that should be reconnected.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm licenses

You can use the `tsm licenses` commands to manage server license tasks like activating or deactivating a Tableau Server product key on- or off-line, and getting associated files for off-line activation or deactivation.

- tsm licenses activate
- tsm licenses atr-configuration get
- tsm licenses atr-configuration set
- tsm licenses deactivate
- tsm licenses get-offline-activation-file
- tsm licenses get-offline-deactivation-file
- tsm licenses list
- tsm licenses refresh

tsm licenses activate

Activates a Tableau Server product key.

Synopsis

```
tsm licenses activate --license-key <product-key> [global options]
```

Options

`-f, --license-file <file.tlf>`

Required if doing offline activation.

Specifies the license file (<file>.tlf) used for offline activation.

`-k, --license-key <product-key>`

Required if activating a valid product key.

Specifies the product key to use for online activation.

`-t, --trial`

Note: This option is only available in versions earlier than 2023.3.0. For trial licensing with later versions, contact your account representative.

Required if activating a trial license.

Activate a trial license.

tsm licenses atr-configuration get

View the Server authentication-to-run (ATR) duration. The ATR duration is the length of time that Tableau Server is authorized to run before the license must be renewed.

Synopsis

```
tsm licenses atr-configuration get --duration [global options]
```

Options

`--duration`

Displays the current authentication-to-run (ATR) in seconds. For example, 432000 (5 days).

tsm licenses atr-configuration set

Set the Server authentication-to-run (ATR) duration. The ATR duration is the length of time that Tableau Server is authorized to run before the license must be renewed.

Synopsis

```
tsm licenses atr-configuration set --duration <duration_in_seconds>
[global options]
```

Options

`--duration <duration_in_seconds>`

Sets the authorization-to-run (ATR) duration (in seconds). For example, 432000 (5 days).

tsm licenses deactivate

Deactivates a Tableau Server product key either online or offline.

Synopsis

```
tsm licenses deactivate --license-key <product-key>
[global options]
```

Options

`-f, --license-file <return_file.tlr>`

Required if doing offline deactivation.

Specifies the license file (<file>.tlf) used for offline deactivation.

`-k, --license-key <product-key>`

Required if deactivating a product key.

Specifies the product key to use for online deactivation.

tsm licenses get-offline-activation-file

Generate an offline activate file to use for activating Tableau Server offline. To learn more, see Activate Tableau Server Offline.

> **Note:** You can only activate one product key at a time unless you're using Server ATR. With Server ATR, you can provide a comma delimited list of product keys in the offline activation file to simultaneously activate multiple product keys.

Synopsis

```
tsm licenses get-offline-activation-file --license-key <product-key>
--output-dir <path> [global options]
```

Options

`-k, --license-key <product-key>`

Required.

Specifies the product key to use for offline activation.

`-o, --output-dir <path>`

Required.

The location where the offline activation file should be saved. This location must exist.

tsm licenses get-offline-deactivation-file

Generate an offline deactivation file to use for deactivating Tableau Server offline. To learn more, see Deactivate Tableau Server Offline.

Synopsis

```
tsm licenses get-offline-deactivation-file --license-key <product-key> --output-dir <path> [global options]
```

Options

`-k, --license-key <product-key>`

Required.

Specifies the product key to use for offline deactivation.

`-o, --output-dir <path>`

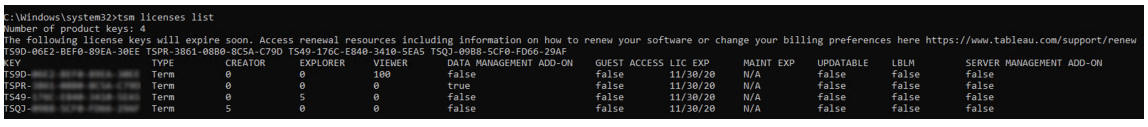
Required.

The existing location where the offline deactivation file should be saved.

tsm licenses list

Lists licenses that are activated on the Tableau Server deployment.

For example, a server with five Creator licenses, five Explorer licenses, 100 Viewer licenses, and Data Management would provide command output similar to the following:



The following fields are returned:

- **KEY**: A globally unique 16-character string that identifies the license.
- **TYPE**: Describes the type of license
  - Term: Term licenses map to a subscription schedule and must be renewed. The expiration date is listed under the LIC EXP field.
  - Perpetual: Perpetual licenses are purchased once and do not need to be renewed but must be refreshed to update the MAINT EXP or maintenance expiration date.
  - Cores: Core licenses are licenses that map to the number of cores on the computers running specific Tableau Server services. Core licensing allows for a guest user access to views on the server or embedded on other web servers. Core licenses also allow for unlimited Explorer and Viewer users.
- **CREATOR**: The number of Creator licenses issued to the Tableau Server deployment.
- **EXPLORER**: The number of Explorer licenses issued to the Tableau Server deployment.
- **VIEWER**: The number of Viewer licenses issued to the Tableau Server deployment.
- **DATA MANAGEMENT**:Tableau Server is licensed with Data Management (`True`/`False`). See About Data Management.
- **GUEST ACCESS**: Tableau Server is licensed for a Guest User. See Guest User. The ability to leverage a Guest User requires Core licensing. See TYPE field.
- **LIC EXP**: The date that the license expires and Tableau Server will stop working. Term licenses expire. See TYPE field. Visit the Tableau Customer Portal to refresh licenses.
- **MAINT EXP**: Applies only to legacy perpetual licenses (TYPE = Perpetual). For Term licenses, this field will output, `N/A`. MAINT EXP displays the date that the maintenance contract for the Tableau Server deployment expires. To update the license maintenance key see Refresh Expiration Date and Attributes for the Product Key. Visit the Tableau Customer Portal to view maintenance purchase history and to purchase additional maintenance.
- **UPDATABLE**: Specifies whether the license is an updatable subscription license (`True`/`False`).
- **LBLM**: Specifies if login-based license management (LBLM) is enabled for the Tableau Server deployment (`True`/`False`). When enabled, LBLM allows users to log into Tableau Server to license their instance of Tableau Desktop or Prep, rather than entering a product key. For more information about LBLM, see Login-based License Management.
- **SERVER MANAGEMENT**: Tableau Server is licensed for Advanced Management (formerly Server Management Add-on) (`True`/`False`). For more information about

Advanced Management, see About Tableau Advanced Management on Tableau Server.

Synopsis

```
tsm licenses list [global options]
```

tsm licenses refresh

Update the maintenance expiration date of all product keys on Tableau Server.

Synopsis

```
tsm licenses refresh [global options]
```

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm login

Use the `tsm login` command to log in to Tableau Services Manager from a remote node.

If the account you are logged in as is a member of the TSM-authorized group, you do not need to provide credentials to run commands when running tsm CLI locally. For more information, see Authenticating with tsm CLI.

Synopsis

```
tsm login [global options]
```

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm logout

You can use the `tsm logout` command to log out of Tableau Services Manager (TSM).

Synopsis

```
tsm logout [global options]
```

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm maintenance

You can use the `tsm maintenance` commands to manage server maintenance tasks like creating backups or restoring Tableau Server from a previously created backup.

- tsm maintenance backup
- tsm maintenance cleanup
- tsm maintenance jmx
  - tsm maintenance jmx disable
  - tsm maintenance jmx enable
- tsm maintenance metadata-services
  - tsm maintenance metadata-services disable
  - tsm maintenance metadata-services enable
  - tsm maintenance metadata-services get-status
- tsm maintenance reindex-search
- tsm maintenance reset-searchserver
- tsm maintenance restore
- tsm maintenance send-logs
- tsm maintenance snapshot-backup (external file store)
  - tsm maintenance snapshot-backup complete
  - tsm maintenance snapshot-backup prepare
  - tsm maintenance snapshot-backup restore
- tsm maintenance validate-backup-basefilepath

- tsm maintenance validate-resources
- tsm maintenance ziplogs

tsm maintenance backup

Creates a backup of the data managed by Tableau Server. This data includes the Tableau PostgreSQL database (the repository) which contains workbook and user metadata, and extract (.hyper files, and .tde files for versions 2024.2 and older) files. This data does not include configuration data. See Perform a Full Backup and Restore of Tableau Server.

Note: Do not use this command on Tableau Server installations with External File Store. See Backup and Restore with External File Store.

Optimizing with topology configurations:

- Co-locating File Store on the same node as the Administration Controller can reduce the length of time it takes to back up Tableau Server by reducing or eliminating the need to transfer data between nodes during the backup process. This is especially true if your organization uses many extracts.
- Co-locating the repository (pgsql) with the Administration Controller node can also help to reduce back up time, but the time savings is less significant than that of the File Store.

The Administration Controller is usually on the initial node, unless you have had an initial node failure and moved the controller to another node.

The backup file is assembled in a temporary location in the data directory and then written to the directory defined in the TSM `basefilepath.backuprestore` variable:

`/var/opt/tableau/tableau_server/data/tabsvc/files/backups/<file-name>.tsbak`

For more information about where backup files are written, and how to change that location, see tsm File Paths. **Note:** Even when you change the backup location, the backup process uses a temporary location in the data directory to assemble the backup file.

Synopsis

```
tsm maintenance backup --file <backup_file> [options]
[global options]
```

Options

`-f, --file <backup_file>`

Required.

For more information about backing up the repository data, see Back up Tableau Server Data for more information.

`-d, --append-date`

Optional.

Append the current date to the end of the backup file name.

Note: Backups created using a schedule automatically include a date/time at the beginning of the backup file name. If you also use the `-d` option your file name will include the date twice. For more information about scheduling backups, see Scheduling and Managing Backups.

`-i, --description <string>`

Optional.

Include the specified description of the backup file.

`--ignore-prompt`

Optional. Added in version 2020.2.

Back up without prompting, even if the File Store is not on the same node as the Administration Controller (usually the initial node). Use this prompt if automating backups (for example, with scripts).

`-k, --skip-verification`

Optional.

Do not verify the integrity of the database backup.

`--multithreaded`

Optional. Added in version 2021.1

By default, a single thread is used when creating a Tableau Server backup. When this option is specified, a backup is created using multiple threads. Two threads are used by default, when this option is specified. You can change the number of threads used by setting the `backup.zstd.thread_count` configuration key.

`--override-disk-space-check`

Optional.

Attempt to create a backup even when there is a low disk space warning.

`-po, --pg-only`

Optional.

Generates only the repository backup.

Important: Do not use the `pg-only` option when generating a backup unless instructed by Tableau Support. This option will only back up the repository and *cannot* be used to restore your Server. Its primary use is for troubleshooting, and Tableau Support will ask you to create a `--pg-only` back up if this is necessary.

`--request-timeout <timeout in seconds>`

Optional.

Number of seconds to wait for the command to finish. Default value is 86400 (1440 minutes).

`-sd, --schedule-days <day[,day]>`

Optional. Added in version 2020.4.

Days on which to run the schedule. Use 1-7 for weekly schedule (1 for Monday, 7 for Sunday), 1-31 for monthly schedules (if a month does not include the specified day, the last day of the month is used). Separate multiple values with commas.

`-si, --schedule-id <ID>`

Optional. Added in version 2020.4.

Specify the ID of an existing schedule you want to update.

`--skip-compression`

Optional.

Create a backup without using compression. This results in a larger backup file but can reduce the amount of time it takes to complete the backup. If using this in a multi-node installation, we strongly recommend you have a File Store instance configured on your initial node.

`-sn, --schedule-name <name>`

Optional. Added in version 2020.4.

Specify the name for a schedule you are creating or updating.

`-sr, --schedule-recurrence <frequency>`

Optional. Added in version 2020.4.

Frequency of schedule recurrence. Valid options are "daily", " weekly", or "monthly".

`-st, --schedule-time <HH:MM>`

Optional. Added in version 2020.4.

The time a schedule should be run, in 24-hour format: HH:MM.

Examples

This example creates a backup called `ts_backup-<yyyy-mm-dd>.tsbak` in the `/var/-opt/tableau/tableau_server/data/tabsvc/files/backups/` directory:

```
tsm maintenance backup -f ts_backup -d
```

This example creates a recurring weekly backup schedule named "weekly-saturday-backup" that runs every Saturday at noon and creates a backup called `<yyyy.mm.dd.hh.mm>-ts_saturday_backup.tsbak`:

```
tsm maintenance backup -f ts_saturday_backup -sr weekly -st 12:00 -sd 6 -sn weekly-saturday-backup
```

For more details on managing scheduled backups, see Scheduling and Managing Backups.

tsm maintenance cleanup

By default the `tsm maintenance cleanup` command deletes temporary files and log files older than one day. Command options can modify retention length and which files are deleted.

The impact of this command depends on whether Tableau Server is running.

- If the server is running, most old files and `http_requests` table entries can be deleted, but any files in use (locked by the operating system) cannot be deleted, so temporary files and active log files are not removed. To delete temporary files and current log files, you must stop the server before running this command. To delete `http_requests` table entries, use the `-q` option.
- If the server is stopped `http_requests` entries cannot be deleted.

If you are running Tableau Server on a distributed deployment, run this command on the node that is running the Administration Controller (also referred to as the *TSM Controller*) process. By default and in most cases, the controller is on the initial node in the cluster.

Note: This command was added in Tableau Server version 10.5.1 and some options were added in version 2018.1.

Synopsis

```
tsm maintenance cleanup [options] [global options]
```

Options

`-a, --all`

Optional.

Perform all cleanup operations with default retention values. Equivalent to running the `cleanup` command with the following options: `-l -t -r -q -ic`.

`--http-requests-table-retention <# of days>`

Optional.

Default: 7 days

Specify the number of days of `http_requests` table entries that should be retained. Use this option with the `-q` option to specify the number of days of table entries to retain, overriding the default of 7 days. This option specifies table entry retention age but does not trigger actual deletion of table entries. Use this together with the `-q` option, which triggers deletion of entries.

`-ic, --sheet-image-cache`

Optional. Added in version 2019.4.

Clear the image cache. This cache can contain images for offline previews, snapshots for subscription email messages, and subscription pdfs, as well as any images requested from the publish rest API endpoint (see rest_api_ref.htm for more information).

`-l, --log-files`

Optional.

Delete log files that are older than 1 day. Files in the subdirectories under `data/tabsvc/logs` will be deleted.

`--log-files-retention <# of days>`

Optional.

Default: 1 (24 hours)

Delete logs older than this number of days. Use this to override the default retention period of 1 day. This command does not apply to temporary files.

`-q, --http-requests-table`

Optional.

Delete old `http_requests` table entries. Tableau Server must be running for table entries to be deleted. This option is ignored if Tableau Server is stopped. This option can be used alone to specify deletion of entries older than the default retention period (7 days), or together with the `--http-requests-table-retention` to specify a non-default retention period.

Note: Deleting `http_requests` table entries permanently removes data that is available to custom administrative views. Be sure removing this data will not impact any custom views you need.

`-r, --redis-cache`

Optional.

Clear the Redis cache.

`--request-timeout <timeout in seconds>`

Optional.

Default: 3600

Wait the specified amount of time for the command to finish.

`-t, --temp-files`

Optional.

Delete all files and subdirectories in the following directories:

- `/var/opt/tableau/tableau_server/data/tabsvc/temp`: To delete files under each Tableau Server process directory, you must stop Tableau Server before running the command. If you execute the command while Tableau Server is running, only directories storing files for expired (not running) sessions are deleted.
- `/var/opt/tableau/tableau_server/data/tabsvc/httpd/temp`
- `/var/opt/tableau/tableau_server/temp`

Examples

This example cleans up all log files older than 2 days old:

```
tsm maintenance cleanup -l --log-files-retention 2
```

tsm maintenance jmx disable

Version: Command added in version 2022.1.

Use the `tsm maintenance jmx disable` command to disable JMX on Tableau Server.

Running this command requires a restart of Tableau Server, including TSM services.

Synopsis

```
tsm maintenance jmx disable [options] [global options]
```

Options

`--ignore-prompt`

Optional.

Disable JMX without prompting.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance jmx enable

Version: Command added in version 2022.1.

Use the `tsm maintenance jmx enable` command to enable JMX on Tableau Server.

This command is interactive, prompting you for applicable options if you don't provide them as command line parameters.

This command requires a restart of Tableau Server, including TSM services.

Synopsis

```
tsm maintenance jmx enable
```

Options

`--access <readonly | readwrite>`

Optional.

Enable JMX with either readonly or readwrite access. Default is `readonly`.

`--ignore-prompt`

Optional.

Enable JMX without prompting for additional security options. This enables JMX with SSL and restarts Tableau Server if you do not include any other command options. For example:

```
tsm maintenance jmx enable --ignore-prompt
```

`--no-ssl`

Optional.

Enable JMX without SSL.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish.

`--ssl-client-auth`

Optional.

Enable JMX requiring client auth for SSL.

`--unsecured`

Optional.

Enable JMX with no security features enabled (this is not recommended). This is the equivalent of setting the `service.jmx_enabled` configuration key to `true` in pre-2022.1 versions.

tsm maintenance metadata-services disable

Version: Command added in version 2019.3.

Use the `tsm maintenance metadata-services disable` command to disable the Tableau Metadata API.

Disabling the Metadata API stops continuous ingestion and indexing of information about the content on Tableau Server, deletes the index of information about the content published to Tableau Server and assets associated with that content, and disables the ability to both query the Metadata API and access Tableau Catalog.

Running this command stops and starts some services used by Tableau Server, which causes certain functionality, such as Recommendations, to be temporarily unavailable to your users.

Synopsis

```
tsm maintenance metadata-services disable
```

Option

`--ignore-prompt`

Optional.

Dismiss the confirmation prompt when disabling the Metadata API.

tsm maintenance metadata-services enable

Version: Command added in version 2019.3.

Use the `tsm maintenance metadata-services enable` command to enable the Tableau Metadata API for Tableau Server.

If Tableau Server is licensed with the Data Management, enabling the Metadata API enables Tableau Catalog.

When enabling the Metadata API, information about the content on Tableau Server is ingested and then indexed to the Metadata API Store. The Metadata API can be used to query schema, lineage, and user managed metadata about the content published to Tableau Server. After the Metadata API is enabled, metadata is continuously ingested and indexed until the Metadata API is disabled.

When running this command, keep the following in mind:

- This command stops and starts some services used by Tableau Server, which causes certain functionality, such as Recommendations, to be temporarily unavailable to your users.
- A new index of metadata is created and replaces the previous index every time this command is used.

For more information about the Tableau Catalog, see, About Tableau Catalog.

Synopsis

```
tsm maintenance metadata-services enable
```

Option

--ignore-prompt

Optional.

Dismiss the confirmation prompt when enabling the Metadata API.

tsm maintenance metadata-services get-status

Version: Command added in version 2019.3.

Use the `tsm maintenance metadata-services get-status` command to get status information on Metadata Services.

Status on Metadata Services indicates if the Metadata API Store has been initialized or if the Tableau Metadata API is running or not.

Synopsis

```
tsm maintenance metadata-services get-status
```

tsm maintenance reindex-search

Use the `tsm maintenance reindex-search` command to rebuild the search index.

Synopsis

```
tsm maintenance reindex-search [options] [global options]
```

Option

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance reset-searchserver

Version: This command was retired (removed) in 2023.3.0 when Search and Browse (also called Search Server) was retired. Search and Browse has been replaced by Index and Search Server.

Note: Running this command on version 2023.3.0 or later will not do anything.

Resets the search server to a clean state, deleting search information and rebuilding the search index.

Synopsis

```
tsm maintenance reset-searchserver [options] [global options]
```

Option

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance restore

Restore Tableau Server using the specified backup file. Restoring a backup file does not restore any configuration data. See Perform a Full Backup and Restore of Tableau Server.

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

Beginning with version 2022.3, backups created using tabadmin ("pre-TSM backups") are not supported. You cannot restore a pre-TSM backup to Tableau Server version 2022.3 or later.

Synopsis

```
tsm maintenance restore --file <file_name> [--restart-server]
[global options]
```

Options

`-f, --file <file_name>`

Required.

Specifies the backup file to restore from.

The `restore` command expects a backup file in the directory defined in the TSM `basefilepath.backuprestore` variable. By default:

`/var/opt/tableau/tableau_server/data/tabsvc/files/backups/`

For more information about file paths and how to change them, see tsm File Paths.

`-ak, --asset-key-file <file_name>`

Optional. Deprecated in version 2021.4.0.

Specify this option only if you are restoring from assets that were created by tabadmin on Tableau Server (versions 2018.1 and earlier).

Name of asset key file to restore from. The asset key file is created by the `tabadmin assetkeys` command. The file must be in the predefined backup/restore location on the server.

`-k, --skip-identity-store-verification`

Optional. Specify this option only if you are restoring from a backup file that was created by tabadmin on Tableau Server (versions 2018.1 and earlier).

Do not use this key in an attempt to change identity store type from Tableau Server that created original backup file. To change the identity store, see Changing the Identity Store .

`-po, --pg-only`

Optional.

Restores only the repository.

`-r, --restart-server`

Optional.

Restart the server after the restore.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance send-logs

Upload the specified file to Tableau and associate it with a support case. To successfully upload files to Tableau, your Tableau Server must be able to communicate with the send-logs server at `https://report-issue.tableau.com`.

Synopsis

```
tsm maintenance send-logs --case <case_number> --email <contact_
email> --file <path/to/file> [global options]
```

Options

`-c,--case <case_number>`

Required.

Support case number.

`-e,--email <contact_email>`

Required.

Contact email.

`-f, --file <path/to/file>`

Required.

Specifies the location and name of the log file archive to send.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance snapshot-backup complete

Version: Command added in version 2020.1 and only available when Tableau Server is configured for External File Store.

Complete the snapshot backup process on Tableau Server. Run this after you have taken a snapshot backup of your external storage.

The *tsm maintenance snapshot-backup prepare* and the *tsm maintenace snapshot-backup complete* commands are used to create a backup of Tableau Server data for Tableau Server installations that are configured with External File Store. For more information, see Backup and Restore with External File Store

Synopsis

```
tsm maintenance snapshot-backup complete [options] [global options]
```

Options

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance snapshot-backup prepare

Version: Command added in version 2020.1 and only available when Tableau Server is configured for External File Store.

Prepares for snapshot backup. Once the preparation step is complete, you may take a snapshot backup of your network storage.

The `tsm maintenance snapshot-backup prepare` and the `tsm maintenace snapshot-backup complete` commands are used to create a backup of Tableau Server data for Tableau Server installations that are configured with External File Store. For more information, see Backup and Restore with External File Store

Synopsis

```
tsm maintenance snapshot-backup prepare [options] [global options]
```

Options

`--include-pg-backup`

Optional. Added in version 2021.1. Prior versions always included a backup of the External Repository.

Optional.

A backup of the Repository is made and copied to the network share. This is only applicable to deployments where both External Repository and External File Store are configured. For more information, see Backup and Restore with External File Store.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance snapshot-backup restore

Version: Command added in version 2020.1 and only available when Tableau Server is configured for External File Store.

Restores the repository backup from the storage snapshot to Tableau Server.

For more information, see Backup and Restore with External File Store.

Synopsis

```
tsm maintenance snapshot-backup restore [options] [global options]
```

Options

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance validate-backup-basefilepath

Version: Command added in version 2022.1.

Validate that the backup/restore base filepath location has correct permissions to allow backup and restore functions to work properly. Run this after setting the basefilepath for backup and restore. For more information, see Change the current file location .

Synopsis

```
tsm maintenance validate-backup-basefilepath [options]
[global options]
```

Options

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance validate-resources

Validate workbooks and data sources for a site. Use this command before migrating a site, to detect issues with site resources such as workbooks and data sources that will cause a site import to fail. Some resource problems can be corrected by republishing from local sources. Other problems might require assistance from Tableau Support.

Synopsis

```
tsm maintenance validate-resources --site-id <site ID>
[global options]
```

Options

`-id,--site-id <site ID>`

Required.

ID for the site whose resources you are validating.

`-r,--repair`

Optional.

Attempt to repair invalid resources. Those that cannot be repaired are noted in output.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish.

tsm maintenance ziplogs

Use the `ziplogs` command to create an archive of Tableau Server log files.

Note: If you cannot run the ziplogs command successfully, you can manually zip the Tableau Server logs. For more information, see Troubleshoot Tableau Server on Linux.

Synopsis

```
tsm maintenance ziplogs [options] [global options]
```

Options

`-a, --all`

Optional.

Include all files except PostgreSQL data.

`-d, --with-postgresql-data`

Optional.

Include the PostgreSQL data folder if Tableau Server is stopped or PostgreSQL dump files if Tableau Server is running. This flag is ignored for deployments running with a Tableau Server External Repository.

`--enddate "<mm/dd/yyyy H:mm>"`

Optional. Time option (`H:mm`) added in version 2021.4.0.

The last date of log files to be included. This option must be used with `--startdate` and cannot be used with `--minimumdate`. If this option is not specified, up to two days of logs will be included, starting at 00:00 GMT.

If you include the time option you must use quotes around date and time. The time option uses GMT, however, the resulting log files will be written using the local time zone of the Tableau Server machine.

Example: If the local time zone of the Tableau Server machine is PDT and you want the log files to begin at 7am PDT and end at 7pm PDT on 07/28/2022, use the following:

```
tsm maintenance ziplogs -f logs.zip --startdate "07/28/2022
14:00" --enddate "07/29/2022 02:00"
```

`-f, --file <name>`

Optional.

Specify a name for the zipped archive file. If no name is provided the archive is created as logs.zip. The file is written to the directory defined in the TSM `base-filepath.log_archive` variable. By default:

`/var/opt/tableau/tableau_server/data/tabsvc/files/log-archives/`

For more information about file paths and how to change them, see tsm File Paths.

`-i, --description <string>`

Optional.

Include the specified description of the archive file.

`-l, --with-latest-dump`

Optional.

When any service crashes, Tableau Server generates a dumpfile. Set this option to include the most recent service crash dumpfile.If you do not set this option, then no dumpfile will be included in the resulting ziplog.

`-m, --minimumdate <mm/dd/yyyy>`

Optional.

Earliest date of log files to be included. If not specified, a maximum of two days of log files are included. Format of date should be "`mm/dd/yyyy`". This option cannot be used with `--startdate` and `--enddate` or `--all`.

`--nodes`

Optional. Added in version 2020.3.

Specify the nodes for which to create a ziplog file. If not specified, ziplog files are created for all nodes and saved to the initial node. Separate nodes with a comma. For example, to gather logs for nodes 2 and 4:

```
tsm maintenance ziplogs --nodes node2,node4
```

`-o, --overwrite`

Optional.

For an overwrite of an existing ziplog file. If a file by the same name already exists and this option is not used, the ziplogs command will fail.

By default the file is written to:

`/var/opt/tableau/tableau_server/data/tabsvc/files/log-archives/`

For more information about file paths and how to change them, see tsm File Paths.

`--request-timeout <seconds>`

Optional.

Number of seconds to wait for the command to finish. Default value is 7200 (120 minutes).

`--startdate "<mm/dd/yyyy H:mm>"`

Optional. Time option (`H:mm`) added in version 2021.4.0.

The last date of log files to be included. This option must be used with `--enddate` and cannot be used with `--minimumdate`. If this option is not specified, up to two days of logs will be included, starting at 00:00 GMT.

If you include the time option you must use quotes around date and time. The time option uses GMT, however, the resulting log files will be written using the local time zone of the Tableau Server machine.

Example: If the local time zone of the Tableau Server machine is PDT and you want the log files to begin at 7am PDT and end at 7pm PDT on 07/28/2022, use the following:

```
tsm maintenance ziplogs -f logs.zip --startdate "07/28/2022
14:00" --enddate "07/29/2022 02:00"
```

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm pending-changes

Use the `tsm pending-changes` commands to apply, discard, or view pending configuration and topology changes to Tableau Server.

Passwords and secrets that you enter during TSM configuration are encrypted after you save them. Secrets remain encrypted until, during, and after you apply pending changes. For more information about secret storage, see Manage Server Secrets.

- tsm pending-changes apply
- tsm pending-changes discard
- tsm pending-changes list

tsm pending-changes apply

Use the `tsm pending-changes apply` command to apply pending configuration and topology changes to Tableau Server.

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt.

Synopsis

```
tsm pending-changes apply [global options]
```

Options

`-iw, --ignore-warnings`

Optional.

Ignore warning level constraints.

`--ignore-prompt`

Optional.

Suppress the prompt for restart. This only suppresses the prompt. The restart behavior is unchanged.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

tsm pending-changes discard

Use the `tsm pending-changes discard` command to discard pending configuration and topology changes to Tableau Server.

Synopsis

```
tsm pending-changes discard [options] [global options]
```

Options

`--config-only`

Optional.

Discard only pending configuration changes.

`--topology-only`

Optional.

Discard only pending topology changes.

tsm pending-changes list

Lists pending configuration and topology changes to Tableau Server. Any changes that do not require a server restart will be listed as not requiring a restart. If none of the pending changes require a restart, a message displays saying the changes do not require a server restart. If any change in the list requires a restart, the entire list of pending changes will result in a restart. For more information on dynamic configuration or topology changes, see Tableau Server Dynamic Topology Changes.

Synopsis

```
tsm pending-changes list [options] [global options]
```

Options

`--config-only`

Optional.

List only pending configuration changes.

`--topology-only`

Optional.

List only pending topology changes.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm register

You can use the `tsm register` command to register Tableau Server. This command lets you either display the registration file template (using the `--template` option) or provide the path to a completed registration file (using the `--file` option). You must use one of these two options when calling the `tsm register` command.

Synopsis

```
tsm register --template | --file <registration-filename>
[global options]
```

Options

`--file <registration-filename>`

Required.

Path to the file that contains the registration data.

`--template`

Required.

Display registration filetemplate.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm reset

Use the `tsm reset` command to clear the initial admin user so that you can enter a new one. After you run `tsm reset` you must rerun the `tabcmd initialuser` command to create a new initial admin. The new name cannot be the same username as the previous admin user.

If your organization is using Active Directory or LDAP for the Tableau identity store, then the account and password you specify must match an account in the directory.

Synopsis

```
tsm reset[option] [global options]
```

Option

`-d,--delete-all-sessions`

Optional.

Delete all active user sessions when the server is reset.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm restart

You can use the `tsm restart` command to restart Tableau Server. The command stops the server if necessary, and then starts it.

Synopsis

```
tsm restart [global options]
```

Option

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm schedules

Note: These commands only apply to backup jobs scheduled using the tsm maintenance backup command. They do not apply to jobs run by the backgrounder (ex: extract refreshes, flows, subscriptions).

You can use the `tsm schedules` commands to manage scheduled backup jobs. To create a scheduled backup job, use the `tsm maintenance backup` command. For details, see tsm maintenance backup. For more details on managing scheduled backups, see Scheduling and Managing Backups.

- tsm schedules delete
- tsm schedules list
- tsm schedules resume
- tsm schedules suspend
- tsm schedules update

tsm schedules delete

Delete the specified schedule.

Synopsis

```
tsm schedules delete [-si <scheduleID> | -sn <scheduleName>]
[global options]
```

Options

`-si,--schedule-id <scheduleID>`

Required if --schedule-name is not used.

Id of the schedule to delete.

`-sn,--schedule-name <scheduleName>`

Required if --schedule-id is not used.

Name of the schedule to delete.

tsm schedules list

List schedules on the server.

Synopsis

```
tsm schedules list [--next-run | --schedule-id <scheduleID> | --
schedule-name <scheduleName>] [global options]
```

Options

`-nr,--next-run`

Optional.

Sort the schedules by their "next run" time, earliest to latest.

`-si,--schedule-id <scheduleID>`

Optional.

Id of the schedule to list the details of.

`-sn,--schedule-name <scheduleName>`

Optional.

Sort the schedules by their names, in alphabetical order.

tsm schedules resume

Resume the specified suspended schedule.

Synopsis

```
tsm schedules resume [-si <scheduleID> | -sn <scheduleName>]
[global options]
```

Options

`-si,--schedule-id <scheduleID>`

Required if --schedule-name is not used.

Id of the schedule to resume.

`-sn,--schedule-name <scheduleName>`

Required if --schedule-id is not used.

Name of the schedule to resume.

tsm schedules suspend

Suspend the specified schedule.

Synopsis

```
tsm schedules suspend [-si <scheduleID> | -sn <scheduleName>]
[global options]
```

Options

`-si,--schedule-id <scheduleID>`

Required if --schedule-name is not used.

Id of the schedule to suspend.

`-sn,--schedule-name <scheduleName>`

Required if --schedule-id is not used.

Name of the schedule to suspend.

tsm schedules update

Update the specified schedule.

Synopsis

```
tsm schedules update[-si <scheduleID> -st <time_to_run> -sr <recur-
rence_frequency> -sd <day-or-days>] [global options]
```

Options

`-si,--schedule-id <scheduleID>`

Required.

Id of the schedule to update.

`-sr,--schedule-recurrence <frequency>`

Required.

Recurrence frequency of the schedule. Valid options are "daily", "weekly", or "monthly".

`-st,--schedule-time <HH:MM>`

Required.

The time a schedule should be run, in 24-hour format: HH:MM.

`-sd,--schedule-days <day[,day]>`

Optional.

Days on which to run the schedule. For weekly schedules, use 1-7 where 1 is Monday, and 7 is Sunday. For monthly schedules, use 1-31. If a day doesn't exist for a specific month (30 for February, for example) the last valid day of the month is used. Separate multiple values with commas.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm security

Use the `tsm security` commands to configure Tableau Server support for external (gateway) SSL or repository (Postgres) SSL. Repository SSL configuration includes the option to enable SSL over direct connections from Tableau clients—including Tableau Desktop, Tableau Mobile, and web browsers—to the repository.

- tsm security authorize-credential-migration
- tsm security cancel-credential-migrations
- tsm security custom-cert
  - tsm security custom-cert add
  - tsm security custom-cert delete
  - tsm security custom-cert list
- tsm security custom-indexandsearch-ssl
  - tsm security custom-indexandsearch-ssl add
  - tsm security custom-indexandsearch-ssl list

- tsm security custom-tsm-ssl
  - tsm security custom-tsm-ssl disable
  - tsm security custom-tsm-ssl enable
  - tsm security custom-tsm-ssl list
- tsm security external-ssl
  - tsm security external-ssl disable
  - tsm security external-ssl enable
  - tsm security external-ssl list
- tsm security kms
  - tsm security kms set-mode aws
  - tsm security kms set-mode azure
  - tsm security kms set-mode local
  - tsm security kms status
- tsm security maestro-rserve-ssl
  - tsm security maestro-rserve-ssl disable
  - tsm security maestro-rserve-ssl enable
- tsm security maestro-tabpy-ssl
  - tsm security maestro-tabpy-ssl disable
  - tsm security maestro-tabpy-ssl enable
- tsm security regenerate-internal-tokens
- tsm security repository-ssl
  - tsm security repository-ssl disable
  - tsm security repository-ssl enable
  - tsm security repository-ssl get-certificate-file
  - tsm security repository-ssl list
- tsm security rotate-coordination-service-secrets
- As of the 2020.2 release, to configure Rserve and TabPy analytics extensions, use the Tableau Server admin pages. See Configure Connections with Analytics Extensions.

Prerequisites

Before you configure SSL, you must acquire certificates, and then copy them to the computer that runs the Tableau Server gateway process. Additional preparation is required for enabling direct connections from clients. To learn more, see the following articles:

Configure SSL for External HTTP Traffic to and from Tableau Server

Configure SSL for Internal Postgres Communication

For information about mutual (two-way) SSL, see Configure Mutual SSL Authentication and tsm authentication mutual-ssl commands.

tsm security authorize-credential-migration

Authorizes a Tableau user to migrate embedded credentials from a Tableau Server installation to a Tableau Cloud site using Content Migration Tool. Both Tableau Server and Tableau Cloud must have an Advanced Management license to migrate content. For more information, see Migrate Workbooks and Data Sources with Embedded Credentials.

You can cancel authorization using the `tsm security cancel-credential-migrations` command.

Synopsis

```
tsm security authorize-credential-migration --source-site-url-namespace <Tableau Server site ID> --destination-site-url-namespace <Tableau Cloud site ID> --destination-server-url <Tableau Cloud site url> --authorized-migration-runner <username> --destination-public-encryption-key <public key>
```

Options

`--source-site-url-namespace`

Required. Site ID of the Tableau Server site. The site ID is used in the URL to uniquely identify the site.

For example, a site named West Coast Sales might have a site ID of west-coast-sales.

`--destination-site-url-namespace`

Required. Site ID of the Tableau Cloud site. The site ID is used in the URL to uniquely identify the site.

`--destination-server-url`

Required. URL of the pod that your Tableau Cloud site is deployed to. The URL you specify must include a trailing slash (`/`).

Your pod is shown in the first portion of the site URL after signing in to Tableau Cloud. For example, `https://10az.online.tableau.com/` is the United States - West (10AZ) pod. For more information about pods, see the Salesforce Trust page.

`--authorized-migration-runner`

Required. Username of the Tableau Server user authorized to migrate embedded credentials.

`--destination-public-encryption-key`

Required. Specify the public key generated on the Tableau Cloud site.

`--expiration-time-in-days`

Optional. Number of days before authorization expires. Default value is 7 days.

Version: Retired in version 2023.1. Beginning in 2023.1.0 this option is no longer valid and will generate an error if used. The expiration value is hard-coded as 7 days.

Example

The following example authorizes user “admin” to migrate workbooks and published data sources with embedded credentials from Tableau Server site “ExampleA” to Tableau Cloud site “ExampleB”. The authorization will expire in 9 days.

```
tsm security authorize-credential-migration --source-site-url-
namespace ExampleA --destination-site-url-namespace ExampleB --des-
tinationServerUrl https://10ay.online.tableau.com/ --authorized-
```

```
migration-runner admin --destination-public-encryption-key <public
key> --expiration-time-in-days 9
```

tsm security cancel-credential-migrations

Cancels granted authorizations for migrating embedded credentials using Content Migration Tool. For more information, see Migrating Workbooks and Data Sources with Embedded Credentials.

Synopsis

```
tsm security cancel-credential-migrations --source-site-url-
namespace <Tableau Server site ID>
```

Options

`--source-site-url-namespace`

Required. Site ID of the Tableau Server site. The site ID is used in the URL to uniquely identify the site.

For example, a site named West Coast Sales might have a site ID of west-coast-sales.

tsm security custom-cert add

Adds a custom CA certificate to Tableau Server. This certificate is optionally used to establish trust for TLS communication between a SMTP server and Tableau Server.

If a custom certificate already exists, this command will fail. You can remove the existing custom certificate using the `tsm security custom-cert delete` command.

Note: The certificate that you add with this command may be used by other Tableau Server services for TLS connections.

As part of your disaster recovery plan, we recommend keeping a backup of the certificate file in a safe location off of the Tableau Server. The certificate file that you add to Tableau Server will

be stored and distributed to other nodes by the Client File Service. However, the file is not stored in a recoverable format. See Tableau Server Client File Service.

Synopsis

```
tsm security custom-cert add --cert-file <file.crt>
[global options]
```

Options

`-c, --cert-file <file.crt>`

Required. Specify the name of a certificate file in valid PEM or DER format.

tsm security custom-cert delete

Removes the server's existing custom certificate. Doing this allows you to add a new custom certificate.

Synopsis

```
tsm security custom-cert delete[global options]
```

tsm security custom-cert list

List details of custom certificate.

Synopsis

```
tsm security custom-cert list[global options]
```

tsm security custom-indexandsearch-ssl add

Add custom certificates for Index and Search Server for Tableau Server 2023.1 and newer. The SSL implementation is based on Opensearch.org TLS implementation. See Configuring TLS certificates for more information.

`--admin <file.crt>`

Required.

Admin certificate file. Specify the path to a valid PEM-encoded x509 certificate with the extension .crt.

`--admin-key <file.key>`

Required.

Specify the path to a valid RSA or DSA private key file (PKXA #8), with the extension .key by convention.

`-- ca <file.crt>`

Required.

Trusted CA file. Specify the path to a valid PEM-encoded x509 certificate with the extension .crt.

`--node <file.crt>`

Required.

Node certificate file. Specify the path to a valid PEM-encoded x509 certificate with the extension .crt. This command will distribute this certificate to each node in the cluster. Use a wild card certificate to allow the full array of node Distinguished Names (DNs) in a single certificate.

`-- node-key <file.key>`

Required.

Specify the path to a valid RSA or DSA private key file (PKXA #8), with the extension .key by convention.

Synopsis

```
tsm security custom-indexandsearch-ssl add --node <file.crt> --admin
 <file.crt> --node-key <file.key> --admin-key <file.key> --ca
<file.crt> [parameters] [global options]
```

tsm security custom-indexandsearch-ssl list

List details of Index and Search Server SSL custom certificate configuration.

Synopsis

```
tsm security custom-indexandsearch-ssl list[global options]
```

tsm security custom-tsm-ssl disable

Disable the custom SSL certificate for connections to TSM Controller. Revert back to an automatically-managed, self-signed certificate.

Synopsis

```
tsm security custom-tsm-ssl disable [global options]
```

tsm security custom-tsm-ssl enable

Enable the custom SSL certificate for connections to TSM Controller for Tableau Server 2023.1 and newer. If you have already enabled SSL and need to update an expired certificate, use this command.

`-cf,--cert-file <file.crt>`

Required.

Specify the path to a valid PEM-encoded x509 certificate with the extension .crt. The subject name on certificate must match host name or IP address of the Tableau computer where the Administration Controller is running. By default, the Administration Controller runs on the initial node of a Tableau Server deployment.

`-kf,--key-file <file.key>`

Required.

Specify the path to valid RSA or DSA private key file (PKXA #8), with the extension .key by convention. This key cannot be passphrase-protected.

`--chain-file <file.crt>`

Optional.

Specify the path to a certificate chain file (.crt)

The chain file is a concatenation of all the certificates that form the certificate chain for the server certificate.

All certificates in the file must be x509 PEM-encoded and the file must have a .crt extension (not .pem).

`--skip-validation`

Optional
Pass this option to skip certificate authority root verification.

Synopsis

```
tsm security custom-tsm-ssl enable --key-file <file.key> --cert-file
<file.crt> [global options]
```

tsm security custom-tsm-ssl list

List details of TSM custom certificate configuration.

Synopsis

```
tsm security custom-tsm-ssl list[global options]
```

tsm security external-ssl disable

Removes the server's existing SSL configuration settings and stops encrypting traffic between external clients and the server.

Synopsis

```
tsm security external-ssl disable [global options]
```

tsm security external-ssl enable

Enable and specify certificate and key files for SSL over external HTTP communication.

Synopsis

```
tsm security external-ssl enable --cert-file <file.crt> --key-
file <file.key> [options] [global options]
```

Options

`--cert-file <file.crt>`

Required. Specify the name of a valid PEM-encoded x509 certificate with the extension .crt.

`--key-file <file.key>`

Required. Specify a valid RSA or DSA private key file, with the extension .key by convention.

`--chain-file <chainfile.crt>`

Specify the certificate chain file (.crt)

A certificate chain file is required for Tableau Desktop on the Mac. In some cases, a certificate chain file may be required for Tableau Mobile.

Some certificate providers issue two certificates for Apache. The second certificate is a chain file, which is a concatenation of all the certificates that form the certificate chain for the server certificate.

All certificates in the file must be x509 PEM-encoded and the file must have a .crt extension (not .pem).

`--passphrase`

Optional. Passphrase for the certificate file. The passphrase you enter will be encrypted while at rest.

Note: If you create a certificate key file with a passphrase, you cannot reuse the SSL certificate key for SAML.

`--protocols <list protocols>`

Optional. List the Transport Layer Security (TLS) protocol versions you want to allow or disallow.

TLS is an improved version of SSL. Tableau Server uses TLS to authenticate and encrypt connections. Accepted values include protocol versions supported by Apache. To disallow a protocol, prepend the protocol version with a minus (-) character.

Default setting: `"all, -SSLv2, -SSLv3"`

This default explicitly does not allow clients to use SSL v2 or SSL v3 protocols to connect to Tableau Server. However, we recommend that you also disallow TLS v1 and TLS v1.1.

Before you deny a specific version of TLS, verify that the browsers from which your users connect to Tableau Server support TLS v1.2. You might need to preserve support for TLSv1.1 until browsers are updated.

If you do not need to support TLS v1 or v1.1, use the following command to allow TLS v1.2 (using the value `all`), and explicitly deny SSL v2, SSL v3, TLS v1, and TLS v1.1.

```
tsm security external-ssl enable --cert-file file.crt --key-
file file.key --protocols "all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1"
```

tsm security external-ssl list

Displays a list of settings related to the configuration of gateway external SSL. The list includes the names of the certificate files in use, but not their location.

Synopsis

```
tsm security external-ssl list [global options]
```

tsm security kms set-mode aws

Set the KMS mode to AWS.

You will need the full ARN string from AWS KMS. This string is in the "General configuration" section of the AWS KMS management pages. The ARN is presented in this format: arn:aws:kms:<region>:<account>:key/<CMK_ID>, for example, `arn:aws:kms:us-west-2:867530990073:key/1abc23de-fg45-6hij-7k89-1l0mn1234567`.

For more information, see AWS Key Management System.

Synopsis

```
tsm security kms set-mode aws --key-arn "<arn>" --aws-region
"<region>" [global options]
```

Options

`--key-arn`

Required. The `--key-arn` option takes a direct string copy from the ARN in the "General configuration" section of the AWS KMS management pages.

`--aws-region`

Required. Specify a region as shown in the Region column in the Amazon API Gateway table.

Example

For example, if your AWS KMS instance is running in `us-west-2` region, your account number is `867530990073`, and your CMK key is `1abc23de-fg45-6hij-7k89-1l0mn1234567`, then the command would be:

```
tsm security kms set-mode aws --aws-region "us-west-2" --key-arn
"arn:aws:kms:us-west-2:867530990073:key/1abc23de-fg45-6hij-7k89-
1l0mn1234567"
```

tsm security kms set-mode azure

Set the KMS mode to Azure Key Vault.

> **Note:** The KMS mode will display as "Azure Key Vault" when you run `tsm security kms status`, but you set it as "azure".

You will need the name of the Azure key vault and the name of the key in Azure.

For more information, see Azure Key Vault.

Synopsis

```
tsm security kms set-mode azure --key-name "<key_name>" --vault-name "<vault_name>" [global options]
```

Options

`--key-name`

Required. The name of the asymmetric key stored in the Azure Key Vault.

`--vault-name`

Required. Name of the Azure Key Vault.

Example

For example, if your Azure Key Vault is named `tabsrv-keyvault` and your key is `tabsrv-sandbox-key01`, then the command would be:

```
tsm security kms set-mode azure --key-name "tabsrv-sandbox-key01" --vault-name "tabsrv-keyvault"
```

tsm security kms set-mode local

Set or reset the KMS mode to local. Local is the default KMS mode. For more information, see Tableau Server Key Management System.

Synopsis

```
tsm security kms set-mode local [global options]
```

tsm security kms status

View the status of KMS configuration. The status returned includes:

- Status: OK indicates that the KMS is accessible by Tableau, or by the controller node if a multi-node installation.
- Mode: Local, AWS, or Azure Key Vault. Indicates what KMS mode is being used.
- Encrypt and decrypt master encryption key:

  KMS stores a collection of master extract keys (MEKs). Each MEK has:

  - An ID, for example, 8ddd70df-be67-4dbf-9c35-1f0aa2421521
  - Either a “encrypt or decrypt key” or “decrypt-only key” status. If a key is "encrypt or decrypt", Tableau Server will encrypt new data with it. Otherwise, the key will only be used for decryption
  - A creation timestamp, for example, "Created at: 2019-05-29T23:46:54Z."
  - First transition to encrypt and decrypt: a timestamp indicating when the key became an encrypt or decrypt key.
  - Transition to decrypt-only: a timestamp indicating when the key transitioned to decrypt-only.

Other values returned depend on the KMS mode.

When the KMS mode is AWS, the following is returned:

- The ARN (ID) of the customer master key (CMK) .
- The region the CMK is in.
- The ID of the root master key (RMK) in use. The RMK is a key that is encrypted by the CMK. Tableau Server decrypts the CMK by making calls to AWS KMS. The RMK is

then used to encrypt/decrypt the master extract key (MEK). The RMK can change, but there will be only one at a time.

When the KMS mode is Azure Key Vault, the following is returned:

- Vault name: The name of the Azure key vault.
- Azure Key Vault key name: The name of the key in the vault.

Synopsis

```
tsm security kms status [global options]
```

tsm security maestro-rserve-ssl disable

Disable the Rserve connection.

For more information, see Use R (Rserve) scripts in your flow.

tsm security maestro-rserve-ssl enable

Configure a connection between an Rserve server and Tableau Server version 2019.3 or later.

For more information, see Use R (Rserve) scripts in your flow.

Synopsis

```
tsm security maestro-rserve-ssl enable --connection-type <maestro-
rserve-secure | maestro-rserve> --rserve-host <Rserve IP address or
host name> --rserve-port <Rserve port> --rserve-username <Rserve
username> --rserve-password <Rserve password> --rserve-connect-
timeout-ms <RServe connect timeout>
```

Options

`--connection-type`

Select `maestro-rserve-secure` to enable a secure connection or `maestro-rserve` to enable an unsecured connection. If you select `maestro-rserve-secure`,

specify the certificate file path in the command line.

`--rserve-host`

Host

`--rserve-port`

Port

`--rserve-username`

Username

`--rserve-password`

Password

`--rserve-connect-timeout-ms`

The connect timeout in milliseconds. For example `--rserve-connect-timeout-ms 900000`.

tsm security maestro-tabpy-ssl disable

Disable the TabPy connection.

For more information, see Use Python scripts in your flow.

tsm security maestro-tabpy-ssl enable

Configure a connection between a TabPy server and Tableau Server version 2019.3 or later.

For more information, see Use Python scripts in your flow.

Synopsis

```
tsm security maestro-tabpy-ssl enable --connection-type <maestro-tabpy-secure | maestro-tabpy> --tabpy-host <TabPy IP address or host name> --tabpy-port <TabPy port> --tabpy-username <TabPy username> --tabpy-password <TabPy password> --tabpy-connect-timeout-ms <TabPy connect timeout>
```

Options

`--connection-type`

Select `maestro-tabpy-secure` to enable a secure connection or `maestro-tabpy` to enable an unsecured connection. If you select `maestro-tabpy-secure`, specify the certificate file -cf<certificate file path> in the command line.

`--tabpy-host`

Host

`--tabpy-port`

Port

`--tabpy-username`

Username

`--tabpy-password`

Password

`--tabpy-connect-timeout-ms`

The connect timeout in milliseconds. For example `--tabpy-connect-timeout-ms 900000`.

tsm security regenerate-internal-tokens

This command performs the following operations:

1. Stops Tableau Server if it is running.
2. Generates new internal SSL certificates for Postgres repository the search server.
3. Generates new passwords for all of the internally managed passwords.
4. Updates all Postgres repository passwords.
5. Generates a new encryption key for asset key management and encrypts the asset key data with the new key.
6. Generates a new encryption key for configuration secrets (master key) and encrypts the configuration with it.
7. Reconfigures and updates Tableau Server with all of these secrets. In a distributed deployment, this command also distributes the reconfiguration and updates across all nodes in the cluster.
8. Regenerates a new master key, adds it to the master keystore file, and then creates new security tokens for internal use.
9. Starts Tableau Server.

If you plan to add a node to your cluster after you have run this command, then you will need to generate a new node configuration file to update the tokens, keys, and secrets that are generated by this command. See Install and Configure Additional Nodes.

For more information about internal passwords see Manage Server Secrets.

Synopsis

```
tsm security regenerate-internal-tokens [options] [global options]
```

Options

`--ignore-prompt`

Optional.

Perform a restart (if necessary) without prompting. This option only suppresses the prompt. The restart behavior is unchanged.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

tsm security repository-ssl disable

Stop encrypting traffic between the repository and other server components, and stop support for direct connections from Tableau clients.

Synopsis

```
tsm security repository-ssl disable [global-options]
```

tsm security repository-ssl enable

When the repository is local, enables SSL and generates the server's .crt and .key files used for encrypted traffic between the Postgres repository and other server components.

Starting in version 2021.4, when using an external repository, imports the server's .crt and 'key files used to encrypt traffic between external PostgreSQL repository and Tableau Server components.

Enabling this also gives you the option to enable SSL over direct connections from Tableau clients to the server.

Synopsis

```
tsm security repository-ssl enable [options] [global options]
```

Options

`-i, --internal-only`

Optional. This option only applies when the repository is local to Tableau Server and is not configured external to Tableau Server. This option should not be used for Tableau Server configured with External Repository.

When set to `--internal-only`, Tableau Server uses SSL between the repository and other server components, and it supports but does not require SSL for direct connections through **tableau** or **readonly** users.

If this option is not set, Tableau Server requires SSL for traffic between the repository and other server components, as well as for direct connections from Tableau clients (for connections through the **tableau** or **readonly** users).

When you specify this option, you must also complete the steps described in Configure Postgres SSL to Allow Direct Connections from Clients.

`-c, --certificate`

Optional. Added in version 2021.4. This option is only applicable to Tableau Server configured with External Repository and can be used to enable or disable SSL connections post installation.

This option allows you to enable the use of SSL/TSL connections between Tableau Server and the External Repository. When using this option, provide the full path to the SSL certificate file including the file name for the External Repository. This file is the same as the one used when enabling the external repository.

tsm security repository-ssl get-certificate-file

Get the public certificate file used for SSL communication with the Tableau repository. SSL must be enabled for repository communication before you can retrieve a certificate. The

certificate file is distributed automatically to internal clients of the repository in the Tableau Server cluster. To enable remote clients to connect over SSL to the repository, you must copy the public certificate file to each client.

This command works only for Tableau Server that uses a local Repository and will result in an error when Tableau Server is configured with an External Repository.

Synopsis

```
tsm security repository-ssl get-certificate-file [global-options]
```

Options

`-f, --file`

Required.

Full path and file name (with .cert extension) where the certificate file should be saved. If a duplicate file exists it will be overwritten.

tsm security repository-ssl list

Returns the existing repository (Postgres) SSL configuration.

Synopsis

```
tsm security repository-ssl list [global-options]
```

tsm security rotate-coordination-service-secrets

Version: Added in version 2022.1

Generates new certificates, keys, and trust stores used by the Coordination Service for secure connections.

Synopsis

```
tsm security rotate-coordination-service-secrets [options]
[global options]
```

Options

`--coord-svc-restart-timeout <seconds>`

Optional.

Wait the specified number of seconds for Coordination Service to restart. Default: 1200 (20 minutes).

`--ignore-prompt`

Optional.

Perform a restart (if necessary) without prompting.

`--request-timeout <seconds>`

Optional.

Wait the specified number of seconds for the command to finish. Default: 1800 (30 minutes).

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm settings

You can use the `tsm settings` commands to export (get) and import (set) configuration values.

- tsm settings clone
- tsm settings export
- tsm settings import

Important: The server configuration file referenced in this topic includes a copy of the master keystore file used for encrypting configuration secrets. We strongly recommend

that you take additional measures to secure the node configuration file, using mechanisms as described in Securing secrets for import and export operations.

tsm settings clone

Create a "clone payload" that consists of the Tableau Server installation's configuration and topology (including external services and ports). This payload can be used to recreate an exact copy of the Server installation. This command is designed to work best with a Server installation that includes an external repository and external filestore. For details on using the clone payload to create a copy of an installation, see Clone Tableau Server.

Synopsis

```
tsm settings clone --output-directory <output-directory>
[global options]
```

Options

`-d, --output-directory <output-directory>`

Required.

Specifies the location to which the clone payload will be written.

tsm settings export

Export the current server configuration and topology to a file.

The following files are not exported or imported with the `tsm settings import` or `tsm settings export` commands. You must manage these files manually:

- SAML certificate file
- SAML key file
- SAML IdP metadata file
- OpenID.static.file

- Kerberos.keytab file
- LDAP Kerberos keytab file
- LDAP Kerberos conf file
- Mutual SSL certificate file
- Mutual SSL revocation file
- Customization header logo file
- Customization sign-in logo file
- Customization compact logo file

Synopsis

```
tsm settings export --output-config-file <path/to/output_file.json>
[global options]
```

Options

`-f, --output-config-file <file>`

Required.

Specifies the location and name of the file created by this operation.

tsm settings import

Import server configuration or topology.

The following files are not exported or imported with the `tsm settings export` or `tsm settings import` commands. You must manage these files manually:

- SAML certificate file
- SAML key file
- SAML IdP metadata file
- OpenID.static.file
- Kerberos.keytab file
- LDAP Kerberos keytab file
- LDAP Kerberos conf file
- Mutual SSL certificate file
- Mutual SSL revocation file
- Customization header logo file

- Customization sign-in logo file
- Customization compact logo file

Synopsis

```
tsm settings import --import-config-file <path/to/import_file.json>
[global options]
```

Options

`-f,--import-config-file <FILE>`

Required.

Path to input file.

`--config-only`

Optional.

`--topology-only`

Optional.

`-frc, --force-keys`

Optional.

Force a key to be added to configuration even if it did not previously exist.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm sites

You can use the `tsm sites` commands to export an existing site for import to a new site (also referred to as site migration), and to import the new site. An `unlock` command is available in case an error leaves a site locked.

The `tsm sites` commands will use your local file store to hold the export and import data. If you are running a multinode Tableau cluster, then you must run the `tsm sites` commands on a Tableau Server that is running the Data Engine process. For information about the Data Engine process and the processes that require it, see Tableau Server Processes.

Note: When migrating sites between instances of Tableau Server, the target site must be on a version of Tableau Server that is the equal to or later than the version of Tableau Server for the source site. Both the source and target sites must be from supported versions of Tableau Server.

For comprehensive steps for migrating a site, see Export or Import a Site.

- tsm sites export
- tsm sites import
- tsm sites import-verified
- tsm sites unlock

tsm sites export

Export a specified Tableau Server site to a .zip file. You can export a site to archive its settings at a specific point in time, or to complete the first step of a site migration process.

Note: The `tsm sites import` and `tsm sites export` commands can leave a site in a locked state if an error occurs. To unlock a site, use the `tsm sites unlock` command.

Synopsis

```
tsm sites export --site-id <source-siteID> --file <export-file>
[options] [global options]
```

Options

`-f,--file <export-file>`

Required.

Specify the name of the file to which Tableau Server saves all of the site's information.

This file is generated to the directory defined in the TSM `basefilepath.site_export.exports` variable. By default:

`/var/opt/tableau/tableau_server/data/tabsvc/files/siteexports`

For more information about file paths and how to change them, see tsm File Paths.

`-id,--site-id <source-siteID>`

Required.

The site ID for the site you are exporting. You can get the site ID from the URL when you're signed in to the site from a web browser. For information about locating the site ID, see Prepare the Source and Target Sites.

`-ow,--overwrite`

Optional.

Overwrite an export file of the same name that already exists.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 43200 (720 minutes).

tsm sites import

This command uses the .zip file you created using `tsm sites export` to generate a set of .csv files that show how the exported source site settings will map to the new target site.

By default, the .zip file is generated and saved to the `siteexports` directory at:

`/var/opt/tableau/tableau_server/data/tabsvc/files/siteexports`

Before you use this command, you must copy the .zip file to the directory in which Tableau will expect it. This location is defined in the TSM `basefilepath.site_import.exports` variable. By default, the import directory is:

`/var/opt/tableau/tableau_server/data/tabsvc/files/siteimports`

For more information about file paths and how to change them, see tsm File Paths.

> **Note**: The `tsm sites import` and `tsm sites export` commands can leave a site in a locked state if an error occurs. To unlock a site, use the `tsm sites unlock` command.

Synopsis

```
tsm sites import --file <export-file.zip> --site-id <target-siteID>
[options] [global options]
```

Options

`-f,--file <export-file.zip>`

Required.

Name of the .zip file created by the `tsm sites export` process, and which you must copy to the import directory. By default:

`/var/opt/tableau/tableau_server/data/tabsvc/files/siteimports`

`-id,--site-id <target-siteID>`

Required.

The site ID for the new site you are importing to (the target site). For information about locating the site ID, see Prepare the Source and Target Sites.

`-c,--continue-on-ignorable-errors`

Optional.

Continue site import if errors occur which can be ignored. These errors can indicate issues with the import of a specific workbook or data source.

`-k,--no-verify`

Optional.

Skip verification of mapping files.

`-m, --override-schedule-mapper <mapping-file.csv>`

Optional.

Schedule mapping file to override the normal mapping by name.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 7200 (120 minutes).

tsm sites import-verified

Specify the directory that contains an exported site's .csv mapping files, to import to a new site. This is the final step of a site migration process.

Synopsis

```
tsm sites import-verified --import-job-dir <importjob-directory> --
site-id <target-siteID> [options] [global options]
```

Options

`-id,--site-id <target-siteID>`

Required.

The site ID for the new site you are importing to (the target site). For information about locating the site ID, see Prepare the Source and Target Sites.

`-w, --import-job-dir <importjob-directory>`

Required.

The parent of the `mappings` directory that contains the .csv files from the exported (source) site. The name of this parent directory includes the import id and date and time. For example:

```
/var/opt/tableau/tableau_server-
/data/tabsvc/files/siteimports/working/import_ff00_
20180102022014457
```

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 7200 (120 minutes).

tsm sites unlock

Use this command to unlock a site.

Options

`-id,--site-id <target-siteID>`

Required.

The site ID for the site you are unlocking. For information about locating the site ID, see Prepare the Source and Target Sites.

`-d, --desired-state <state to leave unlocked site in>`

Optional.

The state the site should be left in after it is unlocked. Options are "active" and "suspended". The default is "active" if not specified.

For example:

```
tsm sites unlock -id mysite -d suspended
```

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 300 (5 minutes).

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm start

You can use the `tsm start` command to start Tableau Server. If the server is already running this command does nothing.

Synopsis

```
tsm start [option][global options]
```

Option

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm status

You can use the `tsm status` command to display the status of Tableau Server and individual services (processes) that run as part of Tableau Server.

Synopsis

```
tsm status [global options]
```

Options

`-v, --verbose`

Optional.

Display status for every node in the Tableau Server cluster.

`tsm status` will return one of these potential statuses for a Tableau Server node:

- `RUNNING`: The node is running without error statuses for any service or process.
- `DEGRADED`: The node is running with one or more primary services - such as the repository - in an error state. If you have a single instance of the Messaging service and it fails, Tableau Server will continue to function but the status shows as degraded and event messages may be lost. For more information, see Tableau Server Messaging

Service.

- `ERROR`: All primary services or processes are in an error state on the node.
- `STOPPED`: The node is stopped, with no error statuses.

When running `tsm status` with the `--verbose`option, TSM will return a status for each individual service (process). Possible status messages include:

- `is running`: The service is running.
- `status is unavailable`: The status cannot be determined - such as when services are starting up.
- `is in a degraded state`: The service is running, but returning errors. This status indicates the service failed to install properly, has not been configured, or has failed in some way.
- `is in an error state`: The service is running, but returning errors. This status indicates the service failed to install properly, or has not been configured.
- `is synchronizing`: The File Store process is synchronizing with another instance of File Store.
- `is decommissioning`: The File Store process is being decommissioned.
- `is running (Active Repository)`: The active repository is running. This is the expected status.
- `is running (Passive Repository)`: The passive repository is running. This is the expected status when there are two repositories configured.
- `is stopped`: The service is stopped. This does not mean a service is in an error or problem state. Some services only run when needed (the Database Maintenance service for example).

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm stop

You can use the `tsm stop` command to stop Tableau Server. If Tableau Server is already stopped, this command does nothing.

Synopsis

```
tsm stop [option] [global options]
```

Options

`--ignore-node-status <nodeID>`

Optional.

Ignore the status for the specified node or nodes when determining if the server has stopped. Useful if removing a bad node. Separate multiple nodes with commas.

For example, if nodes 2, 3 and 5 are not properly responding: `tsm stop --ignore-node-status node2,node3,node5`

Note: Option added in version 2020.1

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm topology

Use the `tsm topology` commands to prepare File Store nodes for safe removal or to put them back into read-write mode. You can also initiate a repository failover, get a list of nodes or ports, get the bootstrap configuration file required to add additional nodes to your cluster, remove nodes, configure external repository, and external File Store.

Important: When making changes to topology, you need to apply those pending changes for the changes to take effect. For more information, see tsm pending-changes.

- cleanup-coordination-service
- deploy-coordination-service
- external-services
  - gateway
    - gateway disable
    - gateway enable
    - gateway update
  - list
  - repository
    - disable
    - enable
    - replace-host
  - storage (filestore)
    - storage disable
    - storage enable
    - storage switch-share
- failover-repository
- filestore
  - decommission
  - recommission
- list-nodes
- list-ports
- node-nickname
  - list
  - remove

tsm topology cleanup-coordination-service

Note: Beginning with version 2020.1.0, all coordination service ensemble commands require input for a "y/n" prompt confirming that a server restart will take place. To run these commands without input, include the `--ignore-prompt` option.

Use the `tsm topology cleanup-coordination-service` command to remove the non-production Tableau Server Coordination Service ensemble after you deploy a new ensemble. This command removes the old Coordination Service instances on all nodes in the non-production Coordination Service ensemble and is required after you deploy a new Coordination Service ensemble. To learn more about Coordination Service ensembles, see Deploy a Coordination Service Ensemble .

In version 2020.1.0 and later, the `tsm topology deploy-coordination-service` command also removes the old ensemble. There is no need to run this command separately unless the deployment fails.

Synopsis

```
tsm topology cleanup-coordination-service [option] [global options]
```

Option

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 2700 (45 minutes).

tsm topology deploy-coordination-service

Note: Beginning with version 2020.1.0, all coordination service ensemble commands require input for a "y/n" prompt confirming that a server restart will take place. To run these commands without input, include the `--ignore-prompt` option.

You can use the `tsm topology deploy-coordination-service` command to deploy the Tableau Server Coordination Service. This command deploys a Coordination Service ensemble, which is a set of Coordination Service instances that run on specified nodes in your server cluster. To learn more about Coordination Service ensembles, including how many nodes in your cluster should have a Coordination Service instance, see Deploy a Coordination Service Ensemble .

In version 2020.1.0 and later, the `tsm topology deploy-coordination-service` command also removes the old ensemble. There is no need to run the `cleanup-coordination-service` command separately.

Synopsis

```
tsm topology deploy-coordination-service --nodes <nodeID,nodeID,...>
[option] [global-options]
```

Options

`-n, --nodes <nodeID,nodeID,...>`

Required.

Node IDs of nodes to include in the new Coordination Service ensemble, separated by commas. You can specify 1, 3, or 5 Coordination Service nodes, depending on the total number of nodes in your cluster. For more information, see The Coordination Service Quorum.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 2700 (45 minutes).

tsm topology external-services gateway disable

Disable all instances of Independent Gateway on Tableau Server.

Synopsis

```
tsm topology external-services gateway disable [options] [global options]
```

Options

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 2700 (45 minutes).

tsm topology external-services gateway enable

Enable instances of Independent Gateway on Tableau Server .

Synopsis

```
tsm topology external-services gateway enable [options] [global options]
```

Options

`-c, --config <configuration-file>`

Required

Specifies the name of the JSON file containing configuration details for all instances of Independent Gateway.

tsm topology external-services gateway update

Use this command to update the configuration of Independent Gateway in Tableau Server. You need to do this if you add or remove additional instances of Independent Gateway, or if you upgrade Independent Gateway. Gather any changes on the Independent Gateway computers and update the configuration file before running this command.

Synopsis

```
tsm topology external-services gateway update [option] [global options]
```

Option

`-c, --config <configuration-file>`

Required

Specifies the name of the JSON file containing configuration details for all instances of Independent Gateway.

tsm topology external-services list

Use the tsm topology external-service-list command to get a the service that is used for Tableau Server External Repository. For example, if you have configured Tableau Server to use Amazon RDS, you will see the following message:

These externally configured services are in use by Tableau Server:

- pgsql

Synopsis

```
tsm topology external-service list [global options]
```

Option

There are no options for this command.

tsm topology external-services repository disable -n nodeN

Use the tsm topology external-services repository disable command to stop using the external repository and reconfigure the installation to use a local repository. This will migrate the data to a local repository and configure Tableau Server to use the local repository.

Synopsis

```
tsm topology external-services repository disable -n nodeN
```

Option

`-n, --node-name <nodeID>`

Required.

Specifies the node ID of the node where the repository should be moved to.

Important: This does not stop or delete the RDS instance. For more information on how to delete an RDS instance, see Deleting a DB Instance on the AWS web site.

tsm topology external-services repository enable

Use the tsm topology external-services repository enable command to configure Tableau Server to use an external repository. This command can be used during installation of a new Tableau Server to configure the external repository. If this command is run on an already existing and running Tableau Server, it will migrate the data from the local node to the external repository and configure Tableau Server to use the external repository after the migration is complete.

Synopsis

```
tsm topology external-services repository enable -f <filename>.json
-c <ssl certificate file>.pem
```

Options

--f <file name>

Required.

Full path and file name where the configuration file is saved. For more information, see Re-Configure Tableau Server Repository.

--c <ssl certificate file>

Required for versions 2021.2 and 2021.2.1. Optional for versions 2021.2.2 and later.

For SSL configurations, download the certificate file and specify the file for use with this option.

1. **Amazon RDS:** See Using SSL to Encrypt the Connection to a DB Instance.
2. **Azure Database:** See Configure TLS connectivity for Azure Database for PostgreSQL.
3. **Stand-alone PostgreSQL Instance:** See Configure SSL .

--no-ssl

Optional. This option is available in version 2021.2.2 and later.

This means SSL is not required when connecting to the External Repository. If you do not need to use encrypted connections, you must also configure the External Repository to allow unencrypted connections. When you use this option, connections will be encrypted if the external repository is configured to support TLS/SSL connections. Otherwise, Tableau Server will use non-encrypted connections.

Skips the check to see if the external repository is already configured for use with Tableau Server. This option is typically not recommended, as it may lead to the same

repository being used by multiple Tableau Server installations which can cause errors. This option may be useful for testing or disaster recovery purposes.

--skip-state-check

Optional. This option is available in version 2022.3.0 and later.

Skips the check to see if the external repository is already configured for use with Tableau Server. This option may be useful for testing or disaster recovery purposes but is not recommended for normal use, as it may result in the same repository being used by multiple Tableau Server installations.

tsm topology external-services repository replace-host

This command updates Tableau Server configuration settings to use the specified external repository. Use the tsm topology external-services repository replace-host command to reconfigure Tableau Server to use the new external repository immediately without moving data to it from your current external repository. You may need to manually migrate the data. You should only do this after you have fully evaluated and understand the impact of the potential data loss.

This command can be used in the following scenarios:

- Planned expiration of the SSL certificates used by RDS instances: RDS instances need to be updated with the new certificates, and Tableau Server needs to be configured to use the new certificate file to connect to the RDS instance.
- Disaster recovery: Use this to connect to a new RDS instance in disaster recovery scenarios. For more information, see Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS)

Synopsis

```
tsm topology external-services repository replace-host -f <file-
name>.json -c <ssl certificate file>.pem
```

Options

-f <file name>

Required.

Full path and file name where the configuration file is saved. For more information, see Re-Configure Tableau Server Repository.

-c <ssl certificate file>

Optional.

The certificate file is the certificate to be imported to allow connections to the instance. For RDS, this is the CA cert used to sign the certificate of the instance. This is usually the latest root certificate **rds-ca-XXXX-root.pem** file. Use this parameter to update Tableu server if the certificate has changed on the RDS instance.

For more information, see Using SSL/TLS to Encrypt a Connection to a DB Instance.

For more information on how to get the .pem file, see Using SSL to Encrypt a Connection to a DB Instance.

--ignore-prompt

Optional.

Run this command without prompts.

tsm topology external-services storage disable

Configure Tableau Server to run File Store locally. Use this command to disable External File Store and move the File Store data to your Tableau Server.

Synopsis

```
tsm topology external-services storage disable [options] [global options]
```

Options

`-fsn <nodeID, nodeID,...>`

Required

Specify the nodes that you want to configure File Store. You can specify more than one node. The data will be migrated to the first node in the list and then replicated to other nodes.

For more information, see Reconfigure File Store .

tsm topology external-services storage enable

Configure Tableau Server with External File Store. External File Store uses SAN or NAS to store File Store data.

Synopsis

```
tsm topology external-services storage enable [options] [global options]
```

Options

`--network-share <network share mount point>`

Required

Specify the mount point of the network share you want to use for your External File Store. For example: `/mnt/<network share name>/tableau`

For more information, see Reconfigure File Store .

tsm topology external-services storage switch-share

Use this command to move your external services to a different network share. An example of this might be when your current network attached storage is at the end of life and you need to use a new network attached storage with new hardware. For more information, see Reconfigure File Store .

Synopsis

```
tsm topology external-services storage switch-share [option] [global options]
```

Option

```
--network-share <network share mount point>
```

Required

Specify the mount point of the network share you want to switch to. For example: `/mnt/<network share name>/tableau`

tsm topology failover-repository

You can use the `tsm topology failover-repository` to manually initiate a repository failover from the current active repository to the second, passive repository.

The `tsm topology failover-repository` command is persistent. The failover repository remains the active repository until you issue the command again, or, if Tableau Server is configured for it, until automatic failover occurs. If you have a preferred active repository configured, use the `--preferred` option to switch back to that repository. For more information about configuring a preferred active repository, see Tableau Server Repository. If Tableau Server is configured for high availability, failover of the repository is automatic when necessary. Use the failover-repository command to manually fail over the repository.

Synopsis

```
tsm topology failover-repository --preferred | --target <node_id> [global options]
```

Options

```
-r, --preferred
```

Required if -t or --target is not used.

Use the configured preferred node as the target for repository failover.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

`-t, --target <node_id>`

Required if -r or --preferred is not used.

The node id of the target node onto which failover will occur. Find the node id by using the `tsm topology list-nodes` command.

tsm topology filestore decommission

You must use the `tsm topology filestore decommission` command to prepare a file store node or nodes for safe removal. This command puts the specified nodes into read-only mode and ensures there is no unique content on the specified nodes.

If decommissioning results in a single file store node, you must use the `--override` option or the decommission will fail.

Synopsis

```
tsm topology filestore decommission --nodes <nodeID,nodeID,...>
[options] [global options]
```

Options

`-n, --nodes <nodeID,nodeID,...>`

Required.

List of one or more nodes to decommission, specified by node ID and separated by commas.

`--delete-filestore`

Optional.

Forces the removal of the file store, even if it has not been decommissioned. You should only use this option if the node the file store is on is in a error state and decommissioning cannot be done. Any unique files on the node will be permanently deleted.

`-o, --override`

Optional.

Overrides warnings or failures that would normally occur if removing the target File Store node would reduce the number of remaining file store nodes to one. This option cannot be used with the `--delete-filestore` option.

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

tsm topology filestore recommission

Use the `tsm topology filestore recommission` command to revert any decommissioned nodes back to read-write mode.

Synopsis

```
tsm topology filestore recommission --nodes <nodeID,nodeID,...>
[global options]
```

Options

`-n, --nodes <nodeID,nodeID,...>`

Required.

List of one or more nodes to recommission, specified by node ID and separated by commas.

tsm topology list-nodes

Display the nodes in the cluster and (optionally) the services on each node.

Synopsis

```
tsm topology list-nodes [options] [global options]
```

Options

-v, --verbose

Optional.

Shows each node ID, the node role (for more information, see `set-node-role` below), the node address, and the processes on each node.

tsm topology list-ports

Display the ports in the cluster.

Synopsis

```
tsm topology list-ports [options] [global options]
```

Options

`--node-name <nodeID>`

Optional.

Specify the node to list ports for.

`--service-name`

Optional.

Specify the service to list ports for.

tsm topology node-nickname list

Display the node nicknames for nodes in the cluster.

Synopsis

```
tsm topology node-nickname list [options] [global options]
```

Options

`--nodes <nodeID,nodeID,...>`

Optional.

Specify the node IDs of the nodes to list with nicknames.

tsm topology node-nickname remove

Remove the nickname from the specified node or nodes.

Synopsis

```
tsm topology node-nickname remove [options] [global options]
```

Options

`--all`

Required if --nodes is not specified.

Remove the nicknames from all nodes in the cluster.

`--nodes <nodeID,nodeID,...>`

Required if --all is not specified.

Specify the node ID of the node or nodes whose nicknames should be removed.

tsm topology node-nickname set

Set the nickname for the specified node.

Synopsis

```
tsm topology node-nickname set [options] [global options]
```

Options

`-id, --node <nodeID>`

Required.

Specify the node to set the nickname for.

`-nn, --nickname <name>`

Required.

The nickname for the specified node.

tsm topology nodes get-bootstrap-file

You can use the `tsm topology nodes get-bootstrap-file` command to get the bootstrap file that is required to add a new node to the cluster.

Important: The bootstrap file contains a copy of the master keystore file used for encrypting the configuration secrets. The file can also embedded credentials which are valid for a predetermined amount of time (see tabadmincontroller.auth.expiration.minutes) and serve as a session cookie. We strongly recommend that you take additional measures to secure the

bootstrap file.

The following command set provides an example method to encrypt the bootstrap file output. This method is similar to the encryption process described in more detail at Securing secrets for import and export operations.

Note however, the method here must be passed as separate arguments with trailing `&& \` operators as follows:

```
mkfifo -m 600 /tmp/secure1 && \

tsm topology nodes get-bootstrap-file --file /tmp/secure1 && \

gpg --symmetric --batch --yes --passphrase-file ~/.secret-
s/pgppassphrase.txt --cipher-algo AES256 --output encrypted.enc <
/tmp/secure1 && \

rm /tmp/secure1
```

Synopsis

```
tsm topology nodes get-bootstrap-file --file <path\file>.json
[global options]
```

Options

`-f,--file <file>`

Required.

Full path and file name where the configuration file will be saved. If a duplicate file exists it will be overwritten.

`-nec,--no-embedded-credential`

Optional.

Added in version 2019.3.

By default embedded credentials are included in the bootstrap file. Use this option if credentials should not be included in the bootstrap file. Embedded credentials are temporary, and expire based on the value of the `tabadmincontroller.auth.expiration.minutes`configuration key, by default 120 minutes.

> **Note:** You can disable the ability to include embedded credentials at the server level, using a configuration option. For more information, see features.PasswordlessBootstrapInit.

tsm topology remove-nodes

Remove nodes from the cluster.

To complete removal of a node, you also must run the `tsm pending-changes apply` command. Some scenarios require that you move or redeploy processes before removing nodes. See Remove a Node.

If you remove a node and want to re-add it to the cluster, you need to first run the obliterate script to clean Tableau off it, then reinstall the node using the normal process for adding a new node. For more information, see Remove Tableau Server from Your Computer and Install and Configure Additional Nodes.

> **Note**: To remove a node from a cluster it must have been configured with a process at some point in the past. If you are removing a node on which you've not configured any processes, then you must add a process on it, run `tsm pending-changes apply`, and then remove the node.

Synopsis

```
tsm topology remove-nodes --nodes <nodeID,nodeID,...>
[global options]
```

Options

`-n, --nodes <nodeID,nodeID,...>`

Required.

Specify the node or nodes to remove. If specifying multiple nodes, separate node IDs with a comma.

tsm topology set-node-role

Set the Backgrounder and Extract Queries node roles. This determines the type of tasks that will be performed on the nodes. The following node roles can be useful if you have a multi-node cluster. Different node roles may require licenses for Advanced Management or Data Management, or for both. For more information about license requirements, see Workload Management through Node Roles.

Note: Making configurations to node roles require a restart of the server and will require some downtime. For more information, see tsm pending-changes.

Synopsis

```
tsm topology set-node-role [options] [global options]
```

Options

`-n, --nodes <nodeID,nodeID,...>`

Required.

List of one or more nodes to set node roles for, specified by node ID and separated by commas and without spaces between nodes.

```
-r --role <all-jobs,flows,no-flows,extract-refreshes,sub-
scriptions,extract-refreshes-and-subscriptions,no-extract-
refreshes,no-subscriptions,no-extract-refreshes-and-sub-
scriptions,extract-queries,extract-queries-interactive>
```

Required

Sets the role for the nodes specified. The valid values for this option are:

- all-jobs: Backgrounder will run all types of jobs.
- flows :Backgrounder will run only flow run jobs.
- no-flows: Backgrounder will not run flow run jobs.
- extract-refreshes: Backgrounder will run only extract refresh jobs. This includes, incremental refreshes, full refreshes, encryption and decryption of all extracts including extracts that flow outputs generate.
- subscriptions: Backgrounder will run only subscription jobs.
- extract-refreshes-and-subscriptions: Backgrounder will run extract-refreshes, encryption and decryption of all extracts including extracts that flow outputs create, and subscription jobs.
- no-extract-refreshes: Backgrounder will run all jobs except extract-refreshes, extract encryption and decryption including extracts created from flow outputs.
- no-subscriptions: Backgrounder will run all jobs except subscriptions.
- no-extract-refreshes-and-subscriptions: Backgrounder will run all jobs except extract-refreshes, encryption and decryption of all extracts including extracts created from flow outputs, and subscriptions.
- extract-queries: The nodes selected will run as all-jobs and will prioritize the processing of extract queries.
- extract-queries-interactive: The nodes selected will run as all-jobs and will prioritize the processing of interactive extract queries, such as those that run when

a user is looking at their screen and waiting for an extract-based dashboard to load. This is an advanced setting and it should only be used if the cluster has a heavy subscription and alert job workload that causes users to experience degraded performance on viz load times that run around the same time as scheduled loads.

- system: Backgrounder will run only system maintenance jobs that interact with other Tableau Server processes, such as cleaning crashed jobs, reaping database events, and synching Active Directory.
- no-system: Backgrounder will run all jobs except system maintenance jobs.

tsm topology set-ports

Set the ports for a service instance.

Synopsis

```
tsm topology set-ports --node-name <nodeID> --port-name <port_name>
--port-value <port_value> [options] [global options]
```

Options

`-i, --instance <instance_id>`

Optional.

Specifies the instance id of the service. Defaults to 0 (zero) if not specified.

`-n, --node-name <nodeID>`

Required.

Specifies the node ID of the node.

`-pn, --port-name <port_name>`

Required.

The name of the port to be set, in this format: `service_name:port_type`. If no port type is specified, the primary port is assumed. For port name syntax, see Dynamically mapped ports.

`-pv, --port-value <port_value>`

Required.

The port to set.

`-r, --restart`

Optional.

Suppress the prompt for restart and restart Tableau Server when necessary.

tsm topology set-process

Set the number of instances of a process on a node. If a node already has the specified process, the number is updated to match the specified count.

- You can only set one process at a time. If you specify more than one process, any process after the first one will be silently ignored.
- You must set a process one node at a time. If you specify more than one node, the command will display an "invalid node name" error.

When you update the number of processes on nodes, you also need to apply pending changes. In most cases this also requires a server restart (you will be prompted), but there are special cases where you can make dynamic topology changes without needing to restart the server. For more information, see Tableau Server Dynamic Topology Changes.

Note: For a complete list of process names, see Tableau Server Processes.

Synopsis

```
tsm topology set-process --count <process_count> --node <nodeID> --
process <process_name> [global options]
```

Options

`-c, --count <process_count>`

Required.

The process count (number of instances) to set.

`-n, --node <nodeID>`

Required.

Specifies the node ID of the node on which to set the process.

`-pr, --process <process_name>`

Required.

The name of the process to be set.

tsm topology toggle-coordination-service

Note: Beginning with version 2020.1.0, all coordination service ensemble commands require input for a "y/n" prompt confirming that a server restart will take place. To run these commands without input, include the `--ignore-prompt` option.

You can use the `tsm topology toggle-coordination-service` command to switch between coordination service ensembles. To learn more about Coordination Service ensembles, see Deploy a Coordination Service Ensemble .

In version 2020.1.0 and later, the `tsm topology deploy-coordination-service` command also switches to the new ensemble. There is no need to run this command separately.

Synopsis

```
tsm topology toggle-coordination-service [option] [global options]
```

Option

`--request-timeout <timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm user-identity-store

You can use the `tsm user-identity-store` commands to modify the settings of the identity store for Tableau Server after the initial configuration.

The initial configuration of the identity store is part of the installation process. See Configure Initial Node Settings.

For introduction to identity store concepts, see Identity Store.

For LDAP/Active Directory configuration reference table, see External Identity Store Configuration Reference.

- get-group-mappings
- get-user-mappings
- list
- set-connection
- set-group-mappings
- set-user-mappings
- verify-group-mappings
- verify-user-mappings

tsm user-identity-store get-group-mappings [options]

Displays identity store group mappings.

Synopsis

```
tsm user-identity-store get-group-mappings [global options]
```

tsm user-identity-store get-user-mappings [options]

Displays identity store user mappings.

Synopsis

```
tsm user-identity-store get-user-mappings [global options]
```

tsm user-identity-store list [options]

Lists user-identity configuration.

Synopsis

```
tsm user-identity-store list [options] [global options]
```

Options

`-v, --verbose`

Optional.

Lists all configuration parameters.

tsm user-identity-store set-connection [options]

Sets identity store connection parameters.

Synopsis

```
tsm user-identity-store set-connection --kerbkeytab <kerbkeytab>
[options] [global options]
```

Options

`-b,--bind <username and password | Kerberos>`

Optional.

Set LDAP bind type.

`-d,--domain <domain>`

Optional.

Domain name.

`-hn,--hostname <hostname>`

Optional.

The hostname of the LDAP server. You can enter a hostname or an IP address for this value. The host that you specify here will be used for user/group queries on the primary domain. In the case where user/group queries are in other domains, Tableau Server will query DNS to identify the appropriate domain controller.

`-kc,--kerbconfig <kerbconfig>`

Optional.

Kerberos configuration file path.

`-kp,--kerbprincipal <kerbprincipal>`

Optional.

Kerberos Principal.

`-kt,--kerbkeytab <kerbkeytab>`

Required.

Kerberos keytab file path.

`-l,--port <port>`

Optional.

Set LDAP Port value.

`-lp,--ldappassword <ldappassword>`

Optional.

LDAP Password.

`-lu,--ldapusername <ldapusername>`

Optional.

Set LDAP Username value.

`-n,--nickname <nickname>`

Optional.

NetBIOS name (nickname).

tsm user-identity-store set-group-mappings [options]

Sets identity store group mappings and configures LDAP directories that implement an arbitrary or custom schema.

Synopsis

```
tsm user-identity-store set-group-mappings [options]
[global options]
```

Options

`-b,--basefilter <groupbasefilter>`

Optional.

Set group BaseFilter value.

`-cn,--classnames <group_classnames>`

Optional.

Override default user classname values (contains "group" string) with the values you set here. You can provide multiple classnames separated by commas.

`-d,--description <description>`

Optional.

Group description.

`-e,--groupemail <groupemail>`

Optional.

Group email value.

`-m,--member <member>`

Optional.

Set the group members.

`-n,--groupname <groupname>`

Optional.

Name of the group.

tsm user-identity-store set-user-mappings [options]

Sets identity store user mappings and configures LDAP directories that implement an arbitrary or custom schema.

Synopsis

```
tsm user-identity-store set-user-mappings --certificate <certificate> [options] [global options]
```

Options

`-c,--certificate <certificate>`

Optional.

Users' certificate file location.

`-cn,--classnames <user_classnames>`

Optional.

Override default user classname values ("user" and "inetOrgPerson") with the values you set here. You can provide multiple classnames separated by commas.

`-dn,--displayname <displayname>`

Optional.

Display name of the user.

`-e,--email <email>`

Optional.

Users' email address.

`-jp,--jpegphoto <jpegfile>`

Optional.

Users' jpeg image location.

`-m,--memberof <groupname>`

Optional.

Group that the user is a member of.

`-t,--thumbnail <thumbnail>`

Optional.

Users' thumbnail location.

`-ub,--basefilter <userbasefilter>`

Optional.

Users' BaseFilter.

`-uu,--ldapusername <ldapusername>`

Optional.

User name.

tsm user-identity-store verify-group-mappings [options]

Validates configuration for LDAP group mapping.

Synopsis

```
tsm user-identity-store verify-group-mappings --verify <group_name>
[global options]
```

Options

`-v,--verify <group_name>`

Optional.

Name of group to search for.

tsm user-identity-store verify-user-mappings [options]

Validates configuration for LDAP user mapping.

Synopsis

```
tsm user-identity-store verify-user-mappings --verify <user_name>
[global options]
```

Options

`-v,--verify <user_name>`

Optional.

Name of user to search for.

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm version

You can use the `tsm version` command to get the versions of TSM and Tableau Server.

Synopsis

```
tsm version [global options]
```

Global options

`-h, --help`

Optional.

Show the command help.

`-p, --password <password>`

Required, along with `-u` or `--username` if no session is active.

Specify the password for the user specified in `-u` or `--username`.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

`-s, --server https://<hostname>:8850`

Optional.

Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address. For example `https://<tsm_hostname>:8850`. If no server is specified, `https://<localhost | dnsname>:8850` is assumed.

`--trust-admin-controller-cert`

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

`-u, --username <user>`

Required if no session is active, along with `-p` or `--password`.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm File Paths

Certain tsm commands read files from or write files to default locations. These default locations are determined by `basefilepath` configuration keys defined for each command. You can use tsm to view the current value of the keys, and to change the locations.

Default locations for files

During the `tsm maintenance backup`, `restore`, `send-logs`, and `ziplogs` processes, and the `tsm sites export` and `sites import` processes, Tableau Server uses default locations for the files created or used by these commands.

For details on disk space requirements for backing up Tableau Server, see Disk Space Usage for Backup.

By default:

- tsm maintenance commands:
    - **backup**—The backup `.tsbak` file is created in a temporary location in the data directory on the initial node and then saved in:

      `/var/opt/tableau/tableau_server/data/tabsvc/files/backups/`
    - **restore**—The restore process restores a backup file from:

      `/var/opt/tableau/tableau_server/data/tabsvc/files/backups/`
    - **send-logs**—The send-logs sends the logs file from:

      `/var/opt/tableau/tableau_server/data/tabsvc/files/backups/`
    - **ziplogs**—The ziplogs file is generated in:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/log-
archives
```

- tsm sites
    - **export**—The export .zip file is generated to the following directory:

      ```
      /var/opt/tableau/tableau_server-
      /data/tabsvc/files/siteexports
      ```

    - **import**—During the import process, Tableau Server looks for files in:

      ```
      /var/opt/tableau/tableau_server-
      /data/tabsvc/files/siteimports
      ```

Get the current file location

You can see the current file location for a specific command using `tsm configuration get`:

- For tsm maintenance commands:
    - backup, restore, and send-logs:

      ```
      tsm configuration get -k basefilepath.backuprestore
      ```

    - ziplogs:

      ```
      tsm configuration get -k basefilepath.log_archive
      ```

- For tsm sites commands:
    - export

      ```
      tsm configuration get -k basefilepath.site_export.exports
      ```

- import

```
tsm configuration get -k basefilepath.site_import.exports
```

Change the current file location

You can change the expected file locations using the `tsm configuration set` command to update the `basefilepath` variables. For details about specific base file paths, see tsm configuration set Options.

Changing a `basefilepath` variable does not move existing files from the original directory to the new directory. If you want existing backup, restore, log files, or site export or import files to reside in the new directory you specify, you must move them manually. You are responsible for creating the new location and for setting the correct permissions to allow tsm access to any files that will be placed there, and to the directory structure containing those files. For more information about permissions and tsm, see Files and Permissions in TSM. If you change the backup/restore base file path, you should run the tsm maintenance validate-backup-base-filepath command (available in version 2022.1 and later) to verify the permissions are properly set.

The `tsm maintenance backup` command assembles the backup in a temporary location in the data directory before saving the backup file to the location specified by the `base-filepath.backuprestore` variable. Changing the basefilepath does not impact where the `tsm maintenance backup` command assembles the backup file.

- For tsm maintenance commands:
  - To change the backup, restore, or send-logs directory, run the following command:

    ```
    tsm configuration set -k basefilepath.backuprestore -v
    "/new/directory/path"
    ```

  - To change the ziplogs directory:

```
tsm configuration set -k basefilepath.log_archive -v
"/new/directory/path"
```

- For tsm sites commands:
  - To change the sites export directory:

    ```
    tsm configuration set -k basefilepath.site_export.exports
    -v "/new/directory/path"
    ```

  - To change the sites import directory:

    ```
    tsm configuration set -k basefilepath.site_import.exports
    -v "/new/directory/path"
    ```

After you change a default file location you need to do the following:

1. Apply pending changes:

   ```
   tsm pending-changes apply
   ```

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

2. Stop Tableau Server:

   ```
   tsm stop
   ```

3. Restart the TSM Controller (as *tableau* system account):

   ```
   sudo su -l tableau -c "systemctl --user restart tabad-
   mincontroller_0.service"
   ```

4. Wait several minutes for the controller to restart. You can confirm the controller has restarted with this command:

   ```
   tsm status -v
   ```

   When you can run that command and the Tableau Server Administration Controller is listed as 'running' the controller has restarted.

5. Start Tableau Server:

   ```
   tsm start
   ```

Entity Definitions and Templates

You can use the entity definitions and templates with the `tsm settings import` command to configure various Tableau Server settings.

Configuration File Example

This article provides an example of a complete JSON configuration file, with gatewaySettings and identityStore entities specified. In addition, a configuration key sets the gateway timeout to 900 seconds.

Your configuration file will look different depending on the options you need to set.

You might set multiple .json configuration files during installation. To set the values for each file in Tableau Server, you run the following command, once for each configuration file:

```
tsm settings import -f path-to-file.json
```

After you set the configuration files, run tsm pending-changes apply to apply the changes from all of the .json files you've set.

```
{
   "configEntities": {
      "gatewaySettings": {
         "_type": "gatewaySettingsType",
         "port": 80,
         "publicHost": "localhost",
         "publicPort": 80
      },
      "identityStore": {
         "_type": "identityStoreType",
         "type": "local",
         "domain": "example.lan",
         "nickname": "EXAMPLE"
      }
    },
     "configKeys": {
        "gateway.timeout": "900"
     }
}
```

Entities vs keys

As shown in the example above, there are two classes of configuration parameters: configEntities and configKeys.

configEntities

Certain types of configuration are done through entity sets that map to specific scenarios, such as the identity store and gateway configurations. When you pass a set of configEntities with the `tsm settings import -f path-to-file.json` command, TSM validates the configuration. If values passed are invalid, TSM will provide an error. This enables you to make changes during the configuration process, rather than experience a configuration failure at initialization or run time.

Entities can be set only by including a configEntities block in a .json file.

> **Important**: All files that are referenced in configEntities must be located on the local computer. Do not specify UNC paths.

configKeys

Entities cover only a small portion of the configuration values that can be set. Hundreds of keys correspond to parameters stored in .yml files. Tableau Server uses these parameters to store all of the configuration information for all services.

You can set individual keys with the tsm configuration command. But during deployment, setting them along with other configuration scenarios in JSON files, as shown above, is more convenient.

Unlike configEntities, configKeys are not validated.

> **Note:** We do not recommend setting parameters that are not documented in tsm configuration set Options.

gatewaySettings Entity

You must configure the gateway settings for the Tableau Server computer.

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f /path/to/file.json

tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--`

`ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Gateway settings

The gateway settings in the template below specify the HTTP settings for Tableau Server. We recommend using SSL/TLS. Tableau Server is hard-coded to use port 443 for SSL/TLS. Therefore, if you enable SSL, you do not need to update the gatewaySettings entity.

Configuration template

Use this template to configure the gateway settings.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

```
{
  "configEntities": {
        "gatewaySettings": {
         "_type": "gatewaySettingsType",
          "port": 80,
          "sslRedirectEnabled": true,
          "publicHost": "localhost"
        }
       }
 }
```

Configuration file reference

This table includes all of the options that can be included with the `"gatewaySettings"` entity set.

`_type`

Required.

Value: `"gatewaySettingsType"`

Do not change.

`port`

Specifies HTTP port. Default is port 80.

`sslRedirectEnabled`

Options: `true` or `false`.

`publicHost`

Specifies host name for http/s service.

`trustedIPs`

Specifies trusted IP addresses that communicate with Tableau Server. Trusted IP addresses include upstream proxy servers and servers that are used for trusted authentication with Tableau Server. See Configuring Proxies and Load Balancers for Tableau Server and Add Trusted IP Addresses or Host Names to Tableau Server.

If you are running Tableau Server in a cluster then all other nodes of the cluster will automatically be included in the corresponding configuration file that this entity updates. Therefore, if you specify a new value for `trustedIPs`, then you must include the IP addresses for the other nodes in the value.

This option takes a list of strings, which requires passing each IP or host in quotes, separated by a comma (no space) and within brackets. For example: `["192.168.1.101","192.168.1.102","192.168.1.103"]` or `["webserv1","webserv2","webserv3"]`.

`trustedHosts`

Specifies trusted IP addresses that communicate with Tableau Server. Typically, this value contains a list of upstream proxy servers. The values in trustedHosts are used to

determine client request targets.

If you are running Tableau Server in a cluster then all other nodes of the cluster will automatically be included in the corresponding configuration file that this entity updates. Therefore, if you specify a new value for `trustedIPs`, then you must include the IP addresses for the other nodes in the value.

This option takes a list of strings, which requires passing each IP or host in quotes, separated by a comma (no space) and within brackets. For example: `["192.168.1.101","192.168.1.102","192.168.1.103"]` or `["webserv1","webserv2","webserv3"]`.

identityStore Entity

Tableau Server requires an identity store to store user and group information. Review Authentication and Identity Store topics before configuring the identity store for the first time. After you have installed the identity store on Tableau Server, you cannot change it without reinstalling the server.

Important: All entity options are case sensitive.

Before you begin

Review the following information:

- If you will not be using the local identity store, then you will be using some version of LDAP. In this case, work with your directory/LDAP administrator to configure Tableau Server for your LDAP schema and bind requirements.
- Tableau Server configuration is optimized for Active Directory. If you are installing into Active Directory, we recommend configuring the identity store with Configure Initial Node Settings.
- LDAP bind is independent of user authentication. For example, you can configure Tableau Server to use simple bind to authenticate to the LDAP directory and then configure Tableau Server to authenticate users with Kerberos after installation.

- Do not connect to LDAP with simple bind over an unsecured connection. By default, LDAP with simple bind sends data in cleartext. Use LDAPS to encrypt traffic with simple bind. See Configure Encrypted Channel to LDAP External Identity Store.
- To use Kerberos authentication for the LDAP bind with Tableau Server service, then you'll need a keytab file for GSSAPI bind, as described in the sections below. See also, Understanding Keytab Requirements. In the context of Kerberos, GSSAPI bind is all you need during the base installation of Tableau Server. After you install the server, you can then configure Kerberos for user authentication and Kerberos delegation to data sources.
- In this topic, we make the distinction between *LDAP* (the protocol for connecting to directory services) and an *LDAP server* (an implementation of a directory service). For example, `slapd` is an LDAP server that is part of the OpenLDAP project.
- Validate the LDAP configuration before initializing the server, see Configure Initial Node Settings.
- Import JSON configuration files only as part of the initial configuration. If you need to make LDAP changes after you have imported the JSON configuration file and initialized Tableau Server, do not attempt to re-import the JSON file. Rather, make individual key changes with native tsm commands or with `tsm configuration set`. See External Identity Store Configuration Reference.

Configuration templates

The JSON templates in this section are used to configure Tableau Server with different identity store scenarios. Unless you're configuring a local identity store, you will need to select and edit a configuration file template that is specific to your LDAP environment.

Consider using the Tableau Identity Store Configuration Tool to help generate your LDAP JSON configuration file. The tool itself is not supported by Tableau. However, using a JSON file created by the tool instead of creating a file manually does not change the supported status of your server.

Select an identity store configuration template to edit:

- Local
- LDAP - Active Directory
- OpenLDAP - GSSAPI Bind
- OpenLDAP - Simple Bind

For more explanation about configuration files, entities, and keys see Configuration File Example.

Local

Configure local as the identity store type if your organization does not already have an Active Directory or LDAP server for user authentication. When you select local as the identity store type, you use Tableau Server to create and manage users.

An alternative way to configure Tableau Server for local identity store is to run Setup GUI and select "Local" during the installation process. See Configure Initial Node Settings.

```
{
  "configEntities": {
    "identityStore": {
       "_type": "identityStoreType",
       "type": "local"
     }
   }
}
```

Important

The LDAP configuration templates below are examples. The templates, as presented, will not configure LDAP connectivity in your organization. You must work with your directory administrator to edit the LDAP template values for a successful deployment.

Additionally, all files that are referenced in configEntities must be located on the local computer. Do not specify UNC paths.

LDAP - Active Directory

Tableau Server configuration is optimized for Active Directory. If you are installing into Active Directory, configure the identity store with Configure Initial Node Settings.

An encrypted connection to Active Directory is required.See Configure Encrypted Channel to LDAP External Identity Store.

If for some reason you are unable to configure the identity store to communicate with Active Directory with TSM web interface, use this JSON template to configure Tableau Server to connect to Active Directory. This template uses GSSAPI (Kerberos) bind to authenticate Tableau Server service to Active Directory. Tableau Server includes support for Active Directory schema. Therefore, if you set the `"directoryServiceType"` option to `"activedirectory"` then you do not need to provide schema info in the `"identityStoreSchemaType"` option.

If you are installing Tableau Server for Linux into Active Directory, and the computer where you are installing Tableau Server is already joined to the domain, then the computer will already have a Kerberos configuration file and a keytab file. Strictly speaking, you can use these files for GSSAPI bind, but we don't recommend using them. Instead, contact your Active Directory administrator and request a keytab specifically for the Tableau Server service.

```
{
  "configEntities":{
    "identityStore": {
                "_type": "identityStoreType",
                "type": "activedirectory",
                "domain": "your-domain.lan",
                "nickname": "YOUR-DOMAIN-NICKNAME",
                "directoryServiceType": "activedirectory",
                "bind": "gssapi",
                "kerberosKeytab": "<path to local key tab file>",
                "kerberosConfig": "/etc/krb5.conf",
                "kerberosPrincipal": "your-principal@YOUR.DOMAIN"
                }
```

```
        }
}
```

We recommend binding to Active Directory with GSSAPI. However, you can connect with simple bind and LDAPS. To connect with simple bind, change `bind` to `simple`, remove the three Kerberos entities, and add the `port`/`sslPort`, `username`, and `password` options. The following example shows Active Directory with simple bind json.

```
{
  "configEntities":{
       "identityStore": {
               "_type": "identityStoreType",
               "type": "activedirectory",
               "domain": "your-domain.lan",
               "nickname": "YOUR-DOMAIN-NICKNAME",
               "directoryServiceType": "activedirectory",
               "hostname": "optional-ldap-server",
               "sslPort": "636",
               "bind": "simple",
               "username": "username",
               "password": "password"
               }
       }
}
```

OpenLDAP - GSSAPI bind

Use the template below to configure OpenLDAP with GSSAPI bind. Do not use this template if your organization is running Active Directory. If you are installing into Active Directory, use the template above, LDAP - Active Directory.

In most cases, organizations that use OpenLDAP with GSSAPI (Kerberos) will use a keytab file to store credentials. In the following example, a keytab file is used for authentication credentials.

However, you can provide credentials through the `username` and `password` entities.

You can also specify both a keytab and a username and password pair. In this case, Tableau Server will attempt to use the keytab, but if authentication fails for any reason it will fallback and use the username and password credentials.

```
{
  "configEntities":{
               "identityStore": {
                       "_type": "identityStoreType",
                       "type": "activedirectory",
                       "domain": "your-domain.lan",
                       "nickname": "YOUR-DOMAIN-NICKNAME",
                       "directoryServiceType": "openldap",
                       "bind": "gssapi",
                       "kerberosKeytab": "<path to local key tab file>",
                       "kerberosConfig": "/etc/krb5.conf",
                       "kerberosPrincipal": "your-principal@YOUR.DOMAIN",
                       "identityStoreSchemaType": {
                          "userBaseFilter": "(objectClass=inetOrgPerson)",
                          "userUsername": "user",
                          "userDisplayName": "displayname",
                          "userEmail": "email",
                          "userCertificate": "certificate",
                          "userThumbnail": "thumbnail",
                          "userJpegPhoto": "photo",
                          "groupBaseFilter": "(objectClass=groupofNames)",
                          "groupName": "groupname",
                          "groupEmail": "groupemail",
                          "groupDescription": "groupdescription",
                          "member": "member",
                          "distinguishedNameAttribute": "",
                          "serverSideSorting": "",
                          "rangeRetrieval": "",
                          "userClassNames": ["inetOrgPerson","someClass2"],
                          "groupClassNames": ["groupOfU-
niqueNames1","groupOfUniqueNames2"]
                          }
```

```
                    }
            }
}
```

OpenLDAP - Simple bind

```
{
  "configEntities":{
               "identityStore": {
                       "_type": "identityStoreType",
                       "type": "activedirectory",
                       "domain": "my.root",
                       "nickname": "",
                       "hostname": "optional-ldap-server",
                       "port": "389",
                       "directoryServiceType": "openldap",
                       "bind": "simple",
                       "username": "cn=username,dc=your,dc=domain",
                       "password": "password",
                       "identityStoreSchemaType": {
                          "userBaseFilter": "(objectClass=inetOrgPerson)",
                          "userUsername": "user",
                          "userDisplayName": "displayname",
                          "userEmail": "email",
                          "userCertificate": "certificate",
                          "userThumbnail": "thumbnail",
                          "userJpegPhoto": "photo",
                          "groupBaseFilter": "(objectClass=groupofNames)",
                          "groupName": "groupname",
                          "groupEmail": "groupemail",
                          "groupDescription": "groupdescription",
                          "member": "member",
                          "distinguishedNameAttribute": "",
                          "serverSideSorting": "",
                          "rangeRetrieval": "",
                          "userClassNames": ["inetOrgPerson","someClass2"],
```

```
                              "groupClassNames": ["groupOfU-
niqueNames1","groupOfUniqueNames2"]
                              }
                  }
  }
}
```

Configuration template reference

Shared identity store options

`type`
: Where you want to store user identity information. Either `local` or `activedirectory`. (If you want to connect to any LDAP server, select `activedirectory`.)

`domain`
: The domain of the computer where you installed Tableau Server.

`nickname`
: The nickname of the domain. This is also referred to as the NetBIOS name in Windows environments.
The `nickname` option is required for all LDAP entities. If your organization does not require a nickname/NetBIOS, then pass a blank key, for example: `"nickname": ""`.

LDAP GSSAPI bind options

`directoryservicetype`
: The type of directory service that you want to connect to. Either `activedirectory` or `openldap`.

`kerberosConfig`
: The path to the Kerberos configuration file on the local computer. If you are installing into Active Directory, we don't recommend using the existing Kerberos configuration file or keytab file that may already be on the domain-joined computer. See Identity Store.

`kerberosKeytab`

The path to the Kerberos keytab file on the local computer. It is recommended that you create a keytab file with keys specifically for Tableau Server service and that you do not share the keytab file with other applications on the computer. For example, on Linux, you might place the keytab file in the `/var/opt/tableau/keytab` directory.

`kerberosPrincipal`

The service principal name for Tableau Server on the host machine. The keytab must have permission for this principal. Do not use the existing system keytab at `/etc/krb5.keytab`. Rather, we recommend that you register a new service principal name. To see principals in a given keytab, run the `klist -k` command. See Understanding Keytab Requirements.

LDAP simple bind options

`directoryservicetype`

The type of directory service that you want to connect to. Either `activedirectory` or `openldap`.

`hostname`

The hostname of the LDAP server. You can enter a hostname or an IP address for this value. The host that you specify here will be used for user/group queries on the primary domain only. If user/group queries are in other domains (not in the primary domain), Tableau Server will not use this value, but instead will query DNS to identify the appropriate domain controller.

`port`

Use this option to specify the non-secure port of the LDAP server. Plaintext is usually 389.

`sslPort`

Use this option to enable LDAPS. Specify the secure port of the LDAP server. LDAPS is usually port 636. To use LDAPS you must also specify hostname option. See Configure Encrypted Channel to LDAP External Identity Store.

`username`

The user name that you want to use to connect to the directory service. The account that you specify must have permission to query the directory service. For Active Directory, enter the username, for example, `jsmith`. For LDAP servers, enter the distinguished name (DN) of the user that you want to use to connect. For example, you might enter `cn=username,dc=your-local-domain,dc=lan`.

`password`

The password of the user that you want to use to connect to the LDAP server.

LDAPS and subdomains

If you're enabling LDAPS in Active Directory and connecting to subdomains, you'll need to run the following TSM command to configure the LDAPS port (TCP 636) for subdomains. The command takes arguments that specify `subdomainFQDN:port`.

Example: `tsm configuration set -k wgserver.domain.ldap.domain_custom_ports -v subdomain1.lan:636,subdomain2.lan:636,subdomain3.lan:636`

For more information, see tsm configuration set Options.

Shared LDAP options

The following options can be set for generic LDAP, OpenLDAP, or Active Directory implementations.

`bind`

The way that you want to authentication communication from the Tableau Server service to the LDAP directory service. Enter `gssapi` for GSSAPI (Kerberos).

`domain`

In Active Directory environments, specify the domain where Tableau Server is installed, for example, "example.lan".
For non-AD LDAP: the string you enter for this value is displayed in the "Domain" column of user management tools. You can enter an arbitrary string, but the key cannot be blank.

root

LDAP only. Do not specify for Active Directory.
If you do not use a dc component in the LDAP root or you want to specify a more complex root you need to set the LDAP root. Use the "o=my,u=root" format. For example, for the domain, `example.lan`, the root would be `"o=example,u=lan"`.

`membersRetrievalPageSize`

This option determines the maximum number of results returned by an LDAP query. For example, consider a scenario where Tableau Server is importing an LDAP group that contains 50,000 users. Attempting to import such a large number of users in a single operation is not a best practice. When this option is set to 1500, Tableau Server imports the first 1500 users in the first response. After those users are processed, Tableau Server requests the next 1500 users from the LDAP server, and so forth. We recommend that you modify this option only to accommodate the requirements of your LDAP server.

`identityStoreSchemaType` options

If you configure an LDAP connection to an LDAP server, you can enter schema information specific to your LDAP server in the `identityStoreSchemaType` object.

Important If you are connecting to Active Directory (`"directoryServiceType": "activedirectory"`), then do not configure these options.

`userBaseFilter`

The filter that you want to use for users of Tableau Server. For example, you might specify an object class attribute and an organization unit attribute.

`userUsername`

The attribute that corresponds to user names on your LDAP server.

`userDisplayName`

The attribute that corresponds to user display names on your LDAP server.

`userEmail`
: The attribute that corresponds to user email addresses on your LDAP server.

`userCertificate`
: The attribute that corresponds to user certificates on your LDAP server.

`userThumbnail`
: The attribute that corresponds to user thumbnail images on your LDAP server.

`userJpegPhoto`
: The attribute that corresponds to user profile images on your LDAP server.

`groupBaseFilter`
: The filter that you want to use for groups of users of Tableau Server. For example, you might specify an object class attribute and an organization unit attribute.

`groupName`
: The attribute that corresponds to group names on your LDAP server.

`groupEmail`
: The attribute that corresponds to group email addresses on your LDAP server.

`groupDescription`
: The attribute that corresponds to group descriptions on your LDAP server.

`member`
: The attribute that describes the list of users in a group.

`distinguishedNameAttribute`
: The attribute that stores the distinguished names of users. This attribute is optional, but it greatly improves the performance of LDAP queries.

`serverSideSorting`

Whether the LDAP server is configured for server-side sorting of query results. If your LDAP server supports server-side sorting, set this option to `true`. If you are unsure whether your LDAP server supports this, enter `false`, as misconfiguration may cause errors.

`rangeRetrieval`

Whether the LDAP server is configured to return a range of query results for a request. This means that groups with many users will be requested in small sets instead of all at once. LDAP servers that support range retrieval will perform better for large queries. If your LDAP server supports range retrieval, set this option to `true`. If you are unsure whether your LDAP server supports range retrieval, enter `false`, as misconfiguration may cause errors.

`groupClassNames`

By default Tableau Server looks for LDAP group object classes containing the string “group”. If your LDAP group objects do not fit the default class name, override the default by setting this value. You can provide multiple classnames separated by commas. This option takes a list of strings, which requires passing each class in quotes, separated by a comma (no space) and within brackets. For example: `["basegroup","othergroup"]`.

`userClassNames`

By default Tableau Server looks for LDAP user object classes containing the string “user” and “inetOrgPerson”. If your LDAP user objects do not use these default class names, override the default by setting this value. You can provide multiple classnames separated by commas. This option takes a list of strings, which requires passing each class in quotes, separated by a comma (no space) and within brackets. For example: `["userclass1",userclass2”]`.

Importing the JSON file

After you have finished editing the JSON file, pass the file and apply settings with the following commands:

```
tsm settings import -f path-to-file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

kerberosSettings Entity

Before you configure Kerberos authentication, review Kerberos Requirements.

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f /path/to/file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Configuration template

Use this template to configure Kerberos settings.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

After you have finished with the initial configuration of Kerberos authentication, use the tsm authentication kerberos <commands> sub-category to set additional values.

```
{
  "configEntities": {
       "kerberosSettings": {
               "_type": "kerberosSettingsType",
               "enabled": "true",
               "keytabFile": "/path/to/keytab_file"
        }
     }
 }
```

Configuration file reference

The following list includes all of the options that can be included with the `"kerberosSettings"` entity set.

`Option`
: Value

`enabled`

: Options: `true` or `false`.

: Enables Kerberos authentication.

`keytabFile`

: Required.

: Path to valid Kerberos keytab file.

`dBClasses`
: Comma-separated list of database classes for global credentials. May be required for connecting to Cloudera data sources.

mutualSSLSettings Entity

Before you configure mutual SSL, review Configure SSL for External HTTP Traffic to and from Tableau Server.

The `mutualSSLSettings` entity combines both SSL and mutual SSL configuration. Mutual SSL requires that external SSL has been enabled and properly configured.

The TSM entities use JSON and key-value pairs. Use the configuration file template below to create a .json file. Provide values for the appropriate keys for your environment, and then pass the .json file to Tableau Server with the following commands:

```
tsm settings import -f <path-to-file.json>

tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Configuration template

Use this template to configure mutual SSL settings.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

```
{
  "configEntities": {
    "mutualSSLSettings": {
```

```
        "_type": "mutualSSLSettingsType",
        "sslEnabled": true,
        "proxyLogin": false,
        "clientCertRequired": true,
        "caCertFile": "required",
        "keyFileName": "required",
        "keyPassphrase": "",
        "chainFile": "",
        "revocationFile": "",
        "redirect": false,
        "fallbackToPassword": true,
        "protocols": "",
        "cipherSuite": "",
        "forceHttpsForPublicEmbed": false
    }
  }
}
```

Configuration file reference

`sslEnabled`

Enable SSL. This is a prerequisite to enabling mutual SSL.

`clientCertRequired (MutualSSL)`

Set to true to enable mutual SSL authentication. Set to false to disable.

`caCertFile (MutualSSL)`

Required.

Specify the CA-issued certificate file for two-way SSL. The file path must be readable by Tableau Server.

`certFileName`

Specify the file that contains the concatenation of PEM encoded CA certificates that form the certificate chain for the server certificate.

Alternatively the referenced file can be the same as caCertFile when the CA certificates are directly appended to the server certificate for convenience.

`keyFileName`

If the key is not combined with the certificate, use this configuration key to point to the key file. If you have both an RSA and a DSA private key, you can configure both in parallel (for example, to also allow the use of DSA ciphers).

`keyPassphrase`

Optional. Passphrase for the certificate file. The passphrase you enter will be encrypted while at rest.

Note: If you create a certificate key file with a passphrase, you cannot reuse the SSL certificate key for SAML.

`revocationFile`

Specifies the file path for an SSL CA Certificate Revocation List (.crl) file.

`Redirect`

Default: true. Specifies whether Tableau Server should redirect http requests as https requests to the appropriate endpoint.

`clientCertMapping (MutualSSL)`

Specifies the method for retrieving the user name from the certificate.

Accepted values: `ldap`, `upn`, `cn`

- For a server using local authentication, the default setting is `upn`(User Principal Name).
- When Tableau Server authentication is configured for Active Directory (AD), the default is `ldap` (Lightweight Directory Access Protocol). This tells the server to go to AD to validate the user, and it ignores the names inside the certificate.

You can set `cn` for either authentication type to use the CN in the Subject DN in the certificate.

For more information, see Mapping a Client Certificate to a User During Mutual Authentication.

`fallbackToPassword (MutualSSL)`

Set to true to give users the option to sign in to Tableau Server through their user name and password if mutual SSL authentication fails. Set to false to disallow this fallback option.

`protocols`

List the Transport Layer Security (TLS) protocol versions you want to allow or disallow.

Default value: `"all -SSLv2 -SSLv3"`

However, we recommend the using the following setting:

```
"all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1"
```

For more information, see tsm security external-ssl enable. For general information, see the Apache online documentation.

`cipherSuite`

List ciphers to allow or disallow for SSL.

Default value:
`"HIGH:MEDIUM:!aNULL:!MD5:!RC4:!3DES:!CAMELLIA:!IDEA:!SEED"`

See the OpenSSL ciphers page for cipher list format. Use caution when changing this option. The default values disallow ciphers that are no longer considered adequately secure.

`proxyLogin`

Default: false. Indicates that Tableau Server uses a proxy for SSL on sign-in only. It controls the protocol the server reports to Tableau Desktop for sign-in APIs.

`forceHTTPForPublicEmbed`

Default value: false. Forces the code for embedded views to use SSL.

openIDSettings Entity

Before you configure OpenID authentication, review Requirements for Using OpenID Connect.

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f path-to-file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Configuration template

Use this template to configure OpenID settings.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

After you have finished with the initial configuration of OIDC, use the tsm authentication openid <commands> sub-category to set additional values.

```
{
       "configEntities": {
           "openIDSettings": {
               "_type": "openIDSettingsType",
               "enabled": true,
               "clientId": "required",
               "clientSecret": "required",
               "configURL": "required if staticFile value is not set",
               "staticFile": "required if configURL value is not set",
               "externalURL": "required"
               }
         }
}
```

Configuration file reference

The following list includes all of the options that can be included with the `"openIDSettings"` entity set.

`_type`

Required.

Do not change.

`enabled`

Required.

Set to `true`.

`clientId`

Required.

Specifies the provider client ID that your IdP has assigned to your application. For example, `"laakjwdlnaoiloadjkwha"`.

`clientSecret`

Required.

Specifies the provider client secret. This is a token that is used by Tableau to verify the authenticity of the response from the IdP. This value is a secret and should be kept securely.

For example, `"fwahfkjaw72123="`.

`configURL`

Required.

Specifies provider configuration URL. If you do not specify a configuration URL, then delete this option and specify a path and file name for `staticFile` instead.

`staticFile`

Required.

Specifies the local path to the static OIDC discovery JSON document. If you do not specify a static file, then delete this option and specify a url for `configURL` instead.

`externalURL`

Required.

The URL of your server. This is typically is the public name of your server, such as `http://example.tableau.com`.

`connectionTimeout`

Optional.

Specifies connection timeout span in seconds. Default is `10`.

`readTimeout`

Optional.

Specifies read timeout span in seconds. Default is `30`.

`ignoreDomain`

Set this to `true` if the following are true:

- You are using email addresses as usernames in Tableau Server
- You have provisioned users in the IdP with multiple domain names
- You want to ignore the domain name portion of the `email` claim from the IdP

Before you proceed, review the user names that will be used as a result of setting this option to `true`. User name conflicts may occur. In the case of a user name conflict, the risk of information disclosure is high. See Requirements for Using OpenID Connect.

`ignoreJWK`

Set this to `true` if your IdP does not support JWK validation. In this case, we recommend authenticating communication with your IdP using mutual TLS or another network layer security protocol. Default is `false`.

`customScope`

Specifies a custom scope user-related value that you can use to query the IdP. See Requirements for Using OpenID Connect.

`idClaim`

Change this value if your IdP does not use the `sub`claim to uniquely identify users in the ID token. The IdP claim that you specify should contain a single, unique string.

`usernameClaim`

Change this value to the IdP claim that your organization will use to match user names as stored in Tableau Server.

`clientAuthentication`

Specifies custom client authentication method for OpenID Connect.

To configure Tableau Server to use the Salesforce IdP, set this value to `client_secret_post`.

`iFramedIDPEnabled`

Set to `true` to allow IdP displayed in an iFrame. The IdP must disable clickjack protection to allow iFrame presentation.

samlSettings Entity

This article contains a template and reference for configuring server-wide SAML on Tableau Server, using a configuration file with keys and values for the `samlSettings` entity. This information supplements the SAML configuration steps in Configure Server-Wide SAML.

To create a SAML configuration template and apply it to Tableau Server, you complete the following steps:

1. Review the following two sections that describe the template and how it's structured (Template categories and definitions and samlSettings configuration template).
2. Paste the JSON code shown in the template into a new text file, and save it using a .json extension.

3. Use the SAML configuration entity reference to help you provide values where required.

4. Add optional key/value pairs specific to your environment. For example, if your SAML certificate key file requires a passphrase, you will need to specify the password in the `wgserver.saml.key.passphrase` parameter using the tsm configuration set command.

5. Pass the configuration file to Tableau Server.

Template categories and definitions

The template uses placeholders for each key value. These placeholders are categorized as follows:

- **Required:** Attributes with the `"required"` value must be replaced with valid data before you run the configuration command. Review the configuration file reference for valid values.
- **Hard-coded:** Attribute names that are prepended with an underscore (_), for example `"_type"` hold hard-coded values. Do not change these values.
- **Default values:** Attributes that are set to a value that is not `"required"` are default values. These are required attributes that you can change as appropriate for your environment.
- **Empty sets:** Values that are empty (`""`) can be passed as they are, or you can provide a value for your installation.

Important: All entity options are case sensitive.

samlSettings configuration template

Paste this code into a text file and customize it for your environment, using the reference below.

```
{
  "configEntities": {
    "samlSettings": {
      "_type": "samlSettingsType",
      "enabled": true,
      "returnUrl": "required",
      "entityId": "required",
      "certFile": "required",
      "keyFile": "required",
      "idpMetadataFile": "required",
      "idpDomainAttribute": "",
      "idpUsernameAttribute": "required"
    }
  }
}
```

SAML configuration entity reference

The following list includes all of the options you can include with the `"samlSettings"` entity set.

`idpMetadataFile`

Required. The path and file name for the XML file generated by the IdP. The XML metadata must include the user name attribute (assertion).

If you completed the steps described in Configure Server-Wide SAML the value you enter here would be:

`/var/opt/tableau/tableau_server/data/saml/<metadata-file.xml>`

`enabled`

true | false

Required. Indicates whether SAML authentication is enabled. Do not set this option to `true` before setting other required SAML configuration options.

`returnURL`

This is typically the external URL that Tableau Server users enter in their browser to access the server, such as `https://tableau_server.example.com`. This value is used to create the ACS URL attribute when configuring the IdP.

`entityId`

Required. Service provider (in this case, Tableau Server) entity ID value.

Identifies your Tableau Server configuration to the IdP. We recommend that you enter the same value as the `returnURL` option.

`idpUsernameAttribute`

Required. In the IdP metadata, find the attribute that is used to specify user name values, and enter the name of that attribute. Default is `username`.

`certFile`

Required. Enter the location and file name of the x509 certificate (.crt) file for SAML. For example:

```
/var/opt/tableau/tableau_server/data/saml/<file.crt>
```

For more information, see SAML Requirements and Configure Server-Wide SAML.

`keyFile`

Required. Specify the location of the private key (.key) file that accompanies the certificate file. For example:

```
/var/opt/tableau/tableau_server/data/saml/<file.key>
```

> **Note**: If you are using a RSA PKCS#8 key that requires a passphrase, you must set the passphrase using a configKey entity (see Configuration File Example) or with tsm configuration set. The key for the passphrase using these methods is `wgserver.saml.key.passphrase`. The value must be a non-null string.

`idpDomainAttribute`

For organizations that use LDAP or Active Directory, this value specifies which SAML attribute Tableau Server will reference to determine the domain name. For example, if your IdP specifies the domain name in the `domain` attribute, then you would specify `domain` for this value. **Note**: For organizations that have users signing in from multiple domains, this value is required.

If you do not provide a value for this key, the value used depends on the Tableau Server identity store setting:

- For local identity store, the `idpDomainAttribute` value is ignored.
- For Active Directory or LDAP identity stores, Tableau uses the FQDN from the configuration setting `wgserver.domain.default`.

  To get the value for `wgserver.domain.default`, you can run the following command:

  ```
  tsm configuration get --key wgserver.domain.default
  ```

`desktopNoSAML`

true | false

Optional. Allow users to use SAML authentication when they sign in from Tableau Desktop.

By default this is not set, so the effective behavior is equivalent to setting it to false. If single sign-on from Tableau client applications does not work with your IdP, you can set this to true to disable SAML authentication through Tableau Desktop.

`appNoSAML`

true | false

Optional. Allow using SAML to sign in from older versions of Tableau Mobile app. Devices running Tableau Mobile app version 19.225.1731 and higher ignore this option. To disable devices running Tableau Mobile app version 19.225.1731 and higher, disable SAML as a client login option on Tableau Server.

`logoutEnabled`

true | false

Optional. Enables single logout for users who have logged on with SAML. The default is `true`.

The IdP configuration metadata must include a single logout endpoint with POST binding.

This setting applies only for server-wide SAML

When set to `false`, Tableau Server will not attempt single logout.

`logoutUrl`

Optional. Enter the URL to redirect to after users sign out of the server. Setting this option requires that `logoutEnabled` is set to `true`.

By default this is the Tableau Server sign-in page. You can specify an absolute or a relative URL.

`maxAuthenticationAge`

Optional. Specifies the maximum number of seconds allowed between a user's authentication with the IdP and processing of the AuthNResponse message. The default value is -1 which means maxAuthenticationAge is unset or ignored by default. Prior to February 2022, the default value was 7200 (2 hours).

To optimize session length use the same timeout value as is set on the IdP.

`maxAssertionTime`

Optional. Specifies the maximum number of seconds, from creation, that a SAML assertion is usable. Default value is 3000 (50 minutes).

`sha256Enabled`

true | false

Optional. The type of signature Tableau Server will use when sending messages to the IdP. When set to `true`, Tableau Server will sign messages with the SHA 256 signature algorithm. When set to `false`, Tableau Server will sign messages with SHA 1. Default is `true`.

This option sets the signature algorithm to the following messages that Tableau Server signs:

- AuthnRequest messages when `signRequests` is enabled.
- LogoutRequest messages if `logoutEnabled` is enabled.

`signRequests`

true | false

Optional. Specifies whether Tableau Server will sign the AuthnRequests that are sent to the IdP. Signed requests are not always necessary for all IdPs. We recommend signing requests to ensure the most secure option when configuring SAML. To verify whether your IdP accepts signed request, inspect the IdP metadata: if `wantAuthnRequestsSigned` is set to `true`, then your IdP will accept signed requests.

Default value is `true`. To disable signed requests, set this option to `false`.

`acceptableAuthnContexts`

Optional. Sets the `AuthNContextClassRef` SAML attribute. This optional attribute enforces validation of certain authentication "contexts" in IdP initiated flows. Set a comma-separated set of values for this attribute. When this attribute is set, Tableau Server validates that the SAML response contains at least one of the values listed. If the

SAML response does not contain one of the configured values, authentication will be rejected, even if the user has successfully authenticated with the IdP.

Leaving this option blank will result in default behavior: any successfully authenticated SAML response will result in a user being granted a session within Tableau Server.

`iFramedIdpEnabled`

true | false

Optional. Default value is `false`, meaning that when users select the sign-in button on an embedded view, the IdP's sign-in form opens in a pop-up window.

When you set it to true, and a server SAML user who is already signed in navigates to a web page with an embedded view, the user will not need to sign in to see the view.

You can set this to true only if the IdP supports signing in within an iframe. The iframe option is less secure than using a pop-up, so not all IdPs support it. If the IdP sign-in page implements clickjack protection, as most do, the sign-in page cannot display in an iframe, and the user cannot sign in.

If your IdP does support signing in via an iframe, you might need to enable it explicitly. However, even if you can use this option, it disables Tableau Server clickjack protection for SAML, so it still presents a security risk.

Pass the configuration file to Tableau Server

After you have provided an appropriate value for each entity you include in the configuration template, use the following commands to pass the .json file and apply settings to Tableau Server.

```
tsm settings import -f path-to-file.json

tsm pending-changes apply
```

See also

After you complete the initial SAML configuration, use tsm authentication mutual-ssl <commands> to set additional values.

For the command-line reference for configuring SAML, see tsm authentication saml <commands>.

sapHanaSettings Entity

Use the sapHanaSettings entity to configure SAML delegation for SAP HANA.

Review Configure SAP HANA SSO before you continue.

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f path-to-file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

SAP HANA SAML settings

The settings in the template below specify the settings for Tableau Server in a SAML delegation scenario with SAP HANA.

Configuration template

Use this template to configure trusted authentication settings.

All that are referenced in configEntities must be located on the local computer. Do not specify UNC paths.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

```
{
  "configEntities": {
       "sapHanaSettings": {
       "_type": "sapHanaSettingsType",
       "enabled": "true",
       "usernameFormat": "username",
       "usernameCase": "preserve",
       "certFile": "path-to-cert_file",
       "keyFile": "path-to-key_file"
        }
       }
}
```

Configuration file reference

This table includes all of the options that can be included with the `"gatewaySettings"` entity set.

`enabled`

Required.

Values: `true` or `false`

`usernameFormat`

Values: `username`, `domain_and_username`, or `email`

Specifies user name credential format.

`usernameCase`

Values: `lower`, `upper`, or `preserve`

Specifies user input name case.

`certFile`

Specifies file path and name for the certificate file on the local computer.

For example, "`/var/opt/tableau/tableau_server/data/saml/saml_cert.crt`".

`keyFile`

Specifies file path and name for the certificate key on the local computer.

For example, "`/var/opt/tableau/tableau_server/data/saml/saml_key.der`".

shareProductUsageDataSettings Entity

Before you configure this entity, see Product usage data to better understand the behavioral and usage data that Tableau collects. (Your confidential database values are never included.)

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f /path/to/file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Configuration template

Use this template to configure product usage data settings.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

```
{
  "configEntities": {
       "shareProductUsageDataSettings": {
               "_type": "shareProductUsageDataSettingsType",
               "enabled": "true"
        }
     }
 }
```

Configuration file reference

The following list includes all of the options that can be included with the `"shareProductUsageDataSettings"` entity set:

`_type`

Required value: `shareProductUsageDataSettingsType`

`enabled`

Options: `true` or `false`

The default, `true`, shares product usage data from your server with Tableau.

trustedAuthenticationSettings Entity

Before you configure trusted authentication, review Trusted Authentication.

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f /path/to/file.json
```

```
tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Configuration template

Use this template to configure trusted authentication settings.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

After you have finished with the initial configuration of trusted authentication, use the tsm authentication trusted <commands> sub-category to set additional values.

```
{
"configEntities": {
       "trustedAuthenticationSettings": {
              "_type": "trustedAuthenticationSettingsType",
              "trustedHosts": ["webserv1","webserv2","webserv3"]
              }
       }
}
```

Configuration file reference

The following list includes all of the options that can be included with the `"trustedAuthenticationSettings"` entity set.

`trustedHosts`

Required.

IP address or host names of web servers that request trusted tickets from Tableau Server.

This option takes a list of strings, which requires passing each IP or host in quotes, separated by a comma (no space) and within brackets. For example: `["192.168.1.101","192.168.1.102","192.168.1.103"]` or `["webserv1","webserv2","webserv3"]`.

The values you specify overwrite previous settings. Therefore, you must include the full list of hosts when you configure this value.

`tokenLength`

Optional.

The value can be set to any integer between 9 and 255, inclusive.

Determines the number of characters in each trusted ticket. The default setting of `24` characters provides 144 bits of randomness. This option is ignored unless `useV2Tickets` is set to `true`, which is not a recommended best practice.

`logLevel`

Optional.

`all|debug|info|warn|error|fatal|off`

Default: `info`

Specifies logging level for processes related to creating and redeeming trusted tickets. See Change Logging Levels.

`timeoutInSeconds`

Optional.

Default: `180`

Specifies the length of time (in seconds) to invalidate trusted tickets after they are created.

`tryCount`

Optional.

Integer.

Default: `10`

Specifies the number of times to attempt to create a trusted ticket entry.

`use9DigitToken`

Optional.

`true` | `false`

Default: `false`

When set to `true`, tickets are 9 digits long (as in version 8.0 and earlier) and the option `tokenLength` is ignored. This option is intended for temporary support of legacy code.

Warning: Setting this option to `true` severely and negatively impacts the security strength of trusted ticket authentication.

`useV2Tickets`

Optional.

`true` | `false`

Default: `false`

Specifies whether Tableau Server should return a legacy URL format for trusted ticket requests. The legacy URL format includes a 24 character, Base64-encoded string. Beginning with Tableau Server 10.3, the URL that is returned has been updated and includes a Base64-encoded UUID and a 24 character secure random string. Only set option this to `true` if you have deployed trusted tickets with custom code that requires the legacy URL format. We recommend instead, updating your custom code to accept the new URL format.

web-data-connector-settings Entity

This entity is used to manage web data connector (WDC) settings. To learn more about using WDCs in Tableau Server, see Web Data Connectors in Tableau Server and tsm data-access.

Use the configuration file template below to create a .json file. After you have filled in the options with the appropriate values, pass the .json file and apply settings with the following commands:

```
tsm settings import -f /path/to/file.json

tsm pending-changes apply
```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Web data connector settings

The web data connector (WDC) settings in the template below specify whether WDCs are enabled, whether refresh of WDCs is enabled, and the primary and secondary safe lists. The safe lists indicate which WDC URLs are approved for use in your Tableau Server installation, and the domains or URLs that a connector can send requests to and receive requests from.

Configuration template

Use this template to configure the WDC settings.

Important: All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

Single WDC

```
{
  "configEntities": {
    "web-data-connector-settings": {
      "_type": "webDataConnectorSettingsType",
      "refreshEnabled": true,
      "whitelist": {
        "https://www.example.com:443/wdc/": {
          "secondaryWhitelist": [
          "https://www.example.com/.*",
          "https://www.coolapi.com/.*"
          ]
        }
      },
      "enabled": true
    }
  }
}
```

Multiple WDCs

```
{
  "configEntities": {
    "web-data-connector-settings": {
      "_type": "webDataConnectorSettingsType",
      "refreshEnabled": true,
      "whitelist": {
        "https://www.example.com:443/wdc/": {
          "secondaryWhitelist": [
          "https://www.example.com/.*"
          ]
        },
        "https://www.mysite.com:443/coolwdc/": {
          "secondaryWhitelist": [
          "https://www.mysite.com/.*",
          "https://www.coolapi.com/.*"
          ]
        }
      },
      "enabled": true
    }
  }
}
```

Configuration file reference

This table includes all of the options that can be included with the `web-data-connector-settings` entity set.

`_type`

Required.

Value: `webDataConnectorSettingsType`

Do not change.

`refreshEnabled`

Set to `false` to disable refresh of WDCs. Defaults to `true`.

`whitelist`

Required.

Can contain one or more matching sets of safe lists and secondary safe lists (one set per WDC). The first URL provided is the safe list, where you specify the WDC URL and port, formatted as follows:

`<scheme>://<host>:<port>/<path>`

For many WDCs the `<port>` value is 443, which is the default port for HTTPS, but you can check the value for your connector by looking at the data source details on Tableau Server or Tableau Cloud.

`secondaryWhitelist`

Required.

Specifies the domains or URLs that a connector can send requests to and receive requests from, for example, external JavaScript libraries, REST APIs, or local files. To add an entire domain to this secondary safe list, you can use a wildcard expression `.*` at the end of the URL, as shown in the following example:

```
https://www.example.com/.*
```

`enabled`

Set to `false` to disable use of WDCs. Defaults to `true`.

tabcmd

Note: The tabcmd command-line utility version 2.0 is available at Tableau tabcmd. This new version allows you to run tabcmd commands on MacOS and Linux, and to authenticate using personal access tokens (PATs). Version 2.0 is built on public endpoints available in the Python-based Tableau Server Client (TSC). This latest version has limited support for Tableau Server.

Tableau provides the tabcmd command-line utility which you can use to automate site administration tasks on your Tableau Server site. For example, creating or deleting users, projects, and groups.

> **Note:** In Tableau versions prior to 2024.1, tabcmd version 1 does not work for exporting vizzes.

This utility is included with Tableau Server, and is automatically installed on the server nodes. You can also run it from other computers, even computers that are not part of your Tableau Server installation, but to do so you need to download the tabcmd installer from the Tableau website. For more information, see Install tabcmd below.

Install tabcmd

> **Note:** These instructions are for installing the tabcmd 1.0 command-line utility. To install the tabcmd 2.0 command-line utility, go to Tableau tabcmd (new window).

When Tableau Server or Tableau Cloud is upgraded to a new version, if an updated version of tabcmd is required, you can download it from the Tableau *Server* Releases page on the Tableau website.

For Tableau Server, we recommend you download the version that matches your server version. For Tableau Cloud, we recommend you always download the latest version to avoid

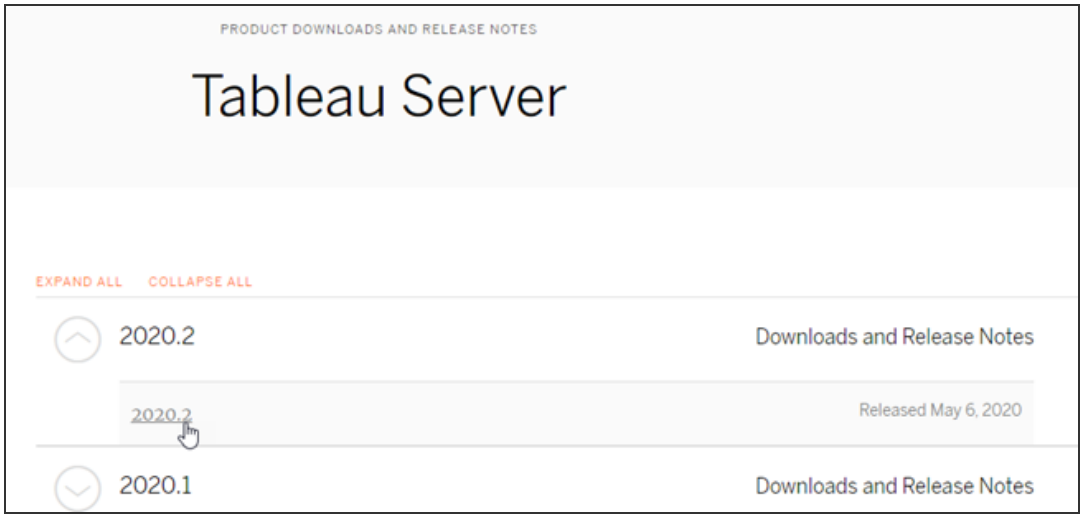
issues caused by version incompatibilities. In either case, using an out of date version of tabcmd can cause errors and unpredictable results.

1. Open a web browser and go to the Tableau Server Releases page. Go to this page even if you use Tableau Online.

2. If you're using:

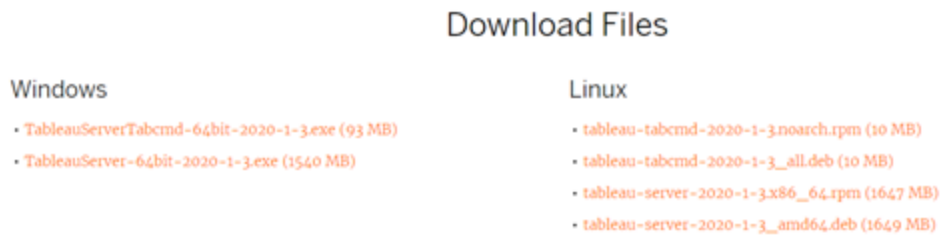
   - **Tableau Cloud**, use Tableau tabcmd 2.0 (new window).
   - **Tableau Server (Windows or Linux):** select the release that matches your server version.

   In either case, if the expanded information shows maintenance releases, select the latest maintenance release or the one that matches your server version.



   This takes you to the release notes page, called Resolved Issues, where you can read about security improvements and resolved issues.

3. Scroll to the **Download Files** section under the resolved issues, select the tabcmd download link that is compatible with the computer on which you'll run the tabcmd commands.



The remaining steps refer to this computer as “the tabcmd computer.”

4. Save the installer to the tabcmd computer, or a location accessible from that computer (a mounted drive, for example).

5. Complete the installation steps as appropriate for the operating system of the tabcmd computer:

- ## Windows

  By default tabcmd is installed to `C:\Program Files\Tableau\Tableau Server\<version>\extras\Command Line Utility`. You can change this during installation and recommend that you install tabcmd to a folder named `tabcmd` at the root of the C:\ drive (`C:\tabcmd`). This can make it easier to locate and run, and will accommodate some limitations with the Windows operating system if you add the tabcmd directory to the Windows PATH.

  > **Note** The tabcmd Setup program does not add the tabcmd directory to the Windows PATH variable. You can add it manually, or you can include the full path to tabcmd each time you call it.

  You can install tabcmd in two ways on Windows:

- Double-click the installer to follow the steps in the UI:

  a. Accept the license agreement.

  b. If you want to install to a non-default location, click **Customize** and type or browse to the location you want to install tabcmd to.

  c. Click **Install**.

     If you are prompted by Windows Defender Firewall or User Account Control, click **Allow access**.

- Run the installer from a command prompt:

  a. Open a command prompt as administrator on the tabcmd computer.

  b. Navigate to the directory where you copied the tabcmd installer.

  c. Install tabcmd:

     ```
     tableau-setup-tabcmd-tableau-<version_code>-
     x64.exe /quiet ACCEPTEULA=1
     ```

     To install to a non-default location:

     ```
     tableau-setup-tabcmd-tableau-<version_code>-
     x64.exe /quiet ACCEPTEULA=1 INSTALLDIR-
     R="<path\to\install\directory>"
     ```

     For example:

     ```
     tableau-setup-tabcmd-tableau-<version_code>-
     x64.exe /quiet ACCEPTEULA=1 INSTALLDIR-
     R="C:\tabcmd"
     ```

     For a complete list of command line options you can use with the tabcmd installer, run the installer with a /?. For more information on

tabcmd installer command line options, see Install Switches and Properties for tabcmd (Windows).

The tabcmd Setup program creates logs in `C:\Users\<user>\AppData\Local\Temp` you can use if you have problems installing tabcmd. The logs use the naming convention `Tableau_Server_Command_Line_Utility_(<version_code>)_##############.log`.

- Linux

  Note: To run tabcmd on a Linux computer, you must have Java 11 installed. On RHEL-like systems, this will be installed as a dependency when you install tabcmd. On Ubuntu systems, you need to install Java 11 separately if it is not already installed.

  As of July 2022, Debian distributions are no longer supported. For more information, see this Tableau Community post.

  a. Log on as a user with sudo access to the tabcmd computer.

  b. Navigate to the directory where you copied the `.rpm` or `.deb` package that you downloaded.

     - On RHEL-like distributions, including CentOS, run the following command:

       ```
       sudo yum install tableau-tabcmd-<version>.noarch.rpm
       ```

     - On Ubuntu, run the following command:

```
sudo apt-get install ./tableau-tabcmd-<version>_
all.deb
```

To uninstall tabcmd from a Linux computer, see the documentation for the Linux variety you are running.

6. (Optional) Add the fully qualified location where tabcmd is installed to your system path to allow you to run tabcmd commands without changing to that location, or specifying the location with each command. Steps to do this depend on the type and version of your operating system. For more information, see PATH_(variable).

How to use tabcmd

The basic steps for using tabcmd are as follows:

1. Open the Command Prompt as an administrator.

   Note: Do not use PowerShell to run tabcmd commands on Windows. Using PowerShell can cause unexpected behavior.

2. On a Windows computer, if you installed tabcmd on a computer other than the initial node, change to the directory where you installed tabcmd.

   On a Linux computer, you do not need to change to the install directory.

3. Run the tabcmd command.

When you use tabcmd, you must establish an authenticated server session. The session identifies the server or Tableau Cloud site and the user running the session. You can start a session first, and then specify your command next, or you can start a session and execute a command all at once.

Important: If you are using tabcmd to perform more than one task, you must run tasks one after another (serially), rather than at the same time (in parallel).

Commands (such as `login`) and the options (such as `-s`, `-u`, etc.) are not case sensitive, but the values you provide (such as `User@Example.com`) are case sensitive.

Examples

The following command demonstrates starting a session with the Tableau Server named *tab-server.mycompany.com*:

```
tabcmd login -s http://tabserver.mycompany.com -u admin -p mypass-
word
```

The next example shows a command that deletes a workbook named *Sales_Workbook*:

```
tabcmd delete "Sales_Workbook"
```

Here's how to accomplish all of the above with one command—note that you do not need `login` here:

```
tabcmd delete "Sales_Workbook" -s http://tabserver.mycompany.com -u
admin -p mypassword
```

A Tableau Server can run multiple sites. When a workbook is on the Default site of a multi-site server you don't need to specify Default, the above command is sufficient. However, if the command applies to something on a site other than Default, you need to specify the site ID for that site (see login). Here's the same command for a workbook that's on the West Coast Sales site (site ID `wsales`):

```
tabcmd delete "Sales_Workbook" -s http://tabserver.mycompany.com -t
wsales -u admin -p mypassword
```

The options `-s`, `-t`, `-u`, and `-p` are among the tabcmd global variables, which can be used with any command.

For more information, see tabcmd Commands.

Status messages and logs

When a command is successful, tabcmd returns a status code of zero. A full error message for non-zero status codes is printed to **stderr**. In addition, informative or progress messages may be printed to **stdout**.

A full log named **tabcmd.log** that includes debugging, progress, and error messages is written to `<home dir>/.tableau/tabcmd/`.

tabcmd Commands

Note: The tabcmd command-line utility version 2.0 is available at Tableau tabcmd. This new version allows you to run tabcmd commands on MacOS and Linux, and to authenticate using personal access tokens (PATs). Version 2.0 is built on public endpoints available in the Python-based Tableau Server Client (TSC). This latest version has limited support for Tableau Server.

> Looking for Tableau Server on Windows? See tabcmd Commands.

You can use the following commands with the tabcmd command line tool:

addusers (to group)
createextracts
creategroup
createproject
createsite
createsiteusers
createusers
decryptextracts
delete *workbook-name* or *datasource-name*
deleteextracts
deletegroup
deleteproject
deletesite

addusers *group-name*

Adds users to the specified group.

Example

```
tabcmd addusers "Development" --users "users.csv"
```

Options

`--users`

Add the users in the given `.csv` file to the specified group. The file should be a simple list with one user name per line. User names aren't case sensitive. The users should already be created on Tableau Server.

For more information, see CSV Import File Guidelines.

`--[no-]complete`

When set to `complete` this option requires that all rows be valid for any change to succeed. If not specified, `--complete` is used.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

createextracts

Creates extracts for a published workbook or data source.

Options

`-d, --datasource`

The name of the target data source for extract creation.

`--embedded-datasources`

A space-separated list of embedded data source names within the target workbook. Enclose data source names with double quotes if they contain spaces. Only available when creating extracts for a workbook.

`--encrypt`

Create encrypted extract.

`--include-all`

Include all embedded data sources within target workbook. Only available when creating extracts for workbook.

`--parent-project-path`

Path of the project that is the parent of the project that contains the target resource. Must specify the project name with --project.

`--project`

The name of the project that contains the target resource. Only necessary if --workbook or --datasource is specified. If unspecified, the default project 'Default' is used.

`-u, -url`

The canonical name for the resource as it appears in the URL.

`-w, -workbook`

The name of the target workbook for extract creation.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

creategroup *group-name*

Creates a group. Use `addusers` (for local groups) to add users after the group has been created. Use `syncgroup` (for Active Directory groups) to create and synchronize a Tableau Server group with an Active Directory group.

Example

```
tabcmd creategroup "Development"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token

remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

createproject *project-name*

Creates a project.

Example

```
tabcmd createproject -n "Quarterly_Reports" -d "Workbooks showing
quarterly sales reports."
```

Options

`-n, --name`

Specifies the name of the project that you want to create.

`--parent-project-path`

Specifies the name of the parent project for the nested project as specified with the `-n` option. For example, to specify a project called "Nested" that exists in a "Main" project, use the following syntax: `--parent-project-path "Main" -n "Nested"`.

`-d, --description`

Specifies a description for the project.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

createsite *site-name*

Creates a site.

Examples

Create a site named West Coast Sales. A site ID of `WestCoastSales` will be automatically created, the site will have no storage quota limit, and site administrators will be able to add and remove users:

```
tabcmd createsite "West Coast Sales"
```

Create a site named `West Coast Sales`with a site ID of `wsales`:

```
tabcmd createsite "West Coast Sales" -r "wsales"
```

Prevent site administrators from adding users to the site:

```
tabcmd createsite "West Coast Sales" --no-site-mode
```

Set a storage quota, in MB:

```
tabcmd createsite "West Coast Sales" --storage-quota 100
```

Options

`-r, --url`

Used in URLs to specify the site. Different from the site name.

`--user-quota`

Maximum number of users that can be added to the site.

`--[no-]site-mode`

Allows or denies site administrators the ability to add users to or remove users from the site.

`--storage-quota`

In MB, the amount of workbooks, extracts, and data sources that can be stored on the site.

`--extract-encryption-mode`

The extract encryption mode for the site can be **enforced**, **enabled** or **disabled**. For more information, see Extract Encryption at Rest.

`--run-now-enabled`

Allow or deny users from running extract refreshes, flows, or schedules manually. **true** to allow users to run tasks manually or **false** to prevent users from running tasks manually. For more information, see Server Settings (General and Customization).

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

createsiteusers *filename.csv*

Adds users to a site, based on information supplied in a comma-separated values (CSV) file. If the user isn't already created on the server, the command creates the user before adding that user to the site.

The CSV file must contain one or more user names and can also include (for each user) a password, full name, license type, administrator level, publisher (yes/no), and email address. For information about the format of the CSV file, see CSV Import File Guidelines.

As an alternative to including administrator level and publisher permissions in the CSV file, you can pass access level information by including the `--role` option and specifying the site role you want to assign users listed in the CSV file.

By default, users are added to the site that you're logged in to. To add users to a different site, include the global `--site` option and specify that site. (You must have permissions to create users on the site you specify.)

If the server contains multiple sites, you can't add server (system) administrators through the `createsiteusers` command. Use `createusers` instead. If you specify the `ServerAdministrator` site role for the `--role` option, the command returns an error. If the CSV file includes `System` as value for administrator, the value is ignored and the user is assigned

the `Unlicensed` license type.

If the server contains only one site (the default site), you can specify `system` for the administrator value for a user, or even assign the `ServerAdministrator` site role using the `--role` option, if you want all users in the CSV file to be server administrators.

By default, this command creates users using a synchronous operation (it waits for all operations to complete before proceeding). You can use the `--no-wait` option to specify an asynchronous operation.

Local authentication

If the server is configured to use local authentication, the information in the CSV file is used to create users.

Active Directory authentication

If the server is configured to use Active Directory authentication, user information is imported from Active Directory, and password and friendly name information in the CSV file is ignored. Further, if a user is specified in the CSV file but no corresponding user exists in Active Directory, the user is not added to Tableau Server. For Active Directory users, because the user name is not guaranteed to be unique across domains, you must include the domain as part of the user name. You can specify this as either `domain\username` or `username@domain.com`; however, we recommend using the `domain\username` format. For more information, see User Management in Deployments with External Identity Stores.

Example

```
tabcmd createsiteusers "users.csv" --role "Explorer"
```

Options

`--admin-type`

Deprecated. Use the `--role` option instead.

`--auth-type`

Sets the authentication type (`Local` or `SAML`) for all users in the `.csv` file. If unspecified, the default is `Local`.

Note: To use SAML authentication, the site must be configured for site-specific SAML in Tableau Server settings. For information, see Configure Site-Specific SAML.

`--[no-]complete`

Deprecated. Default error behavior: if there are more than 3 errors within a ten-row span, then the command will fail.

`--no-publisher`

Deprecated. Use the `--role` option instead.

`--nowait`

Don't wait for asynchronous jobs to complete.

`--publisher`

Deprecated. Use the `--role` option instead.

`--role`

Specifies a site role for all users in the `.csv` file. When you want to assign site roles using the --role option, create a separate CSV file for each site role.

Valid values are: `ServerAdministrator`, `SiteAdministratorCreator`, `SiteAdministratorExplorer`, `SiteAdministrator`, `Creator`, `ExplorerCanPublish`, `Publisher`, `Explorer`, `Interactor`, `Viewer`, and `Unlicensed`.

The default is `Unlicensed` for new users and unchanged for existing users. Users are added as unlicensed also if you have a user-based server installation, and if the `createsiteusers` command creates a new user, but you have already reached the limit on the number of licenses for your users.

> **Note:** On a multi-site Tableau Server, if you want to assign the `Server-Administrator` site role using the `--role` option, use the `createusers` command instead of `createsiteusers`.

`--silent-progress`

Don't display progress messages for the command.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

createusers *filename.csv*

Create users in Tableau Server, based on information supplied in a comma-separated values (CSV) file.

The CSV file must contain one or more user names and can also include (for each user) a password, full name, license type, administrator level, publisher (yes/no), and email address. For information about the format of the CSV file, see CSV Import File Guidelines.

As an alternative to including administrator level and publisher permissions in the CSV file, you can pass access level information by including the `--role` option and specifying the site role you want to assign users listed in the CSV file.

If the server has only one site (the default site), the user is created and added to the site. If the server has multiple sites, the user is created but isn't added to any site. To add users to a site, use `createsiteusers`.

If you have a user-based server installation, and if the command creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user.

Local authentication

If the server is configured to use local authentication, the information in the CSV file is used to create users.

Active Directory authentication

If the server is configured to use Active Directory authentication, user information is imported from Active Directory, and password and friendly name information in the CSV file is ignored. Further, if a user is specified in the CSV file but no corresponding user exists in Active Directory, the user is not added to Tableau Server. For Active Directory users, because the user name is not guaranteed to be unique across domains, you must include the domain as part of the user name. You can specify this as either `domain\username` or `user-`

`name@domain.com`; however, we recommend using the `domain\username` format. For more information, see User Management in Deployments with External Identity Stores.

Example

```
tabcmd createusers "users.csv" --role "ServerAdministrator"
```

```
tabcmd createusers "users.csv"
```

Options

`--admin-type`

Deprecated. Use the `--role` option instead.

`--[no-]complete`

Deprecated. Default error behavior: if there are more than 3 errors within a ten-row span, then the command will fail.

`--no-publisher`

Deprecated. Use the `--role` option instead.

`--nowait`

Don't wait for asynchronous jobs to complete.

`--publisher`

Deprecated. Use the `--role` option instead.

`-r, --role`

Specifies a site role for all users in the `.csv` file. When you want to assign site roles using the --role option, create a separate CSV file for each site role.

Valid values are: `ServerAdministrator`, `SiteAdministratorCreator`, `SiteAdministratorExplorer`, `SiteAdministrator`, `Creator`, `ExplorerCanPublish`, `Publisher`, `Explorer`, `Interactor`, `Viewer`, and `Unlicensed`.

On a multi-site server, the command doesn't assign the user to a site. Therefore, the only site roles the command can successfully assign are `ServerAdministrator` and `Unlicensed`. If you specify any other site role, the command assigns the `Unlicensed` role.

On a single-site server, the user is created and added to the default site using the role that you specify.

If you have a user-based server installation, and if the command creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user.

`--silent-progress`

Don't display progress messages for the command.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

decryptextracts

Decrypt all extracts on a site. If no site is specified, extracts on the default site will be decrypted. For more information, see Extract Encryption at Rest.

Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

Example

```
tabcmd decryptextracts "West Coast Sales"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

delete *workbook-name* or *datasource-name*

Deletes the specified workbook or data source from the server.

This command takes the name of the workbook or data source as it is on the server, not the file name when it was published.

Example

```
tabcmd delete "Sales_Analysis"
```

Options

`-r, --project`

The name of the project containing the workbook or data source you want to delete. If not specified, the “Default” project is assumed.

`--parent-project-path`

Specifies the name of the parent project for the nested project as specified with the `-r` option. For example, to specify a project called "Nested" that exists in a "Main" project, use the following syntax: `--parent-project-path "Main" -r "Nested"`.

`--workbook`

The name of the workbook you want to delete.

`--datasource`

The name of the data source you want to delete.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

deleteextracts

Deletes extracts for a published workbook or data source.

Options

`-d, --datasource`

The name of the target data source for extract deletion.

`--embedded-datasources`

A space-separated list of embedded data source names within the target workbook. Enclose data source names with double quotes if they contain spaces. Only available when deleting extracts for a workbook.

`--encrypt`

Create encrypted extract.

`--include-all`

Include all embedded data sources within target workbook.

`--parent-project-path`

Path of the project that is the parent of the project that contains the target resource. Must specify the project name with --project.

`--project`

The name of the project that contains the target resource. Only necessary if --workbook or --datasource is specified. If unspecified, the default project 'Default' is used.

`-u, -url`

The canonical name for the resource as it appears in the URL.

`-w, -workbook`

The name of the target workbook for extract deletion.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

deletegroup *group-name*

Deletes the specified group from the server.

Example

```
tabcmd deletegroup "Development"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either

an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

deleteproject *project-name*

Deletes the specified project from the server.

Using `tabcmd`, you can specify only a top-level project in a project hierarchy. To automate tasks you want to perform on a project within a parent project, use the equivalent Tableau REST API call.

Example

```
tabcmd deleteproject "Designs"
```

Option

`--parent-project-path`

Specifies the name of the parent project for the nested project as specified with the command. For example, to specify a project called "Designs" that exists in a "Main" project, use the following syntax: `--parent-project-path "Main" "Designs"`.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an

empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

deletesite *site-name*

Deletes the specified site from the server.

Example

```
tabcmd deletesite "Development"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

deletesiteusers *filename.csv*

Removes users from from the site that you're logged in to. The users to be removed are specified in a file that contains a simple list of one user name per line. (No additional information is

required beyond the user name.)

By default, if the server has only one site, or if the user belongs to only one site, the user is also removed from the server. On a Tableau Server Enterprise installation, if the server contains multiple sites, users who are assigned the site role of **Server Administrator** are removed from the site but aren't removed from the server.

If the user owns content, the user's role is change to **Unlicensed**, but the user isn't removed from the server or the site. The content is still owned by that user. To remove the user completely, you must change the owner of the content and then try removing the user again.

If the user was imported from Active Directory, the user is removed from the site and possibly from the server. However, the user isn't deleted from Active Directory.

Example

```
tabcmd deletesiteusers "users.csv"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

deleteusers *filename.csv*

Deletes the users listed in the specified comma-separated values (`.csv`) file.

The `.csv` file should contain a simple list of one user name per line.

Example

```
tabcmd deleteusers "users.csv"
```

Options

`--[no-]complete`

When set to `--complete` this option requires that all rows be valid for any change to succeed. If not specified, `--complete` is used.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

editdomain

> **Note:** As a best practice, you should back up Tableau Server before you edit the domain. The domain is saved in the Tableau repository, and if it's incorrectly changed, administrators may not be able to sign in.

Changes the nickname or full domain name of an Active Directory domain on the server. A domain “nickname” is the Windows NetBIOS domain name.

You can modify the nickname for any domain the server is using. In general, you can modify the full domain name for any domain except the one that you used to sign in. However, if the user name that you're currently signed in with exists in both the current domain and the new domain, you can modify the full name for the current domain.

To ensure that Tableau Server can connect to other Active Directory domains, you must also specify secondary domains thatTableau Server connects to by setting the `wgserver.domain.whitelist` option with TSM. For more information about secondary domains and configuring the connection, see wgserver.domain.whitelist .

Review User Management in Deployments with External Identity Stores to understand how multiple domains, domain name mapping, and user names interact with Tableau Server.

To see a list of domains, use listdomains.

Examples

```
tabcmd editdomain --id 2 --nickname "new-nickname"
```

```
tabcmd editdomain --id 3 --name "new-name"
```

Options

`--id`

The ID of domain to change. To get a list of domain IDs, use use listdomains.

`--name`

The new name for the domain.

`--nickname`

The new nickname for the domain.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

editsite *site-name*

Changes the name of a site or its web folder name. You can also use this command to allow or deny site administrators the ability to add and remove users, or prevent users from running certain tasks manually. If site administrators have user management rights, you can specify how many users they can add to a site.

Examples

```
tabcmd editsite wc_sales --site-name "West Coast Sales"
```

```
tabcmd editsite wc_sales --site-id "wsales"
```

```
tabcmd editsite wsales --status ACTIVE

tabcmd editsite wsales --user-quota 50
```

Options

`--site-name`

The name of the site that's displayed.

`--site-id`

Used in the URL to uniquely identify the site.

`--user-quota`

Maximum number of users who can be members of the site.

`--[no-]site-mode`

Allow or prevent site administrators from adding users to the site.

`--status`

Set to `ACTIVE` to activate a site, or to `SUSPENDED` to suspend a site.

`--storage-quota`

In MB, the amount of workbooks, extracts, and data sources that can be stored on the site.

`--extract-encryption-mode`

The extract encryption mode for the site can be **enforced**, **enabled** or **disabled**. For more information, see Extract Encryption at Rest. Depending on the number and size

of extracts, this operation may consume significant server resources.

`--run-now-enabled`

Allow or deny users from running extract refreshes, flows, or schedules manually. **true** to allow users to run tasks manually or **false** to prevent users from running tasks manually. For more information, see Server Settings (General and Customization).

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

encryptextracts

Encrypt all extracts on a site. If no site is specified, extracts on the default site will be encrypted. For more information, see Extract Encryption at Rest.

Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

Example

```
tabcmd encryptextracts "West Coast Sales"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

export

Exports a view or workbook from Tableau Server and saves it to a file. This command can also export just the data used for a view. View data is exported at the summary level. To export detail-level data, you must use the Tableau Server UI. For details, see Download Views and Workbooks.

Note the following when you use this command:

- **Permissions**: To export, you must have the **Export Image** permission. By default, this permission is Allowed or Inherited for all roles, although permissions can be set per workbook or view.
- **Exporting data**: To export just the data for a view, use the `--csv` option. This exports the summary data used in a view to a .csv file.
- **Specifying the view, workbook, or data to export**:
  - Use part of the URL to identify what to export, specifically the "`workbook/view`" string as it appears in the URL for the workbook or view. Don't use

the “friendly name,” and exclude the `:iid=<n>` session ID at the end of the URL.

For example, the Tableau sample view *Order Details* in the *Superstore* workbook has a URL similar to this: `<server_name>/#/views/Superstore/OrderDetails?:iid=2`

To export the *Order Details* view, use the string `Superstore/OrderDetails`.

Do *not* use `Superstore/Order Details`, or `Superstore/OrderDetails?:iid=2`.

- If the server is running multiple sites and the view or workbook is on a site other than Default, Use `-t <site_id>`.
- To export a workbook, get the URL string by opening a view in the workbook, and include the view in the string you use.

  In the above example, to export the *Superstore* workbook, use the string `Superstore/OrderDetails`.
- To export a workbook, it must have been published with **Show Sheets as Tabs** selected in the Tableau Desktop Publish dialog box.

  Note: The Tableau workbook that contains the administrative views can't be exported.
- To filter the data you download, add a parameter filter using this format:

  `?<filter_name>=value`

  or, if filtering on a parameter and that parameter has a display name that matches the name of a measure or dimension:

  `?Parameters.<filter_name>=value`

- **The saved file's format**: Your format options depend on what's being exported. A workbook can only be exported as a PDF using the `--fullpdf` argument. A view can be exported as a PDF (`--pdf`) or a PNG (`--png`).
- **The saved file's name and location** (optional): If you don't provide a name, it will be derived from the view or workbook name. If you don't provide a location, the file will be saved to your current working directory. Otherwise, you can specify a full path or one that's relative to your current working directory.

  Note: You must include a file name extension such as `.csv` or `.pdf`. The command doesn't automatically add an extension to the file name that you provide.

- **Dashboard web page objects not included in PDF exports**: A dashboard can optionally include a web page object. If you're performing an export to PDF of a dashboard that includes a web page object, the web page object won't be included in the PDF.
- **Non-ASCII and non-standard ASCII characters and PDF exports**: If you're exporting a view or workbook with a name that includes a character outside the ASCII character set, or a non-standard ASCII character set, you need to URL encode (percent-encode) the character.

  For example if your command includes the city `Zürich`, you need to URL encode it as `Z%C3%BCrich`:

  ```
  tabcmd export "/Cities/Sheet1?locationCity=Z%C3%BCrich" -full-
  pdf
  ```

Clearing the Cache to Use Real-Time Data

You can optionally add the URL parameter `?:refresh=yes` to force a fresh data query instead of pulling the results from the cache. If you're using tabcmd with your own scripting and the `refresh` URL parameter is being used a great deal, this can have a negative impact

on performance. It's recommended that you use `refresh` only when real-time data is required—for example, on a single dashboard instead of on an entire workbook.

Examples

Views

```
tabcmd export "Q1Sales/Sales_Report" --csv -f "Weekly-Report.csv"
```

```
tabcmd export -t Sales "Sales/Sales_Analysis" --pdf -f "C:\Tableau_
Workbooks\Weekly-Reports.pdf"
```

```
tabcmd export "Finance/InvestmentGrowth" --png
```

```
tabcmd export "Finance/InvestmentGrowth?:refresh=yes" --png
```

Workbooks

```
tabcmd export "Q1Sales/Sales_Report" --fullpdf
```

```
tabcmd export "Sales/Sales_Analysis" --fullpdf --pagesize tabloid -f
"C:\Tableau_Workbooks\Weekly-Reports.pdf"
```

Options

`-f, --filename`

Saves the file with the given filename and extension.

`--csv`

View only. Export the view's data (summary data) in `.csv` format.

`--pdf`

View only. Export as a PDF.

`--png`

View only. Export as an image in `.png` format.

`--fullpdf`

Workbook only. Export as a PDF. The workbook must have been published with **Show Sheets as Tabs** enabled.

`--pagelayout`

Sets the page orientation (`landscape` or `portrait`) of the exported PDF. If not specified, its Tableau Desktop setting will be used.

`--pagesize`

Sets the page size of the exported PDF as one of the following: `unspecified`, `letter`, `legal`, `note folio`, `tabloid`, `ledger`, `statement`, `executive`, `a3`, `a4`, `a5`, `b4`, `b5`, or `quarto`. Default is `letter`.

`--width`

Sets the width in pixels. Default is 800 px.

`--height`

Sets the height in pixels. Default is 600 px.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an

empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

get *url*

Gets the resource from Tableau Server that's represented by the specified (partial) URL. The result is returned as a file.

Note the following when you use this command:

- **Permissions**: To get a file, you must have the **Download/Web Save As** permission. By default, this permission is allowed or inherited for all roles, although permissions can be set per workbook or view.
- **Specifying a view or workbook to get**: You specify a view to get using the "`/views/<workbookname>/<viewname>.<extension>`" string, and specify a workbook to get using the "`/workbooks/<workbookname>.<extension>`" string. Replace `<workbookname>` and `<viewname>` with the names of the workbook and view as they appear in the URL when you open the view in a browser and replace <extension> with the type of file you want to save. Don't use the session ID at the end of the URL (`?:iid=<n>`) or the "friendly" name of the workbook or view.

  For example, when you open a view *Regional Totals* in a workbook named *Metrics Summary*, the URL will look similar to this:

  ```
  /views/MetricsSummary_1/RegionalTotals?:iid=1
  ```

  Use the string `/views/MetricsSummary_1/RegionalTotals.<extension>` to get the view.

Use the string `/workbooks/MetricsSummary_1.<extension>` to get the workbook.

When downloading workbooks and views from Tableau Server, the content of the .twb or .twbx file is stored in plain text. All data, including filter values that may give semantic clues to the data, will be readable by anyone who opens the file.

- **File extension**: The URL must include a file extension. The extension determines what's returned. A view can be returned in PDF, PNG, or CSV (summary data only) format. A Tableau workbook is returned as a TWB if it connects to a published data source or uses a live connection, or a TWBX if it connects to a data extract.

  Note: If you're downloading a view to a PDF or PNG file, and if you include a `--filename` parameter that includes the .pdf or .png extension, you don't have to include a .pdf or .png extension in the URL.

- **The saved file's name and location** (optional): The name you use for `--filename` should include the file extension. If you don't provide a name and file extension, both will be derived from the URL string. If you don't provide a location, the file is saved to your current working directory. Otherwise, you can specify a full path or one that's relative to your current working directory.

- **PNG size** (optional): If the saved file is a PNG, you can specify the size, in pixels, in the URL.

Clearing the cache to use real-time data

You can optionally add the URL parameter `?:refresh=yes` to force a fresh data query instead of pulling the results from the cache. If you're using tabcmd with your own scripting, using the `refresh` parameter a great deal can have a negative impact on performance. It's recommended that you use `refresh` only when real-time data is required—for example, on a single dashboard instead of on an entire workbook.

Examples

Views

```
tabcmd get "/views/Sales_Analysis/Sales_Report.png" --filename
"Weekly-Report.png"

tabcmd get "/views/Finance/InvestmentGrowth.pdf" -f "Q1Growth.pdf"

tabcmd get "/views/Finance/InvestmentGrowth" -f "Q1Growth.pdf"

tabcmd get "/views/Finance/InvestmentGrowth.csv"

tabcmd get "/views/Finance/InvestmentGrowth.png?:size=640,480" -f
growth.png

tabcmd get "/views/Finance/InvestmentGrowth.png?:refresh=yes" -f
growth.png
```

Workbooks

```
tabcmd get "/workbooks/Sales_Analysis.twb" -f "C:\Tableau_Work-
books\Weekly-Reports.twb"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

initialuser

Create the initial Server administrative user on a server that doesn't have an initial administrative user defined. This creates a Server administrator. This doesn't create a TSM administrator.

Enclose values in single quotes.

> **Note**: The **tabcmd initialuser** command doesn't require authentication to Tableau Server, but you must run the command on the initial server node.

Notes:

- The **tabcmd initialuser** command doesn't require authentication to Tableau Server, but you must run the command on the initial server node.
- The `username` value cannot include an at sign (@) unless the user name suffix matches Tableau Server's primary domain. For example, if Tableau Server connects to domain "myco.com", a user name of "user@example.com@myco.com" cannot be used.

Examples

```
tabcmd initialuser --username 'admin' --password 'password' --
server http://localhost
```

```
tabcmd initialuser --username 'admin' --password 'password' --
friendly 'Tableau Admin' --server http://localhost
```

To prompt for the password in the shell, don't include the `--password` parameter in the command. For example:

```
tabcmd initialuser --username 'admin' --server http://localhost
```

Options

`-f, --friendly`

Creates the initial Server administrative user with the display name.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

listdomains

Displays a list of the Active Directory domains that are in use on the server, along with their nicknames and IDs. If the server is configured to use local authentication, the command returns only the domain name `local`.

Example

```
tabcmd listdomains
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

listsites

Returns a list of sites to which the logged in user belongs.

Example

```
tabcmd listsites --username adam --password mypassword
```

Options

`--get-extract-encryption-mode`

The extract encryption mode for the site can be **enforced**, **enabled** or **disabled**. For more information, see Extract Encryption at Rest.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an

empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

login

Logs in a Tableau Server user.

Use the `--server`, `--site`, `--username`, `--password` global options to create a session.

Note: When you use the **tabcmd login** command, you can't use SAML single sign-on (SSO), even if the server is configured to use SAML. To log in, you must pass the user name and password of a user who has been created on the server. You will have the permissions of the Tableau Server user that you're signed in as. For more information, see Set Users' Site Roles and Permissions.

If you want to log in using the same information you've already used to create a session, just specify the `--password` option. The server and user name stored in the cookie will be used.

If the server is using a port other than 80 (the default), you will need to specify the port.

You need the `--site` (`-t`) option only if the server is running multiple sites and you're logging in to a site other than the Default site. If you don't provide a password you will be prompted for one. If the `--no-prompt` option is specified and no password is provided the command will fail.

Once you log in, the session will continue until it expires on the server or the `logout` command is run.

Example

Logs user jsmith in to the Tableau Server running on their local machine:

```
tabcmd login -s http://localhost -u jsmith -p password
```

Logs administrator in to the Sales site on sales-server:

```
tabcmd login -s http://sales-server -t Sales -u administrator -p
password
```

```
tabcmd login -s http://sales-server:8000 -t Sales -u administrator
-p password
```

Logs administrator in to the Sales site on sales-server using SSL, but doesn't validate the server's SSL certificate:

```
tabcmd login --no-certcheck -s https://sales-server -t Sales -u
administrator -p password
```

Establishes a forward proxy and port for localhost:

```
tabcmd login --proxy myfwdproxyserver:8888 -s http://localhost -u
jsmith -p password
```

Logs user jsmith in to the reverse proxy using SSL:

```
tabcmd login -s https://myreverseproxy -u jsmith -p password
```

Options

`-s, --server`

If you're running the command from a Tableau Server computer that's on your network, you can use `http://localhost`. Otherwise, specify the computer's URL, such as `http://bigbox.myco.com` or `http://bigbox`.

If the server is using SSL, you will need to specify `https://` in the computer's URL.

For Tableau Cloud, specify the full URL including the pod that your site is deployed to. For example: `https://prod-useast-b.online.tableau.com`.

`-t, --site`

Include this option if the server has multiple sites, and you're logging in to a site other than the default site.

The site ID is used in the URL to uniquely identify the site. For example, a site named West Coast Sales might have a site ID of west-coast-sales.

`-u, --username`

The user name of the user logging in. For Tableau Cloud, the user name is the user's email address.

`-p, --password`

Password for the user specified for `--username`. If you don't provide a password you will be prompted for one.

`--password-file`

Allows the password to be stored in the given `filename.txt` file rather than the command line, for increased security.

`-x, --proxy`

Use to specify the HTTP proxy server and port (Host:Port) for the tabcmd request.

`--no-prompt`

Don't prompt for a password. If no password is specified, the `login` command will fail.

`--no-proxy`

Don't use an HTTP proxy server.

`--cookie`

Saves the session ID on login. Subsequent commands won't require a login. This value is the default for the command.

`--no-cookie`

Don't save the session ID information after a successful login. Subsequent commands will require a login.

`--timeout SECONDS`

The number of seconds the server should wait before processing the `login` command. Default: 30 seconds.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

logout

Logs out of the server.

Example

```
tabcmd logout
```

publish *filename.twb(x)*, *filename.tds(x)*, or *filename.hyper*

Publishes the specified workbook (.twb(x)), data source (.tds(x)), or extract (.hyper) to Tableau Server.

If you're publishing a workbook, by default, all sheets in the workbook are published without database user names or passwords.

The permissions initially assigned to the workbook or data source are copied from the project that the file is published to. Permissions for the published resource can be changed after the file has been published.

If the workbook contains user filters, one of the thumbnail options must be specified.

Example

```
tabcmd publish "analysis.twbx" -n "Sales_Analysis" --db-username
"jsmith" --db-password "secret-password"
```

```
tabcmd publish "analysis_sfdc.hyper" -n "Sales Analysis"
--oauth-username "user-name" --save-oauth
```

If the file isn't in the same directory as tabcmd, include the full path to the file.

Example

```
tabcmd publish "\\computer\volume\Tableau Workbooks\analysis.twbx" -
n "Sales_Analysis" --db-username "jsmith" --db-password "secret-pass-
word"
```

```
tabcmd publish "\\computer\volume\Tableau Workbooks\analysis_sfd-
c.hyper" -n "Sales Analysis" --oauth-username "username" --save-
oauth
```

Options

`-n, --name`

Name of the workbook or data source on the server. If omitted, the workbook, data source, or data extract will be named after filename.

`-o, --overwrite`

Overwrites the workbook, data source, or data extract if it already exists on the server.

`-r, --project`

Publishes the workbook, data source, or data extract into the specified project. Publishes to the “Default” project if not specified.

`--parent-project-path`

Specifies the name of the parent project for the nested project as specified with the `-r` option. For example, to specify a project called "Nested" that exists in a "Main" project, use the following syntax: `--parent-project-path "Main" -r "Nested"`.

`--db-username`

Use this option to publish a database user name with the workbook, data source, or data extract.

`--db-password`

Use this option to publish a database password with the workbook, data source, or extract.

`--save-db-password`

Stores the provided database password on the server.

`--oauth-username`

The email address of the user account. Connects the user through a preconfigured OAuth connection, if the user already has a saved access token for the cloud data source specified in `--name`. Access tokens are managed in user preferences.

For existing OAuth connections to the data source, use this option instead of `--db-username` and `--db-password`.

`--save-oauth`

Saves the credential specified by `--oauth-username` as an embedded credential with the published workbook or data source.

Subsequently, when the publisher or server administrator signs in to the server and edits the connection for that workbook or data source, the connection settings will show this OAuth credential as embedded in the content.

If you want to schedule extract refreshes after publishing, you must include this option with `--oauth-username`. This is analogous to using `--save-db-password` with a traditional database connection.

`--thumbnail-username`

If the workbook contains user filters, the thumbnails will be generated based on what the specified user can see. Can't be specified when `--thumbnail-group` option is set.

`--thumbnail-group`

If the workbook contains user filters, the thumbnails will be generated based on what the specified group can see. Can't be specified when `--thumbnail-username`option

is set.

`--tabbed`

When a workbook with tabbed views is published, each sheet becomes a tab that viewers can use to navigate through the workbook. Note that this setting will override any sheet-level security.

`--append`

Append the extract file to the existing data source.

`--replace`

Use the extract file to replace the existing data source.

`--disable-uploader`

Disable the incremental file uploader.

`--restart`

Restart the file upload.

`--encrypt-extracts`

Encrypt extracts when you publish a workbook, data source, or extract to the server. For more information, see Extract Encryption at Rest.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token

remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

publishsamples

Description

Publishes Tableau Sample workbooks to the specified project. Any existing samples will be overwritten.

Syntax

```
tabcmd publishsamples -n [project name] [Global options]
```

Example

Publish samples to the Inside Sales project on the Default site, as user jsmith.

```
tabcmd publishsamples -n "Inside Sales" -t "" -s localhost --user-
name "jsmith" --password "secret-password"
```

Options

`-n, --name`

Required. Publishes the Tableau samples into the specified project. If the project name includes spaces, enclose the entire name in quotes.

`--parent-project-path`

Specifies the name of the parent project for the nested project as specified with the `-n` option. For example, to specify a project called "Nested" that exists in a "Main" project, use the following syntax: `--parent-project-path "Main" -n "Nested"`.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

reencryptextracts

Reencrypt all extracts on a site with new encryption keys. This command will regenerate the key encryption key and data encryption key. You must specify a site. For more information, see Extract Encryption at Rest.

Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

Examples

```
tabcmd reencryptextracts "Default"
```

```
tabcmd reencryptextracts "West Coast Sales"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

refreshextracts *workbook-name* or *datasource-name*

Performs a full or incremental refresh of extracts belonging to the specified workbook or data source.

This command takes the name of the workbook or data source as it appears on the server, not the file name when it was published. Only an administrator or the owner of the workbook or data source is allowed to perform this operation.

Note: This method will fail and result in an error if your Server Administrator has disabled the **RunNow** setting for the site. For more information, see Tableau Server Settings.

Examples

```
tabcmd refreshextracts --datasource sales_ds
```

```
tabcmd refreshextracts --project "Sales External" --datasource
sales_ds
```

```
tabcmd refreshextracts --project "Sales External" --parent-project-
path "Main" --project "Sales External" --datasource sales_ds

tabcmd refreshextracts --workbook "My Workbook"

tabcmd refreshextracts --url SalesAnalysis

tabcmd refreshextracts --workbook "My Workbook" --addcalculations

tabcmd refreshextracts --datasource sales_ds --removecalculations
```

Options

`--incremental`

Runs the incremental refresh operation.

`--synchronous`

Adds the full refresh operation to the queue used by the Backgrounder process, to be run as soon as a Backgrounder process is available. If a Backgrounder process is available, the operation is run immediately. The refresh operation appears on the Background Tasks report.

During a synchronous refresh, `tabcmd` maintains a live connection to the server while the refresh operation is underway, polling every second until the background job is done.

Note: The `--synchronous` option isn't available for data sources refreshed with Tableau Bridge.

`--workbook`

The name of the workbook containing extracts to refresh. If the workbook has spaces in its name, enclose it in quotes.

`--datasource`

The name of the data source containing extracts to refresh.

`--project`

Use with `--workbook` or `--datasource` to identify a workbook or data source in a project other than *Default*. If not specified, the Default project is assumed.

`--parent-project-path`

Specifies the name of the parent project for the nested project as specified with the `--project` option.

For example:

- To specify a project called "Nested" that exists in a "Main" project, use the following syntax:
- `--parent-project-path "Main" --project "Nested"`
- To specify a project called "Nested2" that is nested within the "Nested" project:
- `--parent-project-path "Main/Nested" --project "Nested2"`

`--url`

The name of the workbook as it appears in the URL. A workbook published as “Sales Analysis” has a URL name of “SalesAnalysis”.

`--addcalculations`

Use with `--workbook` to materialize calculations in the embedded extract of the workbook or `--datasource` to materialize calculations in the extract data source. Adds the operation to the queue used by the Backgrounder process. If a Backgrounder process is available, the operation runs immediately. This operation appears on the Background Tasks for Extracts administrative view.

`--removecalculations`

Use with `--workbook` or `--datasource` to remove calculations that were previously materialized. Adds the operation to the queue used by the Backgrounder process. If a Backgrounder process is available, the operation runs immediately. This operation appears on the Background Tasks for Extracts administrative view.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

reset_openid_sub

Clears OpenID Connect identifiers (sub values) that have already been associated with Tableau Server identities. See Changing IdPs in Tableau Server for OpenID Connect.

Example

```
tabcmd reset_openid_sub --target-username jsmith
```

Options

`--target-username`

Clears sub value for the specified individual user.

`--all`

Clears sub values for all users.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

removeusers *group-name*

Removes users from the specified group.

Example

```
tabcmd removeusers "Development" --users "users.csv"
```

Options

`--users`

Remove the users in the given `.csv` file from the specified group. The file should be a simple list with one user name per line.

`--[no-]complete`

Requires that all rows be valid for any change to succeed. If not specified `--complete` is used.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is

saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

runschedule *schedule-name*

Runs the tasks in the specified schedule for the site that you're currently logged into. You can't run this for all sites using tabcmd. To run the tasks in the schedule for all sites, log into the web interface, from the **Schedules** page, select **All Sites**, and then do a **Run Now** on the schedule.

This command takes the name of the schedule as specified on the server.

This command isn't available for Tableau Cloud.

> **Note:** This method will fail and result in an error if your Server Administrator has disabled the **RunNow** setting for the site. For more information, see Tableau Server Settings.

Example

```
tabcmd runschedule "5AM Sales Refresh"
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

set *setting*

Enables the specified setting on the server. Details about each setting can be seen on the Maintenance page on the server.

Use an exclamation mark in front of the setting name to disable the setting. You can enable or disable the following settings:

- `allow_scheduling`
- `embedded_credentials`
- `remember_passwords_forever`

Example

```
tabcmd set embedded_credentials
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

syncgroup *group-name*

Synchronizes a Tableau Server group with an Active Directory group. If the Tableau Server group doesn't already exist, it's created and synchronized with the specified Active Directory group.

If the group name itself includes an "@" (other than as the domain separator) you need to refer to the symbol using the hex format "\0x40".

Example

```
tabcmd syncgroup "Development"
```

```
tabcmd syncgroup "Dev\0x40West"
```

Note: If you synchronize a group that you're a member of, changes that you make using this command don't apply to your user. For example, if you use this command to remove the administrator right from users in a group that you're a member of, you're still an administrator when the command finishes.

Options

`--grant-license-mode <grant-license-mode>`

Specifies whether a role should be granted on sign in. Default is `on-sync`. Valid values are `on-login`, `on-sync`. If no value is specified, `on-sync` is assumed and the default role will be grated when the group is synchronized. For more information, see Modifying user roles with Grant role on sign in.

`--no-publisher`

Deprecated. Use the `--role` option instead.

`--overwritesiterole`

Allows a user's site role to be overwritten with a less privileged one when using `--role`. By default, a user site role can be promoted when using `--role`, but can't be demoted. Because the `--overwritesiterole` option will demote user site roles, use it with caution.

`--publisher`

Deprecated. Use the `--role` option instead.

`-r, --role`

Specifies a site role for users in the group. The default is `Unlicensed`.

Valid values are: `SiteAdministratorCreator`, `SiteAdministratorExplorer`, `SiteAdministrator`, `Creator`, `ExplorerCanPublish`, `Publisher`, `Explorer`, `Interactor`, `Viewer`, `Unlicensed`.

`--silent-progress`

Don't display progress messages for the command.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

upgradethumbnails

Starts and stops the Upgrade Thumbnails job. To learn more, see Upgrade Thumbnails Job.

Examples

To start the Upgrade Thumbnail job:

```
tabcmd upgradethumbnails --server <serverURL>
```

To stop the in progress Upgrade Thumbnail job:

```
tabcmd upgradethumbnails --server <serverURL> --stop
```

Options

`--stop`

When specified, stops the in progress Upgrade Thumbnails job. If this option isn't specified, the Upgrade Thumbnail job will be started.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

validateidpmetadata

Identifies Tableau Server sites that are configured with IdPs using the insecure digest algorithm, SHA-1. This command also identifies IdPs that are using certificates with an insufficient RSA key size or elliptic curve size.

Note: This command is only available for site-specific SAML. For more information, see Configure Site-Specific SAML.

Options

`--digest-algorithms <ALGORITHMS>`

A space-separated list of digest algorithms. Legal values are `sha1`and `sha256`. If not specified, server uses values from server configuration setting, wgserver.saml.blocklisted_digest_algorithms.

`--min-allowed-elliptic-curve-size <SIZE>`

If not specified, server uses values from server configuration setting, wgserver.saml.min_allowed.elliptic_curve_size.

`--min-allowed-rsa-key-size <SIZE>`

If not specified, server uses values from server configuration setting, wgserver.saml.min_allowed.rsa_key_size.

`--site-names <SITENAMES>`

A space-separated list of site names on which to perform certificate validation. If not specified, then all sites are inspected.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

version

Displays the version information for the current installation of the tabcmd utility.

Example

```
tabcmd version
```

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

`-t, --site`

Indicates that the command applies to the site specified by the Tableau Server site ID, surrounded by single quotes or double quotes. To specify the Default site, use either an empty string with single or double quotes (" or "") or use Default in double quotes ("Default"). Site ID is case-sensitive when using a cached authentication token. If you do not match case you may be prompted for a password even if the token is still valid.

`-x, --proxy`

Host:Port

Uses the specified HTTP proxy.

`--no-prompt`

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

`--no-proxy`

When specified, an HTTP proxy will not be used.

`--no-certcheck`

When specified, tabcmd (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default, the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default, the process will wait until the server responds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

Tip: For Tabcmd 1.0 commands available for Tableau Cloud, see tabcmd commands.

Install Switches and Properties for tabcmd (Windows)

Note: The tabcmd command-line utility version 2.0 is available at Tableau tabcmd. This new version allows you to run tabcmd commands on MacOS and Linux, and to authenticate using personal access tokens (PATs). Version 2.0 is built on public endpoints available in the Python-based Tableau Server Client (TSC). This latest version has limited support for Tableau Server.

You can use the following switches when installing the Tableau Server Command Line Utility (tabcmd) version 2019.4.0 or later from the command line on Windows.

Note: There are no equivalent switches for the Linux version of the tabcmd installer.

Switch	Description	Comments
`/install` | `/repair` | `/uninstall` | `/layout` "<directory>"	Run Setup to either install, repair, or uninstall tabcmd, or with `/layout`, create a complete local copy of the installation bundle in the directory specified.	Default is to install, displaying UI and all prompts. If no directory is specified on a fresh install, `C:\Program Files\Tableau\Tableau Server\<version>\extras\Command Line Utility` is assumed.
`/passive`	Run Setup with minimal UI and no prompts.	
`/quiet` | `/silent`	Run Setup in unattended, fully silent mode. No UI or prompts are displayed.	**Note:** Use either `/silent` or `/quiet`, not both.
`/norestart`	Run Setup without restarting Windows, even if a restart is	**Note:** In certain rare cases, a restart cannot be suppressed, even when this option is used. This is most likely when an earlier system restart was

	necessary.	skipped, for example, during installation of other software.
`/log "<log-file>"`	Log information to the specified file and path. By default log files are created in the user's `%TEMP%` folder with a naming convention of `Tableau_Server_Command_Line_utility_<version_code>.log`.	If no file location is specified, the log file is written to the user's TEMP folder (`C:\Users\<username>\AppData\Local\Temp`). Check this log file for errors after installation. Example: `<Setup file> /silent /log "C:\Tableau\Logs\tabcmd-Install" ACCEPTEULA=1`

Properties	Description	Comments
`ACCEPTEULA=1|0`	Accept the End User	If not included when using `/passive`, `/silent` or `/quiet`, Setup

	License Agreement (EULA). Required for quiet, silent, and passive install. 1 = accept the EULA, 0 = do not accept the EULA.	fails silently. If included but set to 0, Setup fails.
`INSTALLDIR-R="<path\to\installation\directory>"`	Install tabcmd to the specified non-default install location.	Specifies the location to install tabcmd. If not used, tabcmd is installed to `C:\Program Files\Tableau\Tableau Server\<version_code>\extras\Command Line Utility` Example: `<Setup file> /silent INSTALLDIR-R="C:\tabcmd"`

Troubleshooting

You can use the following topics to troubleshoot and resolve issues with Tableau Server.

Troubleshoot Tableau Server on Linux

Follow the suggestions in this topic to resolve common issues with Tableau Server. For additional troubleshooting steps based on process status viewed on the Status page, see Troubleshoot Server Processes.

The following table displays the default locations of the installation, data, logs, and scripts directories:

Directory	Default location
Installation:	`/opt/tableau/tableau_server`
Data:	`/var/opt/tableau/tableau_server/data`
Logs:	`/var/opt/tableau/tableau_server/data/tabsvc/logs/`
Scripts:	`/opt/tableau/tableau_server/packages/scripts.<version_code>/`

General Troubleshooting Steps

Many Tableau Server issues can be addressed or tested with one or more of these basic steps:

Clean install

Install Tableau Server on Linux on a computer that has never had Tableau installed on it. If you are reusing a computer or VM that has had a previous version of Tableau Server installed, follow the steps in Remove Tableau Server from Your Computer to clean Tableau off your computer before you install the new version.

If you run into problems installing Tableau Server you may need to entirely remove Tableau from your computer and do a clean install. See Remove Tableau Server from Your Computer for details.

Disk space

Make sure there is enough disk space on each computer running Tableau Server. Limited disk space can cause a failure to install, a failure to upgrade, or problems running Tableau Server.

Remove old log files

If you are running out of disk space you can clean up old Tableau Server log files. These can take up space and as a best practice you should remove them regularly.

Version 10.5.x

If you have version 10.5.1 and higher, run this command at a terminal prompt to clean up log files you do not need:

```
tsm maintenance cleanup
```

Version 10.5.0

If you are running version 10.5.0 of Tableau Server on Linux, the cleanup command is not available and you need to run these commands at a terminal prompt:

```
sudo find /var/opt/tableau/tableau_server/data/tabsvc/temp/* -mtime
+2 -type f -delete
```

```
sudo find /var/opt/tableau/tableau_server/data/tabsvc/logs/* -mtime
+2 -type f -delete
```

Important: The Linux file system makes it possible to delete files that are open and if you do this the Tableau processes may not be able to recreate the files. This will result in empty log files. To fix this you can stop Tableau Server, restart the TSM Controller, and restart Tableau again:

1. Stop Tableau Server:

   ```
   tsm stop
   ```

2. Restart the TSM Controller:

   ```
   sudo systemctl restart tabadmincontroller_0.service
   ```

3. Wait several minutes for the controller to restart. You can confirm the controller has restarted with this command:

   ```
   tsm status -v
   ```

   When you can run that command and the Tableau Server Administration Controller is listed as 'running' the controller has restarted.

4. Start Tableau Server:

   ```
   tsm start
   ```

Manually gather logs

If you cannot run `tsm maintenance ziplogs` for any reason (for example, if you have a critical failure before you run `tsm initialize`), you can manually collect and zip the logs by running these commands in a terminal window on each node in the Tableau Server deployment:

```
cd /var/opt/tableau/tableau_server/data/tabsvc/

cp /var/opt/tableau/tableau_server/logs/app-install.log logs

cp ~/.tableau/tsm/tsm.log logs

tar -czvf ~/logs.tar.gz logs
```

This creates a file called logs.tar.gz in your home directory. You can upload or send this file to Tableau.

Restart server

Restart Tableau Server. Issues related to indexing and processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau Server, use the `tsm restart` command. This will stop all the processes associated with Tableau Server and then restart them.

Edit installation and configuration files using Linux

You should edit or create any files used to install or configure Tableau Server on Linux using a Linux operating system. Files created using Microsoft Windows will cause errors in Tableau Server on Linux installation and configuration because Linux operating systems end files with a line-feed (LF) character, whereas Windows ends files with a carriage-return character and a line-feed character (CR LF). Non-Linux (CR LF) file endings can cause errors during Automated Installation of Tableau Server if they appear in the `config.json`, `reg_templ.json` or `secrets` files used by the automated installer. Non-Linux (CR LF) file endings can also cause errors during registration or when configuring identity store settings or gateway settings.

Check systemd logs

If Tableau Server will not start, and you do not find anything useful in the Tableau logs (see Work with Log Files for more information) you can check the systemd logs for messages related to the TSM Service starting and stopping. The logs are stored at `/var/log/messages` (RHEL-like distros) or`/var/log/syslog`(Ubuntu). We recommend using the `journalctl` command to search and parse the systemd logs.

Installing Tableau Server

Install fails due to hardware requirements

Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. For details on requirements, see Before you install....

Install fails due to timeouts

If you install Tableau Server on a computer with limited resources, for example, a computer that just meets the minimum hardware requirements, you can run into problems where tsm commands timeout due to slow response. You can specify a longer timeout by using the global `--request-timeout` option on all tsm commands. For more information on the `--request-timeout` option, see for example, tsm initialize.

Install fails with "Failed to initialize the instance of the temporary database"

Tableau Server on Linux only supports UTF-8 character encoding. If your Linux locale is missing the UTF-8 encoding, your installation can fail with an error similar to this one:

```
Failed to initialize the instance of the temporary database
```

To check if your locale is using UTF-8 encoding, run the `localectl` command at a command prompt. The resulting output should look something like this (your locale may be different):

```
[tableauserver-centos1a ~]$ localectl
System Locale: LANG=en_US.UTF-8
[tableauserver-centos1a ~]$
```

If the `LANG` value does not include `.UTF-8` then you need to run `localectl` to add it:

```
sudo localectl set-locale LANG=<your_locale>.UTF-8
```

> **Note:** In some cases `localectl` may not complete (timeout) if your version of `systemd` is old. Updating systemd may fix this problem and allow you to set the UTF-8 encoding. On RHEL-like systems, use this command to update `systemd`: `sudo yum update systemd`

Installation fails on a virtual machine in Parallels

Parallels is currently not supported. If you install Tableau Server on a Linux virtual machine in Parallels, the install might fail.

Tableau Server doesn't start

If Tableau Server does not start or is running in a degraded state, run the `tsm restart` command. This will shut down any processes that are running, and restart Tableau Server.

Cannot start Tableau Server after installation

Tableau Server might not start if your computer's hostname changes after installation. One of the main reasons why the hostname might change is if you use the cloud-init package on CentOS. If you use the cloud-init package, *reboot the computer* where you want to install Tableau Server before you begin the installation process. Alternatively, you can fix the hostname without rebooting by running the following command:

```
sudo hostnamectl set-hostname `hostnamectl --static`
```

The cloud-init package is commonly used to initialize new virtual machines, configure SSH public key authentication, and more. For example, some CentOS images use cloud-init, and cloud-init is commonly used in OpenStack deployments. However, the version of cloud-init included by default in the CentOS 7.x repositories (cloud-init 0.7.5-10.el7.centos.1) has a known issue that prevents your computer from displaying its Fully Qualified Domain Name (FQDN) along with its hostname until after it restarts.

Because the Tableau Server installation process uses your computer's hostname to configure server processes and generate TLS certificates, Tableau Server might not start if it is configured to use a hostname without the FQDN.

To determine if your computer is displaying the correct hostname, run the `hostnamectl` command. In the following example, the command displays a transient hostname which indicates that it will not return the FQDN and must be restarted.

```
$ hostnamectl
   Static hostname: server01.example.com
Transient hostname: server01
[...]
```

Alternatively, in the following example, the command displays the correct hostname and FQDN:

```
$ hostnamectl
   Static hostname: server01.example.com
[...]
```

Cannot create initial administrator account with multiple Active Directory (AD) domains

When you create the initial administrator account on Tableau Server, you might see the following error if you selected AD as the authentication type:

```
Failed to authenticate username and password
```

This occurs when Tableau Server attempts to connect to multiple AD domains. For example, you might see this error if you install Tableau Server on a computer that is part of one domain and you attempt to authenticate AD users that are part of another domain.

Fonts

Tableau Server uses the fonts installed on the system to render workbooks based on the fonts used when a workbook was created. When a font is not available, Tableau Server will use the closest equivalent based on font families; this is true for both Windows and Linux Servers. On Linux Servers missing fonts may be more obvious because Linux ships with fewer included fonts than Windows and OS/X systems do. This matters because many workbooks are authored in Tableau Desktop on Windows or on Mac.

Tableau Server on Linux ships with the following fonts:

- Arial
- Courier
- Georgia
- Times New Roman
- Verdana
- Trebuchet MS
- Tableau Font

Workbooks which use fonts other than these may appear differently than expected when viewed on Tableau Server on Linux, due to missing fonts. To resolve this issue, install the appropriate fonts onto all nodes in your Tableau Server installation.

Support for Asian character sets

If you see empty boxes where you expect to see Asian characters in workbooks that are displayed on Tableau Server, then you should install the language-appropriate font packages in your Linux environment.

Initializing Tableau Server

TSM initialization fails because the `tableau` user account exists but is not a member of the group `tableau`

When you install and initialize Tableau Services Manager (TSM) and Tableau Server, the initialization script (`initialize-tsm`) creates the users and groups needed to run, or confirms that the existing ones are configured with the required characteristics. By default the script creates a user called `tableau` and adds it to a group called `tableau`. If a `tableau` user already exists but is not part of the `tableau` group, the script fails with a warning.

If this happens you can fix the conflict by using a `--unprivileged-user` flag to specify a different user, and the user will be created and added to the `tableau` group.

For example, to specify a user named `tableauserver`, you would run the script from the `/opt/tableau/tableau_server/packages/scripts.<version_code>` directory using this command:

```
sudo ./initialize-tsm --unprivileged-user="tableauserver" --
accepteula
```

For a complete list of options that can be used with the `initialize-tsm` script, use the -h option:

```
sudo ./initialize-tsm -h
```

Error initializing Tableau Server on unsupported system locale

If you attempt to install Tableau Server on a computer with a locale that is not one of the supported locales, you will get an error during installation.

Tableau Server will run on a system using one of the following locales:

de_DE, en_GB, en_US, es_ES, fr_FR, it_IT, ja_JP, ko_KR, pt_BR, zh_CN, zh_TW

fr_CA (as of version 2022.3)

th_TH, sv_SE (version 2023.1)

Any other locale will generate the error.

Error initializing Tableau Server when en_US.utf8 is not included in locale list

If you attempt to install Tableau Server on a computer that does not have `en_US.utf8` in the locale list, the initialization will fail with an error. To see if `en_US.utf8` is included, type `locale -a` at a shell prompt.

If `en_US.utf8` is not listed, you can en_us to the locale list by typing `sudo locale-gen en_US.UTF-8` at a shell prompt on Ubuntu, or `sudo localedef -i en_US -f UTF-8` at a shell prompt on RHEL-like distributions.

Error: status 10 - initializing Tableau Server when data directory path includes a period

If you attempt to install Tableau Server and specify a data directory with a path that includes a period ("."), initialization will fail with errors including:

```
Connection timed out
```

and

```
ERROR: TSM services returned status 10
```

To avoid this issue, choose a data directory that does not include a period in its path.

Error initializing Tableau Server after reinstallation

If you uninstall and reinstall Tableau Server, you can encounter an error initializing Tableau Server. For example, you might see the following error:

```
ERROR com.tableau.tabadmin.webapp.asyncjobs.JobStepRunner - Running
step WaitForConfigure failed
```

```
com.tableau.tabadmin.webapp.exceptions.ServiceFailedStateException
```

This error occurs when artifacts remain from a previous installation that cause services to fail to start. To prevent this error, use the `tableau-server-obliterate` script in the `/opt/tableau/tableau_server/packages/scripts.<version_code>` folder. For more information about completely removing Tableau Server, see Remove Tableau Server from Your Computer.

Activating Tableau Server

Tableau Server license activation fails

In certain cases activation of the Tableau product key using the `tsm licenses activate -k <product_key>` command fails with an error:

```
License Server not available
```

This can happen if your computer is unable to connect through TCP port 443 to the Tableau licensing server at `licensing.tableau.com`.

To resolve this you need to configure your network and/or host-based firewalls to allow access to that address and port, or activate Tableau offline. For more information, see Activate Tableau Server Offline.

Reindexing Tableau Server Search & Browse

Problems that can be solved by reindexing Search & Browse

Symptoms of an index that needs to be rebuilt include:

- A blank list of sites when a user attempts to log in
- A blank list of projects when a user tries to select a project
- Missing content (workbooks, views, dashboards)
- Unexpected or inaccurate alerts (for example, an "refresh failed" alert on a workbook that does not include an extract)

If you see any of these behaviors, rebuild the Search & Browse index using the `tsm maintenance reindex-search` command.

Restarting Tableau Server

Restarting Tableau Server or applying changes fails

If one of the Tableau Server services fails, you might see an error when you attempt to restart the server or apply configuration changes.

To see if a failed service is causing the error, type the following command:

```
tsm status -v
```

To find out why a service failed, view the tabadminagent and tabadmincontroller log files in the data directory. For example, a service might fail because of concurrency issues or port configuration issues. Please include any issues you encounter in your feedback.

As a workaround, you can attempt to resolve the failure by removing and re-adding the service in TSM. Once the service has started, you can retry your previous configuration change or retry restarting the server with the `tsm restart` command.

Error restarting Tableau Server after adding or configuring a node

If you add a or configure the node without a Gateway process, Tableau Server might fail to restart and you could see errors like these:

```
ERROR : com.t-
ableau.tabadmin.configuration.PortConfigurationExtractor - Unable to
find port config key worker1.gateway.port
```

and

```
Message: Missing port configuration value for key 'work-
er1.gateway.port'
```

These errors appear in the gateway.log file and occur when a Tableau Server node is configured with either an Application Server or VizQL Server but without a Gateway. A Gateway process is required if either Application Server or VizQL Server is running on a node.

Backup/Restore

Problems related to restoring a backup created by Tableau Server can be the result of permissions issues. Proper permissions are necessary for both the file that TSM is restoring, and the location of the file. When TSM handles the backup, it puts the file in a default location and sets permissions appropriately. You can run into permission problems if you are restoring a backup that was copied to your Linux server, or a backup from a non-default location on your server. For details about using a non-default location, including how to change the location, see tsm File Paths.

Errors may include:

```
Server Was Denied Access to File
```

or

```
Restoring the backup '<backup>.tsbak' was unsuccessful
```

or

```
Comparing authentication methods failed
```

The Tableau Server backup and restore processes must have:

- Read permission—The processes need to access the `.tsbak` backup file directly.
- Execute permission—The processes also require execute permissions to the directory structure in which the `.tsbak` file is placed.

When TSM creates a backup in the default location, it sets the permissions it needs. If you copy a file to the Linux server, or move it to a non-default directory, the permissions may not allow the TSM processes proper access. You need to verify that both the file, and the directory tree that contains it, allow access by the TSM user *tableau*. The file permissions must give the *tableau* user read access to the `.tsbak` file. You can do this by setting the group on the file to the *tableau* group, and giving that group read access. The directory permissions

must give the *tableau* user read access. You can do this by setting the group on the directory to the *tableau* group, and giving that group read and execute access on the directories.

For detailed information about TSM and file permissions, see Files and Permissions in TSM.

File locations

Changing basefilepath does not change the location of an existing file

Several tsm commands write files to default locations. You can change these default locations for each command using a tsm set command, but doing so does not move any existing files from the original location to the new one, and it does not create the new location. You are responsible for creating the new location, and for making sure it has the correct permissions to allow tsm access to any files in the location, and the entire directory structure that contains the files.

For more information about changing default locations for backup, restore, site import and export, and ziplogs files, see tsm File Paths.

For information about tsm permissions, see Files and Permissions in TSM.

TSM commands

TSM command line does not show progress for long-running tasks

If you run a tsm command such as restore or ziplogs that takes more than 2 hours to complete, the command will continue to run until completion on the server. To display the progress of the job, use the `tsm jobs reconnect` command.

Opening Firewall ports

Manually opening firewall ports on Ubuntu

The current version of Tableau Server does not support the `ufw` firewall that is used on Ubuntu. For customers that do not want to install `firewalld` on Ubuntu, another option is to manually open those ports. The following steps will confirm that `ufw` is running, and open TCP ports 8850 and 80 to connections from any source address:

1. Run the following command to confirm `ufw` is running:

   ```
   sudo ufw status
   ```

   If the result is `Status: inactive`, you will need to enable `ufw` and ensure that you can continue to connect via `ssh`, which is outside the scope of these release notes.

2. Run the following command to allow access to port 8850:

   ```
   sudo ufw allow 8850
   ```

3. Run the following command to allow access to port 80:

   ```
   sudo ufw allow 80
   ```

OpenID fails on first sign-in attempt

If you have configured Open ID Connect authentication for Tableau Server, the first sign-in attempt fails. To successfully log in, users must retry authentication after the initial failure.

Administrative views do not display

The Status tab of Tableau Server includes links to visualizations that display server metrics. These visualizations require the PostgreSQL driver to access the appropriate data from the Tableau Server repository. The PostgreSQL driver is not installed automatically, so if you have not installed the driver, the views will not display. For more information, see Database Drivers.

> **Note:** To use administrative views, the PostgreSQL driver must be installed on any node that runs the VizQL Server process.

Changing locale on view

When you change your user locale after opening a view, any subsequent attempt to open the view will fail with an "unexpected error." You can still open views that you have not previously opened.

To work around this issue, sign out of Tableau Server after changing your locale, and then sign back in. All views will display properly.

Work with Log Files

Tableau Server creates log files as a normal part of its activities. You may need to use the server log files when you are troubleshooting issues with Tableau Server or if Tableau Support requests logs from you to help you resolve an issue.

You can create a zipped log file archive using the `tsm maintenance ziplogs` command. The zipped archive contains copies of the logs you can unzip and look at, or send to Tableau Support. Once you have a copy of the archive, you can delete the archive from your server. For more information on log file archives, see Log File Snapshots (Archive Logs).

This collection of topics provides information about how to create log file archives, the contents of specific log files, and details about when and how you might want to look at a log.

Contents of Tableau Server Logs

Every Tableau Server process writes information about what it is doing to its own log file. Singly these give detailed information on the actions of each process. Taken together these log files contain detailed information about internal communication between components of Tableau Server while processing users' requests or performing automated tasks. Tableau Server logs only contain technical information useful for troubleshooting; the status of different components, actions taken by different processes, communication attempts, queries to the database (not including results), and timings of requests, for example.

Log files could contain some specific data such as names of database servers, as well as their IP addresses and ports, names or IP addresses of Tableau Server computers, and URLs and names of the workbooks and views accessed by users.

Log files do not contain any sensitive customer data such as passwords, results of the queries, or data shown on the views.

Note: When logging at the `DEBUG` level, full environment information is gathered when Tableau starts. This means that if you have any sensitive information in an environment variable, it may be included in a log. Logging at the default `INFO` level only gathers safe environment information.

The `tsm maintenance ziplogs` command allows users to not only generate a zipped archive of log files, but also to include Tableau Server repository data if the `-d` option is specified. The repository contains metadata from Tableau Server (for example, usernames, groups, projects, permissions on Tableau Server, extract refresh schedules). The repository also includes layout and connection information for the workbooks, but does not have any data such as passwords, actual data from the database or data shown on the view.

Data displayed in views comes from extract files or databases, and is cached in memory. It is not saved in logs or, in the case of live connections, in separate files on Tableau Server computers. Extract files are stored on Tableau Server computers as `.hyper` files in the `dataengine` folder, but are never included in the zipped log archive.

Investigating Tableau Server Issues

The range and complexity of possible issues with Tableau Server means that there is no simple process you can use to investigate all problems, but a general approach would include these steps:

1. **Clean up** existing log files to reduce their size. For more information, see Remove Unneeded Files.

   Important: If there is a chance you will want to get help from Tableau Support troubleshooting an issue, be sure to create a zipped archive of your logs before cleaning them up. The clean up can delete important information Support may need. For details on creating log archives, see Log File Snapshots (Archive Logs).

2. **Set the appropriate logging level**. This is something that Tableau Support will instruct you on. For more information including impact of different log levels, see Change Logging Levels.
3. **Reproduce the issue** you are troubleshooting so the logs capture the events related to the problem.
4. **Create an archive** of the logs. For more information see Log File Snapshots (Archive Logs).

   > **Important:** Use this archive when looking at the log files. You should not edit, move or delete any files directly on the server.

5. **Review the TSM Administration Controller log** (`/tabadmincontroller/tabadmincontroller_node<n>-<n>.log`) to understand any configuration or deployment done by TSM from the command line, Web UI, or API, including jobs started by TSM. Start with the controller log. This is where you'll get most useful information.

   Note: The `tsm.log` is less wordy than the `tabadmincontroller_*.log` but can provide useful, complimentary troubleshooting information.

6. **Review the Apache logs** (`/httpd/access.####_##_##_##_##_##.log` and `/httpd/error.log`) for requests that may be related to the issue you are investigating.

   The Apache logs will contain a fair amount of "noise" that does not apply to issues you are experiencing.

   - If you find a request that seems to be related to your issue, search the`vizqlserver` directory for entries that include the unique request ID from the Apache logs.
   - Look for the response code and message associated with the request ID.
   - Search for the name of the workbook, view, dashboard, or data source that is related to your issue. Make sure to look for a relevant timestamp.

- If you find a request that seems to be related to your issue, look at the response code associated with the request. (200s are good, 500s indicate problems.)
- Locate the unique request ID associated with the request you've identified (the unique request ID is a 24 character alphanumeric string at the very end of the request).

7. **Review the log archive** further to search for other messages and possible errors.

   - Use the request ID from the Apache logs to search the `vizqlserver` folder of the log archive for files containing related log entries. Look for indications of a problem (for example, error messages or long-running queries).
   - The free, open source tool, Logshark can be a useful option for reviewing log archives. For more information, see Troubleshooting Tableau Server in the Tableau Blueprint.

8. **Review script logging**.

   Tableau Server includes logs for most of the bash scripts that are included in the scripts directory at `/opt/tableau/tableau_server-/packages/scripts.<version_code>/`. These logs are saved to the `/var/tmp` directory each time a script is run.

   By default: `/var/opt/tableau/tableau_server/data/tabsvc/logs/`

9. **Contact support**

   If you are not able to solve the issue yourself, or if requested by Tableau Support, send the zipped archive to Tableau.

Tableau Server Logs and Log File Locations

Tableau Server generates log files as a normal part of its functioning. Each service that runs as part of Tableau Server generates its own logs. These log files include information about what is happening on the server, what the service or process is doing, and what, if any errors or warnings are generated. The extent of information in the logs depends on which service is writing the logs, what the logging levels are set to, and what is happening on the server.

Looking for Tableau Server on Windows? See Server Log File Locations.

Log files can be useful in helping to identify and fix issues that Tableau Server is having. In some cases, system administrators may be able to look at logs and find clues to what is happening, but in most situations the Tableau Server logs are most useful for Tableau Support. When you open a case with Support, you may be asked to send log files from your server.

Note: The specific directories and logs generated by Tableau Server depend on the version of server you are running, and which processes you have configured. New services and processes are added periodically to support new functionality. For details about processes or services you might find logs for, see Tableau Server Processes.

Tableau Server log files on an active cluster

As a best practice you should not edit or delete log files in an active Tableau Server installation. Doing this can cause unexpected behavior or server downtime. Most Tableau Server logs are written to a location in the data directory. Some logs are written to other locations.

The easiest and safest way to gather and view server log files is to create a log archive, which is a zipped collection of logs from all nodes in a cluster. If you think you may need old logs for any reason, for example, to compare with new logs after doing an upgrade, or to send to Tableau Support when troubleshooting a server issue, create a zip archive, and move the archive to a safe location that is not part of your Tableau Server infrastructure. For more information about log files in a log archive, see Server Log Files in a zipped archive.

Logs can take up a good deal of space, especially on a heavily used server. You can use the tsm maintenance cleanup command to remove logs you no longer want or need. but if you think you may need your existing logs, consider archiving them before cleanup.

Primary log locations on a working Tableau Server installation

Most of the Tableau Server logs are written to the data directory, `/var/-opt/tableau/tableau_server/data/tabsvc/logs/`. Subdirectories are created for each instance of a service, with a name that includes the service name and the version code. For example:

`/var/opt/tableau/tableau_server/data/tabsvc/logs/backgrounder`

Configuration file locations on a working Tableau Server installation

In addition to logs for each service or process,, a `config` subdirectory contains configuration information about the service.

`/var/opt/tableau/tableau_server/data/tabsvc/config/backgrounder`

Tableau Support may ask you to gather some of these if you are working with them on a server issue. The contents can be analyzed by Support.

Logs that are not written in the primary location

A few logs are not part of the main set of logs, and are written to locations other than the normal log directories:

- The TSM log. The `tsm.log` file is located in `<home dir>/.tableau/tsm`.
- The install log. The `app-install.log` file is located in `/var/-opt/tableau/tableau_server/logs`.
- The upgrade log. The `app-upgrade.log` file is located in `/var/-opt/tableau/tableau_server/logs`.
- Bash script logs. Most Tableau Server bash scripts located in the /scripts directory (`/opt/tableau/tableau_server/packages/scripts.<version>`) generate their own logs. These are written to the `/var/tmp` directory each time a script is run.

Server Log Files in a zipped archive

You may want to look at Tableau Server log files, or need to send them to Tableau Support if you have a problem with your server. Use the `tsm maintenance ziplogs` command to create a zipped archive of log files from all nodes in your installation. By default, Tableau

Server log file archives are gathered in a zip file called `logs.zip`, but you can specify a different file name when you create the archive. You can copy the archive from the server to a local computer and open it there, or send it to Tableau Support.

When you unzip the archive, a directory is created for each node in the cluster, and in that directory are sub-directories for each service or process using this naming convention:

```
<service_name>_<instance>.<version>.<build>
```

If there are multiple instances of a service on a node, there will be multiple directories for that service, one for each instance. For example, if you have two Backgrounders on a node, you will see directories like these:

```
backgrounder_0.<version>.<build>
backgrounder_1.<version>.<build>
```

The specific directories and logs in the zip file depend on what version of Tableau Server you have, and which processes you have configured. For details about processes or services you might find logs for, see Tableau Server Processes.

Log File Snapshots (Archive Logs)

Tableau Server includes functionality to generate a snapshot of log files for archival purposes. If you plan to clean up and delete old log files as part of regular server maintenance, you may consider archiving log files to an off-server storage location before deleting them.

Or, if you are working with Tableau Support on a case, the support engineer may request a server log file snapshot.

This topic describes:

- How to generate a log file snapshot
- How to send the snapshot directly to Tableau Support from the Tableau Server administration tools
- How to download the snapshot
- How to delete archived logs

Use the TSM web interface

1. Open TSM in a browser:

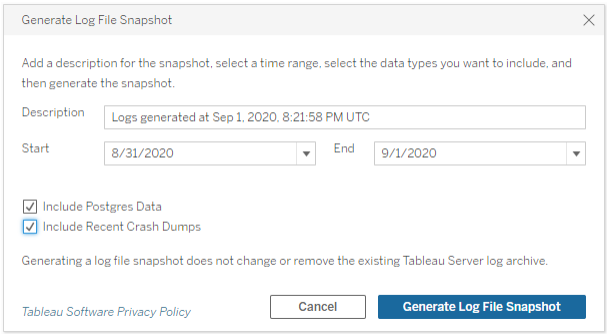
   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Maintenance**.

3. Generate a log file snapshot.

   a. On the Server Maintenance page, under Log Files, click **Generate Log File Snapshot**.

      An options dialog displays:



   b. On the Options page, enter or select the options you want, including a **Description**, time **Range** of log files to be included, and the optional types of logs to be included (**Include Postgres Data**, **Include Recent Crash Dumps**), then click **Generate Log File Snapshot**.

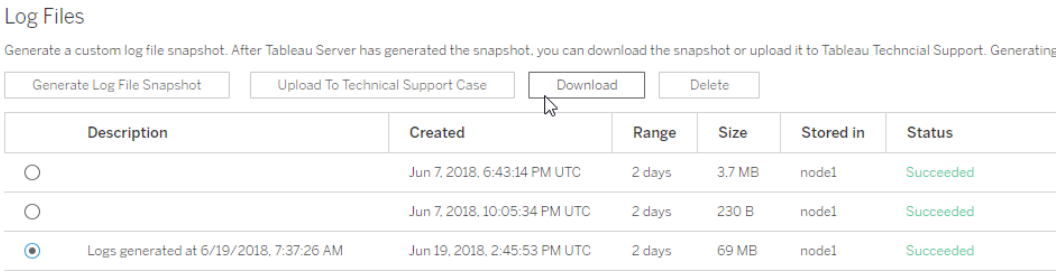
The log file snapshot is saved to a fixed location on the computer where TSM and Tableau Server are installed. If you have a multi-node installation, the snapshot is saved to the initial node of the cluster. The location is specified by the `basefilepath_log_archives` variable.

By default the snapshot is saved to:

`/var/opt/tableau/tableau_server/data/tabsvc/files/log-archives`

You can find the current location by querying the `basefilepath.log_archives` setting, and change the location by specifying a new value for `basefilepath.log_archive`. For more information, see tsm File Paths.

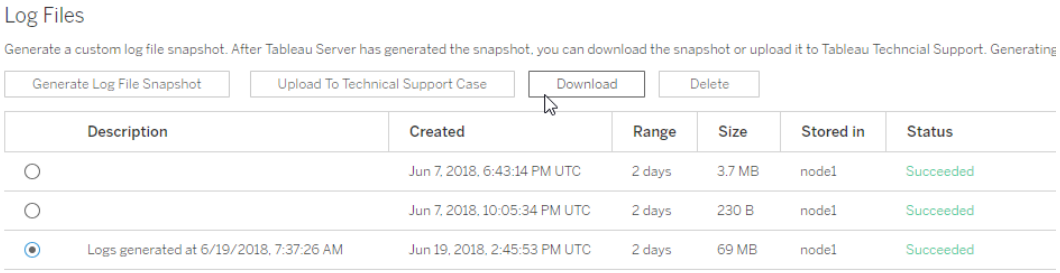
4. After you generate the snapshot, you can select it and upload it to Technical Support, download it to your local machine, or delete it:



5. To download a local copy of the snapshot or to delete it, select the snapshot under Log Files and then select the appropriate **Action**.

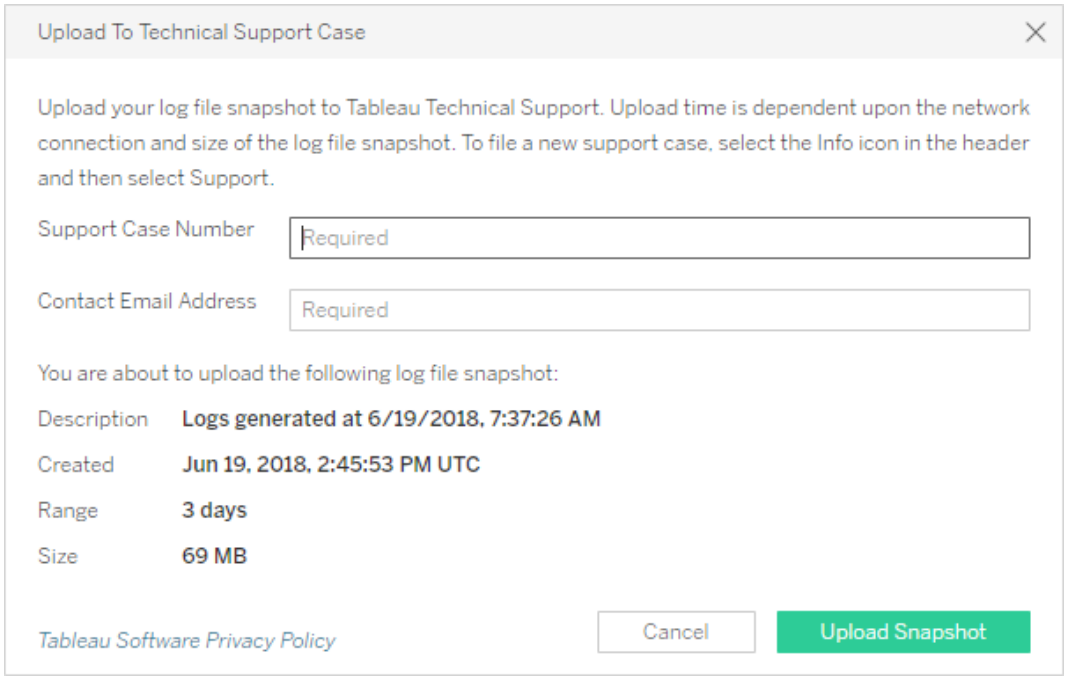
Uploading log snapshots for Tableau Support

1. Click the **Maintenance** tab.
2. Select the snapshot you want to send.



3. Click **Upload to Technical Support Case**.
4. In the dialog that displays, enter the **Support Case Number** and your **Contact Email**

Address, then click **Upload Snapshot**.



For additional methods to send log archives to Tableau Support, see Sending Large Files to Tableau in Salesforce Help.

Use the TSM CLI

You create a snapshot archive of Tableau Server log files using the `tsm maintenance ziplogs` command.

By default, this command creates a zip file containing all of the log files. If you are running a distributed installation of Tableau Server, perform this step from the initial node. Logs from all nodes will be included in the zip file.

> **Note:** If you cannot run the ziplogs command successfully, you can manually zip the Tableau Server logs. For more information, see Troubleshoot Tableau Server on Linux.

To create a log file snapshot:

1. On the initial node, open a terminal session.

2. Type the following command:

   ```
   tsm maintenance ziplogs -l -f <filename>
   ```

   where `<filename>` is name of the zipped archive file you want to create. Choose a unique name with no spaces. If an existing ziplog with the same file name already exists the creation of the file will fail unless you include the `-o` option to force an overwrite, delete the existing file, or specify a different name in the command.

   You can specify a time range for the snapshot and the types of logs to include. For example, if you know when an error occurred, use the `--startdate` and `--enddate` options to capture logs from a few hours before and after the error:

   ```
   tsm maintenance ziplogs -f <filename> --startdate "<mm/dd/yyyy
   H:mm>" --enddate "<mm/dd/yyyy H:mm>"
   ```

   For more information, see tsm maintenance ziplogs.

   The log file snapshot is saved to a fixed location on the computer where TSM and Tableau Server are installed. If you have a multi-node installation, the snapshot is saved to the initial node of the cluster. The location is specified by the `basefilepath_log_archives` variable.

   By default the log file snapshot is saved to:

   ```
   /var/opt/tableau/tableau_server/data/tabsvc/files/log-archives
   ```

   You can find the current location by querying the `basefilepath.log_archives` setting:

   ```
   tsm configuration get -k basefilepath.log_archive
   ```

   and change the location by specifying a new value for `basefilepath.log_archive`:

```
tsm configuration set -k basefilepath.log_archive -v
"<drive>:/new/directory/path"
```

For more information, see tsm File Paths.

Sending log archives to Tableau Support

You can send log files to Tableau Support as a part of a customer support case (a customer support case number is required). Before sending a log file, use `tsm maintenance zip-logs` command to combine the log files into a single zip file archive.

- In a terminal session, type the following command:

  ```
  tsm maintenance send-logs -f <zip file name> -c <case number> -
  e <email address>
  ```

  where `<case number>` is your support case number, `<email address>` is your contact email for this support case, and `<zip file name>` is the file name of your archive with .zip file extension.

For additional methods to send log archives to Tableau Support, see Sending Large Files to Tableau in Salesforce Help.

Change Logging Levels

By default, Tableau Services Manager (TSM) and Tableau Server log events at the **Info** level. You can change this if you need to gather more information (if you are working with Tableau Support, for example).

As a best practice you should not increase logging levels except when troubleshooting an issue, as instructed by Support. You should only set a logging level to debug when investigating a specific issue. Changing log levels can have these impacts:

- Increasing the log level to `debug` or `trace` increases the amount of information being logged and can have a significant impact to performance. Reproduce the issue and then reset the logging level back to info.
- Setting the log level to `warn` or `error` can reduce the amount of information so much that it is not useful for Tableau Support.

> **Note:** When logging at the `DEBUG` level, full environment information is gathered when Tableau starts. This means that if you have any sensitive information in an environment variable, it may be included in a log. Logging at the default `INFO` level only gathers safe environment information.

Logging Levels

The following logging levels are listed in order of increasing amount of information logged:

- off
- fatal
- error
- warn
- info (the default)
- debug
- trace

Change Logging Levels

Set logging levels for TSM and Tableau Server processes using **tsm configuration set** configuration keys. The key you use depends on which component of TSM or Tableau Server you want to change the logging level for.

Dynamic log level configuration

In version 2020.2 we introduced dynamic configuration. The capability has been expanded in subsequent releases. If you are only changing logging levels for one or more of these components, and are running the appropriate version of Tableau, you can change the logging levels without restarting Tableau Server.

These logging levels are dynamically configurable, beginning with these versions:

- 2020.2 - tsm services (`tsm.log.level`) and control application services (`tsm.-controllerapp.log.level`).
- 2020.3 - backgrounder (`backgrounder.log.level`), cluster controller (`cluster-controller.log.level`), data server (`dataserver.log.level`), file store

(`filestore.log.level`), data source properties (`tdsservice.log.level`), and VizQL server (`vizqlserver.log.level`).
- 2020.4 - adds interactive microservice container (`tomcatcontainer.log.level`) and application server (`vizportal.log.level`).

Configuration Keys for Changing Logging Levels

This table includes both dynamically configurable keys and those that are not dynamically configurable.

Configuration key	**Location of affected logs** (path begins with `/var/-opt/tableau/tableau_server-/data/tabsvc/logs/`)
`tsm.log.level` Changes TSM logging levels for: cli-entfileservice, licenseservice, tabad-minagent, tabadmincontroller, tabsvc	`/<service>/<service>_node<n>-<instance>.log` example: `/cli-entfileservice/clientservice_node1-0.log`

`tsm.controlapp.log.level`
 	
Changes TSM logging levels for: control applications
	`/<service>/control_<service>_node<n>-<instance>.log`
	
	examples: `/cli-entfileservice/control_cli-entservice_node1-0.log`
	
	`/filestore/control_filestore_node1-0.log`

`<process>.native_api.-log.level`	`/vizqlserver/*.txt`

Valid process names are backgrounder, vizportal, vizqlserver, dataserver **Note:** These are not dynamically configurable.	
`backgrounder.log.level` Changes logging levels for: Backgrounder	`/backgrounder/*.log`
`clustercontroller.log.level` Changes logging levels for: Cluster Controller	`/clustercontroller/*.log`
`dataserver.log.level` Changes logging levels for: Data Server	`/dataserver/*.log`
`filestore.log.level` Changes logging levels for: File Store	`/filestore/*.log`
`gateway.log.level` Changes logging levels for: Gateway control processes	`/gateway/*.log`
`gateway.httpd.loglevel` **Note:** added in version 2021.3.0 Changes logging levels for: Gateway	`/gateway/*.log`
`hyper.log.level` Changes logging levels for: Hyper	`/hyper/*.log`

`tdsservice.log.level` Changes logging levels for: Data Source Properties service	`/tdsservice/*.log`
`tomcatcontainer.log.level` Changes logging levels for microservices in: Interactive Microservice Container and Non-Interactive Microservice Container	`/tomcatcontainer/*.log`
`vizportal.log.level` Changes logging levels for: Application Server	`/vizportal/*.log`
`vizqlserver.log.level` Changes logging levels for: VizQL Server	`/vizqlserver/*.log`

For more information, see tsm configuration set Options.

If you are only changing dynamically configurable logging levels, you do not need to stop or start the server (for more information, see Dynamic log level configuration above). If you are changing other logging levels, you may need to stop Tableau Server before changing the logging levels, and restart it afterward. If this is the case, you will be prompted.

On a multi-node installation of Tableau Server, set logging levels from the initial node.

To change the logging level:

1. (Optional for dynamically configurable logging levels in 2020.2.0 and later) Stop Tableau Server by opening a command prompt and typing:

   ```
   tsm stop
   ```

2. Set the logging level to by typing `tsm configuration set -k <config.key> -v <config_value>`

   where `<config.key>` is one of the keys in the above table and `<config_value>` is a valid logging level.

   Examples:

   - `tsm configuration set -k backgrounder.native_api.log.level -v debug`
   - `tsm configuration set -k tsm.log.level -v debug`
   - `tsm configuration set -k tsm.controlapp.log.level -v debug`

3. Apply pending changes by running the `tsm pending-changes apply` command.

4. (Optional, only if server is stopped) Start Tableau Server by running the following command:

   ```
   tsm start
   ```

Reset Logging Levels

After you reproduce the issue and gather the information related to the issue, reset the logging levels so there is no lingering performance impact and no additional disk space used up.

Reset the logging level back to its default (info) using the appropriate command with a `-d` option. You need to apply pending changes after resetting the level, and if you are resetting logging levels for Tableau Server processes, you may need to stop the server before making the change, and start it after applying the pending changes.

Examples:

- `tsm configuration set -k backgrounder.native_api.log.level -d`
- `tsm configuration set -k tsm.log.level -d`

Troubleshoot Tableau Server Install and Upgrade

Follow the suggestions in this topic to resolve common issues with Tableau Server. For additional troubleshooting steps based on process status viewed on the Status page, see Troubleshoot Server Processes.

General Troubleshooting Steps

Many Tableau Server issues can be addressed with some basic steps:

1. Make sure there is enough disk space on each computer running Tableau Server. Limited disk space can cause a failure to install, a failure to upgrade, or problems running Tableau Server.
2. Restart Tableau Server. Issues related to processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau Server, use the `tsm restart` command. This will stop all the processes associated with Tableau Server and then restart them.
3. Reindex Tableau Server. Issues related to indexing can be resolved by reindexing Tableau Server. To reindex Tableau Server, use the `tsm maintenance reindex-search` command. For more information, see Reindexing Tableau Server Search & Browse below.
4. Restart the computer on which Tableau Server is running. Some issues, such as those related to data source connectivity, can be resolved by restarting the server computer.

Common Tableau Server Install Issues

Installation logs location

The install log, `app-install.log`, is located in `/var/opt/tableau/tableau_server-/logs`.

The upgrade log, `app-upgrade.log`, is located in `/var/opt/tableau/tableau_server/logs`.

Multiple install attempts fail

If you attempt to install Tableau Server and the install fails, any subsequent installation attempts are likely to fail unless you run the `tableau-server-obliterate` script to clean Tableau off the computer.

A failed install attempt can leave the computer in a state that causes subsequent attempts to also fail with errors that don't seem directly related to a previous install attempt. One possible error is:

```
Enabling and starting all services
+ services=(appzookeeper* tabadmincontroller* tabsvc* licenseser-
vice* fnplicenseservice* tabadminagent* clientfileservice*)
+ systemctl_user enable appzookeeper_0.service 'tabadmincontroller*'
'tabsvc*' 'licenseservice*' fnplicenseservice_0.service 'tabad-
minagent*' 'clientfileservice*'
++ id -ru a_tabadminpoc
+ local unprivileged_uid=222954
+ su -l a_tabadminpoc -c 'XDG_RUNTIME_DIR=/run/user/222954 systemctl
--user enable appzookeeper_0.service tabadmincontroller* tabsvc*
licenseservice* fnplicenseservice_0.service tabadminagent* cli-
entfileservice*'
Failed to execute operation: No such file or directory
```

To fix this problem, run the `tableau-server-obliterate` script to clean up any left over remnants of the previous install attempt and then restart the computer. For more information, see Running the tableau-server-obliterate script .

Important: If you created a backup of Tableau (`<file>.tsbak`) you want to keep (for example, to restore to your new installation), copy that file to a safe location on another computer to guarantee it is not removed when you clean up your Tableau computer.

Install fails due to hardware requirements

Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. The requirements apply to all computers on which you are installing Tableau Server. For details on minimum hardware requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server.

Install or upgrade fails due to CPU requirements

Beginning in version 2020.4.0 Tableau Server requires CPUs that support SSE4.2 and POPCNT instruction sets. You cannot install or upgrade Tableau Server 2020.4.0 or oater on computers that have CPUs which do not support these instruction sets.

You may see this error message when installing a new installation, or in preparation for upgrading an existing installation:

```
Your computer's processor doesn't meet the minimum requirements
that Tableau requires to install the software. If you are using a
VM, make sure Processor compatibility mode is off.
```

The SSE4.2 and POPCNT instruction sets have been common for more than 10 years and most newer CPUs support them, but if you get an error related to processor minimum requirements when attempting to install or upgrade Tableau Server on a Virtual Machine (VM), Processor compatibility mode may be enabled on the VM. To successfully install or upgrade Tableau on a VM, make sure the Processor compatibility mode is turned off.

Common Tableau Server Upgrade Issues

Upgrade logs location

The upgrade log, `app-upgrade.log`, is located in `/var/opt/tableau/tableau_server/logs`.

Maps do not display or display incompletely after upgrading

Beginning with Tableau version 2019.2, the internet access requirements changed for maps. If you are upgrading from version 2019.1.x or earlier to version 2019.2.x or later, and maps

are not displaying as expected, confirm that your environment is configured to allow access on port 443 to `mapsconfig.tableau.com` and `api.mapbox.com`.

In version 2019.1.x or earlier, access was necessary to `maps.tableausoftware.com`.

For more details on internet access requirements, see Communicating with the Internet.

Upgrade script error: "Tableau Server Version change validation failed."

When upgrading, if you run the `upgrade-tsm` script from the `scripts.<version_code>` directory for the earlier version, the upgrade will fail with an error:

```
Tableau Server Version change validation failed.
Tableau Server <version> is already installed.
```

If you get this error, change to the `scripts.<version_code>` directory for the version you just installed and run the script from there.

Upgrade multi-node, initializing additional node fails with "Enter your credentials again" error

If you attempt to initialize an additional node when upgrading Tableau Server and see this error:

```
Enter your credentials again. The credentials you enter must provide
administrative access to the computer where you generated the con-
figuration file.
```

this is an indication that the node is unable to connect to or communicate with the initial node. This can happen for multiple reasons:

- The credentials you entered are not valid or you mistyped them. The credentials must be for a user who has administrative permissions on the computer where Tableau Server was first installed. You do not need to use the credentials of the user who created the bootstrap file but doing so will ensure you are using valid credentials.
- The local firewall of the computer you are trying to add is not allowing communication to the initial node. For more information, see Local firewall configuration.

Upgrading fails due to lack of disk space

If there is not enough disk space for the Tableau Server Setup program to run and do the upgrade, the installation will fail. The amount of disk space required will depend on the size of your repository database and the number and size of your extracts.

To free up disk space:

1. Create a log archive snapshot using the `tsm maintenance ziplogs` command.

   After you create the ziplogs file, save it to a safe location that is not part of your Tableau Server installation.

2. Clean up unnecessary files using the `tsm maintenance cleanup` command. For more information, see Remove Unneeded Files.

Upgrade fails on RebuildSearchIndex job

Beginning with version 2020.1.x, the final step in an upgrade is to rebuild the search index. At this point all services have been upgraded, so if this job fails, you can manually reset the search server by running the `tsm maintenance reset-searchserver` command. You do not need to obliterate and start over.

The error will be:

```
An error occurred while rebuilding search index.
```

To reset the search server :

1. On the initial node, open a terminal session.

   This must be a new terminal session because the upgrade script updates system environment for the new version.

2. Rebuild the search index using the tsm maintenance reset-searchserver command.

Upgrade fails on 2022.1 and later

After upgrading Tableau Server 2022.1 (or later), restoring a Tableau Server backup as part of your upgrade process can cause the following error:

"The backup cannot be restored because Tableau Server uses the new identity service tables by default."

This issue occurs because Tableau Server 2022.1 (and later) uses an identity schema that is different from the identity schema used by the backup. To resolve this issue, see Troubleshoot Issues with the Identity Migration.

Upgrade fails on 2020.4.0 or later

Beginning with version 2020.4.0, the Checkpoint Upgrade feature allows you to retry a failed upgrade. In general, this is most useful for experienced server administrators and IT professionals who are comfortable with Tableau Server log files and are willing to search through them. But the feature can help in all failed upgrades because it allows you to rerun the upgrade-tsm script, and the script is run from the last successful step, saving time. For those with experience, it may be possible to identify problems like disk space problems, or permissions issues, correct them, and rerun the upgrade.

If you are upgrading to version 2020.4.0 or later and the upgrade fails, the following steps may help you to complete the upgrade:

- Rerun the `upgrade-tsm` script. Upgrade failures are sometimes a result of timeouts during the upgrade process, and rerunning the script can allow the upgrade to get beyond intermittent or occasional timing issues. This is also a step that is safe to do, and easy. Rerunning the script will do no harm, and at worst, the upgrade will fail again at the same point, but without needing to go through any previous steps.

  The script is located in the `\scripts` directory:

  `opt/tableau/tableau_server/packages/scripts.<version_code>/upgrade-tsm`

If your Tableau Server upgrade isn't successful when you rerun the upgrade-tsm script, and you are comfortable with Tableau Server logs, you can take these additional troubleshooting steps:

- Look at the output of the script in the command window. Useful error messages may help you identify the cause of the upgrade failure and give you some ideas for how to correct the issue.
- Look in the `app-upgrade.log` file. Any errors that are displayed at the command line will also appear in the `app-upgrade.log` file, often with more details.
- Look in the `tabadmincontroller.log` file. Upgrade problems that aren't easily identifiable in the above two instances are likely the result of an issue in a job. The `tabadmincontroller.log` file may have more information that helps you diagnose the issue.

  Note: For information about log file locations, see Tableau Server Logs and Log File Locations.

Upgrade fails due to permission problems with the backup/restore file location

With versions of Tableau Server before 2022.1.0, If the file location for the backup/restore file does not have the correct permissions, the upgrade script will fail with an error about not being able to read the backup file or not being able to restore the repository.

Beginning with version 2022.1, the upgrade script confirms the permissions of file location for the backup/restore file before starting the upgrade so the file can be written to and read from the location during the upgrade to the new version of Tableau Server.

The errors will be similar to these:

```
The tableau user does not have permission to read the backup file:
<backup/restore basefilepath>.

Repository restore failed.
```

```
An error occurred during installation.
An error occurred while restoring repository.
```

The location used by TSM for backup and restore is defined by the `base-filepath.backuprestore` configuration key and has a default that the installation program sets up with correct permissions, but these may be impacted by organization IT rules or if you change the location to one you have created yourself. A new command available starting in 2022.1 allows you to check the permissions on the backup/restore file location immediately after creating it, to avoid any permission-related problems. For details about that command, see tsm maintenance validate-backup-basefilepath.

For details about the backup/restore file path, see tsm File Paths.

Upgrade succeeds but published data sources cannot be accessed

In limited, specific scenarios, after upgrading Tableau Server from version 2021.3 to early versions of 2023.1 or 2023.3, attempts to connect to or refresh existing published data sources fail with this error:

```
java.io.FileNotFoundException: Unable to fetch data from any other host. This may indicate a lost or invalid folder.
```

This could happen if:

1. You upgrade a Tableau Server installation that was version 2021.3.x at any point (you could be running 2021.3 or have upgraded from 2021.3 to a 2022.x version)

   and

2. You upgrade that installation *to* early versions of 2023.1 or 2023.3

No impact

There are no problems in the following situations:

- In all other upgrade paths *from* 2021.3
- In all other upgrade paths *to* 2023.1 or 2023.3

- In all fresh installations of 2023.1 and 2023.3

More information

As of September 16, 2024, all problematic versions have been removed from the download site. If you need to upgrade to version 2023.1.x or 2023.3.x, upgrade to maintenance release versions 2023.1.16 or higher, or 2023.3.9 or higher.

For more information about this issue, see the Known Issue.

Common Settings Import Issues

Import of settings file causes "not present on any node" validation error due to missing services

If you are upgrading by installing a new version of Tableau Server and importing a settings file from an earlier version, you may encounter topology validation errors when running the `tsm settings import` command.

This can happen when you export a settings file from an older version of Tableau Server and import it into a new version, and new services have been added to Tableau between the two versions.

Errors will be similar to this (the specific service may be different):

```
>tsm settings import -f 20183-export.json

Pending topology set.
There are 1 topology validation errors/warnings.

Service 'elasticserver' is not present on any node in the cluster.
Service: Elastic Server
```

To resolve this issue, add any missing services to Tableau Server:

1. For any service that generated a validation error, add the service with an instance count of 1.

   For example, if the Elastic Server is not present in the cluster, set the process instance count to 1 using the service name that appears in the first line of the validation error message:

   ```
   tsm topology set-process -n node1 -pr elasticserver -c 1
   ```

   Repeat this step for each service that results in an error.

2. When you have no more warnings or errors, apply the pending changes:

   ```
   tsm pending-changes apply
   ```

Your settings should be imported successfully.

Import of settings file causes "configuration value you specified does not match" error

If you are installing a new version of Tableau Server and import a settings file from an earlier version, you may encounter configuration validation errors when running the `tsm settings import` command. These can occur when a settings file includes a configuration value that has since been removed from Tableau.

The error will look similar to this (the configuration key may be different):

```
>tsm settings import -f 20183-export.json
Configuration error: At least one configuration value you specified
does not match a known configuration key. This applies to the fol-
lowing keys: '[features.TsmConfigFileService]'
Use this parameter to override unknown key error: --force-keys
```

To resolve this issue, edit the settings file you are importing to remove the reference to the configuration key or keys in the error:

1. Copy the JSON settings file and save the copy for backup.

2. Open the JSON settings file in a plain text editor.

3. Locate and delete the entire line that includes the key. In this example, `features.TsmConfigFileService`:

```
"configKeys" : {
  "config.version" : 19,
  "tabadmincontroller.port" : "8850",
  "endpoints.enabled" : false,
  "endpoints.health.enabled" : true,
  "features.TsmConfigFileService" : true,
  "tableau_projects.language" : "en",
```

   The above is an example of a small section of an exported settings file and is not intended to represent the entire contents of the file.

4. Save the settings file and import it again.

You may encounter additional errors related to topology validation. For information about solving those errors, see Import of settings file causes "not present on any node" validation error due to missing services above.

"You cannot directly modify instances of the Coordination Service" error

This error can occur in two situations:

- When you import a Tableau Server settings file into an installation that has a different Coordination Service topology than the settings file does
- When you attempt to configure the Coordination Service using the `tsm topology set-process` command

If you see this error after importing a settings file:

The Tableau Server settings file has a different Coordination Service topology than the target server does. This can happen if you are upgrading Tableau Server by installing a new version and importing a settings file from an earlier version. If you have not explicitly deployed a Coordination Service ensemble on the target server, it has a single instance of Coordination Service, on the initial node.

To correct this error you can take either correct the mismatch from the command line, or by editing the settings import file. You can also discard all pending changes, deploy the Coordination Service on the target computer to match the settings in the import file, and reimport the settings file.

To correct the mismatch from the command line, for each node that generates an error, use the tsm topology set-process command to revert the instance count of Coordination Service.

1. Run the `tsm pending-changes list` command. The output shows you which nodes have changes.

2. Find the node or nodes where the Coordination Service count is changed.

   For example, if the settings file had a Coordination Service instance on node2, but the target system did not have any Coordination Service instance on that node, the count for node 2 would show as changed from 0 to 1 by the import of the settings file:

   ```
   C:\Windows\system32>tsm pending-changes list
   Configuration
   There are no pending configuration changes.
   Topology
   node2:
               Coordination Service
                                   New Instance Count:1
                                   Old Instance Count:0
   ```

3. Use the `tsm topology set-process` command to set the count back to the "Old Instance" value.

   For the example above:

   ```
   tsm topology set-process -n node2 -c 0 -pr "Coordination Service"
   ```

4. Once you have reset any Coordination Service instance count that was changed, apply

pending changes:

```
tsm pending-changes apply
```

If you see the error when setting the process count for Coordination Service manually:

This error can also occur if you attempt to update the Coordination Service directly, using the `tsm topology set-process` command instead of the `tsm topology` commands for managing the Coordination Service. If you tried this:

1. Use the `tsm pending-changes discard` command to discard the pending changes.
2. Use the correct commands for configuring the Coordination Service. For more information, see Deploy a Coordination Service Ensemble .

Starting Tableau Server

Tableau Server cannot determine if it fully started

In some instances Tableau Server may report that it could not determine if all components started properly on startup. A message displays: "Unable to determine if all components of the service started properly."

If you see this message after starting, verify that Tableau Server is running as expected by using a `tsm status -v` command.

If the status shows as running ("Status: RUNNING"), then the server successfully started and you can ignore the message. If the status is DEGRADED or STOPPED, see "Tableau Server doesn't start" in the next section.

Tableau Server doesn't start

If Tableau Server does not start or is running in a degraded state, run the `tsm restart` command from a command prompt. This will shut down any processes that are running, and restart Tableau Server.

Reindexing Tableau Server Search & Browse

Problems that can be solved by rebuilding Search & Browse index

Symptoms of an index that needs to be rebuilt include:

- A blank list of sites when a user attempts to log in
- A blank list of projects when a user tries to select a project
- Missing content (workbooks, views, dashboards)
- Unexpected or inaccurate alerts (for example, an "refresh failed" alert on a workbook that does not include an extract)

If you see any of these behaviors, reset and rebuild the Search & Browse index using the `tsm maintenance reset-searchserver` command.

Activating Tableau Server

Tableau Server license activation fails

In some instances Tableau Server license activation may fail. Error messages can range from a very generic one:

- `An error has occurred`

To more specific messages:

- `Function flxActCommonLicSpcPopulateFromTS returned error 50030, 71521,`
- `No license found for 'Tableau Server'`

To resolve this issue, try these solutions in the order listed:

Confirm you can access the licensing server

The Tableau licensing service was moved to a new data center on October 6, 2018. This means any environments that required special configuration (static IP safe listing for example) to access licensing.tableau.com or licensing.tableau.com will need to be updated before you can activate, refresh, or deactivate a Tableau product key.

To test access, type the URL and the port of the licensing server in a browser:

```
https://licensing.tableau.com:443
```

and:

```
https://atr.licensing.tableau.com/_status/healthz
```

If you are able to access the server, a "Test success" message displays for the first server, and an "OK" message displays for the second.

Tableau Server needs to make a connection to the following internet locations for licensing purposes:

- atr.licensing.tableau.com:443
- licensing.tableau.com:443
- register.tableau.com:443
- o.ss2.us
- s.ss2.us
- crt.rootca1.amazontrust.com
- crt.sca1b.amazontrust.com
- crt.sca0a.amazontrust.com
- crt.sca1a.amazontrust.com
- crt.sca2a.amazontrust.com
- crt.sca3a.amazontrust.com
- crt.sca4a.amazontrust.com
- *.digicert.com
- ocsp.*.amazontrust.com

- crl.*.amazontrust.com
- crt.rootg2.amazontrust.com

Requests to the above domains may be on port 80 or 443. Port 80 is used for certificate validation (revocation, certificate chain, etc). Port 443 is used for SSL connections.

Requests to the `ocsp.*.amazontrust.com` and `crl.*.amazontrust.com` domains are managed by Amazon for certificate revocation information. See ACM certificate characteristics for more information. We recommend that you install the Amazon root certificates in the certificate trust store on the computer running Tableau. To download and install the Amazon root certificates, see Certificate Authorities on the Amazon Trust Services web site.

Verify the date and time

Verify the date and time on the initial Tableau Server computer is correct. If the clock is set to a time and date earlier than the current date, Tableau Server cannot be activated.

Force the product key to be read again

1. On the initial Tableau Server computer, log on as a user with sudo access.
2. Change to the Tableau Server bin directory. By default this is:

   ```
   /opt/tableau/tableau_server/packages/bin.<version_code>/
   ```

3. Type the following commands:

   ```
   tsm stop

   ./lmreread
   tsm start
   ```

Send the contents of trusted storage to Tableau Support

If FlexNet Licensing Services is installed and running but you're still seeing an error, there might be a problem with the Tableau product key information. To resolve this issue, complete the following steps to create a file of the key information located in trusted storage.

1. On the initial Tableau Server computer, log on as a user with sudo access.
2. Type the following command:

   ```
   serveractutil -view > <machine_name>-LicResults.txt
   ```

   This creates the `<machine_name>-LicResults.txt` file in your current directory. If you don't have write permissions for that location and see an error, change to a location where you do have permission to create a file and run the command again.
3. Contact Tableau Support (http://www.tableau.com/support/request) and include the `<machine_name>-LicResults.txt` file that you created.

tabcmd Installation Problems

Installing tabcmd separately

tabcmd is automatically installed on the initial Tableau Server node when you install Tableau Server, but if you want to run it on another computer, you need to download and install tabcmd separately. For details, see Install tabcmd.

Problems installing tabcmd on Linux

tabcmd requires Java 11 to run properly. On RHEL-like systems, this will be installed as a dependency when installing tabcmd. On Debian-like systems, you need to install Java 11 separately if it is not already installed.

As of July 2022, Debian distributions are no longer supported. For more information, see this Tableau Community post.

Java is not installed

If you see errors similar to this when installing tabcmd, confirm that Java 11 is installed on your Linux computer:

```
Cannot find 'java' in your PATH. Install 'java' and make sure it is
in your PATH to continue.
```

Incorrect version of Java is installed

If you see errors similar to these, confirm that Java 11 is installed:

```
Exception in thread "main" java.lang.UnsupportedClassVersionError:
com/tableausoftware/tabcmd/Tabcmd : Unsupported major.minor version
52.0
```

or.

```
*** Uncaught exception NoClassDefFoundError: javax/xm-
l/bind/JAXBException
*** See the logs for the stacktrace.
```

systemd User Service Failures

You may receive one of the following errors when upgrading or when running `initialize-tsm` during a fresh installation:

- "Failed to get D-Bus connection: No such file or directory"
- "$XDG_RUNTIME_DIR not found"
- "systemd unit user@<userID> is not running. Check /var/log/messages or /var/-log/syslog."

Background

As of 2018.1, Tableau Server uses the `systemd` user service to manage processes. This means there is a `systemd` process that runs as the unprivileged user. By default, Tableau Server Setup creates an unprivileged account named `tableau`. The Tableau Server processes are spawned from the `systemd` process and not the system-wide `systemd` process, which runs as root.

> **Important**: This troubleshooting note applies primarily to RHEL 7-based distros. However, if you see one of these errors, it's possible that the same issues exist on Ubuntu distros.

The `systemd` user service is not used as commonly as the normal `systemd` process manager. Red Hat disabled the `systemd` user service in RHEL 7 (and thereby all distros that come from RHEL, like CentOS, Oracle Linux 7, Amazon Linux 2). However, RedHat has assured Tableau that running the `systemd` user service is supported as long as the service is re-enabled.

Upgrading from Tableau Server on Linux 10.5

If you are upgrading from Tableau Server 10.5, check that the unprivileged user has a valid shell and home directory. For Tableau Server 10.5, Tableau deliberately created the unprivileged user with shell set to `/sbin/nologin` and home directory "`/`". If the unprivileged user was created by `initialize-tsm`, then during upgrade to 2018.1 Tableau updates the shell and home directory.

However, if you created the unprivileged user during the initial installation of 10.5, then you will get an error when trying to upgrade.

To fix this, you must set the shell to `/sbin/nologin` and the home directory "`/`", and then run upgrade again.

Fresh installation error troubleshooting

Verify that the `systemd` user service is running.

Check by running the command, `ps -fww $(pgrep -f "systemd --user")`

If the `systemd` user service is not running, then something prevented it from starting.

Follow this list to troubleshoot:

- Check the logs in `/var/log/messages`
- Run `journalctl`
- Verify that any customization that you may have done to your PAM configuration includes has not removed `pam_systemd.so`.

If the RHEL 7 PAM file, `/etc/pam.d/system-auth` is missing the following line:

```
-session optional pam_systemd.so
```

then it must be added back for Tableau Server to function.

- If `-session optional pam_systemd.so` is present in your PAM configuration, the user service cannot start, and the error message `$XDG_RUNTIME_DIR not found` is showing in `/var/log/messages` , do not attempt to set the environmental variable. In this scenario, the error is not accurate.

  The real error is that the PAM module `pam_systemd.so` is unable to allocate the user session. The default configuration suppresses error messages from `pam_systemd.so`. To surface error messages and debug messages, change the line in `/etc/pam.d/system-auth` from `-session optional pam_systemd.so` to `session optional pam_systemd.so debug`. (Removing the leading hyphen will surface the error messages, and adding `debug` will surface more verbose logging.)

  Now you can look in `/var/log/messages`, `/var/log/secure` and `/var/log/audit/audit.log` files to see error messages.

Example

You may see the following error message:

```
systemd-logind: Failed to mount per-user tmpfs directory /run/user/0: Permission denied
```

In this case, searching the error online leads to the Redhat KB article, https://access.redhat.com/solutions/2460611.

The article recommends updating the `selinux-policy` package by running `sudo yum update selinux-policy`.

In some cases, upgrading from version 3.12.X to 3.13.X fixes a `$XDG_RUNTIME_DIR not found` problem. Be sure to run `sudo reboot` after updating the package.

Troubleshoot Job Failures Due to Service Failures

Beginning in Tableau Server version 2021.1, a new set of error messages help you understand when jobs fail due to an issue with a service. This topic explains the messages and what they can mean.

An error message will display in this format:

```
<nodeId>
<service>_<instanceId>.<version>: <error>
```

There are four categories of errors:

- **Missing status** - If a service is unable to report its status, for example, if a node is down, if tabadminagent is unable to report status, or if a service has failed in a way that results in its dependent services not being able to be installed, this will show up as a "missing status" error.
- **Failed to update configuration** - If a service is unable to update its configuration file, this shows up as a "failure to update configuration" error. This may occur during upgrades, when attempting to apply pending changes to new services. See the service's control app log, `<dataDir>\tabsvc\logs\<service>\control-<service>_<nodeId>-<instanceId>` for more details about the error.
- **Failed to reach the requested state. Current state:<currentState>** - If a service cannot be installed/removed/started/stopped, this results in a "failed to reach requested state" error. This can happen during upgrades when new services are being installed and old ones removed. Possible options for <currentState> are: DEPLOY_FAILED, INSTALL_FAILED, DISABLE_FAILED, ENABLE_FAILED, CONFIGURE_FAILED, UNINSTALL_FAILED, REMOVE_FAILED. Most common are: INSTALL_FAILED, UNINSTALL_FAILED and REMOVE_FAILED. See the service's control app log, `<dataDir>\tabsvc\logs\<service>\control-<service>_<nodeId>-<instanceId>` for more details about the error.
- **Failed to start/stop. Current status: <currentStatus>** - This occurs if a service that should be running is stopped or if a service that should be stopped continues to run. Possible values in <currentStatus> are: ACTIVE, BUSY, PASSIVE, UNLICENSED, DOWN, STATUS_UNAVAILABLE and DEGRADED. The first three (ACTIVE, BUSY, PASSIVE) are considered "running" statuses. The last four are a "stopped" status. See

the main service log, `<dataDir>\tabsvc\logs\<service>\<service>_<nodeId>-<instanceId>.log`, for more details about the error.

An example of an error message is:

```
This job failed due to unexpected error: 'Ser-
viceOperationTimeoutException'
One or more services failed to reach their expected state.
node1:
     vizportal_0.2021.4.0.0: Failed to reach requested state. Cur-
rent state: INSTALL_FAILED
```

Beginning in version 2021.3, an additional error message has been added for the second and third error types that will match the errors found in the control app logs.

Troubleshoot Server Sign in Problems

There are several different sign in options between Tableau Services Manager (TSM) and Tableau Server.

- **TSM**—If you are not able to sign into TSM, make sure you are using credentials for a user who has administrative rights to the computer where TSM is installed. This user may or may not also be a Tableau Server administrator. This is true whether you are signing in to the Web UI or the CLI. For more information, see Sign in to Tableau Services Manager Web UI.
- **Tableau Server**—
    - Administrators: if you are signing into Tableau Server as an administrator, you must use credentials for a user who has an administrator role in Tableau Server. You create the initial administrator when you first install Tableau, but can add other users as administrators once Tableau is installed and running. For more information, see Sign in to the Tableau Server Admin Area.

- Non-administrative users: If you are signing into Tableau Server as a user, you need to be use credentials for a user who has been added to Tableau Server. For more information, see Sign in to Tableau Server or Tableau Cloud.

  Note: If users with valid credentials are unable to sign into Tableau Server, make sure you have not added a node without applying pending changes. If you have a pending new node, signing into Tableau Server may not be possible.

Troubleshooting scenarios

Troubleshoot Licensing

This topic includes instructions for troubleshooting issues related to Tableau Server licensing.

Handle an unlicensed server

Tableau offers two licensing models: role-based and core-based. To learn more about role-based and core-based licensing, see Licensing Overview.

role-based licensing requires each active user account to be covered by a license. role-based licenses have a defined capacity, or number of users that they allow. Each user is assigned a unique user name on the server and is required to identify themselves when connecting to the server.

Core-based licensing has no constraints on the number of user accounts in the system, but it does restrict the maximum number of processor cores that Tableau Server can use. You can install Tableau Server on one or more machines to create a cluster, with the restriction that the total number of cores in all the machines does not exceed the number of cores you have licensed and that all of the cores on a particular machine are covered by the license.

Unlicensed role-based server

The most common reason for a server that has role-based licensing to be unlicensed is an expired product key or an expired maintenance contract.

Unlicensed core-based server

A core-based server can become unlicensed for a variety of reasons, such as an expired product key or when Tableau Server nodes running licensed processes cannot contact the Tableau Server node running the License Manager service. To learn more about licensed processes, see Tableau Server Processes.

When the server is unlicensed you may not be able to start or administer the server. You can, however, manage your licenses using the tsm licenses command.

Unlicensed server administrator

All Tableau Server administrators require a user license. Tableau Server administrators will always consume the highest role available. If a Creator product key is activated, the Tableau Server Administrator(s) will take this role. If the highest role available on Tableau Server is an Explorer, the Server Administrator will take the Explorer role. If Creator licenses are added to the server, any existing Server Administrator accounts using Explorer licenses will automatically convert to use Creator licenses.

TSM administrator accounts do not require licenses.

If the license that the server administrator is using expires, then the account will become unlicensed and will be unable to sign in.

Verify the expiration date of your license(s) for the administrators on the server:

- Run tsm licenses list.
- Compare the date with the date displayed in the Tableau Customer Portal.
- If the portal does not display the date that you expect, contact Customer Success.
- To renew your license, visit the Tableau renewal web page.
- Run the tsm licenses activate command to activate a new license for the administrator account(s).

If the TSM date matches the portal date and the following refresh operation fails, contact Tableau Support.

If the license for your administrator account has expired or will expire soon, you will need to activate a new license for the account. Alternatively, you can unlicense a non-administrator user to free a license for the server administrator account.

If a Tableau Server administrator is using a Creator, Explorer or Viewer license and their license expires, they will use another license of the same type, if available. If no license seats are available the user will become “unlicensed”.

Important: Do not restart Tableau Server until you have activated a new license or transferred a site role for the server administrator account.

Troubleshoot role-based licensing

This section provides information about resolving issues that can occur when adding the role-based Viewer, Explorer and Creator licenses to Tableau Server or Tableau Cloud, or when these licenses expire. The highest available license type is Creator, followed by Explorer, and finally Viewer. To learn more about role-based licensing, see Licensing Overview.

A user or administrator is unlicensed due to license expiration

To avoid having users unexpectedly become unlicensed or move to another site role, you should always do one of the following before the license that they are currently using expires:

- Renew and activate a replacement license. If a user occupies a Creator, Explorer or viewer license and their license expires, they will use another license of the same type, if available.
- Change the site role of those users to allow the use of a license that is not due to expire.

To learn how site roles can be changed to require a different license, see Set Users’ Site Roles.

The reassignment of users to new licenses is governed by the following logic:

- When a Server Administrator user occupies a Creator license and their license expires (with no replacement licenses available), they are reassigned to an Explorer license if any Explorer licenses are available. This license reassignment occurs in order of most recent login. Server Administrators displace other users who might be currently using an Explorer license. If no Creator or Explorer licenses are available a Server Administrator becomes unlicensed.
- When a non-Server Administrator user occupies a Creator license and their license expires (with no replacement licenses available), they become unlicensed. To avoid having these users become unlicensed, change their site role prior to license expiration. This is especially important for users in the Site Administrator Creator site role, who must move to the Site Administrator Explorer site role before their Creator license expires to avoid losing Site Administrator capabilities.
- When a non-Server Administrator user occupies an Explorer or Viewer license and their license expires (with no replacement licenses available), they are upgraded to a higher license type, if licenses of that type are available. Specifically, the following occurs when a license expires:
    - Users who occupy an Explorer license will move to a Creator license, if available (with no change to site role).
    - Users who occupy a Viewer license will move to an Explorer license, if available. If no Explorer licenses are available, these users will move to a Creator license, if available (with no change to site role).
    - If no licenses are available at the higher license types, those users are moved to Unlicensed.

Users are reassigned to a new license as described above in order of most recent login, with lower license types reassigned first (first Viewer, then Explorer, and then Creator).

For example: Two users with a Viewer license, a user with the Creator license, and two Server Administrators with a Creator license all have their licenses expire. Four unexpired Explorer licenses are available for these users. In this situation, the following occurs in the order shown below:

1. The user with a Viewer license who logged in most recently is reassigned to an Explorer license.
2. The second user with a Viewer license is reassigned to an Explorer license.

3. The Server Administrator user with a Creator license who logged in most recently is reassigned to an Explorer license, and then the second Server Administrator with a Creator license is reassigned to the remaining Explorer license.
4. The user with the Creator license becomes unlicensed.

Server Administrator site role is unchanged when using a Creator license

Server Administrators gain Creator capabilities if Creator licenses are available in Tableau Server, with no change to their site role name. All other Tableau Server and Tableau Cloud users gain Creator licenses only if assigned to a site role that includes Creator in its name.

Licenses are not immediately available

When you add a role-based license to Tableau Server, those licenses become available to all users when you restart Tableau Server.

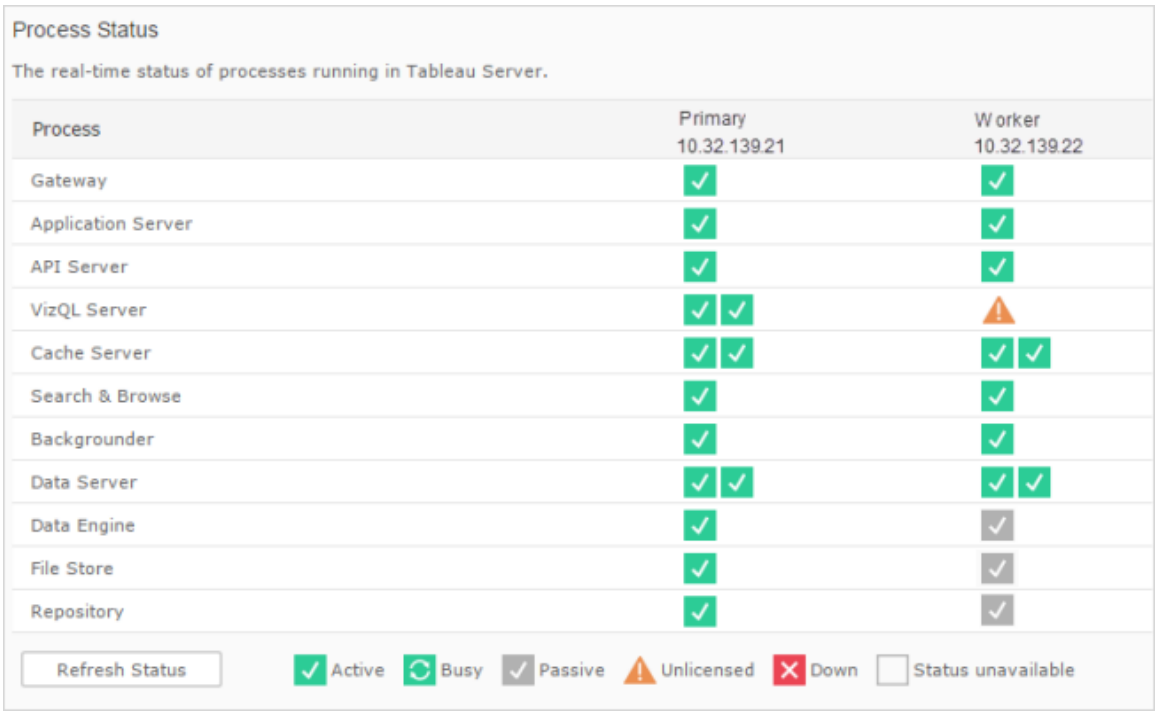
A user with a Viewer license cannot open Tableau Server or Tableau Cloud workbooks from Tableau Desktop

A user with a Viewer license who also has a separate Tableau Desktop license will be unable to open workbooks on Tableau Server or Tableau Cloud using Tableau Desktop. To open workbooks such using Tableau Desktop, that user will need an Explorer or Creator license on Tableau Server or Tableau Cloud.

Handle an Unlicensed Server Process

There are several status indicators on the Tableau Server Status page that help you understand the state of Tableau Server processes. An orange-color status box, "Unlicensed", indicates that one of the server processes is unable to retrieve the Tableau Server license information.

In the image below, one of the VizQL processes is unlicensed:



There may be several reasons why a process is unable to access licensing information. For example, there may be network issues preventing a process running on an additional node from communicating with the licensing service on the initial node. Or, the unlicensed process may be getting sent more requests than it can accept at a particular moment and can't handle the licensing request. The impact to users depends on which process is unable to confirm its license, and whether there are other instances of the process on one of the server nodes. In the case of the unlicensed VizQL process above, some users may be able to access views while others cannot.

To resolve the problem, stop, then start Tableau Server.

Tableau Services Manager (TSM) Command Timeout

When Tableau Server is configured with two instances of the repository and failover to the backup repository occurs, TSM attempts to restart the original repository so that it is available as a backup. If this cannot be done for any reason, subsequent TSM commands can fail due to timeouts while waiting for the original repository to recover.

Commands that can be impacted include:

- tsm maintenance restore
- tsm maintenance reindex-search
- tsm reset
- tsm security regenerate-internal-tokens
- tsm sites export
- tsm sites import

If any of these commands is failing, and you have a repository that is not recovering, remove the repository from the server topology, apply pending changes, and re-add it.

Troubleshooting Tableau Services Manager (TSM) Backup

Backup fails to start because services do not start

When you back up Tableau Server, one of the first steps taken is to confirm that key services are running, and, if they are not, to start them. If these services cannot be started:

- Active Repository
- File Store
- Cluster Controller

any attempt to back up Tableau Server will fail with one of the following errors:

```
An error occurred starting one or more of the following services:
Active Repository, File Store, Cluster Controller.
```

```
One or more of the following services did not start in a timely
fashion: Active Repository, File Store, Cluster Controller.
```

To successfully back Tableau Server up, make sure these processes can start.

Cookie Restriction Error

When a user signs in to Tableau Server, a session cookie is stored in their local browser. The stored cookie is how Tableau Server maintains that the signed in user has been authenticated and can access the server. Because the cookie is set with the same domain or sub-domain as

the browser's address bar, it is considered a first-party cookie. If a user's browser is configured to block first-party cookies, they will be unable to sign in to Tableau Server.

When a user signs in to Tableau Server via an embedded view, or in an environment where trusted authentication has been configured, the same thing happens: a cookie is stored. In this case, however, the browser treats the cookie as a third-party cookie. This is because the cookie is set with a domain that's different from the one shown in the browser's address bar. If a user's web browser is set to block third-party cookies, authentication to Tableau Server will fail. To prevent this from occurring, web browsers must be configured to allow third-party cookies.

Troubleshoot Subscriptions

"The view snapshot in this email could not be properly rendered."

If you receive a subscription with this error message, there could be several reasons:

- **Missing credentials**: Some views are published with embedded credentials. You may receive the above error if the embedded credentials are now out-of-date, or if the view was republished without the embedded credentials.
- **Database temporarily down**: If the view has a live database connection and the database was temporarily down when the subscription was being generated, you might receive the above error.
- **Background process timeout**: By default, the background process that handles subscriptions has a timeout value of 30 minutes per view for the rendering of a view. If rendering a view goes beyond this time limit, the next view in the workbook results in a failed job due to the timeout. In the majority of cases, this default is plenty of time. However, if the background process is handling an extraordinarily large and complex dashboard, that may not be enough time. You can check the Background Tasks for Non Extracts admin view to see if that's the case. To increase the timeout threshold, use the `tsm configuration set subscriptions.timeout` command.

Can't see images in email

For images of content to display in a subscription email, users subscribed to views, in addition to **View** permissions, must also have **Download Image/PDF** permissions. For more information, see Permissions.

Can't subscribe

If you can see a view on Tableau Server and it has a subscription icon () in the upper right corner, you can subscribe to it.

To subscribe to a view, Tableau Server needs to be correctly configured (described in Manage Subscriptions) and the view you're subscribing to must either have embedded credentials for its data source or not rely on credentials at all. Examples of the latter include a workbook that connects to an extract that isn't being refreshed, or a workbook whose data is in a file that was included with the workbook at publish time. Embedding credentials is a step that happens in Tableau Desktop (see the Tableau Help for details).

No subscription icon

It's possible to see a view but be unable to subscribe to it. This can happen becausefor several reasons:

- **No subscriptions have been scheduled**: If no subscriptions have been scheduled or all subscription schedules are disabled, the subscription icon will not appear. To set a schedule for subscriptions, see Create or Modify a Schedule.
- **The view uses a live database connection**: Views with live database connections, where you're prompted for your database credentials when you first click the view, aren't available for subscription. A subscription includes a view (or workbook), data, and a schedule. To deliver the data required for the view, Tableau Server either needs embedded database credentials or data that doesn't require credentials. Where live database connections are concerned, Tableau Server doesn't have the credentials, only the individual users do. This is why you can only subscribe to views that either don't require credentials or have them embedded.
- **Tableau Server is configured for trusted authentication**: You may also be able to see a view but be unable to subscribe to it (no subscription icon) if Tableau Server is

configured for trusted authentication. See Ensure Access to Subscriptions for more information.

Receiving invalid or "broken" subscriptions

If you configured subscriptions on test or development instances of Tableau Server in addition to your in-production instance, disable subscriptions on your non-production instances. Keeping subscriptions enabled on all instances can result in your users receiving subscriptions that appear to be valid, but which don't work, or receiving subscriptions even though they've unsubscribed from the view or workbook.

Missing attachments

You can add a PDF attachment to your subscription if your administrator has it enabled. If the PDF attachment is missing from your subscription, it might be because the size of the PDF exceeds either the email server size limit or the maximum size limit set by server administrators. In Tableau Server, the maximum size limit for PDF attachments to subscriptions can be adjusted through the tsm configuration option `subscriptions.max_attachment_size_megabytes`. For more information, see Configure Server Event Notification and Set Up a Site for Subscriptions.

Starting in Tableau 2024.1, you can send emails with your own sending server, which will allow you to send attachments with a maximum email size of 10MB.

To enable this feature, navigate to site settings, find the **Customize Email Notifications** section, and check the box next to **Use your SMTP server**.

> Note: If you're using your own sending server, Tableau will attempt to send a subscription email with a subset of the attachment that is under the 10MB limit, but this is not guaranteed. If Tableau can't send the attachment, you'll see a message letting you know that the attachment is too large to send.

Suspended Subscriptions

By default, a subscription is suspended after 5 consecutive subscription failures. To change the threshold number of subscription failures that can occur before they are suspended, use the tsm configuration set option, `backgrounder.subscription_failure_threshold_for_run_prevention`. This sets the threshold for the number of consecutive failed subscriptions necessary before suspending the subscription. This is a server-wide setting.

Only Server administrators can configure the threshold number of subscription failures before a subscription is suspended. For information on setting this threshold, see Set up a Server for Subscriptions.

By default, administrators are not emailed when a subscription is suspended, but can opt-in to suspension emails per site through **My Account Settings**.

Resume suspended subscriptions

Administrators and subscription owners can resume subscriptions in several ways:

- from My Subscription tab in Content Settings
- from the Subscriptions tab per workbook
- from the Subscriptions tab under Tasks (Server Admins only)

When a subscription is resumed, the alert failing count goes back to zero. The next evaluation of the subscription will occur at the next scheduled evaluation time.

Can't set subscription frequency to "When Data Refreshes"

You can set subscriptions to run when an extract refreshes if the workbook uses a connection to a published extract. When creating or modifying a subscription, you might not see a **Frequency** option if the workbook uses:

- More than one extract refresh
- A live data connection

Subscriptions not arriving ("Error sending email. Can't send command to SMTP host.")

You may see the above error in Windows Event Viewer if subscriptions aren't arriving and your SMTP server is using encrypted (TLS) sessions. To send subscriptions to an SMTP server that is configured with TLS, you must configure secure SMTP on Tableau Server. See Configure SMTP Setup. (If you're experiencing this error, note that Tableau Server will still indicate that subscriptions are being sent in the Background Tasks for Non Extracts admin view.)

Missing data quality warnings or sensitivity labels

Data quality warnings and sensitivity labels are included in subscription emails when:

- Tableau Server or Tableau Cloud is licensed with Data Management. For more information, see About Data Management.
- Tableau Catalog is enabled. For more information, see Enable Tableau Catalog.
- In site settings, the check box under **High-Visibility Data Labels in View and Workbook Subscriptions** is selected. (In earlier versions, the check box is under **Data Quality Warnings in Subscriptions**.)

Server Administrator Reference

You can learn more about Tableau Server processes, ports, and accounts and permissions.

Tableau Server Processes

This topic describes the options for setting the process configuration. To configure Tableau Server processes, you need to specify which processes and how many instances should run on each node. You do this using the tsm topology set-process command. For more information, see Changing the number of processes on a node.

> Looking for Tableau Server on Windows? See Tableau Server Processes.

Except where explicitly noted in the table below, applying changes in processes will stop Tableau Server if it is running when you apply those changes. After changes are applied, Tableau Server is returned to the state it was in before process configuration, so if the server was running, it will be restarted.

Important: Your process topology will depend on your organizational needs.

Licensed processes

Some of the processes that are installed as a part of Tableau Server are "licensed" processes. Licensed processes need a valid Tableau Server license in order to run. Other processes that are installed as a part of Tableau Server are not tied to a valid license. This has the following impact:

- Every licensed process needs to regularly contact the Tableau Server License Manager service that runs on the initial Tableau Server computer to verify they are licensed. If they cannot confirm there is a valid license, for example, if the initial node is not available, the process will not run and Tableau Server may not function properly or reliably.
- If you have a core-based Tableau Server license, the cores on any node with a licensed process will count against the total count of licensed cores.

  Note: If you have Data Management and a core-based license, you will need to understand how the licensed processes will count against the total count of licensed cores that come with each license. For more information, see License Data Management.

The "Licensed" column in the table below identifies those processes that require a valid license, and which impact the count of cores in core-based licenses.

Tableau Server Processes	These processes have a status of `running` when Tableau

Server is running, and `stopped` when Tableau Server is stopped.				
Name shown in `tsm status -v`	**Name used with `tsm topology set-process`**	**Purpose**	**Notes**	**Licensed**
Analytics Extensions Microservice	`analytics-icsextensions`	The Analytics Extensions Microservice supports a set of functions to pass expressions to analytics extensions for integration with R, Python, and Einstein Discovery.	Automatically installed on any node where Application Server (VizPortal) is installed.	No
Application Server	`vizportal`	The Application Server (VizPortal) handles the web application, REST API calls, and supports browsing and searching.	When Application Server is installed, Data Engine is also installed, unless the node already has an instance of Data Engine. When the first instance of Application Server is installed on a node, the Interactive Container Service is also installed.	Yes
Ask Data	Cannot be con-	The Ask Data	Runs automatically on all	No

	figured manually.	service is used by the Ask Data feature.	nodes where Data Server is running.	
Authentication Added: 2022.1	Cannot be configured manually.	The Authentication service handles the identity migration process and identity pools. After the identity migration process is complete and as part of identity pools management, the Authentication service is responsible for the following: • Searching for a user identity in the appropriate	Status for this process is only available through tsm CLI. Identity migration must be complete and the Identity Service turned on to set up and user identity pools. For more information, see About the Identity Migration and Provision and Authenticate Users Using Identity Pools.	No

		Identity Service table, using the returned universal unique identifier (UUID) to query the legacy system_ user table and find the appropriate system user, and then granting them a user session, which com-		

		• Importing immutable user identifiers and additional user attributes when the configured identity store is external (AD or LDAP).		
Backgrounder	`backgrounder`	The Backgrounder runs server tasks, including extract refreshes, subscriptions, 'Run Now' tasks, and tasks initiated from tabcmd.	When Backgrounder is installed, Data Engine is also installed, unless the node already has an instance of Data Engine. Backgrounder is a single-threaded process. You can add more instances of backgrounder to a node to	Yes

			expand the capacity of the node to run jobs in parallel. In most situations you can increase or decrease the number of backgrounder instances on an existing node of a running server without causing a stop and restart the server. The exception is when you have Data Management licensed and you add the first instance of backgrounder to a node or remove the last instance of backgrounder from the node. For more information, see Tableau Server Dynamic Topology Changes.	
Cache Server	`cacheserver`	The Cache Server is a query cache distributed and shared across the server cluster. This in-	The cache is single-threaded, so if you need better performance you should run additional instances of cache server. We recommend you	No

		memory cache speeds user experience across many scenarios. VizQL server, backgrounder, and data server (and application server to a lesser extent) make cache requests to the cache server on behalf of users or jobs.	install no more than a maximum of six cache server instances, limiting each node to no more than two. Our testing indicates that installing more than a total of six cache server instances in a Tableau Server installation provides no improvement, and could have a negative impact on performance.	
Cluster Controller	`cluster-controller`	The Cluster Controller is responsible for monitoring various components, detecting failures, and running failover when needed.	Automatically installed on every node.	No
Collections Added: 2021.2.0	`collections`	The Collections service provides meta data for	The Collections service is installed on the first node where the Application (vizportal) is	No

		the collections and favorites features.	installed. For high availability, we recommend you install an instance of the Collections service on every node that has an instance of the Application Server installed. For more information, see Tableau Server Collections Service.	
Connection Pooling Added: 2023.1.0		The Connection Pooling service is intended to offer an option for pooling database connections to the repository.	The Connection Pooling service is installed on the every node in a disabled state. It cannot be enabled except in TSM. It should **not** be enabled except by instructions from Tableau Support.	No
Content Exploration Added: 2021.1.0	`content-texploration`	The Content Exploration service extends search and browse capabilities for Tableau Server. It also depends on Index and Search Server	The Content Exploration service is installed on the initial node. For high availability, we recommend you install an instance of the Content Exploration service on every node that has Application Server installed. For more	No

		to do its operations.	information, see Tableau Server Content Exploration Service.	
Data Engine	Cannot be configured manually.	The Data Engine creates data extracts and processes queries.	Automatically installed when you install File Store, VizQL Server, Application Server (VizPortal), Data Server, Prep Flow Authoring, or Backgrounder. **Note:** When File Store is configured externally, Data Engine is no longer installed with File Store. For more information see Tableau Server External File Store.	Yes
Data Profiling Added: 2021.4.0	`dataprofiling`	The Data Profiling service handles column profiling requests for the Virtual Connection editor.	Only installed when you have a Data Management license. Then, automatically installed on any node where Backgrounder is installed.	Yes - Requires a Data Management license
Data Server	`dataserver`	The Data Server manages connections to Tableau Server data	When Data Server is installed, Data Engine is also installed, unless the node already has an instance of Data Engine.	Yes

		sources.		
Data Source Properties Added: 2020.1.0	`tdsservice`	The Data Source Properties service provides published data source metadata to client services like Ask Data.	Data Source Properties is added by default on the initial node. A Tableau Server installation must include at least one instance. For performance reasons, we recommend installing the Data Source Properties service on any node that has Application Server (VizPortal) installed.	No
Data Stories Added: 2023.1.0		The Data Stories service handles the analytical engine that powers the Data Stories dashboard extension.	Data Stories is automatically added to every node of the Tableau Server installation. It cannot be configured manually.	No
Elastic Server Added: 2019.1 Removed: 2022.1 Starting in	`elasticserver`	Elastic Server is used by Ask Data to index data and by Content Exploration service to index searchable content.	Elastic Server processes can be running on more than one node in a cluster. Optionally, they can be moved to any node. It is recommended to have an odd number of Elastic Server processes running.	No

version 2022.1 Index and Search Server is used instead.			The Elastic Server heap size can be configured by using the `elasticserver.vmopts` TSM configuration option. For more information, see tsm configuration set Options.	
Extract Service Added: 2021.4.0 Retired: 2023.3.0. In this and later versions, Virtual Connections are handled by Backgrounder.	`extract-service`	The Extract service manages extracts of Virtual Connections.	Only installed when you have a Data Management license. Then, automatically installed on any node where Backgrounder is installed.	Yes - Requires a Data Management license
File Store	`filestore`	File Store can be configured to run locally on Tableau	When File Store is installed, Data Engine is also installed, unless the node already has an	Local File Store: No External

		Server or externally using SAN or NAS storage. When configured locally: The File Store automatically replicates extracts across Data Engine nodes.	instance of Data Engine or if File Store is configured externally.	File Store: Requires Advanced Management license.
Gateway	`gateway`	The Gateway is a web server that handles all requests to Tableau Server from browsers, Tableau Desktop, and other clients.	Required on any node with an instance of VizQL Server, Vizportal, or Tableau Prep Flow Authoring.	No
Index and Search Server Added: 2022.1	`index-andsearch-server`	Index and Search Server is based on AWS OpenSearch. Tableau uses the search capability of	Index and Search Server can be configured on more than one node in a cluster. We recommended an odd number of Index and	No

		Open Search to index data for Ask Data and the Content Exploration service. This server process is the replacement for Elastic Server which is no longer used in version 2022.1 and later. As of version 2023.3.0, this process replaces Search and Browse.	Search Server processes running in total. On Tableau Server Clusters that have 3 or more nodes, we recommend that you configure Index and Search Server on at least three different nodes. The Index and Search Server heap size can be configured by using the `tsm set configuration` command with the `indexandsearchserver.vmopts` option. For more information, see tsm configuration set Options.	
Internal Data Source Properties Added : 2020.1.0	`tdsnativeservice` Cannot be configured manually.	The Internal Data Source Properties service is an internal service that only communicates with the Data Source Prop-	One instance of Internal Data Source Properties is automatically configured on each node that has an instance of Data Source Properties on it.	No

		erties service.		
Messaging Service Added: 2019.4.0	`act-ivemqserver`	The Messaging Service is used to support communication between microservices in Tableau Server.	Automatically installed on initial node when you install Tableau Server. One instance of the service is required. On multi-node installations of Tableau Server, you can move the Messaging Service to a different node. In version 2020.1 you can add a second instance of the Messaging Service on an additional node to provide some redundancy (in 2019.4 you cannot configure more than one instance in a cluster). For more information, see Tableau Server Messaging Service.	No
Metrics Service Added: 2020.2.0 Retired: 2024.2	`metrics`	The Metrics Service is responsible for reading and writing Metric data in Tableau Server.	Automatically installed on initial node with a single instance when you install Tableau Server. One instance of the service is required. You can add additional	No

			instances as necessary . We recommend at least one instance of the Metrics service on each node in a multi-node installation of Tableau Server. For more information, see Tableau Server Metrics Service. (The legacy Metrics feature was retired in February 2024 for Tableau Cloud and in Tableau Server version 2024.2. For more information, see Create and Troubleshoot Metrics (Retired).)	
Minerva Service Added: 2021.4.0	`minerva`	The Minerva service executes queries to Virtual Connections.	Only installed when you have a Data Management license. Then, automatically installed on any node where Backgrounder is installed.	Yes - Requires a Data Management license
NonRelational Storage Service Added: 2021	`nrs`	The NonRelational Storage service is a microservice managed and used intern-	Automatically installed on the initial node of Tableau Server. This service cannot be managed by system admin-	No

3.1.0 Retired: 2024.1.0 **Note:** The service appears in the CLI in versions 2023.x but is non-functional.		ally by other Tableau services.	istrators.	
Virtual Connections Service Added: 2021.4.0	`published-connections`	The Virtual Connections service handles queries to Virtual Connections.	Only installed when you have a Data Management license. Then, automatically installed on any node where Backgrounder is installed.	Yes - Requires a Data Management license
Query Gateway Microservice Added: 2021.4.0	`querygateway`	The Query Gateway Microservice routes queries to the appropriate microservice, depending on the query type and source.	Only installed when you have a Data Management license. Then, automatically installed on any node where Backgrounder is installed.	Yes - Requires a Data Management license
Query Policy Service	`querypolicy`	The Query Policy service provides	Only installed when you have a Data Management license. Then,	Yes - Requires a Data

Added: 202-1.4.0 Retired: 2023.3.0		information about Data Policies when processing queries.	automatically installed on any node where Backgrounder is installed.	Management license
Repository	`pgsql`	The PostgreSQL repository is the main database for Tableau Server. It stores workbook and user metadata. When Tableau Catalog (or Tableau Metadata API) is enabled, the repository stores Tableau content and external assets metadata.	You are limited to a maximum of two instances of the repository in a cluster, and must have at least three nodes in the cluster to add a second repository instance.	No
Resource Limits Manager Added in: 2022.1	`"Resource Limits Manager"`	The Tableau Server Resource Limits Manager tracks back-	The Resource Limits Manager is automatically and by default installed on the Initial node of Tableau Server. We do	No Requires Advanced Man-

		grounder resource usage in relation to the set resource limits to make sure the resource limits are applied correctly.	not recommend adding more processes or configuring this on additional nodes of the Tableau Server.	agement
SAML Service	Cannot be configured manually.	The SAML Service acts as a proxy between Tableau Server and SAML Identity Providers (IdPs).	Automatically installed on each node where you install Tableau Server. Shows a status of `stopped` in output of `tsm status -v` unless site SAML is enabled. You cannot configure the SAML Service manually.	No
Search And Browse Retired: 2023.3.0	`searchserver`	The Search and Browse service handles fast search, filter, retrieval, and display of content metadata on the server.	Beginning in version 2022.3.0, Search and Browse is no longer used. The Index and Search Server completely replaces this. Beginning in version 2023.3.0, it is no longer installed. **Note:** If you are running Tableau Server version	No

			2022.3 or 2023.1, you *should not configure more than one instance of Search and Browse* for any installation. Configuring more than one instance can, in rare cases, result in stability issues.	
Tableau Prep Conductor	`flowprocessor`	The Tableau Prep Conductor runs flows and processes flows for ingestion by Data Catalog. It leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output.	By default, it is automatically enabled on a node where backgrounder is enabled. If the node role is set to exclude flows, then Tableau Prep Conductor is not installed on that node. For more information, see Workload Management through Node Roles. Starting in 2020.4 Data Management is not needed to enable this process on Tableau Server.	Yes
Tableau Prep Flow Authoring Added in	`floweditor`	Provides the interactive Prep Flow experience in the browser.	When Tableau Prep Flow Authoring (`floweditor`) is installed, Data Engine, Tableau Prep Flow Service	Yes

version 2020.4.			(`flowqueryservice`), and Gateway are also installed, unless the node already has an instance of each of those.	
Tableau Prep Minerva Service Added as Tableau Prep Flow Service in version 2020.4 and renamed in version 2021.2.	`flowminerva` Note: previously `flowqueryservice`	Used by Tableau Prep Flow Authoring (`floweditor`) for querying datasources.	By default, it is automatically enabled on a node where Tableau Prep Flow Authoring (`floweditor`) is enabled.	Yes
Tableau Statistical Service Added: 2022.1 Retired: 2023.3	`statsservice`	The Tableau Statistical Service manages the statistical engine for Explain Data and predictive modeling functions.	Automatically installed on any node where VizQL is installed. For more information, see Tableau Statistical Service.	No
VizQL Server	`vizqlserver`	The VizQL Server loads and renders	When VizQL Server is installed, Gateway and Data Engine are also	Yes

		views, computes and executes queries.	installed, unless the node already has an instance of Gateway and Data Engine. In most situations you can change the number of VizQL instances on an existing node of a running server without causing a stop and restart the server. An exception is if you are adding VizQL to an existing node that did not previously have VizQL or any other process that also installs Gateway and Data Engine. For more information, see Tableau Server Dynamic Topology Changes.	
VizData Service Added: 2024.2	`vizdata-service`	The VizData Service manages connections to published data sources on Tableau Server.	An instance of VizData Service is installed for every instance of Data Server.	Yes
VizData Native Ser-	`vizdatan-ativeservice`	The VizData	An instance of VizData Native Service is added	Yes

vice Added: 2024.2		Native Service communicates with the VizData Service on Tableau Server.	for each instance of VizData Service.	
Tableau Microservice Container Processes These processes are automatically added when the first instance of Backgrounder or Application Server is added to a node. If all instances of Backgrounder or Application Server are removed from a node, the microservice container process is also removed. Container status depends on the status of the microservices within the container. If all microservices are running, the container process has a status of `running`. If all microservices are stopped, the container process status is `error`. If one or more microservices is running while others are not, the container service has a status of `degraded`. For more information, see Tableau Server Microservice Containers.				
Interactive Microservice Container		Container process for internal Tableau Server microservices that are bundled together for ease of deployment and scalability purposes.	These containers and the microservices they contain cannot be manually configured. The microservices may change over time.	No
Non-Inter-	`non-`	Container pro-	These containers and	No

active Microservice Container	`interactive`	cess for internal Tableau Server microservices that are bundled together for ease of deployment and scalability purposes.	the microservices they contain cannot be manually configured. The microservices may change over time.	
Tableau Services Manager (TSM) Processes These processes have a status of `running` once TSM has been initialized, and remain running even when Tableau Server is stopped.				
Activation Service Added in version 2021.1	Cannot be configured manually.	The Activation Service, also known as the authorization-to-run service (ATR), enables you to activate Tableau Server without running out of licenses. It provides short-term leases of configurable duration until the product key expires.	Automatically installed on the initial node beginning in versions 2023.1.3, 2022.3.7, 2022.1.15 and later. In earlier versions, this is automatically installed on the initial node when ATR is enabled.	No

Administration Agent	Cannot be configured manually.	The TSM Agent monitors the Coordination Service for changes to configuration or topology and delivers new configurations to each service (configuration) or deploys new services and removes old ones (topology)	Automatically installed on each node where you install Tableau Server. You cannot configure the Administration Agent manually. For more details, see Tableau Server Administration Agent.	No
Administration Controller	Cannot be configured manually, except to move it to another node. For more information, see Recover from an Initial Node Failure.	The TSM Controller handles requests to TSM and orchestrates configuration and topology changes and workflow across service processes. The Controller also serves as the REST API endpoint (HTTPS).	Automatically installed when you install TSM on the initial node. You cannot configure the Administration Controller manually except to move it to another node. For more information, see Recover from an Initial Node Failure. For more details, see Tableau Server Administration Controller	No

Client File Service	`client-fileservice`	The Client File Service (CFS) manages most shared files in a multinode cluster. For example, authentication related certificates, keys, and files (OpenID, mutual SSL, SAML, and Kerberos), and customization files are managed by CFS.	Automatically installed on the initial node. No other instances are installed unless you explicitly configure them. See Configure Client File Service . In multi-node deployments, we recommend you configure an instance of CFS on each of the nodes where you deploy the Coordination Service. Redeploying the Coordination Service does not have any impact on CFS. CFS does not display in the Status page or the Configuration page but is visible in the output of the `tsm status -v` command. To view or set instances of CFS, use the tsm topology command.	No
Coordination Service	Cannot be set with `tsm topology set-process`.	The Coordination Service serves as the single source	Automatically installed on the initial node. No other instances are installed unless you expli-	No

		of truth.	citly deploy a new Coordination Service ensemble. For details, see Deploy a Coordination Service Ensemble .	
License Manager	Cannot be configured manually.	The License Manager handles licensing.	Automatically installed on the initial node when you install TSM. A single instance of this is installed on a Tableau Server cluster. The License Manager process should only be manually configured if the initial node fails. For more information, see Recover from an Initial Node Failure	No
Tableau Server Maintenance Processes These processes have a status of `stopped` unless they are actively running to complete a job.				
Database Maintenance	Cannot be configured manually.	The Database Maintenance service is responsible for performing maintenance operations on the Tableau Server repository.	Automatically installed on each node where you install Tableau Server. Shows a status of `stopped` in output of `tsm status -v` unless it is actively performing database maintenance.	No

			Maintenance can include updates related to enabling remote access to the repository and changing passwords used to access the repository. You cannot configure the Database Maintenance service manually.	
Backup/Restore	Cannot be configured manually.	The Backup and Restore service is responsible for performing backup and restore operations on the data stored in the Tableau Server repository and file store.	Automatically installed on each node where you install Tableau Server. Shows a status of `stopped` in output of `tsm status -v` unless it is performing a backup or restore operation. You cannot configure the Backup and Restore service manually.	No
Site Import/Export	Cannot be configured manually.	The Site Import and Export service is responsible for migrating Tableau Server sites between	Automatically installed on each node where you install Tableau Server. Shows a status of `stopped` in output of `tsm status -v` unless it is performing an import	No

		server clusters.	or export. You cannot configure the Site Import and Export service manually.	

Process workflow

The Tableau Server processes and how they interact depend on what action or activity is taking place. For example, the processes that are used, and how they interact, differ when you publish a workbook from those used when you sign in using SAML. For some interactive views on process workflow, see the below workbook. This allows you to select a particular workflow and follow it step by step from start to finish.

Disclaimer: This workbook is published on Tableau Public and is not maintained by Tableau documentation. We cannot guarantee that it is up-to-date with the latest version of Tableau Server.

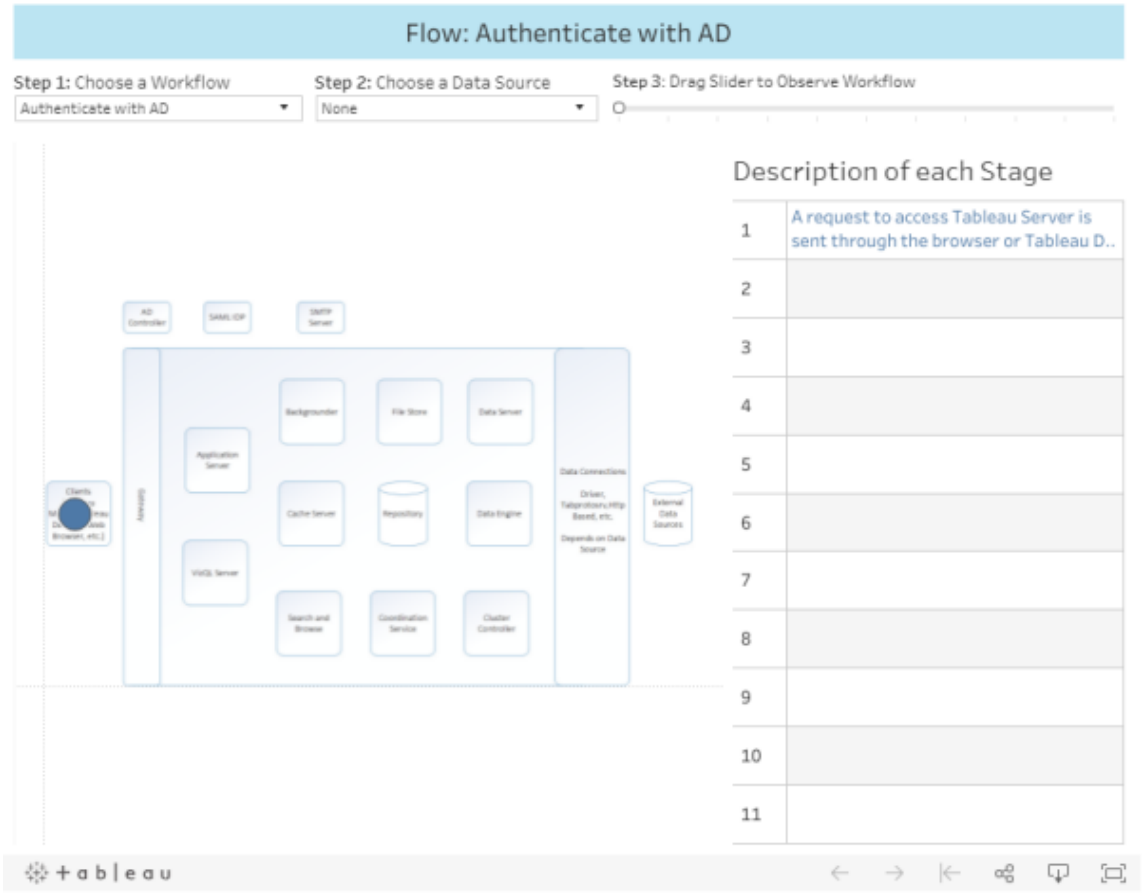


Tableau Server Administration Agent

The Administration Agent monitors the Coordination Service for changes to configuration or topology and delivers new configurations to each service (configuration) or deploys new services and removes old ones (topology). The Administration Agent also checks each of the services for status and reports this back to the Coordination Service. This process will be automatically configured on each node of the cluster during installation—no explicit configuration is required or possible.

The Administration Agent may also be referred to as the *TSM Administration Agent.*

Process	Administration Agent
Status	Status of the Administration Agent process is not visible on the Status Page.

	Use the TSM CLI to view status. For more information, see View Server Process Status
Logging	Logs generated by the the Administration Agent process are located in `/var/opt/tableau/tableau_server/data/tabsvc/logs/tabadminagent`. For more information, see Tableau Server Logs and Log File Locations

What happens when an Administration Agent process fails? All other Tableau Server processes running on the same node will display as “unavailable” on the TSM status page. Tableau Server will continue to work as expected, however you will not be able to make configuration/topology changes to the cluster. Failed Administration Agent processes automatically restart as long as the computer itself is otherwise healthy. If the Administration Agent doesn’t start up on the node, you can try to start the service manually by running the following command:

```
sudo su -l tableau

systemctl --user start tabadminagent_0
```

Tableau Server Administration Controller

The Administration Controller process hosts the TSM REST API for configuring and managing your Tableau Server deployment. There can only be a single instance of the Administration Controller in the entire cluster.

This process will be automatically configured on the initial node of the cluster during installation—no explicit configuration is required.

The Administration Controller is also referred to as the *TSM Controller*and the *TSM Administration Controller*.

Process	Administration Controller
Status	Status of the Administration Controller process is visible on the Status Page, displayed as **TSM Controller**. For more information, see View Server Process Status

Logging	Logs generated by the Administration Controller process are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/tabad-mincontroller`. For more information, see Tableau Server Logs and Log File Locations

What happens when the Administration Controller process fails?

If the Administration Controller fails, the Tableau Server cluster should continue to function; however, tsm commands and the TSM web UI will be unavailable. You will not be able to make any changes or updates to the configuration or topology until the Administration Controller is back up and running. Like other TSM services, Administration Controller is automatically restarted if it is stopped or has failed.

If the Administration Controller doesn't start up on the node, you can try to start the service manually by running the following command:

```
sudo su -l tableau

systemctl --user start tabadmincontroller_0
```

Moving the Administration Controller

If the initial node fails, you need to move the Administration Controller and the Licensing Service to a different node so that Tableau Server can continue to function. For details on how to do this, see Recover from an Initial Node Failure.

Restarting the Administration Controller

Restart the TSM Administration Controller (as *tableau* system account):

```
sudo su -l tableau -c "systemctl --user restart tabadmincontroller_
0.service"
```

Note: It may take a few minutes for tabadmincontroller to restart. If you attempt to apply pending changes in the next step before the controller has fully restarted, TSM will not be

able to connect to the controller. You can verify that the controller is running by using the `tsm status -v` command. Tableau Server Administration Controller should be listed as "is running".

Tableau Server Application Server

The Application Server (VizPortal) handles the web application and REST API calls. Application Server also supports browsing and searching. To ensure high availability of Application Server, configure instances on each node in the Tableau Server cluster.

Process	Application Server
Status	Status of the Application Server process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the Application Server process are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/vizportal`. For more information, see Tableau Server Logs and Log File Locations

What happens when an Application Server process fails? Requests being handled by that instance will fail, but subsequent requests will be routed to other running Application Server processes. Assuming the node containing the failed Application Server is still running, the failed process should automatically restart within seconds.

Troubleshooting problems with Application Server

Beginning with version 2024.2.0, admins can enable Activity and Resource Tracing (ART) data to trouble issues with Application Server. The ART data captured provides detailed insights of memory and CPU usage. This can be useful if your server is experiencing Server Resource Manager (SRM) restarts due to native memory saturation with Application Server (VizPortal).

Important: When you enable ART, additional entries are written to vizportal java and cpp logs. You should not leave ART enabled after you are finished investigating. Be sure to disable ART to avoid extra disk space usage.

Enable ART data on Tableau Server

To enable ART data:

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run the following tsm commands:
   a. `tsm configuration set -k vizportal.enable_art -v true`
   b. `tsm configuration set -k vizportal.log_art_java -v true`
   c. `tsm configuration set -k vizportal.art_skip_list -v "/v1/re-portEventU-nau-thentic-ated,/v1/-getSes-sionIn-fo,/v1/hasUn-seenNo-tific-ation-s,/v1/re-freshSes-sion,/v1/-getViews,/v1/-getUser-`

```
s,/v1/-
getWork-
book-
s,/v1/-
getView,/v1/-
getServer-
Set-
ting-
sUnauthenticated,/v1/getProjectAncestors,/v1/getWorkbook"
```

d. `tsm restart`

Application Server logs are found here by default:

```
C:\ProgramData\Tableau\Tableau Server\data\tabsvc\logs\vizportal
```

Disable ATR

Disable ART data after completing your investigation. To disable ART, run the following commands at a command prompt:

1. `tsm configuration set -k vizportal.enable_art -v false`
2. `tsm configuration set -k vizportal.log_art_java -v false`
3. `tsm configuration set -k vizportal.art_skip_list -v ""`
4. `tsm restart`

Tableau Server Backgrounder Process

The Backgrounder process runs server jobs, including extract refreshes, subscriptions, flow runs, and data driven alerts. Jobs are initiated both from scheduled tasks and when started manually using ‘Run Now’, REST API, or tabcmd commands.

Process	Backgrounder
Status	Status of the Backgrounder process is visible on the Status Page. For more information, see View Server Process Status

Logging	Logs generated by the Backgrounder process are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/backgrounder`. For more information, see Tableau Server Logs and Log File Locations

What happens if a Backgrounder process goes down? Jobs on the failed Backgrounder process are retried once the Backgrounder process recovers from failure. Most background jobs are scheduled to run periodically, and the same background task will be picked up and performed normally at the next scheduled time by a functioning Backgrounder process.

Failed Backgrounder processes automatically restart as long as the computer itself is otherwise healthy, and the failed jobs will be retried.

To make the Backgrounder process highly available, you should configure one or more instances to run on multiple nodes in the cluster.

Managing Backgrounder Resources

Backgrounders as mentioned earlier in this topic, run server tasks, and can be resource intensive. There are several ways in which you can manage the resources that Backgrounder needs to run the server tasks:

- Increase the number of instances on a specific node: Backgrounder is single-threaded. It can only launch a single job at a time. Adding more Backgrounder instances to a node can increase the number of jobs that can be run in parallel on that node, but keep in mind that each job launched can itself use multiple threads. You can add Backgrounder instances up to one half the number of cores. When deciding where and how many Backgrounders to run, consider that each Backgrounder process launched for a job can use multiple threads, so adding Backgrounder instances may limit the effectiveness of each process. Also keep in mind how other server processes will affect each machine's available capacity.
- Isolate Backgrounder process: If you are running Tableau Server on a multi-node cluster, you can dedicate one or more nodes for running Backgrounder. For more information, see Recommended Baseline Configurations.

- Node Roles: You can also separate the type of jobs or workload that the Backgrounder on a node does. For example, you can have one node dedicated to running extract refreshes only. For more information, see Workload Management through Node Roles.
- Restricting users from manually running jobs: Currently users can run extract refreshes, flow run, and subscriptions using the web interface, REST API, and tabcmd commands, and can run them at any time of the day. This can take up server resources during times that your server is busy doing other activities. Starting in Tableau Server 2020.1, a new Run Now settings allows the server administrator to choose whether or not to allow users to run jobs manually. By disabling the Run Now option, you have better control over how backgrounders are utilized as well as being able to better predict the load. This does not apply or affect jobs that are generated for scheduled tasks. For more information on configuring this setting, see Server Settings (General and Customization).

Related content

- Improving group synchronization performance

Tableau Server Cache Server

The Cache Server provides a shared external query cache. It's a cache of key/value pairs which hold information from previous queries to speed up future requests. To make Cache Server highly available, configure one or more Cache Server processes on multiple nodes of the cluster.

Process	Cache Server
Status	Status of the Cache Server process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the Cache Server process are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/cacheserver`. For more information, see Tableau Server Logs and Log File Locations

What happens when Cache Server process goes down? The consequences are relatively mild. Tableau Server will still work but actions may take longer as they do not have cached

results available. As the queries rerun, the restarted Cache Server gets repopulated, eventually speeding things up for end users. In effect, Cache Server does not have an availability impact; however, it does have an impact on various end user performance scenarios.

To reduce impact on user performance, run multiple processes of this type across the cluster. We recommend you install no more than a maximum of six cache server instances, limiting each node to no more than two. Our testing indicates that installing more than a total of six cache server instances in a Tableau Server installation provides no improvement, and could have a negative impact on performance.

A failed Cache Server process is automatically restarted; as long as the computer itself is otherwise healthy, the Cache Server process will relaunch.

Tableau Server Client File Service

The Client File Service (CFS) stores and distributes files needed by TSM (e.g. certificates, customization files, etc.). Files that are managed by the Client File Service are renamed and obfuscated before they are distributed across the deployment. This process also parametrizes the file attributes that are required by Tableau services. As a result, files are not mapped to a single file location on the file system. Be sure you have an off-box backup of all files managed by CFS.

The following files are managed by CFS:

- SAML certificate file
- SAML key file
- SAML IdP metadata file
- OpenID.static.file
- Kerberos.keytab file
- LDAP Kerberos keytab file
- LDAP Kerberos conf file
- Mutual SSL certificate file
- Mutual SSL revocation file
- Customization header logo file
- Customization sign-in logo file
- Customization compact logo file

The following files are not managed or distributed by CFS:

- External SSL files. The certificate and key files for external SSL are stored and managed by the Coordination Service. You do not need to manually distribute these files.
- SSL files for LDAP external identity store. You must distribute the SSL certificate file manually to each node in the cluster. See Configure Encrypted Channel to LDAP External Identity Store.

The Client File Service functions much like the File Store does for files needed by business services. By default, CFS is only installed on the initial node of your Tableau Server installation. To configure CFS to for high availability, we recommend that you configure an instance of CFS on each of the nodes where you deploy the Coordination Service.

In a cluster, if a node that is running your only instance of CFS fails, any files being managed by CFS will be lost, and you will need to repopulate CFS those files by reimporting certs and custom images, and making any related configuration changes.

Process	Client File Service
Status	Status of the Client File Service process is not visible on the Status Page. Use the TSM CLI to view status. For more information, see View Server Process Status
Logging	Logs generated by the the Client File Service process are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/cli-entfileservice`. For more information, see Tableau Server Logs and Log File Locations

What happens when a CFS process fails? Nothing, as long as there is still at least one functioning CFS process in the cluster. The controller will redirect file transfer requests to the other working CFS process.

Failed Client File Service processes automatically restart as long as the computer itself is otherwise healthy.

Tableau Server Collections Service

The Tableau Server Collections service was added in Tableau Server version 2021.2. The Collections service powers the Collections feature. It provides information about collections, and connects with the Content Exploration service to get meta data about collections and items within collections. The Collections service also supports the Favorites feature.

Server Configuration

The Collections Service is automatically installed on the first node where Application Server (vizportal) is installed.

Multi-Node Configuration

For high availability and better performance, we recommend you install one instance of the Collections Service on every node that is running the Application Server (vizportal).

Process	Collections Service
Status	Status of the Collections Service is visible on the Status Page and from the command line using the `tsm status -v` command. For more information, see View Server Process Status
Logging	Logs generated by the the Collections Service process are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/collections`. For more information, see Tableau Server Logs and Log File Locations

Impact if the Collections Service is not running properly

If the Collections Service stops, it should automatically restart as long as the computer itself is otherwise healthy.

One instance of Collections service

If you have a single instance of the Collections service configured and that instance fails, collections and favorites will not be available.

Multiple instances of Collections service

If you have a multi-node installation of Tableau Server and have configured instances of the Collections service on multiple nodes, when one instance fails, collections and favorites may not be available to some users.

Log Files

The Collections Service creates two sets of log files:

- `control_collections*.log`: These logs will contain information about the service starting and being enabled.
- `collections_*.log`: Any errors or problems are logged here.

For more information, see Tableau Server Logs and Log File Locations.

Tableau Server Content Exploration Service

The Tableau Server Content Exploration Service extends the capabilities of the Search and Browse process and is responsible for indexing all content in Tableau Server. This service is available in Tableau Server 2021.1 and later and required to search and browse external assets such as databases and tables. Databases and tables are only available if you have Tableau Catalog enabled.

The Content Exploration Service is applicable to both quick and filtered search. This service is not based on SOLR but depends on the Elastic Search to perform its functions.

Server Configuration

The Content Exploration Service is automatically installed on the initial node.

Multi-Node Configuration

For high availability and better performance, we recommend you install at least one instance of the Content Exploration Service on every node that is running the Application Server.

Process	Content Exploration Service
Status	Status of the Content Exploration Service is visible on the Status Page and

	can be accessed using the TSM CLI to view. For more information, see View Server Process Status
Logging	Logs generated by the Content Exploration Service are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/con-tentexploration`. For more information, see Tableau Server Logs and Log File Locations

What happens when the Content Exploration Service fails?

If the Content Exploration Service stops working, any search requests for the content type it provides would fail to appear on the search results. If multiple instances of the service are installed, subsequent requests are routed to a healthy instance.

A failed Content Exploration service is automatically restarted; as long as the computer itself is otherwise healthy, the service will relaunch.

Performance Tuning

The Content Exploration Service has the ability to scale up as needed but based on the available memory on the node. However, there is a memory allocation set by default that determines the maximum amount of memory that can be used by the service. While it is not typically recommended that you change this setting if you are seeing performance issues due to insufficient memory allocation, you can run the following commands to change the maximum memory allocation setting.

First, retrieve the current maximum memory allocation by running the following command:

```
tsm configuration get -k contentexploration.vmopts
```

Run the following set command to change the maximum memory allocation by updating the -Xmx value, followed by applying the pending changes. Do not change any other options:

```
tsm configuration set -k contentexploration.vmopts -v "-Xmx<new
value>m -XX:+ExitOnOutOfMemoryError -Dspring.-
profiles.active=monolith"
```

For example:

```
tsm configuration set -k contentexploration.vmopts -v "-Xmx1024m -
XX:+ExitOnOutOfMemoryError -Dspring.profiles.active=monolith"
```

```
tsm pending-changes apply
```

Log Files

The Content Exploration Service creates two sets of log files:

- `control_contentexploration*.log`: These logs will contain information about the service starting and being enabled.
- `contentexploration_*.log`: Any errors or problems are logged here.

For more information, see Tableau Server Logs and Log File Locations.

Search Accuracy

There are couple of ways the Content Exploration Service is used to make sure that search-able content is kept up to date:

- Whenever content managed by Tableau Server or Tableau Cloud is modified, including permissions, the Content Exploration Service re-indexes the relevant documents to keep its search indexes up to date.
- A system generated Backgrounder task that periodically validates that all content is properly synchronized and re-indexes any discrepancies that are found.

Re-indexing

During restore, the restore process will initiate a full re-indexing of the content and external assets managed by Tableau Server. The re-indexing process consumes CPU resources which may be noticeable during backup and restore.

Tableau Server Coordination Service

The Coordination Service is built on Apache ZooKeeper, an open-source project, and coordinates activities on the server, guaranteeing a quorum in the event of a failure, and serving as the source of "truth" regarding the server topology, configuration, and state. The service is installed automatically on the initial Tableau Server node, but no additional instances are installed as you add additional nodes. Because the successful functioning of Tableau Server depends on a properly functioning Coordination Service, we recommend that for server installations of three or more nodes, you add additional instances of the Coordination Service by deploying a new Coordination Service ensemble. This provides redundancy and improved availability in the event that one instance of the Coordination Service has problems.

The hardware for your cluster can have some effect on how well the Coordination Service runs. In particular:

- Memory. The Coordination Service maintains state information in memory. By design, the memory footprint is small, and is typically not a factor in overall server performance.
- Disk speed. Because the service stores state information on disk, it benefits from fast disk speed on the individual node computers.
- Connection speed between nodes. The service communicates continuously between cluster nodes; a fast connection speeds between nodes helps with efficient synchronization.

Process	Coordination Service
Status	Status of the Coordination Service process is not visible on the Status Page. Use the TSM CLI to view status. For more information, see View Server Process Status
Logging	Logs generated by the the Coordination Service process are located in `/var/opt/tableau/tableau_server/data/tabsvc/logs/app-zookeeper`. For more information, see Tableau Server Logs and Log File Locations

Configuration for the Coordination Service

The Coordination Service is installed automatically on the initial node of Tableau Server. If you are running a single-node installation, you do not need to do anything to deploy or configure the Coordination Service. If your installation includes three or more nodes, you'll be prompted to configure a Coordination Service ensemble when you add your third node. This is not required, but is highly recommended as the Coordination Service serves a key function for high availability, acting as the source of "truth" about server topology, configuration, and state.

To configure a Coordination Service ensemble, use the TSM CLI and add the Coordination Service to the nodes you want running it. For details on how to deploy a Coordination Service ensemble, see Deploy a Coordination Service Ensemble .

The Coordination Service Quorum

To ensure that the Coordination Service can work properly, the service requires a *quorum*—a minimum number of instances of the service. This means that the number of nodes in your installation impacts how many instances of the Coordination Service you want to configure in your ensemble.

Number of Coordination Service instances to use

The maximum number of Coordination Service instances you can have in an ensemble on Tableau Server depends on how many Tableau Server nodes you have in your deployment. Configure a Coordination Service ensemble based on these guidelines:

Total number of server nodes	**Recommended number of Coordination Service nodes in ensemble (must be 1, 3, or 5)**	**Notes**
1-2 nodes	1 node	This is the default and requires no changes unless you want to move the Coordination Service off your initial node and onto your additional node.
3-4	3 nodes	

Total number of server nodes	Recommended number of Coordination Service nodes in ensemble (must be 1, 3, or 5)	Notes
nodes		
5 or more nodes	3 nodes or 5 nodes	Five is the maximum number of Coordination Service instances you can install. A 3-node Coordination Service ensemble allows for one of the ensemble nodes to fail without causing Tableau Server to fail. A 5-node ensemble allows for two of the ensemble nodes to fail without causing Tableau Server to fail. For most installations, three Coordination Service nodes are adequate, and because of the I/O-intensive nature of the Coordination Service, this is the most performant configuration. If high availability is your absolute priority, you may want to consider deploying a 5-node Coordination Service ensemble. This provides the most redundancy in the event that one or more nodes fail but will require more system resources. A maximum of two of the ensemble nodes can fail without impacting Tableau Server (as long as any other services on the node also exist on still-functioning nodes). To reduce performance impact, locate the Coordination Service on nodes that are running fewer other services or consider using Coordination Service-only nodes. For details, see Configure Tableau Server for High Availability with Coordination Service-Only Nodes.

If you reduce the number of nodes

If you reduce the nodes in your cluster from three (or more) to two nodes, a warning tells you Tableau Server can no longer support high availability:

```
A minimum of three Tableau Server nodes are required for high avail-
ability. You can add a third node now,
or continue with only two nodes. Continuing with only two nodes
means Tableau Server will not be highly available.
You can always add a third node later. Click OK to continue with 2
nodes, or Cancel to go back and add a node.
```

If you continue, Tableau Server will run, but you will not have any automatic failover of the repository.

Viewing Coordination Service Status

The Coordination Service is not included in the listing when you View Server Process Status. To see the state of the service, you can use the tsm status command:

```
tsm status -v
```

The output from the command shows you whether the service is running:

```
node1: TABLEAUSVR01
Status: RUNNING
'Tableau Server Gateway 0' is running.
'Tableau Server Application Server 0' is running.
'Tableau Server VizQL Server 0' is running.
'Tableau Server VizQL Server 1' is running.
'Tableau Server VizQL Server 2' is running.
'Tableau Server VizQL Server 3' is running.
'Tableau Server Cache Server 0' is running.
'Tableau Server Cache Server 1' is running.
'Tableau Server Coordination Service 0' is running.
'Tableau Server Cluster Controller 0' is running.
'Tableau Server Search And Browse 0' is running.
'Tableau Server Backgrounder 0' is running.
```

```
'Tableau Server Backgrounder 1' is running.
'Tableau Server Data Server 0' is running.
'Tableau Server Data Server 1' is running.
'Tableau Server Data Engine 0' is running.
'Tableau Server File Store 0' is running.
'Tableau Server Repository 0' is running (Active Repository).
'Tableau Server Administration Agent 0' is running.
'Tableau Server Administration Controller 0' is running.
'Tableau Server Service Manager 0' is running.
'Tableau Server License Manager 0' is running.
'Tableau Server Client File Service 0' is running.
'Tableau Server Database Maintenance 0' is stopped.
'Tableau Server Backup/Restore 0' is stopped.
'Tableau Server Site Import/Export 0' is stopped.
'Tableau Server SAML Service 0' is stopped.
```

Tableau Server Data Engine

Hyper is Tableau's in-memory Data Engine technology optimized for fast data ingests and analytical query processing on large or complex data sets. Hyper powers the Data Engine in Tableau Server, Tableau Desktop, Tableau Cloud, and Tableau Public. The Data Engine is used when creating, refreshing or querying extracts. It is also used for cross-database joins to support federated data sources with multiple connections.

Process	Data Engine
Status	Status of the Data Engine process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the Data Engine process are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/hyper`. For more information, see Tableau Server Logs and Log File Locations

Memory and CPU usage

The Data Engine is designed to leverage all available CPU and memory on the machine to provide the fastest response times.

CPU usage

Hyper technology leverages the new instruction sets in CPU and is capable of parallelizing and scaling to all the available cores. Hyper technology is designed to scale to many cores efficiently, and also to maximize the use of each single core as much as possible. This means that you can expect to see up to an average of 75% use of total CPU per hour during query processing. Adding more CPU should result in performance improvement.

> **Note:** The 75% hourly average usage is the default, and should be left unchanged unless you are running Data Engine on a dedicated server node. For more information about running Data Engine on a dedicated node, see Optimize for Extract Query-Heavy Environments.

Modern operating systems such as Microsoft Windows, Apple macOS, and Linux have mechanisms to make sure that even if a CPU is fully used, incoming and other active processes can run simultaneously. In addition, to manage overall resource consumption and to prevent overloading and completely starving other processes running on the machine, the Data Engine monitors itself to stay within the limits set in the Tableau Server Resource Manager (SRM). Tableau Server Resource Manager monitors the resource consumption and notifies Data Engine to reduce the usage when it exceeds the predefined limit.

Since the Data Engine is designed to utilize the available CPU, it is normal to see spikes in CPU usage at times. If however, you see high CPU usage (ex: 95%) for extended periods of time (an hour or more), this can mean a couple of things:

- There is a high load of queries. This can happen if a server is under stress due to overload of multiple client requests and the queries are queuing up. If this happens often, it is an indication that more hardware is required to serve the clients. Adding more CPU in this case should help to improve performance.
- There is one long running query. In this case, the Tableau Server resource Manager will

stop long running queries based on the timeout settings. This was also true for the Tableau Server versions earlier than version 10.5

For more information on Tableau Server Resource Manager, see General Performance Guidelines.

Memory usage

Memory usage of the Data Engine depends on the amount of data required to answer the query. The Data Engine will try to run this in-memory first. A working set memory is allocated to store an intermediate data structure during query processing. In most cases, systems have enough memory to do these types of processing, but if there isn't enough available memory, or if more than 80% of RAM is utilized, the Data Engine shifts to spooling by temporarily writing to disk. The temporary file get deleted after the query has been answered. Therefore, spooling is an indication that more memory may be needed. Memory usage should be monitored and upgraded appropriately to avoid performance issues caused by spooling.

To manage memory resources on the machine, the maximum memory limit for Data Engine is set by Tableau Server Resource Manager (SRM).

Server configuration, Scalability, and Performance

- A single instance of Data Engine is automatically installed per node where an instance of File Store, Application Server (VizPortal), VizQLServer, Data Server, or Backgrounder is installed on Tableau Server. The Data Engine can scale by itself and uses as much CPU and memory as needed, thus removing the need for multiple instances of the Data Engine. For more information on the server processes, see Tableau Server Processes.
- The instance of Data Engine installed on the node where File Store is installed is used for querying data for view requests. The instance of Data Engine installed on the node where backgrounder is installed is used for extract creation and refreshes. This is an

important consideration when you are doing performance tuning. For more information, see Performance Tuning.

- Data Server, VizQL Server, and the Application Server (VizPortal) all use the local instance of Data Engine to do cross-database joins and create shadow extracts. Shadow extract files are only created when you work with workbooks that are based on non-legacy Excel or text, or statistical files. Tableau creates a shadow extract file in order to load the data more quickly.
- In Tableau Server 10.5 one instance of Data Engine is installed automatically when you install backgrounder. The backgrounder process uses the single instance of Data Engine (hyperd.exe) installed on the same node.

Important! There are exceptions to when the Data Engine is installed on the same node as File Store. When File Store is configured external to Tableau Server, Data Engine is no longer installed with File Store. In this configuration where Tableau Server is configured with an External File Store, Data Engine, will continue to be installed with the other process as noted above. In addition, you can also configure Data Engine on a node without other processes - but only when File Store is configured externally. For more information on External File Store, see Tableau Server External File Store.

Scalability:

You can scale up with the new Data Engine: Since cores are fully utilized, adding more cores makes individual queries execute faster which in turn allows for more queries to execute in less time.

Memory usage should be monitored and upgraded appropriately to avoid the performance issues caused by spooling.

For more information on Scalability, see Tableau Server Scalability.

Performance:

Performance benefits

Starting in 10.5, Hyper technology has been integrated with Tableau Data Engine to give you the following key benefits:

- **Faster extract creation:** With Hyper technology, extracts are generated almost as fast as the source system can deliver data, no sorting needed.
- **Support for larger extracts:** Prior to this release, you might have not been able to get all your data into a single extract. With Hyper technology, much larger amounts of data can be included in a single extract.
- **Faster analysis of extracts:** In many cases you will see faster querying of data for larger extracts, or workbooks with complex calculations.

Here are some reasons why the Data Engine powered by Hyper performs better on larger or complex extracts and is optimized for faster querying:

- **Hyper technology is designed to consume data faster**. Unlike in previous versions, the Data Engine does not do any post processing like sorting. With Hyper, post processing steps like sorting are not needed giving the Data Engine the ability to perform better with larger extracts.
- **Hyper technology is memory-optimized**. This means that when needed, all data lives in memory. This results in fast data access times.
- **Hyper technology is CPU optimized**. This means that Data Engine now fully parallelizes the query execution and utilizes available CPU in such a way that query execution time scales almost linearly with the number of cores in the machine.
- **Hyper is a compiling query engine**. Queries are either interpreted or compiled to the machine code for maximum performance and allowing the Data Engine to get most performance out of modern hardware (CPU, large main-memory capacities).

- **Hyper technology uses advanced query optimizations to make queries faster.** Along with many additional advanced techniques such as, materializing min and max values for each column, mini-indices to optimize search ranges, more granular data block-level dictionaries, advanced logic for join and sub-query performance optimizations, the new Data Engine offers many improvements over the previous Tableau Data Engine in terms of performance and scalability.

For more information on performance, start with General Performance Guidelines, and Performance Tuning

Tableau Server Data Server

The Data Server manages connections to published data sources on Tableau Server. To make Data Server highly available, configure one or more Data Server processes to run on multiple nodes of the cluster.

Process	Data Server
Status	Status of the Data Server process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the Data Server process are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/dataserver`. For more information, see Tableau Server Logs and Log File Locations

What happens if a Data Server process fails? Queries running through the Data Server process will fail, resulting in a failed view rendering, extract refresh, or alert. Subsequent requests, including a retry of the failed operation, should succeed as long as a working Data Server is available to accept rerouted requests.

Tableau Server is not dependent on Data Server to function; however, without a running Data Server, workbooks on the server lose the ability to query or to connect to published data sources. Any view that does not use Data Server for any of its data sources will still function correctly.

Tableau Server Data Source Properties Service

Introduced in version 2020.1.0, the Tableau Server Data Source Properties service provides metadata for published data source from the Application Server (VizPortal) to client services like Ask Data.

Note: Tableau's Ask Data feature was retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2.

One instance of the Data Source Properties service is configured on the initial node of Tableau Server. You can add additional instances on the initial node or on other nodes in a multi-node installation. You are required to configure at least one instance of the Data Source Properties service on any node that has an instance of Application Server.

Process	Data Source Properties Service
Status	Status of the Data Source Properties Service is visible on the Status Page and from the command line using the `tsm status -v` command. For more information, see View Server Process Status
Logging	Logs generated by the Data Source Properties Service are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/tdsservice`. For more information, see Tableau Server Logs and Log File Locations

Log files for the Data Source Properties service

The Data Source Properties service creates two sets of log files:

- `control-tdsservice*.log`. These logs will contain information about the service starting and being enabled.
- `tdsservice_*.log`. Any errors or problems are logged here.

For more information, see Log File Snapshots (Archive Logs).

Data Source Properties service in a multi-node cluster

How you configure the Data Source Properties service in a multi-node cluster depends on how you configure your nodes. In any installation of Tableau Server you must have at least one instance of the service. In addition, you are required to configure at least one instance of Data Source Properties on any node that is configured with the Application Server (VizPortal).

Tableau Server File Store

This topic describes File Store process when configured to run locally on Tableau Server. However, File Store can be run locally as well as external to Tableau Server. For more information on Tableau Server External File Store, see Tableau Server External File Store.

The Tableau Server File Store process controls the storage of extracts. When File Store is installed, an instance of the Data Engine is also installed unless the node already has an instance of the data engine. In highly available (HA) environments, the File Store ensures that extracts are synchronized to other file store nodes so they are available if one file store node stops running.

Process	File Store
Status	Status of the File Store process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the File Store process are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/filestore`. For more information, see Tableau Server Logs and Log File Locations

The decommission Command

If you want or need to remove a File Store instance you need to decommission File Store first using the `tsm topology filestore decommission` command. If you don't decommission File Store before you attempt to remove it, you will be prompted to do so. Decommissioning puts the File Store instance into read-only mode and copies any unique data contained in the instance to the other File Store(s) in the cluster. While a File Store instance is

being decommissioned, this shows on the Status page, and once all unique content has been copied to other File Store nodes, the decommissioned node shows as ready to be removed.

Decommissioning File Store when co-located with the Administrative Controller

Tableau backup is fastest when an instance of File Store is located on the same node as the TSM Administrative Controller. If you are removing an instance of File Store that is co-located with the Controller, you will be warned about the performance impact for backups.

Tableau Server Gateway Process

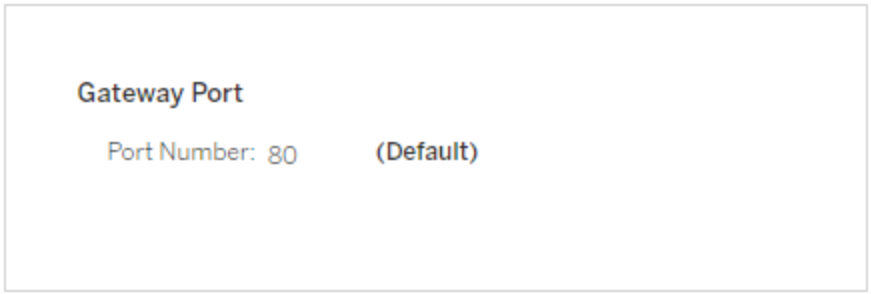
The Tableau Server gateway process is an Apache web server component (`httpd.exe`). Its role is to handle requests to the server from all clients—Tableau Desktop, mobile devices, a proxy, a load balancer, etc.

The server runs a single instance of the gateway process; you can't run more than one per machine. The gateway process is required on any node with an instance of VizQL Server or Vizportal.

Process	Gateway
Status	Status of the Gateway is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the repository are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/httpd`. For more information, see Tableau Server Logs and Log File Locations

Port assignment

By default, the gateway process listens for requests on port 80 (for HTTP requests) and 443 (for SSL requests). When you install Tableau Server on a computer, part of the server configuration makes sure that this port is open in the computer's firewall. If the computer is running a different process that requires port 80 for HTTP, you can change the port assignment for the gateway process during installation. (You cannot change the SSL/HTTPS port.)



If you need to set the HTTP port after installation, then run the following TSM command, where *node* is the node number (for example, `node1`) and *nn* is the new port number:

```
tsm topology set-ports --node-name node --port-name gateway:primary
--port-value nn
```

If you are running Tableau Server with a reverse proxy server, you will need to configure other port and host-related settings on Tableau Server. See Configuring Proxies and Load Balancers for Tableau Server.

Log files for the gateway process

The gateway process creates two sets of log files:

- Activity logs. The name for these log files has the format `access.yy_mm_dd_hh_mm_ss.log`.
- Error logs. All errors are logged in a single file named `error.log`.

For more information, see Log File Snapshots (Archive Logs).

Gateway processes in a cluster

If your server environment is distributed across multiple machines, you can run a single gateway process on each node of the cluster. The most common scenario for running a gateway process on multiple computers in the cluster is that you have a load balancer in front of the cluster. In this scenario, the load balancer distributes requests to any gateway in the cluster. If you need to take a node off line (for example, to perform maintenance on that node), you can

disable the load balancer's routing to that machine. When the maintenance is complete, you can re-enable the node on the load balancer.

You must have a gateway process running on at least one computer in the cluster. If you remove the gateway process from the primary server, you must make sure that another computer in the cluster is running the gateway process. You must also make sure that that computer is reachable by clients.

An instance of the gateway process is required on any node that is configured for one of these processes or services: VizQL Server, Vizportal, or Tableau Prep Flow Authoring.

If the Tableau Server is configured to use SSL, you must make sure that the certificate for SSL support is in the same location on each computer in the cluster that has the gateway process running. For more information about using SSL, see Configure SSL for External HTTP Traffic to and from Tableau Server.

Similarly, if the server installation uses a custom logo, the logo must be in the same location on every computer that is running the gateway process.

If you need to change the port number that the gateway process listens on, as explained earlier, you can use the configuration dialog box or run the following command for each worker computer that is running the gateway process:

```
tsm topology set-ports --node-name node --port-name gateway:primary
--port-value nn
```

Additional information

Configuring Proxies and Load Balancers for Tableau Server

Add a Load Balancer

Index and Search Server

The Tableau Server Index and Search Server process, based on OpenSearch, handles fast search, filter, retrieval, and display of content metadata on your Tableau Server site. Begin-

ning with Tableau Server version 2023.3, Index and Search Server completely replaces the Search and Browse process.

Server Configuration

The Index and Search Server is automatically installed on the initial node.

Multi-Node Configuration

To configure Index and Search Server for high availability, configure the process on multiple nodes. We recommend you configure an odd number of Index and Search Server instances. On Tableau Server Clusters that have 3 or more nodes, you should configure Index and Search Server on at least three different nodes.

Process	Index and Search Server
Status	Status of the Index and Search Server is visible on the Status Page and can be accessed using the TSM CLI to view. For more information, see View Server Process Status
Logging	Logs generated by the Index and Search Server are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/index-andsearchserver`. For more information, see Tableau Server Logs and Log File Locations

What happens when the Index and Search Server fails?

If Index and Search Server fails, users can still sign in to Tableau Server, however the different content types (workbooks, projects, data sources, and views for example) will not appear in the Home page, the search bar or the Explore page. Existing workbooks and Views should still be accessible and fully functional if accessed from a bookmarked url. The content is not actually missing. Rather, the content is not being returned in the search results; it will be displayed again after the Index and Search Server restarts. A failed Index and Search Server is automatically restarted; as long as the computer itself is otherwise healthy, the service will relaunch.

If more than one Index and Search Server is configured and running on multiple nodes when the failure occurs, requests made to a failed Index and Search Server will also fail, but subsequent requests will be routed to working Index and Search Server instances. Each Index and Search Server instance indexes across all nodes in the cluster, therefore as long as one Index and Search Server instance is running, results will still be returned across all nodes.

Performance Tuning

The Index and Search Server heap size can be configured by using the `tsm set configuration` command with the `indexandsearchserver.vmopts` option. For more information, see indexandsearchserver.vmopts

Re-indexing

During restore, the restore process will initiate a full re-indexing of the content and external assets managed by Tableau Server. The re-indexing process consumes CPU resources which may be noticeable during backup and restore.

Tableau Server Internal Data Source Properties Service

The Tableau Server Internal Data Source Properties service was introduced in version 2020.1.0 and communicates with the Data Source Properties service. It is managed internally by Tableau Server, and cannot be configured by an administrator.

An instance of the Internal Data Source Properties service is automatically configured on any node that has an instance of the Data Source Properties service.

Process	Internal Data Source Properties Service
Status	Status of the Internal Data Source Properties Service is visible on the Status Page and from the command line using the `tsm status -v` command. For more information, see View Server Process Status
Logging	Logs generated by the Internal Data Source Properties Service are located in `/var/opt/tableau/tableau_server/data/tabsvc/logs/tdsnativeservice`. For more information, see Tableau Server Logs and Log File Locations

Log files for the Internal Data Source Properties service

The Internal Data Source Properties service creates two sets of log files:

- `control-tdsnativeservice_*.log`. These logs will contain information about the service starting and being enabled.
- `nativeapi_tdsnativeservice_*.log`. Any errors or problems are logged here.

For more information, see Log File Snapshots (Archive Logs).

Internal Data Source Properties service in a multi-node cluster

One instance of the Internal Data Source Properties services is added to any node that has an instance of Data Source Properties configured. Because the Internal Data Source Properties service is managed internally by Tableau, there is nothing an administrator can configure related to the service.

Tableau Server Messaging Service

The Tableau Server messaging service uses Apache ActiveMQ beginning with version 2019.4. This is a publish/subscribe platform that enables secure, scalable, performant, and highly available message-oriented communication for microservices. The Messaging Service is used to support communication between microservices in Tableau Server.

The server runs a single instance of the Messaging Service by default. In version 2020.1 and later, if you have a multi-node instance of Tableau Server you can configure a second instance of the Messaging Service.

Process	Messaging Service
Status	Status of the Messaging Service is visible on the Status Page and from the command line using the `tsm status -v` command. For more information, see View Server Process Status
Logging	Logs generated by the repository are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/act-ivemqserver`. For more information, see Tableau Server Logs and Log File

	Locations

Impact if Messaging Service is not running properly

If the Messaging Service stops or fails to start the consequences depend on whether you have one, or two instances of the Messaging Service configured.

Multiple instances of Messaging Service (multi-node, version 2020.1 and later)

You can configure two instances of the Messaging Service configured if you have version 2020.1 or later, *and* you have a multi-node installation of Tableau Server. In this case, if one instance fails, the other instance handles all traffic, and you can remove and reinstall the failed instance.

One instance of Messaging Service

If you have a single-node installation of Tableau Server, you are limited to a single instance of the Messaging Service. In this case, if the configured instance fails, Tableau Server will continue to function, but the status will show as "Degraded", and any event message that was sent to the Messaging Service while it was down may be lost, even if the service restarts. Event messages include permission changes ro projects.

If the Messaging Service stops, it is automatically restarted as long as the computer itself is otherwise healthy.

Messaging Service in a multi-node cluster

How you configure the Messaging Service in a multi-node cluster depends on what version of Tableau Server you are running. In all cases, you must run one instance of the Messaging Service.

Starting with version 2020.1, you can run two instances of the service in a multi-node environment. We recommend you run two instances as this provides redundancy. You can add a second instance to any node that does not already have the Messaging Service configured. You cannot add more than a combined total of two instances in your multi-node cluster.

Tableau Server Metrics Service

Retirement of the legacy metrics feature

Tableau's legacy metrics feature was retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2. In October 2023, Tableau retired the ability to embed legacy metrics in Tableau Cloud and in Tableau Server version 2023.3. With Tableau Pulse, we've developed an improved experience to track metrics and ask questions of your data. For more information, see Create Metrics with Tableau Pulse to learn about the new experience and Create and Troubleshoot Metrics (Retired) for the retired feature.

The Metrics service is responsible for reading and writing Metric data in Tableau Server. The service is required in order for Metrics to work properly. To make the Metrics service highly available, configure one or more instances of the service to multiple nodes of the cluster. We recommend you configure at least one instance on every node that is running the Application Server (VizPortal).

Process	Metrics Service
Status	Status of the Metrics Service is visible on the TSM Status Page and from the command line using the `tsm status -v` command. For more information, see View Server Process Status
Logging	Logs generated by the Metrics Service process are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/metrics`. For more information, see Tableau Server Logs and Log File Locations

Impact if the Metrics Service fails

The Metrics Service is required for Metrics to function properly on Tableau Server. In order to minimize issues, install multiple instances of the service in your server installation. If you have a single-node instance of Tableau Server, configure at least two instances of the service on your node. If the Metrics Service stops working, any Metrics requests that are in-process will fail. Subsequent requests are routed to a healthy instance of the service.

Metrics Service in a multi-node cluster

The Metrics Service is required for Metrics to function properly on Tableau Server. In order to minimize issues, install multiple instances of the service in your server installation. In multi-node installations, we recommend you configure at least one instance of the Metrics Service on every node that is running the Application Server (VizPortal). This provides redundancy and maximum performance.

Log files for the Metrics service

The Metrics Service creates two sets of log files:

- `control-metrics*.log`. These logs will contain information about the service starting and being enabled.
- `metrics_*.log`. Any errors or problems are logged here.

For more information, see Log File Snapshots (Archive Logs).

Tableau Server Microservice Containers

By default, one instance of the Interactive Microservice Container is added to every node that has Application Server (Vizportal) installed, and one instance of the Non-Interactive Microservice Container is added to every node that has Backgrounder installed. Although you cannot add a Microservice container directly, you can use the TSM CLI to change the number of instances for both Microservice Containers, when necessary. If all instances of Backgrounder or Application Server are removed from a node, the container process is also removed.

Microservice Containers and the microservices:

- Interactive Microservice Container:
    - MessageBus Microservice
    - Relationship Query Microservice
    - Credentials Service
- Non-Interactive Microservice Container:

- Relationship Ingestor Microservice
- External Content Provider Microservice
- Flow Provider Microservice
- Content Provider Microservice

Viewing Microservice Container Status

You can see the status of the Microservice Container processes from the TSM Status page, or from the command line, using the `tsm status -v` command. When you use the TSM Status page to View Server Process Status, the status of each container process is visible, but you cannot see the status of any of the microservices in the containers. When you use the command line, more detail is shown, including the status of each individual microservice.

Microservice Container Status

The status of a container process depends on the status of the microservices within the container. When all microservices within a container process are running as expected, the container status is Active (on the TSM Status page) or `running` (when viewed from the TSM command line). If all microservices within a container process are stopped, the status for the container is Error (on the TSM Status page) or `stopped` (from the TSM command line). If a microservice is stopped but at least one other microservice is running, the container status is Degraded (on the TSM Status page) or `degraded` (from the TSM command line).

When all microservices within a container process have a status of running, the container status is `Active`. If any microservice in a container is in an error state (has a status of `stopped`) the container process status is `degraded`. If all microservices in a container are in an error state, the container status is `error`.

Use the TSM web interface

To view the Microservice Container status from the TSM Status page:

1. Open TSM in a browser:

   http://<tsm-computer-name>:8850

2. Click **Status**:

The page displays the status for the Interactive Microservice Container and Non-Interactive Microservice Container processes, as well as status for other processes running as part of TSM or Tableau Server.

You cannot see the status of any individual microservice within a container process, but if the container process has a status of Active (a green check), the microservices it contains are all running as expected. To see the status of individual microservices, use the TSM command line.

> **Note**: The status of the container processes does not display on the older Tableau Server status page. For details about the two status pages and how they differ, see View Server Process Status.

Use the TSM CLI

To view the Microservice Container status from the TSM command line:

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run the following command:

   ```
   tsm status -v
   ```

   The output from the command shows you the status of the container services and the microservices in them:

   ```
   node1: TABLEAUSVR01
   Status: RUNNING
   'Tableau Server Gateway 0' is running.
   'Tableau Server Application Server 0' is running.
   'Tableau Server Interactive Microservice Container 0' is running.
       'MessageBus Microservice 0' is running.
   ```

```
    'Relationship Query Microservice 0' is running.
'Tableau Server VizQL Server 0' is running.
'Tableau Server VizQL Server 1' is running.
'Tableau Server VizQL Server 2' is running.
'Tableau Server VizQL Server 3' is running.
'Tableau Server Cache Server 0' is running.
'Tableau Server Cache Server 1' is running.
'Tableau Server Coordination Service 0' is running.
'Tableau Server Cluster Controller 0' is running.
'Tableau Server Search And Browse 0' is running.
'Tableau Server Backgrounder 0' is running.
'Tableau Server Backgrounder 1' is running.
'Tableau Server Non-Interactive Microservice Container 0' is
running.
    'Relationship Ingestor Microservice 0' is running.
    'External Content Provider Microservice 0' is running.
    'Flow Provider Microservice 0' is running.
    'Content Provider Microservice 0' is running.
'Tableau Server Data Server 0' is running.
'Tableau Server Data Server 1' is running.
'Tableau Server Data Engine 0' is running.
'Tableau Server File Store 0' is running.
'Tableau Server Repository 0' is running (Active Repository).
'Tableau Server Tableau Prep Conductor 0' is running.
'Tableau Server Elastic Server 0' is running.
'Tableau Server Ask Data 0' is running.
'Tableau Server Administration Agent 0' is running.
'Tableau Server Administration Controller 0' is running.
'Tableau Server Service Manager 0' is running.
'Tableau Server License Manager 0' is running.
'Tableau Server Client File Service 0' is running.
'Tableau Server Database Maintenance 0' is stopped.
'Tableau Server Backup/Restore 0' is stopped.
'Tableau Server Site Import/Export 0' is stopped.
'Tableau Server SAML Service 0' is stopped.
```

```
c:\Program Files\Tableau\Tableau Server-
\packages\scripts.near.18.1216.1859>
```

Tableau Server Repository

Tableau Server Repository is a database that stores server data. This data includes information about Tableau Server users, groups and group assignments, permissions, projects, data sources, workbooks, and extract metadata and refresh information.

The Repository is also referred to as the *PostgreSQL* repository or database .

Process	Repository
Status	Status of the Repository is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the repository are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/pgsql`. For more information, see Tableau Server Logs and Log File Locations

Preferred active repository

When you configure Tableau Server you have the option to specify a node as the preferred active repository. When Tableau Server is configured for repository failover, the preferred active repository node is the one used for the active repository. This is an optional step, and if you do not specify a preferred active repository node, Tableau Server will select the active repository node on startup.

To configure the preferred active repository, use the `tsm configuration set` command to configure the `pgsql.preferred_host` option:

```
tsm configuration set -k pgsql.preferred_host -v "<host_name>"
```

Note: The `host_name` is case-sensitive and must match the node name shown in the output of `tsm status -v`.

Configure a preferred active repository node if you want Tableau Server to select a specific node on startup. You might want to do this if you have a particular server you want to use for your active repository (a computer with more disk space or memory for example), or if you are using custom administrative views. Custom administrative views have embedded connection information that refers to the repository for which you created the views. For more information on connecting to the Tableau Server repository, see Collect Data with the Tableau Server Repository

The failoverrepository Command

If failover occurs and your passive repository becomes the active repository, it remains the active repository until either Tableau Server restarts or you use the `tsm topology failover-repository` command to switch back. Specify the repository you want to be the active one, or specify that the preferred active repository (if configured) should be made active again. For more information, see tsm topology failover-repository.

Tableau Server Resource Limits Manager

This process was introduced in Tableau Server 2022.1.

The Tableau Server Resource Limits Manager tracks backgrounder resource usage in relation to the set resource limits to make sure the resource limits are applied correctly.

Process	Resource Limits Manager
Status	Status of the Resource Limits Manager is visible only in TSM CLI.
Logging	Logs generated by the Content Exploration Service are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/allegro`. • The configuration logs are included in the Backgrounder logs • The resource consumptions logs are included in the vizportal logs. For more information, see Tableau Server Logs and Log File Locations

The Resource Limits Manager is automatically and by default installed on the initial node of Tableau Server. We do not recommend adding more processes or configuring this on additional nodes of the Tableau Server.

The maximum memory usage is set to 512 MB.

What happens when the Resource Limits Manager fails?

The resource limits will no longer be applied but the jobs will continue to run using the available Backgrounder resources. The behavior will be similar to the scenario where there are no set resource limits.

You can see the status using the tsm command - tsm status -v

Tableau Server SAML Service

For Tableau Server installations that have site-specific SAML enabled, there will also be a SAML Service instance running on each node that is configured with Application Server. This service is automatically configured when site-specific SAML has been enabled on the server.

- In version 2023.1.x and later, SAML Service on Tableau Server isn't displayed until site SAML is enabled.
- In version 2022.3.x and earlier, SAML Service on Tableau Server is shown as stopped unless site SAML is enabled.

Process	SAML Service
Status	Status of the SAML Service process is not visible on the Status Page. Use the TSM CLI to view status. For more information, see View Server Process Status
Logging	Logs generated by the the SAML Service process are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/samlservice`. For more information, see Tableau Server Logs and Log File Locations

If this process goes into a failed state, then users will not be able to sign in to Tableau Server if the user request is routed to an Application Server on this node. Like other processes, when a node containing the SAML Service fails, the failed process should automatically restart within seconds.

Tableau Server Search and Browse

Note: Starting in version 2023.3, Search and Browse is retired (no longer installed). It is replaced by the Index and Search Server.

Starting in version 2022.3, Search and Browse is deprecated (installed but no longer used by Tableau Server). If you are running Tableau Server version 2022.3 or 2023.1, you *should not configure more than one instance of Search and Browse* for any installation. Configuring more than one instance can, in rare cases, result in stability issues.

The Search & Browse process, based on Apache SOLR (in Tableau Server versions 2020.4 and earlier) and also known as searchserver, handles fast search, filter, retrieval, and display of content metadata on your Tableau Server site. To configure high availability for the Search & Browse process, configure the process on multiple nodes. Starting in Tableau Server version 2021.1, the functionality of the Search and Browse process is enhanced by the Content Exploration Service. For more information, see Tableau Server Content Exploration Service

Process	Search & Browse
Status	Status of the Search & Browse process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the Search & Browse process are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/searchserver`. For more information, see Tableau Server Logs and Log File Locations

What happens if a Search & Browse process fails? Users can still sign in to Tableau Server, but workbook content will appear to be missing. The content is not actually missing. Rather,

the content is not being returned in the search results; it will be displayed again after the Search and Browse process restarts.

If more than one Search & Browse process is configured and running on multiple nodes when the failure occurs, requests made to a failed Search & Browse process will also fail, but subsequent requests will be routed to working Search & Browse processes. Each Search & Browse process indexes across all nodes in the cluster, therefore if all but one Search & Browse process fails, results will still be returned across all nodes.

Tuning the Search & Browse Process

There are several ways you can change how the Search & Browse process works in Tableau Server. In most cases you should not need to change anything, but here are some configuration changes you can make.

Search & Browse Max Heap Memory

You can configure Search & Browse options using the `searchserver.javamemopts` configuration key. The most important setting you can configure is the one controlling how large the maximum heap memory should be and specified by the `-Xmx` option. By default this is set by Tableau, based on the available physical memory for the server node running the Search & Browse service. The value should be set as high as possible, based on available physical memory and memory usage, but no higher than 24 GB.

When changing the amount of max heap memory, you need to take into account any other services on the system. If you're running low on memory on the computer running Tableau Server, you should consider adding more memory, or scaling down the topology on the machine. If you are running Search & Browse on a node with few other processes, or with a large amount of physical memory, you can consider increasing the amount used by the search service. If you are running Search & Browse on a node with more than the default number of processes, you should consider decreasing the maximum heap memory allocation to avoid having the service attempt to use more memory than is available.

Default maximum heap memory allocations

The table below shows the *default* amount of heap memory Tableau allocates to Search & Browse on a server node, based on available memory:

System memory	SOLR heap memory
<= 16 GB	1 GB
<= 32 GB	2 GB
<= 64 GB	4 GB
<= 128 GB	8 GB
> 128 GB	16 GB

To change the value of max heap memory, set the `-Xmx` value of `search-server.javamemopts` using the `tsm configuration set` command.

First get the current values:

```
C:\WINDOWS\system32>tsm configuration get -k search-server.javamemopts
-Xmx512m -Xms512m -XX:+ExitOnOutOfMemoryError -XX:-UsePerfData
```

Next, change the value of the `-Xmx` option. Include but do not change all other options:

```
tsm configuration set -k searchserver.javamemopts -v "-Xmx8g -Xms512m -XX:+ExitOnOutOfMemoryError -XX:-UsePerfData"
```

For more details about the `searchserver.javamemopts` configuration key, see search-server.javamemopts.

Client session timeouts

You can configure how long Search & Browse clients will wait to establish a connection to the Search & Browse server, and to Coordination Service (Zookeeper). Both timeout values are set to a relatively high value, but if you experience issues browsing to server content, and vizportal and backgrounder logs show timeouts connecting to SOLR, try increasing these settings. If this has no impact, you may be running into limited resources on the Tableau Server computer.

`searchserver.connection_timeout_milliseconds`

`searchserver.zookeeper_session_timeout_milliseconds`

To change the values use the `tsm configuration set` command:

```
tsm configuration set -k searchserver.connection_timeout_mil-
liseconds -v
```

For more details about the `searchserver.connection_timeout_milliseconds` configuration key, see searchserver.connection_timeout_milliseconds.

Zookeeper connection health check timeout at startup

When Tableau Server is starting, resource usage is high, especially related to CPU usage. If Search & Browse does not make a connection to Coordination Service (zookeeper), it will fail to start. To account for this, Tableau Server performs a health check on the Coordination Service before starting Search & Browse.

If your Tableau Server computer is especially busy, or if Search & Browse fails to start, increase this timeout value.

An error is written to the control-searchserver.log files when this health check fails: `Failed zookeeper health check. Refusing to start SOLR.`

To increase the amount of time Tableau Server waits for a successful health check, set the value of `searchserver.startup.zookeeper_healthcheck_timeout_ms` using the `tsm configuration set` command.

```
tsm configuration set -k searchserver.startup.zookeeper_healthcheck_
timeout_ms -v <nnnnnn>
```

For more details about the `searchserver.startup.zookeeper_healthcheck_timeout_ms` configuration key, see searchserver.startup.zookeeper_healthcheck_timeout_ms.

Tableau Statistical Service

The Tableau Statistical Service manages the statistical engine behind Explain Data and predictive modeling functions on Tableau Server. This service is available in Tableau Server 2022.1 through 2023.1.x.

> **Note:** The Statistical Service was retired in version 2023.3.0 and no longer shows up on the status page or in the output of the tsm status command.

Server configuration

The Tableau Statistical Service is automatically installed on any node where VizQL is installed.

Process	Tableau Statistical Service
Status	Status of the Tableau Statistical Service process is visible on the Status Page and can be accessed using the TSM CLI to view. For more information, see View Server Process Status
Logging	Logs generated by the Tableau Statistical Service process are located in `/var/opt/tableau/tableau_server/data/tabsvc/logs/statsservice`. For more information, see Tableau Server Logs and Log File Locations

Impact if the Tableau Statistical Service fails

If the Tableau Statistical Service stops working, Explain Data will be unavailable, and views that contain calculations with predictive functions MODEL_PERCENTILE and MODEL_QUANTILE won't render.

Performance

If there is increased usage of Explain Data or predictive modeling functions on Tableau Server, you may benefit from installing additional instances of the Tableau Statistical Service on nodes running VizQL. For more information on configuring the topology of a Tableau Server node, see Configure Nodes.

Log files

The Statistical Service creates two sets of log files:

- `control_statsservice*.log`: These logs will contain information about the service starting and being enabled.
- `stdout_statsservice_*.log`: These logs will contain information about status and errors.

For more information, see Tableau Server Logs and Log File Locations.

Tableau Server TSM Maintenance Services

There are three TSM Maintenance Services that are installed on every node of the cluster: Database Maintenance, Backup/Restore, and Site Import/Export.

Processes	Database Maintenance, Backup/Restore, and Site Import/Export.
Status	Status of the TSM Maintenance services are not visible on the Status Page. Use the TSM CLI to view status. For more information, see View Server Process Status
Logging	Logs generated by the the Service Manager process are located in `/var/opt/tableau/tableau_server/data/tabsvc/logs/`, in the `datamaintenance`, `backuprestore`, and `siteimportexport` directories. For more information, see Tableau Server Logs and Log File Locations

These services remain stopped unless a specific maintenance task that requires them is initiated by the administrator. Additional high-availability configuration is not required for these services. These services are used only for maintenance tasks such as backup and restore and should not impact the functioning of Tableau Server for the end users.

Tableau Server VizQL Server

The VizQL Server loads and renders views, and computes and executes queries. To achieve high availability for the VizQL Server process, configure one or more instances to run on multiple nodes.

Process	VizQL Server
Status	Status of theVizQL Server process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by theVizQL Server process are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/vizqlserver`. For more information, see Tableau Server Logs and Log File Locations

What happens if a VizQL Server process fails? If there is only one VizQL Server process and it fails, then Tableau Server will no longer be able to render any views. High availability requires configuring redundant VizQL processes. A fairly typical configuration consists of two to four VizQL Server processes on each node.

This simultaneously serves the need for high availability and scalability. If multiple VizQL Server processes are running, then the failure of a single process will result in the failure of any requests and the loss of session data at the time of its failure. Any future requests will be routed to the other working VizQL Server processes on the Tableau Server cluster.

Tableau Prep Conductor

The Tableau Prep Conductor process runs flows and processes flows for ingestion by Data Catalog. It leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output. Starting in 2020.4 Data Management is only

needed to schedule flows to run on Tableau Server. For more information, see Tableau Prep Conductor.

Process	Tableau Prep Conductor
Status	Status of the Tableau Prep Conductor process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the Tableau Prep Conductor process are located in `/var/-opt/tableau/tableau_server/data/tabsvc/logs/flow-processor`. For more information, see Tableau Server Logs and Log File Locations

Tableau Prep Conductor uses the following components to run flows:

- **Backgrounder:** Tableau Prep Conductor uses the Backgrounder process to run flows. Backgrounder is single threaded, so each instance of the Backgrounder process on a node can run one flow at a time. By adding more Backgrounders to a node, you can increase the number of flows that can be run in parallel on that node. The Backgrounder processes can be up to half the number of the physical cores of that node.
- **Connectors:** Prep Conductor uses the supported Tableau Data connectors to connect to data. For a list of supported Connectors, see Supported Connectors.
- **Data Engine:** Any changes to data or transformation steps in your flow that cannot be pushed to the underlying data source are processed using the Data Engine process. For example, SQL Server does not natively support regular expressions. When connecting to SQL Server, Tableau Prep lets you write regular expression calculations. Tableau Prep Conductor uses Data Engine to temporarily load the data and then perform the regular expression.

Performance and Scale Recommendations

- **Isolate flows to a separate node:** Running Tableau Prep Conductor on a separate node will isolate flow workflows from other Tableau workloads. This is highly

recommended since Prep flows are CPU and RAM intensive.

- **Manage flow schedules:** You can control flow execution by creating flow schedules. These schedules let you determine when flows run, how frequently they run, the priority of that schedule, and whether to run items in that schedule serially or in parallel.
- **Add resources:** When scaling your Tableau Prep Conductor environment, we recommend scaling up to an 8 physical cores box per node running as many as 4 backgrounders on each. As you need more resources, we recommend adding more nodes to your server environment.

You can monitor user activity and performance of flows using Administrative views. For more information, see Monitor Flow Health and Performance.

Topology and Configuration

By default, Tableau Prep Conductor is automatically enabled on a node where backgrounder is enabled. If the node role is set to exclude flows, then Tableau Prep Conductor is not installed on that node. For more information, see Workload Management through Node Roles.

It is recommended that you enable Tableau Prep Conductor on a dedicated node to run flows. For more information, see the following topics:

- For new Tableau Server installations, see Step 1 (New Install): Install Tableau Server with Tableau Prep Conductor.
- To enable Tableau Prep Conductor on an existing installation of Tableau Server, see Step 1 (Existing Install): Enable Tableau Prep Conductor.

Tableau Prep Flow Authoring

The Tableau Prep Flow Authoring process provides the interactive Prep Flow experience in the browser, which lets users create and interact with flows on Tableau Server to clean and prepare data. For more information, see Create and Interact with Flows on the Web. It was added in version 2020.4.

Process	Tableau Prep Flow Authoring
Status	Status of the Tableau Prep Flow Authoring process is visible on the Status Page. For more information, see View Server Process Status
Logging	Logs generated by the Tableau Prep Flow Authoring process are located in `/var/opt/tableau/tableau_server/data/tabsvc/logs/flowed-itor`. For more information, see Tableau Server Logs and Log File Locations

Tableau Prep Flow Authoring is enabled by default. It uses these components:

- **Tableau Prep Minerva Service**: Used by Tableau Prep Flow Authoring for querying datasources. By default, it is automatically enabled on a node where Tableau Prep Flow Authoring is enabled.
- **Data Engine**: The Data Engine creates data extracts and processes queries. By default, it is automatically enabled on a node where Tableau Prep Flow Authoring is enabled.
- **Gateway**: The Gateway process handles all requests to Tableau Server from browsers, Tableau Desktop, and other clients. By default, it is automatically enabled on a node where Tableau Prep Flow Authoring is enabled.

Performance and Scale Recommendations

Tableau Prep Flow Authoring can be CPU and RAM intensive. It is recommended to monitor server and adjust your deployment as necessary by isolating flow authoring to a separate node or adding server resources.

Isolate flow authoring to a separate node

Running Tableau Prep Flow Authoring on a separate node will isolate flow authoring workflows from other Tableau workloads. If you have a busy server with many extract refreshes, and a lot of viz editing and viewing and you don't want to cause disruption to this workload, then it is recommended to isolate Prep Flow Authoring to its own server node. This means that all flow editing is directed to a dedicated node. For more information see Configure Nodes.

> **Note:** Prep Web Authoring uses shared services such as the Application Server (VizPortal), the PostgreSQL repository, the Cache Server, and Hyper. If those services are already at capacity, they might also need additional resources because of the additional Prep Web Authoring load.

If you are running Tableau Server on a multi-node cluster, you can dedicate one or more nodes for running Backgrounder. Using the Backgrounder process, you can isolate background workloads such as Tableau Prep Conductor from all your interactive workloads such as Prep Flow Authoring and VizQL Server. For more information, see Tableau Server Backgrounder Process and Workload Management through Node Roles.

For core and user-based deployments, it's strongly recommended that at least one node be dedicated to flows for best performance. While you can run flows on any licensed Server core, the additional resource cores purchases should only run Tableau Prep Conductor, and not extract refreshes or VizQL processes.

Add resources

As you need more resources, you can add more nodes to your server environment. When planning your nodes, several factors can influence your decision on how much additional hardware you need to allocate.

The main things to consider when planning your nodes are:

- The number of concurrent users or concurrent sessions you expect during peak hours. The number of concurrent sessions per user can be set using the TSM option `maestro.sessionmanagement.maxConcurrentSessionPerUser`. For more information, see tsm configuration set Options.
- The number of data input nodes your flows have on average and the amount of data they have. The maximum row sampling limit can be set using the TSM option `maestro.app_settings.sampling_max_row_limit`. For more information, see tsm configuration set Options.

- The complexity of the flows being authored and the number of nodes. Node types like joins, unions, aggregates, and pivots will, in general, require more resources.

License additional offerings

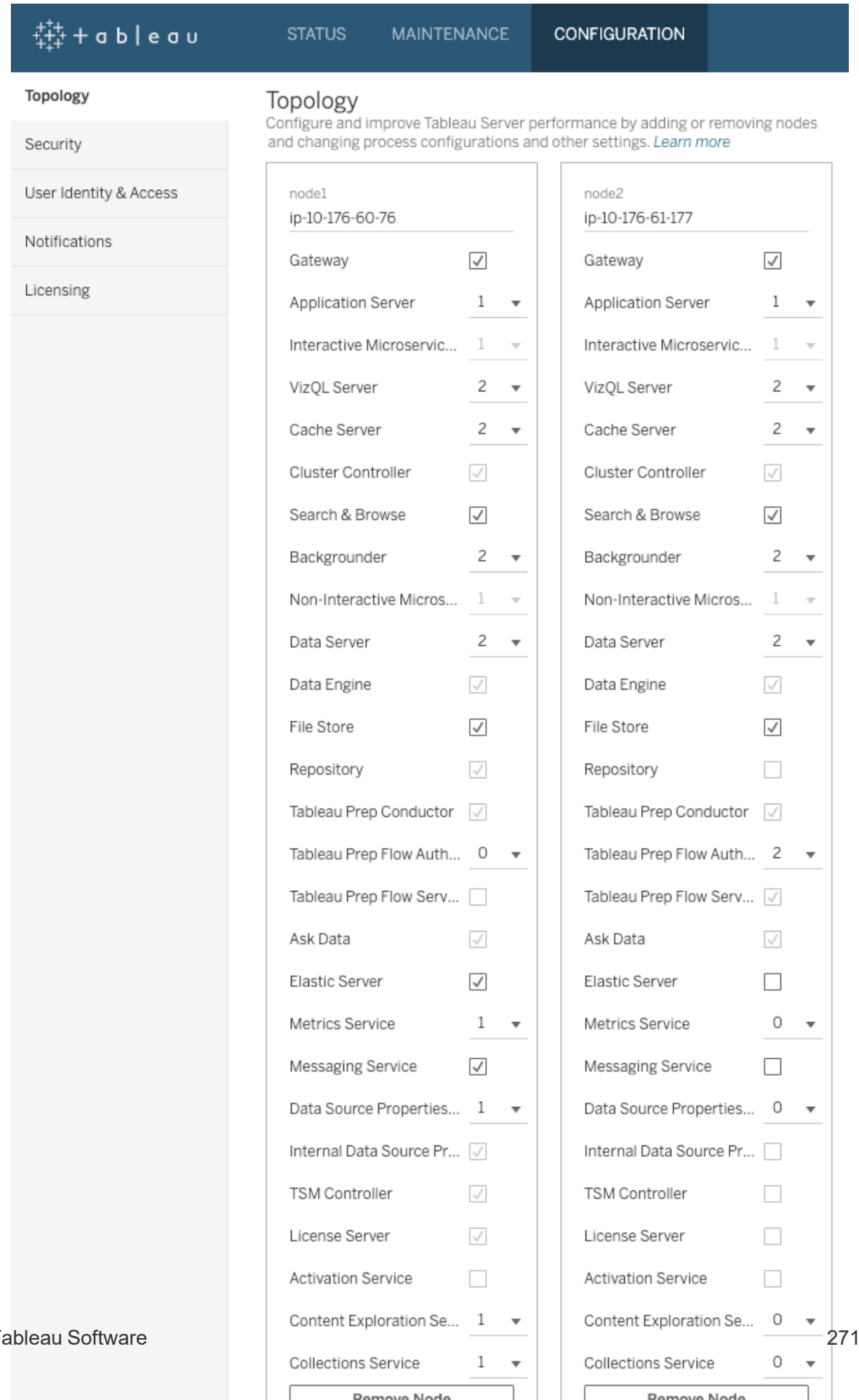
Additional licenses for Data Management and Advanced Management are required for configuring nodes.

OFFERING	Allows you to:
Data Management	Configure a node to run only flows, or configure a node to run all jobs except flows. Tableau Prep Conductor must be running on the node. The Data Management license includes Tableau Prep Conductor, which enables you to schedule and track flows. The license is for a single Tableau Server deployment, which can be role-based or core-based. As a Creator, Data Management is not required to create and edit flows directly on your server.
Advanced Management	Configure where the different types of workloads are processed through node rules. For example, you can run flows on one node and subscriptions and alerts on another node.

Topology and Configuration

To make Tableau Prep Flow Authoring highly available, configure two instances of Tableau Prep Flow Authoring on nodes when enabling it.

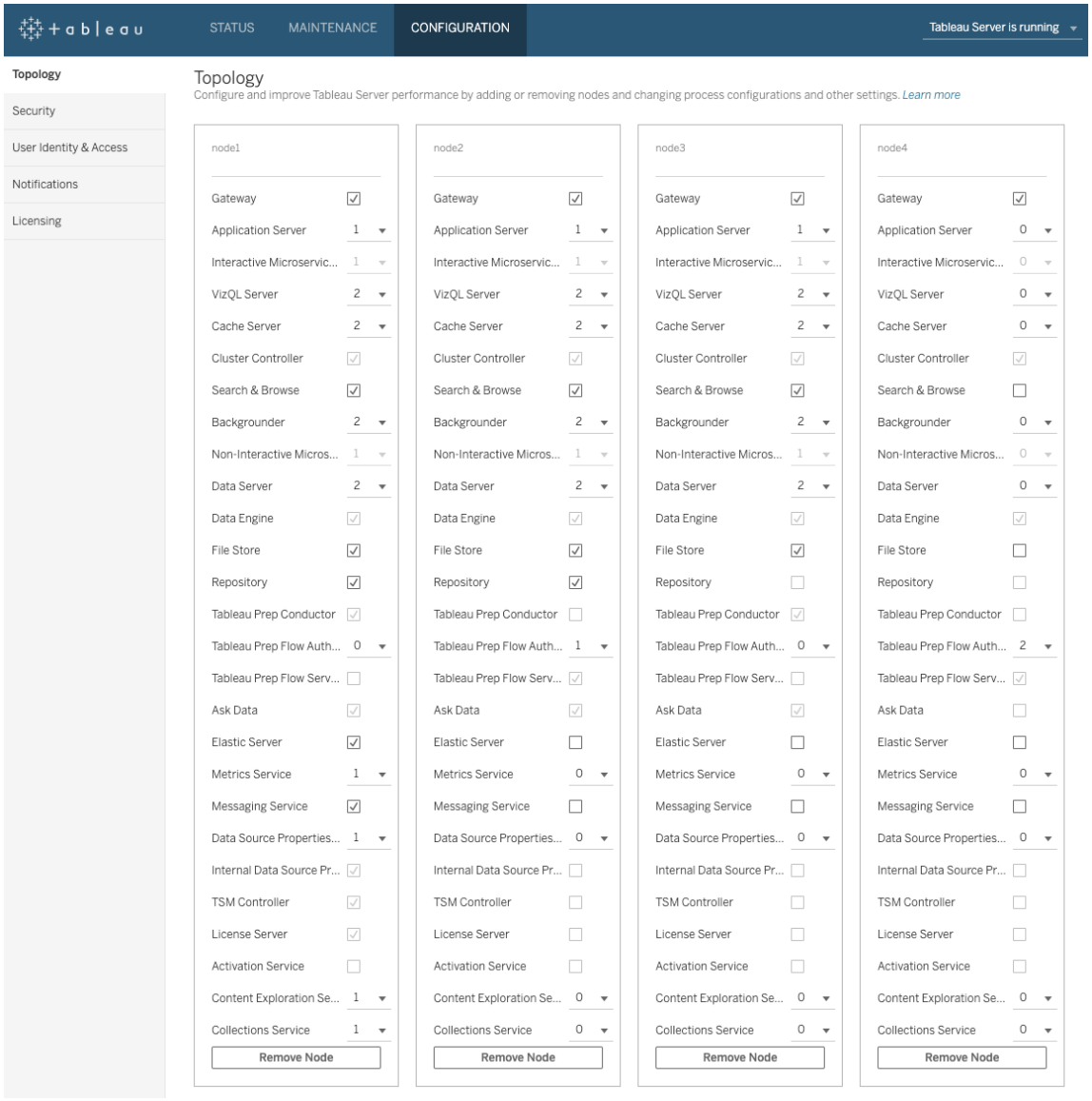
Here is an example of a two node configuration:


+ a b | e a u
STATUS
MAINTENANCE
CONFIGURATION
Topology
Security
User Identity & Access
Notifications
Licensing
Topology
Configure and improve Tableau Server performance by adding or removing nodes and changing process configurations and other settings. Learn more
node1
ip-10-176-60-76
Gateway
Application Server 1
Interactive Microservic... 1
VizQL Server 2
Cache Server 2
Cluster Controller
Search & Browse
Backgrounder 2
Non-Interactive Micros... 1
Data Server 2
Data Engine
File Store
Repository
Tableau Prep Conductor
Tableau Prep Flow Auth... 0
Tableau Prep Flow Serv...
Ask Data
Elastic Server
Metrics Service 1
Messaging Service
Data Source Properties... 1
Internal Data Source Pr...
TSM Controller
License Server
Activation Service
Content Exploration Se... 1
Collections Service 1
node2
ip-10-176-61-177
Gateway
Application Server 1
Interactive Microservic... 1
VizQL Server 2
Cache Server 2
Cluster Controller
Search & Browse
Backgrounder 2
Non-Interactive Micros... 1
Data Server 2
Data Engine
File Store
Repository
Tableau Prep Conductor
Tableau Prep Flow Auth... 2
Tableau Prep Flow Serv...
Ask Data
Elastic Server
Metrics Service 0
Messaging Service
Data Source Properties... 0
Internal Data Source Pr...
TSM Controller
License Server
Activation Service
Content Exploration Se... 0
Collections Service 0

Here is an example of status page for a two node configuration:

Process	node1 ip-10-176-60-76	node2 ip-10-176-61-177
Gateway	✓	✓
Application Server	✓	✓
Interactive Microservice Container	✓	✓
VizQL Server	✓ ✓	✓ ✓
Cache Server	✓ ✓	✓ ✓
Cluster Controller	✓	✓
Search & Browse	✓	✓
Backgrounder	✓ ✓	✓ ✓
Non-Interactive Microservice Container	✓	✓
Data Server	✓ ✓	✓ ✓
Data Engine	✓	✓
File Store	✓	✓
Repository	✓	
Tableau Prep Conductor	✓	✓
Tableau Prep Flow Authoring		✓ ✓
Tableau Prep Flow Service		✓
Ask Data	✓	✓
Elastic Server	✓	
Metrics Service	✓	
Messaging Service	✓	
Data Source Properties Service	✓	
Internal Data Source Properties Service	✓	
TSM Controller	✓	

Here is an example of a four node configuration:



Here is an example of status page for a four node configuration:

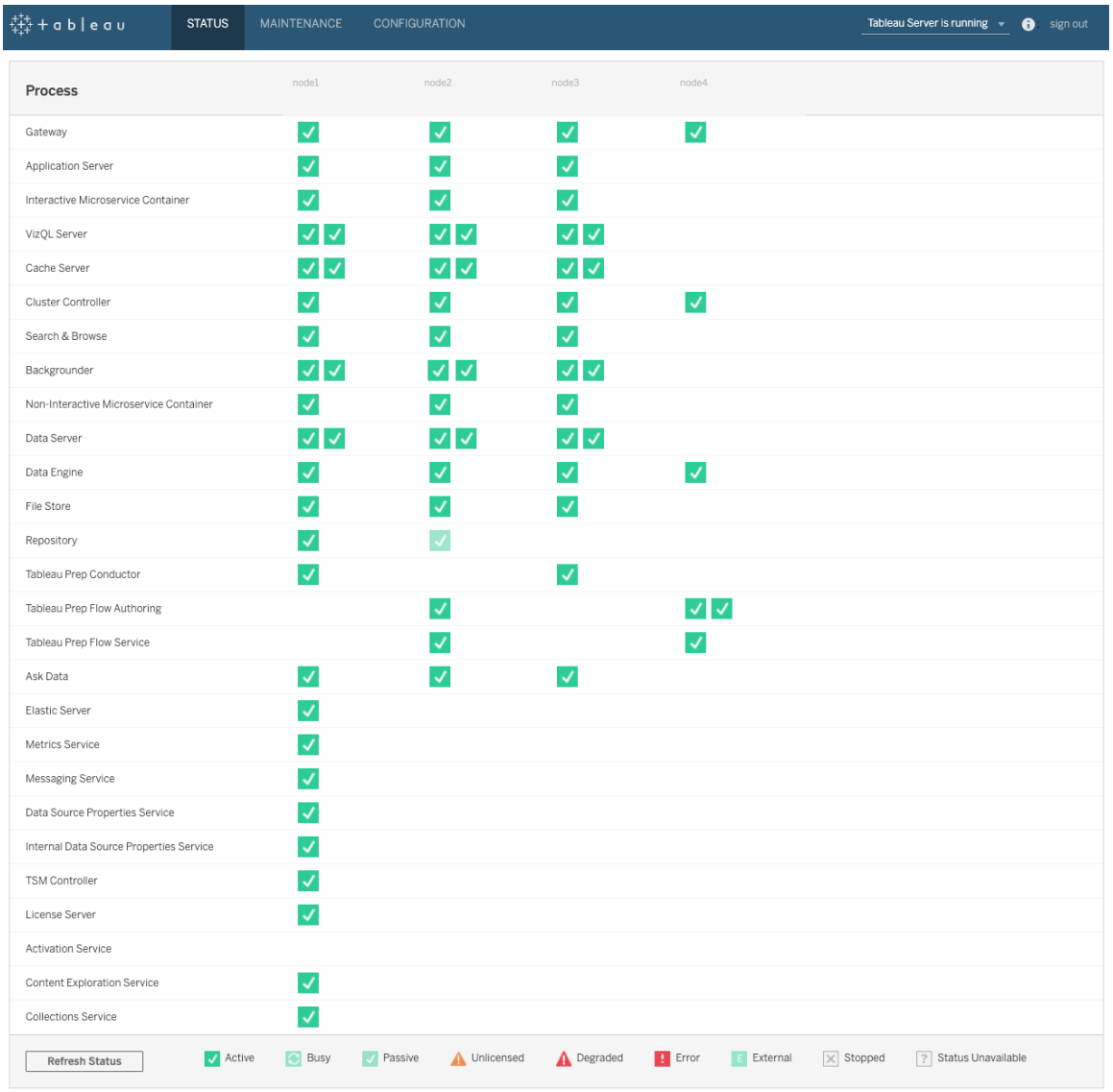


Tableau Server Dynamic Topology Changes

With the introduction of TSM, Tableau Server also introduced the ability to make certain topology changes or updates without restarting the server. These are known as dynamic topology changes, and are possible with the Backgrounder and VizQL Server processes.

You can increase or decrease the number of backgrounder or VizQL Server instances on a node without requiring a Tableau Server restart if the node already has at least one instance of the process running. You must be only changing the number of instances of Backgrounder or

VizQL Server. If you also add or remove another process, or if you are adding the first instance of Backgrounder or VizQL Server to a node or removing the last instance of either from the node, Tableau Server will require a restart.

Dynamic configuration changes

Beginning with version 2020.2.0 of Tableau Server, certain configuration changes can also be made dynamically using configuration keys. You can make dynamic topology changes at the same time you make dynamic configuration changes, without needing to restart Tableau server. For more information about dynamic configuration changes, see Tableau Server Release Notes in What's New and Changed for 2020.2 in Tableau Server.

Example Scenarios

To better understand why this might be useful, consider these examples:

- Backgrounder—At the end of a sales quarter your sales team is using Tableau Server to keep track of their numbers. Dashboards that depend on extracts are showing sales people how they are doing. Any delay in extract refreshes means your team is not seeing the most up-to-date numbers. You can add additional backgrounders to any node that already has at least one backgrounder or VizQL Server, and increase the throughput of extract refreshes, helping to guarantee the numbers are up-to-date as your team finished up their quarter. Later, after the quarterly push, you can reduce the backgrounder instances again to return Tableau Server to its original configuration.
- VizQL Server—Similarly, if Tableau Server is unable to keep up with view refreshes, you can quickly add additional VizQL Server instances to any node that already has at least one instance of either VizQL Server or backgrounder configured. In the above backgrounder example, you might want to remove VizQL Servers temporarily, to accommodate additional backgrounders, and then re-add them back before your users arrive in the morning.

Making dynamic topology changes

You can make dynamic topology changes using the TSM Web UI, or on the command line. To use the Web UI, sign in to TSM using a browser, and on the Configuration tab, update the number of backgrounder or VizQL Server instances for the node you are updating. For details, see Configure Nodes. To make your changes using the TSM CLI, at a command

prompt, run the tsm topology set-process command. For details, see Changing the number of processes on a node.

Impact of dynamic topology changes

When making dynamic topology changes that remove existing instances of VizQL Server or backgrounder, the instances are removed immediately. Be aware of the following potential impacts to users and currently running jobs:

- Backgrounder—Any currently running jobs are terminated. The normal Tableau Server retry logic will restart these jobs, using another backgrounder instance.
- VizQL Server—Any currently active sessions are terminated. Users may see an error message. Refreshing the browser should clear the error.

Best practices

Tableau recommends you test any dynamic topology changes you plan on using, before implementing them in your production environment. This will help you fully understand potential impacts to your users and scheduled refreshes and subscriptions, and allow you to most efficiently take advantage of the flexibility offered by dynamic topology.

Automating dynamic topology changes

You can automate dynamic topology changes. For example, if you have most of your extract refreshes scheduled overnight, and know your server has extra capacity because users are not signed in, you can use a script or other automated deployment tool to add backgrounder instances when they can be most efficiently used, and then remove them before the start of the work day.

To automatically get the status of processes, use the `tsm status -v` command and parse the output in your script. Alternately, you can use the TSM REST API get server status. The API is currently in alpha. For more details, see Get server status.

A sample script to set four instances of backgrounder on node2 might look like this:

```
echo Adding/Removing Processes
tsm topology set-process -pr backgrounder -n node2 -c 4
tsm pending-changes apply
echo Done!
```

Additional information

Tableau Server Processes

Server Process Limits

When you reconfigure processes for Tableau Server, there is a limit to the amount that you can increase the number of process instances. By default, the limit is set to eight. If your machine has enough RAM and CPU cores, and you want to go above this limit, you can change the limit using the `service.max_procs` configuration option. For each process instance, Tableau recommends that the machine running the process have at least 1 GB of RAM and 1 logical CPU core.

To change the maximum number of processes allowed:

1. Type the following command, where `number` is the maximum number of process instances you want to allow:

   ```
   tsm configuration set -k service.max_procs -v <number>
   ```

   For example:

   ```
   tsm configuration set -k service.max_procs -v 10
   ```

2. Next type:

   ```
   tsm pending-changes apply
   ```

   If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart

behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

Tableau Services Manager Ports

The processes and services that make up the components of Tableau Services Manager (TSM) and Tableau Server on Linux use various ports to communicate. By default, most these ports are assigned (mapped) dynamically from a predefined range of ports. The port assignments are made for each service or process when it is installed. You can control what ports are used in a variety of ways.

A small subset of processes do not use dynamic port mapping and behave uniquely. For more information, see Ports that are not dynamically mapped, below.

Note: This operation includes steps that you may need to perform using the TSM command line.

Ephemeral port use

If your operating system is configured to use ephemeral ports, your Tableau traffic may be routed through these, even when configured as described below. This happens at the OS level and is not something you can configure in Tableau. If you want to control port usage, consult your operating system documentation on how to limit ephemeral port usage.

Firewall requirements

If you are running a firewall on the computer where you will be installing Tableau Server, then you will need to open the following default ports for Tableau Server traffic. All port numbers, except 443 can be changed.

Port	TCP/UDP	Used by ...	TYPE OF INSTALLATION	
			All	Distributed / High Availability
80	TCP	Gateway	X	
443	TCP	SSL. When Tableau Server is configured for SSL, the application server redirects requests to this port. Do not change this port.	X	
8850	TCP	Tableau Services Manager.	X	
8060	TCP	PostgreSQL database.	X	
8061	TCP	PostgreSQL backup verification port	X	
8000-9000	TCP	Range of ports reserved by default for dynamic mapping of Tableau processes		X
27000-27009	TCP	Range of ports used by Tableau Server for License service. This range must be open on the node running the License service and accessible from other nodes. By default, the initial node runs the License service.	X	

See Configure Local Firewall.

Port assignment

There are two approaches you can use for port assignment or mapping in TSM:

- **Dynamic port assignment**. This is the default and requires the least intervention by the administrator.

- **Manual port assignment**. This option requires an administrator to individually assign each port.

If you change port assignments and you are running a local firewall, see Local firewall configuration.

Dynamic port assignment

You can control which ports are used by changing the range of ports available to the dynamic assignment process. Doing this leaves dynamic port assignment in place but restricts the ports that can be chosen. With dynamic port assignment in place, you can still choose to assign ports for certain processes manually. This approach enables you to assign specific ports to specific processes, while leaving the others to be mapped dynamically. Using dynamic mapping, with or without some individually assigned ports is the easiest approach, and should satisfy the requirements of most customers. Dynamically assigned ports are preserved if you export your Tableau Server configuration.

By default, ports are assigned for each service or process from available ports between 8000 to 9000. This assignment takes place when services are installed for the first time on a node. After Tableau Server is initialized, you can see which ports are being used by which services or processes by running this command:

```
tsm topology list-ports
```

If you have a multi-node cluster, ports on all nodes are listed.

Changing the port range

For organizations that have specific requirements for ports being used, the easiest way to control this is to change the range from which ports are dynamically selected. You need to do this at installation, by specifying a minimum and maximum port for the range in your configuration file.

Note: The minimum allowable size of your port range will depend on your server installation and how many services or processes you are running. As a general best practice you should not restrict the range too tightly because port assignment is done by selecting random ports within the range, and if you do not allow a large enough range, selection may fail to find an available port.

To limit the range from which available ports are chosen to those between 8300 and 8600, your configuration file would include an entry similar to this:

```
"configKeys": {
    "ports.range.min": "8300",
    "ports.range.max": "8600"
}
```

Blocking specific ports within the range

Beginning with version 2021.1.0, you can specify certain ports that are within the dynamically assigned range but should not be used by Tableau. This is useful if you have other software on your Tableau Server computers that rely on ports within the range Tableau is using.

To specify ports within the range that should not be used by TSM and Tableau Server, use the `ports.blocklist` configuration key:

```
tsm configuration set -k ports.blocklist -v <port>[,<port>,<port>]
```

For example:

```
tsm configuration set -k ports.blocklist -v 8000,8088, 8090
```

Disabling dynamic port assignment

If you need more control of port assignment than you can get through a combination of restricting port range and individually assigning ports, you can disable dynamic port mapping at initial server configuration. Disabling dynamic port mapping requires you to manually assign every port for every process, so we don't recommend this unless you need to control every single port assignment.

If you disable dynamic port mapping, you must configure the port for each process on each node of your installation.

To disable dynamic mapping, your configuration file would include an entry similar to this:

```
"configKeys": {
    "service.port_remapping.enabled": false
}
```

Important: When you disable dynamic port assignments, the License service port range is not included. This range (27000-27009) must be open on the node running the License service and accessible from other nodes. By default, the initial node runs the License service.

Manual port assignment

You can disable automatic port assignment entirely and assign a port for each process individually. If you do this, you must assign a port for every process on every node. You can assign ports either in a configuration file, when the processes are first installed, or after installation, using a TSM command. Only ports assigned at process installation are preserved if you export your Tableau Server configuration.

You can specify individual ports for specific processes, whether or not dynamic mapping is enabled. You might do this if you want a process to use a particular port, or if you've disabled dynamic mapping. There are two ways to specify ports for processes: during installation or after installation.

Configuring ports during installation

We recommend configuring port assignment during the installation process as described here. Changing ports after installation is a much more labor-intesive process.

To configure ports during installation, create a json file that specifies your port configuration. This process is similar to defining a non-default port range, but instead you specify a particular port for a specific service or process. If you are going to assign specific ports, this approach is

the most robust way to do so because the port mapping is preserved if you export the server configuration and topology settings using the `tsm settings export` command.

To define ports at installation, add information to your configuration file to specify the node (`workerN`), process (`servicename`) and instance ID (`instanceid`), port type (`porttype`), and the port to be used. The format looks like this:

```
workerN.{servicename}_{instanceid}.{porttype}.port:X
```

Where:

- `workerN` is an optional parameter and identifies the node for which the remapping applies. Node numbers start with zero (0).
  We recommend you do not include this parameter unless you need to map different ports for the same service on different nodes. If you leave this parameter off, you can map a service port on the initial node, or map the same service port on multiple nodes.
- `servicename` is the name of the process or service that will use the port.
- `instanceid` is the instance of the process. If you are going to be configuring multiple instances of a process on one node, you would need to increment this value for each instance. Start the instanceid at zero (0) and increment it by one (1) for each instance of the process. For services that only install a single instance on any given node, this must be left off.
- `porttype` If setting the primay port, do not include this option.
- `port` is the port the process or service should use.

For example, to set the port for the first instance of the file store process on the initial node to 8500, you would include a configuration file entry similar to this:

```
"configKeys": {
    "filestore_0.port: "8500"
}
```

The example above does not include the optional workerN parameter, so sets the port on all nodes in the cluster. It also leaves off the porttype option because it is setting the primary filestore port.

Important: When specifying port changes with a configuration file, you must include the `--force-keys` parameter with the `tsm settings import` command.

Configuring ports after installation

If you need to change ports after you have installed Tableau Server, use the `tsm topology set-ports` command. This approach allows you to specify a port for a specific process after that process has been installed. You are restricted by these limits:

- You must set ports individually, on each node.
- After you set an individual port, you must run `tsm restart`.
- The port assignments are not preserved if you need to import a Tableau Server configuration using `tsm settings import`.
- Port names use a different syntax for tsm commands than the syntax that is required for configKeys. The table at the end of this topic provides a syntax reference.

For example, to set second instance of the file store on the initial node to use port 8500:

```
tsm topology set-ports --node-name node1 --port-name filestore --
port-value 8500 --instance 2
```

The following example shows how to use shorthand commands to set the JMX ports:

```
tsm topology set-ports -n node1 --port-name vizqlserver:jmx.rmi -pv
9403 -i 1
```

```
tsm topology set-ports -n node1 --port-name vizqlserver:jmx -pv 9404
-i 2
```

Note: Port entries are not validated when you enter them. Therefore, if you use a port that is already assigned, or if you mistype the syntax for a command, Tableau will not give an error until you restart. After restarting, you may see a generic error, *The reconfigure async job failed.*

If you add an incorrect `portname:type` with a valid port, you cannot delete the incorrect

entry. To update the port, you must reassign an unused port to that value to free up the port again.

Ports that are not dynamically mapped

The Tableau Server repository uses two ports that are not dynamically mapped. These each have a default port that you can override using the `tsm configuration set` command and a process-specific parameter.

Port names	Port (default)	Description
pgsql.port	8060	Port for the Tableau Repository (PostgreSQL database). To override this port: `tsm configuration set -k pgsql.port -v <port>`
pgsql.verify_restore.port	8061	Port for verifying the integrity of a repository backup. to override this port: `tsm configuration set -k pgsql.verify_restore.port -v <port>`

Because these ports do not use the dynamic port mapping system, they do not show up in the output of the `tsm topology list-ports` command. To see the value of these you need to use the `tsm configuration get -k <config.value>` command. For example:

```
tsm configuration get -k pgsql.port
```

Controlling port remapping with initialize-tsm

Port assignments are made when services are installed. This means that in order to manually map ports for the TSM-specific processes, you need to assign the ports when you run the initialize-tsm script. The script includes options to specify ports for individual TSM services, as well as options for defining the minimum and maximum of the port range used with dynamic mapping, and you can disable dynamic mapping.

The table below lists the options for ports when running the initialize-tsm script.

Table: initialize-tsm script port options

Script option	Parameter	Description
-i	<port>	Sets the Coordination Service client port.
-e	<port>	Sets the Coordination Service peer port.
-m	<port>	Sets the Coordination Service leader port.
-n	<port>	Sets the TSM agent file transfer port.
-o	<port>	Sets the TSM Controller port.
-l	<min-port>	Sets the bottom of the port range used for dynamically mapping ports.
-r	<max-port>	Sets the top of the port range used for dynamically mapping ports.
--disable-port-remapping		Disables dynamic port mapping. If you do this you must assign ports for every service or process used by TSM and Tableau Server. For more information,

Script option	Parameter	Description
		see Manual port assignment above.

Dynamically mapped ports

This table lists the processes or services that use dynamically mapped ports.

Port names: syntax for json file (configKeys)	Port names: syntax for tsm CLI	Description
activemqserver.port	activemqserver:primary	ActiveMQ Service service port.
activemqserver.openwire.port	activemqserver:openwire	ActiveMQ Service openwire port.
appzookeeper_0.client.port	appzookeeper:client	Coordination Service client port.
appzookeeper_0.peer.port	appzookeeper:peer	Coordination Service peer port.
appzookeeper_0.leader.port	appzookeeper:leader	Coordination Service leader port.
backgrounder_0.port	backgrounder	Backgrounder primary port.
backgrounder_0.debug.port	backgrounder:debug	Backgrounder debug port.
backgrounder_0.jmx.port	backgrounder:jmx	Backgrounder jmx port.

Port names: syntax for json file (configKeys)	Port names: syntax for tsm CLI	Description
backgrounder_0.jmx.rmi.port	backgrounder:jmx.rmi	Backgrounder jmx rmi port.
backgrounder_0.re-commendations.trainer.port	back-grounder:recommendations.trainer	Backgrounder recom-mendations port.
backuprestore.port	backuprestore	Backup/Restore service port.
cacheserver_0.port	cacheserver	Cache server port.
clustercontroller.status.port	clustercontroller:status	Cluster Con-troller status port.
clustercontroller.storage.port	clustercontroller:storage	Cluster Con-troller storate port.
databasemaintenance.port	databasemaintenance	Database Main-tenance port.
dataserver_0.port	dataserver	Data server primary port.
dataserver_0.debug.port	dataserver:debug	Data server debug port.
dataserver_0.jmx.port	dataserver:jmx	Data server jmx port.
dataserver_0.jmx.rmi.port	dataserver:jmx.rmi	Data server jmx rmi port.
filestore.port	filestore	File store

Port names: syntax for json file (configKeys)	Port names: syntax for tsm CLI	Description
		primary port.
filestore.status.port	filestore:status	File Store status port.
gateway.port	gateway	Gateway port. This defaults to 80, and if that is not available, to 8080. If that is not available, it tries 8000. That sequence is followed whether or not dynamic port assignment is enabled or not. If none of those ports are available and dynamic mapping is enabled, it takes an available port within the defined range. The gateway port must be the same on all nodes in a multi-node cluster, so if port 80 is selected on the initial node this is the

Port names: syntax for json file (configKeys)	Port names: syntax for tsm CLI	Description
		port that will be used on all nodes and if it is unavailable on one of the other nodes, gateway port selection will fail.
hyper.port	hyper	Data engine primary port.
hyper.connection.port	hyper:connection	Data engine connection port.
indexandsearchserver.port	indexandsearchserver	Index and Search server primary port.
indexandsearchserver.transport.port	indexandsearchserver:transport	Index and Search server transport port.
licenseservice.vendor_daemon.port	licenseservice:vendor_daemon	License service vendor daemon port. Used for licensing-related communications between nodes in a multi-node installation.
samlservice.port	samlservice	SAML service port.

Port names: syntax for json file (configKeys)	Port names: syntax for tsm CLI	Description
siteimportexport.port	siteimportexport	Site Import/Export port.
tabadmincontroller.port	tabadmincontroller	TSM Controller port.
tabadminagent.columbo.port	tabadminagent:columbo	Administration Agent service discovery port
tabadminagent.filetransfer.port	tabadminagent:filetransfer	TSM Agent file transfer port.
vizportal_0.authentication.port	vizportal:authentication	Application server authentication port.
vizportal_0.authorization.port	vizportal:authorization	Application server authorization port.
vizportal_0.maintenance.port	vizportal:.maintenance	Application server maintenance port.
vizportal_0.microservice.extensions.port	vizportal:.microservice:extensions	Application server extensions port.
vizportal_0.monolith_grpc.port	vizporta:monolith_grpc	Application server GRPC port.
vizportal_0.publishing.port	vizportal:publishing	Application server publishing port.

Port names: syntax for json file (configKeys)	Port names: syntax for tsm CLI	Description
vizportal_0.re-commendations.port	vizportal:recommendations	Application server recom-mendations port.
vizportal_0.port	vizportal	Application server primary port.
vizportal_0.debug.port	vizportal:debug	Application server debug port.
vizportal_0.jmx.port	vizportal:jmx	Application server jmx port.
vizportal_0.jmx.rmi.port	vizportal:jmx.rmi	Application server jmx rmi port.
vizqlserver_0.port	vizqlserver	VizQL server primary port.
vizqlserver_0.debug.port	vizqlserver:debug	VizQL server debug port.
vizqlserver_0.jmx.port	vizqlserver:jmx	VizQL server jmx port.
vizqlserver_0.jmx.rmi.port	vizqlserver:jmx.rmi	VizQL server jmx rmi port.

Enable the JMX Ports

To help you work through a problem with Tableau Server, Tableau Support may ask you to enable the server's JMX ports. These ports can be useful for monitoring and troubleshooting, usually with a tool like JConsole. In versions 2022.1 and later, the JMX ports can be enabled securely and this is the recommended method. In versions earlier than 2022.1, you can only enable the ports unsecured.

Enable secure JMX ports

Beginning with version 2022.1 of Tableau Server, you can enable JMX ports securely. This procedure explains how to enable secure JMX. To enable JMX in earlier versions, see Enable unsecured JMX ports below.

To enable secure JMX ports on Tableau Server:

1. Log on as a user with `sudo` access to the computer where TSM is installed.
2. Run this command:

   ```
   tsm maintenance jmx enable
   ```

   The command has several options you can provide when running it. If you do not provide any options when you run the command, you will be prompted for options based on the answers you give.

   For example:

   ```
   C:\Windows\system32>tsm maintenance jmx enable
   We do not recommend you enable JMX unsecured on a production
   environment. Would you like to enable security features for
   JMX?
   (y/n): y
   JMX access is readonly by default. Would you like to add read-
   write access?
   (y/n): n
   Would you like to enable SSL?
   ```

```
(y/n): y
Would you like to require client authentication (mTLS)?
(y/n): n
Enabling JMX with the specified settings. This will perform a
server restart. Do you want to continue?
(y/n): y
Starting enable JMX asynchronous job.
```

For more information on the command and its options, see tsm maintenance jmx enable.

Enable unsecured JMX ports

For versions of Tableau Server earlier than 2022.1.0, if you are enabling JMX ports they can only be enabled unsecured.

Important Enabling unsecured JMX ports can introduce some security risk. We strongly recommend using secure JMX. If you do not have a version of Tableau Server that supports this, be aware of the risk and mitigate it by limiting access to the JMX ports to the fewest number of clients that's practical for your scenario. You typically limit access using the host's firewall rules, an external security device, or routing rules.

To enable unsecured JMX ports on Tableau Server:

1. Stop the server.
2. Enter the following command:

   ```
   tsm configuration set -k service.jmx_enabled -v true
   ```

3. Apply pending changes:

   ```
   tsm pending-changes apply
   ```

If the pending changes require a server restart, the `pending-changes apply` command will display a prompt to let you know a restart will occur. This prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. If the changes do not require a restart, the changes are applied without a prompt. For more information, see tsm pending-changes apply.

4. Restart the Coordination Service and TSM Administration Controller (as *tableau* system account):

   ```
   sudo su -l tableau -c "systemctl --user restart appzookeeper_0.service"
   ```

   ```
   sudo su -l tableau -c "systemctl --user restart tabadmincontroller_0.service"
   ```

   It may take a few minutes for tabadmincontroller to restart. If you attempt to apply pending changes in the next step before the controller has fully restarted, TSM will not be able to connect to the controller. You can verify that the controller is running by using the `tsm status -v` command. Tableau Server Administration Controller should be listed as "is running".

5. Start the server.

How the JMX Ports Are Determined

By default, the JMX ports assigned dynamically, from within a range of available ports. For details on how port assignment is done, and how to override dynamic mapping, see Tableau Services Manager Ports.

ATRDiag.exe Command Line Reference

You can use the ATRDiag command line utility to manage licensing issues related to the login-based license management and authorization-to-run (ATR) features.

Synopsis

```
ATRDiag -dumpATR -showAll -showVerbose -product ["Tableau Desktop"]
["Tableau Prep"]["Tableau Server"]

-subProduct {subProduct} -version {version} -setVersion {version}

-container -serverDataPath {path}

-log [Trace][Debug][Info][Warn][Error][Fatal][Off] -status -
deleteAllATRs

-deleteInvalidATRs -deleteATR {atrIdToDelete}

-enableATRFeature -disableATRFeature -enableLBLMFeature -dis-
ableLBLMFeature

-requireLBLMFeature -setDuration {seconds}
```

Note: All ATRDiag options are case-insensitive.

Options

`-dumpATR`

Display a valid authorization-to-run (ATR) on the command prompt (if present). Use `-version/-product/-subProduct/-version` to control which ATR is dumped if more than one is valid.

`-showAll`

Enumerate the contents of the registry to see available ATRs.(some of which might not be valid).

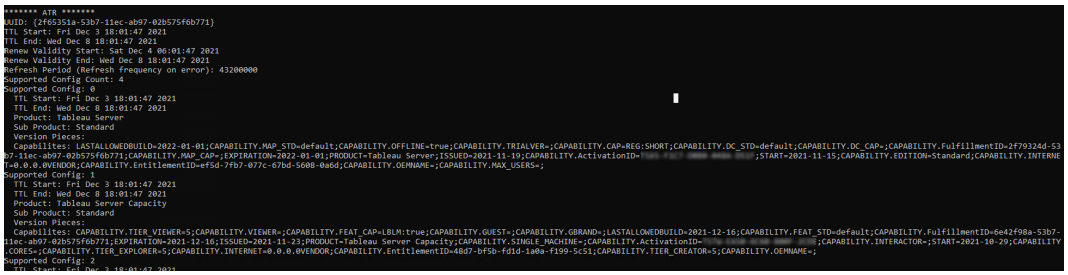
`-showVerbose`

Enumerate the contents of the registry and use `-version/-product/-sub-Product/-version` to provide per-product.

`-product ["Tableau Desktop"]["Tableau Prep"]["Tableau Server"]`

Defaults to "Tableau Desktop". Must specify "Tableau Server" to display Server ATR.

For example: `atrdiag -product "Tableau Server"`



`-subProduct {subProduct}`

Defaults to "Professional".

`-container`

Container mode, only for Tableau Server. Must specify `-product "Tableau Server"`.

`-serverDataPath`

The location of Server data under Container mode. Defaults to `"/var/-opt/tableau/tableau_server/"`.

`-version {version}`

No default; a valid value for this field is "Tableau 2021.1".

`-setVersion {version}`

Persist a default value for -version.

`-log [Trace][Debug][Info][Warn][Error][Fatal][Off]`

Display ATR log information.

`-status`

Provide ATR feature status (enabled or disabled), the license server, and dump the ATR.

`-deleteAllATRs`

Remove all ATRs present on the machine.

`-deleteInvalidATRs`

Remove all invalid ATRs.

`-deleteATR {atrIdToDelete}`

Remove an ATR by ID.

`-enableATRFeature`

Turn on the ATR feature. Must run as an administrator. For use on Tableau Desktop only.

`-disableATRFeature`

Turn off the ATR feature. Must run as an administrator. For use on Tableau Desktop only.

`-enableLBLMFeature`

Turn on login-based license management (LBLM). Must run as an administrator.

`-disableLBLMFeature`

Turn off login-based license management (LBLM). Must run as an administrator.

`-requireLBLMFeature`

Set the login-based license management (LBLM) feature to required. Must run as an administrator.

`-setDuration {seconds}`

Set `ATRRequestedDurationSeconds` to seconds. Must run as an administrator.

Global Options

`-h, --help`

Optional.

Show the command help.

Help Output for initialize-tsm Script

The following help content is the output when you run the following command:

```
sudo ./initialize-tsm -h
```

The initialize-tsm script is installed to `/opt/tableau/tableau_server-/packages/scripts.<version_code>/.`

Output

```
REQUIRED
--accepteula           Indicate that you have accepted the End User
License Agreement (EULA).
                       You can find the EULA in /opt/tableau/tableau_server-
/packages/docs.<version_code>
```

```
OPTIONAL
-c config-name          Set the service configuration name.
                        If not set, the default is "tabsvc".

-d data-directory       Set a custom location for the data directory
                        if it's not already set. If not set, the default is
                        "/var/opt/tableau/tableau_server".

-b bootstrap-file       Optional. Location of the bootstrap file down-
loaded from the Tableau Services Manager
                        on existing node. Must be provided to join existing Tableau
Server cluster.

-u username             Name of the user with admin privileges on
existing Tableau Services Manager.
                        Required if -b option specified.

-p password             Password for the Tableau Services Manager
admin user.
Note: This option was removed beginning in version 2021.3.0 to
improve script security.

-f                      Bypass warning messages.

-g                      Do NOT add the current user to the "tsmadmin"
administrative
                        group, used for default access to Tableau Services Manager,
                        to the "tableau" group, used for easier access to log files.

-a username             The provided username will be used as the
user to be added
                        to the appropriate groups, instead of the user running this
                        script. Providing both -a and -g is not allowed.

-q                              Quiet, suppress output except for errors and warn-
ings.
```

```
-i coordinationservice-client-port      Client port for the coordin-
ation service

-e coordinationservice-peer-port        Peer port for the coordin-
ation service

-m coordinationservice-leader-port      Leader port for the coordin-
ation service

-t licenseservice-vendordaemon-port     Vendor daemon port for the
licensing service

-n agent-filetransfer-port              Filetransfer port for the
agent service

-o controller-port                      Https port for the con-
troller service

-l port-range-min                       Lower port bound for auto-
matic selection

-r port-range-max                       Upper port bound for auto-
matic selection

--disable-port-remapping                Disable automatic port
selection

--unprivileged-user=<value>            Name of the unprivileged account
to run Tableau Server.
Default: "tableau".

--tsm-authorized-group=<value>         Name of the group(s) that allows
authorization to access Tableau Services Manager.
Default: "tsmadmin".
```

```
--disable-account-creation                    Do not create groups or user
accounts for Server and TSM authorization.
                                        However, the values in: unprivileged-user an
group will still be used
                                        in TSM configuration.

--http_proxy=<value>                     Http forward proxy for
Tableau Server. Its value should be http://<proxy_address>:<proxy_
port>/
                                         For example, --http_proxy=http://1.2.3.4:31
y=http://example.com:3128/

--https_proxy=<value>                    Https forward proxy for
Tableau Server. Its value should be http://<proxy_address>:<proxy_
port>/
                                                                    For example, --https
proxy=http://example.com:3128/
                                                                    Take care to use htt
https_proxy environmental variable.
                                                                    Do not specify the h
https_proxy environmental variable.

--no_proxy=<value>                       Environment variable that directs certain UR
to bypass the forward proxy. For example,
                                        --no_proxy=localhost,127.0.0.1,localaddress,

--[no-]activation-service                Specify whether the Tableau
authorization-to-run (ATR) service should be used to activate
Tableau Server.
                                        This option is ideal for cloud-based or virt
but is available to anyone who
                                        can activate their copy of Tableau Server on
for product activation is a permanent
                                        choice that cannot be undone later. If no op
the Tableau authorization-to-run (ATR) service will be used to activ-
ate Tableau Server.
```

Related topics

- Controlling port remapping with initialize-tsm
- Install and Initialize TSM
- System user and groups
- Data directory

Help Output for upgrade-tsm Script

The following help content is the output when you run the following command:

```
sudo ./upgrade-tsm -h
```

The upgrade-tsm script is installed to `/opt/tableau/tableau_server-/packages/scripts.<version_code>/`.

Output

```
Usage: upgrade-tsm --accepteula [optional arguments]

Upgrade Tableau Server cluster to version <version number>.
This script should be run from any Tableau Server cluster node
after Tableau Server <version number> package
has been installed on all nodes.

REQUIRED
--accepteula                              Indicate that you have
accepted the End User License Agreement (EULA).
                                        You can find the EULA in docs directory

-u <value>, --username=<value>            TSM administrator user
name. Required if it is run using a non-TSM administrator
```

```
                                        account on the initial node, or if upgrading
prior to 2019.2.
-p <value>, --password=<value>          TSM administrator password.
Required if the --username option is specified.
                                        If a password is required but not provided,
ted for the password.
OPTIONAL
--debug                                  Print each command as it is
run for debugging purposes. Produces extensive
output.
--trust-admin-controller-cert           Do not validate the server
certificate.
--no-prompt                              Suppress script prompts. You
will only be prompted for missing required
                                        parameters, for example, if you specify a us
password. If the
                                        script needs to stop or restart Tableau Serv
without warning or
                                        prompting. Use this if you automate the upgr
with a script.

--external-repository-config-file=filename
                                        Required if upgrading from a Tableau Server
Tableau Server
                                        that uses a later major version of PostgreSQ
to use an
                                        external repository. The filename is a confi
describing a new
                                        instance of the external repository. The new
use the same
                                        type of external service as the current exte
but with the
                                        supported version of PostgreSQL.

--external-repository-cert-file=filename
                                        Required if upgrading from a Tableau Server
```

```
Tableau Server                                  that uses a later major version of PostgreSQ
figured to use an                               external repository. The filename is an SSL
file for the new                                external repository. The new repository shou
type                                            of external service as the current external
with the                                        supported version of PostgreSQL.
```

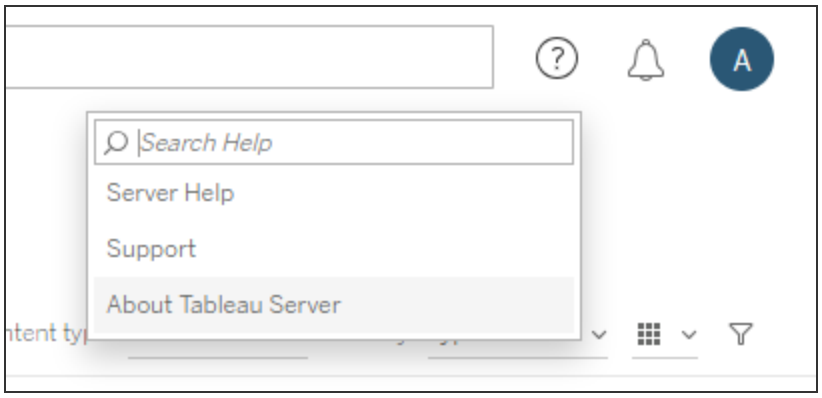
View Server Version

The version of Tableau Server you are running is important. It determines the functionality and capabilities you have access to. Version is also important when you are upgrading, because in some cases how you upgrade depends on which version you are upgrading from, and which version you are upgrading to. Knowing your version is easy, once you understand how to find it.

- Viewing version in Tableau Server—All server users can view the version of Tableau Server from the Help menu in the server web UI.
- Viewing version in Tableau Services Manager (TSM)—TSM administrators can view the versions of TSM and Tableau Server from the TSM command line (CLI).

Viewing the server version from the Tableau Server web UI

- While logged into Tableau Server, click the information icon () and **About Tableau Server**.



The version of Tableau Server is listed in the About Tableau Server dialog box:

Viewing the server version and TSM version from the TSM command line

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).
2. Run the following command:

```
tsm version
```

The output displays the versions of Tableau Services Manager (TSM) and Tableau Server.

For example:

```
C:\>tsm version
Tableau Services Manager command line version 20182.18.1214.0751.
Tableau Server version 20182.18.1214.0751.
```

Short version, long version, and version_code

In most cases, when you need to know your version number, you need to know the "short" version. This version number displays in the About Tableau Server dialog box and is made up of three parts: major, minor, and maintenance versions. The short version number has this format: `nnnn.n.n`. For example: `2018.2.5`.

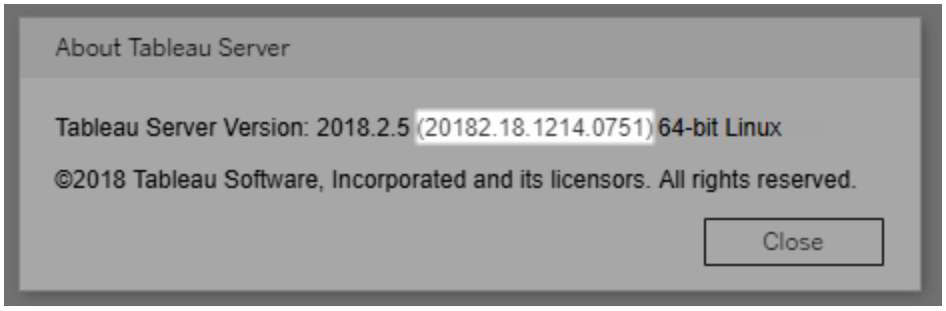
A longer version includes the major and minor version numbers, as well as other build-specific information, and has this format: `nnnnn.nn.nnnn.nnnn`, for example: 20182.18.1214.0751. When you see a reference to `<version_code>` in this documentation, it means this longer version. The most common reference to this `version_code` or long version in the documentation is when we are discussing install locations or paths to the scripts or bin directories.

Tableau Server is installed in a `/packages` directory, with subdirectories that include the long version in the path name. This means that when you are navigating to a location within the `/packages` directory, you need to include the long version. For example, the `/scripts` directory: `/opt/tableau/tableau_server/packages/scripts.<version_code>/`.

For example: `/opt/tableau/tableau server-/packages/scripts.20182.18.1214.0751`

Finding the long version number

This longer version also displays in the About Tableau Server dialog box, in parentheses:



Configure Einstein Discovery Integration

Beginning with version 2021.1.0, Tableau Server supports integration with Einstein Discovery, making Einstein Discovery predictions available to authors and viewers of workbooks and dashboards. Starting in version 2021.2.0, Einstein Discovery predictions is also now available when authoring flows on the web.

Einstein Discovery in Tableau is powered by salesforce.com. Consult your agreement with salesforce.com for applicable terms.

To integrate Einstein Discovery with Tableau Server, there are several necessary configuration steps, including some in Tableau Server, and some in the Salesforce org running Einstein Discovery. This overview outlines these steps for Dashboard extensions, Analytics extensions, and Tableau Prep extensions, and provides links to specific topics with steps for completing the server configuration.

For details on how to use Einstein Discovery predictions in Tableau, including licensing and permission requirements, see Integrate Einstein Discovery Predictions in Tableau in the Tableau Desktop and Web Authoring Help. For information about adding predictions in flows, see Add Einstein Discovery Predictions to your flow.

Einstein Discovery dashboard extensions

The Einstein Discovery dashboard extension allow workbook authors to surface real-time predictions in Tableau. The dashboard extension delivers predictions interactively, on-demand, using source data in a Tableau workbook and an Einstein Discovery-powered model deployed in Salesforce.

To configure Tableau Server for the Einstein Discovery dashboard extension you need to do the following:

1. In Tableau Server:
    a. Enable saved OAuth tokens for data connections and extensions in Tableau Server. Allow Saved Access Tokens

   b. Enable Dashboard extensions for the server. See: Manage Dashboard and Viz Extensions in Tableau Server
2. In Salesforce, in the organization running Einstein Discovery:
   a. Configure CORS in Salesforce.com for Einstein Discover Integration in Tableau Server.
   b. In Salesforce, in the organization running Tableau CRM, create a connected app. See Step 1: Create a Salesforce connected app.
3. In Tableau Server, configure server for saved SF OAuth credentials using information from the connected app. Step 2: Configure Tableau Server for Salesforce.com OAuth

Einstein Discovery analytics extensions

The Einstein Discovery analytics extension gives your users the ability to embed predictions directly in Tableau calculated fields. A table calc script requests predictions from a model deployed in Salesforce by passing its associated prediction ID and input data that the model requires. Use Model Manager in Salesforce to auto-generate a Tableau table calculation script, and then paste that script into a calculated field for use in a Tableau workbook.

To configure Tableau Server for either the Einstein Discovery analytics extension you need to do the following:

1. In Tableau Server:
   a. Enable saved OAuth tokens for data connections and extensions in Tableau Server. Allow Saved Access Tokens
   b. Enable analytics extensions for the server and configure a connection type. See: Configure Connections with Analytics Extensions
2. In Salesforce, in the organization running Einstein Discovery, create a connected app. See Step 1: Create a Salesforce connected app.
3. In Tableau Server, configure server for saved SF OAuth credentials using information from the connected app. Step 2: Configure Tableau Server for Salesforce.com OAuth

Einstein Discovery Tableau Prep extensions

Supported in Tableau Server and Tableau Cloud starting in version 2021.2.0

The Einstein Discovery Tableau Prep extension enables users to embed Einstein predictions directly in their flows when authoring flows on the web.

To configure Tableau Server or Tableau Cloud for the Einstein Discovery Tableau Prep extension you need to do the following:

1. In Tableau Server:
    a. Enable saved OAuth tokens for data connections and extensions in Tableau Server. See Allow Saved Access Tokens
    b. Enable Tableau Prep Extensions for the server. See Enable Tableau Prep Extensions.
2. In Salesforce, in the organization running Einstein Discovery, create a connected app. See Step 1: Create a Salesforce connected app.
3. In Tableau Server, configure server for saved SF OAuth credentials using information from the connected app. Step 2: Configure Tableau Server for Salesforce.com OAuth

Configure CORS in Salesforce.com for Einstein Discover Integration in Tableau Server

In version 2021.1.0 the ability to integrate Einstein Discovery predictions into Tableau Dashboards was added. You can do this using the Einstein Discovery dashboard extension. A prerequisite for this is configuring Cross-Origin Resource Sharing (CORS) in the Salesforce org that hosts Tableau CRM and includes the model and predictions that are going to be used.

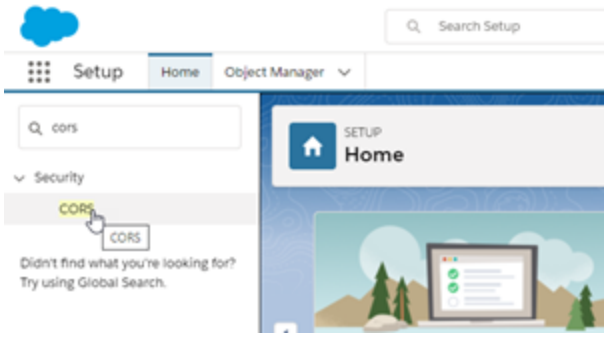
This procedure explains how an administrator in a Salesforce.com organization would do this configuration. You can find more information about CORS in the Salesforce documentation, Configure Salesforce CORS Allowlist.

Configure CORS for Einstein Discovery.

> **Note:** This procedure documents the process in Salesforce Lightning. If you are using the traditional interface, the navigation may be different but the configuration is the same.

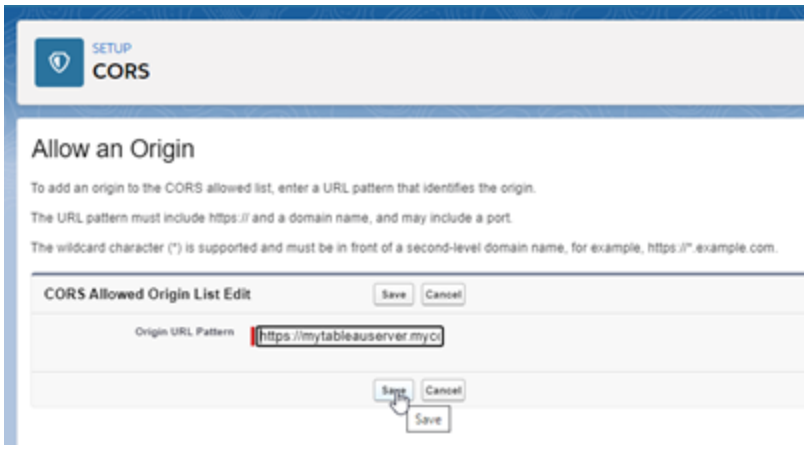
1. Sign in to your Salesforce.com developer account, click your user name in the upper-right, and then select **Setup**.

2. In the left navigation column, search for "cors" and select **CORS** .



3. In **CORS**, in the **Allowed Origins List** section, click **New**.

4. In **CORS Allowed Origin List Edit**, enter the URL of your Tableau server, beginning with "https://".



For more information about the URL pattern, see the Salesforce developer documentation: https://developer.salesforce.com/docs/atlas.en-us.chatterapi.meta/chatterapi/extend_code_cors.htm

5. Click **Save**.

Configure Connections with Analytics Extensions

Tableau supports a set of functions that your users can use to pass expressions to analytics extensions for integration with R, Python, and Einstein Discovery.

> **Note**: You can use R and Python scripts to perform complex cleaning operations in your Tableau Prep flows, but configuration and functionality supported can be different. For information see Use R and Python Scripts in your Flow in the Tableau Prep help.

This topic describes how to configure sites on Tableau Server with analytics extensions.

Because Tableau Server provides an authentication mechanism, it can be more secure to expose analytics extensions functionality to users through Tableau Server than in Tableau Desktop.

For more information about user scenarios and configuring Tableau Desktop, see Pass Expressions Analytics Extensions, in the *Tableau Desktop and Web Authoring Help*.

The configuration steps in this article are specific to Workbooks. For information about how you can use R and Python scripts to incorporate predictive modeling data into your flow, see Use R and Python scripts in your flow in the *Tableau Prep Help*.

Feature change history:

- 2021.2 — You can configure multiple analytics extension connections for each site. (You are limited to a single Einstein Discovery connection per site.)

  For information about how to determine analytics extension usage in workbooks, see Determining analytics extensions usage.
- 2021.1 — Einstein Discover is included as an analytics extension option. Einstein Discovery in Tableau is powered by salesforce.com. Consult your agreement with salesforce.com for applicable terms.
- 2020.2 — You can configure a different analytics extension connection for each site on your server. Prior to this change a single analytics extension configuration applied globally to all sites on the server.
- 2020.1 — This functionality is now called *analytics extensions*. Previously the feature was called "external services."

Server SSL

To configure SSL for analytics extensions, you must install a valid certificate on the computer running Tableau Server. The certificate must be trusted by the computer running Tableau Server. The certificate Subject field or one of the SAN entries on must exactly match the URI of the analytics extensions service configuration.

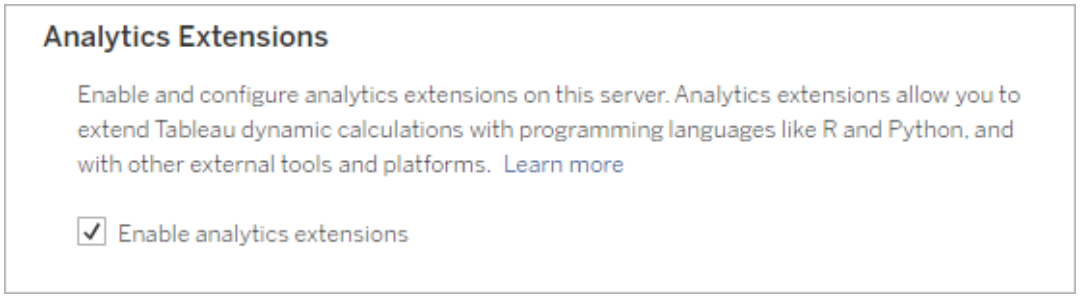
Enable analytics extensions

Before you configure extensions, you must enable analytics extensions server-wide.

1. Sign in to the Tableau Server Admin Area.
   - If you only have a single site (default) on your server, click **Settings**, and then go to Step 2.

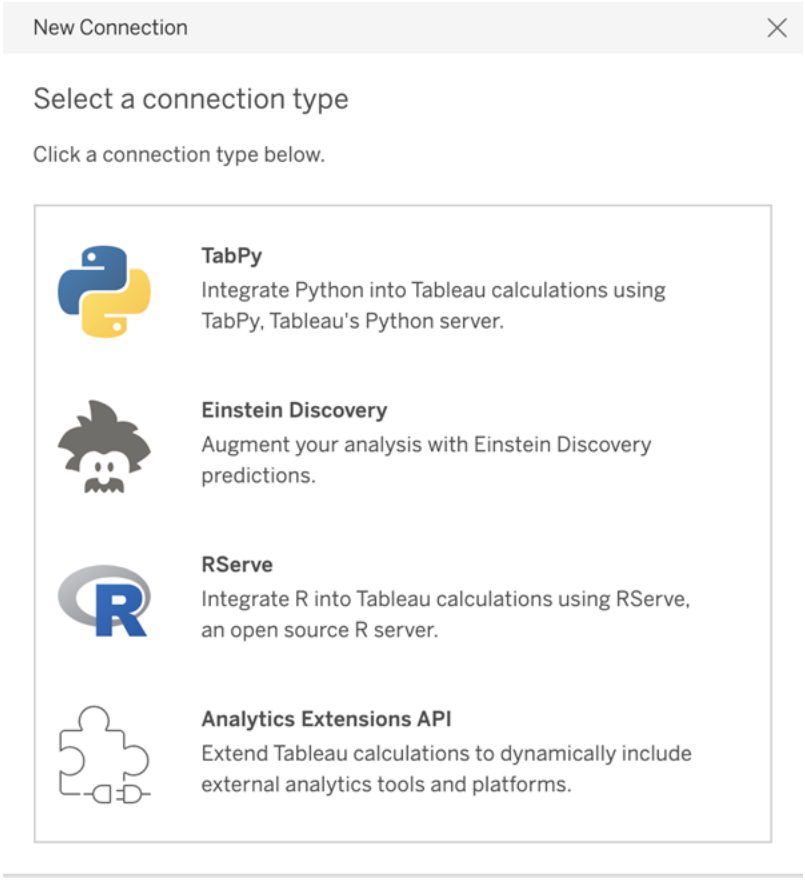
- If you have multiple sites on your server:
  a. Under **All Sites**, click **Manage all sites**.
  b. Click the **Extensions** tab.

2. Scroll to **Analytics Extensions**, select **Enable analytics extensions**, and then click **Save**.



Configure analytics extensions settings

1. Sign in to the Tableau Server Admin Area.
2. On the Settings page, click the **Extensions** tab and then scroll to **Analytics Extensions**. (On multi-site deployments of Tableau Server, navigate to the site where you want to configure analytics extensions, and then click **Settings**>**Extensions**.)
3. **Multi-site deployments only**: you must enable Analytics Extensions on each site. Under Analytics Extensions, select **Enable analytics extensions for site**.
4. Under Analytics Extensions, click **Create new connection**.
5. In the **New Connection** dialog, click the connection type you want to add, then enter the configuration settings for your analytics service:



6. The options you need to configure depend on the connection type you choose:

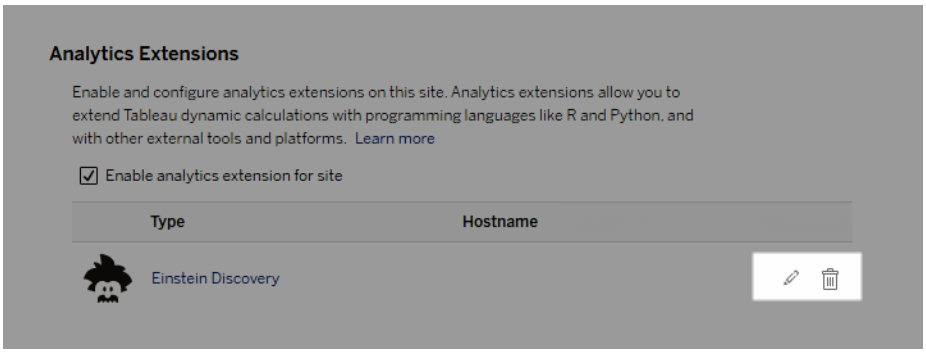
    - For an Einstein Discovery connection, click **Enable**.
    - For TabPy, RServer, and Analytics Extensions API connections, enter the following information:
        - **Connection Name** (Required): Specify the server type you are connecting to. RSERVE supports connections to R using the RServe package. TABPY supports connections to Python using TabPy, or to other analytics extensions.
        - **Require SSL** (Recommended): Select this option to encrypt the connection to the analytics service. If you specify a `HTTPS` URL in the **Hostname** field, then you must select this option.
        - **Hostname** (Required): Specify the computer name or URL where the analytics service is running. This field is case sensitive.
        - **Port** (Required): Specify the port for the service.

- **Sign in with a username and password** (Recommended): Select this option to specify user name and password that is used to authenticate to the analytics service.

7. Click **Create**.

Edit or delete an analytics extension connection

To edit or delete a configuration, navigate to **Analytics Extensions** on the **Extensions** tab of your site.



Click the **Edit** or **Delete** icon and follow the prompts to change the configuration.

Client requirement: Intermediate certificate chain for Rserve external service

As of Tableau Server version 2020.1, you must install a full certificate chain on Tableau Desktop computers (Windows and Mac) that are connecting to a Rserve external connection through Tableau Server. This requirement is due to how Rserve manages the handshake on secure connections.

Importing a root certificate on the Tableau Desktop is not sufficient, the entire certificate chain must be imported onto the client computer.

Script errors

Tableau cannot verify that workbooks that use an analytics extension will render properly on Tableau Server. There might be scenarios where a required statistical library is available on a

user's computer but not on the analytics extension instance that Tableau Server is using.

A warning will be displayed when you publish a workbook if it contains views that use an analytics extension.

> This worksheet contains external service scripts, which cannot be viewed on the target platform until the administrator configures an external service connection.

Determining analytics extensions usage

Beginning with version 2021.2, analytics extensions configurations are mapped at the workbook level. This allows administrators to use custom views to query the Tableau Repository and determine which workbooks are using which extensions, and how often they are used.

To do this you need to join the workbook connections table to tables showing workbook usage. For details about creating and using custom administrative views, see Collect Data with the Tableau Server Repository and Create Custom Administrative Views.

Changing the Identity Store

Infrastructure or business changes may require you to change the identity store on Tableau Server. There are two kinds of identity stores: local and external. When you installed Tableau Server you configured either a local identity store or an external identity store.

When you configure Tableau Server with a local identity store, all user and group information is stored and managed in the Tableau Server repository. In the local identity store scenario, there is no external source for users and groups.

When you configure Tableau Server with an external store, all user and group information is stored and managed by an external directory service. Tableau Server must synchronize with the external identity store so that local copies of the users and groups exist in the Tableau Server repository, but the external identity store is the authoritative source for all user and group data. Examples of external identity stores are OpenLDAP and Active Directory.

For more information about the Tableau identity store, see Identity Store.

You can change from local store to an external store, or you can change from an external store to a local store. In either case, to change the identity store type, you complete these steps:

1. Uninstall and then reinstall of Tableau Server. The procedure for full uninstall and clean install are at the end of this topic.
2. Restore content and permissions.

   In these steps the term "restore" does not refer to using the `TSM maintenance restore` command to restore the backup you are making. You cannot restore a backup (`.tsbak`) created on a Tableau Server instance that uses a different identity store than the target Tableau Server. The backup is a best practice safeguard, in case you need to go back to your original Tableau Server configuration.

Warning

Changing the installation type on Tableau Server can be a complicated and time-consuming process. To avoid data loss or orphaning of content or users, you'll need to plan this process carefully. In all cases, user filters applied to workbooks and data sources will need to be updated manually after the change.

Most importantly, determine how you will transition content and permissions to the new identity store after you reinstall Tableau Server.

Methods for restoring content and permissions

The following list describes two methods for restoring content and permissions after you reinstall Tableau Server. Select the method that best fits with your environmental requirements.

- **Method 1: Use site export and import**—In this method, you start by exporting each site in your existing deployment. Then, you install the new server and configure it for the new identity store type. You then create new users in the default site on the new server. Finally, you import all the original sites. During the import stage, you can map the

original identities to the new users that you created in the default site.

> **Note:** When migrating sites between instances of Tableau Server, the target site must be on a version of Tableau Server that is the equal to or later than the version of Tableau Server for the source site. Both the source and target sites must be from supported versions of Tableau Server.

Because this method exports all content and permissions at each site, it is the best method for organizations that require a high fidelity replica of the content and permissions after the identity store change is complete. Some organizations require an identity store change as the result of an authentication change. In these cases, a different user name syntax is a often a requirement in the new model. This method, which includes a process of mapping original user names to new names, provides flexibility for such scenarios.

- **Method 2: Fresh installation; users republish content**—In this method, you install a new version of Tableau Server and select the new identity store type during setup. You also create new sites. You then create users and give them access, and they republish their workbooks and data sources. Unlike the other method, in this one, you do not reuse any of your existing Tableau Server infrastructure.

  This method is most appropriate for smaller deployments with fairly autonomous and data savvy users. From an administrative perspective, this method is the simplest, since you're not actively porting over content. However, because you rely entirely on users to republish content, this method may not be successful for large organizations or for those where centralized oversight of content is required.

User filters

User filters are domain-specific. Therefore, when the domain of Tableau Server changes or authentication type changes, filters no longer function as expected. Although the user filters are generated by Tableau Server, after they are set by the user, the filters are stored in the

workbooks and data sources. Neither of these methods for changing the identity store modifies the contents of the workbooks or data sources.

As you plan the identity store change, you must also include a final task to correct user filtering in all workbooks and data sources with Tableau Desktop.

User names and the Tableau Identity store

If you are using Method 1, it's helpful to understand how Tableau Server stores user names in the Tableau identity store. Tableau stores all user identities in the repository, which coordinates content permissions and site membership with various services in Tableau Server. Generally, an identity store configured for Active Directory store user names in the format, `domain\username`. Some organizations use a UPN (`jsmith@domain.lan`).

On the other hand, organizations that configure Tableau Server with local identity store usually create standard, truncated user names, such as `jsmith`.

In all cases, these user names are literal strings that must be unique in the Tableau identity store. If you are changing from one identity store type to another, then your target authentication, SSO, or user provisioning solution may require a specific user name format.

Therefore, to maintain all permissions, content, and user viability, one of the following must be true after you change the identity store type:

- The new user names must match the original user names, or
- The original user names must be updated to match a new format.

If an authentication change is driving the identity store change, then the target authentication scheme will likely impose a user name syntax that is different than your original user names. Method 1 includes a process where you can map original user names to new user names.

It's possible that the original user name format will work with the new authentication type. For example, if you used UPN names in a local identity store deployment, you might be able to use the same user names in an Active Directory deployment. You could also use the `domain\username` format for local identity store, as long as users continue to use that format to sign in to Tableau Server.

If you are changing from local identity store to an external Active Directory store, review the topic, User Management in Deployments with External Identity Stores, as part of your planning process.

Method 1: Use site export and import

You must use the same version of Tableau Server for the export and import operations.

1. Export all sites on your server. See Export or Import a Site.
2. Back up, remove, and then reinstall .
3. Create new users on Tableau Server. You should have a new user that corresponds to each user on the original server.
4. Import the sites that you exported in Step 1. See Export or Import a Site. During import, you will be prompted to map the new users to the original users.

Method 2: Fresh installation–users republish content

Even if you do not plan to port content as part of your identity store change, we recommend that you back up the server.

1. Back up, remove, and then reinstall .
2. Create users, sites, and groups.
3. Inform your users of the new Tableau Server, provide them with credentials, and allow them to republish their content.

Back up, remove, and then reinstall

Both methods include the following steps:

1. Back up Tableau Server
2. Remove Tableau Server.
3. Reinstall Tableau Server with the new identity store type.

Step 1: Back up Tableau Server

As a best practice, you should back up the server before proceeding.

Follow the procedure, Create a backup using the TSM command line interface (CLI). Run the `backup` command with the `-d` option. The `-d` option adds the datestamp.

When you are finished, copy the backup file (`.tsbak`) to a safe location that is not a part of your Tableau Server installation.

Step 2: Remove Tableau Server

You must completely remove Tableau Server from the computer. See Remove Tableau Server from Your Computer.

Step 3: Reinstall Tableau Server with new authentication type

1. Go to the Tableau Customer Portal, sign in with your Tableau user name and password, and then download Tableau Server.
2. Install Tableau Server. See Install and Configure Tableau Server more information. During installation, you will select the new identity store type. See Configure Initial Node Settings.

External Identity Store Configuration Reference

Tableau Server supports connecting to an external directory using LDAP. In this scenario, Tableau Server imports users from the external LDAP directory into the Tableau Server repository as system users.

This topic provides a description of all LDAP-related configuration options Tableau Server supports. If you are connecting to Active Directory, we strongly recommend that you automatically configure the LDAP connection with Tableau Server as part of Setup, rather than configuring the connection manually. See Configure Initial Node Settings.

The options listed in this reference can be used for any LDAP-compliant directory. If you do not have experience configuring LDAP, then work with your directory administrator, or with an LDAP expert.

This is a reference topic. For more information about how Tableau Server stores and manages users, start with Identity Store.

Configuration methods

Configuration parameters that enable Tableau Server to connect to your LDAP directory are stored in .yml files. These files are managed and synchronized by various services in Tableau Server. Updating the .yml files must be done using a Tableau Services Manager (TSM) interface.

Do not attempt to update .yml files directly with a text editor. TSM must manage all updates for proper operation.

The .yml configuration files are composed of key-value pairs. For example, the key, `wgserver.domain.username`, takes a username as a value. This key defines the username that will be used to authenticate to the LDAP directory during the bind operation.

There are four different TSM methods that can set yml key values. The four methods are described here, using the `wgserver.domain.username` key as an example to illustrate the different methods:

- configKey key-value pairs—You can update a .yml configuration file key by updating the `wgserver.domain.username` key running tsm configuration set Options, or by including the key in a JSON configuration file under a configKey entity. See Configuration File Example.

  The configKey key-value pairs in a JSON configuration file are the same as those used for `tsm configuration set` but they are set differently. This topic refers to both of these methods as *configKey*.

  Unlike when using configEntities and native tsm commands that are described below, configKey input is not validated. When you set an option with a configKey, the value that you enter is copied as a literal string to the underlying .yml configuration files. For example, for a key where `true` or `false` are the valid inputs, when you configure the key using a configKey key-value pair, you can enter an arbitrary string value and it will be saved for the key. In such cases, invalid values will undoubtedly lead to LDAP configuration errors.

We recommend using configKeys only when no option exists to set the configuration with the other three options listed below (configEntities, a native tsm command, or the TSM Web UI). When using configKeys be sure to double-check your values and be sure to mind case-sensitivity.

- configEntities JSON—You can update a .yml configuration file by passing the `user-name` option in a *configEntities* JSON.

  When you configure a value using configEntities options in a JSON file, the values are validated before they are saved. Values are case-sensitive. For details on how to configure a value using configEntities, see the identityStore Entity example. The JSON file is imported with the tsm settings import command. The options available for configEntities are a subset of all the .yml key-value pairs.

  Validation means that the import command will only succeed if all the values in the JSON file are valid data types. For example, if you enter `no` for a value that only accepts `true` or `false`, then you will receive an error and the configuration is not imported.

  You can only import JSON configuration files only as part of the initial configuration. If you need to make LDAP changes after you have imported the JSON configuration file and initialized Tableau Server, do not attempt to re-import the JSON file. Instead, make individual key changes with native tsm commands if available, or using configKeys and `tsm configuration set`.

- Native tsm commands—You can update a .yml configuration file by passing the `ldapuser` option with the *native tsm command* `tsm user-identity-store`. As with configEntities, values that you enter with the native tsm command are validated before they are saved.

  Not all key-value pairs in a .yml file can be set using native tsm commands.

- TSM GUI—You can set configuration values during Setup, using the TSM GUI. If you are connecting to Active Directory, and configure the Tableau identity store during Setup, with the GUI, then you are prompted for an account with AD read access. The

`wgserver.domain.username` key is set when you enter credentials.

This scenario only works if you are connecting to Active Directory. Tableau Server does not support arbitrary LDAP configuration as part of the GUI Setup process.

Consider using the Tableau Identity Store Configuration Tool to generate your LDAP json configuration file. The Tableau Identity Store Configuration Tool will also generate a list of key/-value pairs that you can set by running tsm configuration set Options. The tool itself is not supported by Tableau. However, using a JSON file created by the tool instead of creating a file manually does not change the supported status of your server.

Configuring Active Directory

If you are configuring Tableau Server to use Active Directory, we recommend using the TSM Web UI during installation. The TSM Web UI is optimized to configure Tableau Server for Active Directory with the minimum necessary input. See Configure Initial Node Settings.

Configuration reference table

configEntities option **(Options are case sensitive)**	**Native tsm command**	**configKey** **(Used with tsm configuration set command or in the configKeys section of a JSON file)**	**Scenario**	**Notes**
type	N/A	wgserver.authenticate	AD, LDAP, Local	Where you want to store user identity information. Values: `local` or `activedirectory`. If you want to connect to any LDAP server, enter `activedirectory`.

sslPort	N/A	wgserv-er.domain.ssl_port	A-D, L-D-A-P	Use this option to specify the secure port of the LDAP server. We recommend secure LDAP for simple bind. LDAPS is usually port 636.
N/A	N/A	wgserv-er.-domain.ldap.starttls.e-nabled	A-D, L-D-A-P	Values: `true` or `false`. Beginning with version 2021.2, this key is set to `true` by default when Tableau Server is configured to connect to Active Directory. As a result, simple bind to LDAP directory is encrypted when a valid SSL/TLS certificate is present in the Tableau key store. For more information, see Configure Encrypted Channel to LDAP External Identity Store. This key is set to `false` by default when Tableau Server is configured to connect to a an arbitrary (but not Active Directory) LDAP server. This key was introduced (but not set) in version 2021.1.
port	N/A	wgserv-er.domain.port	A-D, L-D-A-P	Use this option to specify the non-secure port of the LDAP server. Plaintext is usually 389.

domain	domain	wgserver.domain.default	AD	In Active Directory environments, specify the domain where Tableau Server is installed, for example, "example.lan". For non-AD LDAP: the string you enter for this value is displayed in the "Domain" column of user management tools. You can enter an arbitrary string, but the key cannot be blank. This key is redundant with wgserver.domain.fqdn. The values for both keys must be the same. Native tsm command: Uses tsm user-identity-store set-connection [options] command.
username	ldapusername	wgserver.domain.username	AD, LDAP	The user name that you want to use to connect to the directory service. The account that you specify must have permission to query the directory service. For Active Directory, enter the username, for example, `jsmith`. For LDAP servers, enter the distinguished name (DN) of the user that you want to use to connect. For example, "`cn=-`

				`jsmith,dc=example,dc=lan"`. Native tsm command: Uses tsm user-identity-store set-connection [options] command.
password	ldap-password	wgserver.domain.password	AD, LDAP	The password of the user account that you will use to connect to the LDAP server. Native tsm command: Uses tsm user-identity-store set-connection [options] command.
directoryServiceType	N/A	wgserver.domain.directoryservice.type	AD, LDAP	The type of LDAP directory service that you want to connect to. Values:`activedirectory` or `openldap`.
kerberosPrincipal	kerbprincipal	wgserver.domain.ldap.principal	AD, LDAP	The service principal name for Tableau Server on the host machine. The keytab must have permission for this principal. Do not use the existing system keytab at `/etc/krb5.keytab`. Rather, we recommend that you register a new service principal name. To see principals in a given keytab, run the `klist -k` command. See Understanding Keytab Requirements. Native tsm command: Uses tsm user-identity-store set-connection [options] command.

hostname	host-name	wgserv-er.-domain.ldap.host-name	A-D, L-D-A-P	The hostname of the LDAP server. You can enter a hostname or an IP address for this value. The host that you specify here will be used for user-/group queries on the primary domain. In the case where user-/group queries are in other domains, Tableau Server will query DNS to identify the appropriate domain con-troller. Native tsm command: Uses tsm user-identity-store set-connection [options] command.
mem-ber-sRetriev-alPageSize	N/A	wgserv-er.-domain.ldap.mem-ber-s.retrieval.page.size	A-D, L-D-A-P	This option determines the maximum number of results returned by an LDAP query. For example, consider a scenario where Tableau Server is importing an LDAP group that contains 50,000 users. Attempting to import such a large number of users in a single oper-ation is not a best practice. When this option is set to 1500, Tableau Server imports the first 1500 users in the first response. After those users are pro-cessed, Tableau Server requests the next 1500 users from the LDAP server, and so forth. We recommend that you modify this option only to accommodate the

				requirements of your LDAP server.
N/A	N/A	wgserver.domain.ldap.connectionpool.enabled	AD, LDAP	When this options is set to `true`, Tableau Server will attempt to reuse the same connection when sending queries to the LDAP server. This behavior decreases the overhead of having to re-authenticate with the LDAP server on each new request. Connection pooling only works with simple bind and TSL/SSL bind connections. Connection pooling is not supported for GSSAPI bind connections.
N/A	N/A	wgserver.domain.accept_list	AD	Allows connection from Tableau Server to secondary Active Directory domains. A secondary domain is one that Tableau Server connects to for user synchronization, but is a domain where Tableau Server is not installed. To ensure that Tableau Server can connect to other Active Directory domains, you must specify the trusted domains by setting the `wgserver.domain.accept_list` option with TSM. For more information, see wgserver.domain.accept_list.
N/A	N/A	wgserver.domain.whitelist	AD	**Important:** Deprecated as of version 2020.4.0. Use wgserver.domain.accept_list instead. Allows connection from Tableau Server to secondary Active Directory

				domains. A secondary domain is one that Tableau Server connects to for user synchronization, but is a domain where Tableau Server is not installed. To ensure that Tableau Server can connect to other Active Directory domains, you must specify the trusted domains by setting the `wgserver.domain.whitelist` option with TSM. For more information, see wgserver.domain.whitelist .
kerberosConfig	kerbconfig	No direct mapping	AD, LDAP	The path to the Kerberos configuration file on the local computer. If you are installing into Active Directory, we don't recommend using the existing Kerberos configuration file or keytab file that may already be on the domain-joined computer. See Identity Store Native tsm command: Uses tsm user-identity-store set-connection [options] command.
kerberosKeytab	kerbkeytab	No direct mapping	AD, LDAP	The path to the Kerberos keytab file on the local computer. It is recommended that you create a keytab file with keys specifically for Tableau Server service and that you do not share the keytab file with other applications on the computer. For example,

				on Linux, you might place the keytab file in the `/var/-opt/tableau/keytab` directory. Native tsm command: Uses tsm user-identity-store set-connection [options] command.
nickname	N/A	wgserv-er.domain.nickname	A-D, L-D-A-P	The nickname of the domain. This is also referred to as the NetBIOS name in Windows/Active Directory envir-onments. The `nickname` option is required for all LDAP entities. The value cannot be null. If your organ-ization does not require a nick-name/NetBIOS, then pass a blank key, for example: `""`.
root	N/A	wgserv-er.domain.ldap.root	L-D-A-P	If you do not use a dc component in the LDAP root or you want to specify a more complex root you need to set the LDAP root. Use the "o=my,-,u=root" format. For example, for the domain, `example.lan`, the root would be `"o=example,u=lan"`.
server-SideSorting	N/A	wgserv-er.-domain.ldap.server_side_sorting	L-D-A-P	Whether the LDAP server is con-figured for server-side sorting of query results. If your LDAP server supports server-side sorting, set this option to `true`. If you are unsure whether your LDAP server supports this, enter `false`, as mis-configuration may cause errors.

rangeRetrieval	N/A	wgserver.domain.ldap.range_retrieval	LDAP	Whether the LDAP server is configured to return a range of query results for a request. This means that groups with many users will be requested in small sets instead of all at once. LDAP servers that support range retrieval will perform better for large queries. If your LDAP server supports range retrieval, set this option to `true`. If you are unsure whether your LDAP server supports range retrieval, enter `false`, as misconfiguration may cause errors.
bind	N/A	wgserver.domain.ldap.bind	LDAP	The way that you want to secure communication to the directory service. Enter `simple` for LDAP unless you are connecting to an LDAP server with Kerberos. For Kerberos, enter `gssapi`.
N/A	N/A	wgserver.domain.ldap.domain_custom_ports	LDAP	**Note**: This key is only supported for Tableau Server on Linux. Allows you to map child domains and their LDAP ports. Domain and port are separated by a colon (:) and each domain:port pair is separated by a comma (,) using this format: `FQDN1:port,FQDN2:port` Example: `tsm configuration set -k wgserv-`

				`er.domain.ldap.domain_custom_ports -v child-domain1.lan:3269,child-domain2.lan:3269,child-domain3.lan:389`
distinguishedNameAttribute	N/A	wgserver.domain.ldap.dnAttribute	LDAP	The attribute that stores the distinguished names of users. This attribute is optional, but it greatly improves the performance of LDAP queries. **Important**: Do not set this option as part of the initial configuration. Only set this after you have validated overall LDAP functionality. You must have a dnAttribute set in your organization before setting this key.
groupBaseDn	N/A	wgserver.domain.ldap.group.baseDn	LDAP	Use this option to specify an alternative root for groups. For example, if all of your group are stored in the base organization called "groups," then enter `"o=groups"`.
N/A	classnames	wgserver.domain.ldap.group.classnames	LDAP	By default Tableau Server looks for LDAP group object classes containing the string “group”. If your LDAP group objects do not fit the default class name, override the default by setting this value. You can

				provide multiple classnames separated by commas. If your group names include commas, you must escape them with a backslash (\). For example, if you have a group name, `groupOfNames, top`, then enter `"groupOfNames\, top"`. Tableau LDAP implementation interprets LDAP objects as either user or group. Therefore, be sure that you are entering the most specific class name. Overlapping class names between users and groups may cause conflicts. Native tsm command: Uses tsm user-identity-store set-group-mappings [options] command.
groupBaseFilter	baseFilter	wgserver.domain.ldap.group.baseFilter	LDAP	The filter that you want to use for groups of users of Tableau Server. You might specify an object class attribute and an organization unit attribute. For example: `"(&(objectClass=groupofNames)(ou=Group))"` If `"(&(objectClass=inetOrgPerson)`

				`(ou=People))` " doesn't work in your LDAP implementation, then specify the base filter that works for your Tableau user base. This is a required key. It cannot be blank. Native tsm command: Uses tsm user-identity-store set-group-mappings [options] command.
groupName	groupname	wgserver.domain.ldap.group.name	LDAP	The attribute that corresponds to group names on your LDAP server. Native tsm command: Uses tsm user-identity-store set-group-mappings [options] command.
groupEmail	groupemail	wgserver.domain.ldap.group.email	LDAP	The attribute that corresponds to group email addresses on your LDAP server. Native tsm command: Uses tsm user-identity-store set-group-mappings [options] command.
groupDescription	description	wgserver.domain.ldap.group.description	LDAP	The attribute that corresponds to group descriptions on your LDAP server. Native tsm command: Uses tsm user-identity-store set-group-mappings [options] command.

member	member	wgserver.domain.ldap.group.member	LDAP	Specify the LDAP attribute that contains a list of distinguished names of users that are part of that group. Native tsm command: Uses tsm user-identity-store set-group-mappings [options] command.
N/A	N/A	wgserver.domain.ldap.group.memberURL	LDAP	Specify the name of the LDAP attribute that stores the LDAP query for dynamic groups.
userBaseDn	N/A	wgserver.domain.ldap.user.baseDn	LDAP	Use this option to specify an alternative root for users. For example, if all of your users are stored in the base organization called "users," then enter `"o=users"`.
N/A	classnames	wgserver.domain.ldap.user.classnames	LDAP	By default Tableau Server looks for LDAP user object classes containing the string “user” and “inetOrgPerson”. If your LDAP user objects do not use these default class names, override the default by setting this value. You can provide multiple classnames separated by commas. For example: `"userclass1, userclass2"`. If your names include commas, you must escape them with a backslash (\). For example, if you have a name, `Names, top`, then enter `"Names\, top"`.

				Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.
user-BaseFilter	base-filter	wgserv-er.-domain.ldap.user-.baseFilter	L-D-A-P	The filter that you want to use for users of Tableau Server. You might specify an object class attribute and an organization unit attribute. For example: `"(&(objectClass-s=inetOrgPerson) (ou=People))"` Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.
user-Username	ldapu-ser-name	wgserv-er.-domain.ldap.user-.username	L-D-A-P	The attribute that corresponds to user names on your LDAP server. Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.
user-Dis-playName	dis-play-name	wgserv-er.-domain.ldap.user-.displayname	L-D-A-P	The attribute that corresponds to user display names on your LDAP server. Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.
userEmail	email	wgserv-er.-	L-D-	The attribute that corresponds to user email addresses on your LDAP

		domain.ldap.user-.email	A-P	server. Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.
user-Certificate	cer-tific-ate	wgserv-er.-domain.ldap.user-.usercertificate	L-D-A-P	The attribute that corresponds to user certificates on your LDAP server. Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.
N/A	thum-bnail	wgserv-er.-domain.ldap.user-.thumbnail	L-D-A-P	The attribute that corresponds to user thumbnail images on your LDAP server. Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.
user-JpegPhoto	jpeg-photo	wgserv-er.-domain.ldap.user-.jpegphoto	L-D-A-P	The attribute that corresponds to user profile images on your LDAP server. Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.
memberOf	mem-berof	wgserv-er.-domain.ldap.user-.memberof	L-D-A-P	Group that the user is a member of. Native tsm command: Uses tsm user-identity-store set-user-mappings [options] command.
groupClass-	N/A	wgserv-	L-	By default Tableau Server looks for

Names		er.-domain.ldap.-group.classnames	D-A-P	LDAP group object classes containing the string “group”. If your LDAP group objects do not fit the default class name, override the default by setting this value. For configEntity: This option takes a list of strings, which requires passing each class in quotes, separated by a comma (no space) and within brackets. For example: `["basegroup","othergroup"]`. For configKey: Enter each class, separated by a comma (no space) and within double quotes. For example: `"basegroup,othergroup”`.
user-ClassNames	N/A	wgserver.-domain.ldap.user-.classnames	L-D-A-P	By default Tableau Server looks for LDAP user object classes containing the string “user” and “inetOrgPerson”. If your LDAP user objects do not use these default class names, override the default by setting this value. For configEntity: This option takes a list of strings, which requires passing each class in quotes, separated by a comma (no space) and within brackets. For example: `["user-class1",userclass2”]`. For configKey: Enter each class, sep-

				arated by a comma (no space) and within double quotes. For example: "userclass1,userclass2”.

Calculated configKeys

The following Kerberos-related configKeys are calculated and set according to multiple environmental inputs. As such, they must be set by the native tsm command or configEntities. Do not attempt to set these configKeys manually.

Calculated configKey	**To use the native TSM command:**	**To use configEntity json:**
wgserver.domain.ldap.kerberos.conf, cfs.ldap.kerberos.conf	Set the Kerberos configuration file location with the `kerbconfig` option of tsm user-identity-store set-connection [options] command.	Set the Kerberos configuration file location with the `kerberosConfig` configEntity option.
wgserver.domain.ldap.kerberos.keytab, cfs.ldap.kerberos.keytab	Set the Kerberos keytab file location with the `kerbkeytab` option of tsm user-identity-store set-connection [options] command.	Set the Kerberos ketytab file location with the `kerberosKeytab` configEntity option.

Unsupported configKeys

Some unsupported configKeys are present in underlying .yml configuration files. The following keys are not intended for standard deployments. Do not configure these keys:

- wgserver.domain.ldap.kerberos.login
- wgserver.domain.ldap.guid
- wgserver.domain.fqdn: this key is redundant with wgserver.domain.default. The values for both keys must be the same. Only update wgserver.domain.fqdn if the value does not match wgserver.domain.default.

Basic Product Data

By default Tableau products send usage data to Tableau so that we can understand how customers use our software, and gain insights into where they are successful and where they might run into problems that we can address. For example, this data can help us learn where upgrades are commonly failing and allow us to make product changes to address those issues, or identify what portion of our user base needs informed about a security issue that applies to a specific version of Tableau Server. You can disable the sending of this data at installation time, or later. For details on how to do this, see the instructions for Tableau Desktop or Tableau Server.

Even when you disable the sending of product usage data, certain basic product data is sent to Tableau. This Basic Product Data includes information about products and their processes, including which product or process is running, when they start up, what operating system they are running on, licensing information, which machine or cluster of machines has sent the data (using unique pseudonymized identifiers), and whether the product is configured to send product usage data.

You can disable the sending of Basic Product Data at the machine level, or at the enterprise level, by blocking traffic sent to **prod.telemetry.tableausoftware.com**.

Disabling sharing of Basic Product Data on individual computers

Important: This procedure involves modifying your local `hosts` file. If you do not know what this is, you should not change it. You should only make this change if you understand the implications of making changes to the file, know how to change the file, and have made a backup of the file for safety.

Modifying `hosts` files changes network behavior for computers. Detailed instructions for modifying `hosts` files are provided by operating system providers such as Microsoft, Apple or Linux Distributions.

1. Make a copy of your existing hosts file and save it to a computer that is not your Tableau computer. This is your backup, in case you need to reverse your changes. Do not start modifying the file until you have made a backup copy of it.
2. Modify your computer's `hosts` file to include these lines:

```
# Stops sending Product Usage to Tableau (prod.tele-
metry.tableausoftware.com).
# Learn more here: http:\\tableau.com\derived-data
127.0.0.1    prod.telemetry.tableausoftware.com
```

The first and second lines are comments, explaining the third line.

The third line prevents all traffic to `prod.telemetry.tableausoftware.com` (`http://-prod.telemetry.tableausoftware.com/`) from leaving your local machine by sending it to the Internal host loopback address. The data does not get sent outside the computer.

Disabling the sharing of Basic Product Data at the enterprise level

To disable sending of Basic Product Data on an enterprise level, modify your Network Firewall to prevent outbound traffic to `prod.telemetry.tableausoftware.com`.

This domain is used by Tableau to receive the Basic Product Data about process launch and shutdown. It is also used for the more general Product Usage Data. Blocking traffic to this domain it you will prevent both kinds of data from being sent.

Traffic to this domain will occur on Ports 80 (for initial registration of our Product Data clients) and on Port 443 (for all subsequent traffic). To completely prevent product data from being sent, block all traffic to this domain.

For details on how to configure your network firewall, refer to your vendor or your internal IT department. Tableau cannot provide these instructions.

Archived Content

This is archived content

Deployments on public clouds continue to be supported but the content for third-party public cloud deployments is no longer updated.

For the latest Tableau Server deployment content, see the Enterprise Deployment Guide and the Deploy section of Tableau Server help.

For those customers who have access, we recommend Tableau Cloud. For more details, see:

- Tableau Cloud Manual Migration Guide
- Tableau Cloud Trial for Admin
- Tableau Cloud: Get Started for Admin

Self-Host Tableau Server in a Public Cloud Service

This is archived content

Deployments on public clouds continue to be supported but the content for third-party public cloud deployments is no longer updated.

For the latest Tableau Server deployment content, see the Enterprise Deployment Guide and the Deploy section of Tableau Server help.

For those customers who have access, we recommend Tableau Cloud. For more details, see:

- Tableau Cloud Manual Migration Guide
- Tableau Cloud Trial for Admin
- Tableau Cloud: Get Started for Admin

Introduction

Even if you don’t have your own infrastructure and server hardware, you can deploy an enterprise-level Tableau Server installation in the cloud. Building a cloud-based solution has many benefits over an on-premises installation. For example, the overall total cost of ownership for

building a Tableau Server solution in the cloud is normally much less than a similar on-premises solution because you don't have to buy all of the expensive hardware. In addition, the cloud can provide better uptime, reliability, and fault-tolerance, especially if you deploy your solution across different regions and availability zones.

Looking for Tableau Server on Windows? See Self-Host Tableau Server in a Public Cloud Service.

You can build and scale your Tableau environment in the following cloud environments:

- **Amazon Web Services** - You install and manage Tableau Server on Amazon Web Services (AWS). For more information, see Install Tableau Server in the AWS Cloud.
- **Google Cloud Platform** - You install and manage Tableau Server on the Google Cloud Platform. For more information, see Install Tableau Server on the Google Cloud Platform.
- **Microsoft Azure** - You install and manage Tableau Server on Microsoft Azure. For more information, see Install Tableau Server on Microsoft Azure.
- **Alibaba Cloud** - You install and manage Tableau Server on Alibaba Cloud. For more information, see Install Tableau Server in the Alibaba Cloud.

About Tableau Advanced Management on Tableau Server

Important: As of September 16, 2024, Advanced Management is no longer available as an independent add-on option. Advanced Management capabilities are only available if you previously purchased Advanced Management, or if you purchase certain license edi-

tions - either Tableau Enterprise (for Tableau Server or Tableau Cloud) or Tableau+ (for Tableau Cloud).

Tableau Advanced Management is a collection of features for Tableau Server which provide enhanced security, manageability, and scalability capabilities. The Advanced Management capabilities are available to you if you have Tableau Enterprise.

Note: Tableau Advanced Management includes several Tableau Server features and two separately installed tools: Tableau Content Migration Tool and the Tableau Resource Monitoring Tool. For more information, see the feature table later in this topic.

Advanced Management Licensing Requirements

Advanced Management is licensed on a per Deployment basis, which may be User-Based or Core-Based. A Deployment includes a licensed production Tableau Server installation and two licensed non-production Tableau Server installations that support the production installation. For more information on Deployment, see the EULA Documentation.

- Advanced Management can only be activated on a licensed Tableau Server Deployment. This means that your Tableau Server must be first activated with a valid key that is either User-Based or Core-Based, before applying the Advanced Management product key. For more information on how to acquire the Advanced Management capabilities and get the product key, contact your account manager.
- When the product key is active and enabled, you can use all the features that are included in Advanced Management.
- When the Advanced Management product key is removed or deactivated, you will no longer be able to use the features that require a valid Advanced Management license. Any associated data will not be deleted. Each feature might have slight differences on

what happens when the license expires. For more information on the individual features use the links in the table below.

The following table lists the features that are included and require a valid Advanced Management license:

Feature	Description	Requirements to use the feature	Version
About Tableau Resource Monitoring Tool	Provides a comprehensive look at the health of Tableau Server. With the Resource Monitoring Tool you can identify issues that cause slow load times, extract failures, and other critical issues and can help you proactively address the issues that impact end user experience.	The installation of Tableau Server that you are monitoring must have an Advanced Management license.	Available in 2019.3 and later. Linux support was introduced in version 2020.4.
About Tableau Content Migration Tool	The Content Migration Tool provides an easy way to copy or migrate content between Tableau Server projects, sites and deployments. You can do this between projects on separate Tableau Server installations (for instance, between a development instance of Tableau Server and a production instance of Tableau Server), or between projects on a single Tableau Server installation.	• Both the source Tableau Server (Server that you are moving the content from) and the target Tableau Server (Server that you are moving the content to) must have a valid Advanced Management license. • The Content	Available in version 2019.3 and later.

Feature	Description	Requirements to use the feature	Version
		Migration Tool must be installed on a version of Microsoft Windows that supports .NET 4.6.1 (Windows 7 or later, Windows Server 2008R2 or later).	
Activity Log	The Activity Log writes log events to the vizportal logs folder on the local hard drive for further analysis and auditing.	The Tableau Server must have a valid Advanced Management license.	Available in version 2022.3 and later.
Tableau Server External Repository	Allows you to deploy Tableau Server Repository external to Tableau Server. The Tableau Server Repository is a PostgreSQL database that stores data about all user interactions, extract refreshes, and more.	The Tableau Server that is using an external repository must have a Advanced Management license.	Available in version 2019.3 and later. - AWS supported on Tableau Server version 2019.3 and later. - Azure supported on

Feature	Description	Requirements to use the feature	Version
			Tableau Server version 2020.4 and later.
Workload Management through Node Roles	Using node roles, you can configure where certain types of workloads are processed on your Tableau Server installation. The node roles features allows you to dedicate and scale resources to specific workloads (ex: extract refreshes, subscriptions).	The Tableau Server must have a valid Advanced Management license.	Available in version 2019.3 and later.
Tableau Server Key Management System	Gives you additional functionality to configure Tableau Server to use AWS as the KMS for extract encryption.	Tableau Server must have a valid Advanced Management license.	Available in version 2019.3 and later. - AWS supported on Tableau Server version 2019.3 and later. - Azure supported on Tableau Server version 2021.1

Feature	Description	Requirements to use the feature	Version
			and later.
Tableau Server External File Store	Allows you to use network attached storage(NAS) as your File Store. This removes the need to run File Store locally on your Tableau Server	Tableau Server must have a valid Advanced Management license.	Available in version 2020.1 and later.
Tableau Backgrounder Resource Limits	The Tableau Server Resource Limits Manager tracks backgrounder resource usage in relation to the set resource limits to make sure the resource limits are applied correctly.	Tableau Server must have a valid Advanced Management license.	Available in version 2022.1 and later.
Tableau Server Independent Gateway	Tableau Server Independent Gateway allows you to install a Tableau Server-managed reverse proxy on a separate computer. Install Independent Gateway in your DMZ to provide security for your Server installation and all your data. The Independent Gateway can be used in a variety of configurations, and can be scaled to meet your client connection demands.	Tableau Server must have a valid Advanced Management license.	Available in version 2022.1 and later.

Activating the Advanced Management product key

The Advanced Management license is applied to a Tableau Server Installation and can be used for both User-Based and Core-Based installations.

Here is a quick overview of how you can activate Advanced Management on your Tableau Server Installation.

Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850

2. Click **Licensing** on the **Configuration** tab. Click **Activate License**.

3. Enter or paste your Advanced Management product key and click **Activate**.

4. On the **Register** page, enter your registration information and click **Register**.

5. Follow the prompts and restart Tableau Server after registration is complete.

   Note: If this is a new Tableau Server installation, and you apply an Advanced Management Key before you apply the Tableau Server product key, you will see an error. You may however, continue the installation and apply the Tableau Server product key using the same steps described above.

Use the TSM CLI

1. Open a command prompt as administrator on a node in the Tableau Server cluster.

2. Run the following command with your Advanced Management product key to activate the license:

   ```
   tsm licenses activate -k <server-management-add-on-product key>
   ```

> **Note:** If this is a new Tableau Server installation, run the command twice, first with the Tableau Server product key and then with an Advanced Management product key.

3. Apply the changes and restart the Server:

```
tsm pending-changes apply
```

Who can do this

Only Server Administrators can activate Advanced Management license keys.

About Tableau Resource Monitoring Tool

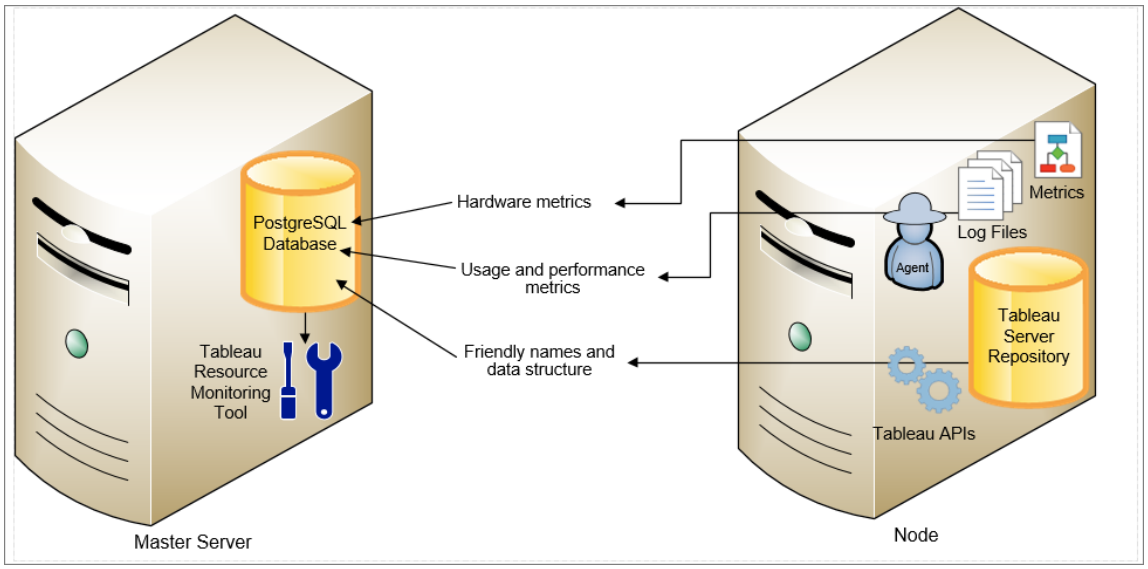
This document gives you an overview of the Tableau Resource Monitoring Tool.

What is Resource Monitoring Tool?

The Resource Monitoring Tool is a tool that you can use to monitor the health and performance of your Tableau Server. It gathers data from your Tableau Server to provide a comprehensive look at the health of Tableau Server. Using this tool, you can identify the cause of slow load times, extract failures, and other critical issues. To use the Resource Monitoring Tool, you must have Advanced Management enabled on your server. For more information on Advanced Management, see About Tableau Advanced Management on Tableau Server

The Resource Monitoring Tool has two main components: Resource Monitoring Tool Server (RMT Server) and Agent.

- The RMT Server is where the data from Tableau Server is collated and served up through a web interface. This is also where you can configure, monitor, and analyze the health and performance of Tableau Server.
- An Agent runs on each of the nodes in your Tableau cluster to monitor their performance and activity. The following diagram illustrates the interaction between a Tableau Server node and the RMT Server.



Get Started with Tableau Resource Monitoring Tool

This article will help you get started with Tableau Resource Monitoring Tool. It contains links to other articles about information you need to prepare before installing Resource Monitoring Tool, links to upgrade and other useful resources.

In an effort to align with our company values of Equality, we have changed non-inclusive terminology where possible. Because changing terms in certain places can causing a breaking change, we maintain the existing terminology. So, you may continue to see the terms in CLI commands and options, installation folders, configuration files. and other instances. For more information, see About Tableau Help.

Pre-Installation

Product Compatibility with Tableau Server

The Resource Monitoring Tool can be installed to monitor supported Tableau Server versions.

The table below lists the version compatibility between Resource Monitoring Tool(RMT) and Tableau Server:

RMT Version	OS support	Tableau Server Version
2023.3.x	Windows, Linux	2021.1.x-2022.1.x , 2022.3.x, 2023.1.x, 2023.3.x
2023.1.x	Windows, Linux	2021.1.x-2022.1.x , 2022.3.x, 2023.1.x
2022.3.x	Windows, Linux	2021.1.x - 2022.1.x , 2022.3.x
2022.1.x	Windows. Linux	2021.1.x - 2022.1.x
2021.4.x	Windows, Linux	2021.1.x - 2021.4.x
2021.3.x	Windows, Linux	2021.1.x - 2021.3.x
2021.2.x	Windows, Linux	2021.1.x - 2021.2.x
2021.1.x	Windows, Linux	2021.1.x

Resource Monitoring Tool Server (RMT Server) and Agent Compatibility

Generally, we recommend that you install the same version of RMT Server and Agent to be sure that they are compatible. If you have Agents using versions not compatible with the RMT Server version, a critical incident is logged. For more information, see Agent Incidents.

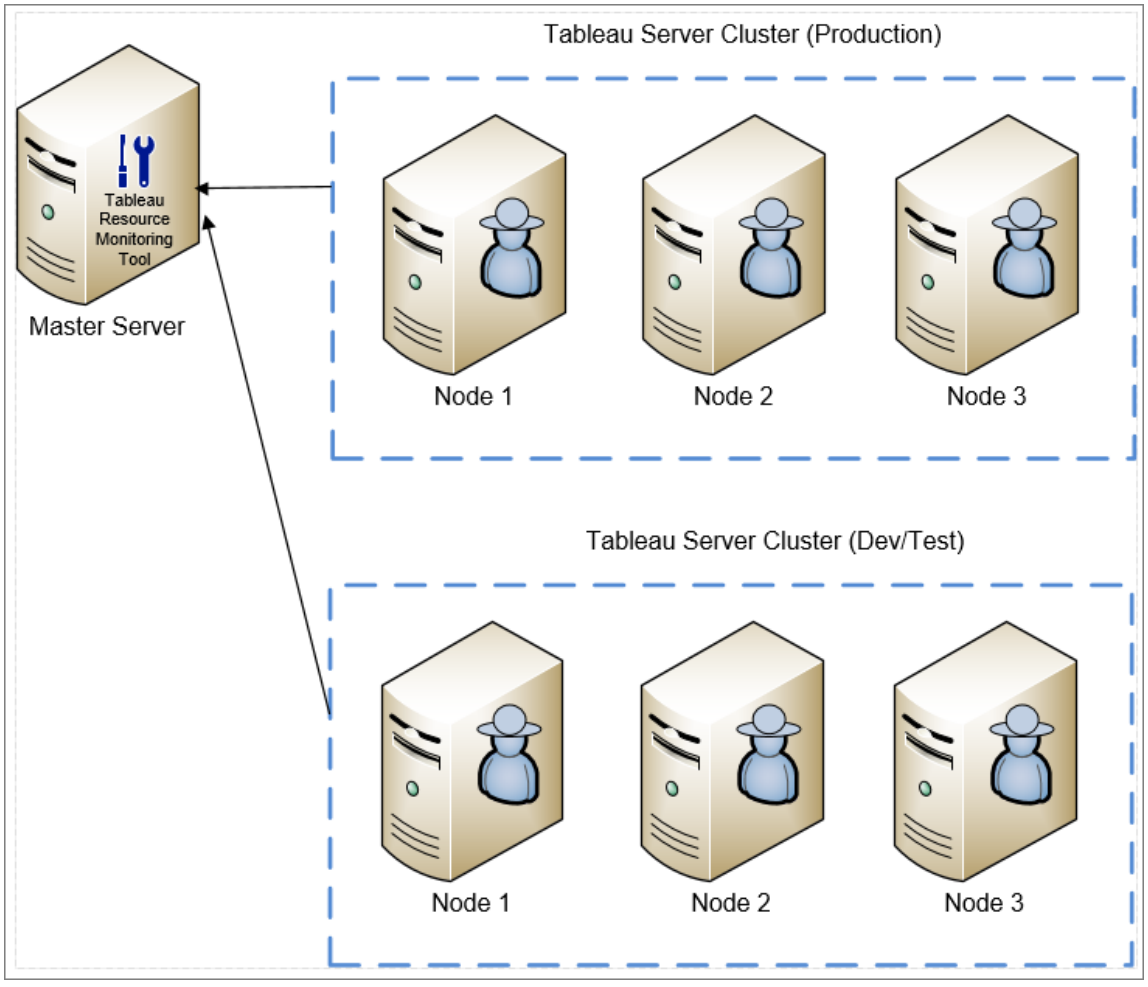
Concepts and Terms

Get familiar with the concepts and terms used in the Resource Monitoring Tool topics. For more information, see Concepts.

Setup Architecture

The Resource Monitoring Tool has two main components: Resource Monitoring Tool Server (RMT Server) and Agent. The RMT Server should be installed on its own machine, and the Agent is installed on each node of your Tableau Server Cluster. For a more details about the setup architecture, see Install the Tableau Resource Monitoring Tool.

A high level view of the Resource Monitoring Tool setup. This diagram shows a three node Tableau Server Cluster.



Minimum Hardware Requirements and Recommendations for Resource Monitoring Tool

Resource Monitoring Tool handles all of the processing, aggregation, storage, and web interface to the monitoring data collected from your Tableau Server installation. For production use, the machine you install Resource Monitoring Tool on should meet or exceed the minimum hardware recommendations. For more details, see Minimum Hardware Requirements and Recommendations for Tableau Resource Monitoring Tool.

Pre-Installation Checklist for Resource Monitoring Tool

Before you install Resource Monitoring Tool, review these action items and complete any necessary steps: Pre-Installation Checklist - Tableau Resource Monitoring Tool

Troubleshoot

Be sure to check our Troubleshoot Tableau Resource Monitoring Tool Issues for answers to common questions before contacting support.

Concepts

This document briefly explains some of Tableau Resource Monitoring Tool core concepts and defines some terminology you will see often.

In an effort to align with our company values of Equality, we have changed non-inclusive terminology where possible. However, you may continue to see the terms in CLI commands and options and other instances. For more information, see About Tableau Help.

Agent

Resource Monitoring Tool Agent runs as a background service on each of your Tableau servers. It must be installed on all the nodes in a Tableau Server installation that you want to monitor.

The agent is a lightweight process that monitors the Tableau activity and general performance of the server it is installed on. It reports those details back to the Resource Monitoring Tool Server.

Resource Monitoring Tool Server (RMT Server)

The Resource Monitoring Tool Server hosts the web application that users interact with. It also does much of the background processing to collate and monitor the data from the agents. We will use the term **RMT Server** to refer to this component.

Environment

Environment is the term used to refer to a single node Tableau Server installation, or a Tableau Server cluster. In the RMT Server web interface, each node of the Tableau Server is called a **Server**.

Resource Monitoring Tool can monitor multiple Tableau Server installations. For example, you may have both production and staging Tableau Servers. In Resource Monitoring Tool, you can create two environments (*Production* and *Staging*) and monitor both of those Tableau Server clusters.

Environment Status

The *environment status* indicates the state of the current Tableau environment. The status is determined by the number and type of incidents and the status of Tableau's various processes. The following sections will describe what each status means in these terms.

OK

No warning or critical incidents have occurred today and there are not currently any failed processes.

Warning

The Environment is in a **Warning** state when any of the following are true:

- At least one warning incident has occurred today.
- A process is **Unlicensed**.
- A process is **Offline**.

Critical

The Environment is in a **Critical** state when any of the following are true:

- At least one critical incident has occurred today.
- The Primary Gateway is **Offline**.
- All instances of a process are **Unlicensed**.
- All instances of a process are **Offline**.

Server

Server is the term used to refer to each of nodes in the Tableau Server installation. Every machine that you install the Resource Monitoring Tool Agent on is considered a Server.

Pre-Installation Checklist - Tableau Resource Monitoring Tool

The following topic provides prerequisites for installing Tableau Resource Monitoring Tool. It also describes the operating system permissions required for the service accounts used.

Machine, Network, and Account Requirements

> **Note:** Resource Monitoring Tool (RMT) server and agent are not Federal Information Processing Standard (FIPS) 140-2 compliant, and will not function properly on operating systems running with FIPS mode enabled.

Complete the following prerequisites before installing the Tableau Resource Monitoring Tool.

1. Make sure you have administrative privileges for all computers that will run the Resource Monitoring Tool. This includes the computer running the RMT Server, and all the Tableau Server nodes in the installation you will be monitoring with Resource

Monitoring Tool.

2. Open port 5672 to inbound TCP traffic on the computer running the RMT Server

   The Resource Monitoring Tool Agent application sends the data that it collects through port 5672, so the machine where the RMT Server is installed will need to allow inbound TCP traffic on that port.

3. Open the following ports:

   - Open port 80 to inbound TCP traffic on the computer running the RMT Server

     This is the port on which the RMT Server will host the web interface. Note that we can change the port number during installation if there is a 3rd party service already using it. To make the web interface accessible from outside the RMT Server, this port will need to be opened.

   - Open port 443 to inbound TCP traffic on the computer running the Resource Monitoring Tool RMT Server.

     This is the port on which connections come from the Agent when testing the connection to the RMT Server.

   - Open port 5672 and 5671 (TLS connections) to outbound TCP traffic on all Agent nodes.

     The Agent application sends the data that it collects to the RMT Server using this port.

   - Open port 80 to outbound TCP traffic on all Agent computers.

     Periodically, the Agent will make requests to the RMT Server web interface.

   - Open port 443 to outbound TCP traffic on all Agent computers.

     This is the port on which the Agent tests its connection to the RMT Server.

For more information on ports and communications, see Tableau Resource Monitoring Tool Communication Ports.

4. Verify that all Agent computers can contact the RMT Server.

   The web interface hosted by the RMT Server is bound to a specific hostname or IP address during the setup process. It will only answer requests sent to that specific hostname. For example, if the RMT Server is configured to be hosted at http://RMTServer/ but the Agents can only contact the RMT Server through http://myrmt-server.myserver.com/ then the requests will not go through.

5. Gather credentials for a Tableau Server Administrator account.

   Resource Monitoring Tool will use the Tableau Server web API to collect information about the content, so it will need credentials for a user that is a Tableau server administrator and can access all sites, projects, etc. We recommend creating one dedicated to Resource Monitoring Tool, but it can be any user that is a Tableau Server administrator.

6. Enable access to the Tableau Server repository database. **This is required starting in version 2022.3**.

   Resource Monitoring Tool accesses the repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the "readonly" database user. For details, see Enable access to the Tableau Server repository.

7. If you want to use SSL connections between Resource Monitoring Tool and Tableau Server Repository, make sure Tableau Server is configured to use SSL for internal Postgres connections. For more information, see Configure SSL for Internal Postgres Communication.

   The Resource Monitoring Tool allows you to use either the certificate file (`server.crt`) or thumbprint for the SSL connections. The certificate file for the Postgres database is installed to:

`/var/opt/tableau/tableau_server/data/tabsvc/config/pgsql_<version>/security`

If you plan to use the certificate file (`server.crt`), copy the certificate file generated by Tableau Server for internal Postgres SSL connections, to the machine where you plan to install the RMT Server. Make sure that the Operating System trusts the certificate.

8. Confirm that the REST API is enabled on Tableau Server (this is the default). Use the `tsm configuration get -k api.server.enabled` command to do this. A return value of `true` means the REST API is enabled. To enable the REST API, use the `tsm configuration set` command. For more information, see api.server.enabled.

9. Review the size of Tableau Server logs. Once Agent Server is installed and configured, the Resource Monitoring Tool processes relevant historical data from Tableau Server logs before data is displayed. If there is a large amount of historical log data, it may take a while to process the information. This might result in a delay of processing newer events on the Server.

   If you are concerned about the delay, and not having historical information does not concern you, you can do the following to clean up the existing files:

   - Remove Unneeded Files, and consider Log File Snapshots (Archive Logs) before you remove log files.

   Note: Performance data like CPU usage and memory usage are not gathered using historical log data and are collected after Agent is installed and configured so cleaning up historical data does not affect performance data.

10. (optional) Gather connection information for a SMTP server that the RMT Server can access.

If you want email notifications, you need to provide the RMT Server with the server name, username and password (if any), and port number for a SMTP Server that it can use to send the email, and the TLS version. Currently, TSL version 1.2 is required, but TLS version 1.3 is also supported. If you want to use TLS version 1.3, make sure you have Open SSL 1.1.1f or greater on the machine where RMT Server will be installed. For more information, see Email notifications

11. (optional) Verify operating system service account permissions for non-default accounts.

    You may need to do this is your organization has specific security requirements that require you to use already-defined system accounts or users.

    You can find details on the default permissions here: Default Installation Permissions - Tableau Resource Monitoring Tool

Who can do this

To do all the steps described above you need be an admin on the machine that you are installing Resource Monitoring Tool, and be a Tableau Server Administrator.

Minimum Hardware Requirements and Recommendations for Tableau Resource Monitoring Tool

Tableau Resource Monitoring Tool handles all of the processing, aggregation, storage, and web interface to the monitoring data collected from your Tableau Server installation.

RMT Server Minimum Hardware Recommendations

For production use, the machine that you install Resource Monitoring Tool on should meet or exceed the hardware recommendations below:

- 8 physical CPU cores (16 vCPUs)
- CPUs must support SSE4.2 and POPCNT instruction sets
- 64GB RAM
- 500GB Disk Space for RMT Server
- SSD drive or similar for performance

For RMT Server installations that match the minimum hardware specified above, the Resource Monitoring Tool can provide performance reporting for a Tableau Server deployments serving up to 10,000 views per hour.

For deployments with more view loads per hour, or very high numbers of background jobs, data delays may occur. In these cases, you may need to upgrade your hardware.

Resource Monitoring Tool Agent - Resource Utilization

The Resource Monitoring Tool Agent monitors the operating system and Tableau Server processes and log files and sends performance metrics and Tableau log data to the Resource Monitoring Tool for near real time processing. The Agent needs to be installed on every machine that is running Tableau Server.

On Tableau Server installations that match Minimum Hardware Requirements and Recommendations for Tableau Server, here is what to expect:

- We typically see the Agent using 0-5% CPU on average with infrequent spikes above that. Our performance target is an average of 10% CPU usage or less.
- The agent typically uses 200 MB RAM or less with infrequent spikes during periods of heavier Tableau Server activity.
- A minimum of 10 GB free disk space is recommended for the machine where Agent is installed.

Installing in a Cloud Environment

Resource Monitoring Tool can be installed on a virtual machine in any cloud environment that you may be using for your Tableau Servers. The virtual machine where RMT Server is installed will need to meet the same minimum hardware requirements described above.

Who can do this

To install Resource Monitoring Tool, you must be all the following:

- Administrator on the machine you are installing Resource Monitoring Tool.
- Tableau Server Administrator.
- Resource Monitoring Tool Administrator.

Default Installation Permissions - Tableau Resource Monitoring Tool

The following topic provides the default permissions set on various system accounts or groups during installation. If your environment or organization requires you to use non-default accounts, this helps you determine what permissions are necessary for Resource Monitoring Tool (RMT) to function properly.

> **Note:** Resource Monitoring Tool (RMT) server and agent are not Federal Information Processing Standard (FIPS) 140-2 compliant, and will not function properly on operating systems running with FIPS mode enabled.

Windows installations

By default Tableau Resource Monitoring Tool creates these accounts when installed:

- NT SERVICE\TableauResourceMonitoringTool
- NT SERVICE\TableauResourceMonitoringToolPostgreSQL
- NT SERVICE\TableauResourceMonitoringToolRabbitMQ

If you want or need to use other accounts in place of the default accounts, you can specify these after installation using the `rmt-admin` command line utility. For details, see rmtadmin service-setup.

The permissions used by these default service accounts are shown below, as well as the minimum required permissions. Your service accounts should match the minimum permissions or the default permissions:

Service Account	Resource	Default permissions	Required minimum permissions
Tableau			

Service Account	Resource	Default permissions	Required minimum permissions
Resource Monitoring Tool Master			
	Application files `Tableau Resource Monitoring Tool\master`	Full control: read, write execute, modify	Full control
	Log directories and files `Tableau Resource Monitoring Tool\master\logs`	Full control: read, write execute, modify	Read, write
	Configuration directory `Tableau Resource Monitoring Tool\master\config`	Full control: read, write execute, modify	Full conyrol
Tableau Resource Monitoring Tool PostgreSQL			
	Application directory `Tableau Resource Monitoring Tool\-prerequisites\postgresql<nn>`	Full control: read, write execute, modify	Full control
	Data directory	Full control:	Full control

Service Account	Resource	Default permissions	Required minimum permissions
	`Tableau Resource Monitoring Tool\data\postgresql<nn>`	read, write execute, modify	
	Logs directory `Tableau\Tableau Resource Monitoring Tool\master\logs\pgsql`	Full control: read, write execute, modify	Read, write .
Tableau Resource Monitoring Tool RabbitMQ			
	Data directory `Tableau Resource Monitoring Tool\data\rabbitmq`	Full control: read, write execute, modify	Full control

	Application files
	`Tableau Resource Monitoring Tool\prerequisites\rabbitmq`
	`Tableau Resource Monitoring Tool\prerequisites\erlang`
		Full control: read, write execute, modify	Full control

	Log files `Tableau Resource Monitoring Tool\master\logs\rabbitmq`	Full control: read, write execute, modify	Read, write

Service Account	Resource	Default permissions	Required minimum permissions
Tableau Resource Monitoring Tool Agent			
	Application files `Tableau Resource Monitoring Tool\agent`	Full control: read, write execute, modify	Full control
	Log directory and files `Tableau Resource Monitoring Tool\agent\logs`	Full control: read, write execute, modify	Read
	Configuration directory `Tableau Resource Monitoring Tool\agent\config`	Full control: read, write execute, modify	Full control

Linux installations

By default Tableau Resource Monitoring Tool creates certain accounts when installed. If you want or need to use groups and users in place of the defaults, you can specify these after installation using the `rmt-admin` command line utility. For details, see rmtadmin service-setup.

The default permissions used by these service accounts are shown below. Your permissions should match these permissions:

Service Account	**Resource**	**per-missions**
Tableau Resource Monitoring Tool Master		
	Application files `/var/opt/tableau/tabrmt/master`	Owner - Full permissions (read, write, execute) Group - read, write, execute Others - read, execute
	Log directories and files `/var/opt/tableau/tabrmt/master/logs`	Owner - Full permissions (read, write, execute) Group - read, write, execute Others - read, execute
	Configuration directory `/var/opt/tableau/tabrmt/master/config`	Owner - Full permissions

Service Account	Resource	per-missions
		(read, write, execute) Group - read, write, execute Others - read, execute
Tableau Resource Monitoring Tool Post-greSQL		
	Application directory `/opt/tableau/tabrmt/prerequisites/postgresql13`	Owner - Full permissions (read, write, execute) Group - read, execute Others - read, execute
	Service directory `/var/opt/tableau/tabrmt/data/postgresql13/pg_*`	Owner - Full permissions (read, write,

Service Account	Resource	per-missions
		execute) Group - none Others - none
	Data directory `/var/opt/tableau/tabrmt/data/postgresql13/`	Owner - Full permissions (read, write, execute) Group - read, write, execute Others - read, execute
	Logs directory `/var/opt/tableau/tabrmt/master/logs/pgsql`	Owner - Full permissions (read, write, execute) Group - read, write, execute Others - execute
	Other directories under the PostgreSQL directory not mentioned above	Owner - read, write

Service Account	Resource	per-missions
	/var/opt/tableau/tabrmt/data/postgresql13/<not mentioned above>	Group - none Others - none
	/var/opt/tableau/tabrmt/data/postgresql13/base /var/-opt/tableau/tabrmt/data/postgresql13/global	Owner - Full permissions (read, write, execute) Group - none Others - none
	/var/-opt/t-ableau/tabrmt/data/postgresql13/certificates	Owner - Full permissions (read, write, execute) Group - Full permissions (read, write, execute) Others - execute
Tableau Resource Monitoring		

Service Account	Resource	per-missions
Tool Rab-bitMQ		
	Data directory `/var/opt/tableau/tabrmt/data/rabbitmq`	Owner - Full permissions (read, write, execute) Group - read, write, and execute Others - read, execute

	Application files
	`/var/-opt/t-ableau/tabrmt/rabbitmq/prerequisites/rabbitmq`
	`/var/-opt/t-ableau/tabrmt/rabbitmq/prerequisites/erlang`
		Owner - Full permissions (read, write, execute)
		Group - read,ex-ecute
		Others - read, execute

	Log files `/var/opt/tableau/tabrmt/master/logs/rabbitmq`	Owner - Full permissions (read, write, execute) Group -

Service Account	Resource	per-missions
		read, write, and execute Others - execute

	Other directories (under `/rabbitmq` directory)
	
	`/var/-opt/tableau/tabrmt/data/rabbitmq/certificates`
	
	`/var/opt/tableau/tabrmt/data/rabbitmq/mnesia`
		Owner - Full permissions (read, write, execute)
		
		Group - read, write, and execute
		
		Others - execute

	`/var/opt/tableau/tabrmt/data/rabbitmq/<other than mentioned>`	Owner - read, execute Group - read, execute Others - none
Tableau Resource Monitoring Tool Agent		

Service Account	Resource	per-missions
	Application files `/var/opt/tableau/tabrmt/agent`	Owner - Full permissions (read, write, execute) Group - read, write, and execute Others - read, execute
	Log directory and files `/var/opt/tableau/tabrmt/agent/logs`	Owner - Full permissions (read, write, execute) Group - read, write, and execute Others - read, execute
	Configuration directory `/var/opt/tableau/tabrmt/agent/config`	Owner - Full permissions (read, write, execute) Group - read, write, and execute

Service Account	Resource	per-missions
		Others - read, execute

Who can do this

To do all the steps described above you need be an admin on the machine that you are installing Resource Monitoring Tool, and be a Tableau Server Administrator.

Resource Monitoring Tool (RMT) Services

The Resource Monitoring Tool has two main components: Resource Monitoring Tool Master (RMT Server) and RMT Agent.

- The RMT Master is where the data from Tableau Server is collated and served up through a web interface. This is also where you can configure RMT, and monitor and analyze the health and performance of Tableau Server.
- An Agent runs on each of the nodes in your Tableau Server cluster to monitor their performance and activity.

This topic describes the services installed with RMT. The tables below list the services installed on the RMT Server machine, and those installed on the RMT Agent machines.

RMT Server services			
Name	**Purpose**	**Log file location**	**Notes**
		Windows: `C:\Program Files\Tableau\Tableau Resource Monitoring Tool\master\logs\`	

		Linux: `/var/-opt/t-ableau/tabrmt/master/logs` **Note:** For additional information about log files, see: Tableau Resource Monitoring Tool Log Files.	
Host	Used during installation and also to ensure the other processes are running.	Windows: `host\YYYYMMDD.log` Linux: `host/YYYMMDD.log`	Responsibilities: • Process watcher: Makes sure other processes are running. • Config File Watcher: Detects and responds to changes in configuration files.
Backgrounder	Processes the information sent from the	Windows: `background\YYYYMMDD-pts.log` Linux: `background/YYYYMMDD-pts.log`	Responsibilities: • rmtadmin: command

	agents and maintains the postgres data RMT collects. Also provides admin command line utility. Generates incidents and notifications based on information processed.		line utility that exposes command line functionality for administrator. • Agent Data Processing: Handles data collection and processing from various agents • TS Status Polling: Periodically polls Tableau Server (TS) status, ensuring

			the system is active and healthy. • Incident Analysis: Analyzes incidents, like errors, failures, reaching thresholds for different measures. • Data Cleanup: Responsible for cleaning up old data. • Ad Hoc Graph Info Collection: Collects data for on-demand

			for monitoring. • Notifications: Manages sending notifications for alerts and incidents. • Data Archiving (Disabled): involve archiving old data to free up resources or for compliance. This is currently disabled.
Director	Kind of a coordination process, and one of it's	Windows: `director\YYYYMMDD-pts.log` Linux: `director/YYYYMMDD-pts.-log`	Responsibilities: • Cluster Status:

	primary respons-ibilities is to query the Tableau Server for licenses status, and sets the sys-tem wide status.		Monitors the health and status of the server cluster and sets the system wide status. • Licensing: query the Tableau Server for licenses status. • Graph Scraping: Extracts data for monitoring and report-ing. Likely has access to server database. • Database Upgrade:

			Handles updates to the database schema or version • Daily Aggregate Processing: Executes daily processing tasks, like aggregating data for reports or analysis.
Web	Runs the web server. Provides user interface. Registers Agents for information gathering	Windows: `web\YYYYMMDD-pts.log` Linux: `web/YYYYMMDD-pts.log`	Responsibilities: • Agent Registration: Manages the registration of agents that monitor or col-

			lect data on Tableau server. • Config UI: Provides a user interface for configuring various system settings and parameters. • Performance / Activity / Content UI: Offers a UI for users to view performance metrics, activity logs, and content-related data in the

			system.
PostgreSQL	Local data repository	Windows: `pgsql\*.log and *.csv` Linux: `pgsql/*.log and *.csv`	
RabbitMQ	Message broker	Windows: `\master\logs\rabbitmq\*.log` Linux: `master/logs/*.log`	

RMT Agent services			
Name	**Purpose**	**Log file location**	**Notes**
		Windows: `C:\Program Files\Tableau\Tableau Resource Monitoring Tool\master\logs\` Linux: `/var/opt/tableau/tabrmt/master/logs` **Note:** For additional information about log files, see: Tableau Resource Monitoring Tool Log Files.	
Host	Used during installation and also to ensure the other processes are running.	Windows: `host\YYYYMMDD.log` Linux: `host/YYYYMMDD.log`	Responsibilities: • Process watcher: Makes sure other processes

			are running. • Config File Watcher: Detects and responds to changes in configuration files.
Backgrounder	Processes the information sent from the agents and maintains the postgres data RMT collects. Also provides admin commandline utility. Generates incidents and notifications based on information processed.	Windows: `background\YYYYMMDD-pts.log` Linux: `background/YYYMMDD-pts.-log`	Responsibilities: • rmtadmin: command line utility that exposes command line functionality for administrator. • Log watcher: Processes the information sent by Tableau server

			from logs.
Web	Runs the web server. Performs agent registration to master for data publishing.	Windows: `web\YYYYMMDD-pts.log` Linux: `web/YYYMMDD-pts.log`	

Install the Tableau Resource Monitoring Tool

The Tableau Resource Monitoring Tool has two components: RMT Server and Agent.

Resource Monitoring Tool prerequisites and the RMT Server must be installed on dedicated hardware to guarantee they have the necessary resources, and that there is no contention for machine resources with other programs or software. These should not be installed on your Tableau Server computer, except in exceptional cases (for example, for limited demonstration purposes in non-production environments).

Agents are installed on all nodes of Tableau Server that you want to monitor.

The installer for installing Resource Monitoring Tool can be found on the Advanced Management Download site. Download all the files that start with **Tabrmt**.

In an effort to align with our company values of Equality, we have changed non-inclusive terminology where possible. Because changing terms in certain places can causing a breaking change, we maintain the existing terminology. So, you may continue to see the terms in CLI commands and options, installation folders, configuration files. and other instances. For more information, see About Tableau Help.

Installation version history

This section calls out significant changes to the installation process. Use this section to understand what changes have been made depending on your current version and the version you are upgrading to.

- Starting in version **2022.3**, you have the option to configure RabbitMQ messaging service and the PostgreSQL database used by Resource Monitoring Tool on a cloud platform. Currently, AWS AMQ for RabbitMQ, and Amazon RDS for PostgreSQL database are supported. With this addition, you can choose to install the repository and message queue service with RMT Server on the same machine or separately on the AWS platform.
- Starting in version **2020.4**, the Resource Monitoring Tool can be installed on Linux machines. Previously Resource Monitoring Tool installation was only supported on Windows machines.
- Starting in **2020.2**, a single installer will install the prerequisites and the RMT Server. You will need only two installers: One for RMT Server that includes prerequisites, and one for installing the Agent.

Before Installing the Resource Monitoring Tool

- Review the following requirements and instructions:
  - Minimum Hardware Requirements and Recommendations for Tableau Resource Monitoring Tool
  - Pre-Installation Checklist - Tableau Resource Monitoring Tool
  - The following OS is supported:
    - AlmaLinux:

      Version 2023.3.1 and later: AlmaLinux 8.x, AlmaLinux 9.x

- Amazon Linux:

  Version 2023.3.1 and later: Amazon Linux 2, Amazon Linux 2023

  Version 2023.3.0 and earlier: Amazon Linux 2

- CentOS:

  Version 2023.3.1 and later: 7.9 and higher (except 8.x), Stream 8.x, Stream 9.x

  Version 2023.3.0 and earlier: 7.9 and higher (except 8.x)

- Oracle:

  Version 2023.3.1 and later: Oracle 8.x, Oracle 9.x

- Red Hat Enterprise Linux (RHEL):

  Version 2023.3.1 and later: 9.x, 8.3+, 7.3+

  Version 2023.3.0: 9.x, 8.3+, 7.3+

  Version 2023.1.x and earlier: 8.3+, 7.3+

- Rocky Linux:

  Version 2023.3.1 and later: Rocky Linux 8.x, Rocky Linux 9.x

- Ubuntu:

  Version 2023.1.1 and later: the latest versions of Ubuntu 16.04 LTS, 18.04 LTS (not 17.04), 20.04 LTS, 22.04

  Version 2023.1.0: the latest versions of Ubuntu 16.04 LTS, 18.04 LTS (not 17.04), and 20.04 LTS

Version 2022.3 and earlier: the latest versions of Ubuntu 16.04 LTS and 18.04 LTS (not 17.04)

Additional notes on Linux distributions:

- Red Hat Enterprise Linux (RHEL), CentOS, Oracle Linux, and Amazon Linux distributions are collectively referred to in this documentation as RHEL-like.
- Previous versions of CentOS and Ubuntu are not supported because Tableau Server requires systemd for process management.
- The version of the installer with the file suffix, .deb, installs on Ubuntu distributions.

  As of July 2022, Debian distributions are no longer supported. For more information, see this Tableau Community post.

- Custom kernels are not supported.

- Make sure the Tableau Server installation you are going to be monitoring is licensed with the Advanced Managementlicense. You must have a valid Advanced Management license to use the Resource Monitoring Tool.
- Review the size of Tableau Server logs. Once Agent is installed and configured, the Resource Monitoring Tool processes relevant historical data from Tableau Server logs before data is displayed. If there is a large amount of historical log data, it may take a while to process the information which in turn might result in a delay of processing newer events on the Server.

  If you are concerned about the delay, and not having historical information does not concern you, you can do the following to clean up the existing files:

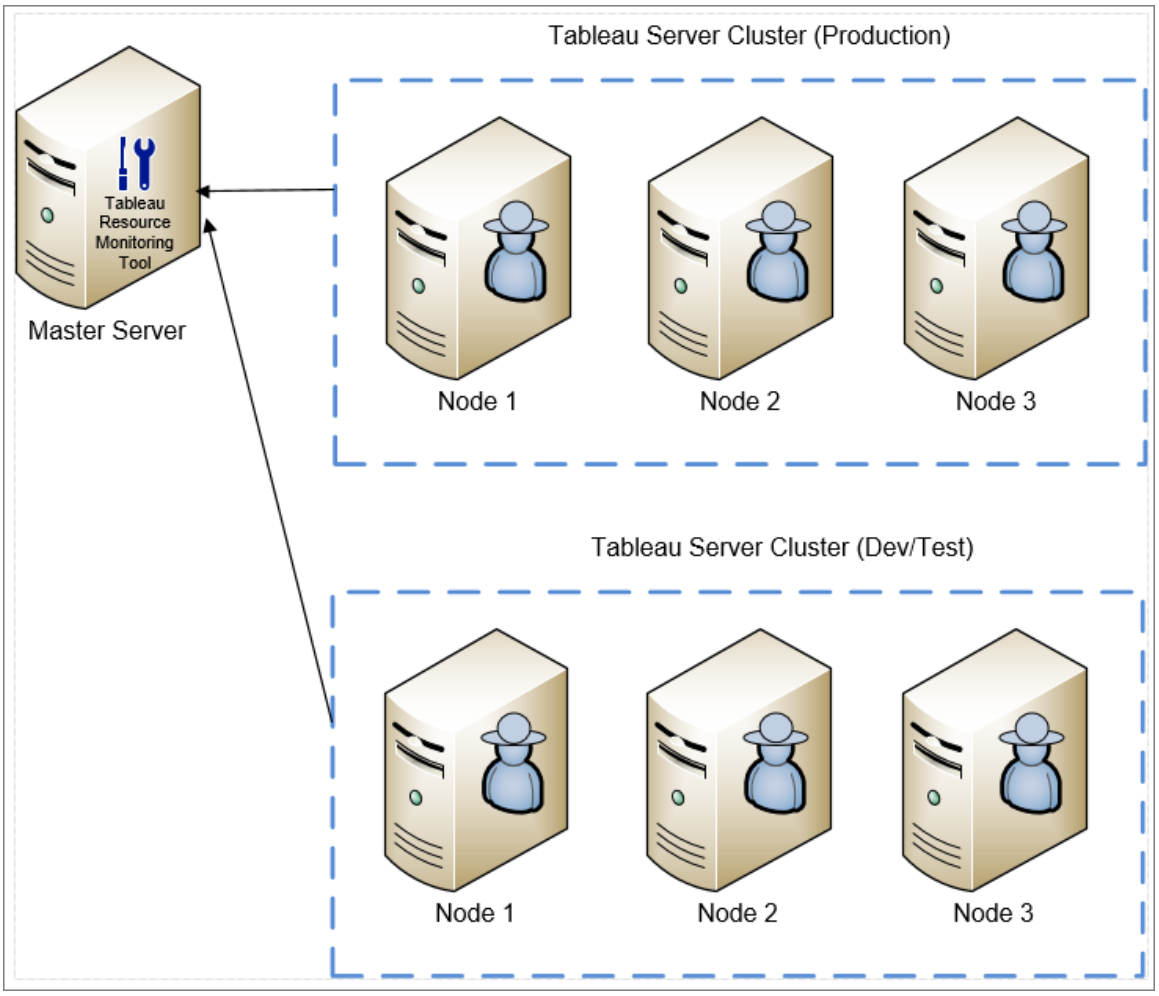
  - Remove Unneeded Files, and consider Log File Snapshots (Archive Logs) before you remove log files.

> **Note:** Performance data like CPU usage and memory usage are not gathered using historical log data and are collected after Agent is installed and configured so cleaning up historical data does not affect performance data.

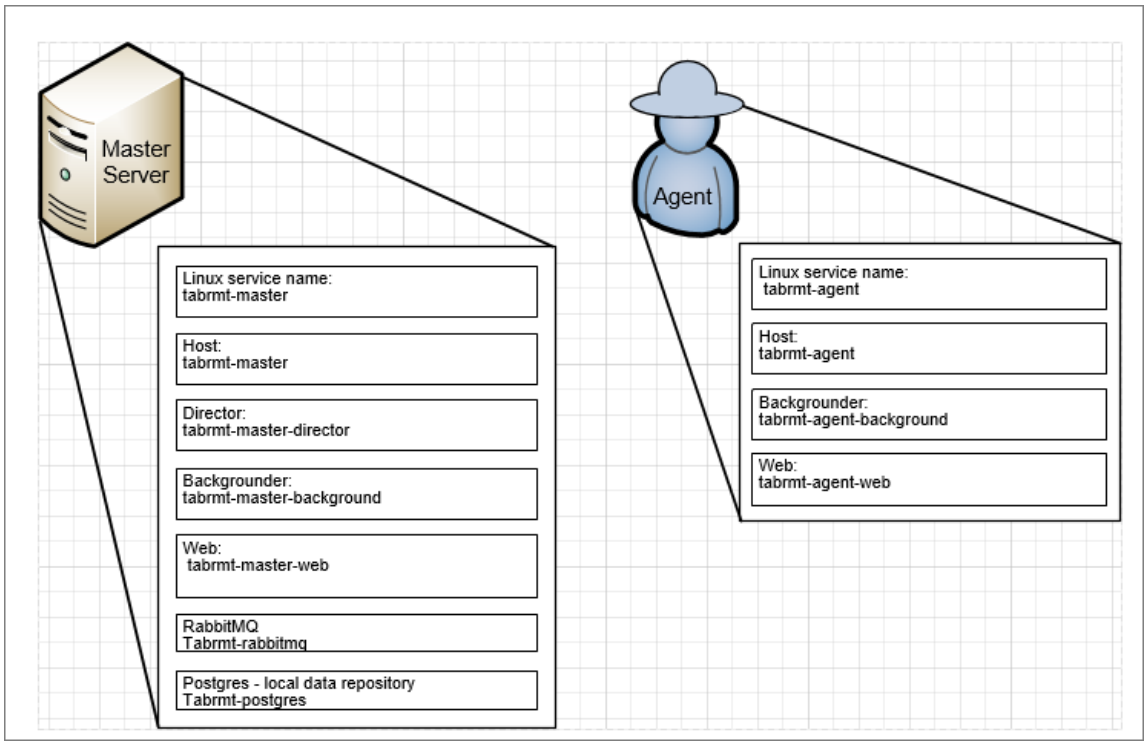
- Enable the readonly built-in user for Tableau Server Repository. This user is used when installing the Resource Monitoring Tool Agent. **This is required starting in version 2022.3.** For more information, see Collect Data with the Tableau Server Repository.
- Determine if you want to configure the repository and message queue service external to Resource Monitoring Tool.

Setup Architecture

A high level view of the Resource Monitoring Tool setup. This diagram shows two Tableau Server clusters monitored by the RMT Server.



A detailed view of the Resource Monitoring Tool components installed on the RMT Server and Agent.



Installation Overview

The RMT Server can be installed on either operating system and does not depend on the operating system that Agent is installed on. This means you can install RMT Server and Agent on different operating systems.

However, Agent is installed on the Tableau Server Cluster nodes, so the operating system choice for the Agent depends on the Tableau Server installation.

You can monitor multiple Tableau Server clusters using the same Resource Monitoring Tool. Each Tableau cluster should be set up as a separate environment.

External Configuration

The repository and the message queue service can both be hosted external to RMT Server on the AWS platform. For hosting the repository, use the AWS RDS instance and for message queue use AWS AMQ. For more details on the configuration, step-by-step instructions, and other details, see:

- External Repository for Tableau Resource Monitoring Tool
- External Message Queue Service (RabbitMQ) for Tableau Resource Monitoring Tool

Installation on Linux

You can install RMT Server and Agent using the command line on Linux operating systems.

Here are the steps to installing Resource Monitoring Tool:

1. Install the RMT Server using command line.

   When you start the Resource Monitoring Toolinstaller, the setup program will install certain programs required to run Resource Monitoring Tool first. The programs installed are RabbitMQ, Erlang, and a dedicated PostgreSQL database. It will then proceed to install the RMT Server.

2. Install the Agent using command line.

   Install the Agent on every one of your Tableau Server nodes. The Agent sends information about Tableau Server usage and performance to the RMT Server for reporting.

HTTPS

As a best practice, you should use HTTPS to protect sensitive information and user credentials.

The **Require HTTPS** setting in the Server configuration is used for communications between the users and the RMT Server. It is also used when you register an Agent. Regular agent communications between Agent and RMT to collect data is done through Rabbit MQ.

Initially, the RMT Server is installed with a self-signed certificate and will use that certificate for HTTPS communication which includes communication during Agent registration. You can use your own certificate to replace the self-signed certificate. This can be done during RMT Server install in the Server Configuration page or after the installation is complete.

SSL Certificate Mode and Requirements

The resource monitoring tool supports the following modes of using SSL Certificates:

- **Default:** This mode uses the default self-signed certificate supplied by the installer.
- **Local:** This mode allows you to specify a file-based certificate in the `/var/-opt/tableau/tabrmt/master/config` folder.

Follow these guidelines and requirements for your certificate:

- You must have a HTTPS certificate (like X.509) for the appropriate domains. This depends on your local security policies and certificate requirements. For example, if the Resource Monitoring Tool is using a CName or SSL passthrough proxy then you might need to use a SAN certificate. For multiple sub-domains, wildcard certificates are supported.
- The Resource Monitoring Tool supports only PKCS #12 and PEM formats.
- The Resource Monitoring Tool web server requires a certificate and a private key, and optionally chain-of-trust.

  The private key can be either RSA or DSA.

  These can be provided in a single file or grouped files.

  - Single file examples:
    - PKCS #12: A single file with the .pfx or .p12 file extensions.
    - PEM: PEM-encoded certificate + private key (plus optionally intermediate CAs chaining up to root CA), in a single file with the .pem extension. The items in the file does not have to be in any specific order.
  - Grouped file examples:
    - PEM-encoded certificate in a .crt or .cer file PLUS
    - PEM-encoded private key in a .key file PLUS (optionally)
    - PEM-encoded certificate authority in one or more .ca files

Default File and Directory locations:

RMT Server:

- Installation directory: `/opt/tableau/tabrmt/master`
- Configuration file: `/var/opt/tableau/tabrmt/master/config.json`
- Logs: `/var/opt/tableau/tabrmt/master/logs`

Agent:

- Installation directory: `/opt/tableau/tabrmt/agent`
- Bootstrap: `/var/opt/tableau/tabrmt/agent/bootstrap`
- Logs: `/var/opt/tableau/tabrmt/agent/logs`

Who can do this

To install Resource Monitoring Tool, you must have all the following:

- User account with full sudo access.
- Tableau Server Administrator site role.
- Resource Monitoring Tool Administrator account.

Next Step

Install the RMT Server:

- Using web interface
- Using command line

Install the RMT Server Using Web Interface

The Resource Monitoring Tool Server (RMT Server) hosts the web application that users interact with. It also does much of the background processing to collate and monitor the data from the Agents. The RMT Server must be installed on dedicated hardware.

> Installation through web interface is currently only supported for Windows Installations. If you want to install on Linux, see Install the RMT Server Using Command Line.

To install the RMT Server:

There are three main steps to installing the RMT Server:

- Installing the Server software
- Configuring the RMT Server
- Creating a new environment.

The detailed step by step process is described below:

1. **Installing the Server software:**

    1. Run the RMT Server setup program.

    2. After reading the EULA, select **I agree to the license terms and conditions**, and click **Install**.

    3. If the User Account Control dialog opens, click **Yes** to allow the installer to make changes.

    4. The RMT Server installer will first install certain prerequisites as the first step. The prerequisites include RabbitMQ, Erlang, and a PostgreSQL database. The PostgreSQL database is used to store usage data gathered from Tableau Server. It will then proceed to install the RMT Server.

    5. At the end of the installation, a **Server Configuration** web page opens.

2. **Configure the RMT Server:**

    1. The page should already be filled in. Make any updates to the values if needed.

        Note: Make a note of the **Host Name**. The web interface used to access the RMT Server uses this format: *https://<hostname>*.

    2. By default, Resource Monitoring Tool uses a self-signed certificate to use for the HTTPS communications with the RMT Server. To leverage your own certificate, replace the thumbprint value with the one for the certificate you want to use. Check the **Require HTTPS** option if you want to mandate secure HTTPS communications.

        Checking the **Require HTTPS** option will also require you to choose a certificate mode and provide additional details if necessary. Mainly there are three options to choose from:

    1. **Default:** This mode uses the default self-signed certificate supplied by the installer.

    2. **Local:** This mode allows you to specify a file-based certificate in the Resource Monitoring Tool**/config** folder. When **Local** mode is selected, the **Certificate Name** field becomes available. The options listed will correspond with the certificate file groups located in the **/config** folder.

    3. **Store:** This mode allows you to enter the thumbprint of a certificate in the Windows certificate store.

       The **Password** field will only be used if the selected certificate requires a password.

3. Test the Server Configuration by clicking **Test Server Configuration**.

4. The **Password** section allows you to configure the password requirements that will apply to the user accounts and will be applied to the Administrative user account you will create a little later.

5. In the **Authentication** section, you can configure the timeout period for the sessions. If the user does not access the session for the set timeout period then the session will expire and they will need to log in again. By default, this is set to 240 minutes. You can also enable the **Sliding Expiration** option to reset the time out period when a session is accessed within the time out period.

6. Click **Save and Restart Server**.

7. When the server has restarted, you are prompted to create an administrative user and password.

8. After signing in using the administrative credentials you just created in the previous step, the Resource Monitoring Tool you will see a message that no Environment has been created. Click **Add an environment** to setup a new

environment.

3. **Create a New Environment:**

    1. Create a new environment: Fill in the details of the environment, like the name an identifier. The Tableau Server REST API and the Tableau Server Repository configurations are used to communicate with Tableau Server. The Tableau Server Repository configuration is optional, but is a preferred method to access Tableau Server.

        You have the option to configure secure encrypted connection when RMT connects to Tableau Server Repository. In order to use SSL connections between RMT and Tableau Server Repository database, Tableau Server must be configured to use SSL. For more information, see Configure SSL for Internal Postgres Communication.

    2. Tableau Repository Configuration:

        In the **Tableau Repository Configuration** section:

        1. In the **SSL Mode** drop down box, select **Prefer SSL** or **Require SSL** to configure SSL connections to Tableau Repository. Choosing **Disable** means SSL will never be used to make Tableau Server Repository connections.

            In the **Prefer SSL** mode, the Resource Monitoring Tool will use SSL in the first attempt, and if that fails the subsequently attempts a non-encrypted connection.

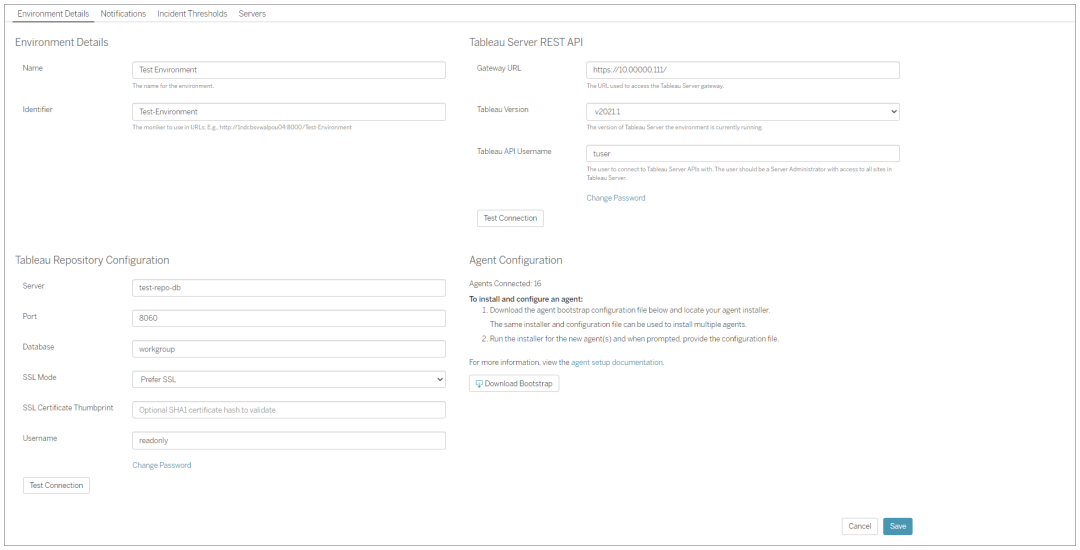
            In the **Require SSL** mode, if the SSL connection fails, the connections to Tableau Server Repository will fail entirely. In this case, Tableau Server REST API connections will be used to communicate with Tableau Server.

        2. You can choose to either supply the thumbprint that was generated by Tableau Server, or copy the **server.crt** file to the Resource Monitoring

Tool Master Server machine. If you choose to copy the certificate file, you don't have to supply the thumbprint. For more information, see Configure Postgres SSL to Allow Direct Connections from Clients.

4. Click **Save**. You will see a new section added to the page - **Agent Configuration**. This section allows you to download the bootstrap file needed to install and configure Agents.

   Note: If you make any updates to the Environment configurations, you must click **Save** before downloading the Bootstrap file.



Who can do this

To install Resource Monitoring Tool, you must have all the following:

- Administrator permissions on the machine you are installing Resource Monitoring Tool.
- Tableau Server Administrator site role.
- Resource Monitoring Tool Administrator account.

Next Step

Install the Agent Using the Web Interface

Install the Agent Using the Web Interface

The Agent is a lightweight process that consumes minimal server resources and sends data to the Resource Monitoring Tool Server (RMT Server). Install the Resource Monitoring Tool Agent on each of your Tableau Server nodes. To install and register an Agent, download the Agent bootstrap configuration file and save it to a location that is accessible from the Resource Monitoring Tool Agent nodes.

> Installation through web interface is currently only supported for Windows Installations. If you want to install on Linux, see Install the Agent Using Command Line.

Before you install

- Download the bootstrap file. **Bootstrap files are only valid for 24 hours after downloading. You will need to regenerate the bootstrap file if the one you are using is older than 24 hours.**
- Starting in version 2021.3, Agent registration will need to communicate both through a https endpoint and RabbitMQ to complete the Agent registration. Make sure both ports 443 and 5672 are open for these communications.

Steps to download the Agent bootstrap file

Use the following steps if you have not yet downloaded the bootstrap file from the RMT Server.

1. Using the web interface (*https://<hostname>*) on the RMT Server, from the **Admin** menu, select **Environments**.
2. The bootstrap file can be downloaded directly from the environment overview tab on the home page.

   > **Note:** If you haven't created an environment as part of the RMT Server setup, follow steps 12- 15 described in the Install the RMT Server Using Web Interface topic.

3. Save the bootstrap configuration file to a location that is accessible from the Tableau Server nodes where you will be installing Resource Monitoring Tool Agent.

Steps to install Agent

To install the Agent on each of your Tableau Server nodes:

1. Run the Agent setup program.
2. After reading the EULA, select **I agree to the license terms and conditions**, and click **Next**.
3. On this page, you will have the option of changing the install location and specify the Run As User account for Agent.

   If you are planning to install to a non-default location, use the guidelines provided in the Installing to a Non-Default Location. The default location is **C:\Program Files\Tableau\Tableau Resource Monitoring Tool\agent**.

   Starting in 2021.4, you must specify a Run As User account for Agent. This account is used to access Tableau Server for gathering monitoring information from Tableau Server nodes.

   For the Agent Run As User account, you must specify the same account that you currently use for the Tableau Server Run As User account. If the account information you provide is not the same as that on Tableau Server, Agent will not be able to gather the monitoring data on that node.
4. If the **User Account Control** dialog opens, click **Yes** to allow the installer to make changes.
5. When the installer is finished an **Agent Registration** web page opens.

   Tip! It may take a while for the web page to open. If the web page fails to open for some reason, use the following URL on the machine that you are installing:

   http://localhost:9002/setup/register

6. Review the information on this page about the Tableau Server log file size, historical data, and the implications. Once you proceed to the next step, you will no longer see this information.

   Once Agent is installed and registered, the Resource Monitoring Tool processes relevant historical data from Tableau Server logs before data is displayed. If there is a large amount of historical log data, it may take a while to process the information which in turn might result in a delay of processing newer events on the Server.

   If you are concerned about the delay, and not having historical information does not concern you, you can do the following to clean up the existing Tableau log files:

   Remove Unneeded Files, and consider Log File Snapshots (Archive Logs) before you remove log files.

   > **Note:** Performance data like CPU usage and memory usage are not gathered using historical log data and are collected after Agent is installed and configured so cleaning up historical data does not affect performance data.

7. Browse to the location of the bootstrap file you downloaded from the RMT Server. Click **Import Bootstrap File**. **Bootstrap files are only valid for 24 hours after downloading. You will need to regenerate the bootstrap file if the one you are using is older than 24 hours.**

8. Once the import is successfully completed, a web page to enter the Server information is displayed. In the **Tableau Server gateway URL** field, enter the URL you use to access Tableau Server.

   Here are some examples on what the URL might look like:

   - https://MarketingServer/ (if you know the server's name)
   - https://10.0.0.2/ (if you know the server's IP address)
   - http://10.0.0.4/ (If your Tableau Server is not enabled to use SSL)

9. Enter the user name and password. You can use the admin user credentials that you created when you installed the RMT server. This user name and password is used for communications between RMT Server and Agent.

10. Click **Test Tableau Server Connection** to verify the Agent is able to reach Tableau Server.

    A **Success** message displays to confirm the Agent can connect to Tableau.

11. The **RMT Server URL** field should already be populated from the information in the bootstrap file. This is the URL of the web page used to do administration tasks, monitor performance and other tasks.

    Here is an example of what the URL might look like:

    - *https://<hostname>*)

12. Click **Test RMT Server Connection** to verify the Agent is able to reach the Resource Monitoring Tool RMT Server.

    A **Success** message displays to confirm the connection to the RMT Server works.

13. The **RMT Server certificate thumbprint** field should be already be populated using the information in bootstrap file. The bootstrap file you saved should have this and the RMT Server URL information. Enter the RMT Server user name and password. This user must have the **Server/Environment Management** role.

    Click **Get Registration Options**.

    This takes you to the **Agent Registration - Message Queue** page.

14. Click **Test Message Queue Connection** to verify the message queue connection is working.

    A **Success** message displays if the connection works.

> **Note:** The **Enable TLS** setting under the message queue section allows you to enable encryption when data is transmitted between the RMT Server and the Agents. It requires additional RabbitMQ setup. For more information, see Encrypted Data Collection.

15. On the final Agent Registration page the environment section should already be filled out. Verify the information and add any Tableau Server node details.
16. Click **Connect to Message Queue**.
17. Click **Register Agent** to complete the agent installation and configuration.

    You will see an option to disable the web interface. After registration the web interface is no longer necessary for the agent and can optionally be disabled. If you need to re-enable the web interface, use the following command:

    ```
    rmtadmin set server.web.run true
    ```

    For more information on Resource Monitoring Toolcommands, see rmtadmin Command Line Utility
18. You can verify that the Agent is connected by navigating to the RMT Server web interface. From the **Admin** menu, select **Environments**, and under environment details you can see the number of Agents that are currently connected.

Installing Agent on a Multi-Node Tableau Server Installation

1. Follow the steps described above to install the Agent on the Tableau Server initial node to connect to the Resource Monitoring Tool.
2. In the **Server** section on the final registration page, check the **Primary Server** option.
3. On the Tableau Server additional nodes, install the Agent using the same bootstrap file that you used to install Agent on the initial node and follow the steps described above.
4. On the final registration page, in the **Environment** section, select the environment you created when installing the Agent on the initial node.
5. In the **Server** section, select the **New Environment Server** option. The Primary Server option should remain unselected.

6. The agent registration process automatically adds this node to your existing environment.

Installing to a Non-Default Location

Tableau recommends using *\Tableau\Tableau Resource Monitoring Tool\agent* as the location for the prerequisites. Example non-default location: **D:\Tableau\Tableau Resource Monitoring Tool\agent**.

To choose a non-default location during installation, use the following steps:

1. Run the Agent installer.
2. On the EULA page, choose **Customize**.
3. Under **Setup Options**, in the **Install location** field, enter the location.
4. Continue with Step 2 of the installation as described Install the Agent Using the Web Interface.

Who can do this

To install Resource Monitoring Tool, you must have all the following:

- Administrator permissions on the machine you are installing Resource Monitoring Tool.
- Tableau Server Administrator site role.

Next Steps

Resource Monitoring Tool Server Configuration

Install the RMT Server Using Command Line

The Resource Monitoring Tool Server (RMT Server) hosts the web application that users interact with. It also does much of the background processing to collate and monitor the data from the Agents. The RMT Server must be installed on dedicated hardware.

This topic describes the steps you can use to install the RMT Server using command line. Command line installation is supported on both Windows and Linux operating systems.

In an effort to align with our company values of Equality, we have changed non-inclusive terminology where possible. Because changing terms in certain places can causing a breaking change, we maintain the existing terminology. So, you may continue to see the terms in CLI commands and options, installation folders, configuration files. and other instances. For more information, see About Tableau Help.

Install on Linux

To install the RMT Server:

1. **Install the RMT Server:**

    1. Download the rpm or .deb RMT Server Installer and save it to a location that you can access from the machine where you plan to install the RMT Server.

    2. Login to the machine where you want the RMT Server installed as a user that has sudo access.

    3. Run the following command to install the RMT Server, where <version> is formatted as major-minor-maintenance:

        For RHEL like distribution including CentOS:

        ```
        sudo yum install <pathtormtserverinstaller>/Tabrmt-Master-x86_64-<version>.rpm
        ```

        For Ubuntu distributions:

        ```
        sudo apt install ./<pathtormtserverinstaller>/Tabrmt-Master-amd_64-<version>.deb
        ```

    This installs the package and the prerequisites including RabbitMQ, Erlang, and a PostgreSQL database. The PostgreSQL database is used to store usage data gathered from Tableau Server. It will then proceed to install the RMT Server.

2. **Initialize RMT Server:**

You must explicitly accept the End User License Agreement (EULA) when you initialize RMT Server. You also have the option to specify non-default configurations. To initialize RMT Server with a default configuration, run this command :

```
sudo /opt/tableau/tabrmt/master/install-scripts/initialize-rmt-master --accepteula
```

The EULA can be found in the `/opt/tableau/tabrmt/master/docs` folder.

Beginning in version 2023.1 you can specify a custom Run As account to be used by RMT, as well as other configuration options. By default, RMT creates and uses an account called `rmt-master` to run under. To specify a custom Run As account to be used by RMT Server, include the `--unprivileged-user` option when you run the initialization script. For information about all the available switches for the `initialize-rmt-master` script, see RMT Server Initialization Script Options.

3. **Configure the RMT Server:**

    1. Run the following command as the `tabrmt-master` user:

        ```
        sudo su --login tabrmt-master
        ```

        ```
        rmtadmin master-setup [options]
        ```

        The configuration options can be supplied either through the command prompt, a configuration file. If you do not supply the options, the default values will be applied except for the administrator password. The administrator username will be set to admin and you will be prompted to provide the password.

        Example command including the required password parameter:

        ```
        rmtadmin master-setup --admin-username=<name of the administrator user> --admin-password=<administrator user password>
        ```

The following table lists the required and some commonly used options to configure the RMT Server. For a full list of the configurations options, see rmtadmin Command Line Utility .

Note: Require HTTPS option ensures secure communications between the RMT Server and users. When you require HTTPS for communications, you must also select a mode for the certificate that should be used for these communications. The table below includes the various options. To learn more about these modes and certificates, see SSL Certificate Mode and Requirements

Option	Required?	Default	Description
admin-password	Yes Password can be supplied in the command line or provide a file with the password to use. If neither is provided, you will be prompted for the password.	n/a	The password for the administrator user.
admin-password-file	No Password can be supplied in	n/a	The file where the password for the administrator user is stored.

Option	Required?	Default	Description
	the command line or provide a file with the password to use. If neither is provided, you will be prompted for the password.		**Note:** `tabrmt-master` user must have access to this file.
admin-username	No	admin	The username for the administrator user.
http-port	No	80	
require-https	No	False	Redirect http traffic to HTTPS.
https-certificate-mode	No	**"Default"** Available options: Default Local	The type of certificate search to perform for the HTTPS certificate. **Default:** This mode uses the default self-signed certificate supplied by the installer. **Local:** Allows you to specify a file-based certificate in the */var/opt/tableau/tabrmt/master/config* folder.
https-cer-	No	Null	The name of the HTTPS cer-

Option	Required?	Default	Description
tificate-local-name	Note: If not specified, the Resource Monitoring Tool is installed with a self-signed certificate and will use that certificate for HTTPS communications.		tificate file without the file extension.
https-certificate-local-password	No	Null	The password to use for the HTTPS certificate.
https-certificate-local-password-file	No	Null	The path to the file containing the password to use for the HTTPS certificate.

4. **Create an environment:**

    1. Run the following command to create an environment:

    ```
    rmtadmin create-env --name=<myenvironment> --api-user-name=<TableauServer API user name> --api-password=<password for the Tableau Server API user account>
    ```

    Configure the environment using the options available for this command. Here are some key configuration options to consider:

- The Tableau Server REST API and the Tableau Server Repository configurations are used to communicate with Tableau Server. The Tableau Server Repository configuration is optional, but is a preferred method to access Tableau Server.
- You have the option to configure secure encrypted connection when RMT connects to Tableau Server Repository. In order to use SSL connections between RMT and Tableau Server Repository database, Tableau Server must be configured to use SSL. For more information, see Configure SSL for Internal Postgres Communication.

The following table lists the some of the common options. To see a full list of options, see rmtadmin Command Line Utility .

Option	Required?	Default	Description
--name	Yes	n/a	The name of the environment.
--gateway-url	Yes	n/a	URL used to access the Tableau Server gateway.
--version	Yes	n/a	Tableau Server version that this environment will be monitoring.
--api-username	No	Null	User name of the account used to connect to Tableau Server APIs. The user account should be a Tableau Server administrator with access to all Tableau Server sites.
--api-password	No (If you specify the	Null	Password of the Tableau Server API user account used to connect to Tableau Server APIs.

Option	Required?	Default	Description
	Tableau API user name, you will either provide the password, or specify the file path and file that has the password)		
--api-password-file	No	Null	The path to the file and the name of the file containing the password of the Tableau Server API user account.
--repository-server	Yes	Null	This is the server name for the PostgreSQL database that in installed with Tableau Server
--repository-port	Yes	Null	The port number of the Tableau Server Repository database.
--repository-username	Yes	Null	Username used to connect to PostgreSQL database installed with the Tableau Server Repository. Resource Monitoring Tool accesses the Tableau Server Repository database directly for

Option	Required?	Default	Description
			performance reasons. For this to work, access to the repository must be enabled, with a password set for the **readonly** database user. For details, see Enable access to the Tableau Server repository.
--repository-password	Yes	Null	Password for the user account used to connect to the PostgreSQL database that is installed with the Tableau Server . Resource Monitoring Tool accesses the Tableau Server Repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the **readonly** database user. For details, see Enable access to the Tableau Server repository.
--repository-password-file	No	Null	The path including the file name where the password for the user account used to connect to the PostgreSQL database that is installed with Tableau Server.
--repository-ssl-mode	No	Prefer	Tableau Server Repository SSL Mode:

Option	Required?	Default	Description
			Prefer SSL or **Require** SSL to configure SSL connections to Tableau Repository. **Disable** to never use SSL to make Tableau Server Repository connections.
--repository-ssl-thumbprint	No	Null	You can choose to either supply the thumbprint that was generated by Tableau Server, or copy the **server.crt** file to the Resource Monitoring Tool Server (RMT Server) machine. If you choose to copy the certificate file, you don't have to supply the thumbprint. For more information, see Configure Postgres SSL to Allow Direct Connections from Clients.

2. Download the bootstrap file to a location that can be accessed from the Tableau Server nodes.

   ```
   rmtadmin bootstrap-file --env=<myenvironment> --file-name=<The absolute or relative path including the file name>
   ```

5. **Optional step - only if not using SSD:** The Resource Monitoring Tool is optimized for SSD by default. If you are not using SSD hardware, run the command:

   ```
   sudo /opt/tableau/tabrmt/master/tabrmt-master optimize --no-ssd
   ```

To install the RMT Server:

1. **Run the RMT Server setup program:**

    1. Download the RMT Server installer and save it to a location that you can access from the machine where you plan to install the RMT Server.

    2. Open the command prompt as an administrator.

    3. Navigate to the location of the RMT Server installer and run the exe file using the following command

        ```
        Tabrmt-Master-64bit-<version>.exe /silent ACCEPTEULA=1
        ```

        Note: Use the /silent or /quiet switch to run the setup unattended and without displaying any UI or prompts.

        The default install folder is usually like this: *C:\Program Files\Tableau\Tableau Resource Monitoring Tool\master*. To provide a different install location run the following command:

        ```
        Tabrmt-Master-64bit-<version>.exe /silent ACCEPTEULA=1
        InstallFolder="D:\Tableau Resource Monitoring Tool\master"
        ```

        Full list of all the install command properties and switches can be found in this section below.

    This installs the package and the prerequisites including RabbitMQ, Erlang, and a PostgreSQL database. The PostgreSQL database is used to store usage data gathered from Tableau Server. It will then proceed to install the RMT Server.

2. **Configure the RMT Server:**

    1. Run the following command and provide the options:

        ```
        rmtadmin master-setup [options]
        ```

The configuration options can be supplied either through the command prompt, a configuration file. If you do not supply the options, the default values will be applied except for the administrator password. The administrator user name will be set to admin and you will be prompted to provide the password.

Example command including the required password parameter:

```
rmtadmin master-setup --admin-username=<name of the admin-
istrator user> --admin-password=<administrator user pass-
word>
```

The following table lists the required and some commonly used options used to configure the RMT Server. For a full list of the configurations options, see rmtadmin Command Line Utility .

Note: Require HTTPS option ensures secure communications between the RMT Server and users. When you require HTTPS for communications, you must also select a mode for the certificate that should be used for these communications. The table below includes the various options. To learn more about these modes and certificates, see Install the Tableau Resource Monitoring Tool

Option	Required?	Default	Description
admin-password	Yes Password can be supplied in the command line or provide a	n/a	The password for the administrator user.

Option	Required?	Default	Description
	file with the password to use. If neither is provided, you will be prompted for the password.		
admin-password-file	No Password can be supplied in the command line or provide a file with the password to use. If neither is provided, you will be prompted for the password.	n/a	The file where the password for the administrator user is stored.
admin-username	No	admin	The username for the administrator user.

Option	Required?	Default	Description
http-port	No	80	
require-https	No	False	Redirect http traffic to HTTPS.
https-certificate-mode	No	**Default** Available options: • Default • Store • Local	The type of certificate search to perform for the HTTPS certificate. **Default:** This mode uses the default self-signed certificate supplied by the installer. **Store:** This allows you to enter the thumbprint of a certificate in the Windows certificate store. **Local:** Allows you to specify a file-based certificate *<installation directory>\config* folder. By default this is *C:\Program Files\Tableau\Tableau Resource Monitoring Tool\master\config*.
https-certificate-store-thumbprint	No	Null	The HTTPS certificate hash/thumbprint to search for in 'store' certificate mode.

Option	Required?	Default	Description
https-certificate-local-name	No	Null Note: If not specified, the Resource Monitoring Tool is installed with a self-signed certificate and will use that certificate for HTTPS communications.	The name of the HTTPS certificate file without the file extension.
https-certificate-local-password	No	Null	The password to use for the HTTPS certificate.
https-certificate-local-password-file	No	Null	The path to the file containing the password to use for the HTTPS certificate.

3. **Create an environment:**

    1. Run the following command to create an environment:

        ```
        rmtadmin create-env --name=<myenvironment> --api-user-name=<TableauServer API user name> --api-password=<password for the Tableau Server API user account>
        ```

        Configure the environment using the options available for this command. Here are some key configuration options to consider:

- The Tableau Server REST API and the Tableau Server Repository configurations are used to communicate with Tableau Server. The Tableau Server Repository configuration is optional, but is a preferred method to access Tableau Server.
- You have the option to configure secure encrypted connection when RMT connects to Tableau Server Repository. In order to use SSL connections between RMT and Tableau Server Repository database, Tableau Server must be configured to use SSL. For more information, see Configure SSL for Internal Postgres Communication.

The following table lists the some of the common options. To see a full list of options, see rmtadmin Command Line Utility .

Option	Required?	Default	Description
--name	Yes	n/a	The name of the environment.
--gateway-url	Yes	n/a	URL used to access the Tableau Server gateway.
--version	Yes	n/a	Tableau Server version that this environment will be monitoring.
--api-username	No	Null	User name of the account used to connect to Tableau Server APIs. The user account should be a Tableau Server administrator with access to all Tableau Server sites.
--api-password	No (If you specify the	Null	Password of the Tableau Server API user account used to connect to Tableau Server APIs.

Option	Required?	Default	Description
	Tableau API user name, you will either provide the password, or specify the file path and file that has the password)		
--api-password-file	No	Null	The path to the file and the name of the file containing the password of the Tableau Server API user account.
--repository-server	Yes	Null	This is the server name for the PostgreSQL database that in installed with Tableau Server
--repository-port	Yes	Null	The port number of the Tableau Server Repository database.
--repository-username	Yes	Null	Username used to connect to PostgreSQL database installed with the Tableau Server Repository. Resource Monitoring Tool accesses the Tableau Server Repository database directly for

Option	Required?	Default	Description
			performance reasons. For this to work, access to the repository must be enabled, with a password set for the **readonly** database user. For details, see Enable access to the Tableau Server repository.
--repository-password	Yes	Null	Password for the user account used to connect to the PostgreSQL database that is installed with the Tableau Server . Resource Monitoring Tool accesses the Tableau Server Repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the **readonly** database user. For details, see Enable access to the Tableau Server repository.
--repository-password-file	No	Null	The path including the file name where the password for the user account used to connect to the PostgreSQL database that is installed with Tableau Server.
--repository-	No	Prefer	Tableau Server Repository

Option	Required?	Default	Description
ssl-mode			SSL Mode: **Prefer** SSL or **Require** SSL to configure SSL connections to Tableau Repository. **Disable** to never use SSL to make Tableau Server Repository connections.
--repository-ssl-thumbprint	No	Null	You can choose to either supply the thumbprint that was generated by Tableau Server, or copy the **server.crt** file to the Resource Monitoring Tool Server(RMT Server) machine. If you choose to copy the certificate file, you don't have to supply the thumbprint. For more information, see Configure Postgres SSL to Allow Direct Connections from Clients.

2. Download the bootstrap file to a location that can be accessed from the Tableau Server nodes.

```
rmtadmin bootstrap-file --env=<myenvironment> --file-name=<The absolute or relative path including the file name>
```

Windows install properties and switches

Switches:

Switch	Description	Comments
`/install | /uninstall`	Run Setup to either install or uninstall Resource Monitoring Tool.	Default is to install, displaying UI and all prompts. If no directory is specified using the InstallFolder property on a fresh install, *C:\Program Files\Tableau\Tableau Resource Monitoring Tool\master* is assumed. If Resource Monitoring Tool is already installed, Setup will assume the same location as the current installation. To completely remove Resource Monitoring Tool including the data directory, use, `/uninstall DELTEDATADIR=1`
`/passive`	Run Setup with minimal UI and no prompts.	
`/quiet | /silent`	Run Setup in unattended, fully silent mode. No web interface or prompts are displayed.	Use either /quiet or /silent, not both.
`/norestart`	Run Setup without restarting Windows, even if a restart is necessary.	In certain rare cases, a restart cannot be suppressed, even when this option is used. This is most likely when an earlier system restart was skipped. For example, if restart was skipped during installation of other software.
`/log <log-file>`	Log information to the specified file and path. By default log files are created in %TEMP% with a nam-	If no file location is specified, the log file is written to the TEMP folder - *C:\Users\<username>\AppData\Local\Temp*. Check this log file for errors after installation.

Switch	Description	Comments
	ing convention of Tableau_Resource_Monitoring_Tool_<version_code>.	For example: `<Setup file> /quiet /log="C:\Tableau\Logs\RmtInstall`

Properties:

Property	Description	Comments
`InstallFolder=<path\to\installation\directory>`	Install to the specified non-default install location.	Specifies the location to install RMT. If not used, RMT is installed to *C:\Program Files\Tableau\Tableau Resource Monitoring Tool\master*. Example: `<Setup file> /silent InstallFolder=r="D:\Tableau\Tableau Resource Monitoring Tool\master"`
`ACCEPTEULA=1|0`	Accept the End User License Agreement (EULA). Required for quiet, silent,	If not included when using /passive, /silent or /quiet, Setup fails silently. If included but set to 0, Setup fails.

Property	Description	Comments
	and passive install on both initial and additional nodes. 1 = true, accept the EULA, 0 = false, do not accept the EULA.	

Who can do this

To install Resource Monitoring Tool, you must have all the following:

Windows

- Administrator permissions on the machine you are installing Resource Monitoring Tool.
- Tableau Server Administrator site role.
- Resource Monitoring Tool Administrator account.

Linux

- Full sudo access for the user account that is used to install the Agent.
- Resource Monitoring Tool Administrator account

Next Step

Install the Agent Using Command Line

RMT Server Initialization Script Options

After installing the RMT Server you need to initialize the Server. By default the only required flag you must include when running the initialization script is `--accepteula`. Other options give you flexibility to customize the installation based on your environment and the security requirements of your enterprise.

--accepteula

Required.

Indicates you have read and accepted the terms of the End User License Agreement (EULA).

`-a <username>`

Optional.

Add the provided username to the appropriate groups, instead of the user running the initialize script. This gives the user access to the resources owned by the groups. This is not the same as the Run As user account.

Default: the user running the initialization script

`-f`

Optional.

Bypass warning messages or distribution version check.

`-h | -?`

Optional.

Displays the script's help text.

`-q`

Optional.

Quiet, suppress output except for errors and warnings.

`--debug`

Optional.

Print each command as it is run for debugging purposes. Produces extensive output.

`--default-group=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the group that allows authorization to access Tableau RMT Agent. If specified then `--unprivileged-user` must also be specified.

`---disable-account-creation`

Version: Added in version 2023.1.0.

Optional.

Accounts/groups, that do not exist, will not be created. If specified, then you must also specify the `--unprivileged-user` parameter and a combination of the `--default-group` and/or other `--rmt-<...>-group` parameters. The user ID and groups those parameters refer to must already exist.

`--rmt-authorized-group=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the group that owns the install base install directory and the `install_dir/-prerequisites` folder. If specified then `--unprivileged-user` must also be specified.

Default: "`rmtmasterapp`" or `--default-group` value

`--rmt-config-group=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the group that owns the `install_dir/master/config` directory. If specified then `--unprivileged-user` must also be specified.

Default: "`rmtmasterconfig`" or `--default-group` value

`--rmt-logs-group=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the group that owns the `install_dir/master/logs` directory. If specified then `--unprivileged-user` must also be specified.

Default: "`rmtmasterlogs`" or `--default-group` value

`--rmt-openssl-group=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the group that owns the `install_dir/prerequisites/openssl` directory.

Default: "`rmtopenssl`" or `--default-group` value

`--rmt-postgres-app-group=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the group that owns the `install_dir/prerequisites/postgresql13` directory.

Default: "`rmtpostgresapp`" or `--default-group` value

`--rmt-postgres-data-group=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the group that owns the install_dir/master/logs directory. If specified then `--unprivileged-user` must also be specified.

Default: "`rmtmasterlogs`" or `--default-group` value

`--rmt-rabbitmq-app-group=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the group that owns the `install_dir/prerequisites/rabbitmq` directory.

Default: "`rmtrabbitmqapp`" or `--default-group` value

`--rmt-rabbitmq-data-group=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the group that owns the `install_dir/data/rabbitmq` directory.

Default: "`rmtrabbitmqdata`" or `--default-group` value

`--unprivileged-user=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the unprivileged user account to run Tableau RMT Master. You cannot change the account after initializing RMT.

Default: "`tabrmt-master`"

Install the Agent Using Command Line

The Agent is a lightweight process that consumes minimal server resources and sends data to the Resource Monitoring Tool Server (RMT Server). Install the Resource Monitoring Tool Agent on each of your Tableau Server nodes. To install and register an Agent, download the Agent bootstrap configuration file and save it to a location that is accessible from the Resource Monitoring Tool Agent nodes.

This topic describes the steps you can use to install the Resource Monitoring Tool Agent using command line. Command line installation is supported on both Windows and Linux operating systems.

In an effort to align with our company values of Equality, we have changed non-inclusive terminology where possible. Because changing terms in certain places can causing a breaking change, we maintain the existing terminology. So, you may continue to see the terms in CLI commands and options, installation folders, configuration files. and other instances. For more information, see About Tableau Help.

Before you install

- Download the bootstrap file and save it to a location that is accessible to the nodes on which you are going to install RMT Agent. **Bootstrap files are only valid for 24 hours**

after downloading. You will need to regenerate the bootstrap file if the one you are using is older than 24 hours.

- Starting in version 2021.3, Agent registration will need to communicate both through a https endpoint and RabbitMQ to complete Agent registration. Make sure both ports 443 and 5672 are open for these communcations.

Install on Linux

To install and configure the Agent:

1. **Install:**

    1. Download the .rpm or .deb Agent Installer and save it to a location that you can access from the machine where you plan to install the Agent.

    2. Run the following command to install the Agent where <version> is formatted as major-minor-maintenance:

        For RHEL like distributions including CentOS:

        ```
        sudo yum install Tabrmt-Agent-x86_64-<version>.rpm
        ```

        For Ubuntu:

        ```
        sudo apt install Tabrmt-Agent-amd64-<version>.deb
        ```

2. Initialize RMT Agent:

    You must explicitly accept the End User License Agreement (EULA) when you initialize RMT Agent. You also have the option to specify non-default configurations. To initialize RMT Agent with a default configuration, run this command :

    ```
    sudo /opt/tableau/tabrmt/agent/install-scripts/initialize-rmt-agent --accepteula
    ```

    The EULA can be found in the `/opt/tableau/tabrmt/agent/docs` folder.

Beginning in version 2023.1 you can specify a custom Run As account to be used by RMT, as well as other configuration options. By default RMT creates and uses an account called `rmt-agent` to run under. To specify a custom Run As account to be used by RMT Agent, include the `--unprivileged-user` option when you run the initialization script. For information about all the available switches for the `initialize-rmt-agent` script, see RMT Agent Initialization Script Options.

3. **Register:**

    1. Log off and log on as the *tabrmt-agent* user so you can run rmtadmin commands which always require that you run as the *tabrmt-agent* user. Also,when you log on again, you create a new session in which group membership changes have taken effect.

        ```
        sudo su --login tabrmt-agent
        ```

    2. Run the following command and provide the path where the bootstrap file is located. Provide a description of the node where the Agent is being installed.

        ```
        rmtadmin register <bootstrap file path\file> --server-name-e=<Friendly name of machine> --server-description=<server description> --username=<name of the RMT admin user>
        ```

        You will be prompted for the password of the RMT admin user.

        > **Note:** The `tabrmt-agent` user defaults to run commands from the base working directory: /var/opt/tableau/tabrmt/agent, so you must specify the file path accordingly. For example, if you placed the bootstrap file in the /var/-opt/tableau/tabrmt/agent/bootstrap/ folder as recommended, the file path would be `/var/-opt/tableau/tabrmt/agent/bootstrap/<bootstrap_file_name>`.

The following table lists the configuration options used to register the Agent:

Option	Required?	Default	Description
--bootstrap file	Yes	<none>	The location of the bootstrap file.
--username	Yes	<none>	This is typically the admin user you created during RMT Server installation.
--password	Yes	<none>	This is the password for the user account
--password-file	No Password can be supplied in the command line or a file that contains the password. If neither is provided, you will be prompted for the password.	<none>	Path including the file name where the password is stored.
-- server-name	No	Host name of machine	Name of the computer that has the Agent Installed. If no option is provided, this field will default to the host name of the machine.
--server-description	No	<none>	Description of the computer that has

Option	Required?	Default	Description
			the Agent installed. If no option is provided, this field will remain blank.

Install on Windows

To install and configure the Agent:

1. Download the bootstrap file to a location that can be accessed from the Tableau Server nodes.

   ```
   rmtadmin bootstrap-file --env=<myenvironment> --filename<The absolute or relative path including the file name>
   ```

2. Run the Agent setup program:

   1. Download the Agent installer and save it to a location that you can access from Tableau Server machines.
   2. Open the command prompt as an administrator.
   3. Navigate to the location of the Agent installer and run the **exe** file using the following command

      ```
      Tabrmt-Agent-64bit-<version>.exe /silent ACCEPTEULA=1
      ```

      Note: Use the /silent or /quiet switch to run the setup unattended and without displaying any UI or prompts.

      The default install folder is usually like this: *C:\Program Files\Tableau\Tableau Resource Monitoring Tool\agent*. To provide a different install location run the following command:

```
Tabrmt-Agent-64bit-<version>.exee /silent ACCEPTEULA=1
InstallFolder="D:\Tableau Resource Monitoring Tool\agent"
```

3. Register the Agent:

    1. Run the following command to register the Agent:

    ```
    rmtadmin register <bootstrap file path\file> --server-name-
    e=<Friendly name of machine> --server-description=<server
    description>
    ```

    The following table lists the configuration options used to register the Agent:

Option	Required?	Default	Description
--bootstrap file	Yes	<none>	The location of the bootstrap file.
--username	Yes	<none>	This is typically the admin user you created during RMT Server installation.
--password	Yes	<none>	This is the password for the user account
--password-file	No Password can be supplied in the command line or a file that contains the password. If neither is provided, you will be prompted for the password.	<none>	Path including the file name where the password is stored.
-- server-	No	Host name	Name of the com-

Option	Required?	Default	Description
name		of machine	puter that has the Agent Installed. If no option is provided, this field will default to the host name of the machine.
-- server-description	No	<none>	Description of the computer that has the Agent installed. If no option is provided, this field will remain blank.

Windows install properties and switches

Switches:

Switch	Description	Comments
`/passive`	Run Setup with minimal UI and no prompts.	
`/quiet | /silent`	Run Setup in unattended, fully silent mode. No web interface or prompts are displayed.	Use either /quiet or /silent, not both.

Properties:

Property	Description	Comments
`InstallFolder=<path\to\i-`	Install	Specifies the location to

Property	Description	Comments
`nstallation\directory>`	to the specified non-default install location.	install RMT. If not used, RMT is installed to *C:\Program Files\Tableau\Tableau Resource Monitoring Tool\master*. Example: `<Setup file> /silent InstallFolder="D:\Tableau\Tableau Resource Monitoring Tool\agent"`
`ACCEPTEULA=1|0`	Accept the End User License Agreement (EULA). Required for quiet, silent, and pass-	If not included when using /passive, /silent or /quiet, Setup fails silently. If included but set to 0, Setup fails.

Property	Description	Comments
	ive install on both initial and additional nodes. 1 = true, accept the EULA, 0 = false, do not accept the EULA.	

Installing Agent on Multi-Node Tableau Server

Run the steps described above on each of the nodes of Tableau Server. On the web interface of the RMT Server you should be able to see all the nodes where the Agent is installed.

Who can do this

To install Resource Monitoring Tool, you must have all the following:

Windows:

- Administrator permissions on the machine you are installing Resource Monitoring Tool.
- Tableau Server Administrator site role.

Linux:

- Full sudo access for the user account that is used to install the Agent.

Next Step

Configure Tableau Resource Monitoring Tool

RMT Agent Initialization Script Options

After installing the RMT Agent you need to initialize the Agent. By default the only required flag you must include when running the initialization script is `--accepteula`. Other options give you flexibility to customize the installation based on your environment and the security requirements of your enterprise.

RMT Agent initialization options

--accepteula

Required.

Indicates you have read and accepted the terms of the End User License Agreement (EULA).

`-a <username>`

Optional.

Add the provided username to the appropriate groups, instead of the user running the initialize script. This gives the user access to the resources owned by the groups. This is not the same as the Run As user account.

Default: the user running the initialization script

`-f`

Optional.

Bypass warning messages or distribution version check.

`-h | -?`

Optional.

Displays the script's help text.

`-q`

Optional.

Quiet, suppress output except for errors and warnings.

`--debug`

Optional.

Print each command as it is run for debugging purposes. Produces extensive output.

`--default-group=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the group that allows authorization to access Tableau RMT Agent. If specified then `--unprivileged-user` must also be specified.

`---disable-account-creation`

Version: Added in version 2023.1.0.

Optional.

Accounts/groups, that do not exist, will not be created. If specified, then you must also specify the `--unprivileged-user` parameter and a combination of the `--`

`default-group` and/or other `--rmt-<...>-group` parameters. The user ID and groups those parameters refer to must already exist.

`--rmt-authorized-group=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the group that owns the install base install directory and the `install_dir/-prerequisites` folder. If specified then `--unprivileged-user` must also be specified.

Default: "`rmtagentapp`" or `--default-group` value

`--rmt-config-group=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the group that owns the `install_dir/agent/config` directory. If specified then `--unprivileged-user` must also be specified.

Default: "`rmtagentconfig`" or `--default-group` value

`--rmt-logs-group=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the group that owns the `install_dir/agent/logs` directory. If specified then `--unprivileged-user` must also be specified.

Default: "`rmtagentlogs`" or `--default-group` value

`--unprivileged-user=<value>`

Version: Added in version 2023.1.0.

Optional.

Name of the unprivileged user account used to run Tableau RMT Agent. You cannot change the account after initializing RMT.

Default: "`tabrmt-agent`"

External Repository for Tableau Resource Monitoring Tool

The Tableau Resource Monitoring Tool uses a PostgreSQL database as its repository to gather and store Tableau Server monitoring data that is used to provide performance and usage information. This database is separate from the Tableau Server database.

Before the 2022.3 release, this PostgreSQL repository was automatically installed with the RMT Server, on the same machine as the RMT Server and this was the only option available. Starting in version 2022.3, you can reconfigure RMT Server to use an externally hosted PostgreSQL database instead of the locally installed database. When RMT Server is configured to use an externally hosted PostgreSQL database, we refer to it as the external repository.

The supported platform for hosting the external repository is an AWS RDS instance. Hosting the repository database externally allows you to manage the availability, backups, and disaster recovery of the repository component of the RMT Server.

With the addition of this option to configure an external repository, RMT Server can be configured in the following ways:

- **Continue to use the locally installed repository:** This means that the PostgreSQL database that is automatically installed with RMT Server on the same machine will be used as the repository for RMT Server, and no further changes are made to this configuration.
- **Configure RMT Server to use an external repository:** This means that the PostgreSQL database that RMT uses is hosted external to RMT Server. This requires

making some configuration changes post installation. Currently only AWS RDS instances are supported as the hosting platform for PostgreSQL database.

In order to setup and manage the external repository, you should have a good understanding of the AWS RDS platform and PostgreSQL database. We recommend, you use the documentation on the AWS site for platform related instructions.

You will also need to know how to configure and manage the external repository configuration for Resource Monitoring Tool. This topic will provide you with that information in the sections below.

New installation of Resource Monitoring Tool

The instructions detailed in this section apply to a new installation of Resource Monitoring Tool. If you have an existing deployment of RMT, and want to move to an external repository configuration, see the Existing Resource Monitoring Tool installation section of this topic.

When you install RMT Server, it automatically installs the repository database on the same machine as the RMT Server. The process to configure the external repository is a post install configuration where you will reconfigure RMT Server to use an externally hosted PostgreSQL database.

Use the following steps to install Resource Monitoring Tool and reconfigure RMT Server to use an external repository:

1. **Create the external repository:** Create an RDS instance to host the external repository with the following recommendations.
    - **Minimum recommended PostgreSQL instance specifications:**
        - db.m6g.2xlarge (8 vcpu, 32 GiB RAM)
        - SSD Storage with 500 GB disk space

- **Instance configuration values:**
  - Max Worker Processes: Total Logical Processors
  - Max Workers Per Gather: Min(Total Logical Processors / 2, 4)
  - Max Parallel Workers: Total Logical Processors
  - Max Parallel Maintenance Workers: Min(Total Logical Processors / 2, 4)
  - Shared Buffers: Total Memory / 4
  - Effective Cache Size: 75% of Total Memory
  - Maintenance Work Memory: Total Memory / 16 Linux, Max 2GB - 1MB in Windows
  - Wal Buffers: Derived from Shared Buffers, likely around 16 MB
  - Work Memory: Derived from Shared Buffers and Max Workers Per Gather

  To learn how to create and connect to a PostgreSQL DB instance on AWS, see this page on AWS documentation site.

2. **Install RMT:** Follow the instructions Install the RMT Server Using Command Line to Install RMT Server, but **skip the steps for creating an environment. You will do that later after configuring RMT Server to use the external repository**. **Note that this will initially install a local PostgreSQL database.**

3. **Test the connection to the external repository:** Test the connection to the new database using psql, a PostgreSQL administrative tool that is installed with the RMT Server;

   - Navigate to the 'bin' folder of the locally installed PostgreSQL installation directory.

     ```
     /var/opt/tableau/tabrmt/prerequisites/postgresql<version number>/bin
     ```

- Run the following command to test the connection. Substitute your own value below for **aws_rds_servername**, and provide the PostgreSQL password you created in AWS when prompted.

  ```
  psql -h <aws_rds_servername> -p 5432 -d postgres -U post-
  gres
  ```

  If you receive an error at this point and cannot connect successfully, review the parameter values in the command against the values from the AWS RDS console.

- Once the connection has been successfully established, you can close the psql session with the following command:

  ```
  \q
  ```

4. Run `rmtadmin master-setup` **to configure RMT Server to use the external repository**. Use the following command examples, and edit the command to reflect your Resource Monitoring Tool installation path, AWS RDS instance name, port number, and the RDS PostgreSQL admin user password:

   - Navigate to the Resource Monitoring Tool RMT Server installation directory:

     ```
     sudo /var/opt/tableau/tabrmt/master
     ```

   - Run the command to configure external repository:

     ```
     rmtadmin master-setup --db-config=external --db-server-
     r=<aws_rds_servername> --db-database=<aws_rds_database_
     name> --db-port=5432 --db-admin-username=postgres --db-
     admin-password=<postgres_user_password>
     ```

5. Now **create an environment** and download the bootstrap file:

- Run the following command to create and environment:

  ```
  rmtadmin create-env --name=<myenvironment> --api-user-
  name=<TableauServer API user name> --api-password=<password
  for the Tableau Server API user account>
  ```

- Download the bootstrap file to register Agents:

  ```
  rmtadmin bootstrap-file --env=<myenvironment> --file-
  name=<The absolute or relative path including the file
  name>
  ```

  The steps are described in full detail here: Install the RMT Server Using Command Line.

6. Follow the instructions Install the Agent Using Command Line to Install and register Agents on Tableau Server Nodes.

Existing Resource Monitoring Tool installation

In this release, to configure external repo, you will essentially have to start with a new installation of Tableau Resource Monitoring Tool using the steps described in the above section.

Upgrade best practices

Here are the general steps you need to follow if you want to upgrade Tableau Resource Monitoring Tool and migrate to an external repository at the same time:

Migration with environment recreation:

1. Upgrade RMT Server and all Agents to 2022.3 or later.
2. Steps through 2-4 are very similar to the steps you would take to do a new install - see New installation of Resource Monitoring Tool above. This mainly involves:

   - Create an AWS PostgreSQL DB Instance.
   - Configure RMT Server to use the external repository.

- Recreate environments and re-register all the Agents.

3. Reconfigure any custom configurations.

Note: You will lose historical data and also need to reconfigure any custom configurations.

Upgrading when the new version of RMT requires a major version PostgreSQL upgrade

When there is a change in the PostgreSQL major version requirement for Resource Monitoring Tool, it is a best practice to upgrade the external repository PostgreSQL version first before you upgrade Resource Monitoring Tool. More details are provided below. To see if you should upgrade your external repository PostgreSQL version, see the product compatibility table.

Use the following steps to upgrade RMT and the RDS instance when a PostgreSQL major version upgrade is required

1. Make a backup of your RDS instance. You will need this in case you need to roll back the upgrade. For more information, see Backing up and restoring an Amazon RDS DB instance topic on the AWS site.

2. Make a copy of the configuration file in the Resource Monitoring Tool directory. The configuration file is located at:

   ```
   /var/opt/tableau/tabrmt/master/config.json
   ```

3. Upgrade the RDS instance to the new version of PostgreSQL For more information, see Upgrading the PostgreSQL DB engine for Amazon RDS topic on the AWS site.

4. Upgrade RMT Server. If the upgrade including the database migration completes successfully, proceed to the next step. If the upgrade fails, see the instructions in this

section on how to recover and roll back the upgrade.

5. Upgrade all the Agents on Tableau Server nodes to the new RMT version. For more information, see Upgrading Resource Monitoring Tool.

Recovering from a failed upgrade

1. Uninstall the upgraded RMT Server.
2. Restore the AWS RDS instance to the version previous to the upgrade. For more information, Backup up and restoring an Amazon RDS DB instance topic on the AWS site.
3. Replace the config file that you backed up prior to upgrade in the following location. You may have to create this folder as it might have been deleted during uninstall:

   ```
   /var/opt/tableau/tabrmt/master/config.json
   ```

4. Install RMT Server which installs a local repository.
5. Configure RMT Server to use the external repository:

   ```
   rmtadmin master-setup --db-config=external --db-server=<aws_
   rds_servername> --db-database=<aws_rds_database_name> --db-port-
   t=5432 --db-admin-username=postgres --db-admin-pass-
   word=<postgres_user_password>
   ```

RMT and PostgreSQL version compatibility

This table lists RMT version 2022.3 and later only, since the external repository is only available from version 2022.3 and later.

RMT Version	PostgreSQL Version shipped with RMT	Supported PostgreSQL Version for external repository
2022.3 -	13.7	13.7

2024.2		

Who can do this

To install Resource Monitoring Tool, you must have all the following:

- User account with full sudo access.
- Tableau Server Administrator site role.
- Resource Monitoring Tool Administrator account.

External Message Queue Service (RabbitMQ) for Tableau Resource Monitoring Tool

The Tableau Resource Monitoring Tool uses RabbitMQ as its message queue service to collect data from Agents and bring them to the RMT Server. This information in the queue is processed and eventually stored in the RMT repository (PostgreSQL database).

Before the 2022.3 release, the RabbitMQ message queue service was automatically installed with the RMT Server and this was the only configuration available. Starting in RMT version 2022.3, you can reconfigure RMT Server to use an externally hosted RabbitMQ service. When RMT Server is configured to use an externally hosted message queue service, we refer to it as the external message queue service.

With this new option added in version 2022.3, RMT Server can be configured in the following ways:

- **Continue to use the locally installed message queue service:** This means that RabbitMQ that is automatically installed with RMT Server on the same machine will be used as the message service for RMT Server, and no further changes are made to this configuration.
- **Configure RMT Server to use an external message queue service:** This means that RAbbitMQ that RMT uses is hosted external to RMT Server. Currently only AWS AMQ is supported as the hosting platform for RabbitMQ. Since all messages from the Agents go through RabbitMQ, by hosting this externally frees up resources from the machine where RMT Server is installed.

To setup and manage the external message queue service, you should have a good understanding of the AWS AMQ platform. We recommend reviewing the documentation on AWS

site. You will also need to know how to configure and manage the external message queue service for RMT. This topic will provide you with that information in the sections below.

New installation of Resource Monitoring Tool

The instructions provided in this section apply to a new installation of Resource Monitoring Tool. If you have an existing installation, and want to move your local RabbitMQ to an externally hosted configuration, see the Existing installations of Tableau Resource Monitoring Tool section of this topic.

Use the following steps to install Tableau Resource monitoring tool and reconfigure RMT Server to use an external message queue service:

1. **Create Amazon AMQ for Rabbit MQ** to host the external message queue service with the following recommendations:

    - For engine type, use RabbitMQ engine. For the version of Rabbit MQ, see the Product compatibility section.
    - Use the same default version as the one used when RabbitMQ is installed locally. For more information, see the product compatibility table.
    - Use a single-Instance broker.
    - Instance specification: mq.m5.large, 2 vCPU/8 GiB RAM.
    - Create a RabbitMQ username/password.

    To learn more, see Working with Amazon MQ for Rabbit MQ on AWS documentation site.

2. **Test the connection from RMT Server** to the message broker by copying the RabbitMQ web console URL from the AWS MQ page, and paste it into a web browser on RMT Server. Log in via the username and password you created when you set up the broker.

3. Follow the instructions in this topic to **install RMT Server**, but **skip the steps for creating an environment. You will do that later after configuring RMT Server to use the external repository**.

4. Run rmtadmin setup as follows to configure the external Rabbit MQ message queue service:

   ```
   rmtadmin master-setup --mq-config=external --mq-server=aws_amq_
   servername --mq-vhost='/' --mq-port=5671 --mq-username=aws_amq_
   username --mq-password='aws_amq_password' --mq-tls-certificate-
   host=aws_amq_servername
   ```

5. Now create an environment and download the bootstrap file.

   - Run the following command to create and environment: `rmtadmin create-env --name=<myenvironment> --api-username=<TableauServer API user name> --api-password=<password for the Tableau Server API user account>`

   - Download the bootstrap file to register Agents: `rmtadmin bootstrap-file --env=<myenvironment> --filename<The absolute or relative path including the file name>`

     The steps are described in full detail in the Install the RMT Server Using Command Line topic.

6. Re-register Agents on Tableau Server Nodes using the instructions in Install the Agent Using Command Line topic.

Existing installations of Tableau Resource Monitoring Tool

In this release, to configure external messaging service, you will essentially have to start with a new installation of Tableau Resource Monitoring Tool using the steps described in the above section.

Upgrade best practices

Here are the general steps you need to follow if you want to upgrade to a version 2022.3 or later and migrate to using an external message queue service at the same time.

Migration with environment recreation:

1. Upgrade RMT Server and all Agents to 2022.3 or later
2. Create an Amazon AMQ broker service.
3. Configure RMT Server to use the external message queue service
4. Recreate environments and re-register all the Agents
5. Recreate any custom configurations.

Note: You will lose some event and hardware processing data and you will also need to reconfigure the incident thresholds

Upgrade steps with enabling TLS for RabbitMQ

Since the agents in versions earlier than 2022.3 have been communicating using unencrypted connection to the RabbitMQ message queue service, when upgrading to version 2022.3 or later, those agents need to be updated to use the new secure connection details. The steps to do this are as follows:

1. After completing the upgrade steps described in the above section, stop all agents by running the following command:

   ```
   rmtadmin stop --agent
   ```

2. Download the bootstrap file for the environment by running the following command:

   ```
   rmtadmin bootstrap-file --env<myenvironment> --filename <The absolute or relative path including the file name>
   ```

3. Run the following command on each of the machines where the Agent is installed:

```
rmtadmin rotate-mq-certificate <BOOTSTRAP_FILE> --username=<RMT
Server Username> --password-file=<RMT Server Password file
name>
```

4. Restart each Agent machine after successfully running the `rmtadmin rotate-mq-certificate` command.

Product Compatibility

This table lists RMT version 2022.3 and later only, since the external message queue is only available from version 2022.3 and later.

RMT Version	RabbitMQ version shipped with RMT	Supoort RabbitMQ version for external message queue service
22.3	3.10.5	3.10.5

Who can do this

To install Resource Monitoring Tool, you must have all the following:

- User account with full sudo access.
- Tableau Server Administrator site role.
- Resource Monitoring Tool Administrator account.

Tableau Resource Monitoring Tool Prerequisites - Licenses

Tableau Resource Monitoring Tool contains the following open source applications:

RMT Version	OPENSSL Version	ERLANG	RABBITMQ	POSTGRESQL
2020.4	1.1.1h	22.3	3.8.3	12.2
2021.1	1.1.1h	23.1	3.8.9	12.4
2021.2	1.1.1i	23.2.6	3.8.14	12.5
2021.3	1.1.1k	23.3.1	3.8.14	12.6
2021.4	1.1.1l	24.0.3	3.8.19	12.6

RMT Version	OPENSSL Version	ERLANG	RABBITMQ	POSTGRESQL
2022.1	1.1.1l	24.1.2	3.9.7	13.3
2022.3	1.1.1q	24.3.4.2	3.10.5	13.7

- **Erlang:** Copyright 2016 Industrial Erlang User Group, Apache 2.0. For more information, see the Erlang and Apache 2.0 sites.
- **RabbitMQ:** Copyright , MPL 2.0, Copyright (c) 2007-2021 VMware, Inc. or its affiliates. For more information, see Mozilla Public License site.

  Resource Monitoring Tool contains a single modified RabbitMQ source file, which is available upon request.
- **PostgreSQL:** Copyright Portions Copyright © 1996-2021, The PostgreSQL Global Development Group, Portions Copyright © 1994, The Regents of the University of California, PostgreSQL license. For more information, see PostgreSQL site.
- **OpenSSL:** Copyright (c) 1998-2019 The OpenSSL Project, Copyright (c) 1995-1998 Eric A. Young, Tim J. Hudson, OpenSSL license, double license under both the OpenSSL License and the original SSLeay license. For more information, see Open SSL License.

Upgrading Resource Monitoring Tool

These instructions are for upgrading an existing installation of Tableau Resource Monitoring Tool.

In an effort to align with our company values of Equality, we have changed non-inclusive terminology where possible. Because changing terms in certain places can causing a breaking change, we maintain the existing terminology. So, you may continue to see the terms in CLI commands and options, installation folders, configuration files. and other instances. For more information, see About Tableau Help.

Note: The Resource Monitoring Tool performs an in-place upgrade, upgrading your current installation to the newer version. Do not uninstall your existing installation **before** upgrading.

Upgrade Notes

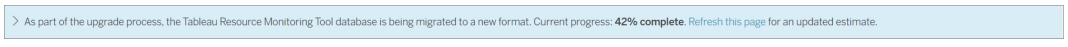
Consider the following before you start the upgrade process:

1. If you are planning to move to use an external repository (PostgreSQL) or external message queue service (RabbitMQ), make sure you review the upgrade best practices section in the following topics:
    - Upgrade best practices for external repository
    - Upgrade best practices for external message queue

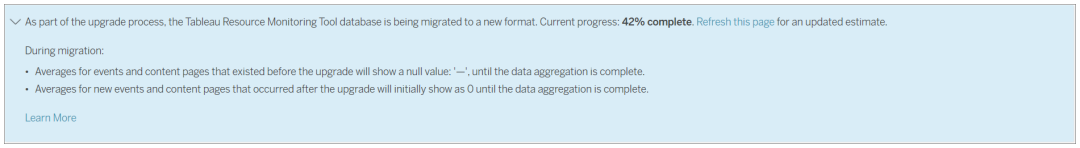
2. Starting in version 2022.3, Tableau Resource Monitoring Tool has built-in encrypted communications between RMT Server and Agents. However, you will need to run `rmtadmin rotate-mq-certificates` on all the Agents to enable TLS. This applies to RabbitMQ that is configured locally on the same machine as RMT Server. For more details on how to do this, see **Upgrade steps with enabling TLS for RabbitMQ** below.

3. You may need to reboot the machine during an upgrade. This usually occurs if files are locked and cannot be updated by the installer. If necessary, you will be prompted by the installer. Because of this possibility, you may want to consider doing your upgrades during non-work hours.

4. Sometimes, the Resource Monitoring Tool will modify the database and in such cases, the upgrade process will include a database migration. In case of a database migration, you will see a message banner at the top of the RMT Server web page with a time estimate for completing the process. An example screen shot is shown below:



On expanding the banner you can review the details of the impact of the upgrade. Changes are unique to each release and the details shown are specific to the release.



5. The Resource Monitoring Tool is backward compatible with Tableau Server versions, but not forward compatible. This means the Resource Monitoring Tool version should be equal to or greater than the Tableau Server version you are monitoring.

   For compatibility reasons, Tableau recommends you upgrade the Resource Monitoring Tool first and then follow with a Tableau Server upgrade. For more information, see Product Compatibility with Tableau Server.

How to Upgrade the Resource Monitoring Tool

We recommend you do an in-place upgrade of Resource Monitoring Tool. If you uninstall Resource Monitoring Tool and install a new version of the database, it may result in data corruption and you will be prompted to reinstall the previous version and re-do the upgrade process.

1. Log in to the Resource Monitoring Tool in a browser and confirm it is working before starting the upgrade.
2. Copy the new version of the RMT Server package to the RMT Server machine.
3. Copy the new version of the RMT Agent package to the machines where you have RMT Agents installed. RMT Agents are installed on Tableau Server nodes you are monitoring.
4. On each Tableau Server node running the RMT Agent service, switch to the tabrmt-agent user and stop the RMT Agent service:

```
sudo su --login tabrmt-agent
rmtadmin stop
exit
```

5. Switch to the tabrmt-master user on the RMT Server and stop the RMT Server service using the following command:

   ```
   sudo su --login tabrmt-master
   rmtadmin stop
   exit
   ```

6. Wait until there are not any running processes with tabrmt in the name before continuing to the next step.

   Once the services have been stopped, it is best practice to check for any Resource Monitoring Tool processes that are running after the services have been stopped: Any with tabrmt-agent or tabrmt-master. This does not include PostgreSQL or RabbitMQ. You can check the status using the following command:

   ```
   rmtadmin status
   ```

7. Run the upgrade commands on the RMT Server. This will upgrade the existing version to the new version:

   For RHEL-like distributions including CentOS:

   ```
   sudo yum install <pathtomasterserverinstaller>/<tabrmt-master-setup-<version>-x86_64.rpm>
   ```

   ```
   sudo /opt/tableau/tabrmt/master/install-scripts/upgrade-rmt-master --accepteula
   ```

   For Ubuntu distributions:

   ```
   sudo apt install <pathtomasterserverinstaller>/<tabrmt-master-setup-<version>-amd_64.deb>
   ```

```
sudo /opt/tableau/tabrmt/master/install-scripts/upgrade-rmt-mas-
ter --accepteula
```

8. Once the RMT Server has been upgraded, upgrade all the RMT Agents by running the following command:

   For RHEL-like distributions including CentOS:

   ```
   sudo yum install <pathtoagentinstaller>/<tabrmt-agent-setup-
   <version>-x86_64.rpm>
   ```

   ```
   sudo /opt/tableau/tabrmt/agent/install-scripts/upgrade-rmt-
   agent --accepteula
   ```

   For Ubuntu distributions:

   ```
   sudo apt install <pathtoagentinstaller>/<tabrmt-agent-setup-
   <version>-amd_64.deb>
   ```

   ```
   sudo /opt/tableau/tabrmt/agent/install-scripts/upgrade-rmt-
   agent --accepteula
   ```

9. Confirm the RMT Server and Agents are running. Start the RMT Server and Agents if they do not automatically restart after the upgrade is complete.

10. To verify RMT Agents have been upgraded, log in to Resource Monitoring Tool in a browser and go to the **Admin** menu, select **Environments**, and click the **Edit Environment** icon to see the environment details. In the **Servers** tab, you can see the version of the RMT Agent. This can be useful to determine which RMT Agents have been upgraded when you have a multi-node Tableau Server cluster.

Upgrade steps with enabling TLS for RabbitMQ

Since the agents in versions earlier than 2022.3 have been communicating using unencrypted connection to the RabbitMQ message queue service, when upgrading to version 2022.3 or

later, those agents need to be updated to use the new secure connection details. The steps to do this are as follows:

1. After completing the upgrade steps described in the above section, stop all agents by running the following command:

   ```
   rmtadmin stop --agent
   ```

2. Download the bootstrap file for the environment by running the following command:

   ```
   rmtadmin bootstrap-file --env<myenvironment> --filename <The absolute or relative path including the file name>
   ```

3. Run the following command on each of the machines where the Agent is installed:

   ```
   rmtadmin rotate-mq-certificate <BOOTSTRAP_FILE> --username=<RMT Server Username> --password-file=<RMT Server Password file name>
   ```

4. Restart each Agent machine after successfully running the `rmtadmin rotate-mq-certificate` command.

Who can do this

To upgrade Resource Monitoring Tool, you will need to have the following permissions:

- User account with full sudo access.
- Tableau Server Administrator.
- Resource Monitoring Tool Administrator.

Uninstalling Resource Monitoring Tool

There are two primary "uninstall" scenarios Resource Monitoring Tool supports:

- **Uninstall Resource Monitoring Tool:** Resource Monitoring Tool can be uninstalled using the `remove` command which removes each Tableau Resource Monitoring Tool service from the computer from which you are running the command. It also removes

data andResource Monitoring Tool user accounts and groups, but preserves configuration files, logs, and backup files, by moving them to a temporary directory under the `/opt/tableau/tabrmt/data` folder.

- **Obliterate Resource Monitoring Tool:** If you want to completely remove Resource Monitoring Tool from a computer, you can use a script provided by Tableau to remove Resource Monitoring Tool and all related files. *This removes all data as well as Resource Monitoring Tool components, so should only be done if you know you want to reset the computer to a pre-Tableau state.* You might need to do this if Technical Support recommends this step when troubleshooting an installation problem. Completely remove Resource Monitoring Tool without uninstalling any version first. The script will uninstall all existing versions found on the computer. If you have already uninstalled your existing version and now want to completely remove Tableau, you can find the script to do so in a temporary location.

Uninstall Resource Monitoring Tool using `remove`:

- For RHEL-like distributions including CentOS:

```
sudo yum remove tabrmt-master

sudo yum remove tabrmt-agent
```

- For Ubuntu distributions:

```
sudo apt remove tabrmt-master

sudo apt remove tabrmt-agent
```

Obliterate Resource Monitoring Tool using `tableau-rmt-obliterate` script:

If you run into issues when uninstalling using the `remove` command, you can use the obliterate script provided by Tableau to remove all of the installation files from your computer. By default, the obliterate script is located at `/opt/tableau/tabrmt/master/install-scripts/tableau-rmt-obliterate` on the RMT Server machine and `/op-`

`t/tableau/tabrmt/agent/install-scripts/tableau-rmt-obliterate` on Agent machines.

If you already attempted to uninstall Resource Monitoring Tool using the `remove` command, the obliterate script is automatically copied to: `/var/tmp/tableau-rmt-obliterate`

The information to run the script is described below:

1. As the root user, run the following command on the RMT Server machine to completely uninstall RMT Server:

   ```
   /var/tmp/tableau-rmt-obliterate -m -y -y -y
   ```

2. As the root user, run the following command on each Agent machine to completely uninstall Agents:

   ```
   /var/tmp/tableau-rmt-obliterate -a -y -y -y
   ```

You can force remove all the files, including the logs and backups, using the parameters:

`-y`

Required.

Removes Resource Monitoring Tool from this computer. Must be specified three times (`-y-y-y`) to confirm.

`-m`

Required to uninstall RMT Server.

Removes RMT Server.

`-a`

Required to uninstall Agent.

Removes Resource Monitoring Tool Agent if installed.

`-k`

Optional.

Does not copy backups to the `logs-temp` directory.

`-g`

Optional.

Does not copy logs to the logs to the `logs-temp` directory.

Who can do this

To uninstall Resource Monitoring Tool, you must have all the following:

- Administrator permissions on the machine you are installing Resource Monitoring Tool.
- Tableau Server Administrator site role.
- Resource Monitoring Tool Administrator account.

Configure Tableau Resource Monitoring Tool

This section includes topics that provide information on how to configure Tableau Resource Monitoring Tool to suit your requirements.

Resource Monitoring Tool Server Configuration

This topic describes the Resource Monitoring Tool Server (RMT Server) configuration options you can set using the web interface.

Post install setup configurations

You can update the configurations you specified during the setup.

Following are the two recommended ways to make configuration changes:

- **To do this using the web interface:** On the machine where RMT Server is installed, go to: *http://<hostname>/setup/server*.
- **To do this using command line**, use `rmtadmin master-setup` command with the `--skip-admin-creation` option to make sure you are not prompted to create the admin user post installation. For more information, see rmtadmin master-setup.

Note: Configuration values are saved in the configuration file. Changes can be made directly to this file, but it is advised to leverage the configuration options in the UI and through the `rmtadmin` command line utility. Changes to the config file will require a restart to be applied.

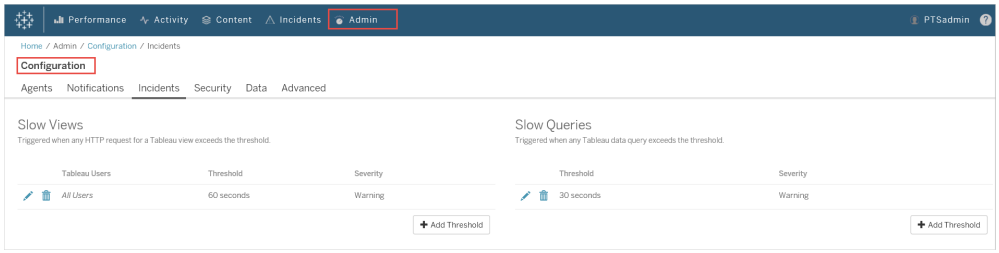
Here are some examples of the type of updates you might want to do after the initial setup.

- Changing SSL Configuration: The default configuration is set to false. You can change this configuration to require HTTPS. Use the rmtadmin master-setup command to update this setting.
- SSL Certificate Updates: You might have completed your initial RMT Server setup using the default certification mode for secure communications, and want to update the certification with your own certificate.
- Changes Ports: Sometimes you will need to use different port numbers than the default based on your network requirements. To learn more about ports and communications between the various components of Resource Monitoring Tool, see Tableau Resource Monitoring Tool Communication Ports.
- Update session expiration: If the user does not access the session for the set timeout period then the session will expire and they will need to log in again. By default, this is set to 240 minutes. You can also enable the **Sliding Expiration** option to reset the time out period when a session is accessed within the time out period.

> **Note:** Adding or removing a Tableau Server node: If you add a new node to the Tableau Server Cluster that you are monitoring using Resource Monitoring Tool, you will need to install and register this node. For more information, see Tableau Server Topology Changes.

To do this through command line, use `rmtadmin-master-setup` command. For more information, see rmtadmin Command Line Utility .

You can configure Resource Monitoring Tool by clicking on “Configuration” under the Admin menu. Configuration sections are divided by tabs. To do this using a configuration file, see RMT Server Configuration File.



Notifications

Notifications can be set at a global level and at the environment level.

You can choose how you want to receive your notifications- through email or Slack, or through both email and Slack.

The following applies to both email and Slack notfications:

Both email and Slack notifications can be set at a global level and at the environment level. To send any notifications, global configurations must be enabled, not just at the environment level. You can also configure different minimum severity levels at which to send notifications - they can be different for email and slack, they can also be different at global and environment levels.

Global configurations are applied to existing environments or any new environments created unless the environment is using custom thresholds.

Slack notification settings

Slack notifications were introduced in Resource Monitoring Tool version 2022.1.

Before you set up Slack notifications in Resource Monitoring Tool, there is some initial setup you need to do in Slack. The full details are described in this Slack article, but here are the main things you will need to do:

1. Create a new or use an existing Slack workspace or app and make sure it is enabled to receive incoming webhooks.
2. Authorize a channel where the notifications should be posted.
3. Copy the Webhook URL that is generated for your workspace.

To set up Slack notification in the Resource Monitoring Tool:

1. From the **Admin** menu, select **Global Configuration**.
2. Under the **Notifications** tab, in the **Slack** section, use the toggle button at the top of the section to enable Slack notifications.
3. Configure the following settings:
    1. **Minimum Severity**: The minimum severity level at which you want to receive Slack notifications.
    2. **Webhook URL**: URL of the incoming webhooks for your Slack workspace. This URL is automatically generated when you enable incoming webhooks for your Slack workspace.
4. Test it using the **Send a Test Message** button to make sure your settings are valid and they work.

To specify the notifications for each environment:

1. From the **Admin** menu, select the environment that you want to modify and choose **Edit Environment**.
2. In the **Notifications** tab, choose **Custom** for configuration type to override the global settings.
3. Set the minimum severity and the Webhook URL.
4. Test it using the **Send a Test Message** button to make sure your settings are valid and they work.

Email notifications

To set up email notification, start by configuring SMTP server settings at the global level. The SMTP server settings configured at the global level are applied automatically to all environments. You can customize certain settings like the minimum severity level and the sender and recipient emails for each environment.

Here are some technical details about how the TLS configuration works:

- RMT Server uses the STARTTLS SMTP option, which upgrades the SMTP connection to TLS after it has been initiated but before the mail content is sent. RMT does not support the REQUIRETLS SMTP option.
- RMT Server uses STARTTLS to encrypt mail in transit to the SMTP server. Whether or not the SMTP server stores the mail encrypted at rest is subject to the SMTP server configuration.

Before you can configure email notifications in the Resource Monitoring Tool, you must have an SMTP server set up and have the following pieces of information:

- **Name** of the SMTP server.
- If you are planning to use encrypted communications, make a note of the **TLS version** that the SMTP server accepts. Currently, **TLS 1.2 is required** by Resource Monitoring

Tool**and TLS 1.3 is supported**.

- The **certificate thumbprint** (optional).
- **Port number** for the SMTP server.
- **Username and password** (optional). Needed only if the SMTP server is configured to authenticate using a username and password.
- **Sender and recipient email addresses** that will be used to send and receive notifications.

To specify the SMTP server settings in theResource Monitoring Tool:

1. From the Admin menu, select **Global Configuration**.
2. In the **Notifications** tab in the **Email** section, select the toggle button at the top of the sections to enable email notifications.
    1. Configure the following SMTP server settings:
        1. **Server:** Provide the fully qualified DNS name of the SMTP server.
        2. **Encryption:** Specify whether you want the communications between RMT Server and the SMTP server encrypted. The option you choose will depend on the SMTP server configuration for encryption and your preference for using encrypted communications between RTM Server and the SMTP server.
            1. **Required:** Use this when your SMTP server is enabled to use encrypted communications and you want to make sure communications are always encrypted. When set to required, the connection will always be attempted using encryption. If the SMTP server is not enabled to use encryption, the RMT server will be unable to communicate with the SMTP server.

2. **Preferred:** Use this if you are not sure about the encryption settings of your SMTP server, but prefer that encrypted communications be used when possible. If the SMTP server is not enabled to use encryption, in this case, non-encrypted communication is used.

3. **Disabled:** Use this if your SMTP is not enabled to use encrypted communications. Communications between RMT Server and the SMTP server are not encrypted.

   > **Note:** If the SMTP server requires encryption, the connection will fail. If the SMTP server is enabled for encryption but does not require it, the connection will succeed.

4. **Options:** This determines how the SMTP server will be verified. You have the following options:

   1. Check Server identity: RMT will verify if the name of the certificate used matches the SMTP server name. If there is no match, the connection will fail.

   2. Trust all hosts: All certificate errors are ignored and overrides the Check Server identity. This should be used only if you are confident that the certificate that the RMT server using is from your server.

5. **TLS version:** The version of TLS that is supported by your SMTP server. There is a default version already selected, but you can select other versions. If multiple TLS versions are selected, the RMT server will use the most secure version compatible with RMT and the SMTP server. TLS version 1.3 requires Open SSL 1.1.1f. If you are planning to use TLS 1.3, make sure the machine where RMT Server is installed has Open SSL 1.1.1f.

6. **Certificate Thumbprint:** This is optional. The SHA1 certificate is used when provided and it must be valid and one that the SMTP server uses. A valid thumbprint will override other certificate errors like expiration dates and Server name mismatches.

7. **Port:** Port setting for the email server.

8. **Username:** Optional. The name of the account used to authenticate to the email server, if the SMTP server is configured with a username and password for authentication.

9. **Password:** Optional. The password for the account used to authenticate to the email server, if the SMTP server is configured with username and password for authentication.

10. **Minimum Severity:** The minimum severity level at which you want to receive email notifications. This will be applied to all environments unless modified at the environment level.

11. **Sender email:** The email address that is used to send the notifications. This will be applied to all environments unless modified at the environment level.

12. **Recipient email(s):** Email addresses of the people who should receive these notifications. This will be applied to all environments unless modified at the environment level.

13. **Test** it using the Send a Test Message button to ensure your settings are valid and can be used to successfully send an email notification.

To customize the settings for an environment:

1. From the **Admin** menu, select the environment that you want to modify and choose **Edit Environment**.

2. In the **Notifications** tab, choose **Custom** for configuration type to override the global settings.
3. Set the severity level, and sender and recipient email information.
4. Test it using the Send a Test Message button to make sure your settings are valid and you are able to send an email notification.

Troubleshoot connection failures

Connection failures can happen due to various reasons, but here are some that could be caused due to configuration issues:

- **Encryption setting errors:** Happens if there is a mismatch of encryption settings between the RMT Server and the SMTP server. For example, if the RMT server is set to require encryption, but the SMTP server is not configured to use encryption. The reverse is also true. If the SMTP server requires encrypted communication and RMT Server is set to disable encryption, the connection will fail.
- **Certificate errors:** When using encrypted communications, things like certificate authority and match between certificate name and the SMTP server are considered unless you choose to explicitly **Trust all hosts** under **Options**.
- **TLS errors:** TLS version support is dependent on the Operating System (OS) on which the RMT Server is installed. Check to see if the version of TLS you selected is supported by the OS. TLS 1.2 is required, but TSL 1.3 is also supported. TLS 1.3 requires Open SSL 1.1.1f or greater. Make sure you have Open SSL 1.1.1f on the machine where RMT Server is installed.

Incident thresholds

Configure the global incident notification settings. These global thresholds apply to all environments by default unless overridden by the individual environments.

See Incidents for information on what incident options are available and how to configure them.

Security

Configure the security settings for Resource Monitoring Tool user accounts.

Data

By default, Resource Monitoring Tool stores two weeks of detailed activity data and ten years of aggregated reporting data.

Data Retention: Detailed activity data from your Tableau Server powers the dashboards and incidents for diagnosis of recent performance incidents. Due to the associated storage and processing requirements, this data is only stored for two weeks by default.

Reporting Data: Aggregated activity data from your Tableau Server is stored for reporting purposes (e.g., the Chargeback report). This data is compact and can easily be stored for many years of historical reporting.

Advanced

Allows configuration of the Resource Monitoring Tool diagnostic logging levels.

See the Log Files for more information about logging.

Who can do this

Resource Monitoring ToolAdministrator or a Resource Monitoring Tool user with **Server-/Environment Management** role.

RMT Server Configuration File

This topic describes the configuration options you can use using the configuration file. To do this using the web interface, see Resource Monitoring Tool Server Configuration .

In an effort to align with our company values of Equality, we have changed non-inclusive terminology where possible. Because changing terms in certain places can causing a breaking change, we maintain the existing terminology. So, you may continue to see the terms in CLI commands and options, installation folders, configuration files. and other instances. For more information, see About Tableau Help.

The configuration file is located at `/var/opt/tableau/tabrmt/master/config.json`.

> **Note:** Configuration values are saved in the configuration file. Changes can be made directly to this file, but it is advised to leverage the configuration options in the UI and through the `rmtadmin` command line utility. Changes to the config file will require a restart to be applied.

You will need to restart the RMT Server after making changes to the configuration file.

Data Retention

By default the Resource Monitoring Tool stores two weeks of detail data and ten years of aggregated reporting data.

This is configurable. Here is an example snippet defining the data retention settings:

```
{
  "db":  {
    "cleanup":  {
      "afterDays":  14,
      "aggregate":  {
        "afterDays":  3650
      }
    }
  }
}
```

SMTP Configuration

An example `config.json` snippet defining the SMTP information:

```
{
  "smtp":  {
    "server":  "localhost",
    "port":  25,
    "username":  "my-username",
```

```
    "password":  "my-password",
    "requireSSL":  true
  }
}
```

The full list of configuration values available in the `smtp` configuration section:

Key	Data Type	Required?	Description
`server`	String	Optional	The SMTP server to use. Default value: `localhost`
`port`	Number	Optional	The SMTP server's port number. Default value: `25`
`username`	String	Optional	The username to use if authentication is required by the server.
`password`	String	Optional	The password to use if authentication is required by the server.
`requireSSL`	Boolean	Optional	`true` if your SMTP server requires SSL, otherwise `false`. Default value: `false`

Notification Configuration

An example `config.json` snippet defining the notification information:

```
{
  "notifications":  {
    "email":  {
      "from":  "sender@domain.com",
      "to":  "recipient1@domain.com,recipient2@domain.com"
    }
```

```
    }
}
```

Global notification information (as shown above) can be overridden per environment in the `environments` section:

```
{
  "environments":  {
    "Environment1":  {
      "notifications":  {
        "email":  {
          "from":  "sender@domain.com",
          "to":  "recipient1@domain.com,recipient2@domain.com"
        }
      }
    }
  }
}
```

The full list of configuration values available in the `notifications.email` configuration section:

Key	Data Type	Required?	Description
`from`	String	Required	The sender's email address.
`to`	String	Required	The recipient email address (es). Multiple addresses must be separated by commas or semi-colons.
`minimumIncidentSeverity`	String	Optional	The minimum incident severity at which emails

Key	Data Type	Required?	Description
			will be sent. Default value: `critical`. Also see Tableau Resource Monitoring Tool - Incidents.

Histogram Configuration

The boundaries used to generate histograms in the web interface are configurable, using an array of values that represent each boundary.

The full list of configurable histograms in the `monitoring.histograms` section:

Key	Data Type	Required?	Description
`viewLoadDuration`	Array of Numbers	Optional	The histogram boundaries for view loads. Values are in milliseconds. Default value: `[1000, 3000, 6000, 10000]`
`externalDataRequestDuration`	Array of Numbers	Optional	The histogram boundaries for external data requests. Values are in milliseconds. Default value: `[1000,`

Key	Data Type	Required?	Description
			`3000, 6000, 10000]`
`backgroundTaskDuration`	Array of Numbers	Optional	The histogram boundaries for background tasks. Values are in milliseconds. Default value: `[60000, 300000, 600000, 1800000]`

As an example, to use the following histogram buckets for everything at a global level:

- ≤ 1 second
- > 1 second and ≤ 10 seconds
- > 10 seconds and ≤ 30 seconds
- > 30 seconds

The configuration would look like:

```
{
  "monitoring":  {
    "histograms":  {
      "viewLoadDuration":  [1000,  10000,  30000],
      "externalDataRequestDuration":  [1000,  10000,  30000],
      "backgroundTaskDuration":    [1000,  10000,  30000]
    }
  }
}
```

Histogram boundaries can also be set per environment. As an example, for an environment whose identifier is “staging-environment” to use the following view histogram buckets:

- ≤ 2.5 seconds
- > 2.5 seconds and ≤ 5 seconds
- > 5 seconds and ≤ 30 seconds
- > 30 seconds and ≤ 1 minute
- > 1 minute and ≤ 10 minutes
- > 10 minutes

The configuration would look like:

```
{
  "environments":  {
    "staging-environment":  {
      "monitoring":  {
        "histograms":  {
          "viewLoadDuration":  [2500,  5000,  30000,  60000,
600000]
        }
      }
    }
  }
}
```

The “staging-environment” would fall back to the global histogram configuration for background tasks.

Minimum TLS Version

By default, RMT will use a secure version of TLS to encrypt traffic. The default minimum version is 1.2, but if you have specific security requirements that mandate that older versions of TLS be disabled, you can modify the `server.minimumTlsVersion` section of the configuration file to enforce a minimum TLS version, as shown in the example snippet below where the minimum version is set to 1.3. The list of valid values for `minimumTlsVersion` are defined in SslProtocols from .Net Core.

```
"server":  {
  "url":  "https://rmtserver:443",
  "https":  {
    "enforce":  true,
```

```
    "certificate":  {
      "mode":  Default,
      "local":  {
        "name":  null,
        "password":  null
       },
      "store":  {
        "certificateThumbprint":  ""}
     },
     "minimumTlsVersion":  "Tls13"
   },
```

Incident Configuration

See Incidents for information on what incident options are available and how to configure them.

RMT ServerLogging

See Tableau Resource Monitoring Tool Log Files.

Agent

The agent service's configuration file is located at `/var/-opt/tableau/tabrmt/master/config.json`

You will need to restart the agent service after making changes to the config file.

Tableau Server Detection

In almost all situations agents will automatically detect the Tableau Server installation and no configuration is needed beyond the standard agent setup process. If desired, you can however manually configure the Tableau Server information through the `config.json` file.

An example `config.json` snippet defining the Tableau Server information needed to run the agent:

```
{
  "agent":  {
    "tableauServer":  {
      "override":  true,
      "productVersion":  2021.4,
      "applicationDirectory":  "/var/opt/tableau/tableau_
server/2021.4"
    }
  }
}
```

The full list of configuration values available in the `agent.tableauServer` configuration section:

Key	Data Type	Required?	Description
`applicationDirectory`	String	Required	The Tableau Server application root directory.
`dataDirectory`	String	Optional	The directory to get Tableau Server data files.
`override`	Boolean	Optional	`true` to have the configuration values take precedent over any automatically detected values. `false` to have the configuration values act as a fallback to the automatically detected values.
`productVersion`	Number	Required	The version num-

Key	Data Type	Required?	Description
			ber of the Tableau Server.

Agent Logging

See Tableau Resource Monitoring Tool Log Files.

Common

Common configuration values are available in both RMT Server and Agent applications. See application-specific sections for guidance on locating the configuration file.

You will need to restart the application service after making changes to the config file.

Encrypted Messaging

To enable encrypted messaging, the RabbitMQ server must be first configured to allow TLS. See the Encrypted Data Collection administrator guide for more details.

When configuring the RMT Server or Agent(s) for encrypted messaging:

- Both the `enabled` flag and the `certificateHostName` must be configured for encryption to be enabled.
- The `certificateHostName` variable MUST match the canonical name (CN=) on the server certificate or the connection will fail.
- The `port` setting in the `mq` section will likely need to be changed based on the TLS port you configured RabbitMQ.

```
{
  "mq":  {
    "port":  5671,
    "tls":  {
      "enabled":  true,
      "certificateHostName":  "foo"
    }
  }
}
```

The full list of configuration values available in the `mq.tls` configuration section:

Key	Data Type	Required?	Description
`enabled`	Boolean	Optional	`true` enables TLS encryption for messaging connections. `false` uses unencrypted connections for messaging. Default value: `false`
`certificateHostName`	String	Optional	`certificateHostName` MUST match the canonical name (CN=) of the server certificate or the connection will fail. Default value: `""`

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-/Environment Management** role.

rmtadmin Command Line Utility

Resource Monitoring Tool includes a command line interface, `rmtadmin.cmd`, located in the installation folder. The default installation folder is:

- RMT Server `/var/opt/tableau/tabrmt/master/`.
- Agent: `/var/opt/tableau/tabrmt/agent/`.

  `rmtadmin` is included in both the **RMT Server** and **Agent** installations. Some commands may vary based on whether you are using the RMT Server or Agent `rmtadmin` command line utility. E.g., the `users` command only works from the *RMT Server*. The `ziplogs` command is available everywhere but only includes the log files from the application the command is run on.

Note: You must run these commands as the `tabrmt-master` user:

```
sudo su --login tabrmt-master
```

Here are the commands that can be used with the `rmtadmin` command line:

Note: The `rmtadmin` commands use both positional parameters and options.
The positional parameters should be specified using only the values. You don't need to specify the actual keyword. The option keyword and the value should be specified using an equal sign.
Example:

```
rmtadmin <command> <positional parameter value> --<option keyword>=<value>
```

- rmtadmin agents
- rmtadmin bootstrap-file
- rmtadmin cleanup
- rmtadmin create-admin-user
- rmtadmin create-env
- rmtadmin delete-env
- rmtadmin data-access
- rmtadmin delete-env-data
- rmtadmin delete-server
- rmtadmin delete-server-data
- rmtadmin deregister-agent
- rmtadmin environments
- rmtadmin get
- rmtadmin help
- rmtadmin master-setup
- rmtadmin passwd
- rmtadmin query
- rmtadmin register
- rmtadmin restart
- rmtadmin rotate-mq-certificate

- rmtadmin rotate-mq-certificates
- rmtadmin-servers
- rmtadmin-service-setup
- rmtadmin set
- rmtadmin start
- rmtadmin stop
- rmtadmin status
- rmtadmin test-env
- rmtadmin update-baseline
- rmtadmin update-env
- rmtadmin users
- rmtadmin version
- rmtadmin ziplogs

In an effort to align with our company values of Equality, we have changed non-inclusive terminology where possible. Because changing terms in certain places can cause a breaking change, we maintain the existing terminology. So, you may continue to see the terms in CLI commands and options, installation folders, configuration files, and other instances. For more information, see About Tableau Help.

rmtadmin agents

Note: Added in version 2021.2

Lists all the registered Agents on Tableau Server nodes for all environments.

This is useful to see where the Resource Monitoring Tool Agent is installed on Tableau Server.

This command only works on the RMT Server.

Synopsis

```
rmtadmin agents [options][global option]
```

Options

`--env`

Optional: Use this option to see the list of Agents for the specific environment. This is the system generated identifier. You can find this by going to **Admin** > **Environments > Edit Environment**. On this page, in the **Environment Details** section, you will see the identifier for the environment.

`--keys`

Optional: includes the agent key in the output.

Example: `rmtadmin agents --env=<myenvironmentidentifier> --keys`

rmtadmin bootstrap-file

Note: Added in version 2021.2

Creates and saves the bootstrap fie used to register Agents on Tableau Server nodes.

Bootstrap file will be created and saved to the specified absolute path or relative path. Relative paths are resolved to the current working directory.

This command only works on the RMT Server.

Synopsis

```
rmtadmin bootstrap-file [options][global option]
```

Options

`--env`

Required. This is the system generated identifier. You can find this by going to **Admin > Environments > Edit Environment**. On this page, in the **Environment Details** section, you will see the identifier for the environment.
The Agent on a Tableau Server node can only registered to one environment at a time.

`--filename`

Optional. The absolute or relative path including the file name. The relative paths are resolved to the current working directory. If not specified, the default output filename is used.

`--force`

Optional. Overwrites the existing file with the same name.

Example: `rmtadmin bootstrap-file --env=<myenvironmentidentifier>`

rmtadmin cleanup

Note: Added in version 2021.2

Deletes the log files that are older than a certain number of days. By default it deletes log files older than 31 days.

This command on both RMT Server and Agent.

Note: If the Resource Monitoring Tool processes are running, the active log files will not be cleaned up by the command. If you want to be sure that all files are deleted, you will need to stop RMT using the rmtadmin_stop command and then run cleanup using the fol-

lowing command, `rmtadmin cleanup --log-files-retention=0`, and then restart the processes using the rmtadmin_start command.

Synopsis

```
rmtadmin cleanup [option][global option]
```

Option

`--log-files-retention`

Optional. Deletes the log files older than the specified number of days.

Example:`rmtadmin cleanup --log-files-retention=<number of days>`

rmtadmin create-admin-user

Version: Added in version 2022.1

Creates an initial admin user if no other users exist.

Note: Beginning in version 2023.1 you can create multiple admin users.

This command on only RMT Server.

Synopsis

```
rmtadmin create-admin-user [options][global option]
```

Options

`--username`

Optional. The username for the admin user account. Defaults to "admin" if not specified.

`--password`

Required. The password for the admin user account.

`--password-file`

Optional. The path to the file containing the password for the admin user account. You can use this if you do not want to type in the password directly and have the password stored in a file that can be accessed.

Example:`rmtadmin create-admin-user --username<admin user name> --password <password for the admin user account>`

rmtadmin create-env

Note: Added in version 2021.2

Creates a new environment.

This command only works on the RMT Server.

Synopsis

```
rmtadmin create-env [options][global option]
```

Options

Option	Required?	Default	Description
--name	Yes	n/a	The name of the environment.
--gateway-url	Yes	n/a	URL used to access the Tableau Server gateway.
--version	Yes	n/a	Tableau Server version that this environment will be monitoring.

Option	Required?	Default	Description
--non-inter-active	No	Interactive prompts are enabled by default.	Disables all interactive prompts.
--no-test	No	API and Repository connection testing is enabled by default.	Disables testing API and repository connections.
--api-username	No	Null	User name of the account used to connect to Tableau Server APIs. The user account should be a Tableau Server administrator with access to all Tableau Server sites.
--api-password	No (If you specify the Tableau API user name, you will either provide the password, or specify the file path and file that has the password)	Null	Password of the Tableau Server API user account used to connect to Tableau Server APIs.
--api-password-file	No	Null	The path to the file and the name of the file containing the password of the Tableau Server API user account.

Option	Required?	Default	Description
--repository-server	Yes	Null	This is the server name for the PostgreSQL database that in installed with Tableau Server.
--repository-database	Yes	Null	This is the name of the PostgreSQL database that is installed with Tableau Server.
--repository-port	Yes	Null	The port number of the Tableau Server Repository database.
--repository-username	Yes	Null	Username used to connect to PostgreSQL database installed with the Tableau Server Repository. Resource Monitoring Tool accesses the Tableau Server Repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the **readonly** database user. For details, see Enable access to the Tableau Server repository.
--repository-password	Yes	Null	Password for the user account used to connect to the PostgreSQL database that is installed with the Tableau Server . Resource Monitoring Tool accesses the Tableau Server Repository database directly for performance reas-

Option	Required?	Default	Description
			ons. For this to work, access to the repository must be enabled, with a password set for the **readonly** database user. For details, see Enable access to the Tableau Server repository.
--repository-password-file	No, but required if you are not providing the password in command prompt or in a script directly.	Null	The path including the file name where the password for the user account used to connect to the PostgreSQL database that is installed with Tableau Server.
--repository-ssl-mode	No	Prefer	Tableau Server Repository SSL Mode: **Prefer** SSL or **Require** SSL to configure SSL connections to Tableau Repository. **Disable** to never use SSL to make Tableau Server Repository connections.
--repository-ssl-thumbprint	No	Null	When configuring Tableau Server PostgreSQL to allow direct connections, Tableau Server creates a certificate and keys. You can choose to either supply the thumbprint for the certificate that was generated by Tableau Server, or

Option	Required?	Default	Description
			copy the **server.crt** file to the Resource Monitoring Tool Server machine. If you choose to copy the certificate file, you don't have to supply the thumbprint. For more information, see Configure Postgres SSL to Allow Direct Connections from Clients.

Example:`rmtadmin create-env --name=<myenvironment> --api-user-name=<TableauServer API user name> --api-password=<password for the Tableau Server API user account> --gateway-url <Tableau Server Gateway URL> --version <Tableau Server version>`

rmtadmin data-access

Note: Added in version 2022.3

Enables or disables access to PostgreSQL database. Note: Access to PostgreSQL database is required starting in version 2022.3 for Resource Monitoring Toolto successfully gather all monitoring data from Tableau Server.

This command only works on the RMT Server

Note: The PostgreSQL database must be restarted for this configuration to take effect.

Synopsis

```
rmtadmin data-access [positional parameter][options] [global option]
```

Positional Parameter

`mode`

Required. The mode that should be used for remote data access to PostgreSQL database. The values should be one of: *None*, *ReadOnly*, *Admin*. When set to *None*, the data access is disabled. ReadOnly, and Admin specify which user account to use to access the database..

Options

`--`

Example:

```
rmtadmin data-access ReadOnly
```

```
rmtadmin restart --db
```

rmtadmin delete-env

Note: Added in version 2021.2

Deletes a specific environment and all data that has been collected for that environment. It also removes the connection and topology information about the Tableau Server that the environment is monitoring and deregisters all the Resource Monitoring Tool Agents.

This command only works on the RMT Server.

Synopsis

```
rmtadmin delete-env [positional parameter][options] [global option]
```

Positional Parameter

`env`

Required. This is the system generated identifier. This is the system generated identifier. You can find this by going to **Admin** > **Environments > Edit Environment**. On this page, in the **Environment Details** section, you will see the identifier for the environment.

Options

`--confirm`

Optional. Confirm that you want to delete the environment. If you do not provide this option, you will be prompted to confirm.

`--skip-agent-disconnect`

Optional. Skips disconnecting the Agent and continues with deregistering the Agent. Use this option if you think the Agent is inaccessible - For example, if the Tableau Server node has been removed or if the Agent has been uninstalled on that node.

Example:`rmtadmin delete-env <myenvironmentidentifier>`

rmtadmin delete-env-data

Permanently deletes all Tableau Server related data collected for a specific environment. Environment configuration, Tableau Server information, and Agent registration will not be removed.

This is useful for clearing all the existing data in an environment without removing the environment itself. After deleting the existing data, new data sent by the Agents will continue to be processed.

This command only works on the RMT Server.

Synopsis

```
rmtadmin delete-env-data [positional parameter][option][global
option]
```

Positional Parameter

`env`

The identifier of the environment for which the data should be deleted. This is the system generated identifier. You can find this by going to **Admin** > **Environments > Edit Environment**. On this page, in the **Environment Details** section, you will see the identifier for the environment.

Option

`--confirm`

Optional. Confirm that you want to delete the environment data. If you do not provide this option, you will be prompted to confirm.

Example: `rmtadmin delete-env-data <myenvironmentidentifier>`

rmtadmin delete-server

Note: Added in version 2021.2

Deletes the configuration information of the Tableau Server node from the environment, deregisters the Agent on that node, and deletes all monitoring data collected specific to that node.

This command only works on the RMT Server.

Synopsis

```
rmtadmin delete-server [options][global option]
```

Options

`--env`

Required. The identifier of the environment that the Tableau Server node is connected to. This is the system generated identifier. You can find this by going to **Admin** > **Environments > Edit Environment**. On this page, in the **Environment Details** section, you will see the identifier for the environment.

`--hostname`

Required to delete information about on or more connected to the environment and if **--all** is not specified.

This is the machine name of the Tableau Server node. Use commas to separate multiple values if specifying more than one host name.

`--all`

Optional, but required if **--hostname** is not specified.

Deletes the information for all the Tableau Server nodes connected to the environment.

`--skip-agent-disconnect`

Optional. Skips disconnecting the Agent and continues with deregistering the Agent. Use this option if the Agent is inaccessible- For example, if the node has been removed from Tableau Server or if the Agent has been uninstalled on that node.

`--confirm`

Confirm that you want to delete all the information about the Tableau Server node. If you do not provide this option, you will be prompted to confirm.

Example:`rmtadmin delete-server --hostname=<machine name of the Tableau Server node>`

rmtadmin delete-server-data

Note: Added in version 2021.2

Deletes the data collected from one or more Tableau Server nodes. Data removed only pertains to the nodes that are specified. If you specify *all*, then all the data collected from all the nodes will be deleted. Environment configuration, Tableau Server information, and Agent registrations will not be deleted.

This command only works on the RMT Server.

Synopsis

```
rmtadmin delete-server-data [options][global option]
```

Options

`--env`

Required. The identifier of the environment that the Tableau Server node is connected to. This is the system generated identifier. You can find this by going to **Admin** > **Environments > Edit Environment**. On this page, in the **Environment Details** section, you will see the identifier for the environment.

`--hostname`

Required to delete information about on or more connected to the environment and if **--all** is not specified.

This is the machine name of the Tableau Server node. Use commas to separate multiple values if specifying more than one host name.

`--all`

Optional, but required if **--hostname** is not specified.

Deletes the monitoring data for all the Tableau Server nodes connected to the environment.

`--confirm`

Confirm that you want to delete all the monitoring data from the Tableau Server nodes. If you do not provide this option, you will be prompted to confirm.

Example: `rmtadmin delete-server-data --all`

rmtadmin deregister

Note: Added in version 2021.2

Deregisters the Agent from the environment. Monitoring data from this node will no longer be collected. Existing data will remain and not be deleted.

This command only works when run on the RMT Agent. Run this on the Agent you want to deregister.

Synopsis

```
rmtadmin deregister [options]
```

Options

`--confirm`

Optional. Confirm that you want to deregister the Agent. This bypasses the confirmation prompt.

`--ignore-master-errors`

Optional. Ignores any errors that occur while communicating with the RTM Server during this process.

Example: `rmtadmin deregister --confirm`

rmtadmin deregister-agent

Note: Added in version 2021.2

Deregisters the specified Agent from the environment. Monitoring data from this node will no longer be collected. Existing data will remain and not be deleted.

This command only works when run on the RMT Server.

Synopsis

```
rmtadmin deregister-agent [options][global option]
```

Options

`--env`

Required if **--key** is not specified.

This is identifier of the environment where the Agent is currently registered. This is the system generated identifier. This is the system generated identifier. You can find this by going to **Admin** > **Environments > Edit Environment**. On this page, in the **Environment Details** section, you will see the identifier for the environment.

`--key`

Optional if **--env** is specified.

This is the Agent key. Use commas to separate multiple values. You can get the key values by running the rmtadmin agents command.

`--hostname`

Optional. Machine name of the Tableau Server node where the Agent is installed. Use commas to separate multiple values.

`--all`

Optional. Use this option to deregister Agents on all the nodes.

`--ignore-agent-errors`

Optional. Ignores any errors that occur if unable to connect with the Agents during this process.

`--skip-agent-disconnect`

Optional. Skips disconnecting the Agent and continues with jderegistering the Agent. Use this option if you think the Agent is inaccessible - For example, if the node has been removed from Tableau Server or if the Agent has been uninstalled on that node.

`--confirm`

Optional. Confirm that you want to deregister the Agents. If you do not provide this option, you will be prompted to confirm.

Example: `rmtadmin deregister-agent --env=<myenvironmentidentifier> --all`

rmtadmin environments

Note: Added in version 2021.2

Lists all the environments on the Resource Monitoring Tool Server.

This command only works on the RMT Server.

Synopsis

```
rmtadmin environments [global option]
```

rmtadmin get

Gets the value of a specified configuration key.

This command can be used both on the RMT Server and Agent.

Synopsis

```
rmtadmin get <config.key> [global option]
```

Positional Parameter

`key`

Name of the configuration key.

Example: `rmtadmin get db:database`

Supported configuration keys

Key	Data Type	Applies To	DEFAULT	Description
`db:database`	String	RMT Server	tabrmtdb	Name of the PostgreSQL database that RMT uses.
`db:readOnlyUsername`	String	RMT Server	readonly	Username of a PostgreSQL user account which has only

Key	Data Type	Applies To	DEFAULT	Description
				read-only access to the RMT data.
db:readOnlyPassword	String	RMT Server	Generated by the installer program.	Password of the PostgreSQL read-only user.
mq:tls:certificateHostName	String	RMT Server and Agent	n/a	Host name in the certificate to use when connecting to RabbitMQ via TLS.
mq:tls:enabled	Boolean	RMT Server and Agent	FALSE	Enable/disable TLS connection to RabbitMQ.
mq:port	Integer	RMT Server and Agent	5672	RabbitMQ host port number.
mq:virtualHost	String	RMT Server and Agent	tabrmt	Connection to the RMT Server or Agent. For Agent, you can modify this setting by registering the Agent with a bootstrap file. For the RMT Server,

Key	Data Type	Applies To	DEFAULT	Description
				the value is managed by the installer.
`server.web.run`	Boolean	Agent	TRUE	Enable or disable the Agent's web interface.

Use the set command to change the current configuration value.

rmtadmin help

Shows general help about the command line interface and the available commands.

This command can be used both on the RMT Server and Agent.

Synopsis

```
rmtadmin help
```

Show help and usage information for a specific command:

This command can be used both on the RMT Server and Agent.

```
 rmtadmin help [command]
```

rmtadmin master-setup

Configures the RMT Server with specified options. Used both during install and post installation configurations.

Synopsis

```
rmtadmin master-setup [options][global option]
```

The configuration options are listed below:

Option	Required?	Default	Description
--admin-password	Yes (Required for initial installation only) Password can be supplied in the command line or provide a file with the password to use. If neither is provided, you will be prompted for the password.	n/a	The password for the administrator user.
--admin-password-file	Yes (Required for initial installation only) Password can be supplied in the command line or provide a file with the password to use. If neither is provided, you will be prompted for the password.	n/a	The password for the administrator user.
--admin-username	No	admin	The username for the administrator user.
--skip-admin-creation **Added:** version	No This option must be and only used when	False	Skip creating the administrator user and password.

Option	Required?	Default	Description
2020.4.0 **Retired:** version 2022.3.0	you making configuration updates post installation. In versions where this option is valid (2020.4.0 - 2022.2.x), if not used during post installation configuration updates, the command will fail and no updates will be made. In versions 2022.3.0 and later, the command will create the admin where necessary.		
--http-port	No	80	
--require-https	No	False	Redirect http traffic to HTTPS.
--https-certificate-mode	No	'Default' Available options: • Default • Store (Windows only) • Local	The type of certificate search to perform for the HTTPS certificate. **Default:** This mode uses the default self-signed certificate sup-

Option	Required?	Default	Description
			plied by the installer. **Store:** This allows you to enter the thumbprint of a certificate in the Windows certificate store. **Local:** Allows you to specify a file-based certificate in the **config** folder.
--https-certificate-store-thumbprint	No	Null	The HTTPS certificate hash/thumbprint to search for in 'store' certificate mode.
--https-certificate-local-name	No	Null Note: If not specified, the Resource Monitoring Tool is installed with a self-signed certificate and will use that certificate for HTTPS communications.	The name of the HTTPS certificate file.
--https-certificate-local-password	No	Null	The password to use for the HTTPS certificate.

Option	Required?	Default	Description
--https-certificate-local-password-file	No	Null	The path to the file containing the password to use for the HTTPS certificate.
--confirm	No	Prompt for confirmation.	Confirms the restart of the RMT Server.
--host	No **Note**: We don't recommended changing this value.	Current configuration value or machine name.	The preferred URL for accessing the Resource Monitoring Tool server. The server will listen to all the IPs regardless of the host name.
--db-config=external	No	None. This is only used when the repository is configured external to RMT Server.	Use this to configure RMT Server to use an external repository. You do not need to specify this if you want the PostgreSQL database to be installed locally.
--db-server	No	Current configuration value or installer default.	This is the server name for the PostgreSQL database that in installed with the Resource Monitoring Tool. If you are using an external repository hosted on AWS RDS, this

Option	Required?	Default	Description
			should be the RDS instance name.
--db-ssl-mode Version: Added in version 2023.1.0	No	`Prefer`	This determines the SSL/TLS encryption for the connection to the PostgreSQL database that in installed with the Resource Monitoring Tool. Options are `Prefer` (the default), `VerifyCA`, and `VerifyFull`. For details, see the Npgsql documentation.
--db-port	No	Current configuration value or installer default.	Port number for the database server.
--db-username	No	Current configuration value or installer default.	Username used to connect to PostgreSQL database installed with the Resource Monitoring Tool. If you are using an external repository hosted on AWS RDS, this should be "postgres".

Option	Required?	Default	Description
--db-password	No	Current configuration value or installer default.	Password for the user account used to connect to the PostgreSQL database that is installed with the Resource Monitoring Tool. If you are using an external repository hosted on AWS RDS, this is the user password you created when creating the RDS instance.
--mq-config=external	No	None This option in only specified when the messaging service is hosted external to RMT Server.	This options configures RMT Server to use an external Rabbit MQ messaging service.
--mq-server	No	Current configuration value or installer default.	The name of the message queue server.
--mq-port	No	Current configuration value or installer default.	The message queue port.
--mq-vhost	No	Current configuration value or installer default.	The message queue virtual host.
--mq-username	No	Current con-	The user name used to

Option	Required?	Default	Description
		figuration value or installer default.	connect to the message queue.
--mq-password	No	Current configuration value or installer default.	The password for the user account used connect to the message queue.
--mq-enable-tls	No	Current configuration value or **false**.	Requires TLS connection to connect to the message queue.
--mq-tls-certificate-host	No	Current configuration value or **null**.	The canonical name of the message queue server. This must match the name on the certificate.
--password-salt	No	Random	A global salt to use to hash the password. This applies to the local user accounts created in the Resource Monitoring Tool.
--password-min-length	No	Current configuration value or **10**.	The minimum length for the password. This applies to the local user accounts created in the Resource Monitoring Tool.
--password-min-numeric	No	Current con-	The minimum required

Option	Required?	Default	Description
		figuration value or **1**.	numeric characters in the password. This applies to the local user accounts created in the Resource Monitoring Tool.
--password-min-special	No	Current configuration value or **1**.	The minimum required special characters in the password. This applies to the local user accounts created in the Resource Monitoring Tool.
--password-min-latin	No	Current configuration value or **5**.	The minimum required latin characters in the password. This applies to the local user accounts created in the Resource Monitoring Tool.
--password-require-mixed-case	No	Current configuration value or **true**.	Requires mixed case characters in passwords. This applies to the local user accounts created in the Resource Mon-

Option	Required?	Default	Description
			itoring Tool.
--auth-timeout-minutes	No	Current configuration value or **240**.	The number of minutes before the user authentication expires for the session.
--auth-sliding-expiration	No	Current configuration or **true**.	Whether or not to reset the authentication timeout period with user activity.

Examples:

To specify the admin password: `rmtadmin master-setup --admin-password=<password> --skip-admin-creation`

To update the port post install: `rmtadmin master-setup --http-port=8000 --skip-admin-creation`

rmtadmin passwd

Resets the password for a specific Resource Monitoring Tool user account.

This command works only on the RMT Server.

Synopsis

```
rmtadmin passwd [positional parameter][global option]
```

Positional Parameter

`username`

Name of the user that you want to change the password for.

Example: `rmtadmin passwd <username>`

rmtadmin query

Executes a raw SQL query against the Resource Monitoring Tool database and saves the results to an output file.

This command works only on the RMT Server.

Synopsis

```
rmtadmin query [positional parameter][options] [global option]
```

Positional Parameter

`sql`

SQL command text to run. You can provide multiple SQL commands. To use a file with the SQL commands, use **@** as a prefix to the name of the file.

Options

`--outfile=VALUE`

Name of the zip file you want to output the query results. Default is *queryresults.zip*

`--force`

Overwrites the existing file.

`--timeout=VALUE`

timeout for the query. Specify this in seconds

`--commit`

Commits any changes made by the SQL command to the database. By default, the SQL command is run as a transaction, but is rolled back at completion.

Example: `rmtadmin query <SQLCommand> --outfile=<path and the output file name>`

rmtadmin register

Note: Added in version 2020.2

Registers the Agent using a bootstrap file. The bootstrap file can be downloaded using the rmtadmin bootstrap-file command.

This command is can only be used on the Agent and should be run on the machine where you want to install the Agent.

Synopsis

```
rmtadmin register [options][positional parameter] [global option]
```

Positional Parameter

`bootstrap`

The file path including the name of the bootstrap file.

Options

`--username`

Required. Name of the admin user created during the RMT Server installation.

`--password`

Required. Password for the user account.

`--password-file`

Path including the file name that contains the password information. Password can be supplied in the command line or a file that contains the password. If neither is provided, you will be prompted for the password.

`--server-name`

Name of the node where Agent is being installed. It defaults to the machine name if no name is specified.

`--server-description`

Custom description for the Server.

Example: `rmtadmin register <bootstrap file name and path> --server-name=<server name>`

rmtadmin restart

Restarts the Resource Monitoring Tool applications. When run from the machine where RMT Server is installed, it restarts the RMT Server application. When run on a node where the Agent is installed, it restarts just the Agent application on that particular node.

This command can be run from both the RMT Server and Agent.

Synopsis

```
rmtadmin restart [options] [global option]
```

Options

At least one option must be specified:

RMT Server:

`--all`

Restarts all services.

`--master`

Restarts the RMT Server.

`--mq`

Restarts the message queue service.

`--db`

Restarts the database service.

Agent:

`--agent`

Restarts the Agent service running on the machine.

Example: `rmtadmin restart --db`

rmtadmin rotate-mq-certificate

Rotates the Agent message queue client certificate on the agent machine. Run this command after running `rmtadmin rotate-mq-certificate` on the Server machine.

This command must be run from the RMT Agent.

Synopsis

```
rmtadmin rotate-mq-certificate [options] [positional parameter]
```

Positional Parameter

`bootstrap`

The file path including the name of the bootstrap file.

Options

`--username`

Required. The username for the master server.

`--password`

Required. Password for the master server user account.

`--password-file`

Path including the file name that contains the master server password information. Password can be supplied in the command line or a file that contains the password. If neither is provided, you will be prompted for the password.

`--confirm-restart`

Confirms service restart and bypasses confirmation prompt.

rmtadmin rotate-mq-certificates

Rotates message queue server and client certificates on the Server machine.

This command must be run from the RMT Server.

Synopsis

```
rmtadmin rotate-mq-certificates [options]
```

Options

`--confirm-restart`

Confirms service restart and bypasses confirmation prompt.

rmtadmin servers

Note: Added in version 2021.2

Lists all the Tableau Server nodes across all environments or a specific environment.

This command only works on the RMT Server.

Synopsis

```
rmtadmin servers [positional parameter][global option]
```

Positional Parameter

`env`

Optional. Specify the identifier of the environment to get a list of Tableau Server nodes registered in that environment. This is the system generated identifier. You can find this by going to **Admin** > **Environments > Edit Environment**. On this page, in the **Environment Details** section, you will see the identifier for the environment.

rmtadmin service-setup

> **Note:** Added in version 2021.4. Used to install, or update the user credential for Resource Monitoring Tool services including RMT Server, Agent, Rabbit MQ, and PostgreSQL database. This is useful to update service information post-install.

This command works on both RMT Server and Agent.

Synopsis

```
rmtadmin service-setup [positional parameter][options][global option]
```

Positional Parameter

At least one of the following values must be specified:

`all`

Updates all available services. Can be run on both RMT and Agent. When run on a Tableau Server node, updates the Agent on that node.

`master`

Updates the RMT Server.

`agent`

Updates the Agent. Can only be run on the Agent (Tableau Server node).

`db`

Updates the database service. Can only be run on RMT Server.

`mq`

Updates the message queue service. Can only be run on RMT Server.

Options

`--confirm`

Optional. Confirm that you want to make updates. Can be run on both RMT and Agent. When run on a Tableau Server node, updates the Agent on that node.

`--append-permissions`

Optional. Appends new permissions. Use this if you do not want to overwrite the existing permissions.

rmtadmin set

Sets the value of a specified configuration key.

This command works both on the RMT Server and Agent.

Synopsis

```
rmtadmin set [positional parameter] [global option]
```

Positional Parameters

`key`

The configuration key you want to change the value for.

`value`

The new value you want to use.

Example: `rmtadmin set mq:port <port number>` where `mq:port` is the key and the `<port number>` is the value.

Supported configuration keys

Key	Data Type	Applies To	DEFAULT	Description
`mq:tls:certificateHostName`	String	RMT Server and Agent	n/a	Host name in the certificate to use when connecting to RabbitMQ via TLS
`mq:tls:enabled`	Boolean	RMT Server and Agent	FALSE	Enable/disable TLS connection to RabbitMQ
`mq:port`	Integer	RMT Server and Agent	5672	RabbitMQ host port number.
`server.web.run`	Boolean	Agent	TRUE	Enable or disable the Agent's web interface.

Options

`--backup`

Creates a backup of the configuration file before making changes.

Use the get command to view the current configuration value.

rmtadmin start

Starts the Resource Monitoring Tool services. When this is run on the RMT Server, it will start the RMT Server only. When run on an Agent, it will start the Agent on the machine you are running the command from.

This command works both on the RMT Server and Agent

Synopsis

```
rmtadmin start [options] [global option]
```

Options

At least one option must be specified:

RMT Server:

`--all`

Starts all services.

`--master`

Starts the RMT Server.

`--mq`

Starts the message queue service.

`--db`

Starts the database service.

Agent:

`--agent`

Starts the Agent service running on the machine.

Example: `rmtadmin start --all`

rmtadmin status

Checks the status of the application and running services on the machine you are running this command on. For RMT Server, the status report confirms that the RMT Server is connected to the database and message queue. For Agents, the status report confirms that the agent is connected to the RMT Server.

This command can be run both on the RMT Server and the Agent.

This command returns the following:

- Application status (running or not)
- License status
- MQ connection
- Queue details
- Memory used
- Disk space used
- Tableau server version on machine
- Process topology of Tableau Server that is stored in Agent config file.

Beginning with version 2024.2.0, when run on the RMT Server, the command returns:

- Disk space usage warning if messaging tables in the local database are using more space than the warning threshold (4 GB). For more information, see Troubleshoot Messaging Tables Disk Usage Warnings

Synopsis

```
rmtadmin status [global option]
```

rmtadmin stop

Stops the Resource Monitoring Tool services. When this command is run on the RMT Server, it will only stop the RMT Server. When run on a machine where Agent is installed, it will only stop the Agent on that machine.

This command can be run both on the RMT Server and the Agent.

Synopsis

```
rmtadmin stop [options] [global option]
```

Options

At least one option must be specified:

RMT Server:

`--all`

Stops all services.

`--master`

Stops the RMT Server.

`--mq`

Stops the message queue service.

`--db`

Stops the database service.

Agent:

`--agent`

Stops the Agent service running on the machine.

Example: `rmtadmin stop --db`

rmtadmin test-env

Note: Added in version 2021.2

Tests the various connection points between Resource Monitoring Tool and Tableau Server. This includes testing Tableau Server Repository connection, Tableau Server API connections, and Advanced Management capabilities.

This command only works on the RMT Server.

Synopsis

```
rmtadmin test-env [positional parameter][global option]
```

Positional Parameter

`--env`

Optional. The identifier of the environment that the Resource Monitoring Tool is monitoringThis is the system generated identifier. You can find this by going to **Admin > Environments > Edit Environment**. On this page, in the **Environment Details** section, you will see the identifier for the environment.

`--force`

Optional. Use this option to update the baseline immediately even if enough data is not

available. By default the command will wait for 50 successful loads to calculate the baseline

rmtadmin update-baseline

Note: Added in version 2021.4

Updates the baseline for all view loads. Baseline is the 95th percentile of 50 successful initial load times for a view.

This command only works on the RMT Server.

Synopsis

```
rmtadmin update-baseline [options][global option]
```

Options

`env`

Required. This is the system generated identifier. You can find this by going to **Admin > Environments > Edit Environment**. On this page, in the **Environment Details** section, you will see the identifier for the environment.

rmtadmin update-env

Note: Added in version 2021.2

Updates the environment settings.

This command only works on the RMT Server.

Synopsis

```
rmtadmin update-env [options][global option]
```

Options

Option	Required?	Description
--non-interactive	No	Disables all interactive prompts.
--no-test	No	Disables testing API and repository connections.
--name	Yes	The name of the environment.
--id	No	Identifier of the environment used in web interface URLs. This is the system generated identifier. You can find this by going to **Admin** > **Environments > Edit Environment**. On this page, in the **Environment Details** section, you will see the identifier for the environment.
--gateway-url	No	URL used to access the Tableau Server gateway.
--version	No	Tableau Server version that this environment will be monitoring.
--api-username	No	User name of the account used to connect to Tableau Server APIs. The user account should be a Tableau Server administrator with access to all Tableau Server sites.

Option	Required?	Description
--api-password	No	Password of the Tableau Server API user account used to connect to Tableau Server APIs.
--api-password-file	No	The path to the file containing the password of the Tableau Server API user account.
--repository-server	No	This is the server name for the PostgreSQL database that in installed with the Resource Monitoring Tool.
--repository-port	No	The port number of the Tableau Server Repository database.
--repository-username	No	Username used to connect to PostgreSQL database installed with the Tableau Server Repository. Resource Monitoring Tool accesses the Tableau Server Repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the **readonly** database user. For details, see Enable access to the Tableau Server repository.
--repository-password	No	Password for the user account used to connect to the PostgreSQL database that is installed with the Tableau Server .

Option	Required?	Description
		Resource Monitoring Tool accesses the Tableau Server Repository database directly for performance reasons. For this to work, access to the repository must be enabled, with a password set for the **readonly** database user. For details, see Enable access to the Tableau Server repository.
--repository-password	No	The path including the file name where the password for the user account used to connect to the PostgreSQL database that is installed with Tableau Server.
--repository-ssl-mode	No	Tableau Server Repository SSL Mode: **Prefer** SSL or **Require** SSL to configure SSL connections to Tableau Repository. **Disable** to never use SSL to make Tableau Server Repository connections.
--repository-ssl-thumbprint	No	When configuring Tableau Server PostgreSQL to allow direct connections, Tableau Server creates a certificate and keys. You can choose to either supply the thumbprint for the certificate that was

Option	Required?	Description
		generated by Tableau Server, or copy the **server.crt** file to the Resource Monitoring Tool Server machine. If you choose to copy the certificate file, you don't have to supply the thumbprint. For more information, see Configure Postgres SSL to Allow Direct Connections from Clients.

Example:`rmtadmin update-env --name=<new name>`

rmtadmin users

Shows a list of the Resource Monitoring Tool user accounts.

This command works only on the RMT Server.

Synopsis

```
rmtadmin users
```

rmtadmin version

Shows the current version information for the RMT Server when run on the RMT Sever machine. It shows the current version information of the Agent installed on the machine that you are running the command from.

This command can be run both on the RMT Server and the Agent.

Synopsis

```
rmtadmin version
```

rmtadmin ziplogs

Creates a ZIP archive file containing the Resource Monitoring Tool log files.

This command works both on the RMT Server and Agent but only includes the log files from the application the command is run on.

Synopsis

```
rmtadmin ziplogs [positional parameter] [option] [global option]
```

Positional Parameter

`filename`

Name of the output zip file. Defaults to log.zip if no name is provided.

Option

`--force`

Overwrites the existing file.

Example: `rmtadmin ziplogs <zip file name> --force`

Global Option

--help

Shows the help for the command.

Tableau Resource Monitoring Tool Communication Ports

This article lists the communication ports the Tableau Resource Monitoring Tool uses, the default port configuration, and how to change these ports when your network requirements needs them to be different than what is set by default.

RMT Server

You can change or update the port information using:

- rmtadmin master-setup command
- Web interface: From the machine where RMT Server is installed, go to: `http://-localhost/setup/server`

RMT Server Communications

Ports used: 9001, 443, and 80

- Setup initially defaults to port 9001, but changes the default to 80 after initial configuration.
- The SSL self signed certificate on initial installation is for port 443. RMT Server listens on port 80 by default and will attempt to forward traffic to 443 for SSL. Any user specified certificate will also be for port 443.
    - The certificates are used during user traffic to the RMT Server and when the Agent is registered using the Web interface.
    - We generally do not recommend changing port 443. If however, you need to change this due to your environment restrictions, you will need to explicitly list the port number when connecting to RMT Server. If you are registering the Agent using the web interface, you must explicitly specify the port number to connect to the RMT Server.
- Port 80 is used for non SSL traffic and can be changed either during initial setup or post installation.
- The RMT Server will reach out directly to the Tableau Server gateway for REST API calls.
- The RMT Server will reach out directly to the Tableau Server repository to query information if PostgreSQL info is supplied (optional). For information about configuring SSL

between RMT server and the Tableau Server repository, see Pre-Installation Checklist - Tableau Resource Monitoring Tool.

RabbitMQ

RabbitMQ is the component (message queue) used to broker information between Agents and the RMT Server.

Ports used: 5671, 5672

- RabbitMQ is installed and listens to port 5671 (TSL), 5672 (non-TSL).
- TLS communication for RabbitMQ is enabled by default.

PostgreSQL Database

Ports used: 5555

- PostgreSQL database is installed and listens to port 5555. This cannot be changed.
- PostgreSQL has an `admin` user that is used by the Resource Monitoring Tool. The `readonly` user is used to connect to the Resource Monitoring Tool PostgreSQL database in .tds files downloaded from the Resource Monitoring Tool web interface. For more information about downloading the .tds files, see Explore Monitoring Data Using Tableau Data Source Files.
- PostgreSQL requires SCRAM-SHA-256 authentication. For more information, see Explore Monitoring Data Using Tableau Data Source Files.

Agent

Ports used: 9002, 443, 5672

- During installation, the Agent installer open a web browser that is only used for registration that listens on port 9002.

- If Agent registration is done using the web interface, the registration process uses port 443. If the registration is done using the command line, the registration process goes through RabbitMQ and uses port 5672.

Tableau Resource Monitoring Tool Response Headers

This article describes how to set custom response headers in Tableau Resource Monitoring Tool. The ability to do this was added in August 2024 maintenance releases of RMT.

Important: Changes to response headers can break RMT. If you make updates to headers, test afterward to confirm that RMT is working as it should.

Viewing and updating response headers

You can view and change response headers using `rmtadmin get` and `rmtadmin set` commands.

To view a list of currently set response headers, run this command:

```
rmtadmin get server.web.responseheader
```

To set or change a response header:

```
rmtadmin set server.web.responseheader.<some header>
```

This sets the value for the given header.

Headers are a string of the complete header and any key-value pairs. For example, this sets the keep alive timeout and max values:

```
rmtadmin set server.web.responseheader.keep_alive "Keep-Alive:
timeout=5, max=997"
```

Invalid headers

RMT allows you to set invalid response headers. If the response header you specify is not a valid one, RMT will warn you but allow you to set the value. It is your responsibility to verify

that RMT is working properly after updating response headers.

Test RMT after making *any changes* to the response headers.

Manage Users

When you install Resource Monitoring Tool, you create an admin user during configuration. That user has permissions to perform all tasks on the RMT Server. You can later add other users and specify which tasks they are able to perform related to RMT.

Add a local user

To add users to RMT, click the **Admin** menu and select **Users** (this option may not be available if you have not configured an environment).

To add a new user in Resource Monitoring Tool:

1. From the **Admin** menu, select **Users**.
2. Click **New User**.
3. Provide a user name, password, and apply the server roles.
4. Click **Save**.

Add a delegated user

Beginning in version 2023.1 users can be authenticated using their domain account credentials rather than using a separate password stored locally in RMT. When a user configured for delegated authentication signs into RMT, RMT passes the user and the password they entered to the computer operating system (OS) for verification.

To add a new, delegated user in Resource Monitoring Tool:

1. From the **Admin** menu, select **Users**.
2. Click **New User**. The default is a local user.
3. Select **Operating System (Delegated)**.
4. Provide a **Username**.

   Be sure to type *just* the user name. Do not include any domain.

When signing into RMT, the user will provide their standard network password and the OS will verify this.

5. Select the server roles the user should have. For more details about server roles, see below.
6. Click **Save**.

Change user authentication

With version 2023.1.0 or later you can change an existing user from one auth type to another. To change an existing user's authentication:

1. Click the Edit icon.
2. Select the **Authentication Mode** you want:
    - If you are changing to **Operating System (Delegated)** mode, you only need to enter the username (without domain). The user will provide their own password at sign in. And existing password stored in RMT is deleted and cannot be restored.
    - If you are changing to **RMT (Local)** mode, you need to enter a username and password. These are stored locally by RMT.
3. Confirm the **Server Roles** selected for the user.
4. Click **Save** to save your changes.

Server Roles in Tableau Resource Monitoring Tool

The Tableau Resource Monitoring Tool has various server roles that you can assign to a user to give them permissions to perform certain tasks.

The following table lists the available roles and what each roles allows the user to do:

Server Roles	Role Description
Download Log Bundles	Download log files.
Download TDS Files	Download **.tds** files.
Generate Chargeback Reports	Create chargeback reports.
Server/Environment Management	Has permissions to update all configurations.

Server Roles	Role Description
User Management	Create and edit users.
Webhook Management	Manage Slack notification settings.

Troubleshoot authentication issues

For details on how to troubleshoot user sign in and authentication issues, see Troubleshoot User Authentication.

Tableau Resource Monitoring Tool - Incidents

Incidents are reported for events that are unusual and may require human attention. Incidents can be configured for a variety of events either at a global level or can be customized for each environment.

To see current incidents reported for an environment, after selecting an environment, from the **Incidents** menu, select **All Incidents** to see a list of all incidents logged and reported.

You may also receive a notification through email or Slack depending on your settings for notifications. To learn more about how notifications work, see Notifications.

You can generally categorize incidents into two types:.

1. Incidents that are set by default and cannot be changed. We will refer to these as "**system-defined incidents**".
2. Incidents that can be configured by you. We will refer to these as "**configurable incidents**".

All incidents have an incident severity level. The severity level is something that is specifically defined. For system-defined incidents, Resource Monitoring Tool sets the severity levels, and for configurable incidents, you set the severity level depending on what is right for your environment.

Here is the list of severity levels:

Severity Level	Key	Description
Info	`information`	When you wish to be informed of an incident that does not cause service distruption.
Warning	`warning`	Incidents causing possible service disruptions.
Critical	`critical`	Incidents causing major service disruption or the service is down entirely.

System-defined incidents

The following is a list of system incidents that are set by default and cannot be changed.

Incident	Incident Level	Description
Agent Down	`warning`	An incident is logged and reported when either one or more Agents are down, and the RMT Server is unable to communicate with the Agent.
Agent Unli-censed	`critical`	This can happen if Tableau Server is not properly licensed to use Tableau Resource Monitoring Tool, or due to connection issues. For more information on the possible causes, see Agent Incidents
Environment Down	`critical`	An incident is logged and reported when Tableau Server is offline.

Configurable incidents

These incidents can be configured based on your environment characteristics and organ-izational priorities. Following are the events that you can configure severity levels and, or thresholds. The links provide more details on what you can configure for each of the events.

- Extract Failure Incidents
- Hardware Incidents: Includes CPU and memory usage, memory availability, disk space, and disk queue length.
- Hyper Spooling Incidents
- Slow Query Incidents
- Slow Views Incidents

To configure incidents at the global level:

1. From the **Admin** menu, select Global Configuration.
2. In the **Incidents** tab, configure the threshold and severity levels for the incidents.

Global configurations are applied to existing environments or any new environments created unless the environment is using custom thresholds.

Environment Down Incidents

Environment Down incidents will be logged as critical when we detect that your Tableau Server is offline. Following are couple of use cases where Tableau Server is considered offline by Tableau Resource Monitoring Tool:

- Resource Monitoring Tool is unable to get a response from Tableau Server. Resource Monitoring Tool will retry three times before the incident is logged.
- If Resource Monitoring Tool detects that a process is down across all the nodes, an environment down incident will be logged.

Resource Monitoring Tool polls Tableau Server's `http://{Tableau-ServerUrl}/admin/systeminfo.xml` page every 30 seconds (by default) to check the status. If the 30 second polling interval is not sufficient, increase the interval to 60 seconds. If the status is offline, or does not reply within the set value for three consecutive polling intervals then a critical incident is created.

Note: The polling interval cannot be set using the rmtadmin set command. This setting can only be changed by editing the config.json file."

An example `config.json` snippet is below:

```
{
  "background":  {
   "run":true,
   "loglevel":"Information",
    "pollRates":  {
      "processStatus":  30000
      }
    }
}
```

Key	Data Type	Required?	Description
`processStatus`	Number	Optional	The amount of time in milliseconds to pause between Tableau Server status polling.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-/Environment Management** role.

Agent Incidents

Agents Unlicensed (critical)

Incompatible Agent Version (critical)

Agent Message Queue Credential Rotation Failure (critical)

Agent Down (warning)

In an effort to align with our company values of Equality, we have changed non-inclusive terminology where possible. Because changing terms in certain places can causing a breaking

change, we maintain the existing terminology. So, you may continue to see the terms in CLI commands and options, installation folders, configuration files. and other instances. For more information, see About Tableau Help.

Agents Unlicensed

Agents Unlicensed incidents will be logged as critical when we detect that your Tableau Server has not been properly licensed to use Tableau Resource Monitoring Tool.

To monitor a Tableau Server using the Resource Monitoring Tool your Tableau Server must be licensed with the Tableau Advanced Management.

Agents will no longer collect data in the unlicensed state and send to the RMT Server. You can however, continue to see the data that was previously collected when the environment was properly licensed.

To resolve this issue, license Tableau Server with Advanced Management. For more information, see About Tableau Advanced Management on Tableau Server.

> **Note:** It may take up to an hour after adding a valid license to Tableau Server, for the incident to be cleared.

There are other possible causes that can cause an **Agents Unlicensed** incident:

- Tableau Server API credentials may not be correct. In the environment configuration page, verify that the username and password are correct and that the account used has the Server Administrator role on Tableau Server.
- Unable to connect to Tableau Server REST API. Make sure that REST API is enabled on Tableau Server and that Tableau Server is running.

  To resolve the REST API connectivity issue, from the **Admin** menu, select **Environments**, click **Edit Environment** on the environment you want to modify, and in the

Environment Details tab, click the **Test Connection** button in the **Tableau Server REST API** section on the right side of the page.

- The RMT Server was unable to connect to Tableau Server, likely due to a network connectivity or similar issue.

Incompatible Agent Version

Agent incompatible incidents will be logged as critical when one or more Agents in the environment are on a version that is not compatible with the RMT Server version.

To resolve the issue, you need to make sure that the Agent is on a version that is compatible with the current RMT Server version.

Use one of the following to resolve this issue:

We recommend that the RMT Server and all Agents be on the same version.

- If the Agent is on a version that is earlier than the minimum compatible version, then you must upgrade the Agent to at least the minimum compatible version that is listed in the incident report.
- If the Agent is on a version that is later than the RMT Server version, we recommend that you upgrade the RMT Server to that same later version as well. You may need to upgrade other Agents depending on the version that they are currently on.

Upgrading Agents:

Use the following steps to upgrade the Agent:

1. Copy the Agent package to the machines where you have Agents installed. Agents are installed on Tableau Server nodes that you are monitoring. The Agent should be on the same version as the RMT Server or use a version that is compatible with the RMT Server version. The incident details provides the minimum compatible version.

2. Stop the Resource Monitoring Tool Agent service on all Tableau Server nodes using the following command:

   ```
   rmtadmin stop --agent
   ```

3. Upgrade all the Agents by running the following command:

   For RHEL like distributions including CentOS:

   ```
   sudo yum install <pathtoagentinstaller>/<tabrmt-agent-setup-<version>-x86_64.rpm>
   ```

   ```
   sudo /opt/tableau/tabrmt/agent/install-scripts/upgrade-rmt-agent --accepteula
   ```

   For Ubuntu and Debian distributions:

   If you are upgrading from version 2020.4 to 2020.4.1 or later:

   ```
   touch /tmp/tabrmt-agent-upgrading.txt && sudo apt install <tab-rmt-agent-setup-<version>-amd_64.deb>
   ```

   ```
   sudo /opt/tableau/tabrmt/agent/install-scripts/upgrade-rmt-agent --accepteula
   ```

   Note: The touch command is only required when upgrading from 2020.4. If you do not run the touch command before installing the package, your existing version of the Resource Monitoring Tool will be uninstalled before upgrading.

   If you are upgrading from version 2020.4.1 to 2020.4.2 or later:

   ```
   sudo apt install <pathtoagentinstaller>/<tabrmt-agent-setup-<version>-amd_64.deb>
   ```

   ```
   sudo /opt/tableau/tabrmt/agent/install-scripts/upgrade-rmt-agent --accepteula
   ```

4. Confirm that the Agent is running and has been upgraded. You can view Agent registration status by navigating to **Admin** -> **Environments** -> **Edit the Environment** -> **servers** tab to see a full list of Tableau Server nodes and the status of the Agent.

Upgrading RMT Server

Upgrade the RMT Server if one or more Agents are on a later version.

Use the following steps to upgrade RMT Server:

1. Copy the new version of the RMT Server package to the machine where RMT Server is installed.

2. Stop the Resource Monitoring Tool on the RMT server using the following command:

   ```
   rmtadmin stop --master
   ```

3. Stop the Resource Monitoring Tool Agents on all Tableau Server Nodes using the following command:

   ```
   rmtadmin stop --agent
   ```

4. Once the services have been stopped, it is best practice to check for any Resource Monitoring Tool processes that are running after the services have been stopped: Any with tabrmt-agent or tabrmt-master. This does not include PostgreSQL or RabbitMQ. You can check the status using the following command:

   ```
   rmtadmin status
   ```

5. Run the upgrade commands on the RMT Server. This will upgrade the existing version to the new version:

   For RHEL like distributions including CentOS:

   ```
   sudo yum install <pathtomasterserverinstaller>/<tabrmt-master-setup-<version>-x86_64.rpm>
   ```

```
sudo /opt/tableau/tabrmt/master/install-scripts/upgrade-rmt-mas-
ter --accepteula
```

For Ubuntu and Debian distributions:

If you are upgrading from version 2020.4 to 2020.4.1 or later:

```
touch /tmp/tabrmt-master-upgrading.txt && sudo apt install <tab-
rmt-master-setup-<version>-amd_64.deb>
```

```
sudo /opt/tableau/tabrmt/master/install-scripts/upgrade-rmt-mas-
ter --accepteula
```

Note: The touch command is only required when upgrading from 2020.4. If you do not run the touch command before installing the package, your existing version of the Resource Monitoring Tool will be uninstalled before upgrading.

If you are upgrading from version 2020.4.1 to 2020.4.2 or later:

```
sudo apt install <pathtomasterserverinstaller>/<tabrmt-master-
setup-<version>-amd_64.deb>
```

```
sudo /opt/tableau/tabrmt/master/install-scripts/upgrade-rmt-mas-
ter --accepteula
```

6. Confirm that the Agent and RMT Server are running. Start the Agent and RMT Server if they do not automatically restart after the upgrade is complete.

Agent Message Queue Credential Rotation Failure

Starting in version 2021.3, as a security best practice, the credentials for connections between Agent and Rabbit MQ must be unique to that Agent. The unique credentials are created during a new installation or updated when upgrading to 2021.3.

During upgrade, if there are network issues, the credential update process can fail, resulting in a critical incident report. This, however, does not interrupt the upgrade process and the upgrade process will continue. After upgrade is complete, the Resource Monitoring tool will retry daily to create the unique credentials. A critical incident report is created for the initial failure and every subsequently failed retry. During this time, Agent will continue to work using the previous credentials until new credentials have been created.

The issue might resolve by itself, but if it persists, do the following:

- Verify that the Agents can connect to RMT Server.
- Check for firewall issues between Agent and RMT Server.

Agent Down

Agent Down incidents will be logged as warning when the Tableau Resource Monitoring Tool is unable to communicate with the Agents on Tableau Server.

Use the following steps to troubleshoot this issue:

1. Make sure that the hardware that hosts the Agents is running and available to communicate with the RMT Server.
2. Make sure that the Agent service is successfully registered, and is running. You can view Agent registration status by navigating to **Admin** -> **Environments** -> **Edit the Environment** -> **servers** tab to see a full list of Tableau Server nodes and the status of the Agent.

Agent polling and incident creation times

- Agent sends a *heartbeat message* to the Resource Monitoring Tool server every 5 minutes.
- The Resource Monitoring Tool server checks to see if the most recent *heartbeat message* from Agent is less than 15 minutes old.
    - If the latest *heartbeat message* received from the Agent is less than 15 minutes old, then the Agent is considered to be online and no incident is created, and any existing **Agent Down** incidents are cleared.

- If the latest *heartbeat message* received from Agent is more than 15 minutes old, then it is considered to be the first failed attempt. Every minute thereafter, a check is done, two more times. If after three attempts (first attempt and two retries) there is still no recent heartbeat message, then an **Agent Down** incident is created.

> **Note:** Based on the polling interval described above, it takes about 17 (15+1+1) minutes for an **Agent down** incident to be created after the Agent is offline.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-/Environment Management** role.

Extract Failure Incidents

Extract Failure incidents will be logged as a warning when there is an extract failure in Tableau.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-/Environment Management** role.

Hardware Incidents

Hardware incidents monitor the server itself. These can be used to help identify server issues that may affect Tableau Server's performance.

You can set thresholds for the following issues:

- CPU Usage
- Available Memory
- Memory Usage
- Free Disk Space

The following incidents are configured by default when you install a new Tableau Resource Monitoring Tool:

- When the available disk space falls below 10 GB for 10 minutes or more a warning incident is logged, and when available disk falls below 5 GB for 10 minutes or more, a critical incident is logged.
- When available memory falls below 8 GB for over 10 minutes, a warning incident is logged.
- When the CPU usage for the entire server is 80% or more for 5 minutes, a warning incident is logged.

Note: Memory related incidents are configured in binary multiples of bytes.

You can configure thresholds using the RMT Server web interface or by updating the configuration file `config.json`.

Use the RMT Server web interface

To set the thresholds for hardware incidents, under the **Admin** menu, select **Configuration**, and go to the **Incidents** tab.

For **CPU Usage**, set the following:

To set the thresholds for hardware incidents, under the **Admin** menu, select **Configuration**, and go to the **Incidents** tab.

Key	Required?	Description
Severity	Required	See Incident Severity Level.
Process	Required	The threshold applies to the entire Tableau Server or for a single process as specified.
Start Threshold	Required	The CPU usage must surpass the value specified before an incident is created and monitored. Set the percent and the duration for this threshold.

Key	Required?	Description
End Threshold	Optional	The CPU usage that must fall below the value specified before an incident is considered resolved.

For **Available Memory**, set the following:

Key	Required?	Description
Severity	Required	See Incident Severity Level.
Start Threshold	Required	The available memory must fall below the value specified before an incident is created and monitored. Set the percent and the duration for this threshold.
End Threshold	Optional	The available memory must be above the value specified before an incident is considered resolved.

For **Memory Usage**, set the following:

Key	Required?	Description
Severity	Required	See Incident Severity Level.
Process	Required	The threshold applies to the entire Tableau Server or for a single process as specified.
Start Threshold	Required	The memory usage must be equal to the value specified before an incident is created and monitored. Set the percent and the duration for this threshold.
End Threshold	Optional	The memory usage must be below the value specified before an incident is considered resolved.

For **Free Disk Space**, set the following:

Key	Required?	Description
Severity	Required	See Incident Severity Level.

Key	Required?	Description
Start Threshold	Required	The free disk space must fall below the value specified before an incident is created and monitored. Set the percent and the duration for this threshold.
End Threshold	Optional	The free disk space must be above the value specified before an incident is considered resolved.

For **Disk Queue Length**, set the following:

Key	Required?	Description
Severity	Required	See Incident Severity Level.
Start Threshold	Required	The disk queue length must be equal to the value specified before an incident is created and monitored. Set the percent and the duration for this threshold.
End Threshold	Optional	The disk queue length must be below the value specified before an incident is considered resolved.

Use the configuration file (config.json)

An example `config.json` snipped defining two hardware incidents:

```
{
 "monitoring":  {
    "incidents":  {
      "triggers":  [
        {
          "counter":  "DiskSpaceAvailableKB",
          "severity":  "warning",
          "threshold":  1048576
        },
        {
          "counter":  "ProcessorTimePercent",
```

```
            "severity":  "warning",
            "threshold":  0.95,
            "thresholdDuration":  300000,
            "endThreshold":  0.90,
            "endThresholdDuration":  5000
          }
        ]
      }
    }
}
```

- The **DiskSpaceAvailableKB** incident will trigger a warning once the available disk space falls below 10 GB.
- The **ProcessorTimePercent** incident will trigger a warning once the CPU has had at least 95% utilization for over 5 minutes. The incident will be considered resolved once the CPU is below 90% utilization for 5 seconds.

The default settings may or may not meet your requirements, and can be changed based on your environment. As an example, for an environment whose identifier was “staging-environment” to trigger a warning when the available disk space falls below 2 GBs, the configuration would look like:

```
{
  "environments":  {
    "staging-environment":  {
     "monitoring":  {
        "incidents":  {
          "triggers":  [
            {
              "counter":  "DiskSpaceAvailableKB",
              "severity":  "warning",
              "threshold":  2097152
            }
          ]
        }
      }
    }
```

```
    }
}
```

Key	Data Type	Required?	Description
`counter`	String	Required	The identifier for the hardware incident to monitor. available options are: • `ProcessorTimePercent` • `DiskSpaceAvailableKB` • `DiskQueueTotalLength` • `MemoryAvailableKB` • `MemoryCommittedKB`
`severity`	String	Optional	See Incident Severity Level. Default value: `Warning`
`threshold`	Number	Required	The threshold that must be surpassed before an incident is monitored.
`thresholdDuration`	Number	Optional	The amount of time in milliseconds to monitor the situation before triggering an incident. If not specified, an incident will be triggered as soon as the `threshold` is reached.
`endThreshold`	Number	Optional	The threshold that must be surpassed before an incident is considered resolved.
endThresholdDuration	Number	Optional	The amount of time in milliseconds to monitor the situation before completing the incident. If not specified, an incident will be resolved as soon as the `endThreshold` is reached. If `endThreshold` is not defined, then `threshold` is used.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-/Environment Management** role.

Hyper Spooling Incidents

Hyper Spooling incidents will be logged as a warning when Hyper queries are spooled to disk. Typically, this happens when there is not enough available memory and the Tableau Server Data Engine process shifts to spooling by temporarily writing to disk. The Tableau Server Data Engine topic describes this in more detail in the memory usage section.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-/Environment Management** role.

Slow Query Incidents

Slow Query incidents are triggered when any data query takes too long to complete. By default, a Slow Query incident will trigger a warning if any data query takes at least 30 seconds to execute.

You can configure thresholds using the RMT Server web interface or by updating the configuration file `config.json`.

Use the RMT Server web interface

To set the thresholds for slow query incidents, under the **Admin** menu, select **Configuration**, and go the **Incidents** tab.

Key	Required?	Description
Severity	Required	See Incident Severity Level.
Duration	Required	The minimum duration for a query to be considered slow. Val-

Key	Required?	Description
		ues are in seconds.

Use the configuration file (config.json)

An example `config.json` snippet defining a Slow Query incident:

```
{
 "monitoring":  {
    "incidents":  {
      "triggers":  [
        {
          "counter":  "DataQueryDuration",
          "severity":  "critical",
          "threshold":  45000
        }
      ]
    }
  }
}
```

This Slow Query incident will trigger if a data query takes at least 45 seconds to execute.

Incidents can be configured per environment. As an example, for an environment whose identifier was “staging-environment” to trigger a warning when a data query takes longer than 30 seconds to execute, the configuration would look like:

```
{
  "environments":  {
    "staging-environment":  {
     "monitoring":  {
       "incidents":  {
          "triggers":  [
            {
              "counter":  "DataQueryDuration",
              "severity":  "warning",
```

```
                "threshold":  30000
              }
            ]
          }
        }
      }
    }
}
```

Key	Data Type	Required?	Description
`counter`	String	Required	Slow Query incidents use the `DataQueryDuration` identifier.
`severity`	String	Optional	See Incident Severity Level. Default value: `Warning`
`threshold`	Number	Required	The minimum duration for a query to be considered slow. Values are in milliseconds.

Who can do this

Resource Monitoring ToolAdministrator or a Resource Monitoring Tool user with **Server-/Environment Management** role.

Slow Views Incidents

Slow View incidents are triggered when any view request takes too long to load.

By default, a slow view incident will trigger a warning if any view request takes at least 1 minute to load.

Configure Slow View Incident Thresholds

To set the thresholds for slow view incidents, under the **Admin** menu, select **Configuration**, and go the **Incidents** tab.

Key	Required?	Description
Severity	Required	See Incident Severity Level.
Duration	Required	The minimum duration for a query to be considered slow. Values are in seconds.
Tableau Users	Required	This threshold is can be either applied to specific users when they make view requests or for all users. Default is **All Users**.
Content	Required	Content includes workbooks and views. This threshold can be applied to a single view or all views. If a workbook is specified, the threshold would be applied to all the views in that workbook. To specify a view, use the share URL. For more information, see How View URLs are structured. You can also exclude certain views or workbooks by specifying exceptions. Default is **All Views**

Only one incident is created per view request. The Resource Monitoring Tool evaluates all of the incident triggers, and if multiple triggers match a specific view request, then the triggers are ranked in order of priority and specificity. The highest ranking trigger is used to create the incident.

For example, a trigger with a severity of critical is ranked higher than a severity of warning.

Encrypted Data Collection

To make sure the data collection from Tableau Server to Tableau Resource Monitoring Tool is encrypted, communications between RMT Server and Agents, and connections to Tableau

Server Repository have to be enabled to use encrypted messaging.

For versions 2022.3 and later

- If RabbitMQ is installed on the same machine as the RMT Server (local configuration), Tableau Resource Monitoring Tool has built-in encrypted communications between RMT Server and agents. There is no set up required for encryption between RabbitMQ and RMT Server.
- If RabbitMQ is hosted external to RMT Server (external configuration), you must follow the same steps as described in the For versions 2022.2 and earlier section in this topic

You still need to enable and configure encrypted communications to Tableau Server repository database in both cases stated above.

Tableau Repository SSL Configuration

1. Make sure Tableau Server is configured to use SSL connections for internal Postgres connections. For more information, see Configure SSL for Internal Postgres Communication. The Resource Monitoring Toolallows you to use either the certificate file or thumbprint for the SSL connections. If you plan to use the certificate file, copy the certificate file generated by Tableau Server for internal Postgres SSL connections, to the machine where you plan to install the RMT Server. For more information, see Configure Postgres SSL to Allow Direct Connections from Clients.

2. On the RMT Server web interface, from the Admin menus, select Environments. Click on the edit environment icon.

   In the **Tableau Repository Configuration** section:

   1. In the **SSL Mode** drop down box, select **Prefer SSL** or **Require SSL** to configure SSL connections to Tableau Repository. Choosing **Disable** means SSL will never be used to make Tableau Server Repository connections.

      In the **Prefer SSL** mode, the Resource Monitoring Tool will use SSL in the first attempt, and if that fails the subsequently attempts a non-encrypted connection.

In the **Require SSL** mode, if the SSL connection fails, the connections to Tableau Server Repository will fail entirely. In this case, Tableau Server REST API connections will be used to communicate with Tableau Server.

2. You can choose to either supply the thumbprint that was generated by Tableau Server, or copy the **server.crt** file to the Resource Monitoring Tool Master Server machine. If you choose to copy the certificate file, you don't have to supply the thumbprint. For more information, see Configure Postgres SSL to Allow Direct Connections from Clients.

For versions 2022.2 and earlier

Encrypted communication between the Agent(s) and RMT Server is possible by performing the following:

- Configuring RabbitMQ with SSL/TLS certificates.
- Configuring the RMT Server and agent(s) to enable encrypted messaging.
- Configuring encrypted connections to Tableau Repository.

RabbitMQ Setup

For details on RabbitMQ server setup please reference RabbitMQ's documentation for TLS Setup.

Tableau Resource Monitoring Tool Setup

After RabbitMQ has been configured for TLS all client applications: the Tableau Resource Monitoring Tool RMT Server and all Agents will need to be configured to enable encrypted messaging. Do the following on the RMT Server web interface:

1. On the machine where RMT Server is installed, go to: *http://<hostname>/setup/server*.
2. In the Message Queue section, check the Enable TLS check box, and provide the Certificate Host Name.
3. Update the port information if needed.

When configuring the RMT Server and agent(s) for encrypted messaging:

- Both the `enabled` flag and the `certificateHostName` must be configured for encryption to be enabled.
- The `certificateHostName` variable must match the canonical name (CN=) on the server certificate or the connection will fail.
- The `port` number will likely need to be changed based on the TLS port you configured on RabbitMQ.

> **Note:** If the Agents were already registered before SSL was configured, then you must re-register the agent. To do this, download the new bootstrap file and re-register the Agent using the new bootstrap file. For more information on re-registering the Agent, see Re-registering an Agent.

Who can do this

To configure encrypted data collection you must be both a Tableau Server Administrator and a Resource Monitoring Tool administrator.

Hardware Changes to RMT Server - Tuning PostgreSQL Database

Sometimes you may need to upgrade or change hardware on the machine where RMT Server is installed. Whenever you make any hardware changes, specifically to memory and CPU, it would be beneficial to tune the underlying PostgreSQL database that is installed with the RMT Server.

When you install RMTr Server, the setup program configures the PostgreSQL database to optimize the use of available hardware on the machine. Example optimizations include managing the buffer and cache size. When you change the hardware on your machine, it may affect the performance.

Use the following steps to make sure the configuration is updated to reflect the change in hardware:

1. Connect to the RMT Server as a user with complete sudo access.
2. Navigate to the "master" directory and run the following command:

```
sudo /opt/tableau/tabrmt/master/tabrmt-master optimize
```

You should see a message that indicates that the configuration file has been updated. It will also state that a restart of the PostgreSQL service is required for the changes to take effect. The updates to the configuration file does not require downtime of the server.

3. Restart the PostgreSQL database. You can do this during off-work hours when the Resource Monitoring Tool is not in use.

Tableau Server Topology Changes

This article will help you understand what you need to do when you make certain topology changes to Tableau Server. The changes include: adding or removing a node, adding or removing a process from a node, and changing the number of processes on a node.

Adding a Node

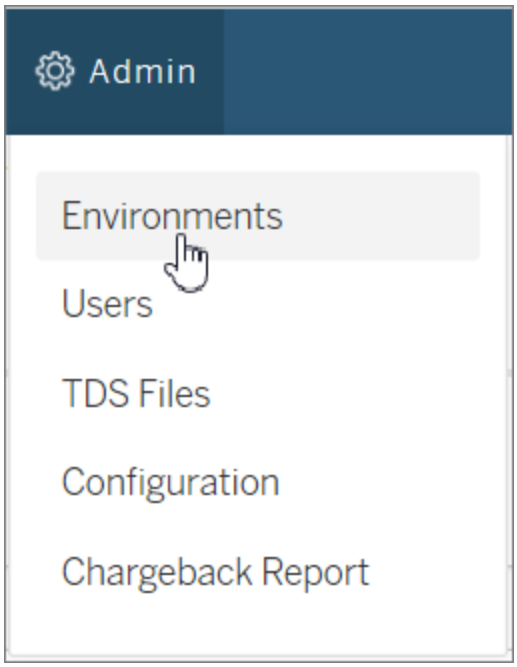
When you add a new node to Tableau Server, you must install the agent on this node. For more information on installing Agent on Tableau Server cluster, see Install the Tableau Resource Monitoring Tool. Until you install the agent, information about this node will not be included in the reports.

Removing a Node

When you remove an existing node from Tableau Server cluster, you must update the environment on the RMT Server.

Use the following steps to update the environment:

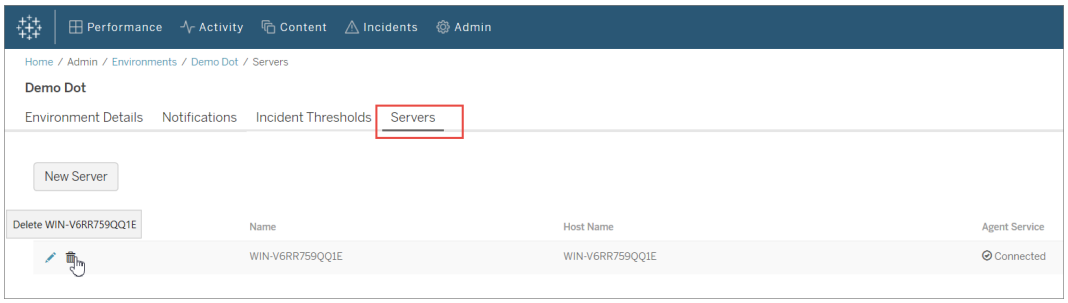
1. Log into the RMT Server web interface.
2. From the **Admin** menu, choose **Environments**. Select the environment that should reflect this change.



3. Choose **Edit Environment**. This opens more detailed information about the environment.

4. From the **Servers** tab, select the node that was removed from Tableau Server, and choose **Delete**.

> Note: The host name should match the name or the IP address of the node.



Re-registering an Agent

There can be situations when you might want to re-register an Agent. One such use case might be when you want to point one node or the entire Tableau Cluster to a different environment. Another example might be if you had to re-image a Tableau Server node and need to install and re-register the Agent. Use the following instructions to re-register the Agent.

1. On the RMT Server web interface, navigate to **Admin** -> **Environments**. Select the environment. From the **Servers** tab, click delete to completely remove the Agent registration for this node on this environment. This also removes historical monitoring data that has been collected on this node. Use this for scenarios where you are ok with losing the history. For example, if you planning to point this node to a new environment, it might be acceptable to remove the historical data and start fresh.

Note: In the RMT Server web interface, each node in a Tableau Server is referred to as a **Server**. The entire Tableau Server constitutes an environment. For more information, see Concepts

Who can do this

To make topology changes, you must be both a Tableau Server Administrator and a Resource Monitoring Tool Administrator.

Tableau Resource Monitoring Tool Log Files

Log Files

Tableau Resource Monitoring Tool includes many components. Each component maintains its own set of log files.

By default, 31 log files with a max file size of 1 GB each will be retained in each set. Excess log files beyond that will be deleted automatically.

In an effort to align with our company values of Equality, we have changed non-inclusive terminology where possible. Because changing terms in certain places can causing a breaking change, we maintain the existing terminology. So, you may continue to see the terms in CLI commands and options, installation folders, configuration files. and other instances. For more information, see About Tableau Help.

Components

Component **For additional information about services installed with RMT, see Resource Monitoring Tool (RMT) Services.**	**Log File Location** `/var/opt/tableau/tabrmt/master/logs`	**Description**
Backgrounder	`background\YYYYMMDD-pts.log`	Logs generated by the general background processor.
Director	`director\YYYYMMDD-pts.log`	Logs generated by the director background pro-

Component **For additional information about services installed with RMT, see Resource Monitoring Tool (RMT) Services.**	Log File Location `/var/opt/tableau/tabrmt/master/logs`	Description
		cessor.
Host	`host\YYYYMMDD.log`	Logs generated by the host.
Web Server	`web\YYYYMMDD-pts.log`	Logs generated by the web server.
RabbitMQ (Message Broker)	`rabbitmq/*.log`	Logs generated by RabbbitMQ
PostgreSQL (local data repository)	`pgsql/*.log and *.csv`	Logs generated by the local PostgreSQL database.

For advanced troubleshooting, verbose logging can also be enabled for more internal activities. These activity logs are stored in the associated component's log directory.

Activity	Log File Location	Description
Database	`logs\*\YYYYMMDD-ef.log`	Logs of internal database queries. By default, only errors are written to this log. Enable by changing `db.logLevel` to

Activity	Log File Location	Description
		`Debug` or higher.
Message Queue	`logs\*\YYYYMMDD-mq.log`	Logs of internal message queue communication. By default, this log will not be generated. Enable by changing `mq.logLevel` to `Debug` or higher.

Log Level Configuration

Resource Monitoring Tool log levels can be adjusted in the following ways:

- Resource Monitoring Tool web interface: From the Admin menu, navigate to Configuration and select the Advanced tab. You can set the log levels and also specify if you want to include database queries and message queue communications.

  Logging levels you can set using the web interface:

Level
Default: Includes Information, Warning, Error, and Critical levels
Verbose: Includes Information, Warning, Error, Critical (everything in the default) and Debug levels

- Settings in the `config.json` files. Log levels are configured independently for each component. The default config file paths are below. This may vary depending on your installation folder.

Install	Default Location
• Tableau Resource Monitoring Tool	`/var/opt/tableau/tabrmt/master/config.json`
Tableau Resource Monitoring Tool Agent	`/var/opt/tableau/tabrmt/agent/config.json`

Logging levels that can be configured using the configuration file:

Level
Trace
Debug
Information
Warning
Error
Critical
None

Below is a snippet demonstrating some default log level settings:

```
{
  "db":  {
    "logLevel":  "Error"
  },
  "mq":  {
    "logLevel":  "Warning"
  },
  "server":  {
    "background":  {
      "logLevel":  "Information"
```

```
    },
    "director": {
      "logLevel": "Information"
    },
    "web": {
      "logLevel": "Information"
    }
  }
}
```

Sending Log Files to Tableau Customer Support

If you are working with Tableau Support and they ask you to send log files, zip the files up before you send them:

1. Connect to RMT Server and each server that has the Resource Monitoring ToolAgent installed.
2. Open a command prompt and run: `rmtadmin ziplogs <output file path>`, to create a ZIP archive of the log files.

For more information on sending log files to Tableau, see the Tableau Knowledge Base.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-/Environment Management** role.

Tableau Log Files

Tableau Resource Monitoring Tool agents monitor the Tableau log files in near real time and send log messages to the RMT Server for processing and reporting.

Below is a list of the specific Tableau log files that are monitored.

All directory paths are relative to the Tableau Server data directory. By default, this is located at: `/var/opt/tableau/tabrmt/data/tabsvc`

Directory	File Name
`logs\backgrounder`	`backgrounder-*.log`
`logs\httpd`	`*.log`
`logs\vizportal`	`vizportal-*.log`
`logs\dataserver`	`dataserver_*.txt`
`logs\vizqlserver`	`vizqlserver_*.txt`

Upgrading Tableau

When upgrading a Tableau Server installation that is monitored by Tableau Resource Monitoring Tool, there are a few additional steps that need to be followed.

Ensure Resource Monitoring Tool supports the new Tableau version

Before upgrading Tableau, you'll want to make sure that the version of Resource Monitoring Tool you have installed supports the version of Tableau that you are upgrading to.

The quickest way to check is to log in to the Resource Monitoring Tool web interface and navigate to the **Admin** | **Environments** screen. Edit the environment you are upgrading and check the **Tableau Version** drop down for the version you are upgrading to.

Stop Agents

Resource Monitoring Tool agents should be stopped while you upgrade Tableau Server. Follow these steps on each machine in your Tableau Server cluster:

Run the following command as the `tabrmt-master` user:

```
sudo su --login tabrmt-master
```

```
rmtadmin stop --agent
```

Upgrade Tableau

Follow the normal process for upgrading Tableau Server. This process is outlined in the Tableau Server documentation.

Update Tableau Version in Resource Monitoring Tool

1. Log in to the Resource Monitoring Tool web interface.
2. Go to **Admin** | **Environments**
3. Edit the environment you upgraded.
4. Modify the **Tableau Version** to match

Restart Agents

After the upgrade is complete you are ready to restart the Resource Monitoring Tool agents. Follow these steps on each machine in your Tableau Server cluster:

Run the following command as the `tabrmt-master` user:

```
sudo su --login tabrmt-master

rmtadmin restart --agent
```

Who can do this

To upgrade Resource Monitoring Tool, you must have all the following:

- Full sudo access for the user on machine you are installing Resource Monitoring Tool.
- Tableau Server Administrator site role.
- Resource Monitoring Tool Administrator account.

Monitor Tableau Server Performance

Tableau Resource Monitoring Tool is used to monitor and analyze Tableau Server health and performance. Performance, usage and hardware metrics are collected through the Agents installed on the Tableau Server nodes and sent to the RMT Server. The aggregated and analyzed data is then displayed in the form or charts and views on the web interface of the RMT Server.

The web interface of the RMT Server has built-in charts and views that you can use to identify what is causing slow load times, extract failures, and other critical issues. For more information, see Monitor Tableau Server Performance with Tableau Resource Monitoring Tool.

You can also download the data that is used to create the pre-built charts and explore it further. For more information, see Explore Monitoring Data Using Tableau Data Source Files.

Monitor Tableau Server Performance with Tableau Resource Monitoring Tool

Tableau Resource Monitoring Tool makes it easy to detect and resolve health and performance problems within your Tableau Server environments. One instance of the Resource Monitoring Tool can provide a single unified interface for administrators to monitor multiple Tableau Servers.

The Resource Monitoring Tool consists of two components:

- Agent, which collects resource usage and performance recording of interactions on Tableau Server.
- RMT Server, which aggregates and displays this performance data in the form of charts on a web interface.

The Resource Monitoring Tool may not be able to provide information for Tableau Server process that are external:

- External Repository: The process status is not monitored and not included in the chartss
- External File Store: The process status is not monitored and the extract file size information is not tracked. The extract failure and time taken to complete is still available.
- External Gateway: The process status is not monitored and not included in the charts.

Pre-built Charts

The web interface of the RMT Server has built-in dashboards and charts that can be used to identify performance bottlenecks and issues. These built-in charts and metrics can be found on the **Performance**, **Activity**, and **Content** pages. You can also set up alerts (called **incidents**) to report outliers or unusual behavior based on thresholds that you configure.

Here is a full list of capabilities that the Resource Monitoring Tool offers to help with monitoring your Tableau Server:

Capability	Component	Where to find it on RMT Server
Configuring Incidents/Alerts	• Hardware resources at Tableau Server node and process level. • View load times. • Query times. • Tableau Server node down events.	**Admin** -> **Environments** ->**Edit Environment** -> **Incidents** or **Notifications** tabs. • Use the **Incidents** tab to configure incident thresholds. • Use the **Notifications** tab to configure when and how you want to receive notifications.
Hardware Resources	• CPU • Memory • Disk • Network	For a high level view, go to **Performance** -> **Select environment** ->**Environment** For a more detailed view, go to **Performance** -> **Select environment** -> **Servers**

Tableau Server Usage	• Concurrent users
	• Aggregated view load times across the server
	• Background tasks
	  • Duration, state, and detailed errors for failed tasks.
	  • Log snippets for failed tasks.
	• Query details
	  • Duration, state,
		Most of this information is on the **Environment** tab of the performance dashboard.
		Performance -> **Select environment** ->**Environment**
		To see query details, go to **Activity** -> **Data Queries** and select a query for more details.
		To see log information, go to **Activity** -> and make a selection from the list to

Capability	Component	Where to find it on RMT Server

	and query text for long-running or expensive queries.
	• Log information for:
	  • Individual view load times.
	  • VizQL session information.
	  • Duration of events during view loads.
	  • Query text, load time, and number of rows returned
	  • Background tasks and run times.
		see more details.

Custom Charts

You can also download the data that is used to create the pre-built charts and explore it further in Tableau. For more information, see Explore Monitoring Data Using Tableau Data Source Files.

Who can do this

Any Resource Monitoring Tool user can view the charts.

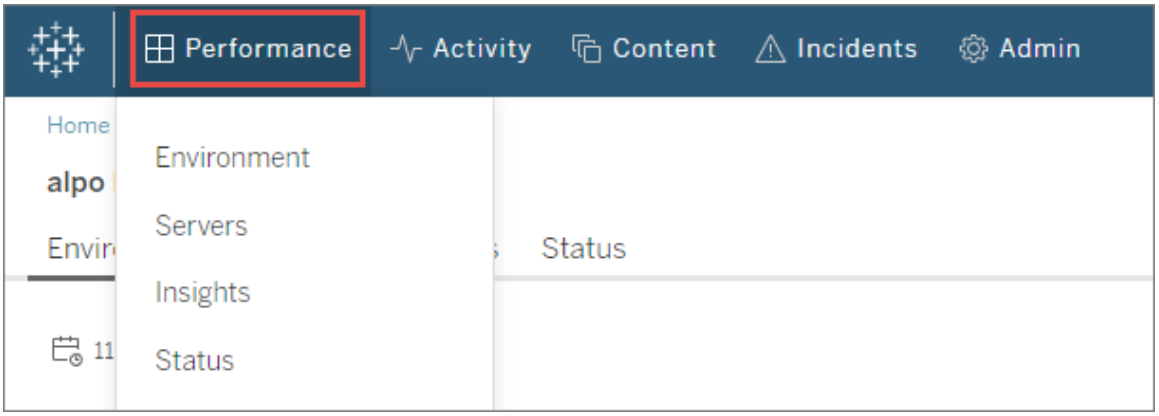
Tableau Resource Monitoring Tool Performance Charts

The Tableau Resource Monitoring Tool includes dashboards that help you monitor and analyze various performance metrics like hardware resource usage, user activity, and the status

of Tableau Server processes. They can be useful in identifying performance bottlenecks and the overall health of your Tableau Server.

This topic describes the information on the charts on the **Performance** page. The performance page provides a dashboard view of the overall health of the Tableau Server and includes the following tabs:

1. **Environment:** A dashboard of the performance metrics specific to the environment you selected.
2. **Servers:** Focuses on the hardware resource metrics.
3. **Insights:** Highlights the slowest views and the longest extract refreshes.
4. **Status:** Tableau Server processes status. Information about each process on the nodes whether they are active, busy, or down. This is similar to what you might see on the Tableau Server Settings page.



Environment Tab

The charts on the environment tab give you an all-up view of the health of the Tableau Server.

- The left side of the page has resource-related metrics: Performance, Tableau Processes, and Background Tasks.
- The right side charts focus on the user activity and the impact: Concurrent Users, Slow View Load Requests, and Total View Load Requests.

Navigation Tips:

At the top left corner of the page, you can select a time line that is applied to all the charts on

this page. You can also select a time range by selecting a part of a specific chart (range selection).
- When you make a range selection in the hardware performance and background task charts, the time line acts as a filter for all the charts on this tab.
- When you make a range selection in the slow view load, and user activity charts, you will be automatically be navigated to the Activity page which provides more details for the selected time range.

Performance Chart

The information on this chart shows you the overall health and usage of the hardware resources for each Tableau Server node. The hardware resources included in this chart are CPU, memory, disk queue, and network.

> **Note:** Network information is not available and not currently supported for Tableau Server running on Linux.

Tableau Processes Chart

Use this chart to get more details about a specific node and the processes running on that node. **Starting in 2021.4**, almost all Tableau Server processes are tracked with a few exceptions like cluster controller and process that are configured external to Tableau Server cluster, like the External Repository, External File Store, and Independent Gateway. This chart allows you to identify the processes that are driving CPU or memory utilization on a particular node. The chart is dynamically updated to highlight the top ten processes that are using the most resources on a node for the selected time period.

> **Note:** Some Tableau Server sub-processes like tabprotosrv, postgres, and gateway, are not capturedby RMT Server and can cause apparent discrepancies between the Performance chart and the sum of all the values from the Tableau Processes chart.

Background Tasks Chart

The chart is an overview of the volume of background tasks, categorized by the task type. Use the toggle button to see the total number of tasks, the median duration of those tasks, and a histogram that shows you the run-time for the selected time period. This can be a quick way to drill down to see which tasks are taking a long time to run.

Concurrent Users Chart

This chart shows you the number of users that have been sending requests in the time period that is selected.

> **Note:** When you select a portion of this chart, it automatically takes you to the related activity page and shows you the information for that specific time period.

Slow View Load Requests Chart

This chart uses a **baseline** that is established for each view and is then uses it to compare the time it takes to render that view to determine if the view is taking longer than expected.

In version 2021.4, the baseline is established by calculating the median value of the first 10 times a specific workbook is rendered successfully.

In version 2021.4.1 and later, the baseline is established by calculating the 95th percentile of 50 times a specific workbook is rendered successfully.

Once the baseline is established, every time that same workbook is rendered in the future, the time taken for a workbook to load is then compared to its baseline. Depending on whether the time taken to render the view falls within the expected range or at varying degrees outside of the expected range, they are categorized as follows:

- **Normal:** <=2x baseline
- **Long:** >=2x baseline

- **Very Long:** >=4x baseline
- **Failed**: Failed to load or resulted in an error

This baseline and the comparisons only apply to initial rendering of the view. It does apply to subsequent actions like filter selections. Also, when you publish a new version of the workbook, it triggers a recalculation of the baseline.

Note: When you select a portion of this chart, it automatically takes you to the related activity page and shows you the information for that specific time period.

For a detailed walk through on how to use this chart and investigate performance with view rendering, see Investigating Slow View Load Requests.

Total View Load Requests Chart

This chart gives you an overall feel for how many views have been rendered over a period of time. This chart is useful to assess the scope of the impact of a particular incident.

Note: When you select a portion of this chart, it automatically takes you to the related activity page and shows you the information for that specific time period.

Server Tab

The charts on this tab provide a more detailed view of the **Performance** chart on the **Environment** tab.

Insights Tab

Slowest Views

This chart shows the slowest views ordered by average view load duration. The orange dot indicator represents that view's longest load duration. The time period that is represented here is the full range of data that is available, default is 2 weeks.

Longest Extract Refreshes

This chart shows the slowest Refresh Extract tasks ordered by duration. Both data source extracts and workbook extracts are considered. They can be distinguished by the icon next to their name. The time period that is represented here is the full range of data that is available, default is 2 weeks.

Status Tab

This tab lists the Tableau Server processes and their status on each node of the Tableau Server Cluster. This does not include Tableau Server processes that are configured external to Tableau Server cluster like External Repository, External File Store, and Independent Gateway.

Who can do this

Any Resource Monitoring Tool user can view the charts.

Related Topics

- Monitor Tableau Server Performance with Tableau Resource Monitoring Tool
- Tableau Resource Monitoring Tool Activity Pages
- Tableau Resource Monitoring Tool Content Pages

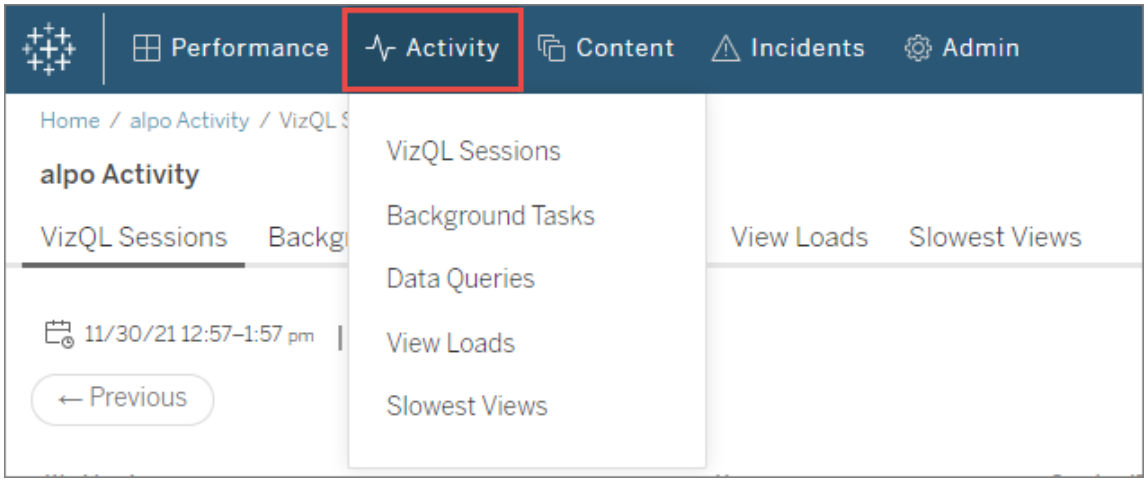
Tableau Resource Monitoring Tool Activity Pages

The Tableau Resource Monitoring Tool includes dashboards that help you monitor and analyze various performance metrics like hardware resource usage, user activity, and the status of Tableau Server processes. They can be useful in identifying performance bottlenecks and the overall health of your Tableau Server.

This topic describes the information on the **Activity** page. The activity page provides the next level of detail to what is seen on the charts on the **Performance** page. The filter selections on either of these dashboards are carried over to the other so you can see corresponding information as you are trying to identify any performance issues.

The activity page includes detailed information about the following:

- VizQL Sessions
- Background Tasks
- Data Queries
- View Loads
- Slowest Views



VizQL Sessions

Shows you a list of all the VizQL sessions in the selected time range. A VizQL session is a set of interactions that a user has with a workbook on Tableau Server.

Click a session ID to see a summary chart and more details such as the requests made during the session, request duration, and related workbook sessions. You can also see any related data queries, incidents that have been reported, and the environment activity filtered to the same time period as the VizQL session's time frame.

Background Tasks

Shows you a list of all the background tasks in the selected time range.

Click the start time of the task to see more details about the background task including a summary that shows you a comparison of the time it took to complete the task versus the average duration of time taken to complete similar tasks. You can also see any related incidents that have been reported and the overall environment activity during the time this task was run.

Click the site name to see more information about the site including the number of workbooks, views, and VizQL sessions for each project on that site.

Data Queries

Shows you a list of all the data queries during the selected time range.

Click on a query to see query performance details, the full query text, and the connection details.

View Loads

Shows you the list of rendered views in the selected time range.

The **Load Time Severity Category** filter allows you to filter views are taking much more time to load than normally expected. This uses the same baseline concept and comparisons used on the **Slow View Load Request** Chart. The baseline is established by calculating the median value of the first 10 times a specific workbook is rendered. Once the baseline is established,

every time that same workbook is rendered in the future, the time taken for a workbook to load is then compared to its own baseline.

The categories for the load time severity are as follows:

- **Normal:** <=2x baseline
- **Long:** >=2x baseline
- **Very Long:** >=4x baseline
- **Failed:** Failed to load or resulted in an error
- **Processing:** The baseline calculation is in process and is not yet established.

Slow Views

Shows you the list of all views that have taken the longest time in descending order. This list is based on average load duration time rather than a comparison against a baseline. You can narrow this list by filtering by duration, publisher, site or a specific view.

- Click on the view to see more details about the load times You can also see any related incidents that have been reported and the overall environment activity during the time this task was run.
- Click on the workbook to see more details about a specific workbook.

Who can do this

Any Resource Monitoring Tool user can view the charts.

Related Topics

- Monitor Tableau Server Performance with Tableau Resource Monitoring Tool
- Tableau Resource Monitoring Tool Performance Charts
- Tableau Resource Monitoring Tool Content Pages

Tableau Resource Monitoring Tool Content Pages

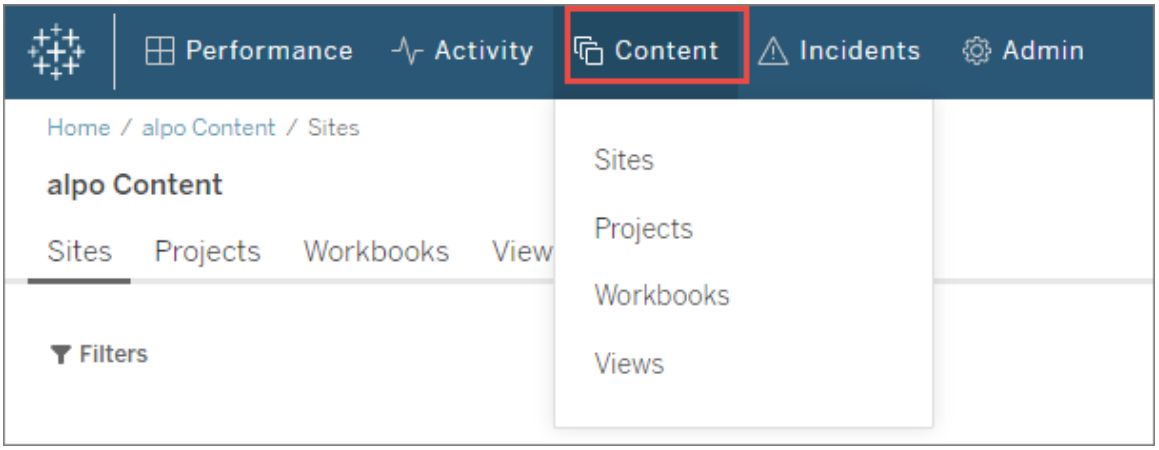
The Tableau Resource Monitoring Tool includes dashboards that help you monitor and analyze various performance metrics like hardware resource usage, user activity, and the status of Tableau Server processes. They can be useful in identifying performance bottlenecks and the overall health of your Tableau Server.

This topic describes the information on the charts on the **Content** page. This page is the main way to look at the details for a specific content item. It is a useful place to start when you need to investigate performance for a specific workbook or a view.

The content page includes detailed information about the following:

- Tableau Sites
- Projects in each Tableau Site
- Workbooks
- Views

You can see performance metrics for the VizQL sessions, data queries related to a project, workbook, or a view. You can also see any related incidents specific to workbooks or views.



Sites

Shows you a list of sites on your Tableau Server environment including the total number of projects, workbooks, and VizQL sessions.

Projects

Shows you a list of all the projects on a site. Click on the project to see a list of all the workbooks, views, and related VizQL sessions, and data queries.

Workbooks

Shows you a list of all the workbooks on a site. Click on a workbook to see the load times, VizQL session information, related data queries, and any reported incidents specific to this workbook.

Views

Shows you a list of all the Views on a site. Click on view to see the load times for the view, VizQL session information, related data queries, and any reported incidents specific to the view.

Who can do this

Any Resource Monitoring Tool user can view the charts.

Related Topics

- Monitor Tableau Server Performance with Tableau Resource Monitoring Tool
- Tableau Resource Monitoring Tool Performance Charts
- Tableau Resource Monitoring Tool Activity Pages

Investigating Slow View Load Requests

The **Slow View Load Requests** chart on the performance page is a useful metric to understand the performance of views and the resulting impact on user interactions on Tableau Server.

The Slow View Load Requests chart shows when views are rendering more slowly than normal on Tableau Server. To do this, the chart uses a **baseline** that is established for each view and is then uses it to compare the time it takes to render that view to determine if the view is taking longer than expected.

In version 2021.4, the baseline is established by calculating the median value of the first 10 times a specific workbook is rendered successfully.

In version 2021.4.1 and later, the baseline is established by calculating the 95th percentile of 50 times a specific workbook is rendered successfully.

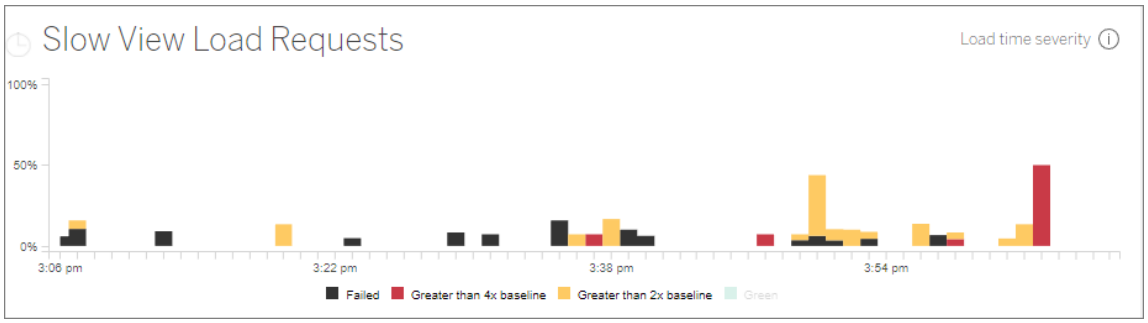
Once the baseline is established, every time that same workbook is rendered in the future, the time taken for a workbook to load is then compared to its baseline. Depending on whether the time taken to render the view falls within the expected range or at varying degrees outside of the expected range, they are categorized as follows:

- **Normal:** <=2x baseline
- **Long:** >=2x baseline
- **Very Long:** >=4x baseline
- **Failed:** Failed to load or resulted in an error

Note: This baseline and the comparisons only apply to initial rendering of the view. It does apply to subsequent actions like filter selections. Also, when you publish a new version of the workbook, it triggers a recalculation of the baseline.

The chart shows the percentage of view loads that are outside of the normal range for the selected time range. So, if you see yellow (Long) or red (Very Long) spikes on this chart, that is the first indication that there is likely a problem.

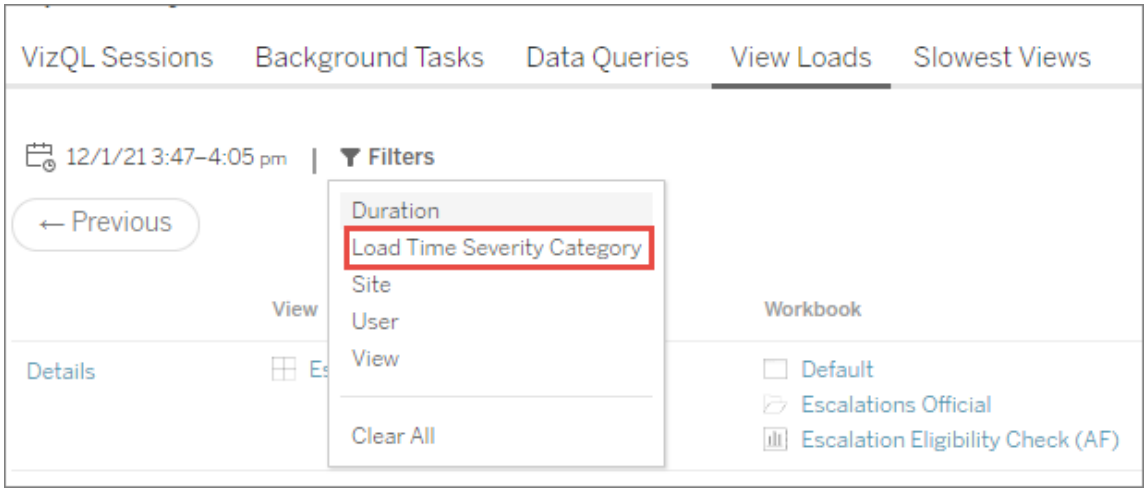
Tip: Selecting the “Past 48 hours” time range might be a good place to start as it will give you some context on activity in the past versus current.



The **Concurrent Users** and the **Total View Load Requests** charts on the same page can be used to see the scope of the impact resulting from slow view loads in the same time range. You can also use the **Tableau Processes** chart to identify any correlations between the slow view load performance and resource usage - like for example, you might see high VizQL Server resource usage on specific nodes during the same time that you are seeing spikes in the **Slow View Load Requests** chart.

If you see a spike in the slow view load request chart, you can then start to drill down further to identify what might be causing the issue - whether the problem is due to a single view or a much broader issue. To do this, within the **Slow View Load Requests** chart, select a range to include a large portion of the slow views. This will take you to the **View Loads** activity page, showing you view load requests for the same time frame.

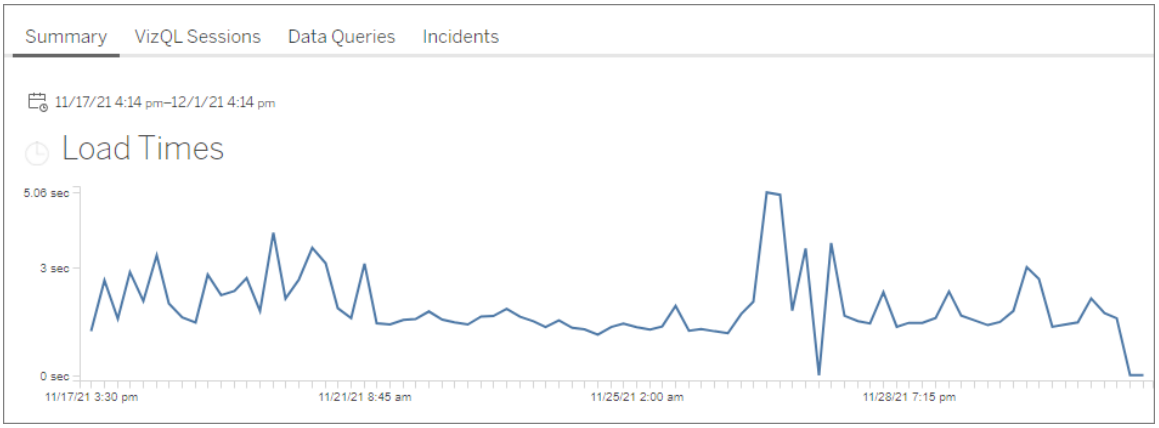
Filter by **Load Time Severity Category** and select the appropriate category for the view loads you want to investigate.



The list should indicate which views might have caused the chart results and can be understood in the following ways:

Multiple views: If the list shows multiple views, this is likely a broader issue with Tableau Server. Go back to the performance page, and take a look at the **Performance**, and **Tableau Server Processes** chart to investigate the resource usage. The **Status** tab on the performance page will show the status of the processes whether they are active, busy, or down. Look for the VizQL Server, Data Server, and the Data Engine processes on this list.

Same view: If the list is mostly one single view, this could mean that there is an issue with that view or workbook. Further investigation might be needed to see what might be causing the problem. Click on the **view name** in the list to see more information about load times, related data queries, and VizQL sessions.



Important! On the **Load Times** chart, if the average load time is consistent even when adjusting the date range to span just before and after the spike began, it means that the baseline calculation was likely created when view loads were heavily cached, causing subsequent view loads to be considered 'slow'. This scenario does not reflect an issue with Tableau Server or the view. If this is the case, you can trigger a recalculation of the baseline by publishing a new version of the workbook.

Here are some resources to help with performance troubleshooting for a specific view:

- Optimize Workbook Performance
- Record and Analyze Workbook Performance

Who can do this

Any Resource Monitoring Tool user can view the charts.

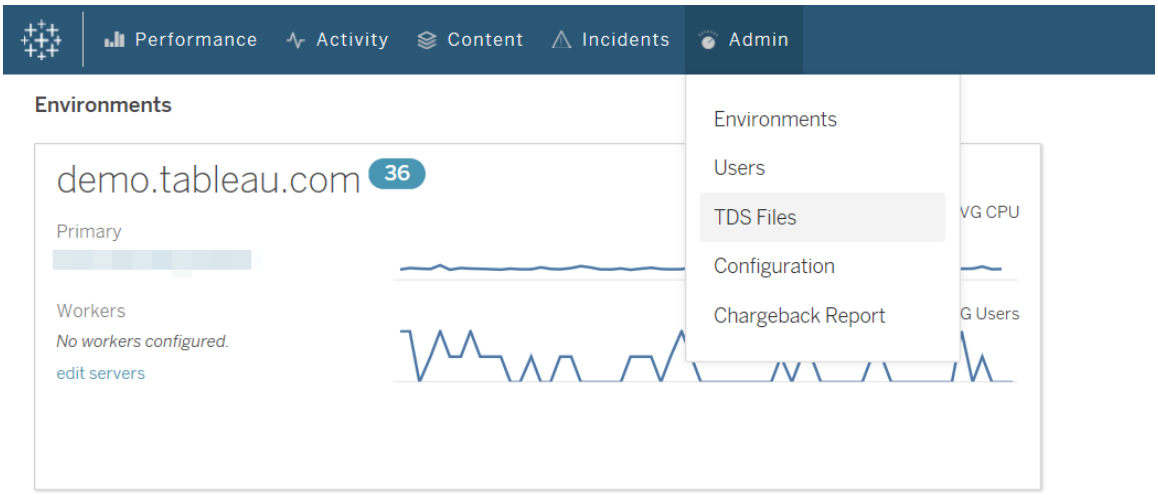
Tools used in Data Collection

Below is a list of the specific classes used to collect monitoring data :

Class	Command	Monitoring Category
`LinuxSystemCpuCollector`	top	CPU
`LinuxProcessPerformanceCollector`	top	Process
`LinuxSystemMemoryCollector`	free	Memory
`LinuxDiskQueueLengthCollector`	iostat	Disk
`LinuxDiskUsageCollector`	df	Disk

Explore Monitoring Data Using Tableau Data Source Files

The Tableau Resource Monitoring Tool includes built-in charts that you can use to monitor and analyze Tableau Server health and performance. The data Tableau Resource Monitoring Tool leverages can be downloaded as Tableau data source (.tds) files for exploration within Tableau Desktop. You can download the .tds files from the **Admin** menu using the Resource Monitoring Tool web interface. This methods works for both local and external repository configurations.



The following is a list of Tableau data source (.tds) files you can download:

- **Background Tasks**: Includes information about scheduled backgrounder tasks like extract refreshes, subscriptions, and flows.
- **Data Queries**: Information about all queries executed by Tableau Server.
- **Gateway Requests**: HTTP requests handled by Tableau Server including VizQL Server session details.
- **Incidents**: Incidents recorded by the Resource Monitoring Tool .
- **Server Performance**: Tableau Server hardware and process information that is gathered by the Resource Monitoring Tool.
- **Tableau Entities**: Information about the Tableau Server sites, projects, workbooks, and views gathered by the Resource Monitoring Tool.

Requirements

- Encryption used is SCRAM-SHA-256 which is supported by Tableau Desktop 2020.4 and later.

Enable access to the Resource Monitoring Tool PostgreSQL database

The Tableau data source (.tds) file contains a connection to the Resource Monitoring Tool PostgreSQL database. Before you can connect to a downloaded .tds file, you need to enable access for the `readonly` user to the Resource Monitoring Tool PostgreSQL database. Once the `readonly` user has access, you can then use the `readonly` username and password to

connect to the Resource Monitoring Tool PostgresSQL database from the .tds file in Tableau Desktop.

Resource Monitoring Tool versions 2022.3 and later:

Resource Monitoring Tool with local repository:

1. On the RMT Server machine, enable access to the Resource Monitoring Tool PostgreSQL database for the `readonly` user:

   ```
   rmtadmin data-access ReadOnly
   ```

2. Restart the Resource Monitoring Tool PostgreSQL database for the configuration change to take effect:

   ```
   rmtadmin restart --db
   ```

3. Retrieve the password for the `readonly` user:

   ```
   rmtadmin get db.readOnlyPassword
   ```

Resource Monitoring Tool with external repository:

You must configure the RDS instance to allow access from Tableau Desktop. Retrieve the username and password for the Resource Monitoring Tool PostgreSQL database and use it to download .tds files. For more information see the documentation on AWS site.

Resource Monitoring Tool versions 2022.2 and earlier:

1. Open the `postgresql.conf` file. By default, the file is located at: `/var/-opt/tableau/tabrmt/data/postgresql<version>`

2. Update `Listen_addresses = 'localhost'` to `Listen_addresses = '*'`.

   Note: You must remove the '`#`' from this line.

3. Open the `pg_hba.conf` file. This file is also located in the same directory as the `postgresql.conf` file. By default the file is located at: `/var/opt/tableau/tabrmt/data/postgresql<version>`

4. Add the following to the end of the `pg_hba.conf` file and then save the file:

   ```
   host all all 0.0.0.0/0 scram-sha-256

   host all all ::/0 scram-sha-256
   ```

5. Restart the Resource Monitoring Tool PostgreSQL database for the configuration changes to take effect:

   ```
   rmtadmin restart --db
   ```

6. Retrieve the password for the `readonly` user:

   ```
   rmtadmin get db.readOnlyPassword
   ```

Connect to the RMT .tds files from Tableau Desktop

After access to the Resource Monitoring Tool PostgreSQL database has been enabled for the `readonly` user, you can then connect to the downloaded .tds files from Tableau Desktop.

1. In Tableau Desktop, go to **File > Open** and select the .tds file downloaded from the Resource Monitoring Tool web interface.

   > **Note**: You might need to install the PostgreSQL database drivers. You can download drivers from www.tableau.com/support/drivers.

   If Tableau Desktop does not automatically connect to the Resource Monitoring Tool Postgres database after opening the .tds file, you might need to manually enter the `readonly` username and password into the edit connection window in Tableau Desktop.

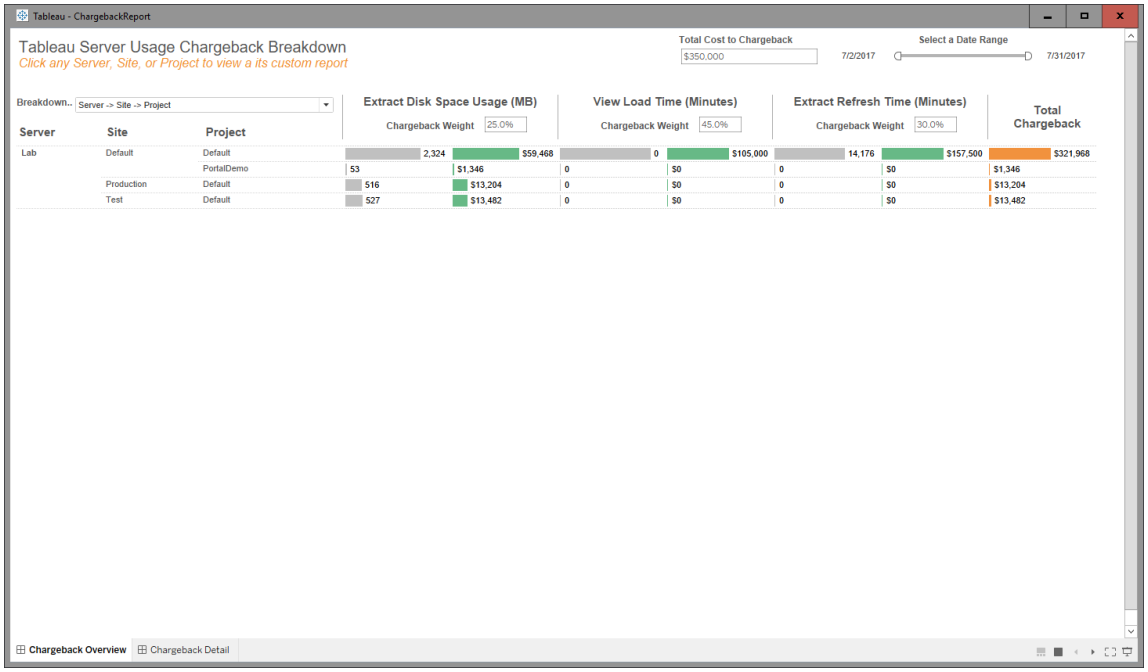
Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Download TDS Files** server role.

Chargeback Reports

Chargeback reports show usage on a per-project or per-site basis and are available for allowed users through the **Admin** menu. The report is generated as a Tableau workbook using a generated extract, allowing you to modify the report or reuse the extract as needed.

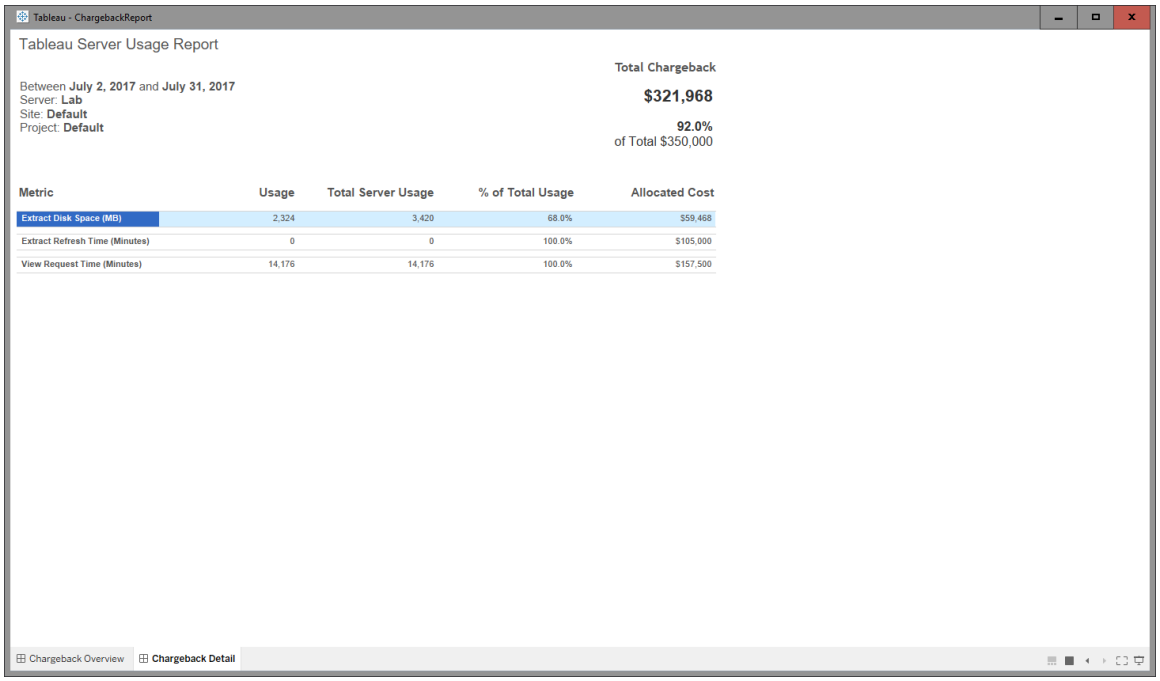
The **Chargeback Overview** worksheet shows a breakdown of various metrics based on site and/or project:



Overview

Dollar amounts can be customized using the total cost field, and by adjusting the cost weights for each metric.

Clicking on a project or site navigates to a customized invoice-style detail report for that project/site:



Detail

Security

User access is controlled through the **Generate Chargeback Reports** server role.

Data Generation

Data for the chargeback report is aggregated at the daily level, with the following metrics included in the data set:

Metric	Description
Extract File Usage	The size, in Kilobytes, of extract files is collected once per day. For the default report this is simply totaled across the time period for ratio/comparison purposes between projects/sites.
Extract Query Duration	The length of time, in milliseconds, that queries running against extract files ran, totaled per day.

Metric	Description
Extract Refresh Task Duration	The length of time, in milliseconds, that extract refresh background tasks ran, totaled per day.
Query Duration	The length of time, in milliseconds, that queries ran, totaled per day. This includes both extract and non-extract queries.
View Request Duration	The length of time, in milliseconds, that requests to render views ran, totaled per day. This includes the duration of any data queries that blocked view rendering.

Who can do this

Resource Monitoring Tool Administrator or a Resource Monitoring Tool user with **Server-/Environment Management** role.

Troubleshoot Tableau Resource Monitoring Tool Issues

This section includes articles that describe troubleshooting steps and tips. If you have any questions or encounter other issues not described here, contact Tableau customer support.

Troubleshoot Missing Hardware Performance Data

Processor (CPU) usage, memory usage, disk queue, and network performance stats are considered hardware performance data in the Tableau Resource Monitoring Tool. The most common places to see this data are:

- On the **Environment Overview** dashboard in the Performance and Tableau Processes charts
- On the **Servers** dashboard

This data is reported in near-real-time by the Resource Monitoring Tool agent processes running on each of your Tableau Server machines.

If these charts show no data for an extended period of time, this may be due to the following reasons:

1. Tableau Server is not licensed correctly. This may be due to missing Advanced Management capabilities. Make sure Tableau Server has the required license. For more information on Advanced Management, see About Tableau Advanced Management on Tableau Server.
2. Connectivity issues between Agent and Tableau Server. Use the steps below to troubleshoot and isolate these issues.

Step 1: Check the Agent connection status

First, check that the Agents are currently connected to the Resource Monitoring Tool. The Agents send a regular heartbeat message to the RMT Server to indicate their connection status.

1. Log in to the Resource Monitoring Tool as an administrator.
2. Navigate to the **Admin** > **Environments** page.
3. Click the Edit link for the environment that is missing performance data.
4. Locate the **Servers** list and ensure that each server shows the Agent Service as **Connected**. You can hover over the **Connected** status for a timestamp of when the last heartbeat message was received.

Step 2: Ensure the Agent is running

If the Agent shows as **Disconnected** in the Resource Monitoring Tool, then the Agent's Windows service may not be running.

1. Connect to the machine the agent is running on.
2. Ensure that the Tableau Resource Monitoring Tool Agent Windows service is running.

Step 3: Ensure the Agent is configured correctly

If the agent shows as **Disconnected** in the Resource Monitoring Tool, but the service is running, then the agent may not be able to reach the Resource Monitoring Tool's message queue.

1. Connect to the machine the agent is running on.
2. Navigate to the agent's installation folder. For example: `/opt/tableau/tabrmt/agent`
3. Run the *rmtadmin status* command.

The *rmtadmin status* command will test the agent's connectivity to the message queue and the Resource Monitoring Tool's RMT Server.

Step 4: Restart the Agent

In some cases, the Agent may be running and all status indicators show success but the agent continues to not send hardware performance data. This is a known issue that can result from transient connectivity errors between the Agent and the message queue. For example, when restarting the message queue server or during brief network interruptions.

To ensure this isn't the case, restart the Agent and wait a few minutes to confirm whether the performance data is working again.

1. Connect to the machine the Agent is running on.
2. Restart the Tableau Resource Monitoring Tool Agent Windows service.
3. Ensure the service starts successfully.
4. Wait at least 10 minutes and then log in to Resource Monitoring Tool and check Servers dashboard to see if any hardware performance data has been received.

Step 5: Verify Run As account configuration

Ensure the run-as accounts for the agents are configured to use the same run-as account that Tableau Server uses to connect to Tableau Server and get the performance and CPU data. It must be the same account that Tableau Server is configured with. Make sure this account has permissions to access Tableau Server logs.

Step 6: Contact Support

If, after following the above steps the issue is still not resolved, please contact support.

The support team will need a copy of the Resource Monitoring Tool log files from the RMT Server and from each of the agents that are having connection issues. For more information on how to collect log files and sending them to Tableau customer support, see Sending Log Files to Tableau Customer Support.

Who can do this

Resource Monitoring Tool Administrators.

Troubleshoot RMT Server Service Interruptions

You may be experiencing service interruptions due to RMT Server stopping on a regular basis. This is most likely because Resource Monitoring Tool has been configured with service accounts that do not conform to your internal IT policies. Many IT departments employ automation tools that will revoke privileges on accounts that are deemed non-compliant with their standards, and are the cause for the service interruptions.

Temporary solution: Restart RMT Server by running `rmtadmin start -master`.

Long term solution: You can either configure the Resource Monitoring Tool to run under an account that is compliant with your internal governance policies (recommended), or alternatively, work with your IT department to obtain an exception from the policy.

Who can do this

You need to be a Administrator on the machine and be a Resource Monitoring Tool Administrator in order to make configuration and database changes.

Troubleshoot Unknown Status of Tableau Server Processes

Under certain conditions you might see that the status of a Tableau Server process as reported as **Unknown**. This is often due to a change in the process configuration in Tableau Server which has not yet been updated in Tableau Resource Monitoring Tool.

> **Note:** After updating any configuration, it may take a few minutes for the status reported by the Resource Monitoring Tool to refresh. By default, the Resource Monitoring Tool checks the Tableau Server status every 15 seconds but it may be longer if you configured it to check less frequently.

Use the following troubleshooting steps to resolve this issue:

Step 1: Check Tableau Server Environment Settings

The Resource Monitoring Tool connects to Tableau Server for monitoring and data collection. If the Tableau Server is upgraded, or the Tableau Server credentials expire then Resource Monitoring Tool will not be able to monitor it as expected.

To confirm that theResource Monitoring Tool is able to connect to your Tableau Server:

1. Go to **Admin** > **Environments list** page.
2. Edit the environment that is having the issue.
3. Confirm that the selected Tableau Server version is correct. When you upgrade your Tableau Server you may need to update the version in the Resource Monitoring Tool.
4. Test the Tableau Server REST API connection using the “Test Connection” button.

   Optionally, test the Tableau Repository connection using the “Test Connection” button.

Step 2: Update Tableau Server Machines and Processes

To correctly monitor a Tableau Server, the Resource Monitoring Tool needs to be configured with a complete record of your Tableau Server machine names and process ports. This information is automatically gathered when creating a new environment but may need manually updated if the Tableau Server configuration is changed.

1. Go to the **Admin** > **Environments** page.
2. Edit the environment that is having the issue.
3. Confirm that the list of **Servers** contains your initial node/gateway Tableau Server machine as well as all additional node machines.
4. Follow the steps below to confirm that each server is configured correctly.

Step 3: Update Machine Name

1. Open this Tableau Server URL: *http://<your TableauServer URL>/admin/systeminfo.xml*.
2. Compare and update the Resource Monitoring Tool configuration with this page to ensure it is up-to-date:
   - Update the server’s Host Name to exactly match the <machine name="{HOST NAME}" /> attribute value.

Step 4: Contact Support

If, after following the above steps the issue is still not resolved, contact Tableau customer support.

VizQL Session details page says the VizQL process is unknown

HTTP requests for a Tableau view are linked to the VizQL process (the actual VizQL PID) that last locked the request's VizQL session ID.

In some cases we may not be able to find a matching VizQL process (PID) and in these cases you will see a message on the view session details page that the VizQL process is unknown.

This can happen in a few rare cases:

- The Tableau cluster has been modified with the addition of a new VizQL worker instance.
- Tableau adjusting the VizQL process's port number to avoid a conflict with another process.

If this happens, sign in to the Resource Monitoring Tool and navigate to the environment admin screen. You'll want to check the process definitions for your servers. Ensure that VizQL processes are defined on the expected servers and with the correct port numbers.

Who can do this

To troubleshoot Tableau Server processes issues, you need to be both a Tableau Server Administrator and a Resource Monitoring Tool Administrator.

Troubleshoot User Authentication

When a Resource Management Tool user is unable to sign in to RMT, there are a few different reasons why this might happen. Your troubleshooting steps depend on what version of RMT you are using, and what kind of authentication the user is configured for.

Troubleshoot RMT user authentication issues

When a user is unable to sign into Resource Management Tool, troubleshoot the problem by verifing the following:

- Is the user name they are entering added as a user in RMT?
- What type of authentication does the user have in RMT?
    - **Local:** If the user has local auth, reset the password and give them the new one.
    - **Delegated** (Version 2023.1.0 and later)**:**
        - If the user has delegated auth, make sure their user name is correctly entered into RMT. It should not include the domain either before or after the user name:
            - Correct: `<username>`
            - Incorrect: `<subnet.network>\<username>` or `<username@<subnet>.<network>`
        - Have the user confirm their credentials by signing into another system that uses their domain credentials.
        - Make sure the user is not locked out due to excessive failed sign in attempts (this is not something RMT handles; they'll need to work with their IT help desk on this).

Using logs to troubleshoot authentication problems

Logs for authentication issues can be found at: `\[Install Directory]\master\logs\web\tabrmt_YYYMMDD.log`.

Error strings to look for:

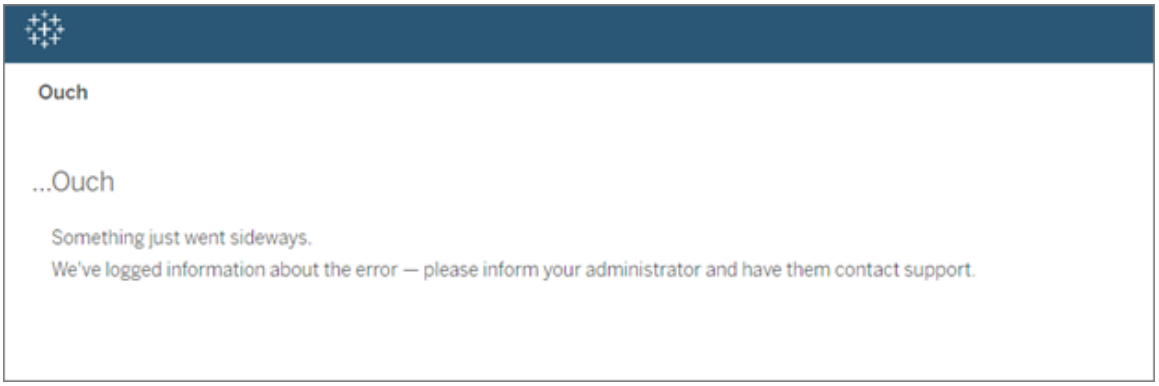
- "Invalid password for user" - A user entered the valid username, mapped to an RMT user assigned Local authentication, but provided an invalid password.
- "User failed to provide the correct password" - A user entered the correct username, mapped to an RMT user assigned delegated authentication, but provided an invalid password.
- "Executing action method \"Tableau.PowerTools.Server.Master.Web.Controllers.AuthenticationController.Login" - A sign in attempt is being made.

Who can do this

You need to be a Administrator on the machine and be a Resource Monitoring Tool Administrator in order to troubleshoot user authentication issues.

Troubleshoot Web Interface Timeouts

When you try to navigate to the Tableau Resource Monitoring Tool web interface, you see the following error:



This can happen when the load time takes longer than the default 30 seconds threshold. The load time can be affected by the following:

- The size of the underlying PostgreSQL database.
- If RMT is connected to Tableau Server environments that push the current scale that RMT can support, this may also have an affect on the load times.
- Port conflict. Make sure the port Resource Monitoring Tool is suing is not in conflict with the ports that Tableau Server is using.

To resolve this issue you can either change the data retention time period or change the threshold to greater than 30 seconds.

To change the data retention time period, use the following steps:

1. Navigate to the Configuration UI as described in this topic.
2. Navigate to the Data tab.

   There are two types of Data Configurations:

   - Data retention that is set to 2 weeks by default. It includes detailed activity data and the data displayed in the charts.

- Reporting data that is set to 10 year time period by default. It includes aggregated historical data and data used for chargeback reports.

3. Tableau recommends changing the data retention time period to 1 week.

To change the timeout threshold, use the following steps:

1. Log into the machine where the RMT Server is installed.
2. Find the "master" configuration file. The file when installed in the default location can be found at:

   ```
   /var/opt/tableau/tabrmt/master/config.json
   ```

3. Add or change the key: **db:timeoutSeconds**

If you need to reduce the size of the PostgreSQL database immediately, Use the rmtadmin command line utility to delete the data. For more information, see rmtadmin Command Line Utility .

Who can do this

You need to be a Administrator on the machine and be a Resource Monitoring Tool Administrator in order to make configuration and database changes.

Troubleshoot Messaging Tables Disk Usage Warnings

In rare situations you may see a disk space usage warning at the command line after running the `rmtadmin status` command:

```
Internal RMT messaging tables total disk usage is now: <nn.nn> KB.
```

A message displays when disk space reaches or exceeds the warning threshold of 4 GB.

The database usually cleans up unnecessary tables automatically, but if this warning displays, take the following steps to clean up the tables manually:

1. Stop RMT Server:

   ```
   rmtadmin stop
   ```

2. Start the database service:

   ```
   rmtadmin start --db
   ```

3. Clean up the unnecessary tables:

   ```
   rmtadmin query "DROP SCHEMA hangfire CASCADE;" --outfile=drop_
   hangfire_schema_output.zip --commit
   ```

4. Start RMT Server:

   ```
   rmtadmin start
   ```

Who can do this

You need to be a Resource Monitoring Tool Administrator in order to run the necessary commands.

Upgrade Power Tools for Server to Tableau Resource Monitoring Tool

Power Tools Server has been rebranded to Tableau Resource Monitoring Tool. The installer has been changed to reflect this rebranding and new ownership. This means the product will install side-by-side with the legacy InterWorks product instead of upgrading it in place.

The 2019.3 version of Power Tools Server (now named the Tableau Resource Monitoring Tool) requires a license key. For information about activating the license key, see Tableau Resource Monitoring Tool Legacy License Key Activation.

Follow these steps to upgrade to Resource Monitoring ToolThe recommended steps to upgrade are:

1. Stop all installed InterWorks agents.
2. Wait for the data processing to complete.
3. Stop and uninstall InterWorks master server.
4. Run the Tableau RMT Server installer to install Tableau RMT server.

5. Stop Tableau RMT server.
6. Copy over *config/config.json* from InterWorks RMT Server directory to Tableau RMT Server directory.
7. If you are using Postgres as your database, the following steps are necessary for the database to function with the security improvements in 2019.3. If using any other database, skip to step 8.

   Postgres SQL Update Instructions:

   - Locate the *postgres pg_hba.config* file. It will be at <*PTS installation directory*>*data**postgresql**pg_hba.config*.
   - Edit the *pg_hba.config* file, entries at the end will look like this: host all all 127.0.0.1/32 trust. Change the last column for all lines from **trust** to **md5**. So the new line should look like this: host all all 127.0.0.1/32 md5.
   - Save the *pg_hba.config* file.
   - Restart the PTS postgresql service.
8. Restart Tableau RMT Server.
9. Run the Tableau Agent installers to install Tableau agents.
10. Stop Tableau Agents.
11. Copy over *config/config.json* from InterWorks agent directory to Tableau Agent directory.
12. Once the Tableau Resource Monitoring Tool installation is complete, and you have confirmed it is working as expected, uninstall the InterWorks/Tableau Power Tools for Server.

Tableau Resource Monitoring Tool Legacy License Key Activation

The 2019.3 version of Power Tools Server (now named the Tableau Resource Monitoring Tool) requires a license key. The instructions below provide the steps to activate this key:

1. The license key will be passed over as a file in format *.dat*.
2. The .dat file needs to be added to the config directory of the Master Server.

   The default location is: C:\Program Files\Tableau\Tableau Resource Monitoring Tool\-master\config.
3. Restart the Master Server service:

   Navigate to **Windows Server Manager** and restart the **Tableau Resource Monitoring Tool** service.

About Tableau Content Migration Tool

This set of articles guides you through setting up, using, and maintaining the Tableau Content Migration Tool.

What is Content Migration Tool?

The Content Migration Tool provides an easy way to copy or migrate content between Tableau Server sites. You can do this between sites on a single Tableau Server installation, or if you have user-based licensing, between sites on separate installations (for example, between a development instance of Tableau Server and a production installation). The Content Migration Tool user interface walks you through the steps necessary to build a "migration plan" that you can use once or as a template for multiple migrations.

Before migrating content, we recommend reviewing the Content Governance section in Tableau Blueprint.

Help and Support

If you have problems that you cannot solve with this documentation, contact Tableau Technical Support.

Getting Started with Tableau Content Migration Tool

This article will help you get started with the Tableau Content Migration Tool. It contains links to other articles about information you need to prepare before installing the Content Migration Tool, and steps to design a migration plan and upgrade existing installations.

Pre-installation

Installation requirements

The Content Migration Tool can only be installed on Windows operating systems. Before installing, you must be able to connect to the Tableau source site (the site you are migrating from) and the destination site (the site you are migrating to) from the computer where Content Migration Tool is installed. Both the source and destination sites must have a valid Advanced Management license. For more information about installing and upgrading Content Migration Tool, see Install Tableau Content Migration Tool.

Compatibility with Tableau Server

The Content Migration Tool supports content migration for Tableau Server versions 2019.3 and later.

The table lists compatible versions of Tableau Server based on the installed version of Content Migration Tool.

CMT Version	Tableau Server Version
2024.3x	2023.1x - 2024.2x
2024.2x	2022.3x - 2024.2x
2024.1x	2022.1x - 2024.1x

CMT Version	Tableau Server Version
2023.1.x	2021.2.x - 2023.1.x
2022.4.x	2021.1.x - 2022.4.x
2022.3.x	2020.4.x - 2022.3.x
2022.2.x	2020.3.x - 2022.2.x
2022.1.x	2020.2.x - 2022.1.x
2021.4.x	2020.1.x - 2021.4.x
2021.3.x	2019.4.x - 2021.3.x
2021.2.x	2019.3.x - 2021.2.x
2021.1.x	2019.3.x - 2021.1.x
2020.4.x	2019.3.x - 2020.4.x
2020.3.x	2019.3.x - 2020.3.x

Compatibility with Tableau Cloud

Content Migration Tool version 2022.2.1 and later support content migration for all Tableau Cloud deployments. We recommend installing the most recent version from the Tableau Advanced Management downloads page to take advantage of the latest features and fixes.

Compatibility with Tableau content

The Content Migration Tool supports migrating workbooks and published data sources saved in the eight most recent versions of Tableau. While you can migrate existing data sources, only data sources that use the connection types in the table below can be changed and modified during migration. For more information, see Data Source Transformations in Migration Plans: Workbooks and Migration Plans: Published Data Sources.

Actian Matrix Google Drive Pivotal Greenplum Database

Actian Vectorwise	HortonWorks Hadoop Hive	PostgreSQL
Amazon Athena	HP Vertica	Progress OpenEdge
Amazon Aurora	IBM DB2	Salesforce
Amazon EMR	IBM Netezza	SAP HANA
Amazon Redshift	Map R Hadoop Hive	SAP Sybase ASE
Apache Drill	Microsoft Access	SAP Sybase IQ
Aster Database	Microsoft Analysis Services	Snowflake
Box	Microsoft Excel	Spark SQL
Cloudera Hadoop	Microsoft Excel Direct	Statistical File
Delimited Text File	Microsoft OneDrive	Tableau Extract
EXASOL	Microsoft SQL Server	Tableau Published Data Source
Firebird	MySQL	Teradata
Google Analytics	OData	Text File
Google BigQuery	Oracle	Web Data Connector
Google Cloud SQL	Oracle Essbase	Other Databases (ODBC)

Post-installation

Limitations when migrating content

Before you start, make sure you understand the limitations when migrating content using the Content Migration Tool. For more information, see Migration Limitations.

Create a migration plan

The Content Migration Tool walks you through migrating content across projects on a single site, to a new site on the same Tableau Server instance, and to sites that exist on different Tableau Server instances. The plan you create can be saved and used again for future migrations. For more information, see Migration Plan Overview.

Install Tableau Content Migration Tool

Installing Tableau Content Migration Tool is straightforward and easy.

Installation requirements

The Content Migration Tool tool is run from a Windows computer and can connect to Tableau Cloud sites and Tableau Server 19.3 and later with a valid Advanced Management license. For more information about compatible versions, see Getting Started with Tableau Content Migration Tool.

The computer that you install Content Migration Tool on must meet the requirements below:

- Microsoft Windows 10 or newer (x64)
- Intel Core i3 or AMD Ryzen 3 (Dual Core)
- 4 GB memory or larger
- Can connect to the source and destination sites. Both sites must have a valid Advanced Management license to migrate content.
- 2 GB HDD or larger. The drive where the `\temp` folder resides must have enough disk space to hold a copy of all content being migrated in a single migration. All content is stored locally on the disk and deleted when the migration is complete.
- Have enough free disk space to hold the application and its logs.

In addition, confirm that the REST API is enabled on Tableau Server (this is the default). Use the `tsm configuration get -k api.server.enabled` command to confirm this. A return value of `true` means the REST API is enabled. To enable the REST API, use the `tsm configuration set` command. For more information, see api.server.enabled.

Install Content Migration Tool

To install the Content Migration Tool:

1. Download the Content Migration Tool installer (`Tabcmt-64bit-<version>.exe`) for your version of Tableau Server from the Tableau Advanced Management downloads page.
2. Run the Content Migration Tool Setup program.

   Note: Running the Content Migration Tool Setup program overwrites the previous version.

3. After reading the EULA, select **I agree to the license terms and conditions**, and click **Install**.
4. If the User Account Control dialog opens, click **Yes** to allow the installer to make changes.

Upgrade Content Migration Tool

Upgrading to the latest version of Content Migration Tool ensures that you can take advantage of the latest features and fixes included with each new version.

Important:

- Running the Content Migration Tool Setup program overwrites the previous version.
- Content Migration Tool doesn't support side-by-side installation of previous versions.

To upgrade Content Migration Tool:

1. Log on to the machine where Content Migration Tool is installed. If there are instances of Content Migration Tool open, save your migration plan and exit the application.
2. Follow the steps listed in Install Content Migration Tool to download the latest installer and complete the upgrade.

Install Content Migration Tool from the command line

You can install Content Migration Tool from the command line if you're a local administrator on the machine.

Install switches

Specify one or more switches in the command line for the installer. For example:

```
Tabcmt-64bit-2022-3-0.exe /quiet /norestart
```

Switch	Description	Comments
`/install` | `/repair` | `/uninstall` | `/layout` "<directory>"	Run Setup to either install, repair, or uninstall Content Migration Tool, or with `/layout`, create a complete local copy of the installation bundle in the directory specified.	Default is to install, displaying UI and all prompts. If no directory is specified on a fresh install, `C:\Program Files\Tableau\Tableau Content Migration Tool` is assumed. If Content Migration Tool is already installed, Setup assumes the same location as the current installation.
`/passive`	Run Setup with minimal UI and no prompts.	Content Migration Tool doesn't start automatically when installed in `/passive` mode. To start Content Migration Tool, open the application manually.
`/quiet` | `/silent`	Run Setup in unattended, fully silent mode. No UI or prompts are displayed.	Content Migration Tool doesn't start automatically when installed in `/silent` or `/quiet` mode. To start Content Migration Tool, open the application manually. **Note:** Use either `/silent` or `/quiet`, not both.
`/norestart`	Run Setup without restarting Windows, even if a restart is necessary.	**Note:** In certain rare cases, a restart cannot be suppressed, even when this option is used. This is most likely when an earlier system restart was skipped, for example, during installation of other

		software.

Who can do this

A user with Administrator access on the machine.

Using Tableau Content Migration Tool

The following steps are designed to guide you through using the Tableau Content Migration Tool:

- Migration Plan Overview
  - Migration Plans: Sites
  - Migration Plans: Source Projects
  - Migration Plans: Workbooks
  - Migration Plans: Published Data Sources
  - Migration Plans: Permissions and Ownership
  - Migration Plans: Migration Scripts
  - Migration Plans: Plan Options
- Using the Tableau Content Migration Tool Console Runner

Tableau Content Migration Tool Use Cases

Tableau Content Migration Tool as the name suggests, is primarily used for moving Tableau Server content from one site to another. However, there are many features in the tool that makes it ideal for accomplishing several tasks related to content migration and maintenance.

Note: In many of the use cases we use the term migration to describe moving content from one environment, site, or project to another. However, technically the Content Migration Tool copies content and does not automatically delete or archive the original or source content.

The information below describes some common use cases where you can leverage the Content Migration Tool.

Content promotion

You can use the Content Migration Tool to create content for development sites and then perform routine migrations to promote content to staging or production environments.

Use the following steps to promote content to production environments:

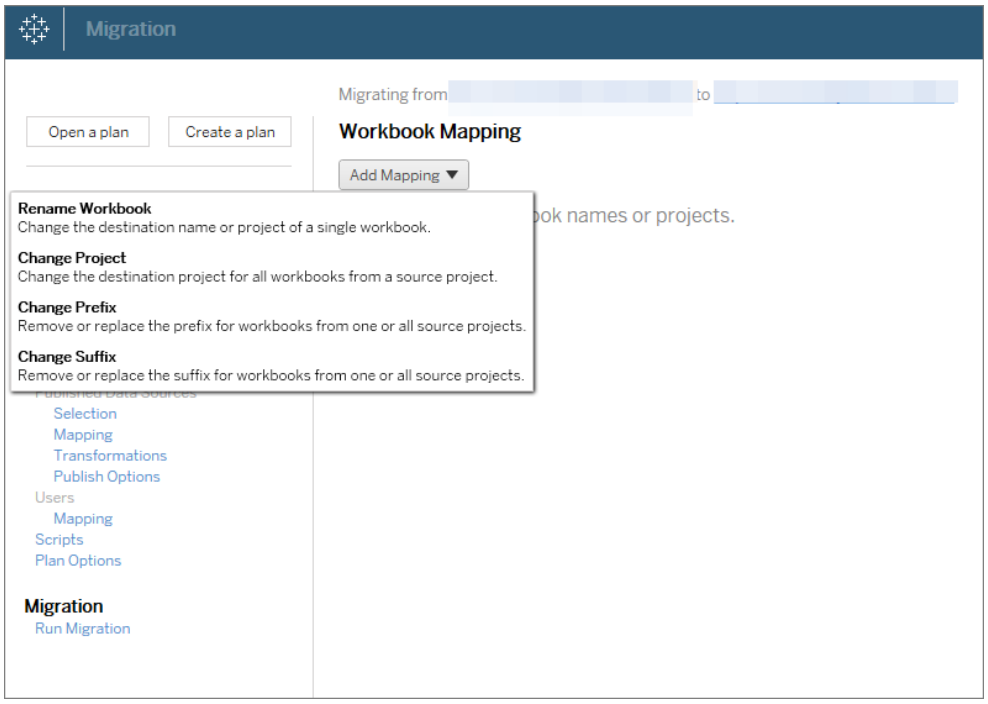
1. **Create a plan** and select the site used for development as the source and the staging or production site as the destination. For more information, see Create a Plan in Migration Plan Overview topic.

   When migrating your workbooks between two projects on the same site, your sign-in credentials for the source and destination may be similar or identical. In this scenario, we recommend using personal access tokens for a more reusable connection. For more information, see Personal Access Tokens.

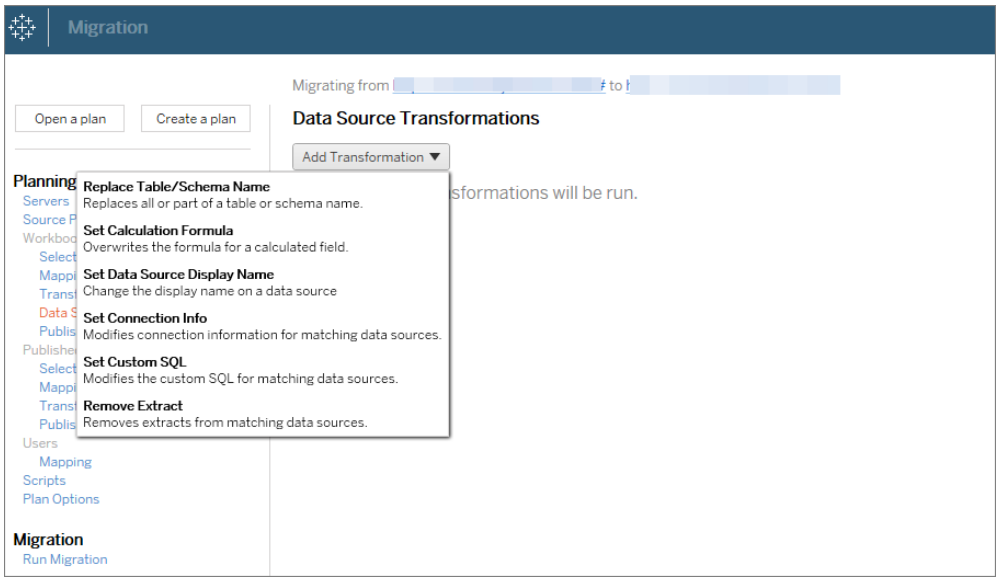
2. **Select the content** you want to migrate from your source site. You can select entire projects, specific workbooks and data sources, and user permissions. For more information, see Planning in Migration Plan Overview topic.

3. If you need to make any **changes or transformations** to the content during this migration, you can configure that in the plan as well. This is referred to as **Mapping**. The types of mapping you can make include:

   - **Changes to workbooks:** Includes renaming workbooks and changing the destination project. For a full list of workbook transformations, see Migration Plans: Workbooks.



- **Changes to data sources:** Includes replacing table or schema names, settings calculation formulas, and setting connection information. For a full list of data source transformations, see Migration Plans: Workbooks (embedded data sources) and Migration Plans: Published Data Sources (published data sources).



- **Changes to Users:** Includes domain, user, and group name changes in the destination.

4. When you are ready, click **Run Migration** to end the Planning phase and prepare to run your plan.

5. **To schedule** this to run regularly, you can script this as a job using the Content Migration Tool Runner and schedule it. For more information on using the Content Migration Tool Runner, see Using the Tableau Content Migration Tool Console Runner.

Tailoring content for customers

When working in a consulting scenario, you can customize content for each of your customers using the Content Migration Tool. Each workbook functions as a template for your migration plan, allowing you to apply styling (text, images, etc.) and replace data sources for specific customers.

Use the following steps to customize content for your customers:

1. **Create a plan** and select the production site as the source and the customer site as the destination. For more information, see Create a Plan in Migration Plan Overview topic.

2. **In the Workbooks** step of the Migration Plan, use workbook mappings and transformations to customize your content. Below are two examples of frequently used transformations. For a full list of workbook transformations, see Migration Plans: Workbooks.

    - To personalize content, you can use the **Replace Image** and **Replace Text** transformations to update the workbook with a customer's company name and logo.

    - When it comes to data sources, you can use the **Replace Table/Schema Name** or **Set Custom SQL** transformations to modify content for your customer.

3. **Verify and run** the plan. When you are ready, click **Run Migration** to end the Planning phase and prepare to run your plan.

Environment migration

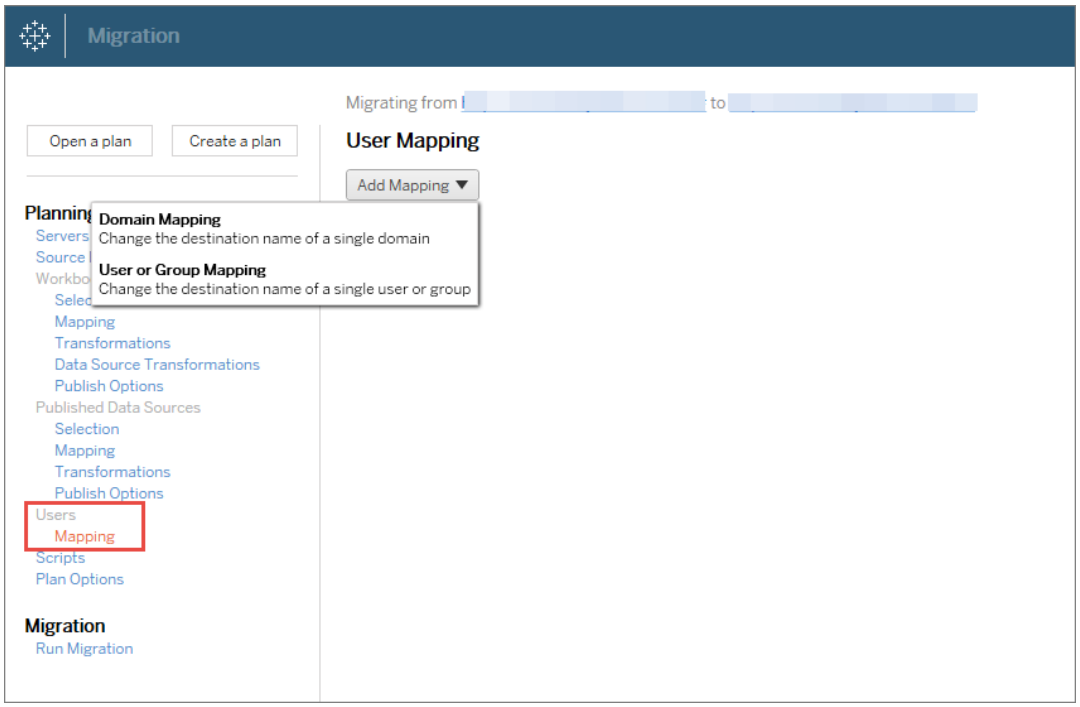
You can use the Content Migration Tool to migrate content between Tableau Server environments.

Use the following steps to migrate content between Tableau deployments:

1. **Create a plan** and **select the site** you want to migrate from as your source. For more information, see Create a Plan in Migration Plan Overview topic.

2. **Select the content** you want to migrate from your source site. You can select entire projects, specific workbooks and data sources, and user permissions.

3. **Create user permissions mappings** to customize and secure content. For more information, see Migration Plans: Permissions and Ownership.



4. **Verify and run** the plan. When you are ready, click **Run Migration** to end the Planning phase and prepare to run your plan.

Tips

- Before you perform an environment migration, make sure you understand the Migration Limitations when using the Content Migration Tool.
- You can migrate your content in stages, test and validate content iteratively before final migration is complete. There is no server downtime when you use this method of migration. It can be a replacement for Site import/export.
- Content Migration Tool migration does not handle embedded credentials, subscriptions, and custom views. These will have to be migrated manually.

External content sharing

You can use the Content Migration Tool to share internal content with external collaborators, without allowing access to your site. This keeps your data secure and allows you to publish only select workbooks and data sources. Once content has been shared, collaborators sign in

to their Tableau Server site to view and make changes, without affecting content stored on your internal server.

Before you continue, make sure the content you share is compatible between the internal (source) and external (destination) sites. The external site should be running the same version of Tableau as the internal site or later. For more information about compatibility, see Make Workbooks Compatible Between Versions in Tableau Desktop help.

Use the following steps to share content externally:

1. **Work with** the external site administrator to determine a user account that has publishing rights on the site. You will use this user account to create the migration plan. For more information, see Set Users' Site Roles and Permissions.

2. **Prepare internal content**. As a best practice, we recommend separating content on the internal server to a project, with locked permissions and strict governance rules. Workbooks and data sources should be clearly labeled to indicate the content is for external use. For more information, see Use Projects to Manage Content Access.

   > **Note:** Content shared with external sites must use data extracts unless the data source is publicly accessible. For information about creating extracts and replacing data sources, see Extract Your Data and Replace Data Sources in Tableau Desktop help.

   If you have implemented row level security, those data sources must be updated to reflect user filters and other details for the external site. For more information about row level security, see Restrict Access at the Data Row Level in Tableau Desktop help.

3. **Create a plan** and select the internal site as the source and the external site as the destination. For more information, see Create a Plan in Migration Plan Overview topic.

4. **Select the content** you want to share with the external site. You can select entire projects, specific workbooks and data sources, and user permissions.

5. **Verify and run** the plan. When you are ready, click **Run Migration** to end the Planning phase and prepare to run your plan.

Validating database migrations

This use case is when you intend to validate content after a migration of the underlying databases. One example of database migration is moving from SQL Server to Snowflake. CMT can help you validate the content built from both data sources is the same before you finalize your migration, but it cannot perform the actual database migration.

Use the following steps to validate database migrations:

1. **Create a plan** and select the Tableau site to use as your source. In this example, we describe a migration between projects on the same Tableau site, so select the same site for your destination. For more information, see Create a Plan in Migration Plan Overview topic.
2. **Configure the migration** to copy your content to a new project. Let's call the **source** project as **Project A**, and the new or the **destination** project as **Project B**.

    - Changes to workbooks: Create a workbook mapping to change Project A to Project B. For a full list of workbook transformations, see Migration Plans: Workbooks.

    - Changes to data sources: Create a data source mapping to change the Project A to Project B. For a full list of data source transformations, see Migration Plans: Published Data Sources.

3. **Verify and run** the plan. When you are ready, click **Run Migration** to end the Planning phase and prepare to run your plan.

4. **Update the content** in **Project B** with the new database connections or replace the data sources. This needs to be done manually by authoring.

5. **Test each workbook** in **Project A** with the copy in **Project B** and review for any inconsistencies in the data due to the change in data source.

6. Once you have confirmed everything is working as expected, **overwrite the content** in Project A with the updated content in Project B.

> **Note:** If the content already exists in the destination project and you do not select the **Overwrite Newer Workbooks** and **Overwrite Newer Data Sources** publish options, the content will not be copied to the destination project.

Geographical content migration

If you are maintaining a geographically distributed, multi-site environment, you will need some of this content to be shared and accessible across all the servers. This use case describes how to migrate content between servers in different geographies. The server can be in the same country or across continents.

Best Practices:
- We recommend prioritizing the content that you most need. Content Migration Tool shouldn't be used to copy entire server environments to multiple geographies.
- We recommend migrating content in one direction only meaning Primary to Secondary. Here we use the term Primary to indicate the source site and Secondary to indicate the destination. You can have one or more destination by creating multiple migration plans.

Use the following steps to migrate between Tableau Server that are distributed in different geographies:

1. **Create a plan** and select the Primary site as the source and the secondary site as the destination. For more information, see Create a Plan in Migration Plan Overview topic.

2. **Select content** that you want to share between the Primary and Secondary.

3. **Verify and run** the plan. When you are ready, click **Run Migration** to end the Planning phase and prepare to run your plan.

4. **To schedule** this to run on regularly, you can script this as a job using the Content Migration Tool Runner and schedule it. For more information on using the Content Migration Tool Runner, see, Using the Tableau Content Migration Tool Console Runner.

5. **Review the content** on the source **periodically** to determine if new items should be added to the migration plan.

Consolidate sites

If you need to combine the content of multiple sites into a single site (if, for example, organizational restructuring has changed how your sites should be arranged), you can use the Content Migration Tool to do this.

> **Note:** Before consolidating sites, make sure you understand the limitations when migrating content using the Content Migration Tool. For more information, see Migration Limitations.

Use the following steps to copy all the workbooks and data sources from one site to another:

1. Create a plan and select the site you want to consolidate as the source. For more information, see Create a Plan in Migration Plan Overview topic.

2. On the Source Projects page, select **All Projects** and click **Next**.

   If a project with the same name exists on the destination site, the content will be migrated to the same folder.

3. On the Project Options page, select your preferences for the destination site and click **Next**.

If the content already exists in the destination project and you do not select the overwrite option, the content will not be copied to the destination project.

4. On the Workbook Selection screen, select **All Workbooks**.
5. (Optional) If you are copying published data sources:
    - From the left navigation menu, under Published Data Sources, click **Selection**.
    - Select **All Data Sources**.
6. Click **Run Migration** and review the migration plan. When you're ready, click **Run** at the bottom of the screen to run the migration.

Repeat these steps until you've consolidated all sites. For more information, see Planning in Migration Plan Overview topic.

Maintenance tasks

You can use the Content Migration Tool to perform a variety of maintenance tasks.

Tagging stale content

Using the Content Migration Tool, you can manage archiving stale content. For example, you can build a plan that runs on a regular schedule that can automatically pick up content tagged as Stale Content and move it to an Archive project. After a certain amount of time, the content in this project can be purged from the system. For more information see, Migration Plans: Workbooks.

Restoring content

You can use the Content Migration Tool to restore content removed (accidentally or purposefully) from the production Tableau Server with content from a backup server. The restoration process is simple and does not require downtime, compared to restoring with a backup file.

Use the following steps to restore content from a backup server:

1. **Create a plan** and select the backup Tableau Server as the source and the production server as the destination. For more information, see Create a Plan in Migration Plan Overview topic.
2. **Select the content** you want to restore from the backup Tableau Server.
3. **Verify and run** the plan. When you are ready, click **Run Migration** to end the Planning phase and prepare to run your plan.
4. **Review the content** on the production server.

Partial backup

Once you have a backup Tableau Server environment, you can use the Content Migration Tool to transfer new content from production to the backup Tableau Server. If you have not configured a backup Tableau Server environment, see the Disaster Recovery for Tableau Server whitepaper for more information.

Notes:
- Before you perform a partial backup, make sure you understand the Migration Limitations when using the Content Migration Tool.You may still need to periodically perform a full backup and restore to backup all Tableau Server content. For more information, see Perform a Full Backup and Restore of Tableau Server.
- The Content Migration Tool shouldn't be used to perform your first backup.

Use the following steps to perform a partial backup of your content:

1. **Create a plan** and select the backup Tableau Server as the source and the production server as the destination. For more information, see Create a Plan in Migration Plan Overview topic.
2. **Select the content** you want to backup. You can select entire projects, specific workbooks and data sources, and user permissions. To migrate only new content, make sure

the publish options **Overwrite Newer Workbooks** and **Overwrite Newer Data Sources** are not selected. For more information, see Migration Plans: Workbooks.

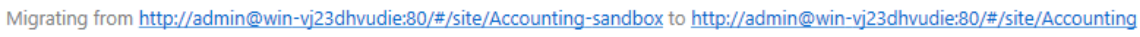
3. **Verify and run** the plan. When you are ready, click **Run Migration** to end the Planning phase and prepare to run your plan.

4. **To schedule** this to run on a regular basis, you can script this as a job using the Content Migration Tool Runner and schedule it. For more information on using the Content Migration Tool Runner, see Using the Tableau Content Migration Tool Console Runner.

Migration Plan Overview

Tableau Content Migration Tool creates a streamlined process for migrating Tableau content between sites and projects. The easy-to-follow plan can be audited, is repeatable, and works via a batch process so any number of workbooks and data sources can be migrated in a simple and efficient process.

The Content Migration Tool will display contextual tips to walk you through creating or editing a migration plan. Once you select the source and destination sites, a summary of your migration will be displayed at the top of the screen as follows:

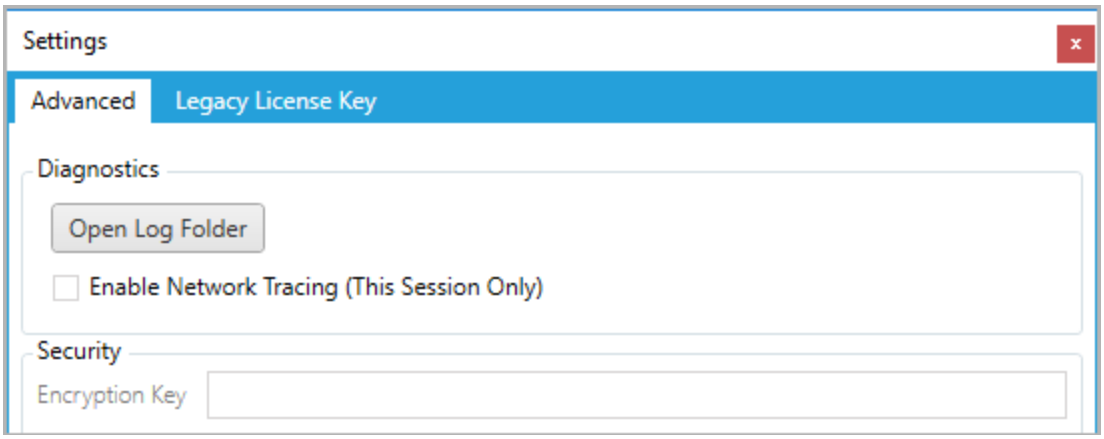


Limitations when migrating content

Before you start, make sure you understand the limitations when migrating content using the Content Migration Tool. For more information, see Migration Limitations.

Encryption keys

Each migration plan file is generated with an encryption key unique to the application that created the plan. Encryption keys can be shared if the migration plan needs to be run through an application that did not originally generate the file. When sharing encryption keys, you will need to overwrite the existing key in the application to run the migration plan. To view your encryption key, select **Help** > **Settings**.



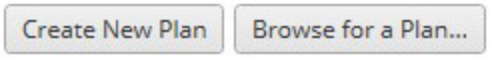
If you will be using the Content Migration Tool Console Runner for migration plans, you must specify the encryption key using the `tabcmt-runner encryption` command before running the plan. For more information, see Using the Tableau Content Migration Tool Console Runner.

Migration process

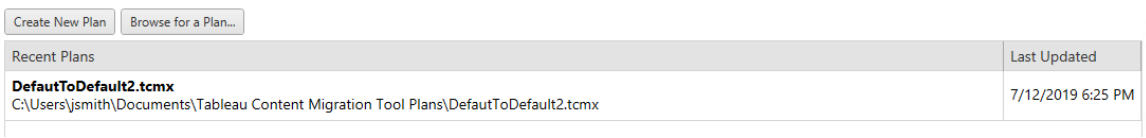
Step 1: Start

The core of the migration process is creating a plan, which you can save and re-use for future migrations or modify and update as needed. The first step is choosing whether to create a new plan, or select a previously saved plan.

To create a new plan, click **Create New Plan**. If you already created a migration plan and want to use it, click **Browse for a Plan**.



By default, all of your saved migration plans will be stored in the `Tableau Content Migration Tool Plans` folder in your My Documents folder. All migration plans are saved with a `.tcmx` extension, with recently accessed plans listed separately to make them easy to select:



You can select a recently accessed plan and duplicate it to modify the plan and save it as a new plan. Select the plan you want to copy and click **Duplicate**.

Step 2: Planning

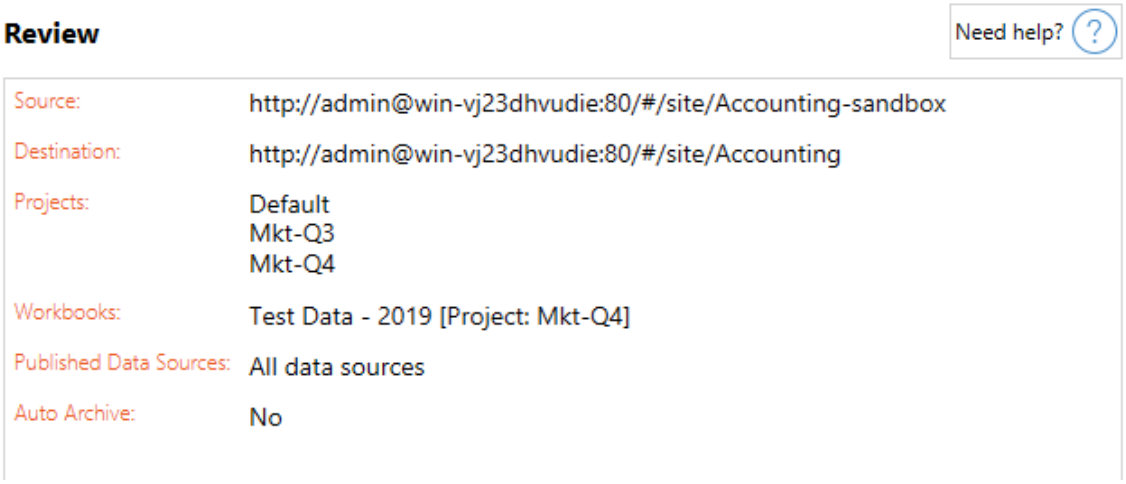
The Content Migration Tool guides you through building or editing your migration plan in six steps.

Click on each step for detailed instructions:

- Migration Plans: Sites
- Migration Plans: Source Projects
- Migration Plans: Workbooks
- Migration Plans: Published Data Sources
- Migration Plans: Permissions and Ownership
- Migration Plans: Migration Scripts
- Migration Plans: Plan Options

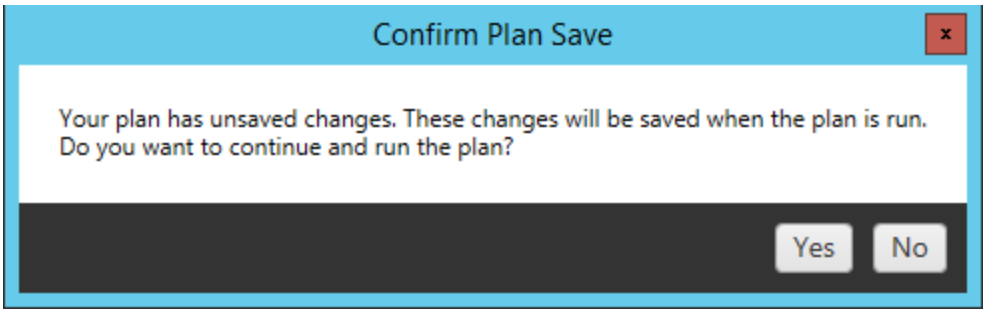
Step 3: Migration

Once you have completed your plan, you are now ready to run the batch process for migration. When you reach the final step of the migration, a plan summary displays for your verification:



If you want to change any aspects of your plan, you can click on a section in the left sidebar to go directly to that phase. When you are ready, click **Run** to begin your migration.

When you click **Run**, the migration tool will prompt you about any unsaved elements of your plan. By default, any unsaved elements will be saved when you click **Yes**. Remember you can always keep your previous plan without making any changes by duplicating it during the Start phase of the migration process.

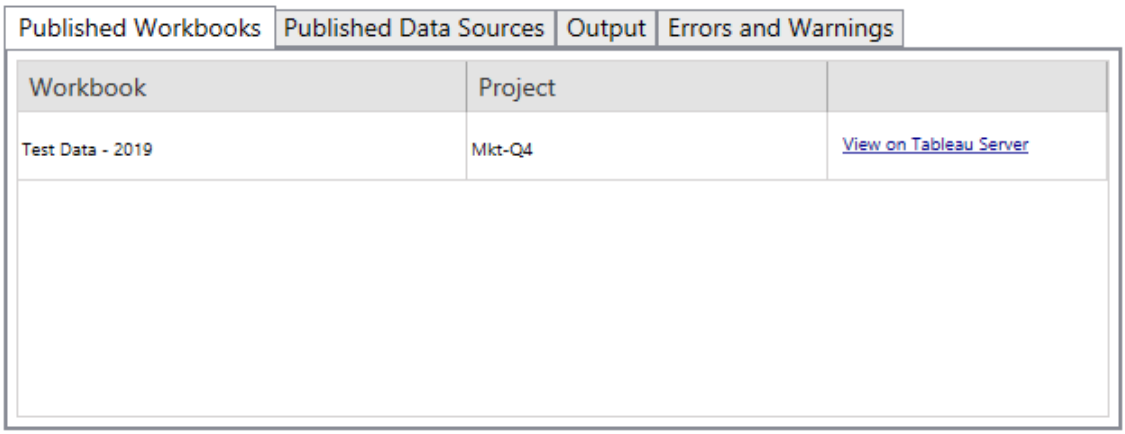


Your migration plan will run and a status bar displays for the overall plan progress and each workbook being sent to the destination server.

When the plan finishes running, you can click the tabs at the bottom of the screen for more information about the migration.

Published workbooks

Published Workbooks details the newly published workbooks and the projects where they were migrated.

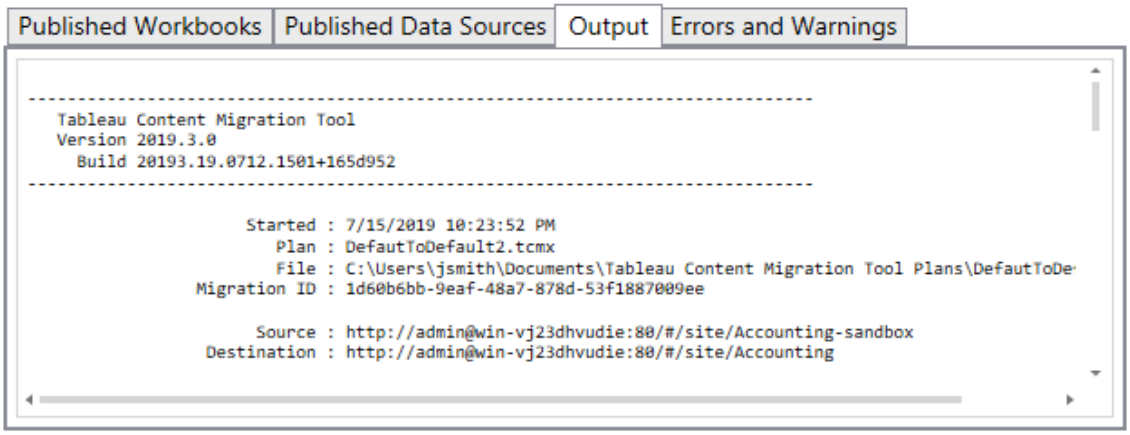


Published data sources

Published Data Sources details the newly published data sources and the projects where they were migrated.

Output

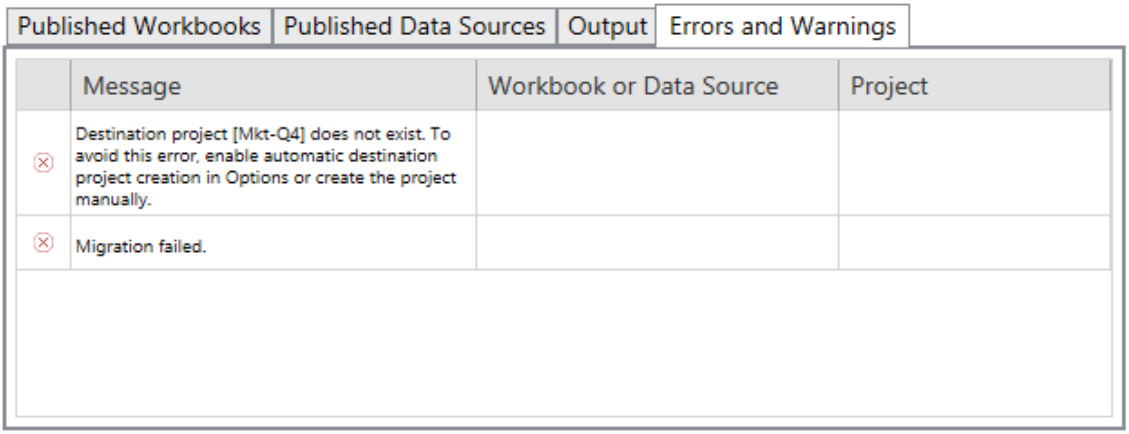
The **Output** tab details the migration log of your plan.



You can save this log by clicking **Save Log**.

Errors and warnings

The **Errors and Warnings** tab highlights any problems that occurred during the migration.



You can correct these and rerun your plan. When you have completed your migration and saved your plan, click **Done** to finish.

Who can do this

Tableau site user with an Explorer role or higher. To migrate content, you must have **View** and **Download/Save a Copy** capabilities for workbooks on the source site and **View** and **Publish** capabilities for target projects on the destination site. For more information, see Permissions.

Migration Limitations

There are certain limitations to migrations when using the Tableau Content Migration Tool. Before creating your migration plan, review the sections below to learn about version compatibility and content that will not be migrated.

Compatibility with Tableau content

The Content Migration Tool supports migrating workbooks and published data sources saved in the eight most recent versions of Tableau. Workbooks and published data sources saved before version 2018.1.x are not supported by CMT. For more information, see Getting Started with Tableau Content Migration Tool.

Configurations

The following configurations are not migrated to the destination site when using the Content Migration Tool.

- Users
- Groups
- Site settings (custom logos, view recommendations, etc.)

Data connections

While you can migrate existing data sources, only data sources that use the connection types in the table below can be changed and modified during migration. For more information, see

Data Source Transformations in Migration Plans: Workbooks and Migration Plans: Published Data Sources.

Actian Matrix	Google Drive	Pivotal Greenplum Database
Actian Vectorwise	HortonWorks Hadoop Hive	PostgreSQL
Amazon Athena	HP Vertica	Progress OpenEdge
Amazon Aurora	IBM DB2	Salesforce
Amazon EMR	IBM Netezza	SAP HANA
Amazon Redshift	Map R Hadoop Hive	SAP Sybase ASE
Apache Drill	Microsoft Access	SAP Sybase IQ
Aster Database	Microsoft Analysis Services	Snowflake
Box	Microsoft Excel	Spark SQL
Cloudera Hadoop	Microsoft Excel Direct	Statistical File
Delimited Text File	Microsoft OneDrive	Tableau Extracts
EXASOL	Microsoft SQL Server	Tableau Server Data Sources
Firebird	MySQL	Teradata
Google Analytics	OData	Text File
Google BigQuery	Oracle	Web Data Connector
Google Cloud SQL	Oracle Essbase	Other Databases (ODBC)

Unsupported content

The following content is not migrated to the destination site when using the Content Migration Tool and will require additional configuration.

Content	Action required
Ask Data lenses	Users must recreate Ask Data lenses on the destination site. For more information, see Create Lenses that Focus Ask Data for Specific Audiences.
Collections	Users must recreate collections on the destination site. For more information, see Collections in Tableau Desktop help.
Comments	Users must re-add comments to views on the destination site. For more information, see Comment on Views in Tableau Desktop help.
Custom views	Users must recreate custom views on the destination site. For more information, see Use Custom Views in Tableau Desktop help.
Data roles	Users must recreate data roles on the destination site. For more information see Use Data Roles to Validate your Data in Tableau Prep Builder help.
Data source certifications	If you have the following site roles and capabilities, you can certify data sources on the destination site. • Site Administrator Creator • Creator or Explorer (can publish) with Project Leader capability on the project containing the data source For more information, see Use Certification to Help Users Find Trusted Data.
Data-driven alerts	Users must recreate data-driven alerts for dashboards and views on the destination site. After data-driven alerts are created, anyone with access to the view can add themselves to existing alerts. For more information, see Send Data-Driven Alerts from Tableau Cloud or Tableau Server in Tableau Desktop help.
Descriptions for workbooks and	If you own the content item or have the appropriate permissions, you can edit the item's description on the destination site. For more inform-

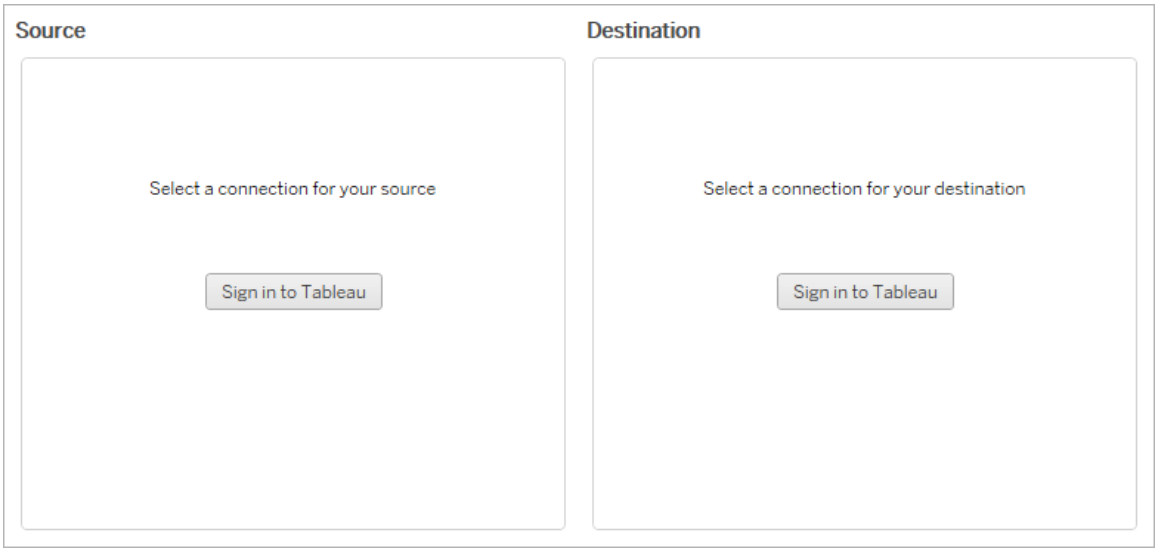
data sources	ation, see Add or edit descriptions in Tableau Desktop help.
Embedded credentials	For security purposes, Tableau Server removes embedded credentials from data sources during the download process. • To include embedded credentials when migrating from Tableau Server to Tableau Cloud, use the Migrate Embedded Credentials for Workbooks and Migrate Embedded Credentials for Data Source publish options. For more information, see Migrate Workbooks and Data Sources with Embedded Credentials. • To include embedded credentials when publishing to Tableau Server sites, use the Set Connection Info data source transformation. For more information, see Migration Plans: Published Data Sources. **Note:** CMT does not support embedded credential migration for OAuth connections. To migrate OAuth credentials, use the Set Connection Info data source transformation.
External assets	Customized attributes for external assets are not migrated to the destination site. For example, tags, certifications, data quality warnings, descriptions, permissions, user contacts, tables, and columns must be recreated. For more information, see Manage Permissions for External Assets.
Extract refresh schedules	Extract refresh schedules cannot be migrated to Tableau Cloud destination sites. To refresh data on Tableau Cloud, you can run extract refreshes manually or create new extract refresh schedules. For more information, see Schedule Refreshes on Tableau Cloud.
Favorites	Users must reselect their favorite content on the destination site. For more information, see Mark Favorites in Tableau Desktop help.
Flows	To run flows on a schedule, users must republish flows to the des-

	tination site with Tableau Prep. For more information, see Publish a Flow to Tableau Server or Tableau Cloud in Tableau Prep help.
Incremental extract refreshes	Incremental extract refreshes are changed to full extract refreshes on the destination site. Users must reconfigure incremental refreshes in Tableau Desktop and publish extracts to the destination site after migration. For more information, see Refresh Extracts in Tableau Desktop help.
Metrics	The historical values for metrics are removed from views, and users must recreate metrics on the destination site. For more information, see Create and Troubleshoot Metrics (Retired). The legacy Metrics feature was retired in February 2024 for Tableau Cloud and in Tableau Server version 2024.2. For more information, see Create and Troubleshoot Metrics (Retired).
Revision history	To migrate previous versions of workbooks to the destination site, users must download the versions they wish to keep and republish the workbook to the destination site. For more information see Work with Content Revisions in Tableau Desktop help.
Subscriptions	Users must resubscribe to views and workbooks on the destination site. For more information, see Create a Subscription to a View or Workbook.
Thumbnails for workbooks and views	Workbooks and views that are migrated using the Content Migration Tool will retain their original thumbnails, even if the migration plan includes transformations that result in the views being rendered differently (for example, if data connections change). To update thumbnails, edit the workbook or view on the destination site and re-save it. For more information, see Edit Tableau Views on the Web in Tableau Desktop help.
Virtual connections	Users must recreate virtual connections on the destination site. For more information, see Create a Virtual Connection.

Migration Plans: Sites

The first step when creating a migration plan in the Tableau Content Migration Tool is to sign in to the source and destination sites.

In the Sites section of the planning phase, you'll sign in to the source and destination sites. The permissions of the user credentials you use govern the sites and projects you see when creating a migration plan. You can only migrate content that the user has access to.



Required permissions and licenses

The user account(s) used to sign in to the source and destination sites must have an Explorer role or higher, and the following permissions for the content you want to migrate:

- View
- Download Workbook/Save a Copy
- Optional: Administrator (to select workbooks, to access a user list)

Both the source and destination sites must have Advanced Management capacities. For more information, see About Tableau Advanced Management on Tableau Server.

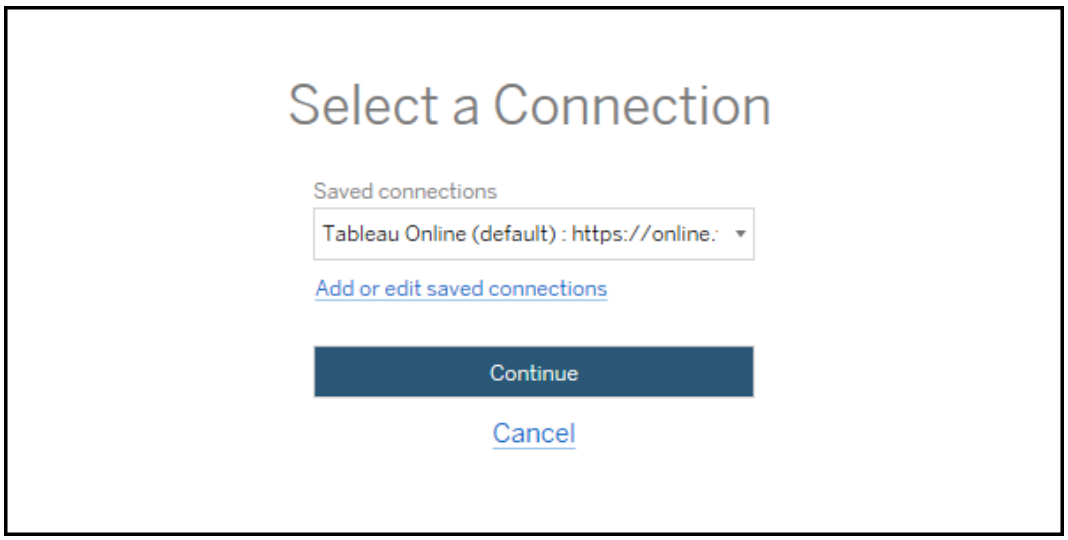
Step 1: Source

Here is the starting point of the migration. Sign in to Tableau Server , then select your *site* to use as the source. Sites are independent silos of workbooks, data, and user lists created within Tableau to group related content for selected users. You can only migrate content from one site at a time.

Sign in to the source site

Use the steps below to sign in to Tableau Cloud or Tableau Server. For more information about signing in to Tableau Cloud with Single Sign-On and Tableau with MFA, see Sign In to Tableau Cloud.

1. Click **Sign in to Tableau**.
2. In the **Select a Connection** dialog window, select a saved connection and click **Continue**.



   If no connections are available, click **Add or edit saved connections** to add a new connection. For more information, see Saved connections.

3. Enter your username and password, and click **Sign In**.

If your server is configured for SAML or Single Sign-On, you are redirected to the Identity Provider sign-in page to complete the authentication process.

4. Select the site you want to use.

To change the source server or update your site selection, click **Select a different source**.

Step 2: Destination

Repeat the sign-in process for the destination site (the site you are migrating content to).

If you are migrating your workbooks between two projects on the same Tableau site, your sign-in credentials for the source and destination site will be identical (including the server URL and site name).

Saved connections

Using saved connections allow you to quickly sign in to the source and destination sites by creating a reusable connection. When adding a saved connection, you must specify the preferred sign-in method for your site.

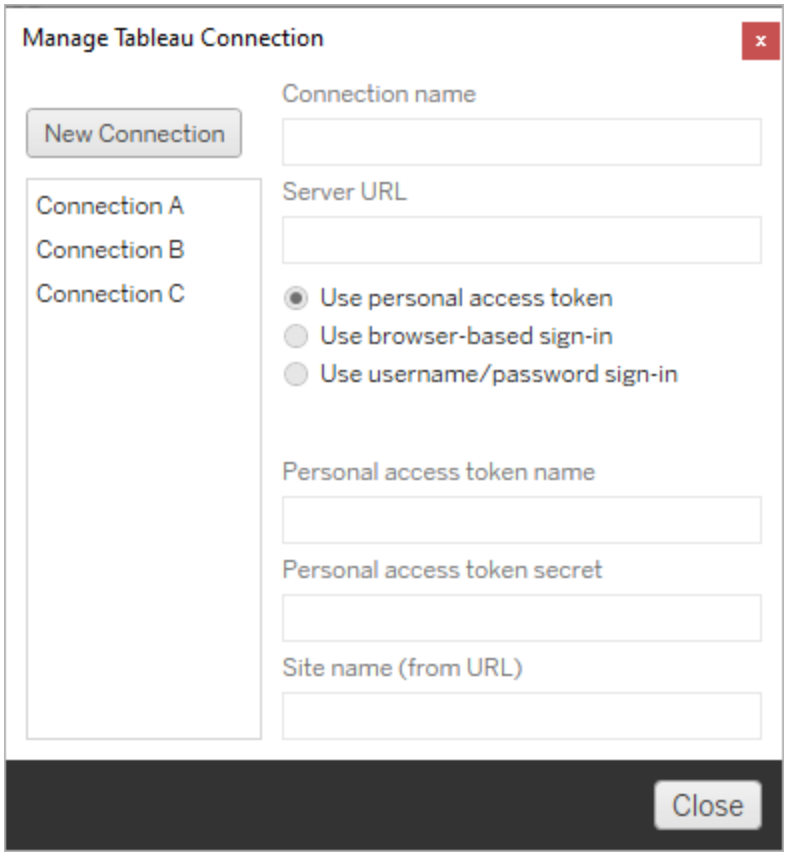
As of 2021.2, the Content Migration Tool supports the following sign-in methods:

- **Personal access tokens:** Allows users to create long-lived authentication tokens for improved security, auditing, and automation of migration plans. Personal access tokens let users sign in without requiring interactive login in the Content Migration Tool. For more information, see Personal Access Tokens.

- **Browser-based sign-in:** Users enter their credentials and complete authentication through an embedded web browser. This option may be similar to how you usually authenticate to Tableau.
- **Username and password sign-in:** Users authenticate through the Content Migration Tool instead of an embedded browser window. This option passes credentials to the server using Tableau REST APIs. You can use username and password sign-in to authenticate to migration plans created before version 2020.3 and when troubleshooting issues that prevent the use of browser-based sign-in.

Add or edit saved connections

A link to **Add or edit saved connections** is displayed at the bottom of the Content Migration Tool, and when signing in to the source and destination sites. Clicking this link will open the **Manage Tableau Connections** window.



Use the steps below to add a saved connection:

1. On the **Manage Tableau Connections** window, click **New Connection**, or select an existing connection to make changes.
2. Enter a **Connection Name** (name to describe your server) and the **Server URL**.

   If you don't include a prefix for the Server URL, the Content Migration Tool will use `http://`.
3. Select the sign-in method for your connection.

   If you're using personal access tokens, see Add saved connections with personal access tokens.
4. Click **Save**.

After you create a saved connection, it's listed in the Select a Connection window next time you sign in to the source and destination sites.

Add saved connections with personal access tokens

Adding a saved connection with a personal access token requires more information than other sign-in methods. You will need to create a new personal access token on the source and destination sites to begin. Personal access tokens should not be shared between applications. For more information, see Personal Access Tokens.

Creating personal access tokens

1. In a web browser, sign in to your Tableau site.
2. At the top of the page, click your profile image or initials, and then select **My Account Settings**.

3. Under **Personal Access Tokens**, enter a descriptive name for your token in the **Token Name** field, and then click **Create new token**.
4. In the resulting window, click **Copy to clipboard** and then close the window.
5. Paste the token secret to a file. Store the file in a safe location.

Adding personal access tokens

1. In the Content Migration Tool, click **Add or edit saved connections**.
2. On the **Manage Tableau Connections** window, enter a **Connection name** and the **Server URL**.

   If you are connecting to Tableau Cloud, you must enter the full pod URL of your site. For example, `https://10ay.online.tableau.com`. Your pod is shown in the first portion of the site URL after signing in to Tableau Cloud.
3. Enter the **Personal access token name** and **Personal access token secret**, obtained when creating a personal access token in the previous section.
4. In the **Site name** field, enter the site name as it appears in the URL, without spaces. This is different than the friendly site name. For example, “Site A” would be “sitea” in a browser URL.
5. Click **Save**.

Step 3: Continue to the next step

After successfully signing in to both source and destination sites, click **Next** to continue to the Migration Plans: Source Projects section of the planning phase.

Who can do this

Tableau site user with an Explorer role or higher. To migrate content, you must have **View** and **Download/Save a Copy** capabilities for workbooks on the source site and **View** and

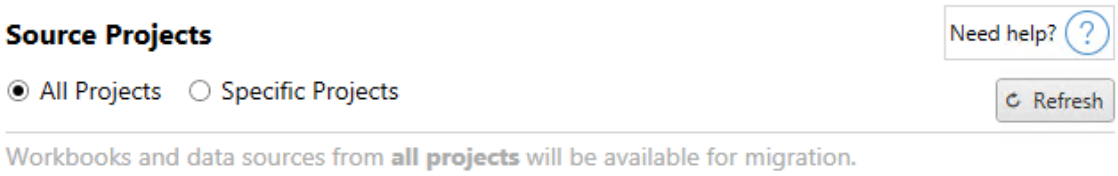
Publish capabilities for target projects on the destination site. For more information, see Permissions.

Migration Plans: Source Projects

The next step in creating a migration plan in the Tableau Content Migration Tool is to select the source projects. Source projects are the projects the workbooks and published data sources will be migrated from. The projects you choose determine which workbooks are available to migrate in the next step of the migration plan.

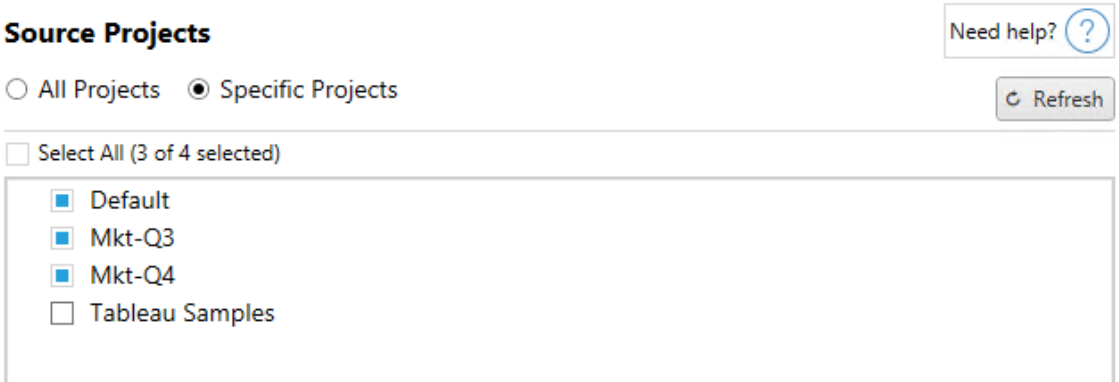
Step 1: Select your source project

There are two options when selecting source projects, **All Projects**, and **Specific Projects**:

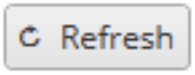


The **All Projects** option selects all projects from the source site you specified in the Servers step. The **Specific Projects** option allows you to select specific projects from the source site.

> **Note:** Source projects must contain workbooks or data sources. Content Migration Tool will not migrate empty projects.

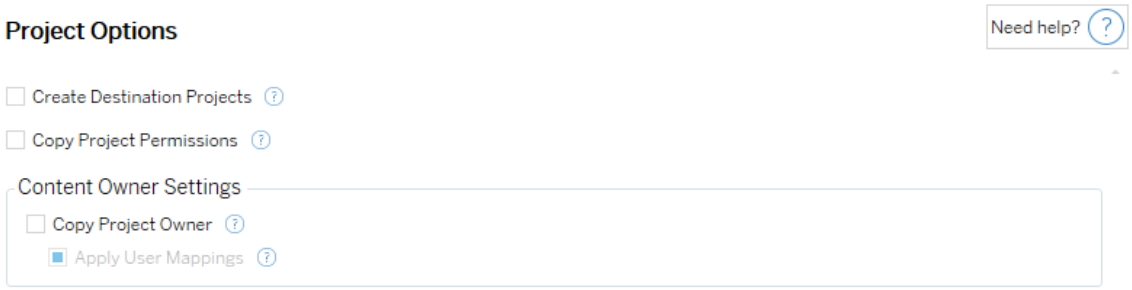


You can select each project individually or use the **Select All** button and then clear selections for the projects you don't want to include. If you make any changes on the source site while on this step, you can use the **Refresh** button to update the projects list.



Step 2: Select project options

Once the source projects are selected, select which project options to apply for the destination location. There are options to create projects that don't exist, in addition to copying project permissions and ownership from the source location. To assign new content ownership based on user mappings, select **Apply User Mappings**.



- **Create Destination Projects**: Automatically create projects that don't exist in the destination location. Content Migration Tool will not create destination projects if the source project is empty or no workbooks or data sources are selected. By default, attempts to migrate to a non-existent project will result in a failed migration.
- **Copy Project Permissions**: Copy source project permissions as closely as possible.
- **Copy Project Owner**: Copy project ownership settings from the source location to assign the project owner.
- **Apply User Mappings**: Apply user mappings to assign content ownership of projects in the destination location. Content ownership won't be applied if the destination project

already exists. For more information, see Migration Plans: Permissions and Ownership.

Step 3: Continue to the next step

After selecting the source projects, click **Next** to continue to the Migration Plans: Workbooks section of the planning phase. If you are migrating workbooks between two projects on the same Tableau site, you'll choose your destination project in the next section.

Who can do this

Tableau site user with an Explorer role or higher. To migrate content, you must have **View** and **Download/Save a Copy** capabilities for workbooks on the source site and **View** and **Publish** capabilities for target projects on the destination site. For more information, see Permissions.

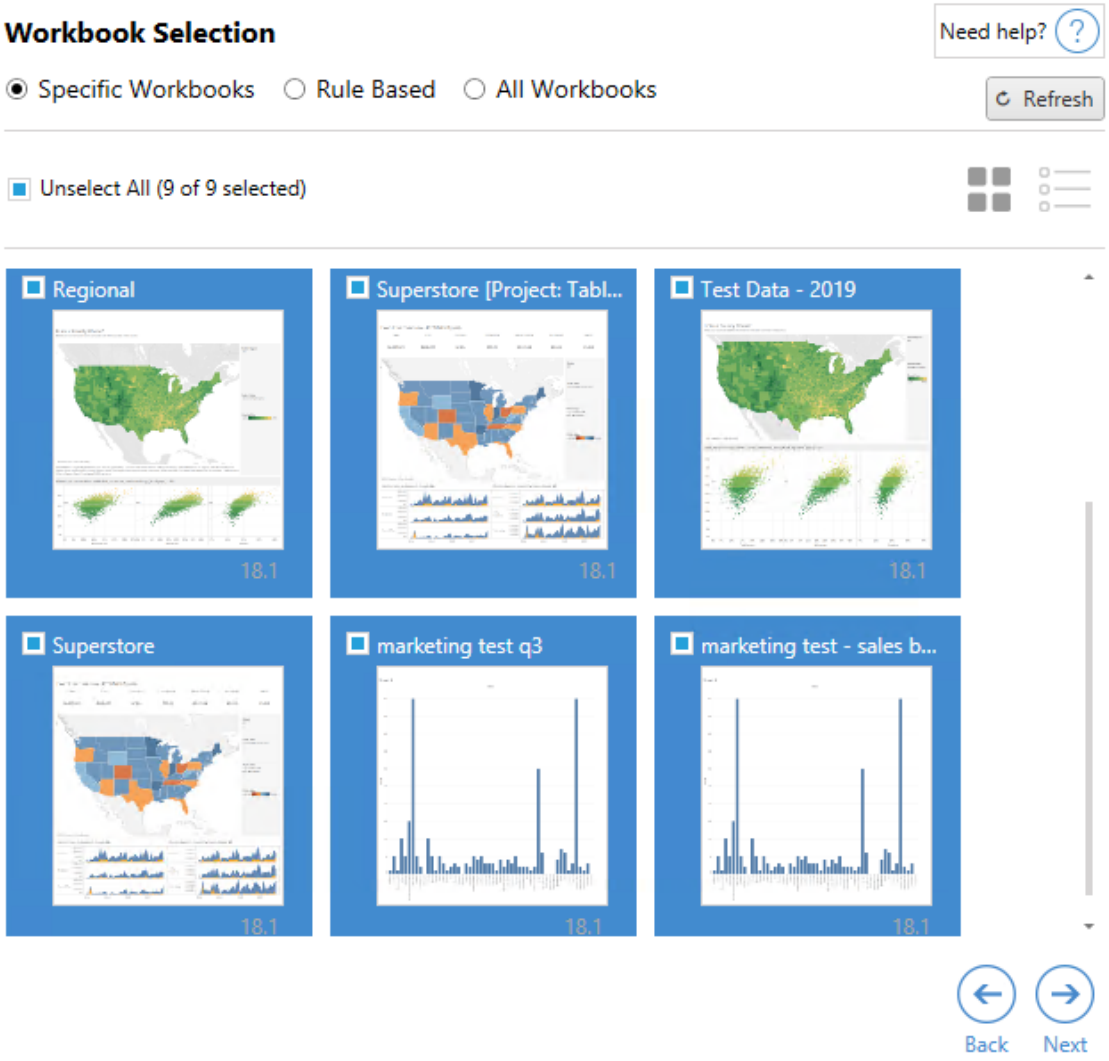
Migration Plans: Workbooks

You have successfully signed in to your source and destination sites and selected projects. The next step is to prepare your workbooks for migration.

Note: If your workbooks or data sources include extracts, be sure you read and understand the information in Migrate Workbooks and Data Sources with Extracts.

Step 1: Workbook selection

All of the workbooks in the source site and selected projects appear on the **Workbook Selection** screen.



If you make any changes to the workbooks in the source site while on this step, you can click **Refresh** to update the workbook listings. There are several different ways to select these workbooks.

Specific Workbooks Selection

There are three buttons in the **Specific** section. Any choices from the Basic section will immediately include the specifically selected workbook in the migration plan. Alternately, you can individually select specific workbooks by clicking on each one.

Select All

This button will select or clear selection of all the workbooks in the site. If additional workbooks are added to the site after the plan is saved, they will not be automatically added the next time the plan is used.

Display:

Thumbnails

The default view shows your workbooks in thumbnail previews to help you differentiate each of them. In this view, mousing over the thumbnail will show previews of the other worksheets and dashboards within that workbook.

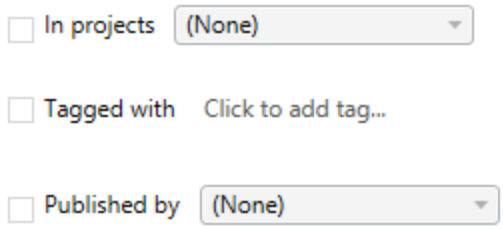
List

The list view is a more succinct listing that also provides additional information, including Workbook Name, Project, Tableau Version, and Last Modified.

Clicking on any of the column headers will sort the workbooks appropriately. Also, mousing over any of the workbooks will also provide a floating preview of the worksheets and dashboards within that workbook. List view is particularly useful if you have a large amount of workbooks in a site.

Rule Based Selection

You can use **Rule Based** selection to choose workbooks based on specific criteria. Rule-based options will create workbook selection criteria to be used when the migration plan is run. Be aware that selecting "all" in any of the **Rule Based** options is different than the **Specific Workbooks** selection. A rule-based "all" selection will always include all workbooks, so any newly added workbooks are included in future migrations.



The **Rule Based** radio button allows you to select workbooks by using the following options:

Workbooks in projects

This menu allows you to select workbooks from specific projects.

Workbooks tagged with

This menu allows you to select workbooks by their tags.

Workbooks published by

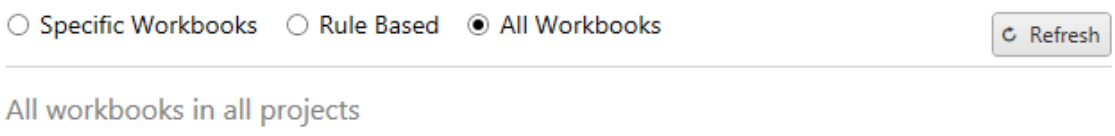
This menu allows you to select workbooks by their author.

With each option, you can select individually or multiple by clicking on the option next to each entry. All selected workbooks will appear in the **Selection Description** box.

All Workbooks Selection

The last option is to select the **All Workbooks** radio button, which selects all workbooks in all projects in the site.

Using the **All Workbooks** radio button is different than selecting all of the workbooks using the **Specific Workbook** method because it will use every workbook in the source site each time the migration plan is used in the future.

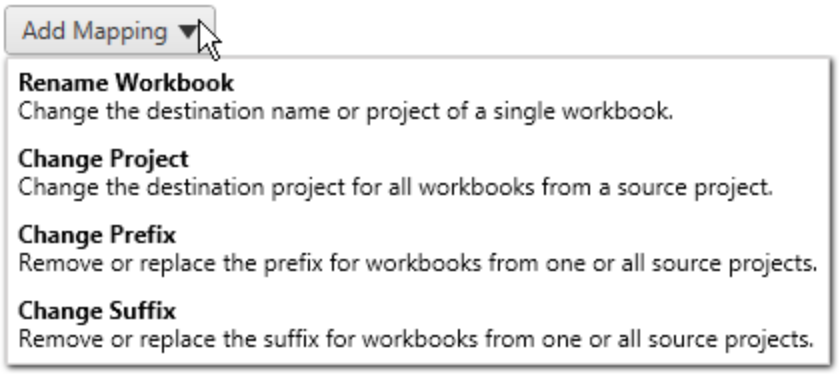


When you are satisfied with your workbook selections, click **Next**.

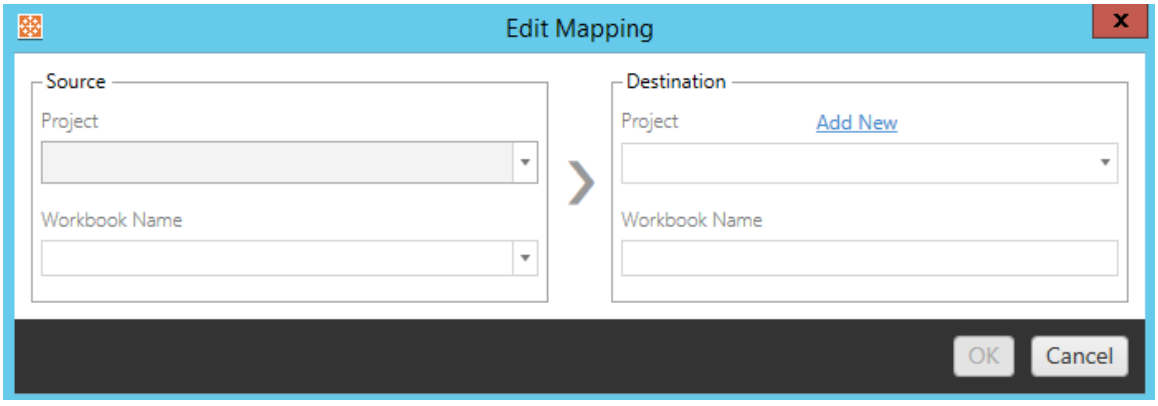
Step 2: Workbook mapping

You can now map your selected workbooks from the source file to the destination file. Mapping allows you to rename source workbooks as they are migrated and choose different destinations. You can also add mapping to change the project, prefix, or suffix for the workbooks as well. Projects can be added to the Destination in this section as well.

If you make no changes here, then the selected workbooks will simply be migrated with the same name and into the same project as the source. If you have not defined projects in your destination site, then they will be migrated into the Default project. To add workbook mapping click the **Add Mapping** button. The following options will appear in the mapping area.

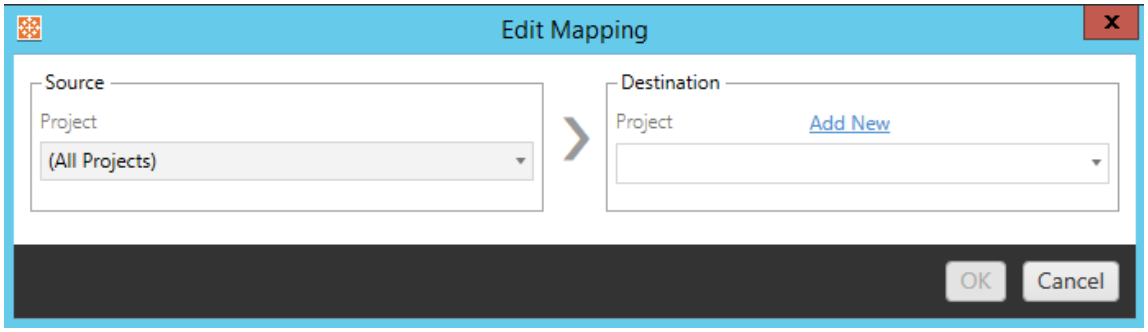


Rename Workbook



This transformation allows you to filter by the **Source** project and select the desired workbook(s) to rename. In the **Destination** field, select which project you would like the workbook to be directed to and enter the desired name.

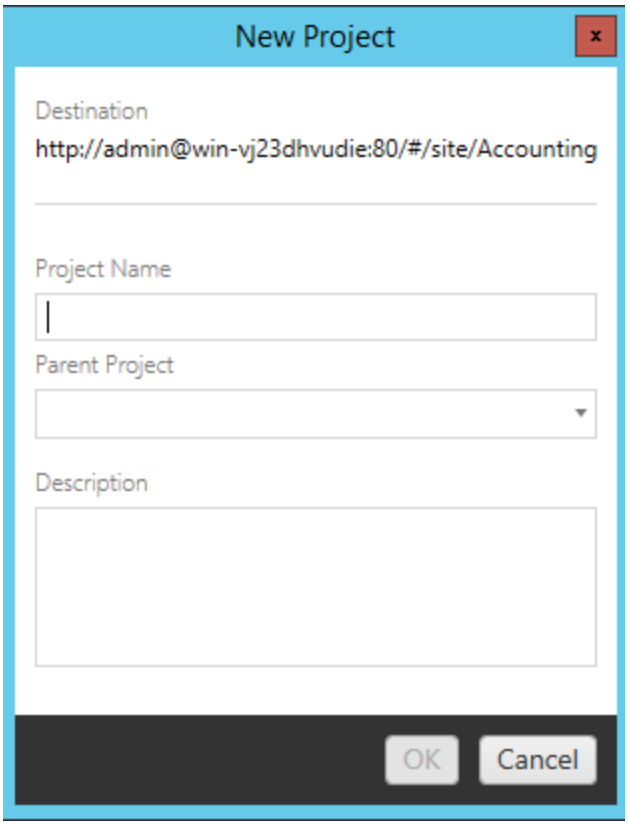
Change Project



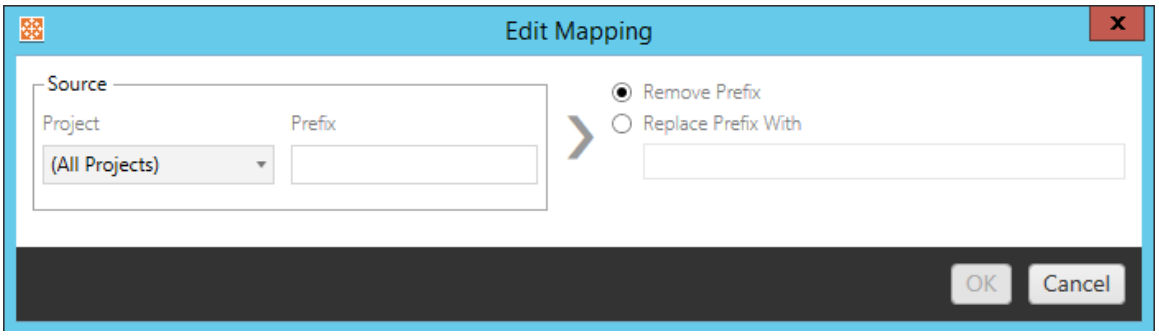
By default, the workbooks are migrated to the same project in the destination. This mapping allows you to change the destination project for all workbooks from a source project.

Add Project

When renaming the workbook or changing the project, the **Add New** option allows you to create a destination project without having to sign in to the destination site and create the project manually. You can create both projects and nested projects using the **Add New** dialog box.



Change Prefix



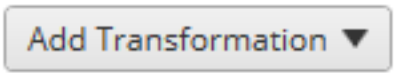
This allows you to remove or replace the prefix for workbooks from one or all source projects.

Change Suffix

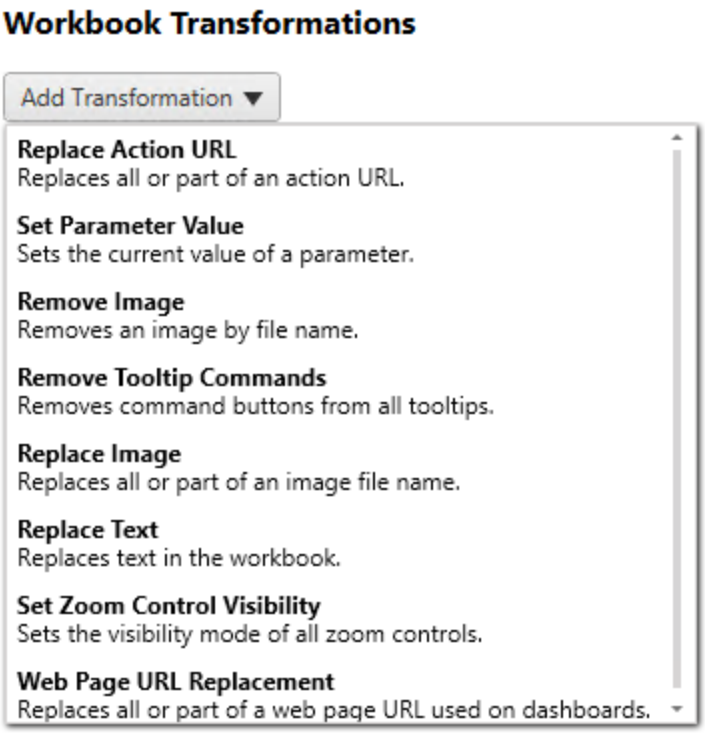
Like the prefix mapping, you can remove or replace the suffix for workbooks from one or all source projects.

Step 3: Workbook transformations

You can change and modify your workbooks by using the Transformation step.

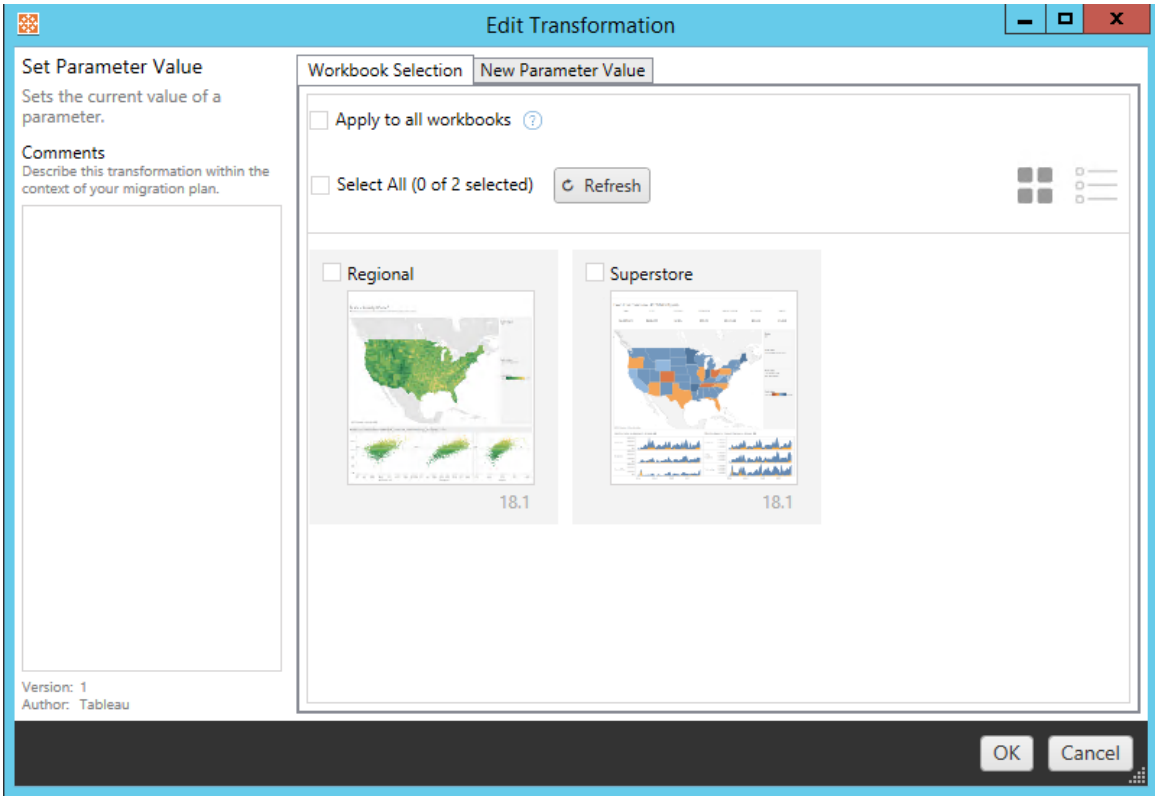


Transformations modify your workbooks in a specified manner. Additional transformations can be included via plug-ins or will be added in future versions of the application. Click on the **Add Transformation** drop-down menu to see the selection of Transformations currently available.

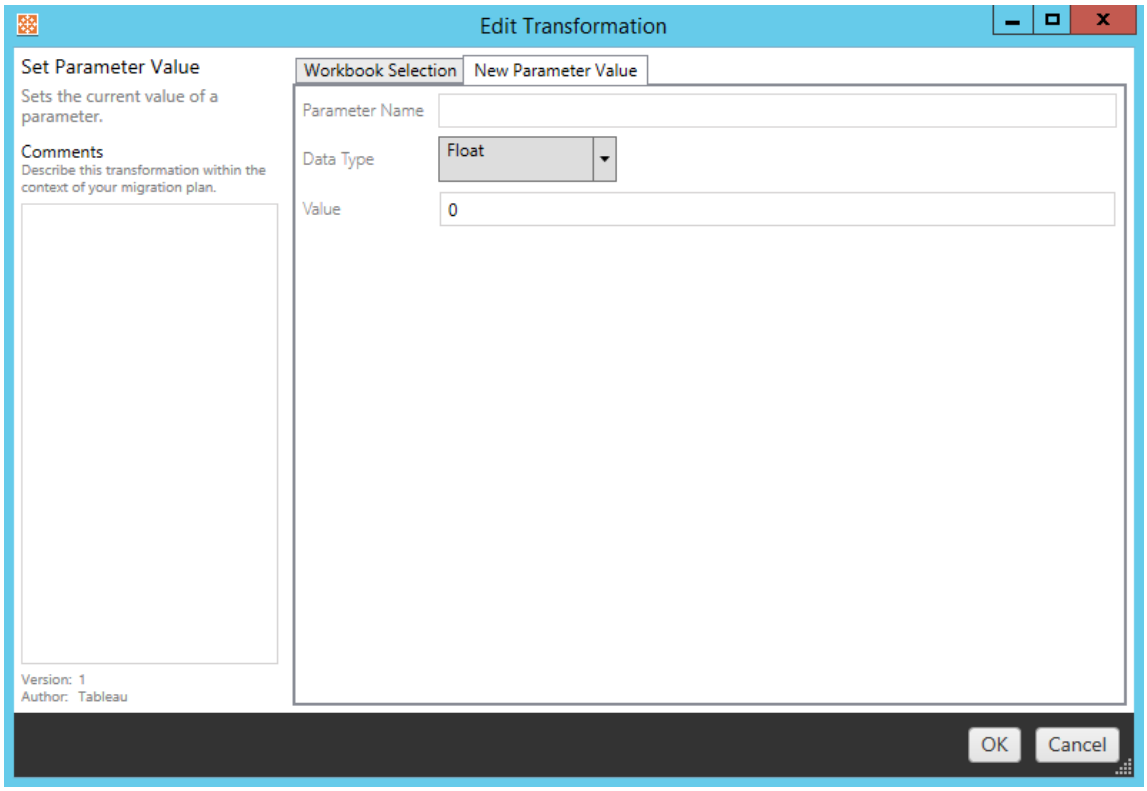


Selecting any of the transformations will bring up the Edit Transformation window, which will allow you to customize it to your selected workbooks. All transformations will be completed in the order that they are listed from top to bottom.

For all of the different types of transformations, there are two basic steps. The first step is to make your selection for the transformation. In this case, select the workbook(s) you want to transform. The selection area is similar to the Workbook Selection section of the Planning phase with all of the features of the Basic selection radio button: **Select/Unselect All**, **Refresh**, **Thumbnail** Display, and **List** Display. At the top of the list, you can select **Select All** workbooks, which is an option to automatically select all workbooks for future transformations. You can also **Refresh** the workbook display window to reflect any changes or updates to the source site.



The second step is to use the options tab to enter the specific selections for whichever transformation you select.



Each of the workbook transformations have different values to be entered on the options tab, and the tab will have different names, depending on the transformation you're editing:

Action URL Replacement

Replace part or all of an URL action inside of the workbook using this transformation. On the options tab, enter the text that should be matched and its replacement value.

Example:

URL: `www.exampledev.com`

Match: `dev`

Replacement: `Prod`

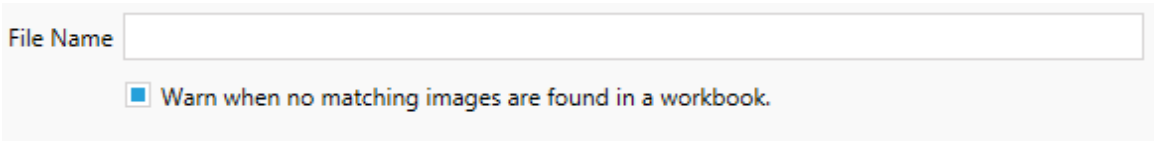
Result: `www.exampleProd.com`

Set Parameter Value

Define a new parameter. On the options tab, enter the name of the Parameter, the data type from the drop-down menu, and the value.

Remove Images

Remove any images (such as a watermark) in the selected workbooks by entering in the file name on the options tab. There is an additional check box to receive a warning during migration if no image is found.

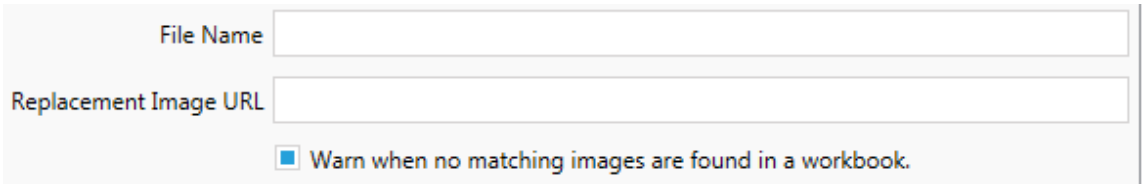


Remove Tooltip Commands

Remove all of the tooltip commands from the selected workbooks. There are no additional options to define for this transformation.

Replace Images

Replace images embedded in the selected workbooks. On the options tab, enter the file name of the current image and the replacement image. You can replace images using a local file path or URL.



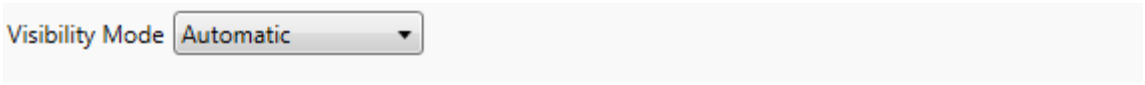
Example:

File Name: `image.png`

Replacement Image URL: `https://www.exampledev.com/replacementImage.png`

Zoom Control Visibility

Set the visibility mode from the drop-down menu: **Automatic**, **Show on Hover**, or Hide on the options tab.



Web Page URL Replacement

Replace part or all of a web page URL used on dashboards using this transformation. On the options tab, enter the text that should be matched and its replacement value.

Example:

URL: `www.exampledev.com`

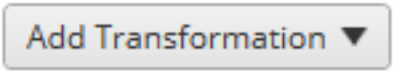
Match: `dev`

Replacement: `Prod`

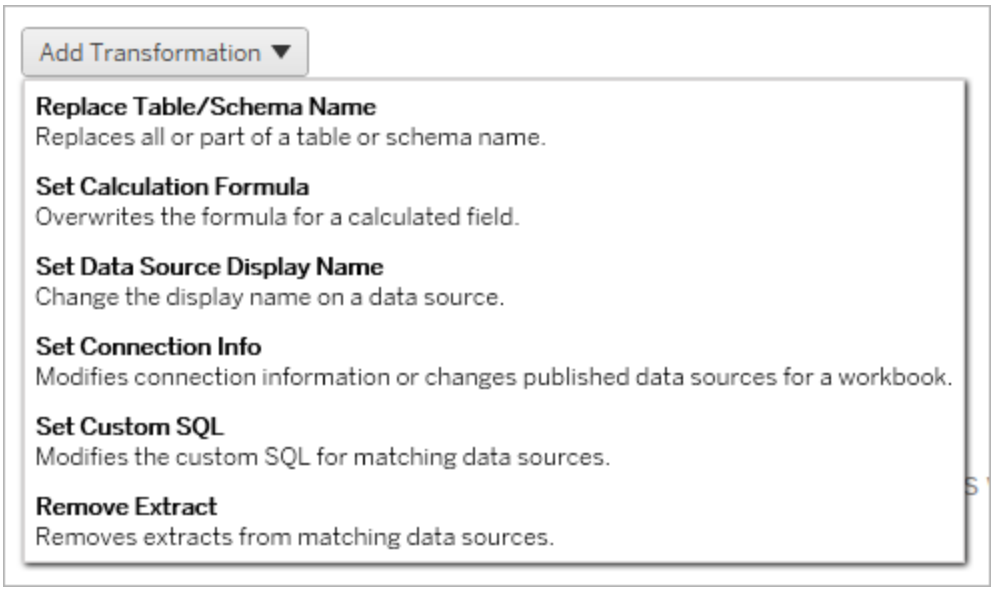
Result: `www.exampleProd.com`

Step 4: Data source transformations

The next step in planning your workbooks for your enterprise migration are your data source transformations. It is similar in function to the Workbook Transformations step. These are for data sources that are packaged within the workbooks. Published data sources are handled in a different step in the process.

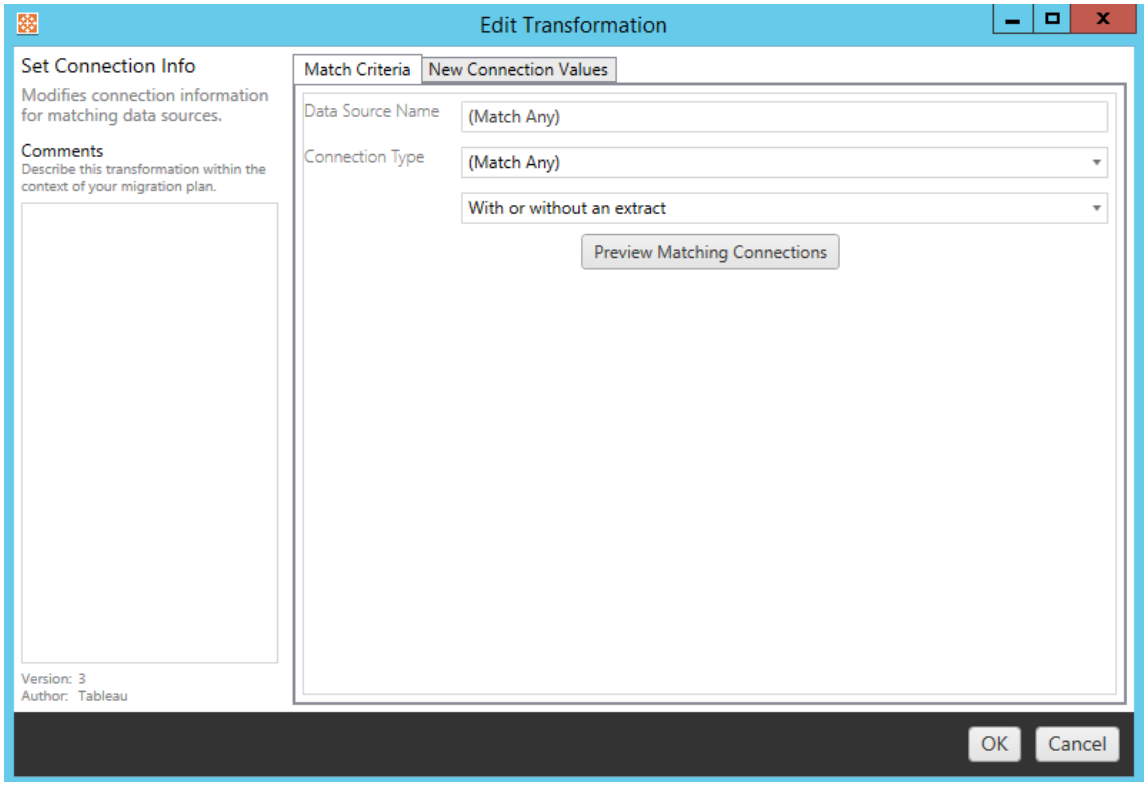


Click on the **Add Transformation** drop-down menu and the following options will appear:

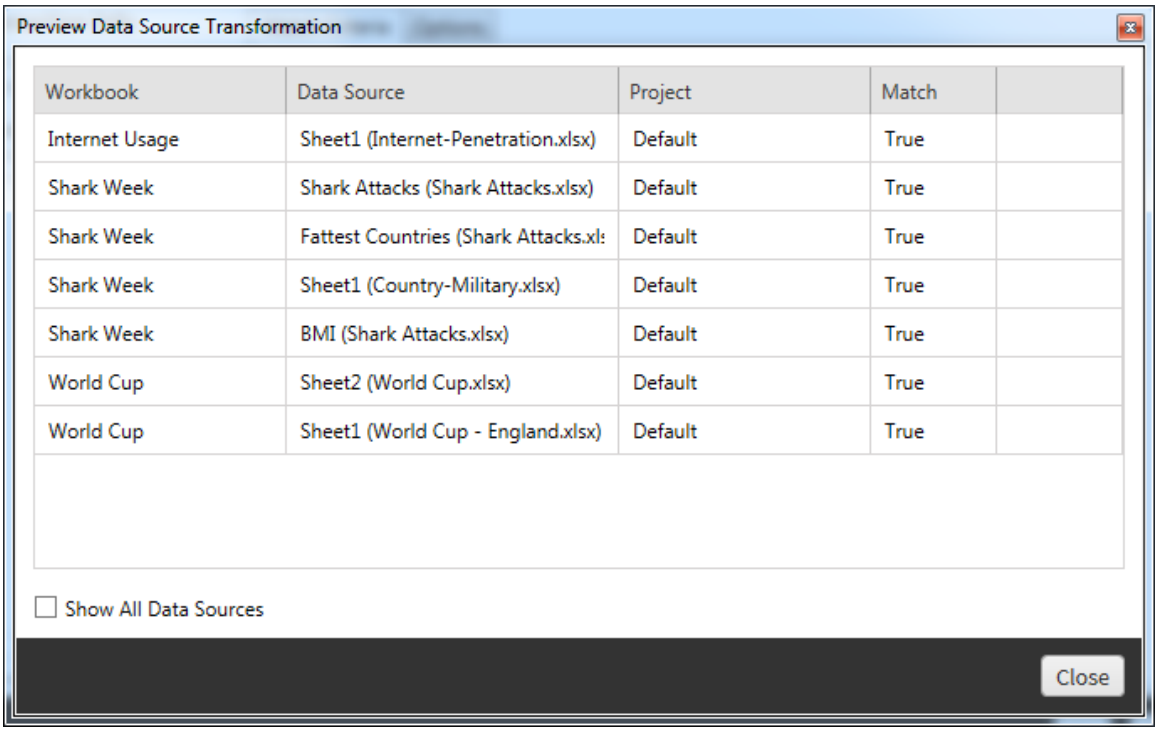


Selecting any of the data source transformations will bring up the Edit Transformation window, which will allow you to customize it to your selected data sources. All transformations will be completed in the order that they are listed from top to bottom.

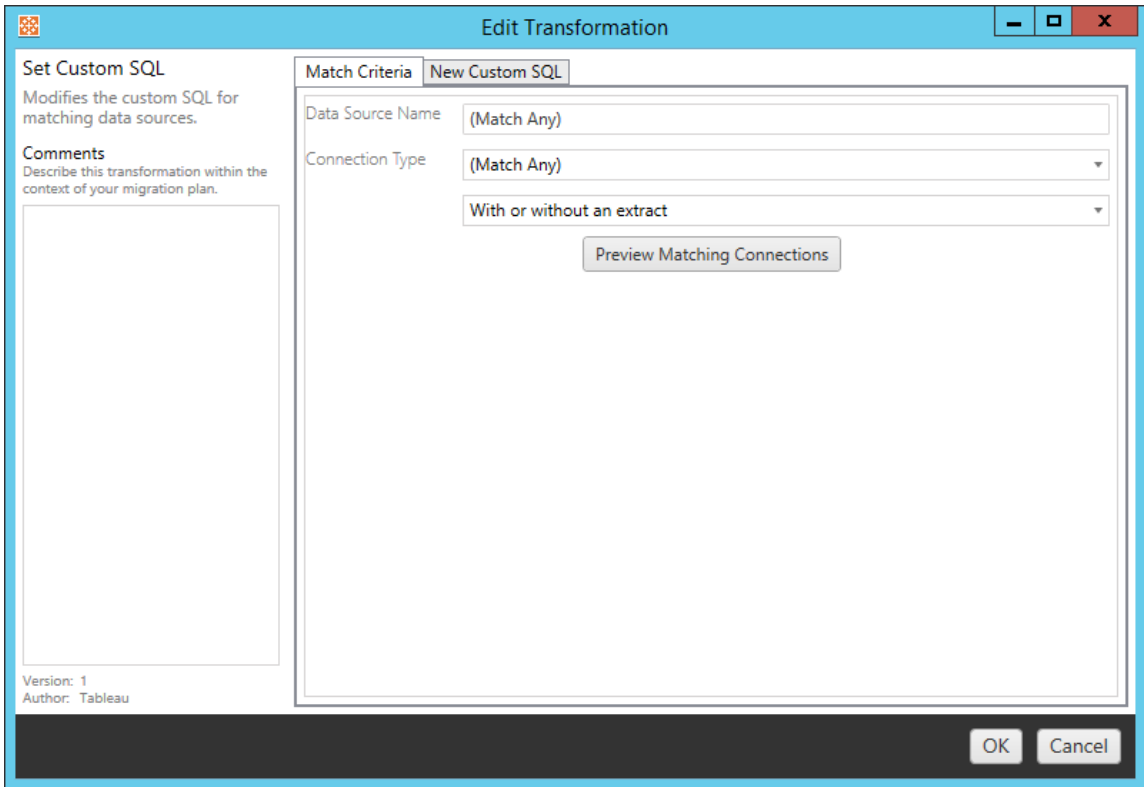
For all of the different types of data source transformations, there are two basic steps. The first step is to enter in the match criteria for the desired data source. Depending on which connection type you select, more fields will appear on the **Match Criteria** tab.



Click on the **Preview Source Connections** to find any connections that match the criteria entered.



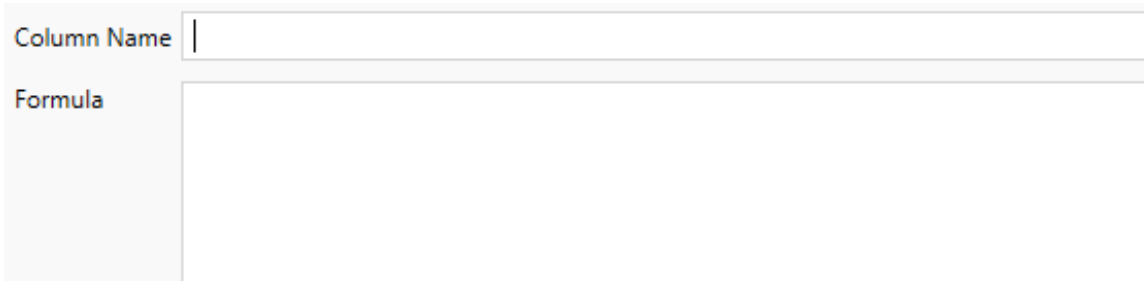
The second step is to use the options tab to enter the specific selections for whichever transformation you select.



Each of the data source transformations have different values to be entered on the options tab:

Set Calculation Formula

On the options tab, you can replace the calculation for a column.



Set Connection Info

On the **New Connection Values** tab, enter the authentication method and connection details for the new data source. Depending on which connection type you select, more fields will appear.

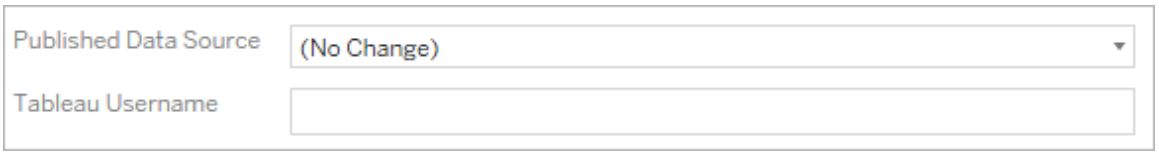
Change published data sources with CMT

As of version 2022.1, select the **Tableau Server (Published Data Source)** connection type to change the published data source for a workbook. This can reduce the manual steps required when migrating workbooks between Tableau environments, for example, promoting content from development to production.

To change the published data source, select a data source from the drop-down menu and enter the **Tableau Username** for authentication. The user must exist on the destination site and have the Connect capability for the published data source.

- For file-based data sources, users will access the workbook and see data based on permissions of the specified Tableau user.
- For all other data sources, users are prompted for their own database credentials when the view or workbook loads.

If Tableau Username isn't specified, only users with the Connect capability can see data in the workbook.



Set Custom SQL

On the **New Custom SQL** tab, enter the name of the custom SQL query you want to modify for **Match Query Name**. The query name must match the custom SQL query name from the phys-

ical layer of the data source. If these names don't match, the transformation will fail. For more information about data modeling and the physical layer, see The Tableau Data Model.

After entering the query name, enter the desired **Custom SQL** in the text field. Be aware that custom SQL can negatively impact the performance of your workbooks if improperly used.

Remove Extract

There is no options tab for this transformation, simply enter in the **Match Criteria** information and the extract will be removed during migration.

In addition, on each of the transformations you can enter notes in the **Comments** section on the left-hand side of the **Edit Transformation** window.

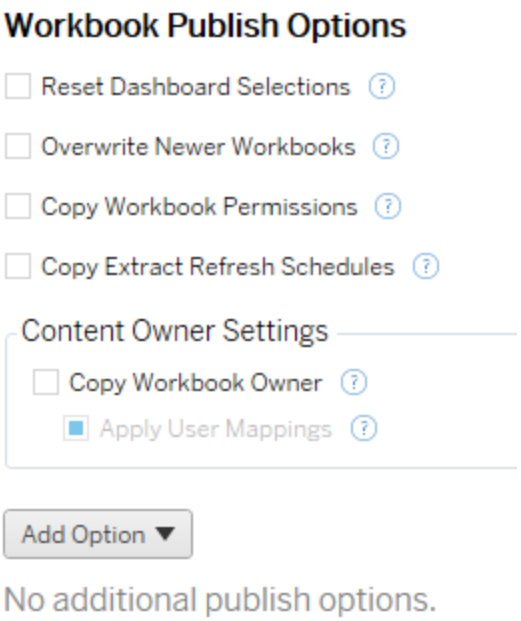
Apply Saved Credentials

Deprecated in version 2022.3. Use the Set Connection Info data source transformation instead.

On the options tab, enter the **Tableau Username** and corresponding **Saved Credentials Username** for the data connection. You can only apply saved credentials for existing data connections on the Account Settings page of your Tableau site. For more information, see Manage Saved Credentials for Data Connections.

Step 5: Publish options

The final step in the Workbooks phase is to select publish options and create transformations for tags, extract refresh schedules, and permissions.



Reset Dashboard Selections

This option deselects all objects on dashboards.

Overwrite Newer Workbooks

If checked, a workbook will be migrated even if it will overwrite a workbook that has been created at the same time or more recently than the moved workbook

Copy Workbook Permissions

When selected, the migration tool will attempt to match source workbook permissions as closely as possible.

Copy Extract Refresh Schedules

When selected, the migration tool will attempt to set the destination workbook extract refresh schedule(s) to schedules matching the source's name.

Note: Extract refresh schedules cannot be created in Tableau Cloud. This option is not available if the destination is a Tableau Cloud site. For more information, see Migration Limitations.

Copy Embedded Credentials for Workbooks

Copy the embedded credentials for data sources embedded in workbooks. Only available when migrating from Tableau Server to Tableau Cloud sites. For more information, see Migrate Workbooks and Data Sources with Embedded Credentials.

Note: CMT does not support embedded credential migration for OAuth connections. To migrate OAuth credentials to the destination site, use the Set Connection Info data source transformation.

Copy Workbook Owner

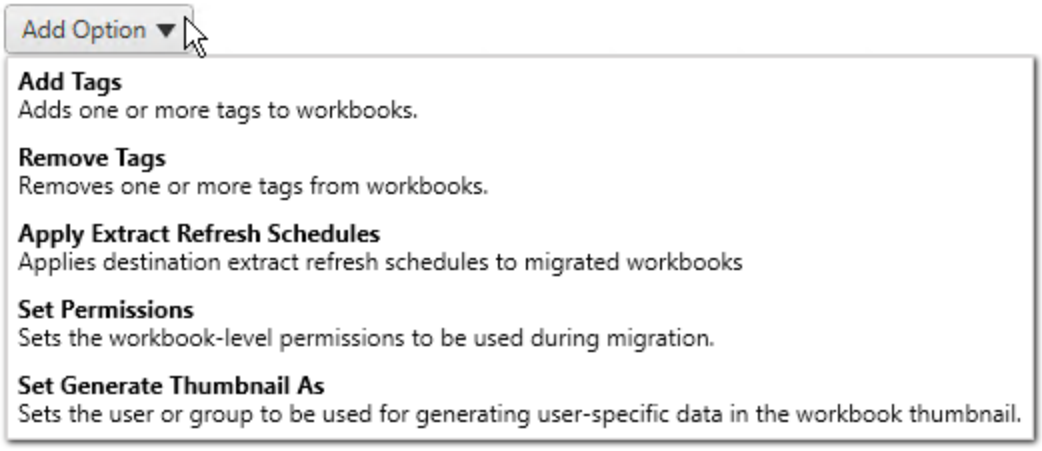
Copy workbook owner settings from the source location to assign the workbook owner. If unselected, the Content Migration Tool user is given ownership of the workbook in the destination location.

Apply User Mappings

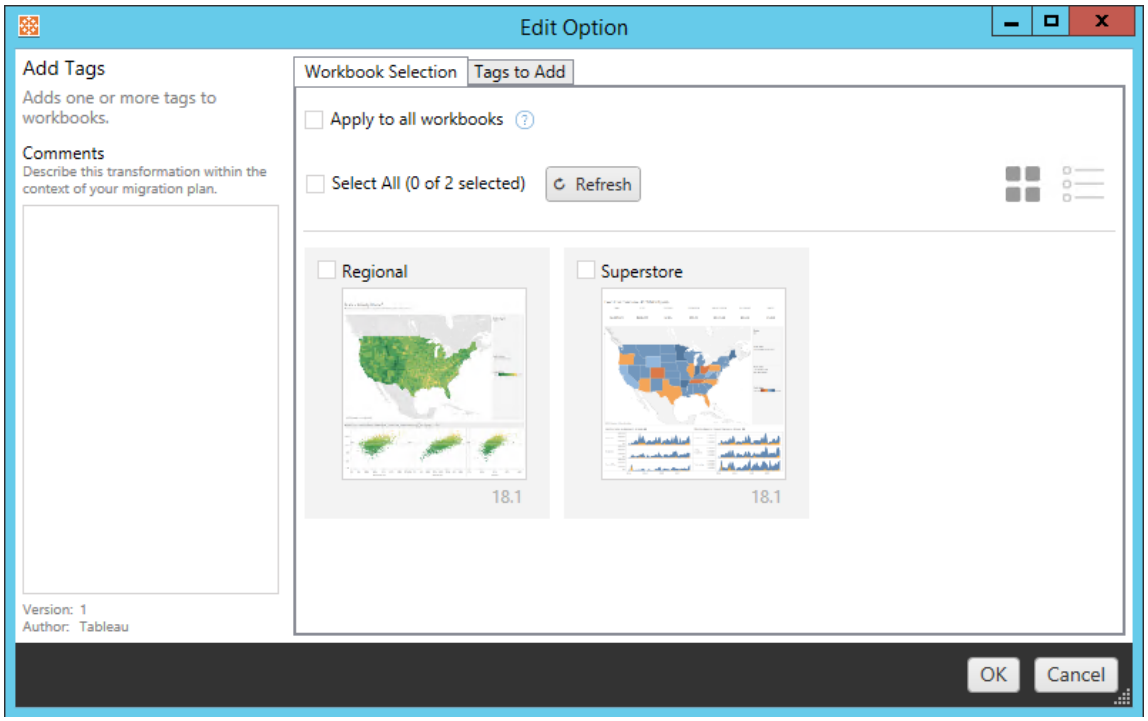
Apply user mappings to assign content ownership. Select this option if there are differences in username syntax in the destination location. For more information, see Migration Plans: Permissions and Ownership.

Add Option

Click on the **Add Option** drop-down menu for the different types of transformations you can add:



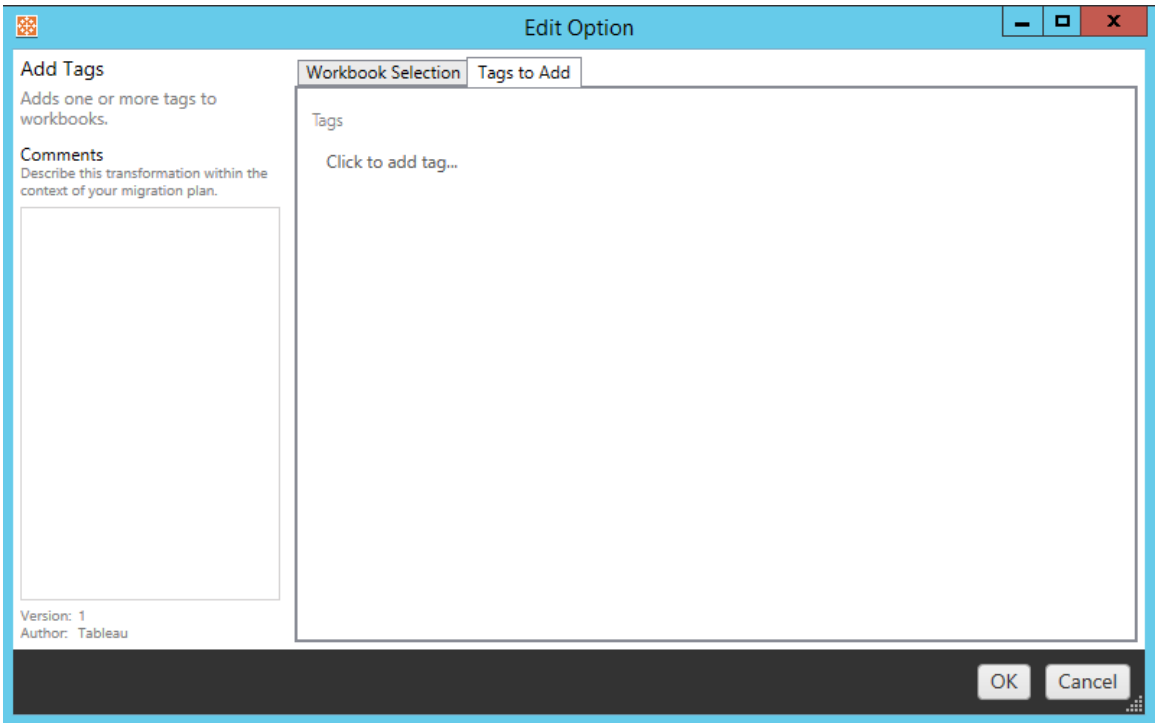
For all of the different types of transformations, there are two basic steps. The first step is to make your selection for the transformation. In this case, select the workbook(s) you want to transform. The selection area is similar to the Workbook Selection section of the Planning phase with all of the features of the Basic selection radio button: **Select/Unselect All**, **Refresh**, **Thumbnail Display**, and **List Display**. At the top of the list, you can select **Select All** workbooks, which is an option to automatically select all workbooks for future transformations. You can also **Refresh** the workbook display window to reflect any changes or updates to the source site.



The second step is to use the options tab to enter the specific selections for whichever transformation you select. **Note:** The options tab will have different names, depending on which transformation you are editing.

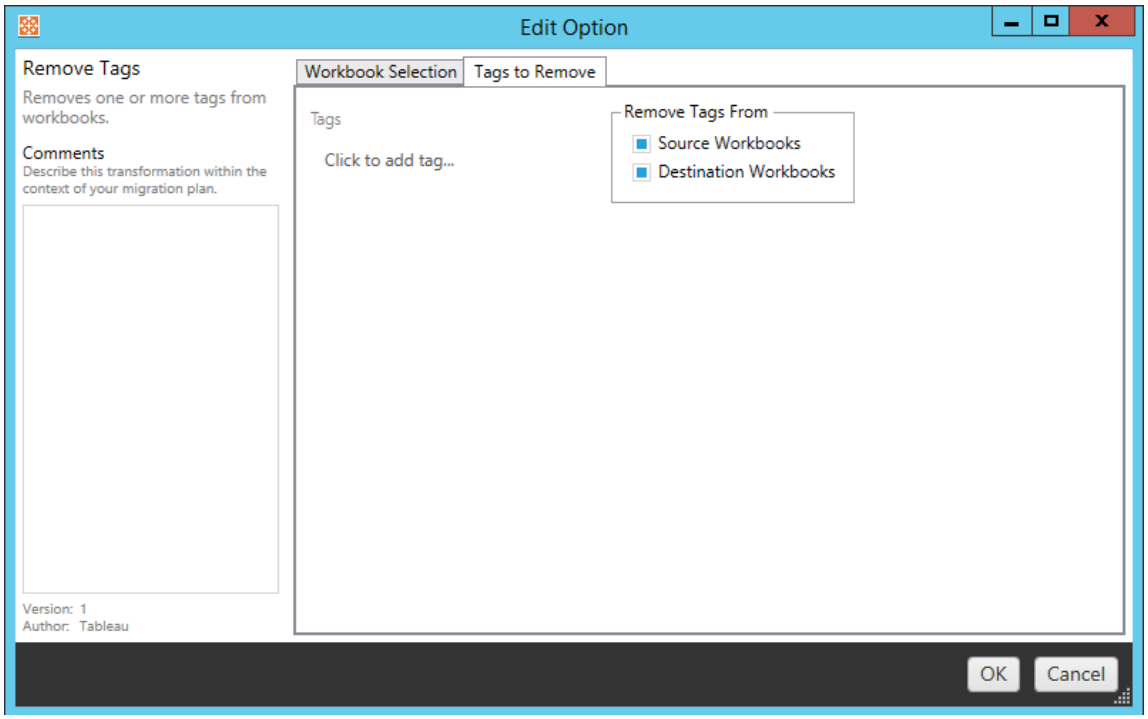
Add Tags

This allows you to add one or more tags to the workbook. If you hover your mouse over a previously entered tag, a blue “X” will appear to allow deletion.



Remove Tags

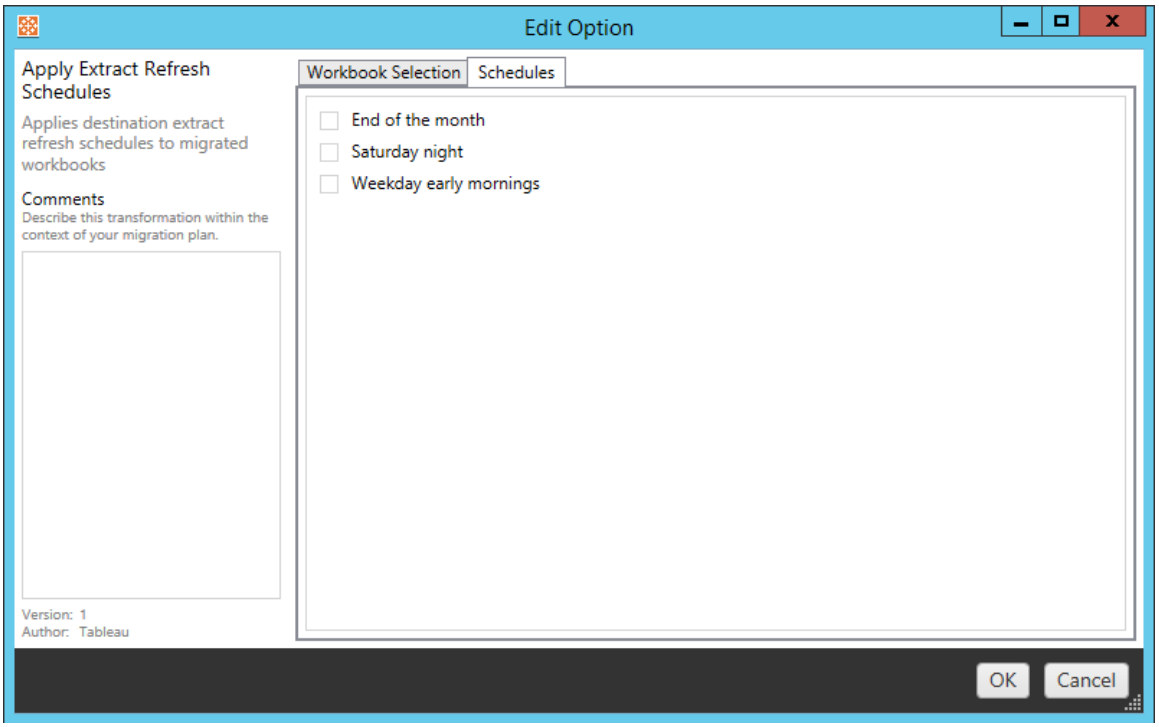
This allows you to add one or more tags to the workbook. If you hover your mouse over a previously entered tag, a blue “X” will appear to allow deletion. You can also choose to remove the tag from the source or destination workbooks.



Apply Extract Refresh Schedules

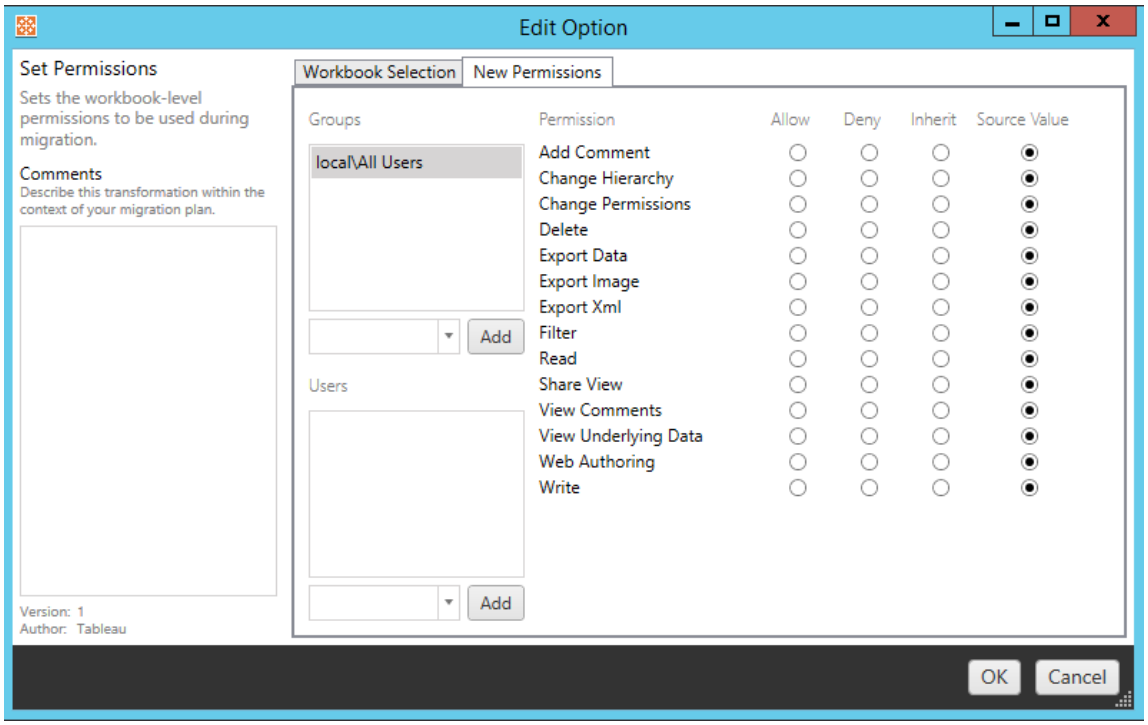
Here you can apply destination extract refresh schedules to migrated workbooks. The list of schedules generated are from the destination.

Note: Extract refresh schedules cannot be created in Tableau Cloud. This option is not available if the destination is a Tableau Cloud site. For more information, see Migration Limitations.



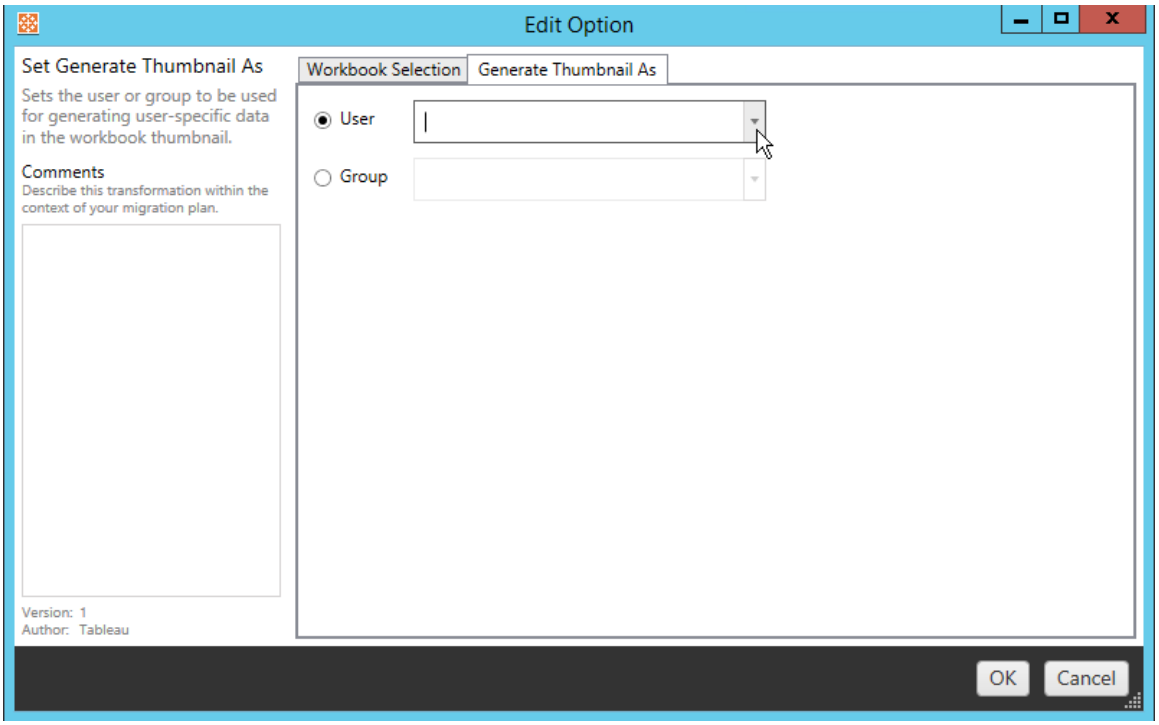
Set Permissions

This transformation is to edit the permissions for the selected workbooks. Enter in a Group or User and then click **Add**. Adjust the permissions as desired. The four different options are to **Allow** the permission, **Deny** the permission, **Inherit**, or to keep the **Source Value**.



Set Generate Thumbnail As

This allows you to set the **User** or **Group** to be used for generating user-specific data in the workbook thumbnail after being migrated. Each option has a drop down to select the desired user or group.



Step 6: Continue to the next step

After selecting your workbooks and preferences, click **Next** to continue to the Migration Plans: Published Data Sources section of the planning phase.

Who can do this

Tableau site user with an Explorer role or higher. To migrate content, you must have **View** and **Download/Save a Copy** capabilities for workbooks on the source site and **View** and **Publish** capabilities for target projects on the destination site. For more information, see Permissions.

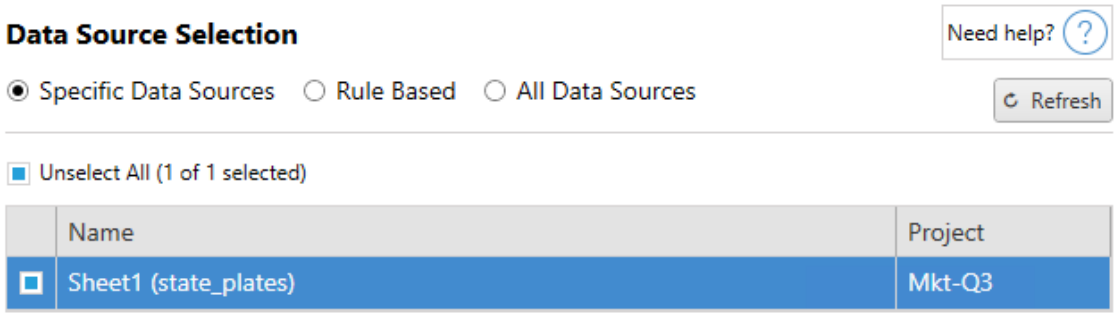
Migration Plans: Published Data Sources

The next step of creating a migration plan in the Tableau Content Migration Tool is to select, map, and add any transformations to your published data sources. The process is very similar to the Workbooks step of the planning phase, particularly the data source mapping step.

Note: If your workbooks or data sources include extracts, be sure you read and understand the information in Migrate Workbooks and Data Sources with Extracts.

Step 1: Selection

Starting the Published Data Sources phase of the migration plan, you'll select any data sources you want to include in the migration plan:

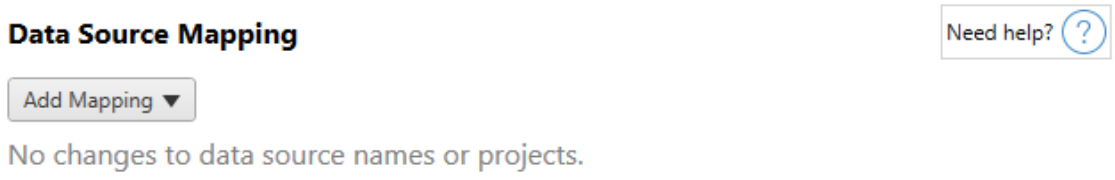


The data sources will only be selected at the moment of migration. You have two methods of selection. Use **Specific Data Sources** to choose one or more published data sources. Click **Refresh** to reload the list of published data sources available.

The second option is **All Data Sources**, which selects every data source in the source site.

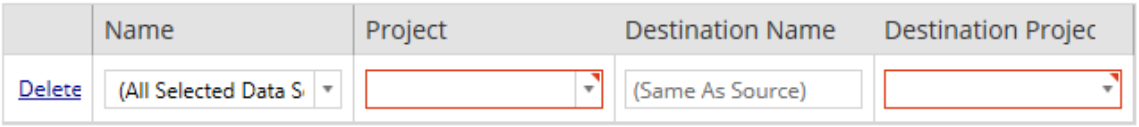
Step 2: Mapping

The next step is to map your source data sources to the new destination. This is similar in functionality to mapping workbooks.



If you make no changes here, then the selected data sources will simply be deployed with the

same name and project as the source. To add data source mapping click **Add Mapping**. The following options will appear in the mapping area.



The entry has the following options:

Delete

Clicking the **Delete** link will delete this mapping entry.

Name

Use the **Name** menu to select the data source you wish to map. You can select **(All Selected Data Sources)** to choose all of the data sources.

Project

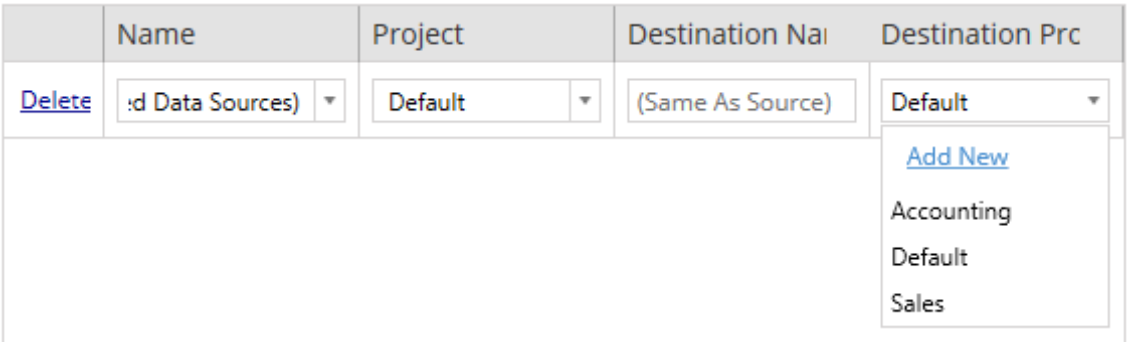
The **Project** is the project of the associated data source names.

Destination Name

By default, the Content Migration Tool will use the same **Destination Name(Same As Source)**, keeping the original name in the Source file, but you can type in a new name here for the destination folder.

Destination Project

If your destination projects have already been created on your site, you can choose which project to place your migrated workbooks or click **Add New** to create a new project. You can create different project destinations for individual data sources.

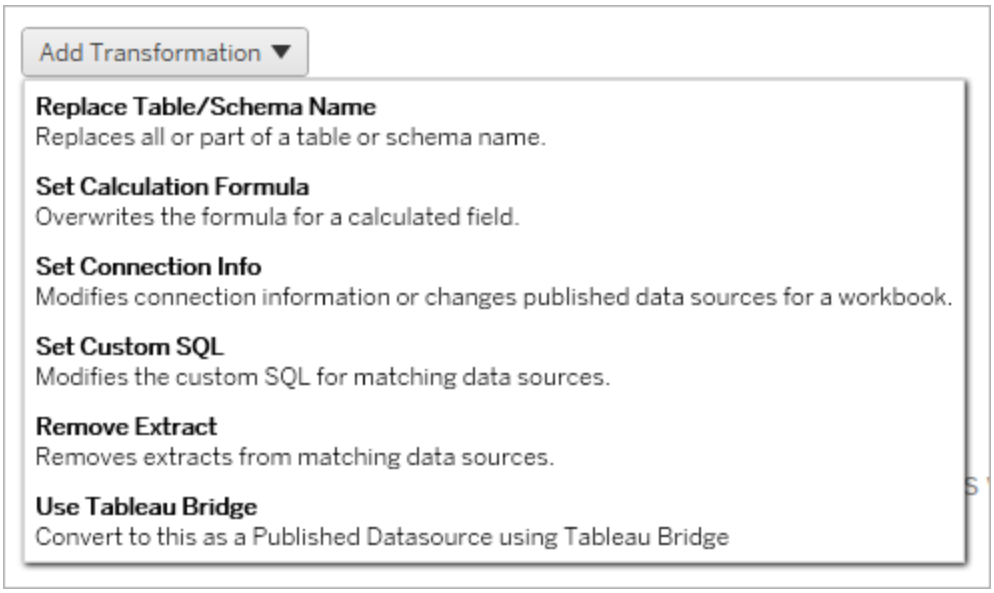


In most situations if multiple mapping entries exist for a single data source, a validation error will be displayed and must be fixed to continue. There is one important exception to this – a data source may match both a specific selection and a project-wide mapping entry. In this instance, the more specific entry will be used.

When you have completed all of the data source mapping necessary, click **Next** to continue.

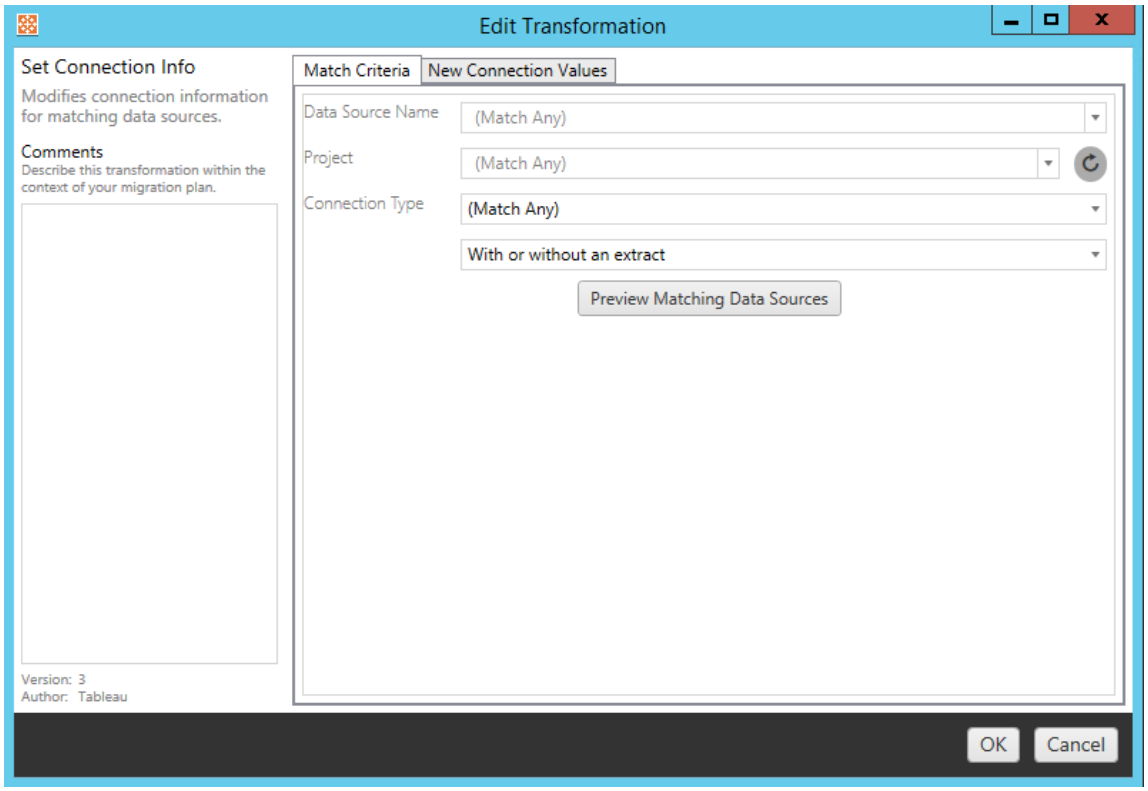
Step 3: Data source transformations

You can modify your data sources by using the transformation option. Click **Add Transformation** to see the list of transformations available.



Selecting any of the data source transformations opens the **Edit Transformation** window. Use this to customize your selected data sources. Transformations are executed in the order they are listed, from top to bottom.

For most data source transformations, there are two basic steps. The first step is to enter in the **Match Criteria** for the desired data source. Depending on which connection type you select, more fields will appear on the **Match Criteria** tab.



The second step depends on which transformation type you are adding. Each of the data source transformations have different values to be entered on the second tab.

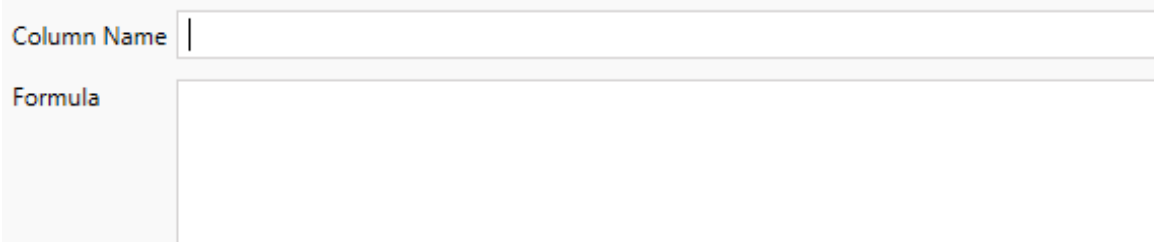
You can add notes in the **Comments** section on the left side of the **Edit Transformation** window for each of the transformations you add.

Replace Table/Schema Name

On the **Options** tab, you can replace all or part of a table or schema name.

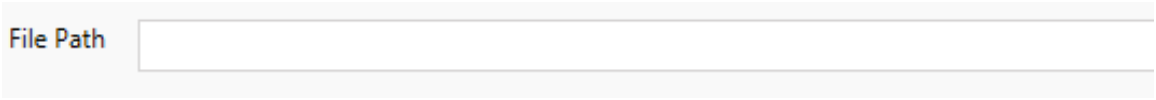
Set Calculation Formula

On the **New Calculation Formula** tab, you can replace the calculation for a column.



Set Connection Info

On the **New Connection Values** tab, enter the authentication method and connection details for the new data source. Depending on which connection type you select, more fields will appear.



Set Custom SQL

On the **New Custom SQL** tab, enter the name of the custom SQL query you want to modify for **Match Query Name**. The query name must match the custom SQL query name from the physical layer of the data source. If these names don't match, the transformation will fail. For more information about data modeling and the physical layer, see The Tableau Data Model.

After entering the query name, enter the desired **Custom SQL** in the text field. Be aware that custom SQL can negatively impact the performance of your workbooks if improperly used.

Remove Extract

There is no **Options** tab for this transformation. Type the **Match Criteria** information and the extract will be removed during migration.

Use Tableau Bridge

There is no **Options** tab for this transformation. Type the **Match Criteria** information and data sources that are within a private network (inaccessible to the public internet) will be allowed to refresh using Tableau Bridge.

The destination Tableau Cloud site must have Tableau Bridge configured before migrating data sources. For information about Tableau Bridge, see Use Tableau Bridge in Tableau Cloud help. After the migration, data sources will need to be assigned a refresh schedule through Tableau Cloud.

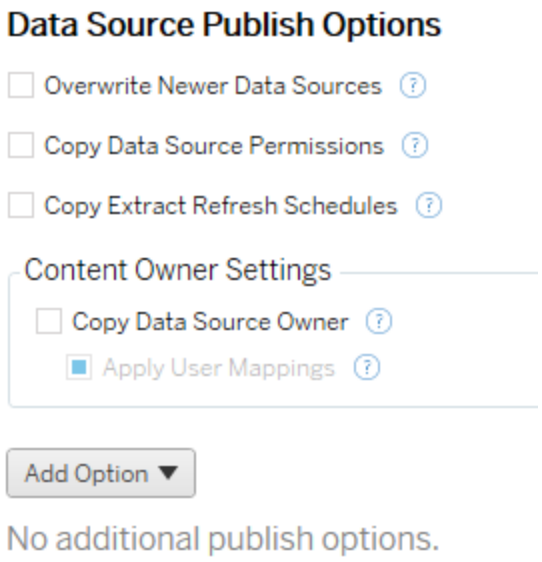
Apply Saved Credentials

Deprecated in version 2022.3. Use the Set Connection Info data source transformation instead.

On the options tab, enter the **Tableau Username** and corresponding **Saved Credentials Username** for the data connection. You can only apply saved credentials for existing data connections on the Account Settings page of your Tableau site. For more information, see Manage Saved Credentials for Data Connections.

Step 4: Publish options

The final step in the Published Data Source phase is to create transformations for permissions and tags and finalize the publish options specific to the data sources.



Overwrite Newer Data Sources

If selected, a data source will be published even if it will overwrite a data source that has been updated more recently.

Copy Data Source Permissions

When selected, the migration tool will attempt to match source published data source permissions as closely as possible.

Copy Extract Refresh Schedules

When selected, the migration tool will attempt to set the destination data source extract refresh schedule to schedules matching the source's name.

> **Note:** Extract refresh schedules cannot be created in Tableau Cloud. This option is not available if the destination is a Tableau Cloud site. For more information, see Migration Limitations.

Copy Embedded Credentials for Data Sources

Copy the embedded credentials for published data sources. Only available when migrating from Tableau Server to Tableau Cloud sites. For more information, see Migrate Workbooks and Data Sources with Embedded Credentials.

> **Note:** CMT does not support embedded credential migration for OAuth connections. To migrate OAuth credentials to the destination site, use the Set Connection Info data source transformation.

Copy Data Source Owner

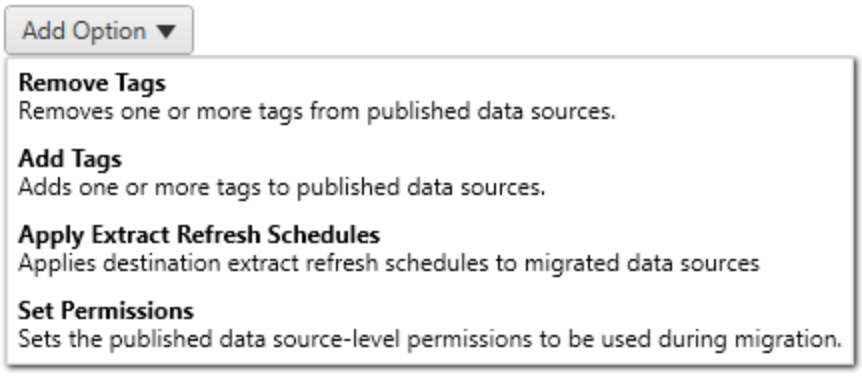
Copy data source owner settings from the source location to assign the data source owner. If unselected, the Content Migration Tool user is given ownership of the data source in the destination location.

Apply User Mappings

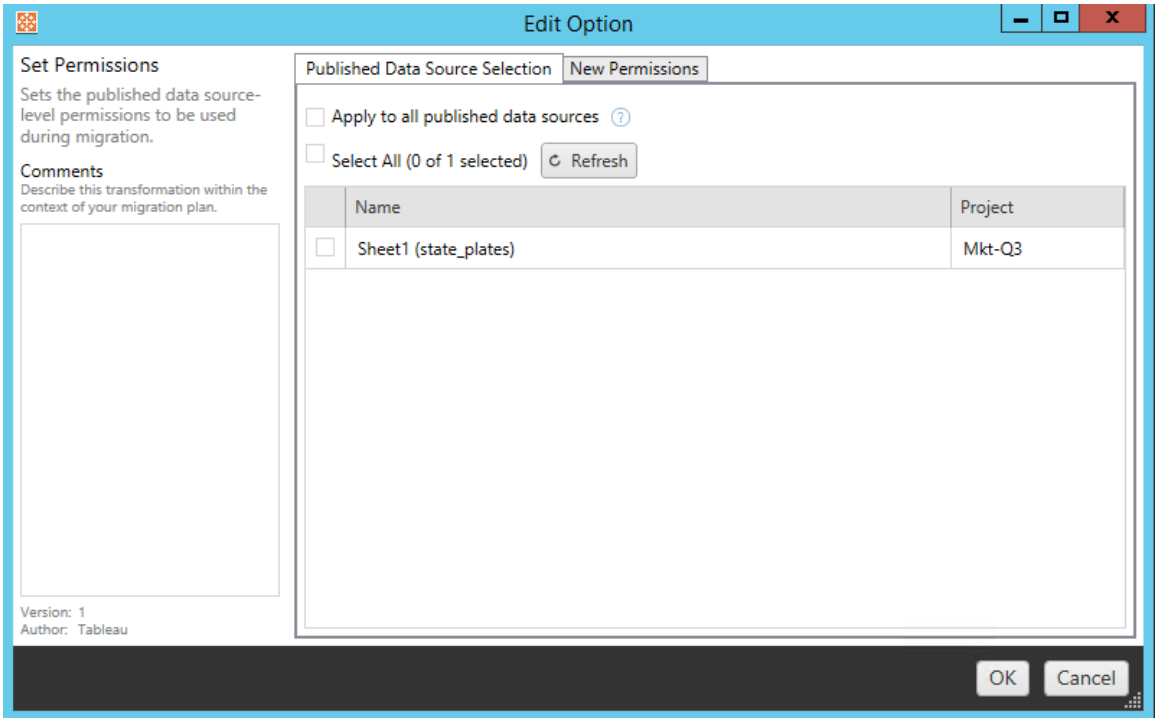
Apply user mappings to assign content ownership. Select this option if there are differences in username syntax in the destination location. For more information, see Migration Plans: Permissions and Ownership.

Add Options

Click on the **Add Option** drop-down menu for the different types of transformations you can add:



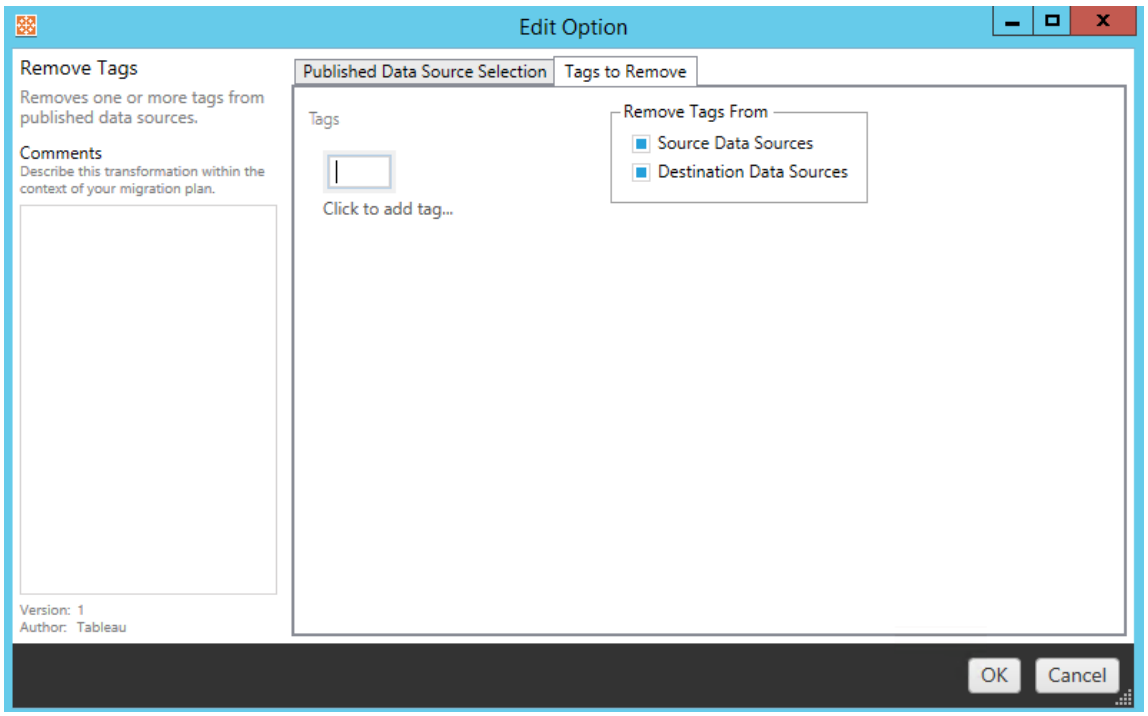
For all of the different types of transformations, there are two basic steps. The first step is to make your selection for the transformation. In this case, select the data source(s) you want to transform. At the top of the list, you can select **Apply to all published data sources**, which is an option to automatically select all data sources for future transformations. You can also **Refresh** the data source display window to reflect any changes or updates to the source site.



The second step is to enter the specific selections for the transformation you select.

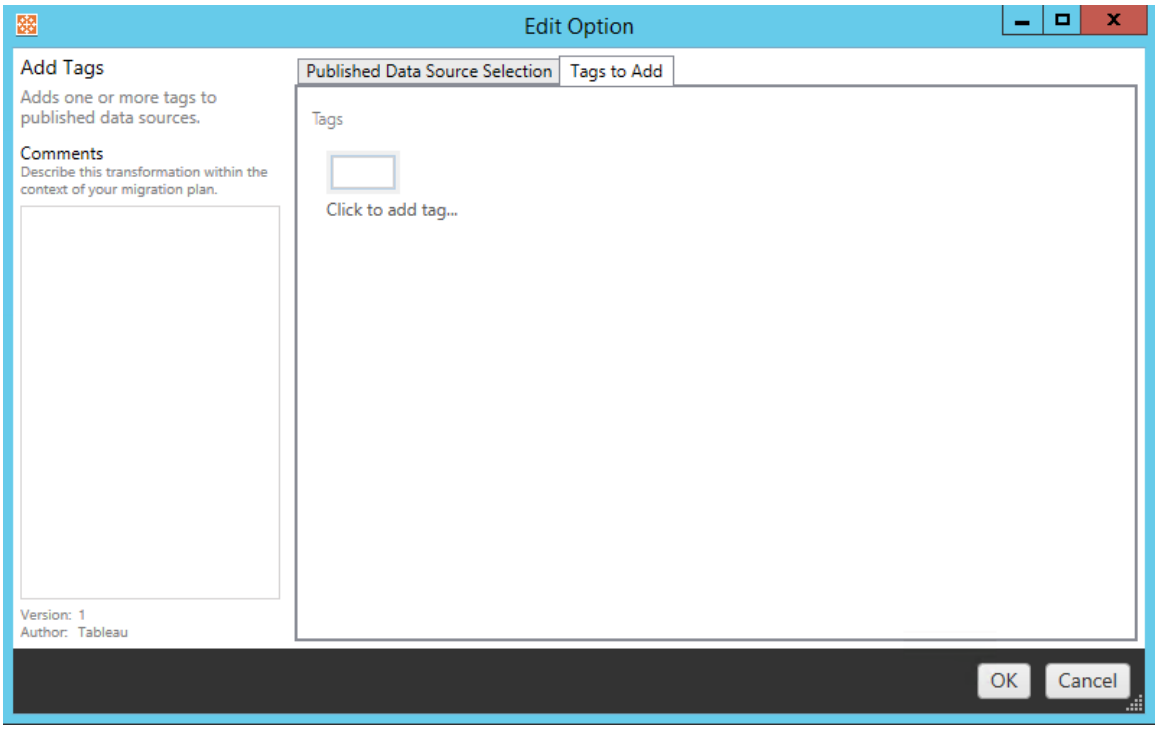
Remove Tags

Once the data sources you would like to remove tags from are selected, enter any tags you want to remove by entering them into the field at the bottom and click **Add**. From this screen, you can also select to remove from the source or destination data sources. If you want to remove a previously entered tag, click on it and press the delete key.



Add Tags

After selecting the data sources desired, enter any tags you want to assign by entering them into the field at the bottom and click **Add**. If you want to remove a tag, click on it and press the delete key.

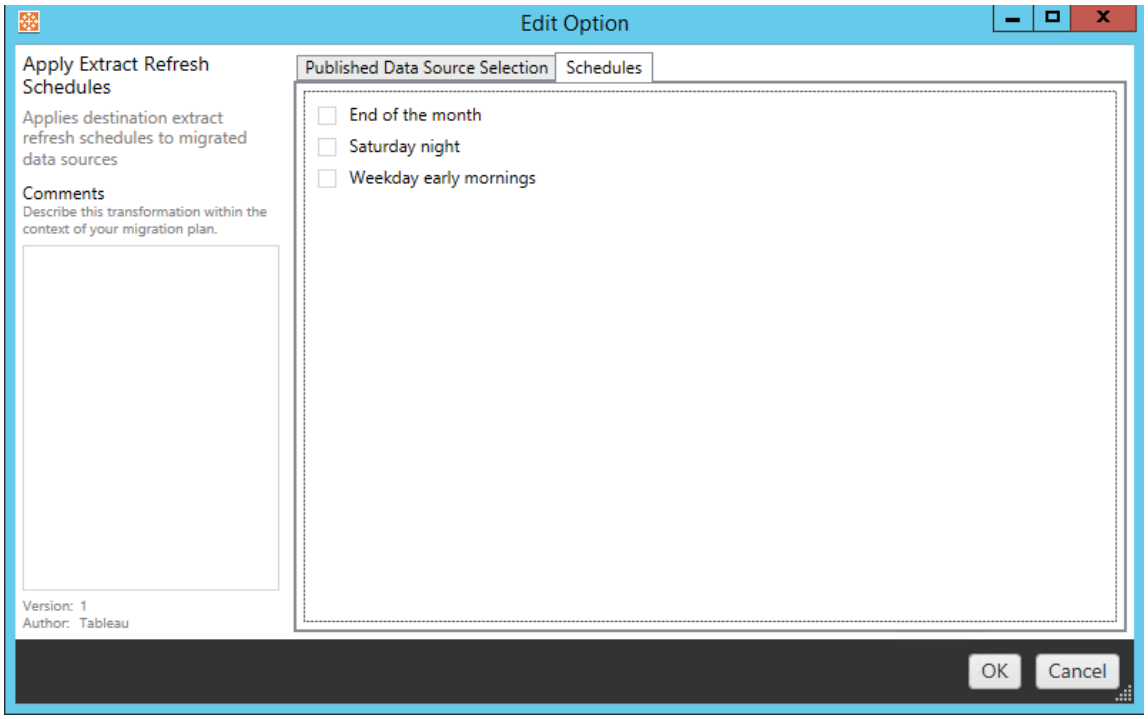


In addition, on each of the transformations you can enter notes in the **Comments** section on the left-hand side of the Edit Transformation window.

Apply Extract Refresh Schedules

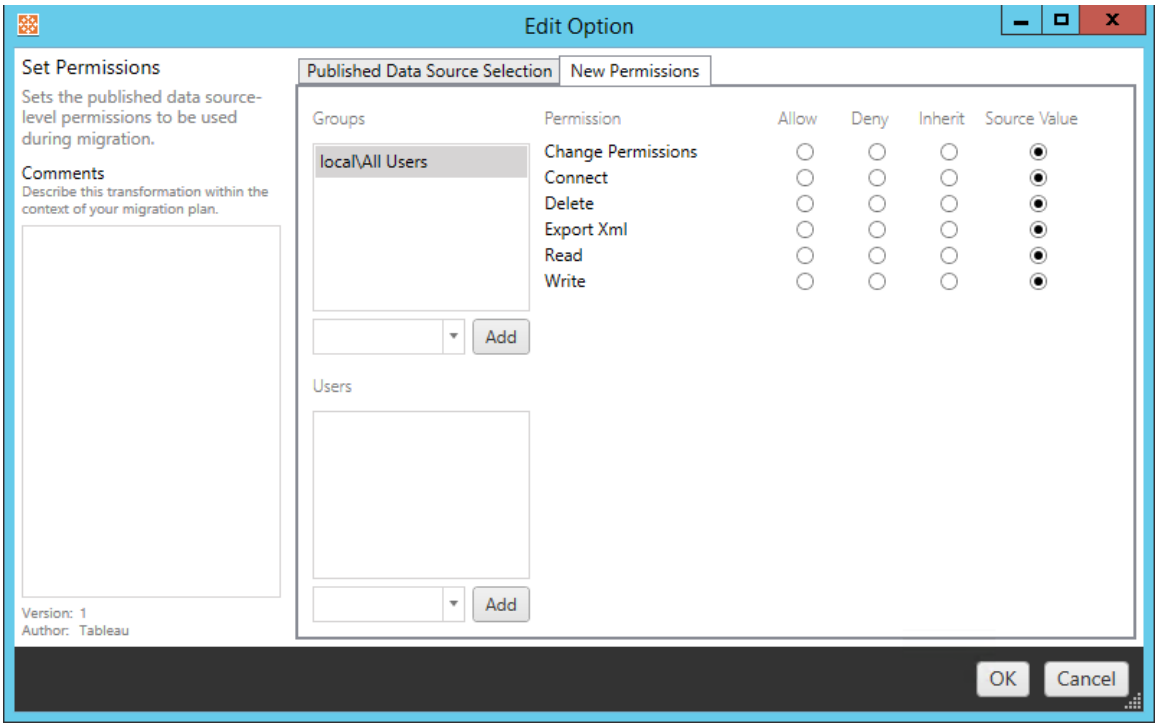
This transformation applies destination extract refresh schedules to migrated data sources. The list of schedules generated are from the destination.

> **Note:** Extract refresh schedules cannot be created in Tableau Cloud. This option is not available if the destination is a Tableau Cloud site. For more information, see Migration Limitations.



Set Permissions

The last type of transformation is to edit the permissions for the selected data sources. Enter in a Group or User and click **Add**. Adjust the permissions as desired. The four different options are to **Allow** the permission, **Deny** the permission, **Inherit**, or to keep the **Source Value**.



Step 5: Continue to the next step

When you are ready, click **Next** to continue to the Migration Plans: Permissions and Ownership section of the planning phase.

Who can do this

Tableau site user with an Explorer role or higher. To migrate content, you must have **View** and **Download/Save a Copy** capabilities for workbooks on the source site and **View** and **Publish** capabilities for target projects on the destination site. For more information, see Permissions.

Migration Plans: Permissions and Ownership

The Content Migration Tool allows you to replicate workbook and data source permissions to Tableau environments in different network domains or have differences in username or group syntax. You can create user permissions mappings to customize and secure content after it has been published to the destination location. Mappings are applied if **Copy Project Permissions**, **Copy Workbook Permissions**, or **Copy Data Source Permissions** have been selected earlier in the planning phase, along with **Apply User Mappings**.

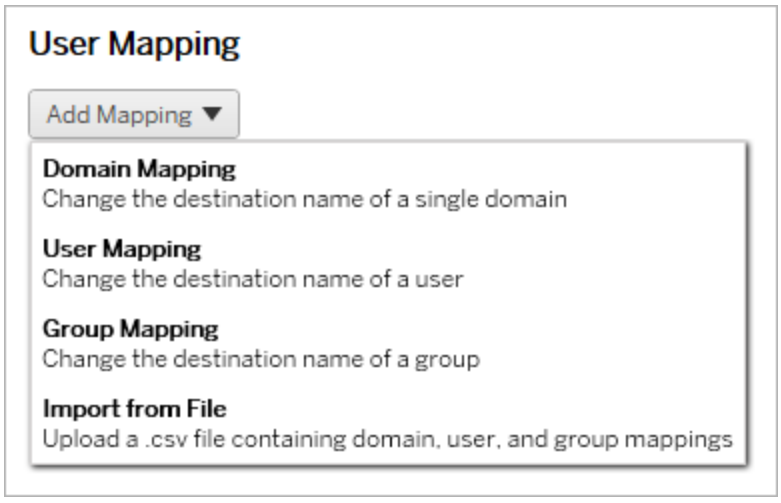
For more information, see Migration Plans: Source Projects, Migration Plans: Workbooks, and Migration Plans: Published Data Sources.

Mapping limitations

- Content Migration Tool will stop the migration process if it fails to find the mapped user or group in the destination location. Subsequent user or group permissions mappings are not checked after the first failure, and the plan must be run again.
- Content Migration Tool cannot replicate permissions if the source content has permissions for multiple users and groups with identical names. This only occurs when there are duplicate user or group names sourced from separate domains.

Step 1: Add mapping

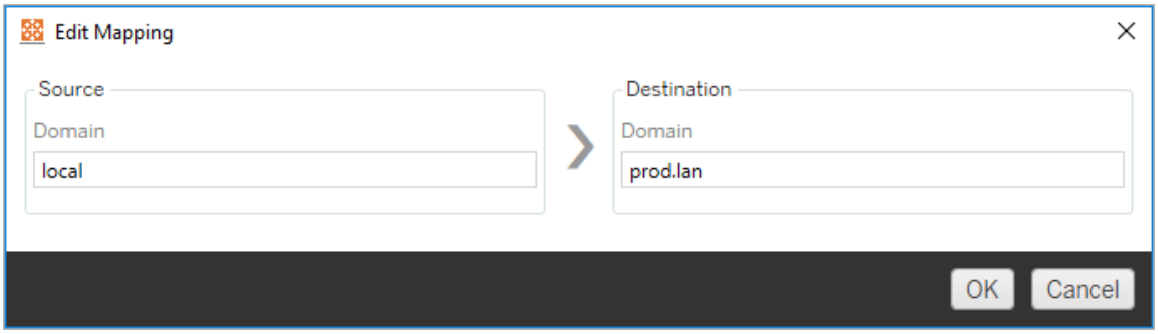
To add user permissions mapping, click **Add Mapping** and select whether to change the name of a domain, user, group or to import mappings from a comma-separated values (CSV) file. If Content Migration Tool is unable to match a permission in the destination location, the source content will not be migrated.



Domain Mapping

Domain permissions mapping applies to all users and groups in the destination location. If you are unsure about the source or destination domain, you can check the user and group

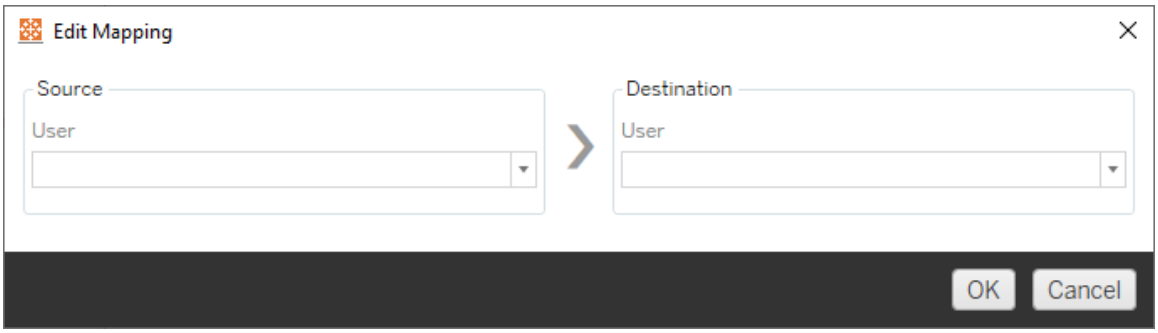
pages on your Tableau site. If local user provisioning has been selected, the domain must be specified as `local`.



User Mapping

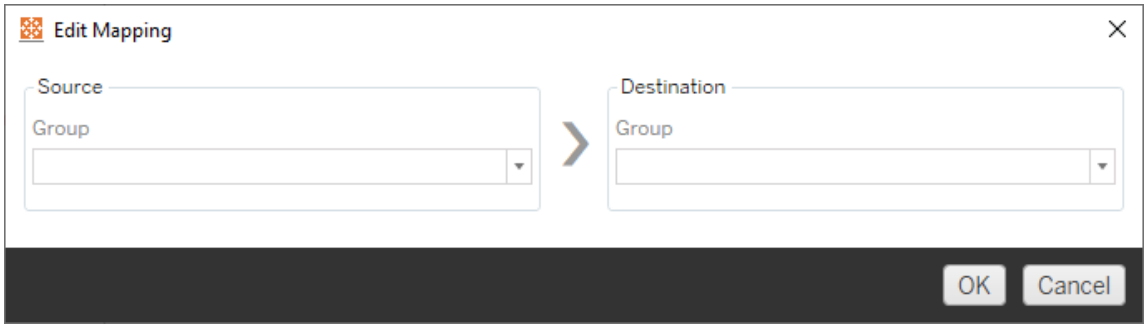
User permissions mapping automatically populates a list of users from the source and destination locations using the syntax `domain\user`. You cannot enter and save the names of users that don't exist.

Note: When migrating between projects on the same site, the destination location and list of users will be the same. You can use mappings to update content ownership from `User_A` to `User_B` on the site.



Group Mapping

Group permissions mapping automatically populates a list of users from the source and destination locations using the syntax `domain\group`. You cannot enter and save the names of groups that don't exist.



Import mappings from a CSV file

Starting in version 2021.4, you can import a CSV file containing domain, user, and group mappings to quickly prepare your data for migration. Importing mappings can reduce the manual steps required to run a migration plan by allowing you to create and edit your mappings in bulk outside of Content Migration Tool. To import mappings, select **Import from File** from the Add Mapping menu.

CSV file format requirements

When you create a CSV file to import mappings, make sure that the file meets the following requirements:

- The file does not include column headings. Tableau assumes that every line represents a mapping.
- The file contains three comma-separated values per row: mapping type, source domain/user/group, and destination domain/user/group.
- Include the domain for user names and groups if the server uses Active Directory authentication or "local" if a local identity store is used.

You must specify "domain," "user," or "group" for mapping type, as shown in the following table. The source and destination columns provide example syntax for Active Directory and a local identity store. Actual values in the CSV file will vary depending on your organization.

Mapping Type	Source	Destination
domain	<domain>	<domain>
user	<domain>\<user name> local\<user name>	<domain>\<user name> local\<user name>
group	<domain>\<group name> local\<group name>	<domain>\<group name> local\<group name>

Import user permissions mappings

To import user permissions mappings in the Content Migration Tool:

1. Click **Add Mapping** and select **Import from File**.
2. In the dialogue window, click **Export CSV** to export a .csv file containing all users and groups from the source site. Edit the resulting file in a text editor to add mappings for the destination site.

   If you already have a mapping file, skip to step 3.

   > **Note:** The exported CSV file doesn't include domains from the source site. Domains must be added manually to the CSV to create domain mappings.

3. Click **Import Mappings** and select the mapping file you want to import.

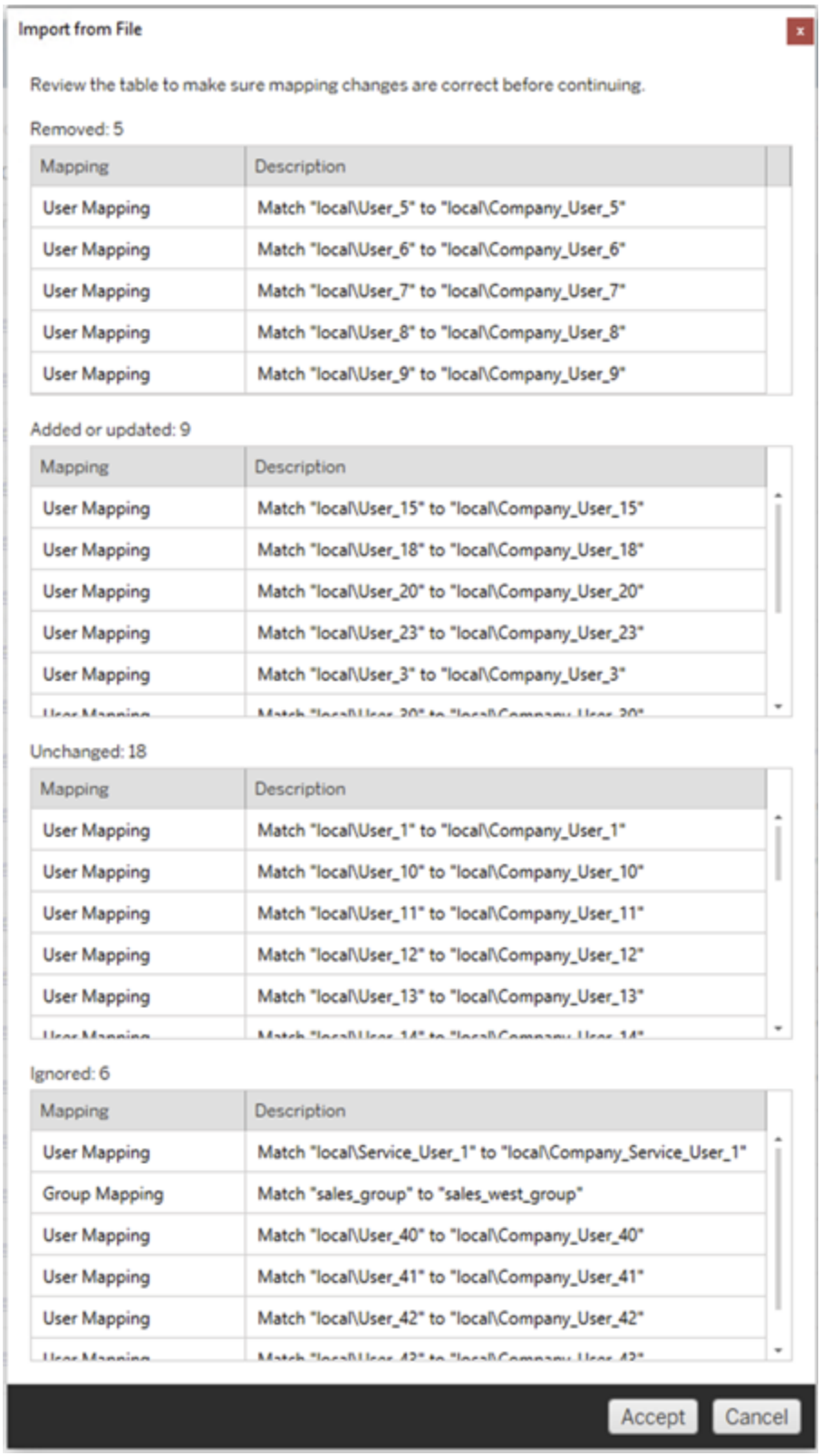
Content Migration Tool will validate the mappings for errors when importing the file. If errors are detected, you must fix each error in the CSV file and then import it again.

CSV import example

The following example shows a CSV file that contains multiple mapping types.

```
user,local\hwilson,companyx.lan\henry.wilson
user,local\jjohnson,companyx.lan\janna.johnson
user,local\mkim,companyx.lan\michele.kim
user,local\fsuzuki,companyx.lan\fred.suzuki
user,local\awang,companyx.lan\alan.wang
user,local\snguyen,companyx.lan\susan.nguyen
user,local\lrodriguez,companyx.lan\laura.rodriguez
user,local\agarcia,companyx.lan\ashley.garcia
group,local\All Users,companyx.lan\All Users
group,local\Finance Team,companyx.lan\Finance Group
domain,dev.mycompany,prod.mycompany
```

A preview window is displayed while importing the CSV that shows mappings removed, added or updated, unchanged, and ignored. Review that the mapping changes are correct and click **Accept**.


Import from File
Review the table to make sure mapping changes are correct before continuing.
Removed: 5
Mapping
Description
User Mapping
Match "local\User_5" to "local\Company_User_5"
User Mapping
Match "local\User_6" to "local\Company_User_6"
User Mapping
Match "local\User_7" to "local\Company_User_7"
User Mapping
Match "local\User_8" to "local\Company_User_8"
User Mapping
Match "local\User_9" to "local\Company_User_9"
Added or updated: 9
Mapping
Description
User Mapping
Match "local\User_15" to "local\Company_User_15"
User Mapping
Match "local\User_18" to "local\Company_User_18"
User Mapping
Match "local\User_20" to "local\Company_User_20"
User Mapping
Match "local\User_23" to "local\Company_User_23"
User Mapping
Match "local\User_3" to "local\Company_User_3"
Unchanged: 18
Mapping
Description
User Mapping
Match "local\User_1" to "local\Company_User_1"
User Mapping
Match "local\User_10" to "local\Company_User_10"
User Mapping
Match "local\User_11" to "local\Company_User_11"
User Mapping
Match "local\User_12" to "local\Company_User_12"
User Mapping
Match "local\User_13" to "local\Company_User_13"
Ignored: 6
Mapping
Description
User Mapping
Match "local\Service_User_1" to "local\Company_Service_User_1"
Group Mapping
Match "sales_group" to "sales_west_group"
User Mapping
Match "local\User_40" to "local\Company_User_40"
User Mapping
Match "local\User_41" to "local\Company_User_41"
User Mapping
Match "local\User_42" to "local\Company_User_42"
Accept
Cancel

Once the mappings are imported successfully, you can edit, delete, or change the mapping order as described in Step 2.

Step 2: Change mapping order

After a permissions mapping is created, you can change the order using the **Up** or **Down** options to determine when it will be handled during the migration. When a domain, user, or group is handled in a permissions mapping, any subsequent permissions mappings for the source domain, user, or group will be ignored.

In the example below, permissions for `User_A` are mapped to `User_B`. Content Migration Tool will ignore the second permissions mapping because `User_A` has already been handled.

	Mapping	Description
Edit Delete Up Down	User Mapping	Match "local\User_A" to "local\User_B"
Edit Delete Up Down	User Mapping	Match "local\User_A" to "local\User_C"

In the example below, the first permissions mapping associates the domain for all users to `prod`. Content Migration Tool will ignore the second permissions mapping because the domain for `User_A` has already been handled.

	Mapping	Description
Edit Delete Up Down	Domain Mapping	Match "local" to "prod"
Edit Delete Up Down	User Mapping	Match "User_A" to "dev\User_B"

Step 3: Continue to next step

When you are ready, click **Next** to continue to the Migration Plans: Migration Scripts section of the planning phase.

Who can do this

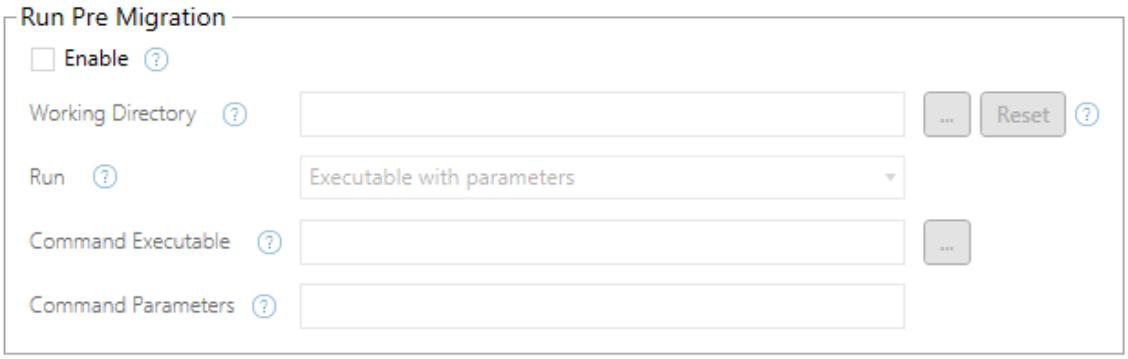
Tableau site user with an Explorer role or higher. To migrate content, you must have **View** and **Download/Save a Copy** capabilities for workbooks on the source site and **View** and **Publish** capabilities for target projects on the destination site. For more information, see Permissions.

Migration Plans: Migration Scripts

The next step of creating a migration plan in the Tableau Content Migration Tool is to create any scripts you want to run with your plan before or after migration.

Step 1: Pre-Migration

The **Run Pre Migration** section of the screen is dedicated to scripts that will run before the migration.



Each field has a help icon you can get information from by moving your cursor over it. To start with your pre-migration scripts, select **Enable**, which will then activate the fields below.

Working Directory

This is the working directory for the script. The default directory is the same folder as the migration plan. Click on the browse button to select a different folder. The **Reset** button will restore the current migration plan folder as the working directory.

Run

This drop down allows you to choose either to run a custom script or an executable with parameters.

Command Executable

If you selected **Executable with Parameters** from the **Run** menu, this field will appear. This is the file path to the command executable to run before migration. Type it in directly or use the browse button to find the executable. This is a required field.

Command Parameters

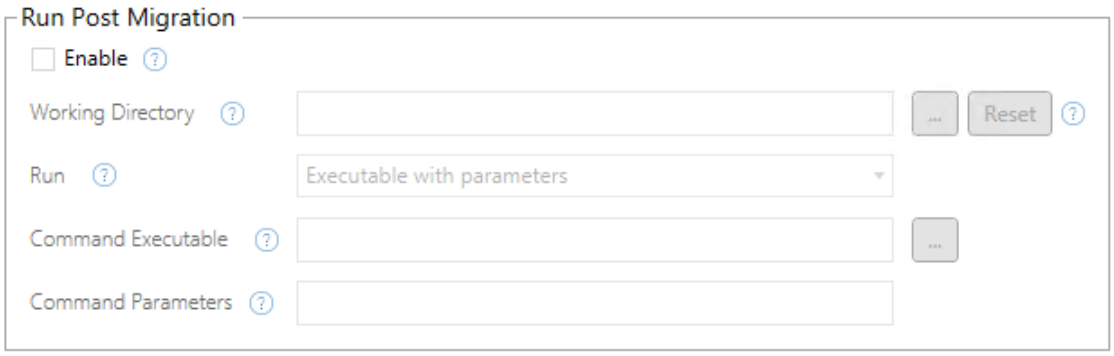
If you selected **Executable with Parameters** from the Run drop-down menu, this field will appear. Enter in command line parameters here to use with the command executable.

Script

If you selected **Custom script** from the **Run** menu, enter in your pre-migration script here. It will be executed as a *.cmd file. This is a required field.

Step 2: Post-Migration

The **Run Post Migration** half of the screen is dedicated to scripts that will run after migration.



Each field has a help icon you can get information from by moving your cursor over it. To start with your post-migration scripts, select **Enable**, which will then activate the fields below.

Working Directory

This is the working directory for the script. The default directory is the same folder as the migration plan. Click on the browse button to select a different folder. The **Reset** button will restore the current migration plan folder as the working directory.

Run

This drop down allows you to choose either to run a custom script or an executable with parameters.

Command Executable

If you selected **Executable with Parameters** from the **Run** menu, this field will appear. This is the file path to the command executable to run before migration. Type it in directly or use the browse button to find the executable. This is a required field.

Command Parameters

If you selected **Executable with Parameters** from the **Run** menu, this field displays. Enter in command line parameters here to use with the command executable.

Script

If you selected **Custom script** from the **Run** menu, enter in your post-migration script here. It will be executed as a *.cmd file. This is a required field.

Step 3: Continue to Next Step

When you are ready, click **Next**.

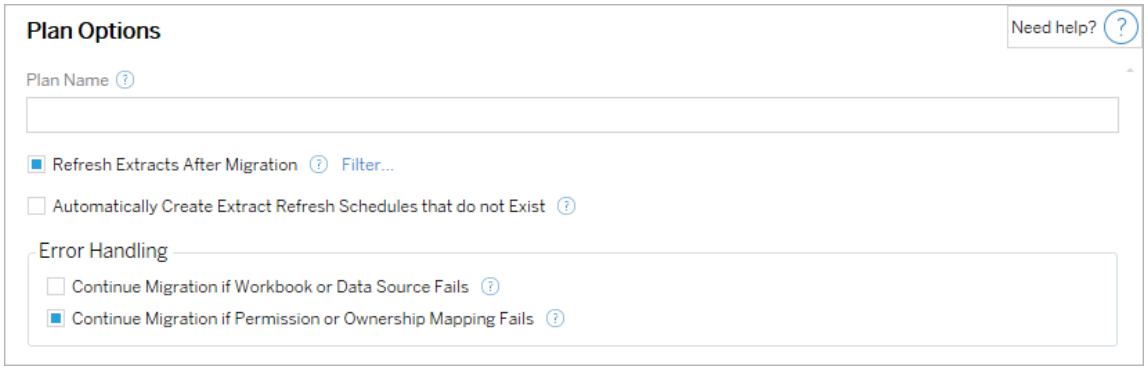
Who can do this

Tableau site user with an Explorer role or higher. To migrate content, you must have **View** and **Download/Save a Copy** capabilities for workbooks on the source site and **View** and **Publish** capabilities for target projects on the destination site. For more information, see Permissions.

Migration Plans: Plan Options

The last step of creating a migration plan in the Tableau Content Migration Tool is configuring the plan options.

Step 1: Configure options



The **Plan Name** is the name of the plan as it will appear in Content Migration Tool. We recommend using a user-friendly name for your plan name.

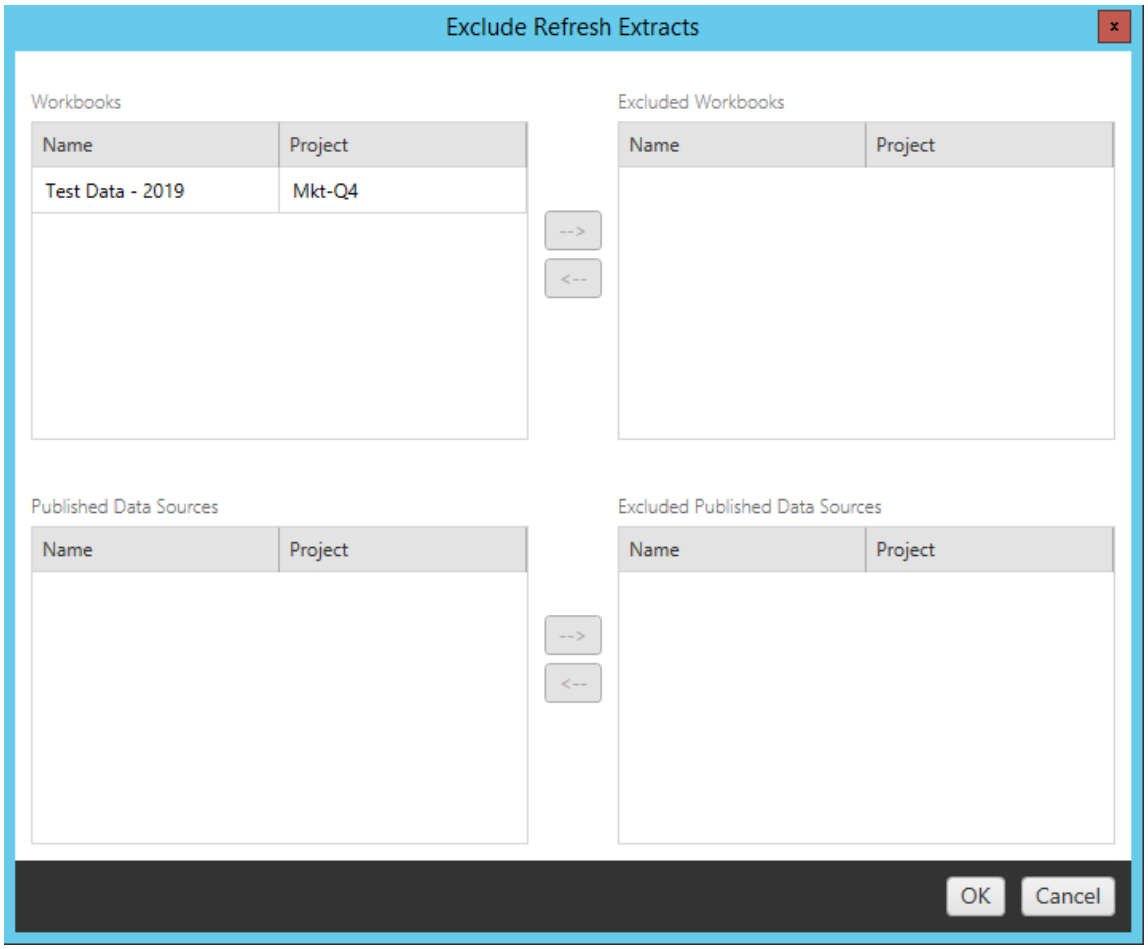
The following are available options:

- **Refresh Extracts After Migration**: If selected, data extracts will be refreshed immediately after migration if Content Migration Tool detects they have been modified during migration. Click the **Filter** link to exclude specific extracts. For more information, see Exclude extract refreshes below.
- **Automatically create Extract Refresh Schedules that do not Exist**: Automatically creates destination extract schedules that do not exist. If not checked, source schedules that do not exist on the destination server will not be copied.

- **Continue Migration if Workbook or Data Source Fails**: If checked, errors migrating a workbook or data source will not cause the migration to stop. The errors will be logged and the migration will continue. Errors during version control will always stop the migration.
- **Continue Migration if Permission or Ownership Mapping Fails**: If checked, errors copying permissions or ownership will not cause the migration to stop. The errors will be logged and the migration will continue.

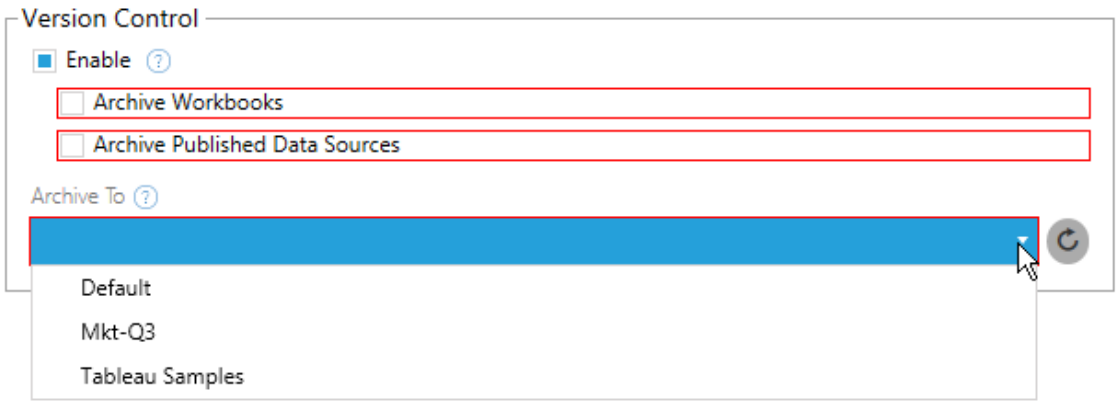
Exclude extract refreshes

By clicking **Filter** next to **Refresh Extracts After Migration**, you can choose the workbooks or published data sources that will not be refreshed automatically. Use the arrow buttons to select the items you want to exclude, and click**OK**.



Step 2: Version control

These options allow you to avoid losing the existing workbooks in the destination site that might be replaced by the migrated workbooks.



Select **Enable** to save previous versions of your content. You can choose to archive workbooks and/or published data sources. Once version control is enabled, you must select a project from the **Archive To** menu, which lists all of the projects in your destination site. We recommend creating a separate archive project to store your versioned content. Click the refresh button to display any projects that have been added or modified on the site.

Step 3: Save plan

Once you have selected your plan options, click **Save Plan** to save your plan for future use. The plan will be saved to the `Documents\Tableau Content Migration Tool Plans` folder on your local machine.

Step 4: Continue to next step

When you are ready, click **Verify & Run** to end the Planning phase and prepare to run your plan.

Who can do this

Tableau site user with an Explorer role or higher. To migrate content, you must have **View** and **Download/Save a Copy** capabilities for workbooks on the source site and **View** and **Publish** capabilities for target projects on the destination site. For more information, see Permissions.

Tableau Server users can publish extracts which are copies, or subsets of the original data. These extracts may be embedded in a workbook or a data source. By default, when you use the Tableau Content Migration Tool to migrate a workbook or data source that contains an extract, that extract is migrated along with the workbook or data source that contains it. The Content Migration Tool gives you a couple options for controlling this behavior:

- **Switching to a Live Connection**

  You can add the **Remove Extract** transformation to your migration plan to remove the extract from your workbook or data source during migration. As always, the source workbook or data source will not be modified. The copy of the workbook or data source migrated to the destination site will have the extract removed from it. This effectively switches the data connection back to a live connection.

- **Refreshing Extracts after Migration**

  You can enable the **Refresh Extracts After Migration** option in your migration plan to have an immediate extract refresh task scheduled after the workbook or data source is migrated.

  We don't recommend using the **Refresh Extracts After Migration** option if your migration plan also uses the **Set Connection Info** transformation to change the data connection to point to a different set of data (for example, a different database server or database). When you change the connection information to point to different data and use the **Refresh Extracts After Migration** option, this can unintentionally expose data in a way that is a potential security issue.

  For more information, see Option 3: Refresh Extracts After Migration.

Changing data connections that use extracts

Tableau data connections are either live connections that directly query a data source, or they are extracts of a data source. Extracts are copies or subsets of the original data and can be

embedded in a workbook or data source. When present, the views will query data from the extract instead of the underlying data source.

Commonly, you'll want to modify the data source connection during the migration so that it points to a different database on the destination site than it did on the source site.

For example, if you are migrating a workbook from your staging site to your production site, you will likely want to update the data connections inside the workbook to connect to your production database. You can implement this by using the **Set Connection Info** transformation in your migration plan. Now you have a migration plan which copies a workbook from staging to production and updates the data connections to point to the production database.

If your workbook uses an extract, additional work is required. In this scenario, the workbook will be migrated and the live data connection updated. However, the views will still show data from the staging database since it still contains the staging database extract - copied from the source (staging) site. There are a few ways to address this.

Option 1: Use Published Data Sources

You can change your workbooks so that they use published data sources instead. This way, the extract will be managed as part of the published data source and migrating updates to the workbooks that use that data source can be simplified by not having to worry about the connection to the live database or the data extract.

Option 2: Remove the Extract During Migration

You can add a **Remove Extract** transformation to your migration plan. This will remove the extract from your workbook, effectively switching the data source to a live connection.

Option 3: Refresh the Extract After Migration

You can use the **Refresh Extracts After Migration** option in your migration plan. This will migrate the extract along with the workbook but will schedule an immediate extract refresh task for that workbook after the migration is complete.

This option is usually not recommended when used in combination with a **Set Connection Info** transformation because of potential security issues that it can introduce.

The issue is that the migrated workbook on your destination site will still show the old (source) extract data for the period between the completion of migration and the completion of the extract refresh task. If the extract refresh task fails, then the old/source extract data will remain until the extract is refreshed.

In a scenario like we've outlined above, migrating from a staging to production environment, this may be acceptable but you should be aware that the users of your workbooks may not be aware that the workbook is showing old/staging data due it being recently migrated and the extract not being refreshed yet.

In other scenarios where you may be using **Set Connection Info** to change data connections to point to a different set of customer or client data, this could introduce serious security issues where the workbook's extract contains data from a different client or customer until the extract has been refreshed post-migration.

One way to mitigate this issue is to implement a 2-stage migration. This approach requires you to create two migration plans, one for each step described below and ensures the workbooks and data sources have an up-t-o-date extract before they are accessible.

- **Stage 1:** Migrate your content to a project on your destination site that only administrators have access to. This migration allows you to use the **Refresh the Extract After Migration** option along with the **Set Connection Info** transformation to update the data connection, because no unauthorized users will have an opportunity to see the old data, even if the extract refresh fails.
- **Stage 2:** After stage 1 is complete and you confirm there is a successful extract refresh, run a second migration plan to migrate the content from the stage 1 destination to the final destination where it is visible to end-users.

Who can do this

Tableau site user with an Explorer role or higher. To migrate content, you must have **View** and **Download/Save a Copy** capabilities for workbooks on the source site and **View** and **Publish** capabilities for target projects on the destination site. For more information, see Permissions.

Migrate Workbooks and Data Sources with Embedded Credentials

Starting in version 2023.1, authorized users can migrate workbooks and published data sources with embedded credentials from Tableau Server to Tableau Cloud. Additional configuration is required before migrating with Content Migration Tool.

> **Note:** Content Migration Tool does not support embedded credential migration for OAuth connections. For more information, see Migration Limitations.

Overview

Migrating embedded credentials using Content Migration Tool (CMT) is available when connecting to Tableau Server as the source site and Tableau Cloud as the destination site. Both sites must have an Advanced Management license.

Now that we've covered the requirements, let's discuss how migration works. You'll need to work closely with the Tableau Cloud site administrator and TSM administrator (sometimes the same person) to allow the feature and authorize a site user. After the feature is activated, the authorized site user builds a migration plan and selects the publish options *Migrate Embedded Credentials for Workbooks* and *Migrate Embedded Credentials for Data Sources*.

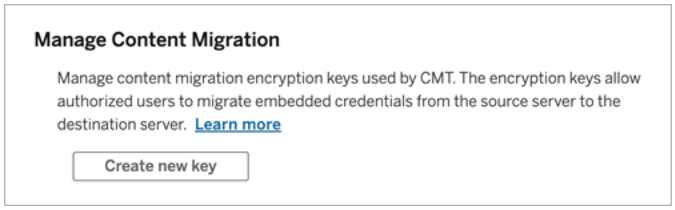
When running the migration plan, all required content credentials are transmitted in an encrypted content manifest from Tableau Server to Tableau Cloud. As CMT publishes content, the destination Tableau Cloud site embeds matched credentials securely from the manifest into the content (workbooks or published data sources). Any problems that occur during migration will appear in the Errors and Warnings tab in CMT. For more information, see Migration Plan Overview.

Allow embedded credential migration

Use the following steps to allow embedded credential migration from Tableau Server to Tableau Cloud.

Tableau Cloud

1. Open a browser window and sign in to Tableau Cloud as a site administrator.
2. Select **Settings** > **General**, and scroll down to **Manage Content Migration**.



3. Click **Create new key** to generate an encryption key pair.

> **Note:** The public key is only displayed once. If you lose the key before completing the configuration, you'll need to generate a new key.

4. In the resulting window, click **Copy to clipboard** and then close the window.
5. Paste the public key to a file and store it in a safe location. The TSM administrator will use the public key to allow migration. You can view the public key expiration date on the Settings page.

TSM Command Line Interface

1. Open a command prompt with an account that is a member of the `tsmadmin` group on a node in the cluster.

2. Use `tsm security authorize-credential-migration` to allow embedded credential migration to the Tableau Cloud site. For more information, see tsm security.

   ```
   tsm security authorize-credential-migration --source-site-url-namespace <Tableau Server site ID> --destination-site-url-namespace <Tableau Cloud site ID> --destination-server-url <Tableau Cloud site url> --authorized-migration-runner <username> --destination-public-encryption-key <public key>
   ```

   Note: When running TSM commands from a remote node, use `tsm login` to authenticate a session with the Tableau Server Administration Controller service before running `tsm security authorize-credential-migration`.

3. (Optional) Use `tsm security cancel-credential-migrations` to cancel granted authorizations. By default, migration authorization will expire in 7 days or the number of days specified with the `--expiration-time-in-days` option.

Content Migration Tool

1. Open Content Migration Tool and select **Create New Plan** or **Browse for a Plan**.
2. On the Sites page, click **Sign in to Tableau**, and connect to Tableau Server as the source and Tableau Cloud as the destination. Embedded credential migration is only available when migrating from Tableau Server to Tableau Cloud.
3. Build your migration plan and select the following Publish Options:
   - On the Workbook Publish Options page, select **Migrate Embedded Credentials for Workbooks**. For more information, see Migration Plans: Workbooks.
   - On the Data Source Publish Options page, select **Migrate Embedded Credentials for Data Sources**. For more information, see Migration Plans: Published Data Sources.
4. When you are ready, click **Verify & Run** to start the migration.

The workbooks and published data sources you selected are migrated to your Tableau Cloud site and should not prompt for authentication. If you experience issues while migrating embedded credentials, see Troubleshooting.

Troubleshooting

This section includes some common migration issues you might encounter and suggestions to resolve them.

There is no option to migrate embedded credentials

You can only migrate embedded credentials from a Tableau Server to a Tableau Cloud site. Tableau Server and Content Migration Tool must be running versions 2023.1 or later. For more information, see Install Tableau Content Migration Tool.

Migrating embedded credentials failed

In the Errors and Warnings tab of CMT, you may receive an error indicating that migrating the embedded credentials failed. This can occur when the public key used to authorize migration has expired.

As a Tableau Cloud site administrator, go to the Settings page and verify that the public key is valid. You'll have to create a new encryption pair to authorize the migration if the public key expires. For more information, see Allow embedded credential migration.

Who can do this?

- Tableau Cloud site administrator and TSM administrator are required to allow embedded credential migration.
- The authorized site user must have an Explorer role or higher. They must also have View and Download/Save a Copy capabilities for workbooks on the source site and View and Publish capabilities for target projects on the destination site.

For more information, see Permissions.

Using the Tableau Content Migration Tool Console Runner

The Tableau Content Migration Tool includes a command-line utility for running migrations, `tabcmt-runner.exe`, located in the installation folder. The default installation folder is `%PROGRAMFILES%\Tableau\Tableau Content Migration Tool`.

Note: The `tabcmt-runner.exe` utility is not the same as the `tabcmt.cmd` command line utility which is used to configure the Content Migration Tool graphical application. For more information about `tabcmt.cmd`, see Using the Tableau Content Migration Tool Command Line Interface.

Usage:

- tabcmt-runner [options] <plan_file.tcmx>
- tabcmt-runner license --remove
- tabcmt-runner license <new license key>
- tabcmt-runner license <license file path> [--passphrase=<license file passphrase>]
- tabcmt-runner encryption --reset
- tabcmt-runner encryption <new_key>
- tabcmt-runner improvement [on|off]
- tabcmt-runner --help
- tabcmt-runner --version
- tabcmt-runner script-warning [on|off]

Options:

- --version
- --help
- --quiet
- --info
- --logfile=VALUE
- --src-user=VALUE
- --src-password=VALUE
- --dest-user=VALUE
- --dest-password=VALUE

- --https=VALUE
- --allow-scripts

Run Plan

Executes a migration plan immediately.

```
tabcmt-runner [options] <plan file>
```

Available options:

- `--logfile=<file name>` sets the file name to log output to
- `--https=<secure|legacy>` sets the HTTPS mode
- `--quiet` disables logging to stdout
- `--src-user=<username>` sets the username of the source connection
- `--src-password=<password>` sets the password of the source connection
- `--dest-user=<username>` sets the username of the destination connection
- `--dest-password=<password>` sets the password of the destination connection

Exit codes:

- 0 indicates that the migration was successful.
- 1 indicates that the migration was successful but warning messages were logged.
- 2 indicates that the migration failed. Specific errors will be included in the log output.

Show Plan Summary

Shows a summary of the migration plan and then exits.

```
tabcmt-runner --info <plan file>
```

help

Shows usage information for the command line utility.

```
tabcmt-runner --help
```

version

Shows the current application version information.

```
tabcmt-runner --version
```

encryption

Reset the encryption key, or specify a new one. You must specify the encryption key before using the `tabcmt-runner` utility, even if you already done so from the Content Migration Tool UI.

```
tabcmt-runner encryption <new_key> | --reset
```

improvement

Default value: `on`

Enables or disables collection of anonymous usage information by the application. This information is completely anonymous and is sent periodically to Tableau to help us improve Content Migration Tool.

Examples

Show whether the improvement program is enabled or disabled:

```
tabcmt-runner improvement
```

Enable or disable the improvement program:

```
tabcmt-runner improvement <on|off>
```

license

Deprecated in July 2022.

This command is only applicable for legacy licenses. Manages a legacy application license for the current user. When using a legacy key, to use the `tabcmt-runner` utility you must activate the license using this command, even if you already activated it from the Content Migration Tool UI.

Examples

Show the current license information:

```
tabcmt-runner license
```

Set/activate a serial key or offline license key:

```
tabcmt-runner license <key>
```

Remove/deactivate the current license:

```
tabcmt-runner license --remove
```

Set/activate using a license file:

```
tabcmt-runner license <file path> [--passphrase=<password>]
```

script-warning

Default value: `on`

Shows a warning message when running a migration plan that includes migration scripts.

> **Note:** This command updates your selection on the Settings page. For more information, see Tableau Content Migration Tool Settings.

Examples

Show if script warning is turned on or off.

```
tabcmd-runner script-warning
```

Turn script warning on or off

```
tabcmd-runner script-warning <on|off>
```

If turned on, you must include the option `--allow-scripts` to execute migration plans.

```
tabcmd-runner --allow-scripts <plan file>
```

Who can do this

To use the console runner, you must have all the following:

- Administrator permissions on the Content Migration Tool machine.
- Tableau site user account with an Explorer role or higher.

- View and Download Workbook/Save a Copy permissions on the source site.
- Publishing rights for the destination site.

Example: Scripting Migration Plans

> **Note:** This topic includes a sample script you can use as the basis for scripting a multi-plan migration that satisfies your needs and environment. This script is intended to be used as a sample only, and not to be run as-is. For detailed instructions on using the console runner, see Using the Tableau Content Migration Tool Console Runner.

Tableau Content Migration Tool command line utility for running migrations can be used to automate the running of a migration plan from an external scheduler (such as Windows Task Scheduler) or from a custom script. The console runner only runs one migration plan (stored in a .edt file) at a time. If you have a group of migration plans you want to run as a group, then you can use a custom script in combination with the Content Migration Tool console runner.

The example below is written in PowerShell and uses the console runner to execute a list of migration plans as a group.

The following example code demonstrates:

- Running multiple migration plans as a group using the console runner.
- Optionally halting deployment of the group of plans immediately when any single migration in the group fails.
- Using the console runner's exit code to determine whether the migration failed or logged warnings.

```
# List of migration plans to execute as a group.
$planFiles = @(
       'customer 1.tcmx',
       'customer 2.tcmx'
)

# True of false whether to continue with the next plan if a
```

```
migration fails.
$continueOnFailure = $false

# Path to the CMT console runner executable
$runnerExe = 'C:\Program Files (x86)\Tableau\Tableau Content Migra-
tion Tool\tabcmt-runner.exe'

# Store the exit code from the previously run migration plan.
$lastResult = -1

# Loop through and run each migration plan one at a time.
$planFiles | % {
       $file = $_

       if ($lastResult -ge 2 -and -not($continueOnFailure)) {
               Write-Warning "Skipping plan because previous migration failed.
`nSkipped plan: $file"
               return
       }

       Write-Verbose "Running migration plan: $file"
       & $runnerExe $file
       $lastResult = $LASTEXITCODE

       if ($lastResult -ge 2) {
               Write-Error "Migration failed. See output or log file for error
details.`nPlan: $file" -ErrorAction 'Continue'
       }
       elseif ($lastResult -eq 1) {
               Write-Warning "Migration completed with warnings. See output or
log file for warning details.`nPlan: $file"
       }
}
```

Who can do this

To script migration plans, you must have all the following:

- Administrator permissions on the Content Migration Tool machine.
- Tableau site user account with an Explorer role or higher.
- View and Download Workbook/Save a Copy permissions on the source site.
- Publishing rights for the destination site.

Using the Tableau Content Migration Tool Command Line Interface

The Tableau Content Migration Tool includes a command line interface, `tabcmt.cmd`, located in the installation folder. The default installation folder is `%PROGRAMFILES%\Tableau\Tableau Content Migration Tool` (32-bit Windows) or `%PROGRAMFILES(x86)%\Tableau\Tableau Content Migration Tool` (64-bit Windows).

> **Note:** The `tabcmt.cmd` utility is not the same as the Content Migration Tool console runner, `tabcmt-runner.exe`. The console runner is a separate command line utility used for running migrations from the command line. For information on using the Content Migration Tool console runner, see Using the Tableau Content Migration Tool Console Runner.

Here are the commands that can be used with the `tabcmt` command line:

- migrate
- help
- update
- version

migrate

Opens a migration plan file to the migrate step in the GUI:

```
tabcmt migrate <plan file>
```

help

Shows general help about the command line interface and the available commands.

Examples

Show all commands available:

```
tabcmt help
```

Show help and usage information for a specific command:

```
tabcmt help <command>
```

license

Deprecated in July 2022.

This command is only applicable for legacy licenses. Manages the application license for the current user.

Examples

Show the current license information:

```
tabcmt license
```

Remove/deactivate the current license:

```
edt license remove
```

Set/activate a serial key or offline license key:

```
tabcmt license <key>
```

Set/activate using a license file:

```
tabcmt license <file path> [--passphrase=<password>]
```

update

Manages the options for application updates.

Examples

Show the current update settings:

```
tabcmt update
```

Enable or disable the automatic update notifications:

```
tabcmt update --disabled=<true|false>
```

Set the URL to detect/download updates from:

```
tabcmt update --url=<url>
```

Enable or disable showing beta updates. Set to false to only show stable release updates.

```
tabcmt update --beta=<true|false>
```

version

Shows the current application version information.

```
tabcmt version
```

Who can do this

To use the command line interface, you must have all the following:

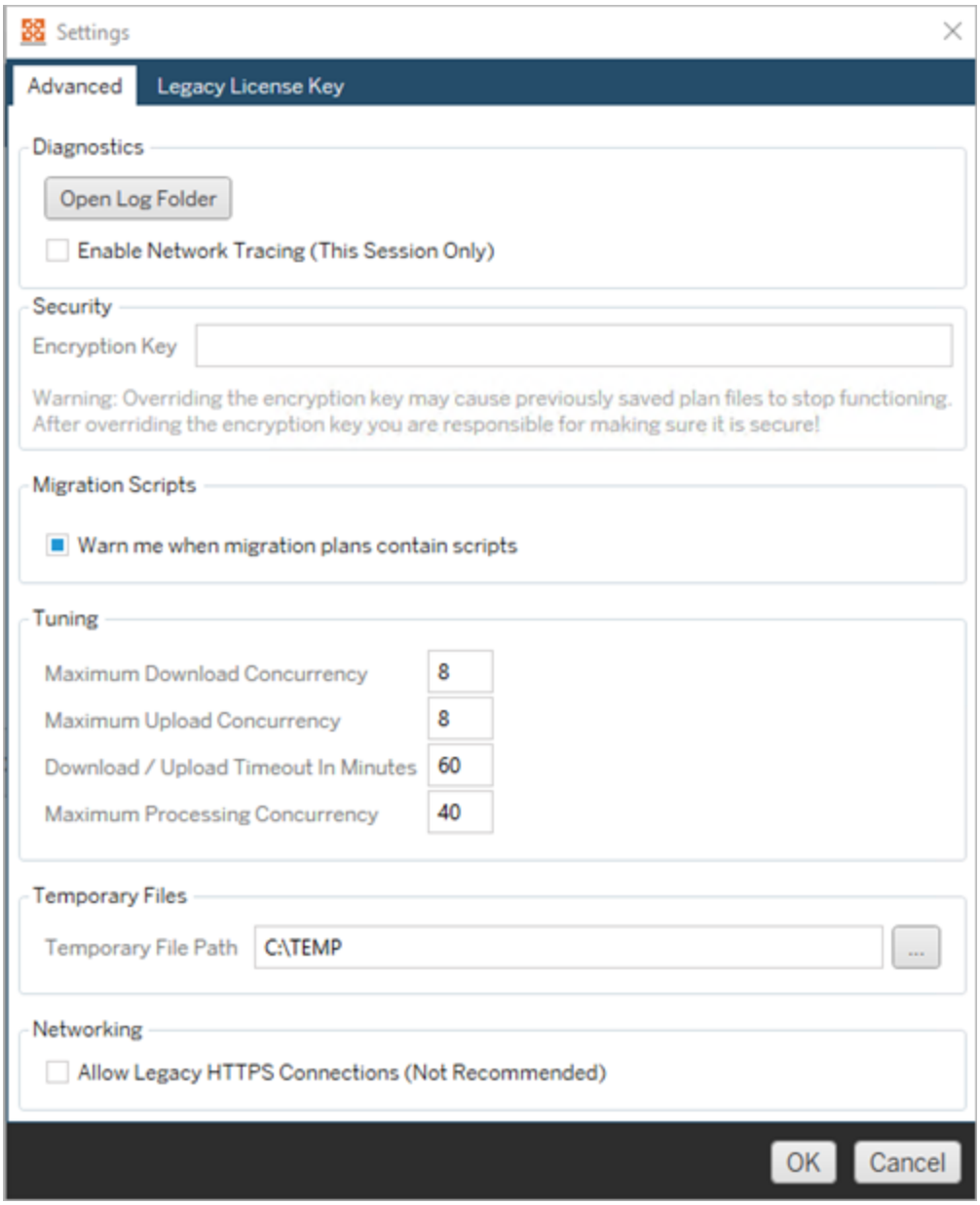
- Administrator permissions on the Content Migration Tool machine.
- Tableau site user account with an Explorer role or higher.
- View and Download Workbook/Save a Copy permissions on the source site.
- Publishing rights for the destination site.

Tableau Content Migration Tool Settings

The Tableau Content Migration Tool default settings work in most cases, but you can change these if you need to, or if you are working with Tableau Support and they ask you to make changes.

To view or update the Content Migration Tool settings:

1. Open Content Migration Tool.
2. Click **Help** > **Settings**. The Settings dialog opens:



Diagnostics—Click **Open Log Folder** to open the logs location. Here you can view the logs, and zip them up if you need to send them to Tableau. For more information, see Tableau Content Migration Tool Log Files.

Select **Enable Network Tracing** if you are working with Support and they ask you to include a network trace in the logs. This applies until you clear the option or restart the Content Migration Tool.

Security—The encryption key is automatically generated on installation. If you change the encryption key, any migration plans with embedded passwords that were created with the previous key cannot be opened. If you have multiple installations of Tableau Content Migration Tool and want to share migration plans, you need to make sure the encryption key used by each instance of the tool is the same.

Migration Scripts—By default, a warning is displayed when running a migration plan that includes migration scripts or executables. Other users can edit these files, so verify that they're safe before running the migration. Toggling this setting on and off will also update your warning preference for the console runner. For more information, see Using the Tableau Content Migration Tool Console Runner.

Tuning—In almost all cases you can leave these set to the defaults. If you are working with Support, they may ask you to change these settings.

Temporary Files—Select a location for temporary files if you want to change the default. This is the location where content is copied during a migration. You may want to change this if the default location does not have enough space to temporarily hold migrated content.

Networking—Selecting **Allow Legacy HTTPS Connections** gives you the ability to connect to Tableau Server installations running with older HTTPS configurations (for example, SSL v3). This is not recommended.

Who can do this

Typically, the tasks listed above can only be done by a user with Administrator access on the machine where Content Migration Tool is installed.

Tableau Content Migration Tool Log Files

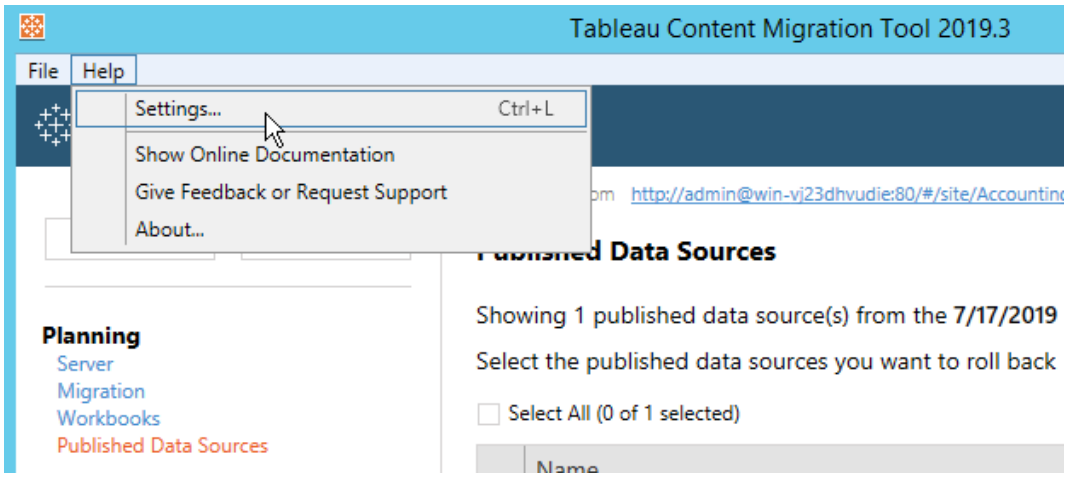
Tableau Content Migration Tool generates log files when you run migrations. These can be helpful for troubleshooting problems.

> **Note:** For information on all the Content Migration Tool settings, see Tableau Content Migration Tool Settings.

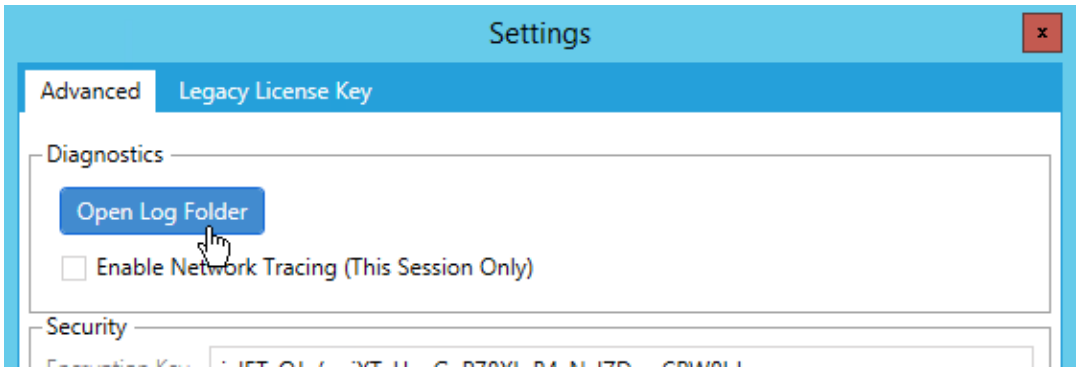
Content Migration Tool Log File Location

To find the Content Migration Tool log files from within the Content Migration Tool:

1. Start Content Migration Tool.
2. Click **Help** and **Settings**:



3. In the **Settings** dialog, click **Open Log Folder**:



A window opens with the log files.

If you are working with Tableau Support and they ask you to send log files, zip the files up before you send them. For more information on sending log files to Tableau, see the Tableau Knowledge Base.

Who can do this

Typically, the tasks listed above can only be done by a user with Administrator access on the machine.

Activity Log

If you have Tableau Server with Advanced Management, activity log files are automatically recorded for detailed analysis and auditing. These logs are stored in the `backgrounder`, `dataserver`, `vizportal`, and `vizqlserver` folders on the local hard drive, using the default path `/var/opt/tableau/tableau_server/data/tabsvc/logs/`. Within each directory, find files named `<process name>-cepp-canonical-events_*.log` that record events and operations.

With the Activity Log, you can:

- View detailed event data for Tableau Server.
- Capture compliance information and keep track of who is doing what on your Tableau sites.

- Audit permissions changes including:
    - Adding or removing users from a group.
    - Moving a piece of content from one project to another.
    - Explicitly changing the permissions on a piece of content.

  You can track permissions change events are essential for implementing a robust book of controls on your Tableau environment. These controls are useful for compliance use cases.

  Supplement the information provided by Admin Insights and Admin Views to track site activity and usage metrics.

  All events include a timestamp and the ID of the actor that performed the event. If relevant, the ID of the piece of affected content is included with the event.

  You can use tools like Splunk or Amazon Cloudwatch to examine the Activity Log. You can use these tools to query log fields and answer questions like:

    - What were the 10 actions last taken by a particular user.
    - Who last performed an event on a piece of content.
    - What was the last action taken on a piece of content.

Audit Permissions Using the Activity Log

Permission auditing allows system administrators to monitor which users have modified access controls to Tableau content. There are two ways to modify access control: *explicit* changes (by changing permission capabilities on a project or content item) and *effective* changes (by changing user site roles, group membership, moving content, and so on). All of these changes are recorded, so administrators can certify that security and access controls are maintained.

For more information about how permission rules are evaluated, see Effective permissions.

Log format

Every action that modifies user or group access to content will get a log entry. Each log entry is structured in a JSON format, with specific keys representing different pieces of information. A log entry contains two parts:

- Metadata: Contains information about when and where an action occurred and what user performed the action.
- Action: Contains information about what piece of content had its permissions changed, what capabilities were changed, and to what values the capabilities were changed.

> **Note:** Activity Log records changes made through the Permissions Dialog UI and REST API. For more information about API methods, see Permissions Methods.

The Activity Log entries are not formatted, and the keys are not sorted in any particular order in the logs. When auditing permissions, you can combine Activity Log data with other data sources to link IDs to names and make the events easier to interpret.

Example

The following is an example log entry showing a group was allowed to connect to a data source.

```
{
event: {
        actorUserId: 39872
        actorUserLuid: "4e6b42bf-9040-4e60-b326-1c56a4fb96f8"
        authorizableType: "DATASOURCE"
        capabilityId: 32
        capabilityValue: "connect"
        contentId: 2099835
        contentName: "Superstore ExtractNeal3"
        eventTime: "2023-01-31T22:44:23.650058Z"
        granteeId: 22
        granteeLuid: "dae0717a-d524-436d-b469-fadeaa22a5dd"
```

```
        granteeType: "Group"
        granteeValue: "GROUP_ALLOW"
        initiatingUserId: 39872
        initiatingUserLuid: "4e6b42bf-9040-4e60-b326-1c56a4fb96f8"
        isError: false
        metadata: {
                applicableToOnline: true
                applicableToServer: true
                comment: "Update Permissions"
                customerAccessible: true
                eventCategory: "security"
                eventType: "update_permissions"
                eventVersion: "1.0"
                internalAccessible: false
        }
        permissionType: explicit"
        siteLuid: "b45e272d-10c7-49d5-9037-e53ce47dbf4e"
}
traceUuid: "3a108a2f-c0ac-4ac7-a5f8-29zf7e064ae1"
}
```


The log entry captures essential information regarding the event, including:

- `eventType` shows an update permissions event occurred
- `permissionType` shows an explicit change to permissions
- `contentId` shows the ID of the content that was modified
- `authorizableType` shows the content type, in this case, a data source
- `capabilityValue` shows the capability that was changed
- `granteeId` shows the grantee that was affected
- `actorUserId` shows the ID of the user who performed the change
- `eventTime` shows the date and time of the change

Events

Log entries contain various event types for permissions changes, such as `content_owner_change` when the content owner changes or `delete_permissions` when an explicit permission rule is deleted on content. For more information about event types, attributes, and when they're recorded, see Activity Log Event Type Reference.

Activity Log Event Type Reference

The following tables describe the Activity Log event types and attributes.

Event type details

The following content describes each event type in Activity Log. Use the alphabetically sorted list of event types on the right, or **ctrl/cmd-f** to go directly to keywords you have in mind.

> **Note:** Timestamps for events are recorded in ISO 8601 UTC.

Common attributes

The following table contains common attributes for all Activity Log events. For event-specific attributes, review the individual event tables.

Attribute Name	Type	Description
actorUserId	integer	ID of the user who performed the action that initiated the event
actorUserLuid	string	LUID of the user who performed the action that initiated the event
eventTime	string	Timestamp when the event occurred
initiatingUserId	integer	ID of the initiating user. For impersonation, it's the ID of the administrative user who initiated impersonation. For standard login, the value is the same as `userId`.

initiatingUserLuid	string	LUID of the initiating user. For impersonation, it's the LUID of the administrative user who initiated impersonation. For standard login, the value is the same as `userLuid`.
licensingRoleName	string	Name of the user's licensing role when the event occurred
serviceName	string	Name of service that initiated the event, such as vizportal, vizqlserver, or sitesaml.
siteLuid	string	LUID of the Tableau site where the event occurred
siteRoleId	integer	The user's site role ID. The value 0 = SiteAdministratorExplorer, 1 = SupportUser, 2 = ExplorerCanPublish, 3 = Explorer, 7 = Guest, 8 = Unlicensed, 9 = Viewer, 10 = Creator, and 11 = SiteAdministratorCreator.
systemAdminLevel	integer	Indicates if the user is a system administrator. The value 10 = System Admin and 0 = Not a system admin.

add_delete_user_to_group

The `add_delete_user_to_group` event is logged when a user is added or removed from a group.

Attribute Name	Type	Description
groupId	integer	The ID of the group
groupLuid	string	The LUID of the group
groupOperation	string	Group operation, either add or delete user to a group
isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error
userId	integer	The ID of the user

userLuid	string	The LUID of the user

background_job

The `background_job` event logs information about jobs run as background tasks. For each job, an event is created to record its various states, including initiation time, queueing, start time, and success or failure.

Attribute Name	Type	Description
args	string	Arguments of the job
duration	long	Duration of the job
eventInitiatedTime	string	Start time of the job
eventState	string	State of the job
isRunNow	bool	Indicates whether the job was initiated manually, by clicking the “Run Now” option on the site or using REST API, or if it was triggered by a schedule. **Note:** Starting in April 2024, jobs triggered by a schedule (`False`) include data for all attributes listed in the table. Attributes for jobs initiated manually (`True`) are under active development, and tentatively scheduled for inclusion in a future release.
jobId	integer	ID of the job
jobLuid	string	LUID of the job
jobType	string	Identifies the background job type associated with the event **Note:** Starting in April 2024, only the `IncrementExtracts`, `RefreshExtracts`, and

		`RefreshExtractsViaBridge` jobs include data for all attributes listed in the table. Attributes for other job types are under active development, and tentatively scheduled for inclusion in a future release.
notes	string	Notes of the job
objLuid	string	Some tasks are specific to a particular workbook or data source. In such cases, the object_luid is the primary key of the relevant item, in either the work-books or data sources tables, as indicated by obj_type.
objName	string	Name of the associated object. Used in conjunction with obj_luid, as described there.
objOwnerLuid	string	A foreign key reference to the user who owns the job target object
objOwnerName	string	Name of the user who owns the job target object
objRepositoryUrl	string	Uniquely identifies a workbook or data source and is used when referencing the object in a URL. The value is derived from the ASCII characters in the workbook or data source name.
objRevision	string	The revision number. Starts with 1.0 and increments by 0.1 with each republication.
objSize	integer	The number of bytes used in storing the job target object information
objType	string	Either a workbook or data source. Used in conjunction with obj_luid.
podName	string	Name of the Tableau pod that handled the job
projectLuid	string	A foreign key reference to the project in which the job target object exists

projectName	string	Name of the project that contains the job target object
projectOwnerEmail	string	Email address of the user who owns the project containing the job target object
projectOwnerLuid	string	A foreign key reference to the user who owns the project containing the job target object
scheduleLuid	string	Schedule LUID of the task; may be null if the job was manually started
scheduleName	sring	Schedule name of the task; may be null if the job was manually started
siteId	integer	ID of the site
siteName	string	Name of the Tableau site
taskId	integer	ID of the task; may be null if the job was manually started.
taskLuid	string	LUID of the task; may be null if the job was manually started.
timeZone	integer	Time zone of the job

content_owner_change

The `content_owner_change` event is logged when the content owner changes.

Attribute Name	Type	Description
contentId	integer	The ID of the content that had the owner changed
contentLuid	string	LUID of the content that had the owner changed
contentName	string	Name of the content that had the owner changed
contentType	string	The type of content, such as data source, workbook, or view

isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error
newOwnerId	integer	The ID of the new content owner
newOwnerLuid	string	The LUID of the new content owner
oldOwnerId	integer	The ID of the old content owner
oldOwnerLuid	string	The LUID of the old content owner

create_delete_group

The `create_delete_group` event is logged when a group is created or deleted.

Attribute Name	Type	Description
groupDomain	string	The domain of the group, such as local
groupId	integer	The ID of the group
groupLuid	string	The The LUID of the group
groupName	string	The name of the group that had its permissions changed
groupOperation	string	Group operation, either create or delete
isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error

create_permissions

The `create_permissions` event is logged when a new explicit permission rule is created.

Note: Deprecated in October 2024. Use the set_permissions event instead.

Attribute Name	Type	Description
authorizableType	string	The type of content that had its permissions changed,

		such as a project or workbook
capabilityId	integer	The ID of the capability. A capability is the ability to perform actions on content, such as view, filter, download, or delete
capabilityValue	string	Description of the capability
contentId	integer	The ID of the content that had the permissions updated
contentLuid	string	The LUID of the content item
contentName	string	The name of the content that had the permissions updated
granteeId	integer	The ID of the grantee
granteeLuid	string	The LUID of the grantee
granteeType	string	The type of grantee, either user or group
granteeValue	string	The updated permissions value, such as 'user allow' or 'group allow'
isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error

delete_all_permissions

The `delete_all_permissions` event is logged when all explicit permission rules for content are deleted, typically when content is deleted.

Attribute Name	**Type**	**Description**
authorizableType	string	The type of content that had its permissions changed, such as a project or workbook
contentId	integer	The ID of the content that had the permissions updated
contentLuid	string	The LUID of the content

contentName	string	The name of the content that had the permissions updated
isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error

delete_permissions

The `delete_permissions` event is logged when an explicit permission rule is deleted on content.

Attribute Name	Type	Description
authorizableType	string	The type of content that had its permissions changed, such as a project or workbook
capabilityId	integer	The ID of the capability. A capability is the ability to perform actions on content, such as view, filter, download, or delete
capabilityValue	string	Description of the capability
contentId	integer	The ID of the content that had the permissions updated
contentLuid	string	The LUID of the content
contentName	string	The name of the content that had the permissions updated
granteeId	integer	The ID of the grantee
granteeLuid	string	The LUID of the grantee
granteeType	string	The type of grantee, either user or group
granteeValue	string	The updated permissions value, such as 'user allow' or 'group allow'
isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error

delete_permissions_grantee

The `delete_permissions_grantee` event is logged when all explicit permission rules for a user are deleted, typically when the user is deleted.

Attribute Name	Type	Description
granteeId	integer	The ID of the grantee
granteeLuid	string	The LUID of the grantee
granteeType	string	The type of grantee, either user or group
isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error

display_sheet_tabs

The `display_sheet_tabs` event is logged when the "Tabbed Views" value is updated on a workbook.

Attribute Name	Type	Description
displayTabs	boolean	Indicates whether sheets of the workbook are displayed as tabs or not
isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error
workbookId	integer	The ID of the workbook

move_content

The `move_content` event is logged when content is moved, for example, moving a workbook between projects.

Attribute Name	Type	Description
contentId	integer	The ID of the content that had the owner changed

contentLuid	string	LUID of the content that had the owner changed
contentName	string	Name of the content that had the owner changed
contentType	string	The type of content, such as data source, workbook, or view
isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error
newContainerLuid	string	LUID of the new container
newContainerType	string	The new container type, such as a project
oldContainerLuid	string	LUID of the previous container
oldContainerType	string	The previous container type, such as a project

project_lock_unlock

The `project_lock_unlock` event is logged when project permissions are locked or unlocked.

Attribute Name	Type	Description
controllingProjectLuid	string	LUID of the project that controls permissions for the nested project
isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error
projectLuid	string	LUID of the project
projectOperation	string	Project operation, either lock or unlock

set_permissions

The `set_permissions` event is logged when an explicit permissions rule is created or updated for a content item.

Attribute Name	Type	Description
authorizableType	string	The type of content that had its permissions changed, such as project or workbook
capabilityId	integer	The ID of the capability. A capability is the ability to perform a certain action on a particular piece of content, such as view, filter, download or delete.
capabilityValue	string	Description of the capability
contentId	integer	The ID of the content that had the permissions set
contentLuid	string	The LUID of the content item
contentName	string	The name of the content that had the permissions set
granteeId	integer	The ID of the grantee
granteeLuid	string	The LUID of the grantee
granteeType	string	The type of grantee, either user or group
granteeValue	string	The set permissions value, such as 'user allow' or 'group allow'
isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error
permissionType	string	The permission type, either explicit or unspecified

site_storage_usage

The `site_storage_usage` event logs the total storage capacity of the site in bytes, the amount of storage used, and the percentage of the total consumed. Administrators can use this data to proactively monitor storage consumption and take action before reaching the site's storage limit.

Attribute Name	Type	Description

actorUsername	string	Username of the user who performed the action that initiated the event
initiatingUsername	string	Username of the initiating user
isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error
totalPercentageStorageQuotaUsed	float	Total percentage of storage usage
totalStorageQuotaLimit	long	Total storage capacity in bytes
totalStorageQuotaUsed	long	Total storage used in bytes

update_permissions

The `update_permissions` event is logged when an explicit permission rule is updated for a content item.

Note: Deprecated in October 2024. Use the set_permissions event instead.

Attribute Name	Type	Description
authorizableType	string	The type of content that had its permissions changed, such as a project or workbook
capabilityId	integer	The ID of the capability. A capability is the ability to perform actions on content, such as view, filter, download, or delete
capabilityValue	string	Description of the capability
contentId	integer	The ID of the content that had the permissions updated
contentLuid	string	The LUID of the content
contentName	string	The name of the content that had the permissions

		updated
granteeId	integer	The ID of the grantee
granteeLuid	string	The LUID of the grantee
granteeType	string	The type of grantee, either user or group
granteeValue	string	The updated permissions value, such as 'user allow' or 'group allow'
isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error
permissionType	string	The permission type, either explicit or unspecified

update_permissions_template

The `update_permissions_template` event is logged when a permission template for a project is updated.

Attribute Name	Type	Description
authorizableType	string	The type of content that had its permissions changed, such as a project or workbook
capabilityId	integer	The ID of the capability. A capability is the ability to perform actions on content, such as view, filter, download, or delete
capabilityValue	string	Description of the capability
contentId	integer	The ID of the content that had the permissions updated
contentLuid	string	The LUID of the content
contentName	string	The name of the content that had the permissions updated
granteeId	integer	The ID of the grantee

granteeLuid	string	The LUID of the grantee
granteeType	string	The type of grantee, either user or group
granteeValue	string	The updated permissions value, such as 'user allow' or 'group allow'
isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error
permissionType	string	The permission type, either explicit or unspecified
templateType	string	The type of permission template used to change permissions, such as workbook or data source

user_create_delete

The `user_create_delete` event is logged when a user is created or deleted.

Attribute Name	Type	Description
forUserName	string	The name of the user whose account was either created, updated or deleted
isError	boolean	Indicates if the audit scenario was completed successfully or failed with an error
siteRole	string	Site role of the user. Determines the maximum level of access a user can have on the site
targetUserId	integer	The ID of the user whose account was either created, updated, or deleted
targetUserLuid	string	The LUID of the user whose account was either created, updated, or deleted
userOperation	string	The action performed on a user, either create, delete, or site role change

Tableau Server Key Management System

Tableau Server has three Key Management System (KMS) options that allow you to enable encryption at rest. One is a local option that is available with all installations of Tableau Server. Two additional options require Advanced Management capabilities, but allow you to use a different KMS.

> **Important:** As of September 16, 2024, Advanced Management is no longer available as an independent add-on option. Advanced Management capabilities are only available if you previously purchased Advanced Management, or if you purchase certain license editions - either Tableau Enterprise (for Tableau Server or Tableau Cloud) or Tableau+ (for Tableau Cloud).

Beginning in version 2019.3, Tableau Server added these KMS options:

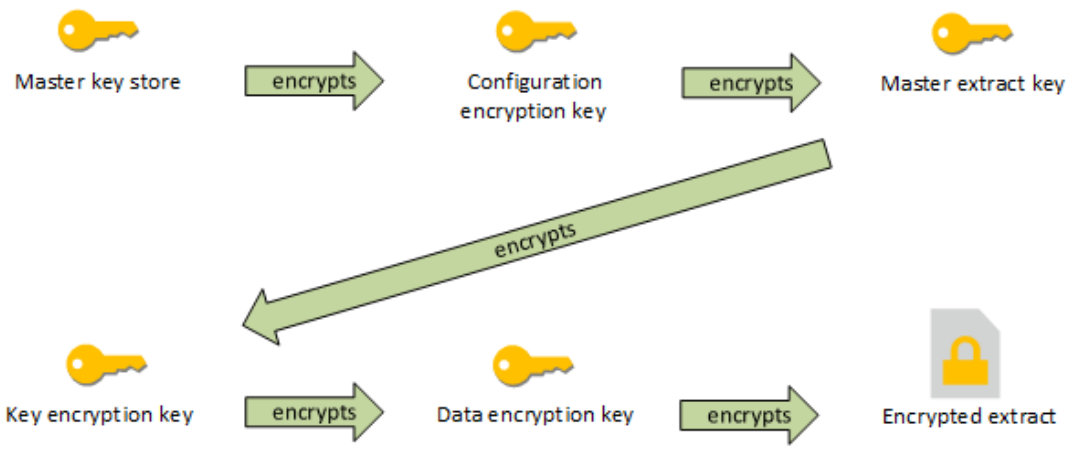
- A local KMS that is available with all installations. This is described below.
- An AWS-based KMS that comes as part of Advanced Management. For details, see AWS Key Management System.

Beginning in version 2021.1, Tableau Server added another KMS option:

- An Azure-based KMS that comes as part of Advanced Management. For details, see Azure Key Vault.

Tableau Server local KMS

The Tableau Server local KMS uses the secret storage capability described in Manage Server Secrets to encrypt and store the master extract key. In this scenario, the Java keystore serves as the root of the key hierarchy. The Java keystore is installed with Tableau Server. Access to the master key is managed by native file system authorization mechanisms by the operating system. In the default configuration, the Tableau Server local KMS is used for encrypted extracts. The key hierarchy for local KMS and encrypted extracts is illustrated here:



Troubleshoot configuration

Multi-node misconfiguration

In a multi-node setup for AWS KMS, the `tsm security kms status` command may report healthy (OK) status, even if another node in the cluster is misconfigured. The KMS status check only reports on the node where the Tableau Server Administration Controller process is running and does not report on the other nodes in the cluster. By default the Tableau Server Administration Controller process runs on the initial node in the cluster.

Therefore, if another node is misconfigured such that Tableau Server is unable to access the AWS CMK, those nodes may report Error states for various services, which will fail to start.

If some services fail to start after you have set KMS to the AWS mode, then run the following command to revert to local mode: `tsm security kms set-mode local`.

Regenerate RMK and MEK on Tableau Server

To regenerate the root master key and the master encryption keys on Tableau Server, run the `tsm security regenerate-internal-tokens` command.

AWS Key Management System

Tableau Server has three Key Management System (KMS) options that allow you to enable encryption at rest. Two of these require Advanced Management capabilities, while a local one is available with all installations of Tableau Server.

> **Important:** As of September 16, 2024, Advanced Management is no longer available as an independent add-on option. Advanced Management capabilities are only available if you previously purchased Advanced Management, or if you purchase certain license editions - either Tableau Enterprise (for Tableau Server or Tableau Cloud) or Tableau+ (for Tableau Cloud).

Beginning in version 2019.3, Tableau Server added these KMS options:

- A local KMS that is available with all installations. For details, see Tableau Server Key Management System.
- An AWS-based KMS that comes as part of Advanced Management. This is described below.

Beginning in version 2021.1, Tableau Server added another KMS option:

- An Azure-based KMS that comes as part of Advanced Management. For details, see Azure Key Vault.

With the release of version 2019.3, Tableau Server supports the AWS key management system (KMS) as part of Advanced Management.

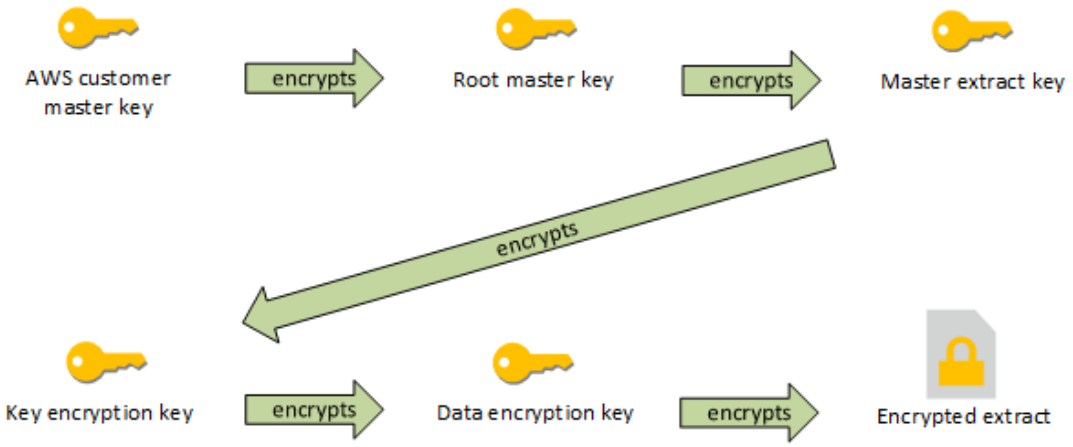
AWS KMS for encryption at rest

> AWS KMS is available as part of Advanced Management in Tableau Server. For more information see About Tableau Advanced Management on Tableau Server.

If your organization is deploying Data Extract Encryption at Rest, then you may optionally configure Tableau Server to use AWS as the KMS for extract encryption. To enable AWS KMS,

you must deploy Tableau Server in AWS EC2. In the AWS scenario, Tableau Server uses the AWS KMS customer master key (CMK) to generate an AWS data key. Tableau Server uses the AWS data key as the root master key for all encrypted extracts. However, even when configured for AWS KMS, the native Java keystore and local KMS are still used for secure storage of secrets on Tableau Server. The AWS KMS is only used to encrypt the root master key for encrypted extracts.

Using AWS to encrypt the master root key provides better security properties by not storing the master key under the same permissions as the extracts.



The key hierarchy when Tableau Server is configured with AWS KMS

Configure AWS KMS for Tableau Server encrypted extracts

To use the AWS customer master key (CMK) to encrypt the root key in the Tableau Server KMS hierarchy, you must configure Tableau Server as described in this section.

Before you begin, verify that you meet the following requirements:

- Tableau Server must be deployed in AWS EC2
- Tableau Server must be configured with Advanced Management. See About Tableau Advanced Management on Tableau Server.
- You must have administrative control of a customer master key (CMK) created in AWS Key Management Service

Step 1: Create CMK and set key policy for Tableau Server in AWS

The following procedures are performed in the AWS KMS service. References are included to AWS documentation.

1. Create the CMK that you will use for Tableau Server. See the AWS topic, Creating Keys.
2. Update the server instance's IAM role.

   Tableau Server needs to be able to authenticate with AWS KMS using the instance's IAM role. The role needs to have a policy attached to it. The policy should be giving the instance permissions to call the "GenerateDataKey" and "Decrypt" actions on the CMK. See IAM Roles for Amazon EC2..

   In a multi-node deployment of Tableau Server, all nodes of the server must be running under roles that have this policy (or equivalent) attached. You can assign the same role to all nodes in the cluster.

3. At a minimum, the CMK must have a key policy where the `Effect` is set to `Allow` the `Prinicpal` (the IAM role that is attached to the server instances) the `Action: GenerateDataKey` and `Decrypt`. See Using Key Policies in AWS KMS.

Step 2: Collect AWS configuration parameters

You will need the full ARN string from AWS KMS. This string is in the "General configuration" section of the AWS KMS management pages. The ARN is presented in this format: arn:aws:kms:<region>:<account>:key/<CMK_ID>, for example, `arn:aws:kms:us-west-2:867530990073:key/1abc23de-fg45-6hij-7k89-1l0mn1234567`.

You will also need to specify the AWS region, which is also included in the ARN string. In the example above, the region is `us-west-2`. The region is where your KMS instance resides. In the next step, you will need to specify a region as shown in the Region column in the Amazon API Gateway table.

Step 3: Configure Tableau Server for AWS KMS

Run the following command on Tableau Server. This command will restart the server:

- `tsm security kms set-mode aws --aws-region "<region>" --key-arn "arn:aws:kms:<region>:<account_number>:key/<CMK_ID>"`

  The `--key-arn` option takes a direct string copy from the ARN in the "General configuration" section of the AWS KMS management pages.

  For example, if your AWS KMS instance is running in us-west-2 region, your account number is 867530990073, and your CMK key is 1abc23de-fg45-6hij-7k89-1l0mn1234567, then the command would be as follows:

  ```
  tsm security kms set-mode aws --aws-region "us-west-2" --key-arn "arn:aws:kms:us-west-2:867530990073:key/1abc23de-fg45-6hij-7k89-1l0mn1234567"
  ```

Step 4: Enable encryption at rest

See Extract Encryption at Rest.

Step 5: Validate installation

1. Run the following command:

   ```
   tsm security kms status
   ```

   The following information may be returned:

   - The ARN (ID) of the customer master key (CMK)
   - The region the CMK is in
   - The ID of the root master key (RMK) in use. The RMK is a key that is encrypted by the CMK. Tableau Server decrypts the CMK by making calls to AWS KMS. The RMK is then used to encrypt/decrypt the master extract key (MEK). The RMK can change, but there will be only one at a time.
   - KMS stores a collection of master extract keys (MEKs). Each MEK has:
     - An ID, for example, 8ddd70df-be67-4dbf-9c35-1f0aa2421521
     - Either a “encrypt or decrypt key” or “decrypt-only key” status. If a key is "encrypt or decrypt", Tableau Server will encrypt new data with it. Otherwise, the key will only be used for decryption
     - A creation timestamp, for example, "Created at: 2019-05-29T23:46:54Z."

- First transition to encrypt and decrypt: a timestamp indicating when the key became an encrypt or decrypt key.
- Transition to decrypt-only: a timestamp indicating when the key transitioned to decrypt-only.

2. View logs after you encrypt and decrypt extracts:

   - Publish extracts to your site and then encrypt them. See Extract Encryption at Rest.
   - Access the extracts with Tableau Desktop or with Web Authoring on a browser (this will decrypt the extracts for use).
   - Search the vizqlserver_node log files for the `AwsKmsEncryptionEnvelopeAccessor` and `AwsKmsEncryptionEnvelope` strings. The default location of the logs are at `/var/opt/tableau/tableau_server/data/tabsvc/logs/`

     Log entry examples that indicate successful configuration include the following:

     - Decrypted the RMK with ID 1abc23de-fg45-6hij-7k89-1l0mn1234567 using the CMK with ARN arn:aws:kms:us-west-2:867530990073:key/1234567d-a6ba-451b-adf6-3179911b760f
     - Using RMK with ID 1abc23de-fg45-6hij-7k89-1l0mn1234567 to decrypt KMS store

     For publishing and extract refreshes related to KMS, search the backgrounder logs. For more information about logs, see Tableau Server Logs and Log File Locations.

Troubleshoot configuration

Multi-node misconfiguration

In a multi-node setup for AWS KMS, the `tsm security kms status` command may report healthy (OK) status, even if another node in the cluster is misconfigured. The KMS status check only reports on the node where the Tableau Server Administration Controller process is

running and does not report on the other nodes in the cluster. By default the Tableau Server Administration Controller process runs on the initial node in the cluster.

Therefore, if another node is misconfigured such that Tableau Server is unable to access the AWS CMK, those nodes may report Error states for various services, which will fail to start.

If some services fail to start after you have set KMS to the AWS mode, then run the following command to revert to local mode: `tsm security kms set-mode local`.

Refresh AWS CMK

Refreshing the AWS CMK is a task that you perform with AWS. By default, the AWS CMK will refresh once a year. See the AWS topic, How Automatic Key Rotation Works. Since the ARN and region do not change, you do not need to update the KMS configuration on Tableau Server for normal CMK refresh scenarios.

After AWS CMK refreshes, you must regenerate the internal RMK and MEKs on Tableau Server. You should also re-encrypt all extracts with the new CMK:

1. Run the `tsm security regenerate-internal-tokens` command to regenerate all internal keys on Tableau Server, including the RMK and MEKs used for extract encryption.
2. Run `tabcmd reencryptextracts <site-name>` to re-encrypt extracts on a given site. Run this command on every site where you are storing encrypted extracts. Depending on the number of encrypted extracts on the site, this operation could consume significant server processing load. Consider running this operation outside of business hours. See Extract Encryption at Rest.

Regenerate RMK and MEK on Tableau Server

To regenerate the root master key and the master encryption keys on Tableau Server, run the `tsm security regenerate-internal-tokens` command.

Back up and restore with AWS KMS

A server backup can be taken in AWS mode with no additional configurations or procedures. The backup contains encrypted copies of the RMK and MEKs. Decrypting the keys requires access and control of the AWS CMK.

For the restore scenario, the server being restored to can be in either KMS mode, including Local. The only requirement is that the server the backup is being restored to has decrypt access to the CMK the backup itself used.

Upon restore, the MEKs from the backup are imported as decrypt-only keys. The RMK is not migrated over. A new RMK is generated as part of the installation/restore process.

Azure Key Vault

Tableau Server has three Key Management System (KMS) options that allow you to enable encryption at rest. Two of these require Advanced Management capabilities, while a local one is available with all installations of Tableau Server.

> **Important:** As of September 16, 2024, Advanced Management is no longer available as an independent add-on option. Advanced Management capabilities are only available if you previously purchased Advanced Management, or if you purchase certain license editions - either Tableau Enterprise (for Tableau Server or Tableau Cloud) or Tableau+ (for Tableau Cloud).

Beginning in version 2019.3, Tableau Server added these KMS options:

- A local KMS that is available with all installations. For details, see Tableau Server Key Management System.
- An AWS-based KMS that comes as part of Advanced Management. For details, see AWS Key Management System.

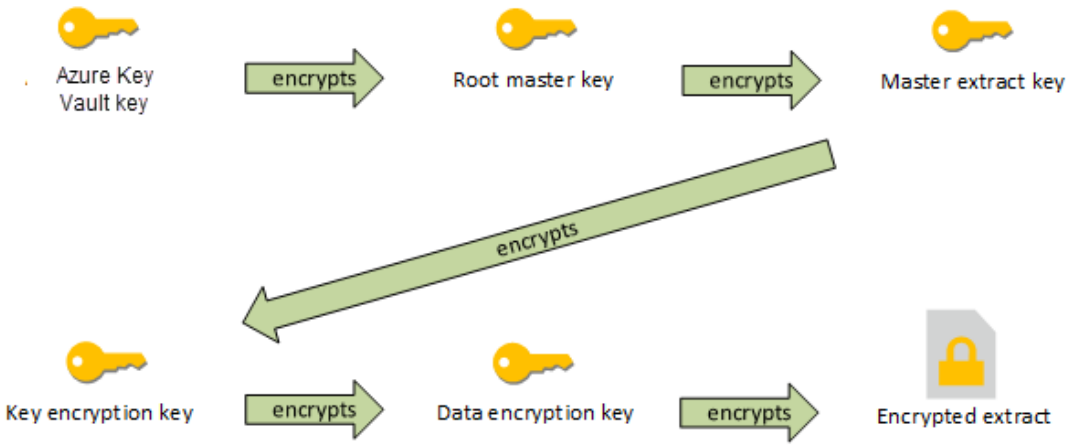
Beginning in version 2021.1, Tableau Server added another KMS option:

- An Azure-based KMS that comes as part of Advanced Management. This is described below.

Azure Key Vault for encryption at rest

> Azure Key Vault is available as part of Advanced Management in Tableau Server beginning in version 2021.1.0. For more information see About Tableau Advanced Management on Tableau Server.

If your organization is deploying Data Extract Encryption at Rest, then you may optionally configure Tableau Server to use Azure Key Vault as the KMS for extract encryption. To enable Azure Key Vault, you must deploy Tableau Server in Azure. In the Azure scenario, Tableau Server uses the Azure Key Vault to encrypt the root master key (RMK) for all encrypted extracts. However, even when configured for Azure Key Vault, the Tableau Server native Java keystore and local KMS are still used for secure storage of secrets on Tableau Server. The Azure Key Vault is only used to encrypt the root master key for encrypted extracts.



The key hierarchy when Tableau Server is configured with Azure Key Vault

Configure Azure Key Vault for Tableau Server encrypted extracts

To use the Azure Key Vault to encrypt the root key in the Tableau Server KMS hierarchy, you must configure Tableau Server as described in this section.

Before you begin, verify that you meet the following requirements:

- Tableau Server must be deployed in Azure.
- Tableau Server must be configured with a Advanced Management license. See About Tableau Advanced Management on Tableau Server.
- You must have administrative control over the key vault in Azure where the key resides.

Step 1: Create a key vault and key for Tableau Server in Azure

The following procedures are performed in the Azure Key Vault service. References are included to Azure documentation.

1. Create the key vault that you will use for Tableau Server. See the Azure topic, Create a key vault.
2. Create a key in the vault. See the Azure topic, Manage keys and secrets.

   The key must be an asymmetric, RSA type, but can be any size (Tableau Server does not care about the key size). We recommend you use the Principle of Least Privilege to have maximum security.

   Tableau requires permissions to perform the GET, UNWRAP KEY, and WRAP KEY commands operations and we recommend you allow access for only these operations for least privilege. Assign the access policy to the VM you are running Tableau Server on.

   In a multi-node deployment of Tableau Server, the access policy must be assigned to all nodes of the server cluster.

Step 2: Collect Azure configuration parameters

You will need the key vault name and the key name from Azure.

Step 3: Configure Tableau Server for Azure Key Vault

Run the following command on Tableau Server. This command will restart the server:

- `tsm security kms set-mode azure --vault-name "<vault name>" --key-name "<key name>"`

  The `--vault-name` and `--key-name` options a direct string copies from your Azure key vault.

For example, if your Azure key vault is named `tabsrv-keyvault` and your key is `tabsrv-sandbox-key01`, then the command would be as follows:

```
tsm security kms set-mode azure --vault-name "tabsrv-keyvault"
--key-name "tabsrv-sandbox-key01"
```

Step 4: Enable encryption at rest

See Extract Encryption at Rest.

Step 5: Validate installation

1. Run the following command:

   ```
   tsm security kms status
   ```

   The following information may be returned:

   - Status: OK (indicates the Key Vault is accessible by the controller node):
   - Mode: Azure Key Vault
   - Vault name: <key_vault_name>
   - Azure Key Vault key name: <key_name>
   - List of available UUIDs for MEKs indicating which key is active
   - Error information if the KMS data is not accessible

2. View logs after you encrypt and decrypt extracts:

   - Publish extracts to your site and then encrypt them. See Extract Encryption at Rest.
   - Access the extracts with Tableau Desktop or with Web Authoring on a browser (this will decrypt the extracts for use).
   - Search the vizqlserver_node log files for the `AzureKeyVaultEnvelopeAccessor` and `AzureKeyVaultEnvelope` strings. The default location of the logs are at `/var/opt/tableau/tableau_server/data/tabsvc/logs/`

For publishing and extract refreshes related to the Azure Key Vault, search the backgrounder logs. For more information about logs, see Tableau Server Logs and Log File Locations.

Troubleshoot configuration

Multi-node misconfiguration

In a multi-node setup for Azure Key Vault, the `tsm security kms status` command may report healthy (OK) status, even if another node in the cluster is misconfigured. The KMS status check only reports on the node where the Tableau Server Administration Controller process is running. It does not report on the other nodes in the cluster. By default the Tableau Server Administration Controller process runs on the initial node in the cluster.

Therefore, if another node is misconfigured so that Tableau Server is unable to access the Azure key, those nodes may report Error states for various services, which will fail to start.

If some services fail to start after you have set KMS to the "azure" mode, then run the following command to revert to local mode: `tsm security kms set-mode local`.

Refresh Azure Key

You refresh the Azure key in Azure. There is no required or scheduled key refresh period. You can refresh your key by creating a new key version in Azure. Because the Key Vault name and Key Name do not change, you do not need to update the KMS configuration on Tableau Server for normal Azure Key refresh scenarios.

Back up and restore with Azure Key Vault

A server backup can be taken in Azure Key Vault mode with no additional configurations or procedures. The backup contains encrypted copies of the RMK and MEKs. Decrypting the keys requires access and control of the Azure Key Vault.

For the restore scenario, the server being restored to can be in either Azure Key Vault or Local KMS mode. The only requirement is that the server to which the backup is being restored has access to the Azure Key Vault the backup itself used.

Tableau Server External File Store

This topic provides an overview of Tableau Server External File Store.

Tableau Server File Store stores extracts and workbook revisions. Typically, Tableau Server File Store is a built-in Tableau Server process that is installed locally on Tableau Server. Starting in Tableau Server 2020.1, you can configure Tableau Server to use an external storage to store the File Store data. The external storage must be a network share, a dedicated file storage that enables multiple users and heterogeneous client devices to retrieve data from a centralized disk capacity. This can be a Server Message Block (SMB) for Windows or a Network File System (NFS) for Linux installations. Users on a local area network (LAN) access the shared storage via a standard Ethernet connection.

With this new feature, Tableau Server can now be configured in two ways:

- Install File Store locally, meaning File Store is installed on the Tableau Server nodes.
- Use External File Store (Beginning in 2020.1).

Why use External File Store?

Using external storage has the following key benefits over installing File Store locally:

- **Centralized location:** When File Store is installed locally, the data needs to be replicated across multiple File Store nodes, which will consume network bandwidth. Moving data to a centralized location will eliminate the need for running File Store on multiple nodes in a Tableau cluster and replication between the nodes. This also reduces the disk space requirements on an individual node and also reduce the network bandwidth usage since the data will not be replicated on multiple nodes.
- **Improving backup time:** Snapshot backup technologies are efficient, and by using snapshot backup of the Tableau Data, you can expect to significantly reduce the amount of time it takes to do a Tableau backup.

Managing External File Store

License Management

To configure External File Store, you must have Advanced Management capabilities for Tableau Server. For more information, see About Tableau Advanced Management on Tableau Server. If you don't have Advanced Management capabilities, you will see the following behavior:

- If you try to configure Tableau Server with External File Store during installation, you will see an error message, but you will be able to continue the installation and Tableau Server File Store will be installed locally.
- If you are already using External File Store, and Advanced Managementis no longer available to your installation, you will see the following behavior:
  - The server will fail on restart.
  - Backups will fail.
  - If you no longer have Advanced Management capabilities, but have a valid Tableau Server license, you can migrate the External File Store to local File Store, to get your server up and running again. For more information on how to migrate from external repository to local repository, see Reconfigure File Store .

Supported Migration Scenarios

- Moving File Store installed locally on Tableau Server to an external managed storage (network attached storage).
- Move File Store from external managed storage to Tableau Server.

Backup and Restore

Backup on Tableau Server with External File Store is different than how you would create backup when File Store is installed locally. For more information on how to do backup and restore on Tableau Server with External File Store, see Backup and Restore with External File Store.

Upgrade Considerations

There are no special steps necessary when upgrading Tableau Server configured with an External File Store. You can follow the normal upgrade process.

High Availability Considerations

Tableau Server does not manage or setup high availability for the External File Store. Your managed storage may have solutions to support redundancy and high availability.

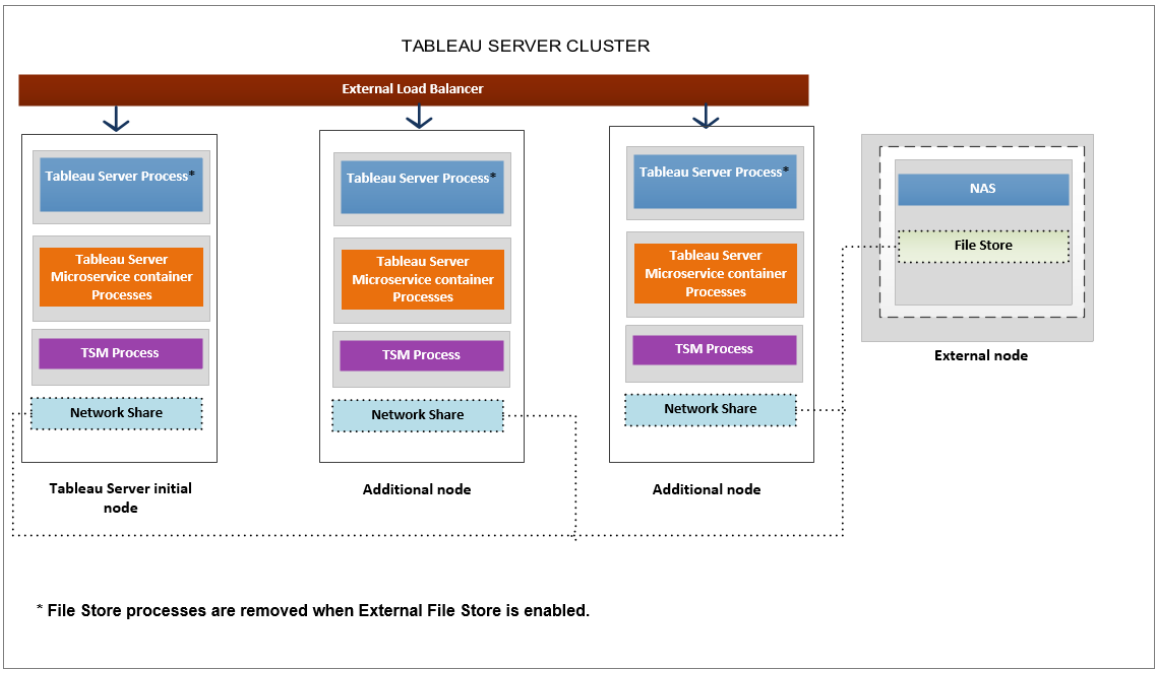
Topology

When you configure Tableau Server with External File Store, you will no longer run File Store locally. The Server status page will indicate that the File Store process is on an external node.

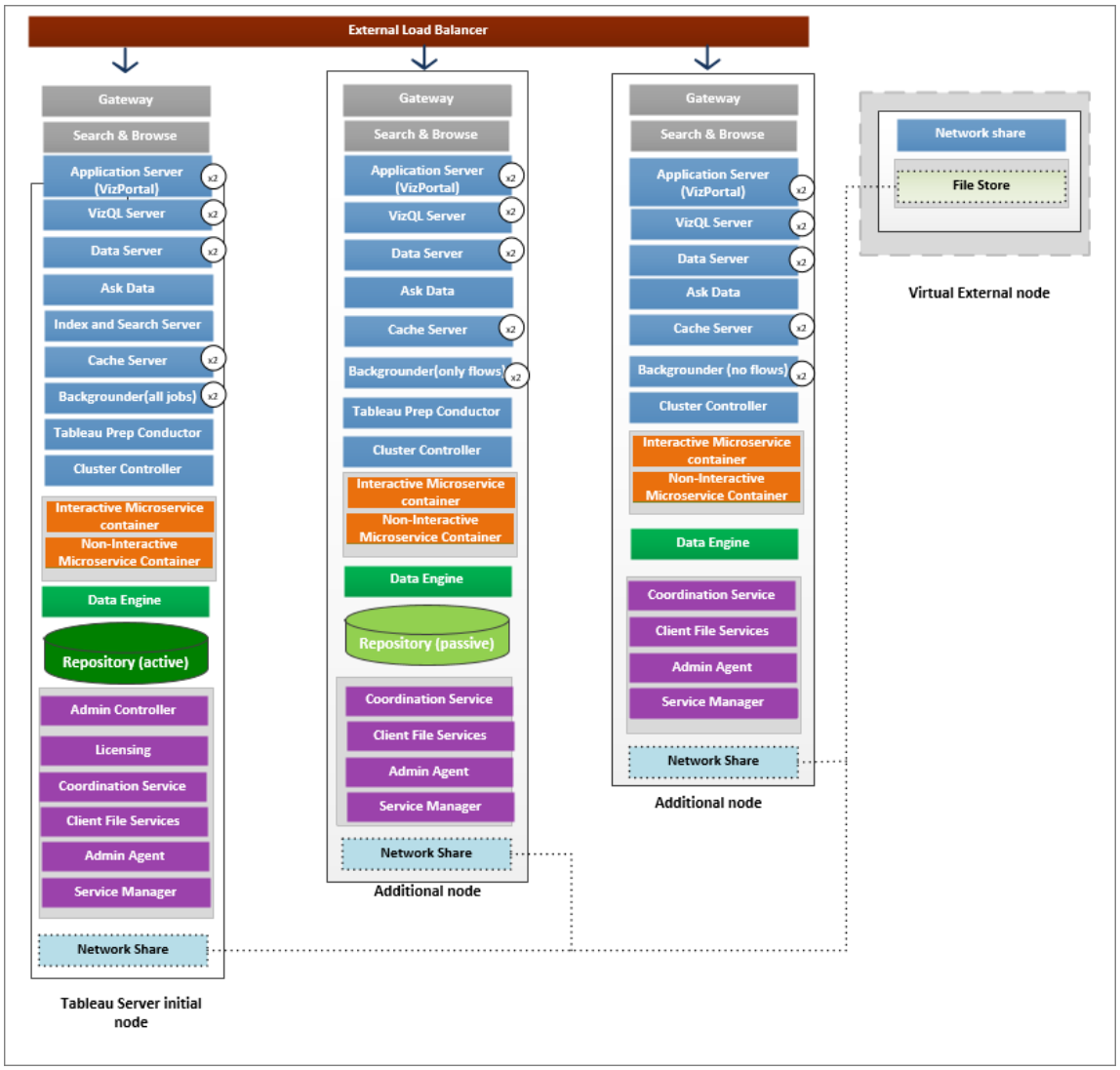
When File Store is configured external to Tableau Server, Data Engine and File Store are no longer co-located. During setup, Data Engine will continue to be automatically installed with other processes as described in Tableau Server Data Engine, except for File Store. However, when you have Tableau Server configured with External File Store, you will be able to install Data Engine on a separate node without any other processes.

When File Store is configured externally, Data Engine will access the File Store data (extracts) on the storage system across the network. To make sure that your overall system performs to your requirements, there are a few things you must consider for your network and storage system. For more information, see Performance Considerations for External File Store.

The diagram below is a summarized version of the Tableau Server topology with External File Store.



The diagram below is a detailed version of the Tableau Server topology with External File Store and shows all the processes installed on each node.



Next

Install Tableau Server with External File Store

Install Tableau Server with External File Store

This topic walks you through the process of configuring network share as your Tableau Server File Store for a new installation. If you are trying to do this on an existing installation of Tableau Server that has File Store running locally, see Reconfigure File Store .

Prerequisites

- You must use Tableau Server 2020.1 or later.
- You must have network share that you can use as your storage option. For recommendations on the storage solution, see Performance Considerations for External File Store.

  Use **NFS** for Linux installations.

  Estimating the storage size: You must take into account the amount of storage needed for publishing and refreshing extracts. In addition, you must also take into account the repository backup size unless you specifically choose the option to do your repository backup separately as described in the Option 2: Back up repository separately topic.

  - Extracts:
    - Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out how many extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.
    - Consider the space needed by the temp directory during an extract refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.
  - Repository Backup:
    - To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base` directory.
    - To obtain the exact size of the repository data, open the backup file and use the size of the `workgroup.pg_dump` file.
- You must have Advanced Management capabilities enabled on your server. To learn more about Advanced Management, see About Tableau Advanced Management on Tableau Server.

Install Tableau Server with External File Store

You can install Tableau Server with File Store using a network share to store Tableau Server data. This solution replaces the need for running the File Store process locally. To learn more about this solution and its benefits, see Tableau Server External File Store.

Use the following steps to install your Tableau Server with External File Store during install:

Step 1: Configure a network share

On your file server:

1. Create and share a directory to use as the Tableau Server External File Store.
2. Make sure the network share is accessible as a directory in the same location on all the Tableau Server nodes.
3. Create the tableau directory in your network share and give full access to **tableau user** and **tableau group**. The tableau user will need read and write permissions to the directory on the network share. We recommend calling the directory '**tableau**'.

   ```
   /mnt/<network share>/tableau/
   ```

4. **Validate that the network share is configured properly**: From Tableau Server run a command to write to a network share and confirm that you are able to write to it.

Step 2: Download and install TSM

1. Download the appropriate installer based on the distribution of Linux you are using.
2. Log on as a user with sudo access to the computer where you want to install Tableau Server.
3. Download the .rpm or .deb installer package.
4. Navigate to the directory where you copied the .rpm or .deb package.
5. Use the package manager to install the Tableau Server.

- On **RHEL-like** distributions, including CentOS, you have the option to install Tableau Server to a non-default location.

- Default location—To install to the default location (/opt/tableau/tableau_server), run the following commands:

  ```
  sudo yum update

  sudo yum install tableau-server-<version>.x86_64.rpm
  ```

- Non-default location—to install to a non-default location, you must use rpm -i. You will also need to install all dependent packages. See the note below.

  Run the following command:

  ```
  sudo rpm -i--prefix/preferred/install/path tableau-server.rpm
  ```

  > **Note:** When you use yum to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau Server. If you want to install to a non-default location, or your organization does not allow you to use yum and you must install using rpm -i, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- On **Ubuntu and Debian**, run the following commands:

  ```
  sudo apt-get update

  sudo apt-get -y install gdebi-core

  sudo gdebi -n tableau-server-<version>_amd64.deb
  ```

Step 3: Initialize TSM

1. Run the following script to start TSM:

   ```
   sudo ./initialize-tsm --accepteula --<optional_parameters>
   ```

The only required parameter for the initialize-tsm script is --accepteula. You must include this parameter to accept the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

```
/opt/tableau/tableau_server/packages/docs.<version_code>/Commercial_EULA.txt
```

2. Log off and log on again to the terminal before you configure Tableau Server.

   When you log on again, you create a new session in which group membership changes have taken effect. The new session also has access to the environment variables added by the initialize-tsm script.

   Alternatively, you can run the following command to update your path for the current session (but not to update your group membership):

   ```
   source /etc/profile.d/tableau_server.sh
   ```

Step 4: Activate and register Tableau Server

Provide the Tableau Server Key and the Advanced Management key in the activate step. You will need to run the following command twice, first with the Tableau Server product key and then with the Advanced Management product key:

```
tsm licenses activate -k <product key>
```

Step 5. Enable External File Store

Configuring Tableau Server with external repository can only be done using TSM CLI.

1. Enable the network storage feature using the following tsm commands:

   ```
   tsm topology external-services storage enable --network-share /mnt/<network share name>/tableau
   ```

   The setup program automatically creates the following directory structure in the share:

   PostgreSQL data backups:

tableau_data/tabsvc/repository_backup

> **Note:** This directory will be created the first time you create a backup.

Extracts and workbook revisions:

tableau_data tabsvc/dataengine/extracts

tableau_data/tabsvc/dataengine/revisions

Step 6: Configure the initial node settings

Follow the instructions provided in the Configure Initial Node Settings topic.

Step 7: Complete the install

You must create the initial administrative account for Tableau Server.

- If you configured a local identity store during setup, then specify a name and password that you want to use.
- If you configured a LDAP or Active Directory identity store during setup, then you must specify a user account that is a member of the directory.

To create the initial user, run the following tabcmd command:

```
tabcmd initialuser --server localhost:80 --username '<new-admin-user-name>'
```

After you run the command, the shell will prompt for an administrative password.

Step 8: Post-installation tasks

After you have created the Tableau Server administrator account, continue your deployment by working through the configuration steps described in the Post Installation Tasks topics.

Who can do this

Tableau Server Administrators can install and configure Tableau Server and External File Store. In addition, you must have permissions and access to configure the network share to use with Tableau Server.

Next

Backup and Restore with External File Store

Reconfigure File Store

Your Tableau Server may be configured with a locally running File Store or an External File Store. This topic describes the steps needed to reconfigure your existing Tableau Server:

- Reconfigure your Tableau Server to use External File Store. This will move your File Store to a network share.
- Reconfigure your Tableau Server to run File Store locally. This will move your File Store from the external storage to your Tableau Server.
- Configure your Tableau Server to use a different storage. An example of this might be when your current network share is at the end of life and you need to use a new network share with new hardware.

Reconfigure Tableau Server with External File Store

Prerequisites

- Tableau Server should be version 2020.1 or later.
- You must have a network share to use as the external storage.

  Use **NFS** for Linux installations.

  Storage and network considerations: See Performance Considerations for External File Store.

Estimating the storage size: You must take into account the amount of storage needed for publishing and refreshing extracts. In addition, you must also take into account the repository backup size unless you specifically choose the option to do your repository backup separately as described in the Option 2: Back up repository separately topic.

- Extracts:
  - Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out how many extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.
  - Consider the space needed by the temp directory during an extract refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.
- Repository Backup:
  - To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base` directory.
  - To obtain the exact size of the repository data, open the backup file and use the size of the `workgroup.pg_dump` file.
- Tableau Server should have a valid and activated Advanced Management license.

Step 1: Upgrade Tableau Server

Upgrade your Tableau Server to 2020.1 or later: Upgrading from 2018.1 and Later (Linux) . If your Tableau Server is already on version 2020.1 or later, you can skip this step.

Step 2: Activate the Advanced Management license

1. View Server Licenses to make sure you have a Advanced Management license activated on your Tableau Server.
2. If you don't have a Advanced Managementon your Tableau Server, use the following tsm command to activate the license. Provide the Advanced Management key as your

product key:

```
tsm licenses activate -k <product key>
```

Step 3: Configure File Store to use an external storage

After completing the upgrade and verifying the licenses, configure Tableau Server with External File Store. This will move any existing data from your local File Store to the external storage of your choice.

1. Configure a network share. On the File Server:
   - Create and share a directory to host the files
   - On all the Tableau Server nodes, map the network share or use a UNC path.
2. Create a tableau directory in your network share and give full access to **tableau user** and **tableau group**. The tableau user will need read and write permissions to the directory on the network share. We recommend calling the directory **tableau**.
3. Enable the network share feature using the following tsm commands:

   ```
   tsm topology external-services storage enable --network-share
   /mnt/<network share name>/tableau
   ```

   The setup program automatically creates the following directory structure and will move the data from the local File Store to the external storage. The local File Stores will be automatically decommissioned during this process.

   Extracts and workbook revisions:

   tableau_data tabsvc/dataengine/extracts

   tableau_data/tabsvc/dataengine/revisions

   PostgreSQL data backups:

tableau_data/tabsvc/repository_backup

> **Note:** This directory will be created the first time you create a backup.

Reconfigure Tableau Server to use local File Store

1. Stop Tableau Server by running the following command:

   ```
   tsm stop
   ```

2. Run the following command to move the File Store data from the external storage to Tableau Server:

   ```
   tsm topology external-services storage disable –fsn <node1, node2>
   ```

3. Start Tableau Server by running the following command:

   ```
   tsm start
   ```

For a Tableau Server cluster, specify the nodes where File Store should be installed. The data is copied to the first node specified in the command and then replicated to the other nodes.

> **Note:** When moving File Store from external to local, make sure Data Engine process is not installed by itself on a separate node and is installed along with one of the core services which include File Store, Application Server (VizPortal), VizQL Server, Data Server. and Backgrounder.
>
> Typically, when you install Tableau Server Data Engine installation happens automatically and is installed on the nodes that have one or more of the core services. However, when you configure Tableau Server to use external storage, you will have the ability to manually install Data Engine on a node on its own without co-locating with the core processes.
>
> If you currently have a Data Engine process installed on a separate node, you can either

choose to install File Store on that node or remove Data Engine from that node, before running the disable command. If you install File Store on a node that currently does not have Data Engine installed, Data Engine will be added automatically.

If you have a Data Engine only node when you run the disable command, it will result in an error.

Configure Tableau Server to use a different external storage

1. Configure the new network share. On the File Server:
    1. Create and share a directory to host the files.
    2. On all the Tableau Server nodes, map the network share or use a UNC path.
2. Create a tableau directory in your network share and give full access to **tableau user** and **tableau group**. The tableau user will need read and write permissions to the directory on the network share. We recommend calling the directory **tableau**.
3. Stop Tableau Server by running the following command:

    ```
    tsm stop
    ```

4. Run the following command to configure Tableau Server to use the new network share:

    ```
    tsm topology external-services storage switch-share --network-share /mnt/<newshare>/tableau
    ```

5. Start Tableau Server by running the following command:

    ```
    tsm start
    ```

Who can do this

Tableau Server Administrators can move File Store locations. In addition, you will need access to the external storage that is used for the External File Store.

Backup and Restore with External File Store

When you have External File Store enabled on your Tableau Server, you cannot use the `tsm maintenance backup` command to do a backup of the Tableau Server repository and File Store Data. Instead, use a "snapshot" backup process to create a point in time snapshot of your network share.

- Tableau Server configured with External File Store
- Tableau Server configured with External File Store and External Repository

Backup strategies:

The backup strategy you use depends on your recovery plan. The snapshot backup process may not be sufficient by itself as it only creates a backup of the File Store (and repository data if requested), and there are other configurations and settings that you may need in order to do a full recovery.

> **Important:** We strongly recommend disabling scheduled tasks before you perform an upgrade. This includes all updates to data content and should be done before you create your pre-upgrade backup. This may involve disabling jobs that are triggered outside of Tableau Server, such as those initiated through REST API-based extract refreshes or using tabcmd.

Here are some scenarios that illustrate when the snapshot backup process may or may not be enough:

- **Standby Tableau Server**—If you maintain a standby Tableau Server to use if your production Server is down, creating a snapshot backup and restoring it to your standby server on a regular schedule may be sufficient. Your backup schedule should be based on your recovery point objective.
- **New Tableau Server, existing configuration not needed**—If you plan to use a new Tableau Server installation in case of a disaster, but don't necessarily need to use the configurations and settings from your existing Tableau Server installation you can install a fresh instance of Tableau Server and use the snapshot to restore your data.
- **New Tableau Server, existing configuration needed**—If you plan to use a new Tableau Server installation that includes your existing configurations and settings as well as your backed up date, you need additional files in along with the snapshot backup. To do a full backup including all the configurations and settings, follow these instructions:
    1. Export topology and configuration data. This exports most of the Tableau Server configuration and topology. For more information, see Perform a Full Backup and Restore of Tableau Server.
    2. Create a network share snapshot of the File Store (and repository data, if desired) as described in the Backup and Restore with External File Store section of this topic.
    3. Document the settings that are not included in the export. These include values for system user accounts, coordination service deployment configuration, and customized settings. For more information, see Perform a Full Backup and Restore of Tableau Server.

Tableau Server configured with External File Store

When you have Tableau Server configured with an External File Store your backup process needs to include creating a point-in-time snapshot backup of the network share with the External File Store. The following procedure describes how to do this.

Note: If you have both an External File Store and an external repository, see Tableau Server configured with External File Store and External Repository.

Creating a snapshot backup

Use the following steps to create a snapshot backup:

1. Prepare for the snapshot backup.

   Run the following command to create a repository backup file and temporarily copy it to the network share. Tableau Server continues to operate normally during the snapshot prepare process. To ensure a consistent snapshot, the internal process that deletes unused extracts will be paused. This process will resume once you complete the backup process described in a later step.

   ```
   tsm maintenance snapshot-backup prepare
   ```

   You should see the following message when the prepare step is complete: **Preparation for snapshot backup succeeded**.

   Confirm that the repository backup file was created on the network share.

   > **Note:** The prepare step creates a backup file of the repository and KMS and Asset keys. If you are using a cloud solution for your encryption key solution, you will need access to the CMK used to decrypt the keys which is necessary for restore. For more information on encryption key solution with AWS, see AWS Key Management System. For more information on encryption key solution with Azure, see Azure Key Vault.

2. Create a snapshot of your network share.

   Use the appropriate process to create a snapshot of your network share. The snapshot is a read-only version of the share, taken at a particular point in time. This will include your File Store data, along with the repository backup file that was temporarily copied to the share in the previous step. The steps you take to create this network share snapshot depend on your network. See your networking documentation for details.

3. Complete the snapshot backup process.

Run the following command to complete the backup process and restart the paused internal processes. This removes the temporary repository backup file that was copied to your network share.

```
tsm maintenance snapshot-backup complete
```

Confirm that the temporary repository backup file was removed from the network share.

Restoring a snapshot backup

These steps apply to both single node and multi-node Tableau Server installations.

1. Stop Tableau Server.

   At the command prompt, run the following command:

   ```
   tsm stop
   ```

2. Restore your File Store data by restoring your network share snapshot to your network. Your specific steps will depend on your network.

3. Restore the repository data.

   Use the following command to restore the repository data:

   ```
   tsm maintenance snapshot-backup restore
   ```

   Note: If you are using a cloud solution for your encryption key solution, make sure the server where the backup is being restored has decrypt access to the cloud instance where the CMK is deployed.

4. Restart Tableau Server.

At a command prompt, run the following command:

```
tsm start
```

Tableau Server configured with External File Store and External Repository

When you have Tableau Server configured with both an External File Store and an external repository, there are special steps you need to take to create a backup. These include creating a point-in-time snapshot backup of the network share with the External File Store, and may include separately backing up your external repository. The following procedure describes how to do this.

Note: If you have an External File Store but are using the default repository, see Tableau Server configured with External File Store.

Backing up the repository

When you have both an External File Store and an external repository, you have two options for backing up the repository data. There are reasons why you might choose each of these:

- Include the repository backup with the network share snapshot:

  Ease of management. You do not have to do a separate backup of the repository, and the backup is in sync with the File Store data.

- Back up the repository independently:

  Doing a backup of an External Repository separately can be faster, especially if you are using a cloud solution that allows for snapshot backups of the instance. The size of the backup file can significantly impact the time it takes to prepare for a network share snapshot.

Option 1: Include repository backup with network share snapshot

Create a snapshot backup

1. Prepare for the snapshot backup.

   Run the following command to create a repository backup file and temporarily copy it to the network share. Tableau Server continues to operate normally during the snapshot prepare process. To ensure a consistent snapshot, the internal process that deletes unused extracts will be paused. This process will resume once you complete the backup process described in a later step.

   ```
   tsm maintenance snapshot-backup prepare --include-pg-backup
   ```

   Note: Starting in 2021.1, if you have both External File Store and External Repository enabled, you *must* use the `--include-pg-backup` option to create the repository backup. If you are upgrading from a version earlier than 2021.1 and you have scripts to run or schedule your backups, and want to continue to include the repository backup, add the **`--include-pg-backup`** option in the prepare command as shown above. For versions prior to 2021.1 you do not need the option, the repository backup is automatically included.

   You should see the following message when the prepare step is complete: **Preparation for snapshot backup succeeded**.

   Confirm that the repository backup file was created on the network share.

   Note: The prepare step creates a backup of the repository and KMS and Asset keys. If you are using a cloud solution for your encryption key solution, you will need access to the CMK used to decrypt the keys which is necessary for restore. For more information on encryption key solution with AWS, see AWS Key

> Management System. For more information on encryption key solution with Azure, see Azure Key Vault.

2. Create a snapshot of your network share.

   Use the appropriate process to create a snapshot of your network share. The snapshot is a read-only version of the share, taken at a particular point in time. This will include your File Store data, along with the repository backup file that was temporarily copied to the share in the previous step. The steps you take to create this network share snapshot depend on your network. See your networking documentation for details.

3. Complete the snapshot backup process.

   Run the following command to complete the backup process and restart the paused internal processes. This also removes the temporary repository backup file that was copied to your network share.

   ```
   tsm maintenance snapshot-backup complete
   ```

Restoring a snapshot backup

These steps apply to both single node and multi-node Tableau Server installations.

Important: If you perform Blue/Green upgrades or manually upgrade Tableau Server 2021.4 (or earlier) using the tsm maintenance (backup and restore) method, you must enable `legacy-identity-mode` before you can restore to Tableau Server 2022.1 (or later). For more information, see Troubleshoot Issues with the Identity Migration.

1. Stop Tableau Server.

   At the command prompt, run the following command:

   ```
   tsm stop
   ```

2. Restore your File Store data by restoring your network share snapshot to your network. Your specific steps will depend on your network.

3. Restore the repository data.

   Use the following command to restore the repository data:

   ```
   tsm maintenance snapshot-backup restore
   ```

   Note: If you are using a cloud solution for your encryption key solution, make sure the server where the backup is being restored has decrypt access to the cloud instance where the CMK is deployed.

4. Restart Tableau Server.

   At a command prompt, run the following command:

   ```
   tsm start
   ```

Option 2: Back up repository separately

This option is recommended only when the host platform for the External Repository allows you to do snapshot backup. If you are using Azure as your host platform, we recommend using Option 1.

Create snapshot backups

1. Prepare for the snapshot backup.

   Run the following command to create a repository backup file and temporarily copy it to the network share. Tableau Server continues to operate normally during the snapshot prepare process. To ensure a consistent snapshot, the internal process that deletes unused extracts will be paused. This process will resume once you complete the backup process described in a later step.

   ```
   tsm maintenance snapshot-backup prepare
   ```

You should see the following message when the prepare step is complete: **Preparation for snapshot backup succeeded**.

> **Note:** The prepare step creates a backup of the KMS and Asset keys. If you are using a cloud solution for your encryption key solution, you will need access to the CMK used to decrypt the keys which is necessary for restore. For more information on encryption key solution with AWS, see AWS Key Management System. For more information on encryption key solution with Azure, see Azure Key Vault.

2. Create a snapshot of your network share.

   Use the appropriate process to create a snapshot of your network share. The snapshot is a read-only version of the share, taken at a particular point in time. This will include your File Store data. The steps you take to create this network share snapshot depend on your network. See your networking documentation for details.

3. Create a backup of the repository: Use the backup technology of platform where you are hosting your external repository to create a backup.

   Important: The snapshot of the network share and the Repository backup must be completed within 3 hours and 30 minutes after completing the Prepare step (step 1). This is to make sure that the File Store and the Repository backups are in sync and to allow the restore to work properly.

   For more information on creating a snapshot of AWS DB instance, see Creating a DB snapshot.

   For more information on creating a backup of Azure DB instance, see Backup and Restore on Flexible Server (PostgreSQL 12 and later) or Backup and Restore on Single Server (PostgreSQL 11 or earlier).

> **Note:** If you are using a cloud solution for your encryption key solution, make sure the server where the backup is being restored has decrypt access to cloud instance where CMK is deployed.

4. Complete the snapshot backup process.

   Run the following command to complete the backup process and restart the paused internal processes.

   ```
   tsm maintenance snapshot-backup complete
   ```

Restoring a snapshot backup

These steps apply to both single node and multi-node Tableau Server installations.

1. Use the database backup of your External Repository. If you are using a cloud platform to host your Repository, this generally requires that you create a new database instance into which you restore the backup.

   For detailed instructions on creating a new instance, see the option for your hosting solution in Install External Repository.

2. Use the instructions in Step 1 of the Install Tableau Server with External PostgreSQL Repository topic to create a configuration file for the new instance.

3. Stop Tableau Server.

   At the command prompt, run the following command:

   ```
   tsm stop
   ```

4. If the restore of the external repository requires a new database instance, use the following command to point Tableau Server to the new database instance:

```
tsm topology external-services repository replace-host -f <file-
name>.json -c <ssl certificate file>.pem
```

The `.json` file is the configuration file you created in Step 2. The certificate file is the SSL certificate you downloaded from the new database instance.

5. Restore your File Store data by restoring your network share snapshot to your network. Your specific steps will depend on your network.

   Note: Some technologies require you to create a new network share when doing a restore. If this applies to your network attached storage, you can do your restore **before** stopping the Tableau Server. If you are restoring the File Store data to a new network share, you must configure Tableau Server to use the new network share. For more information, see Configure Tableau Server to use a different external storage.

6. Run the following command to restore the KMS and Asset keys:

   ```
   tsm maintenance snapshot-backup restore
   ```

   Note: If you are using a cloud solution for your encryption key solution, make sure the server where the backup is being restored has decrypt access to cloud instance where CMK is deployed.

7. Run the following command to restart Tableau Server:

   ```
   tsm start
   ```

Who can do this

Tableau Server Administrators can backup and restore Tableau Server. In addition you must have permission to access and perform snapshot backups on the external storage.

Performance Considerations for External File Store

This topic lists the factors you must consider when you have External File Store to make sure you have optimal performance.

In this scenario where File Store is configured external to Tableau Server, you are storing the extracts on a network share. This means that Tableau Server will be accessing this data across the network. To ensure optimal performance, we recommend the following:

- Use Enterprise grade storage system to to ensure reliability and high data access performance.
- The storage system supports enough read IOPS:
  - Use Solid State Drives. If using spinning disks is the only option, use the fastest and as many as possible.
- The network infrastructure supports the following:
  - At least 10 GB Ethernet to support high speed data transfers between Tableau Server and the storage system.
  - No more than 10 millisecond storage latency between Tableau Server and the storage system.

  The above recommendations are based on testing done by the Tableau team. Your requirements and performance may vary. We highly recommend that you create your own benchmarks to assess the performance and determine resource requirements.

  When creating benchmarks, consider overall performance of Tableau Server with workbook load times as one of the key metrics. This is especially relevant to this configuration since External File Store mostly impacts extract based workbooks.

  You can use Tabjolt to do your benchmarks.

Who can do this

Tableau Server Administrators monitor Tableau Server performance. However, there are network, hardware, and storage considerations that might either need access to make configuration changes to these resources. You may also choose to work with your network administrator to make any changes if you don't have access to these resources.

Tableau Server External Repository

The Tableau Server Repository is a PostgreSQL database that stores data about all user interactions, extract refreshes, and more.

The repository can be installed as locally on the same nodes as the Tableau Server or installed externally:

Local repository: The PostgreSQL Database is installed and deployed locally, meaning it is deployed along with Tableau Server.

External repository: The PostgreSQL Database is deployed externally. The external repository can be installed on Amazon RDS, Azure Database, Google Cloud, or as a stand-alone installation.

For more information about what Tableau Server Repository is in general, see:

- Workgroups Database
- Collect Data with the Tableau Server Repository

The supported hosts for the external repository are:

- Amazon RDS - Beginning in version 2019.3
- Azure Database - Beginning in version 2020.4
- Stand-alone PostgreSQL Instance - Beginning in version 2021.2
- Google Cloud SQL for PostgreSQL Instance - Beginning in version 2021.4

This topic is an overview of the Tableau Server external repository.

External Repository Considerations

Both Amazon RDS and Azure Database offer better scalability, reliability, high availability and security built-in for PostgreSQL. By integrating more closely with these cloud offerings, you will be able to take advantage of these additional benefits.

If you are setting up a stand-alone PostgreSQL instance, you must set up and manage high availability and scale as needed.

Cloud Platform

When using a cloud platform to host your external repository, you have the option of running PostgreSQL instances on hosted infrastructure and managing them yourself or choosing the fully managed service option.

- **Self-managed:** Setting up and managing PostgreSQL instances on hosted infrastructure yourself. For example, if you are using AWS as your cloud platform, you can use EC2 instances to run, manage and maintain PostgreSQL instances.
- **Fully managed:** Select a fully managed service. For example, if you are using AWS as your cloud platform, you can use the RDS option to host your external repository.

One of the key considerations when making a choice between self-managed and fully managed is that in a self-managed option you have the most control but with it comes the responsibility of maintaining the VMs and many database administrative tasks. A fully managed option offers ease of setup, configuration management, and maintenance.

Here is a more comprehensive list of things to consider when choosing between the two:

- Setup and maintenance requirements.
- High availability and disaster recovery options
- Performance, Scalability, and Monitoring capabilities
- Security maintenance
- Operational costs, service costs, personnel costs.

Here is an example of how the two options can be compared for Azure on the Microsoft site: Choose the right PostgreSQL server option in Azure,

Requirements

- Your Tableau Server must be using the following versions:
  - 2019.3 or later to use with AWS
  - 2020.4 or later to use with Azure.

    For the right version of PostgreSQL to use, see Azure Database for PostgreSQL Flexible Server .

    - 2021.2 or later to use with stand-alone PostgreSQL instance. (Can be used for on-premises, Azure VM, or AWS EC2 installations).
    - 2021.4 or later to use with Google Cloud for both PostgreSQL Instance or a stand-alone PostgreSQL on Google Cloud VM.

- Your Tableau Server must have the Advanced Management keys activated.

- Depending on where you are planning to host the External Repository, you will need to be familiar with one of the following:

    - Amazon RDS Database setup and management.
    - Azure Database setup and management.
    - PostgreSQL database setup and management as a standalone installation.
    - Google Cloud PostgreSQL instance setup and management

Versioning

You must be running the correct version of PostgreSQLfor standalone Tableau Server repository. The following table shows Tableau Server version compatibility.

Note: The maximum compatibility version for any Tableau Server version is the minimum major version or any minor update to that. For example, if the PostgreSQL minimum compatible version is 13.4, the maximum compatible version is 13.x where <x> is equal or higher than 4.

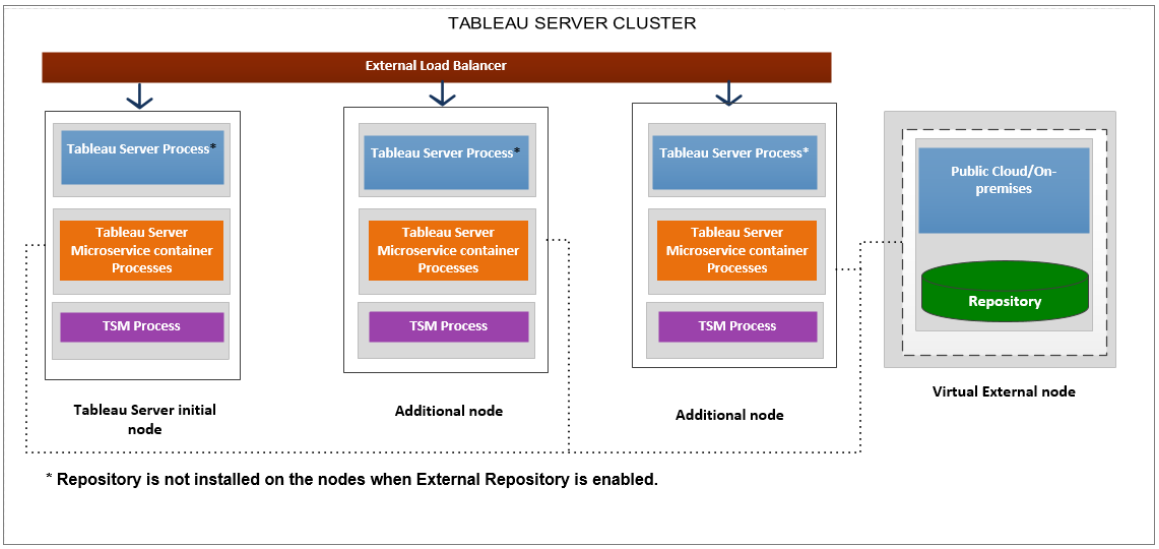
Tableau Server versions	PostgreSQL minimum compatible versions
2021.2.3 - 2021.2.8 2021.3.0 - 2021.3.7 2021.4.0 - 2021.4.3	12.6
2021.2.10 - 2021.2.14 2021.3.8 - 2021.3.13	12.8

2021.4.4 - 2021.4.8	
2021.2.15 - 2021.2.16 2021.3.14 - 2021.3.15 2021.4.9 - 2021.4.10	12.10
2021.2.17 - 2021.2.18 2021.3.16 - 2021.3.17 2021.4.11 - 2021.4.12	12.11
2021.3.26 2021.4.23	12.15
2022.1.0	13.3
2022.1.1 - 2022.1.3	13.4
2022.1.4 - 2022.1.6	13.6
2022.1.7 - 2022.1.16 2022.3.0 - 2022.3.7 2023.1.0 - 2023.1.4	13.7
2022.1.17 - 2022.1.19 2022.3.8 - 2022.3.19 2023.1.5 - 2023.1.15 2023.3.0 - 2023.3.8	13.11
2022.3.20 - 2022.3.x 2023.1.16 - 2023.1.x	13.14

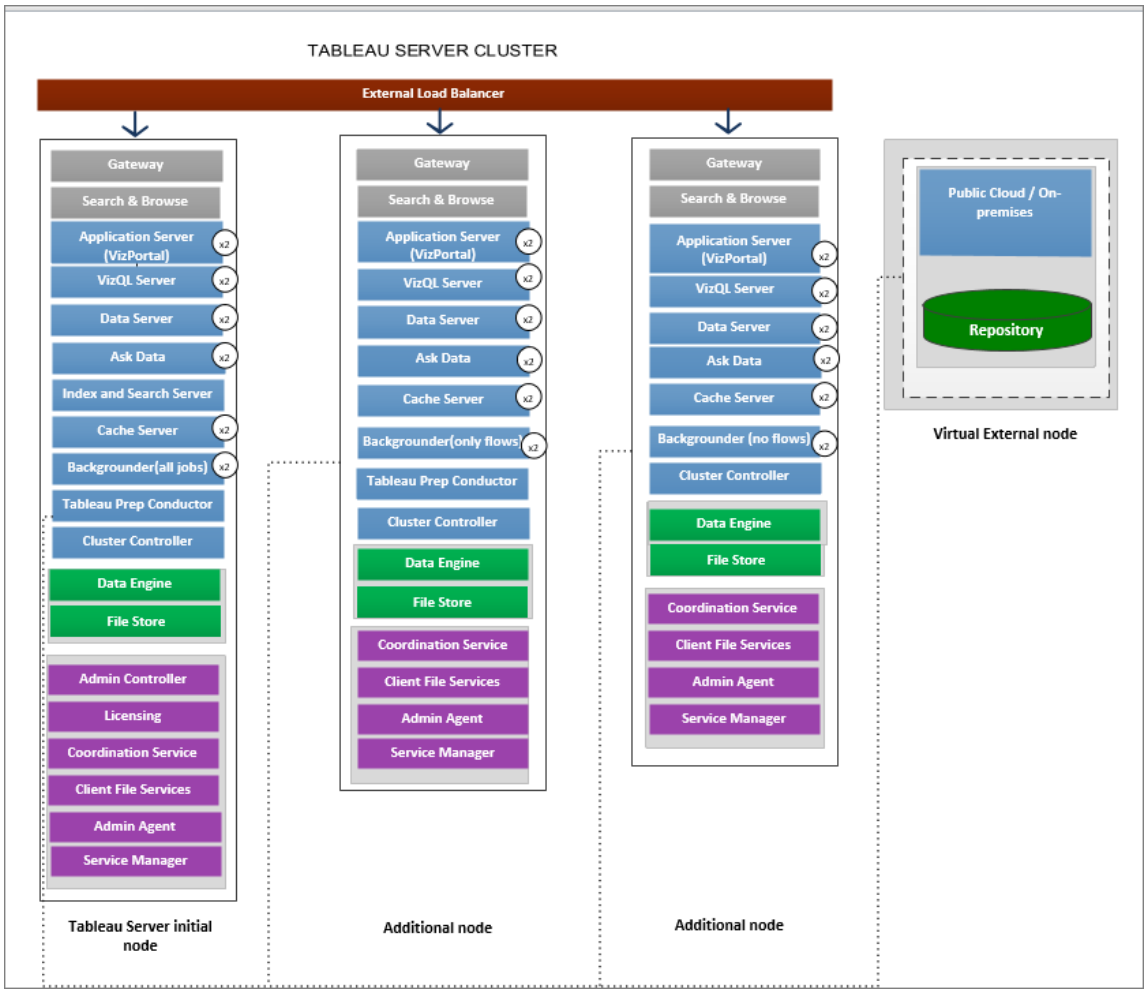
2023.3.9 - 2023.3.x	
2024.0 - 2024.x	15.6

Topology

The diagram below is a summarized version of the Tableau Server topology with External Repository.



The diagram below is a detailed version of the Tableau Server topology with External Repository and shows all the processes installed on each node.



Managing the External Repository

License Management

To enable this feature you must first activate Advanced Management on Tableau Server. For more information, see About Tableau Advanced Management on Tableau Server. If you don't have Advanced Management or if the license is expired, you will see the following behavior:

- If you try to configure Tableau Server to use external repository during installation, you will see an error message, but you will be able to continue the installation and Tableau Server Repository will be installed locally. For single server installations, the repository

is installed on the same machine as Tableau Server. For multi-node installations, the repository is installed on one of the nodes of your Tableau Server cluster.

- If you are already using an external Tableau Server Repository on your Tableau Server installation, and the Advanced Management license expires, the server will fail on restart. If you no longer have Advanced Management capabilities, but have a valid Tableau Server license, you will still be able to create a backup. You can also migrate the external repository to local repository which does not require Advanced Management to get your server up and running again. For more information on how to migrate from external repository to local repository, see Re-Configure Tableau Server Repository.

Supported Migration Scenarios

- Moving your repository from local to external.
- Moving your repository from external to local.

Backup and Restore

If you have only External Repository configured:

The backup and restore process remains is the same for both local and external repository and as described in the Back up Tableau Server Data topic.

- The backup and restore commands work in the same manner for both local and external repository. Backup in the case where Tableau Server uses external repository requires more disk space for backup, so you should keep that in mind when configuring your disk space.
- The default and other supported locations for storing the backup file are the same whether it the repository is local or external.

The following exceptions apply to Tableau Servers using external repository:

- Any custom user accounts that are created in the external Tableau Server repository will be included in the restore, but the passwords for the custom user accounts will not be. The passwords will have to be reconfigured after restore is complete. Custom user accounts are PostgreSQL database user accounts, used by SQL or other database client software to connect to the PostgreSQL database.

  Note: These custom accounts will be disabled as a security measure, but this can be reconfigured.

- Configuration and topology should not be included in the backup. For more information on how to export configuration and topology settings, see Perform a Full Backup and Restore of Tableau Server.

If you have both External Repository and External File Store configured:

You have a couple of options if you have both External Repository and External File Store configured for your Tableau Server. These options allow you to take advantage of the snapshot backup capabilities of the cloud platforms you might be using to host the External Repository and File Store. For more information, and detailed step by step instructions, see Backup and Restore with External File Store.

SSL Connections

You have the option to require or not require TLS/SSL connections from Tableau Server to the External Repository.

If you do not need to use encrypted connections, you must configure the External Repository to allow unencrypted connections, and use the `--no-ssl` option when you configure the External Repository for Tableau Server. For more information, see tsm topology external-services repository enable.

If you want to enable or disable SSL at a later time, post installation, use the tsm security repository-ssl enable or tsm security repository-ssl disable. This option is available starting in 2021.4.

Updating the SSL Certificate

If as part of a planned expiration of the SSL certificate of the RDS or Azure Database, Google Cloud PostgreSQL instance, or a stand-alone PostgreSQL instance, you need to update the instance with the new certificate file, you will also need to update Tableau Server settings to use the new certificate file. You can do this by downloading the latest file and running the *tsm topology external-services repository replace-host* command and providing the new certificate file.

High Availability Considerations

Tableau Server does not manage or setup high availability for the external repository.

- **AWS:** Amazon RDS offers features that can be used to provide high availability and manage failover. For more information, see Amazon RDS High Availability.
- **Azure:** Azure offers features that can be used to provide high availability and manage failover. For more information, see Azure Database High Availability.
- **Google Cloud:** Google Cloud offers features that can be used to provide high availability and manage failover. For more information, see Google Cloud High Availability.
- **Stand-alone PostgreSQL instance:** PostgreSQL offers several features that can be used to provide high availability and manage failover. For more information, see PostgreSQL High Availability.

Upgrade considerations

This applies only if you are using the External Repository configuration with Tableau Server.

If you are using an External Repository, you may need to take additional steps when upgrading:

- **No version change**—If there is no version change in PostgreSQL, there are no special actions required.
- **Minor version change**—If there is a minor version change in PostgreSQL, you need to upgrade your external repository before upgrading Tableau Server. In most cases there

are in-place methods for doing so. The method you use depends on the location of your repository and is beyond the scope of this documentation.

- **Major version change**—If there is a major version change in PostgreSQL, you need to follow the steps described in Upgrade Tableau Server with External Repository for a New Major Version of PostgreSQL.

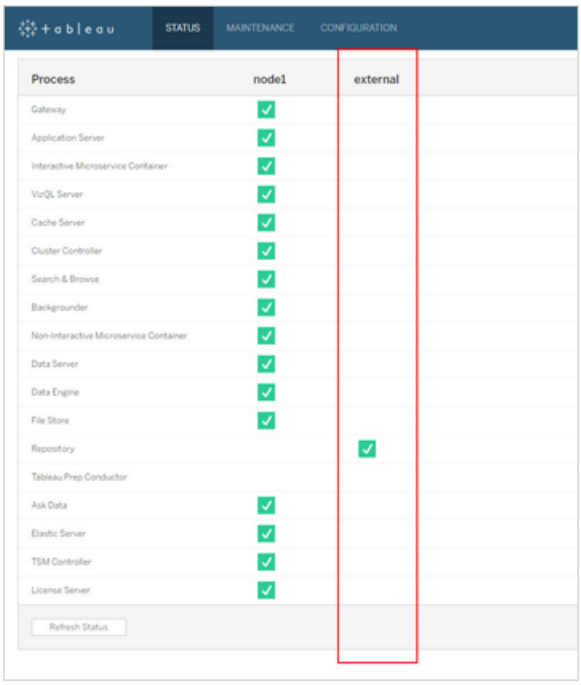
  Steps include:

  1. Creating a new instance of PostgreSQL DB. For more information, see:
     - Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS)
     - Create a Azure Database PostgreSQL Instance on Azure
     - Create a Cloud SQL for PostgreSQL insatnce on Google Cloud
     - Create a PostgreSQL Database as a Stand-alone Installation
  2. Creating a configuration file and download the SSl certificate file for the new instance that you created in Step 1.

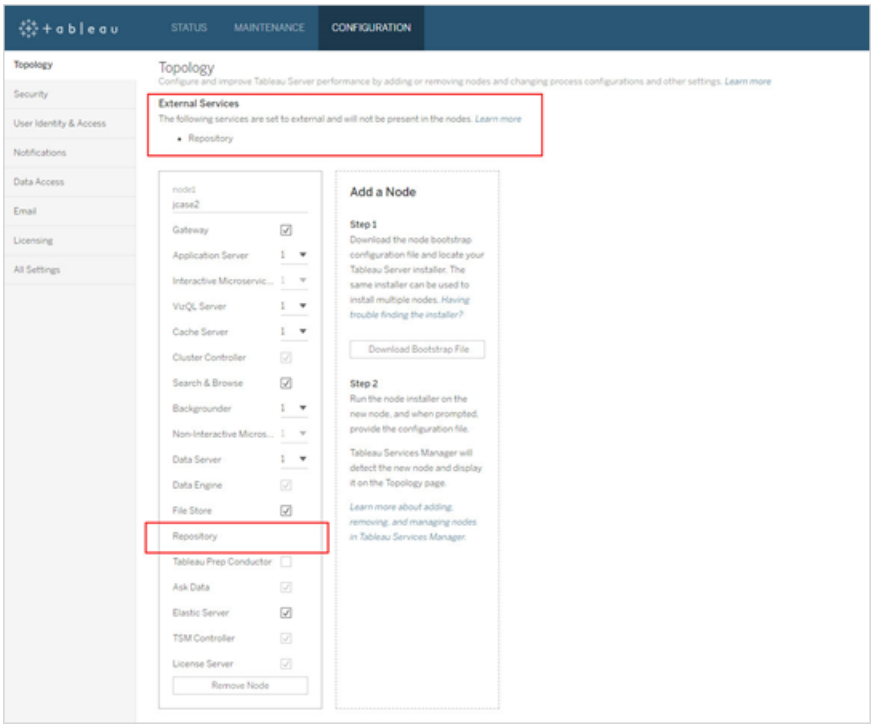
  During upgrade, you will need to point Tableau Server to the new instance using the configuration file. The upgrade process will migrate the content from your current external repository to your new instance. For more details, see Upgrade Tableau Server with External Repository for a New Major Version of PostgreSQL.

Monitoring the Status of the Repository

TSM status page will show the Tableau Server external service as an additional node for your Tableau Server installation.



The topology tab indicates whether there are Tableau Server external services configured:



Getting Logs

Tableau Server logs will not include logs from the external repository. Use the following ways to get the logs specific to your instance:

- **AWS:** For information on setting up logging for your Amazon RDS PostgreSQL instance, see PostgreSQL Database Log Files.
- **Azure:** For information on setting up logging for your Azure Database for PostgreSQL instance see Logs in Azure Database for PostgreSQL.
- **Google Cloud:** For information on how to view and query logs for your PostgreSQL instance, see Cloud SQL logging.
- **Stand alone PostgreSQL Instance:** For information on setting up logging for your stand alone PostgreSQL instance, see Error Reporting and Logging.

Next Steps

- You can use one of the following to create a PostgreSQL database instance:
    - Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS)
    - Create a Azure Database PostgreSQL Instance on Azure
    - Create a PostgreSQL Instance on Google Cloud
    - Create a PostgreSQL Database as a Stand-alone Installation
- Install Tableau Server with External PostgreSQL Repository
- Re-Configure Tableau Server Repository
    - Migrate from local to external
    - Migrate from external to local

Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS)

Beginning in version 2019.3, you can host your External Repository on the AWS Cloud Platform. This topic describes how to create a AWS RDS PostgreSQL DB instance to use as your Tableau Server external repository.

To see a full list of hosts that you can use for the your external repository, see Tableau Server External Repository.

Requirements and Recommendations

- At a minimum use an instance with 8vCPUs and 32GB RAM. This is the minimum recommended AWS RDS instance size to use for Tableau Server external repository, but the exact requirements will vary with your requirements and usage. We recommend using 16 vCPUs and 128GB RAM Amazon RDS memory optimized instance types for good performance.

  You can start with a smaller instance type and if you find later that you need a larger instance type, you can upgrade your existing RDS instance. For more information, see Upgrading your RDS Instance .

  Secure communication between Tableau Server and the external PostgreSQL DB instance using SSL connections is **not** required but recommended.

- The PostgreSQL DB instance must be reachable by all nodes in the Tableau Server cluster. One of the ways you can do this is by making the PostgreSQL DB instance a member of a security group that has the necessary permissions to be accessed by all the nodes in the Tableau Server cluster.

- The version of PostgreSQL should match the version used by Tableau Server when installed locally. Tableau Server 2020.4 uses PostgreSQL version 12.

- Tableau auto-generates passwords for internal use by internal database users. These passwords are 32 characters long and consist of lower-case letters and numbers. They cannot be accessed or configured by an admin. Your external PostgreSQL instance may allow you to set a password policy (this will depend on the platform you are using). If you specify a policy that includes character types other than numbers and lower-case letters, this can cause errors while configuring Tableau Server to use the external repository.

Create a PostgreSQL DB instance on Amazon RDS

Step 1: Create a parameter group

PostgreSQL parameters that you set for a local PostgreSQL instance in the *postgresql.conf* file are maintained in the DB parameter group for your DB instance. When you create a DB instance, the parameters in the associated DB parameter group are loaded.

From the Tableau Server perspective, most of the parameters can be set to defaults. You can modify the parameter values if you have specific performance or logging requirements, but we strongly recommend that the following parameters be left to default values and not be changed:

- `standard_conforming_strings`
- `escape_string_warning`

We also recommend the value for the `work_mem` be set to at least 16384 to help avoid performance issues.

For more information and a full list of PostgreSQL Parameters, see Working with PostgreSQL Parameters, on the AWS site.

Step 2: Create a PostgreSQL DB instance on Amazon RDS

To create a new PostgreSQL DB instance, follow the instructions provided on the Amazon documentation site.

Following are configuration options and the recommended values for the new PostgreSQL DB instance:

- **Instance Specifications**
  - Use the parameter group created in **Step 1**.
  - Use the compatible version of PostgreSQL for the version of Tableau Server you are using. For a full list of PostgreSQL versions compatible with Tableau Server versions, see Product Compatibility.
  - Use DB instance class that is db.m4.2xlarge or larger.
  - Allocate at least 100GB of storage.

    - Storage type and Provisioned IOPS: leave default (recommendations may change depending on load testing).

- **Settings**

    - You **must** use **rails** as the master username.

      This is a requirement for the external repository to work properly with Tableau Server.

    - Pick a password that meets AWS's requirements.

- **Network and Security**

    - Make sure that the RDS instance can be reached by all the Tableau Server nodes. This most often involves creating a security group that allows access from the nodes.

- **Database Options**

    - Don't create an initial database. The Database name should be left blank, as Tableau Server will create the needed databases in the RDS instance.

    - The port can be anything, but we recommend leaving it as the default 5432.

    - Set the DB Parameter Group to the one created in **Step 1**.

    - Leave the IAM DB authentication as disabled.

- **Encryption**

    - You can choose whether or not you want encryption.

- **Backup**

    - This is for AWS's automated backups, not Tableau Server's backups. You can specify the settings that meets the requirements.

- **Monitoring**

    - You can specify the settings based on your requirements.

- **Log Exports**

    - You can specify the settings based on your requirements.

- **Maintenance**

    - Disable auto minor version upgrade. Tableau Server is built to use a specific version of PostgreSQL. and you will be prompted to upgrade the PostgreSQL version if needed, during Tableau Server Upgrade.

- **Delete Protection**

    - You can specify the settings based on your requirements.

Step 3: Get the PostgresSQL DB Instance Endpoint

After creating the PostgreSQL database instance, you can't use it until it's completed initialization by AWS and this can take several minutes. Once the instance is ready, get the endpoint information that you will use to configure Tableau Server to use this instance for the Tableau Server Repository.

Step 4: Download the SSL certificate file (.pem file)

Secure connections between Tableau Server and the External Repository using SSL is **not** required, but recommended.

If you want to set up secure connections between Tableau Server and the External Repository, you will need the .pem file when you configure Tableau Server to use the external DB instance for your Tableau Server Repository. For more information, see Using SSL to Encrypt the Connection to a DB Instance.

If you do not need to use secure connections between Tableau Server and External Repository, you need to configure the RDS instance to allow unencrypted connections.

Important: If as part of a planned expiration of the SSL certificate of the RDS instance, you need to update your RDS instance with the new certificate file, you also need to update

Tableau Server settings to use the new certificate file. You can do this by downloading the latest file and running the *tsm topology external-services repository replace-host* command and providing the new certificate file.

Configuring High Availability for your PostgreSQL DB

Tableau Server does not manage or setup high availability for the external repository. Amazon RDS offers high availability features that can be used to provide high availability, manage fail-over, etc. For more information, see Amazon RDS High Availability.

Disaster Recovery for your PostgreSQL DB

In the event of a disaster, you may need to setup a new RDS instance. There are other scenarios where you may need to recover from an issue with the RDS instance. For example, when you upgrade your Tableau Server, you might also need to upgrade the PostgreSQL version on your RDS instance. In the event that your PostgreSQL upgrade is not successful you might have to use a new RDS instance. In such scenarios, to configure your Tableau Server to use the new RDS instance, use the following steps:

1. **Restore the snapshot to a new RDS instance.** AWS does not support restoring a snapshot to an existing RDS instance. For more on RDS snapshot backup and restore, see Amazon RDS Backup and Restore.

2. **Create a new JSON settings file** containing connection information for the new RDS instance. For more information on creating a JSON settings file, see **Step 1** in Install Tableau Server with External PostgreSQL Repository.

3. **Use the `tsm topology external-services repository replace-host` command** to point your Tableau Server to the new RDS instance.

   For more information on the `tsm topology external-services repository replace-host` command, see tsm topology.

Who can do this

Only Tableau Server Administrators can configure Tableau Server to use the external repository. You will also need an AWS account to create a RDS instance.

Next Steps

For new installs: Install Tableau Server with External PostgreSQL Repository

If you want to configure your existing Tableau Server to use an external repository, see Re-Configure Tableau Server Repository.

Create a Azure Database PostgreSQL Instance on Azure

Beginning in version 2020.4, you can host your External Repository on the Azure Cloud Platform. This topic describes how to create a Azure Database for PostgreSQL instance to use as your Tableau Server external repository.

Requirements and Recommendations

- We recommend that you use 8 vCore memory optimized server with 50 GB of storage for Tableau Server external repository, but the exact requirements will vary with your requirements and usage. If you already have a Tableau Server, review the usage of your existing repository to determine your storage needs.

  You can also scale your resources if you find that you need more. For more information, see Scaling your PostgreSQL Azure Database resources.

- Secure communications between Tableau Server and the external PostgreSQL DB instance using SSL is recommended, but not a requirement.

  If you do not want to use secure connections between Tableau Server and External Repository, you should configure the Azure Database to allow unencrypted connections.

- The PostgreSQL DB instance must be reachable by all nodes in the Tableau Server cluster. The database instance must be set up to allow connections from all the Tableau Server nodes. There are two ways to set this up:

  - This is most secure way: Configure Azure Database for PostgreSQL instance to only allow private access via the Virtual Network service endpoint. For more

information, see Use Virtual Network service endpoints and rules for Azure Database for PostgreSQL and Create and Manage VNet service endpoints.

You may also want to review the overview topic on Azure virtual networks.

- Alternatively Azure Database for PostgreSQL can be configured to allow connections from a range of public IP addresses. This method exposes the Azure Database endpoint to public access on the internet.

- When setting up the Azure Database instance, we recommend using **postgres** as the Administrator user name. If you choose to use a different user name, make sure that the user name does not start with **pg**, or **azure**. The user name also cannot be **rails**, **tblwgadmin**, **tableau**, **readonly**, or **tbladminviews**.

- The version of PostgreSQL should match the version used by Tableau Server when installed locally. Tableau Server 2020.4 uses PostgreSQL version 12.

- Tableau auto-generates passwords for internal use by internal database users. These passwords are 32 characters long and consist of lower-case letters and numbers. They cannot be accessed or configured by an admin. Your external PostgreSQL instance may allow you to set a password policy (this will depend on the platform you are using). If you specify a policy that includes character types other than numbers and lower-case letters, this can cause errors while configuring Tableau Server to use the external repository.

Create a Database PostgreSQL instance on Azure

Step 1: Create a delegated subnet for the Azure Database for PostgreSQL instance

This step is a prerequisite for setting up private access for your networking option when you create the instance. Setting up private access to the database is a must for secure communications. This let the virtual machines created anywhere in that Virtual Network to connect to the database instance, but none outside of the Virtual Network is able to do so.

On the same virtual network where you are currently hosting your Tableau Server, create a new delegated subnet for the Azure Database instance. For more information on setting up

private access see Networking Options for Azure Database for PostgreSQL - Flexible Server on the Azure website.

Step 2: Create an Azure Database for PostgreSQL instance

To create a new Azure Database for PostgreSQL, follow the instructions provided on the Azure documentation site.

Following are configuration options and the recommended values for the new PostgreSQL DB instance:

- **Server Details**
    - Specify None as the Data source to create a new server.
    - For Admin user name, we recommend using **postgres** as the Administrator user name. If you choose to use a different user name, make sure that the user name does not start with **pg**, or **azure**. The user name also cannot be **rails**, **tblwgadmin**, **tableau**, **readonly**, or **tbladminviews**.
    - Pick a password that meets Azure's requirements.
    - Use the compatible version of PostgreSQL for the version of Tableau Server you are using. For a full list of PostgreSQL versions compatible with Tableau Server versions, see Product Compatibility.
    - Allocate at least 512GB of storage.
- **Compute and Storage**
    - At a minimum, use Flexible Server with General Purpose computer tier, and Standard_D8s_v3 (8 vCores, 32 GB RAM) compute size.
- **Network Options**
    - Select Private Access (Virtual Network). This ensures private and secure communications for the database.
- **High availability**
    - Enable the high availability option per your requirements.
- **Backup**

- Set the retention period per your requirements. This is for Azure automated backups, not Tableau Server's backups. You can specify the settings that meets the requirements.

Step 3: Configure a server-level firewall rule

Once the database is created, configure a server-level firewall rule to allow access to the Tableau Server nodes.

Make sure that the Database instance can be reached by all the Tableau Server nodes using the dedicated subnet described in Step 1.

Step 4: Configure the Azure Database for PostgreSQL Instance.

From the Tableau Server perspective, most of the parameters values for the instance can be set to defaults. You can modify the parameter values if you have specific performance or logging requirements, but we strongly recommend that the following parameters be left to default values and not be changed:

- `standard_conforming_strings`
- `escape_string_warning`

We also recommend the value for the `work_mem` be set to at least 16384 to help avoid performance issues.

For information on how to configure server parameters, see this Azure documentation.

Step 5: Get the PostgresSQL DB Instance Endpoint

Once the instance is ready, get the endpoint information that you will use to configure Tableau Server to use this instance for the Tableau Sever Repository.

Step 6: Download the SSL certificate file

Secure communications between Tableau Server and the External Repository using SSL is **not** required but recommended.

If you want to set up secure connections between Tableau Server and the External Repository, download the certificatie file. You will need this certificate file when you configure Tableau Server to use this external repository. For more information, see Configure TLS connectivity for Azure Database for PostgreSQL.

If you do not need to use secure connections between Tableau Server and External Repository, configure the Azure Database instance to allow unencrypted connections.

Configuring High Availability for your PostgreSQL DB

Tableau Server does not manage or setup high availability for the external repository. Azure offers high availability features that can be used to provide high availability. For more information, see Azure Database High Availability.

Disaster Recovery for your PostgreSQL DB

In the event of a disaster, you may need to setup a new Azure Database for PostgreSQL instance. There are other scenarios where you may need to recover from an issue with the database instance. In such scenarios, to configure your Tableau Server to use the new Azure Database instance, use the following steps:

1. **Restore the backup to a new Azure Database instance.** In Azure Database for PostgresQL, performing a restore creates a new server from the original server's backups. For more on Azure Database for PostgreSQL backup and restore, see Azure Database for PostgreSQL Backup and Restore.

2. **Create a new JSON settings file** containing connection information for the new Azure Database for PostgreSQL instance. For more information on creating a JSON settings file, see **Step 1** in Install Tableau Server with External PostgreSQL Repository.

3. **Use the `tsm topology external-services repository replace-host` command** to point your Tableau Server to the new Azure Database for PostgreSQL instance.

For more information on the `tsm topology external-services repository replace-host` command, see tsm topology.

Who can do this

Only Tableau Server Administrators can configure Tableau Server to use the external repository. You will also need an Azure account to create the Azure Database.

Next Steps

For new installs: Install Tableau Server with External PostgreSQL Repository

If you want to configure your existing Tableau Server to use an external repository, see Re-Configure Tableau Server Repository.

Create a PostgreSQL Instance on Google Cloud

Beginning in version 2021.4, you can host Tableau Server External Repository on the Google Cloud Platform. This topic describes how to create a PostgreSQL instance on Google Cloud to use as your Tableau Server external repository.

To see a full list of hosts that you can use for your external repository, see Tableau Server External Repository.

Requirements and Recommendations

- At a minimum, use high memory machine type with 8vCPUs and 32GB RAM. This is the minimum recommended PostgreSQL instance size to use for Tableau Server external repository, but the exact requirements will vary with your requirements and usage. We recommend high memory instance type with 16 vCPUs and 128GB RAM for good performance in most scenarios.
- Secure communication between Tableau Server and the external PostgreSQL DB instance using SSL connections is not required but recommended.

- The PostgreSQL instance must be reachable by all nodes in the Tableau Server cluster. One of the ways you can do this is by making the PostgreSQL instance a member of a security group that has the necessary permissions to be accessed by all the nodes in the Tableau Server cluster.
- The version of PostgreSQL must should be a supported version. For more information, see Product Compatibility for supported version information.
- Tableau auto-generates passwords for internal use by internal database users. These passwords are 32 characters long and consist of lower-case letters and numbers. They cannot be accessed or configured by an admin. Your external PostgreSQL instance may allow you to set a password policy (this will depend on the platform you are using). If you specify a policy that includes character types other than numbers and lower-case letters, this can cause errors while configuring Tableau Server to use the external repository.

Create a Database PostgreSQL instance on Google Cloud

Step 1: Create a new PostgreSQL instance

Create a PostgreSQL instance using the directions provided on the Google website, here.

We recommend using a high memory machine type, with 16 vCPUs, and 128GB RAM.

Step 2: Configure database flags for your PostgreSQL Instance

From the Tableau Server perspective, most of the parameters values for the instance can be set to defaults. You can modify the parameter values if you have specific performance or logging requirements, but we strongly recommend that the following parameters be left to default values and not be changed:

- `standard_conforming_strings`
- `escape_string_warning`

We also recommend the value for the `work_mem` be set to at least 16384 to help avoid performance issues.

For more information on database flags, see this topic on the Google website.

Step 3: Get the PostgreSQL DB Instance Endpoint

Once the instance is ready, get the endpoint information that you will use to configure Tableau Server to use this instance for the Tableau Server Repository.

Step 4: Download the SSL certificate file

Secure communications between Tableau Server and the External Repository using SSL is not required but recommended.

To setup secure connections between Tableau Server and the External Repository, you must use the certificate file when you configure Tableau Server to use this external repository. For more information, see Configuring SSL/TLS certificates on the Google website.

Configuring High Availability for your PostgreSQL DB

Tableau Server does not manage or setup high availability for the external repository. Google Cloud offers high availability features that can be used to provide high availability. For more information, see Enable High Availability on an Instance on the Google website.

Disaster Recovery for your PostgreSQL DB

In the event of a disaster, you may need to set up a new PostgreSQL Database for the PostgreSQL instance. There are other scenarios where you may need to recover from an issue with the database instance. In such scenarios, to configure your Tableau Server to use the new PostgreSQL instance, use the following steps:

1. **Restore the backup to a new PostgreSQL instance.**In the Google Cloud platform, you can choose to either restore to the same instance or create a new instance. For more information, see Restoring an instance on the Google website.

   We recommend creating a new instance and do the following steps to recover.

2. If this is an new instance, **Create a new JSON settings file** containing connection information for the new Azure Database for PostgreSQL instance. For more information

on creating a JSON settings file, see **Step 1** in Install Tableau Server with External PostgreSQL Repository.

3. **Use the `tsm topology external-services repository replace-host` command** to point your Tableau Server to the new Azure Database for PostgreSQL instance.

   For more information on the `tsm topology external-services repository replace-host` command, see tsm topology.

Who can do this

Only Tableau Server Administrators can configure Tableau Server to use the external repository. You will also need an Google Cloud account to create the PostgreSQL database instance.

Next Steps

For new installs: Install Tableau Server with External PostgreSQL Repository

If you want to configure your existing Tableau Server to use an external repository, see Re-Configure Tableau Server Repository.

Create a PostgreSQL Database as a Stand-alone Installation

Beginning in version 2021.2, you can host the Tableau Server repository separately as a stand alone installation. This is different from using a managed cloud service such as AWS RDS, or Azure Database. This configuration can be done on-premises, on AWS EC2, or on an Azure VM. Such an installation of the Tableau Server Repository will be to referred to as stand-alone External Repository.

To see a full list of hosts that you can use for the your external repository, see Tableau Server External Repository.

This topic provides guidance on the requirements and configurations that are necessary for Tableau Server to connect to a PostgreSQL installation and use it as the Tableau Server External Repository. This topic does not provide you with the detailed instructions on how to

install PostgreSQL We recommend that you follow the documentation on the PostgreSQL site for this information.

Requirements and Recommendations

- **Hardware Recommendations:** CPU and storage depend on your requirements. For smaller installations, you should have at least 50 GB of disk storage, and a quad processor (or 4 virtual cores) system with 32 gig of RAM. Review the guidance in this topic to calculate the disk space requirements for backup and restore. The general recommendation is to start with more hardware resources and scale back after monitoring.
- **Networking:** The PostgreSQL database instance must be reachable from all nodes in the Tableau Server cluster. One of the ways you can do this is by making the PostgreSQL database instance a member of a security group that has the necessary permissions to be accessed by all the nodes in the Tableau Server cluster.
- **Version Compatibility:** The version of PostgreSQL should match the version of the Tableau Server Repository when installed locally. For more information about compatibility, see Product Compatibility.
- **Security:** Secure connections between Tableau Server and the External Repository using SSL is **not** required, but recommended.

  If you do not want to set up secure connections between Tableau Server and External Repository, you should configure the stand-alone PostgreSQL Database to allow unencrypted connections.

Create a stand-alone PostgreSQL Database Instance

Step 1: Install and initialize PostgreSQL

1. Use the PostgreSQL documentation to install PostgreSQL database instance to serve as the External Repository for Tableau Server. You may want to set up a PostgreSQL database cluster to meet any high availability requirements you may have.
2. Install the contrib package that includes the uuid-ossp extension. This module is used to

generate the UUIDS that Tableau Server uses for keys in the database.

3. Initialize the PostgreSQL instance.

Step 1: Configure your PostgreSQL Instance

You will be using two configuration files to configure your PostgreSQL instance:

- `pg_hba` : This is the configuration file for host-based authentication.
- `postgresql.conf`: This is the general server configuration file.

By default these files are located here:

`/var/lib/pgsql/12/data` (This may be different depending on the distribution)

Super User Settings

Choose a user name that meets your requirements. We recommend using **postgres** as the Administrator user name. If you choose to use a different user name, make sure that the user name does not start with **pg**. The user name also cannot be **rails**, **tblwgadmin**, **tableau**, **readonly**, or **tbladminviews**.

This is a requirement for the external repository to work properly with Tableau Server.

Network and Security

Make sure that the database instance can be reached by all the Tableau Server nodes. This most often involves creating a security group that allows access from the nodes.

Database Options

The port can be anything, but we recommend leaving it as the default `5432`.

Update Parameters

From the Tableau Server perspective, most of the parameters values for the instance can be set to defaults. You can modify the parameter values if you have specific performance or log-

ging requirements, but we strongly recommend that the following parameters be set to default values and not be changed:

- `standard_conforming_strings`
- `escape_string_warning`

We also recommend the value for the `work_mem` be set to at least `16384` to help avoid performance issues.

Configure remote connections

Use the following steps to make updates to the configuration files:

1. By default, the configuration in the `postgresql.conf` is configured to only listen to local connections. Enable remote connections by making the following changes in the connections and authentication section of the `postgresql.conf` file:

   Add this line to allow remote connections:

   ```
   listen_addresses = '*'
   ```

2. Restart the PostgreSQL instance.

Configure SSL

Secure connections between Tableau Server and the External Repository is **not** required, but recommended.

To configure encrypted connections between Tableau Server and the External Repository, follow the guidance and the detailed steps described below:

When configuring Tableau Server to use the stand-alone PostgreSQL database instance, you will need to provide a trusted root certificate authority (CA) which is used to verify the connection to the server. Ideally, the stand-alone PostgreSQL instance's server certificate should specify a resolvable hostname so Tableau Server can use sslmode, **verify-full**. This mode verifies that the PostgreSQL server's certificate was signed by a trusted CA and that the hostname in the PostgresSQL Server's certificate matches the hostname used to connect to the

PostgreSQL instance. However, if that is not possible, sslmode, **verify-ca** will just verify that the Postgres server's certificate was signed by a trusted CA.

The following procedure provides the general steps to generating a root CA certificate on the PostgreSQL Server. For more detailed information, read the SSL documentation on the PostgreSQL website (The link points to version 12):

1. Generate signing root certificate authority (CA) key.
2. Create the root CA certificate.
3. Create the certificate and related key (for example - `server.csr` and `server.key`) for the PostgreSQL Server. The subject name for the certificate must match the DNS name of the PostgreSQL Server. The subject name is set with the -subj option with the format "`/CN=<private DNS name>`".
4. Sign the new certificate with the CA certificate that you created in step 2.
5. Copy the crt and key files to the data directory (`/pgsql/<version>/data`).
6. The `pg_hba.conf` file controls the connections to the database. Add the following line to allow remote connections. For example:

   ```
   host all all 10.0.0.0/8 md5
   ```

7. To enable SSL add or update the `postgresql.conf` file with:

   ```
   ssl = on
   ```

   To restrict connections to only SSL only, use *hostssl* instead of *host*.

High Availability and Disaster Recovery

Tableau Server does not manage or setup high availability for the external repository. PostgreSQL database supports several solutions for these purposes including replication and log-shipping. For more information, see the high availability documentation on the PostgreSQL website.

In the event of a disaster, if you need to setup a new PostgreSQL instance, make sure to follow these steps to configure Tableau Server to use the new instance.

1. **Create a new JSON settings file** containing connection information for the new RDS instance. For more information on creating a JSON settings file, see **Step 1** in Install Tableau Server with External PostgreSQL Repository.

2. **Use the `tsm topology external-services repository replace-host` command** to point your Tableau Server to the new PostgreSQL instance.

   For more information on the `tsm topology external-services repository replace-host` command, see tsm topology.

Who can do this

Only Tableau Server Administrators can configure Tableau Server to use the external repository. If you are using AWS EC2 or Azure VM to set up a stand-alone External Repository, you need to have accounts to access these platforms.

Install Tableau Server with External PostgreSQL Repository

This topic describes how to install and configure Tableau Server to use an external service for Tableau Server Repository.

Before you install

- You must have Advanced Management activated on your Tableau Server. For more information about Advanced Management, see About Tableau Advanced Management on Tableau Server.

- Your Tableau Server environment must be one of the following:

  - Public Cloud Services:
    - AWS cloud services.
    - Azure cloud services. For more information on Tableau Server installation on Azure, see Install Tableau Server on Microsoft Azure.

    - Google cloud services. For more information on Tableau Server installation on Google Cloud, see Install Tableau Server for Healthcare on the Google Cloud Platform.
  - On-premises: This is Tableau Server running on the hardware located in your organization or company and not on a public cloud.

- You must have an instance of PostgreSQL database install and ready. You will also need the endpoint of your PostgreSQL DB instance.
  - Tableau Server on **AWS**
    - For a fully managed Server option, using Amazon RDS, follow the guidance detailed in Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS).
    - For a self-managed Server option: Use AWS EC2, and Create a PostgreSQL Database as a Stand-alone Installation
  - Tableau Server on **Azure**:
    - For a fully managed Server option using Azure DB, see Create a Azure Database PostgreSQL Instance on Azure.
    - For a self-managed Server option use Azure VM, and Create a PostgreSQL Database as a Stand-alone Installation .
  - Tableau Server on **Google Cloud**:
    - For a fully managed Server option using Google Cloud PostgreSQL instance, follow the guidance in Create a PostgreSQL Instance on Google Cloud
    - For a self-managed Server option use Google Cloud VM, and Create a PostgreSQL Database as a Stand-alone Installation
  - If you are installing this on-premises, see Create a PostgreSQL Database as a Stand-alone Installation

- Download the SSL certificate:

  Secure connections between Tableau Server and the External Repository are **not** required, but recommended.

  If you want to set up SSL connections for communications between Tableau Server and the External Repository, do the following:

- Amazon RDS: See Using SSL to Encrypt the Connection to a DB Instance.
- Azure Database for PostgreSQL: See Configure TLS connectivity for Azure Database for PostgreSQL.
- Google Cloud database: See Configure SSL/TLS certificates.
- Stand alone PostgreSQL database: The CA certificate that you used to configure SSL for the database should be copied to the Tableau Server initial node. For more information on configuring SSL for your PostgreSQL database, see Configure SSL .

Install and Configure Tableau Server

Step 1: Create a configuration file

Create a json file with the following configuration settings:

```
{
 "flavor":"<flavor name>",
 "masterUsername":"<admin user name>",
 "masterPassword":"<password>",
 "host":"<instance host name>",
 "port":5432
}
```

- **flavor:** This is the type of external service you are going to use for Tableau Server repository.
  - Amazon RDS: use “rds”
  - Azure Database: use "azure"
  - Google Cloud Database: use "gcp"
  - Stand-alone PostgreSQL database: use "generic"
- **masterUsername:**
  - Amazon RDS: Use "rails" for the user name. This is the user that you specified when creating the RDS instance.

    You must use "rails" as the masterUsername. This is required for the external repository to work with Tableau Server properly.

    - Azure Database, Google Cloud PostgreSQL instance, or Standalone PostgreSQL Database: Choose a user name that meets your requirements. We recommend using **postgres** as the Administrator user name. If you choose to use a different user name, make sure that the user name does not start with **pg**, or **azure**. The user name also cannot be **rails**, **tblwgadmin**, **tableau**, **readonly**, or **tbladminviews**.

- **masterPassword:** This is the same password you specified when creating the PostgreSQL database instance.
- **host:** This is the endpoint of your PostgreSQL database instance.
- **port:** The database port you specified when creating the PostgreSQL DB instance.

Step 2: Install Tableau Server and Configure the External Repository

Using TSM CLI:

1. Install and Initialize TSM: Follow the instructions provided in this topic and complete steps 1-5 which runs the setup program and installs TSM.
2. Activate and Register Tableau Server: Provide the Tableau Server Key and the Advanced Management key in the activate step. You will need to run the following command twice, first with the Tableau Server product key and then with the Advanced Management product key:

   ```
   tsm licenses activate -k <product key>
   ```

3. Configure Initial Node Settings: Follow the instructions provided in the topic to configure the initial node settings.

   Important! Do not run the **Initialize and Start Tableau Server** step when you configure the initial node. After completing the other steps in the Configure Initial Node Settings topic, return to this page and follow the rest of the instructions.

4. Configure Tableau Server to use the external repository by using the following commands:

- Specify the external repository settings using the json file that you created in the previous step:

  ```
  tsm topology external-services repository enable -f <file-name>.json -c <ssl certificate file>
  ```

  The json file is the file that you created in the first step with the configuration settings.

  > **Note:** The SSL certificate is needed only if you are using encrypted connections between Tableau Server and the External Repository. If this is not a must for you, you must specify the --no-ssl option. In this case, the tsm command would look like this:
  > ```
  > tsm topology external-services repository enable -f <file-name>.json --no-ssl
  > ```

- Apply the changes:

  ```
  tsm pending-changes apply
  ```

Step 3: Complete tsm Initialize

To initialize and start Tableau Server:

```
tsm initialize --start-server --request-timeout 1800
```

Step 4: Complete the install

Add an Administrator Account and complete the installation.

Who can do this

Tableau Server Administrators can install and configure Tableau Server.

Re-Configure Tableau Server Repository

Your Tableau Server may be configured to use either a local or an external repository. This topic describes the steps needed to reconfigure your existing Tableau Server with one of the

following options:

- Move a local Tableau Server Repository to an external repository and configure your Tableau Server to use an external repository.
- Move the external Tableau Server Repository to your local Tableau Server installation, and configure your Tableau Server to use the local repository. This means that the Tableau Server repository will be installed on the same machine or machines as your Tableau Server.

To learn more about these options and external repositories, see Tableau Server External Repository.

Move local repository to external

Tableau Server must be stopped to migrate from a local repository to an external repository.

Use the following steps to move Tableau Server Repository from local to external:

1. Activate the Advanced Management product key on your Tableau Server if it is not already activated. Advanced Management license is required to configure your Tableau Server with an external repository.
2. Configure Amazon PostgreSQL DB instance to use as the external repository.
    1. Amazon: Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS).
    2. Azure Database: Create a Azure Database PostgreSQL Instance on Azure.
    3. Google Cloud Database: Create a PostgreSQL Instance on Google Cloud
    4. Stand-alone PostgreSQL Instance: Create a PostgreSQL Database as a Stand-alone Installation .
3. Create a json file with the following configuration settings:

```
{
 "flavor":"<flavor name>",
 "masterUsername":"<admin user name>",
 "masterPassword":"<password>",
 "host":"<instance host name>",
```

```
 "port":5432
}
```

- **flavor:** This is the type of external service you are going to use for Tableau Server repository.
  - Amazon RDS: use “rds”
  - Azure Database: use "azure"
  - Google Cloud Database: use "gcp"
  - Stand-alone PostgreSQL database: use "generic"
- **masterUsername:**
  - **Amazon RDS:** Use "rails" for the user name. This is the user that you specified when creating the RDS instance.

    You must use "rails" as the masterUsername. This is required for the external repository to work with Tableau Server properly.
  - **Azure Database, Google Cloud instance, and Stand-alone PostgreSQL instance:** Choose a user name that meets your requirements. We recommend using **postgres** as the Administrator user name. If you choose to use a different user name, make sure that the user name does not start with **pg**, or **azure**. The user name also cannot be **rails**, **tblwgadmin**, **tableau**, **readonly**, or **tbladminviews**.
- **masterPassword:** This is the same password you specified when creating the PostgreSQL database instance.
- **host:** This is the endpoint of your PostgreSQL database instance.
- **port:** The database port you specified when creating the PostgreSQL DB instance.

4. Run the following TSM CLI command to configure Tableau Server to use external repository:

```
tsm topology external-services repository enable -f file.json -
c <ssl certificate file>.pem
```

> **Note:** The SSL certificate is needed only if you are using encrypted connections between Tableau Server and the External Repository. If this is not a requirement for you, you must specify the `--no-ssl` option. In this case, the tsm command would look like this:
> `tsm topology external-services repository enable -f <file-name>.json --no-ssl`

The json file is the file that you created in the first step with the configuration settings. The SSL certificate file can be downloaded as described in this topic.

Running the above command will migrate the local repository to your new external PostgreSQL DB instance.

Move external repository to local

Use the following steps to move Tableau Server Repository from external location to the local installation:

1. Run the following TSM CLI command to move the repository to a specific node:

   ```
   tsm topology external-services repository disable -n nodeN
   ```

2. If you are setting up HA for your repository, install the repository on a second node. For more information, see Example: Install and Configure a Three-Node HA Cluster.

   > **Note:** To install the repository on a second node, you must run the command described in the previous step first. The first step migrates your external repository to the local repository. You can then install the repository on a second node on your Tableau Server.

Who can do this

Tableau Server Administrators can reconfigure external repository. You will also need to have access to create PostgreSQL database instance on Amazon or Azure.

Upgrade Tableau Server with External Repository for a New Major Version of PostgreSQL

When there is a change in the PostgreSQL major version requirement for Tableau Server, there are some specific instructions you must follow to ensure that your Tableau Server upgrade is successful. For example, Tableau Server 2020.4 requires that the repository use PostgreSQL version 12. This is a major version change from PostgreSQL version 9.x used in Tableau Server versions earlier than 2020.4. So if you are upgrading from an earlier version of Tableau Server to version 2020.4 or later, you will need to take the steps described in the following sections to complete the upgrade.

This topic also includes the product compatibility between PostgreSQL and Tableau Server.

Before you upgrade

You cannot do an in-place upgrade to update the PostgreSQL version on your existing PostgreSQL DB instance on Amazon RDS or Azure Database for PostgreSQL DB. Instead, you must create a new instance and point the Tableau Server to the new instance during upgrade. Use the following information to create a new instance and prepare for the upgrade:

1. Create a new instance of PostgreSQL Database instance:
    1. Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS)
    2. Create a Azure Database PostgreSQL Instance on Azure
    3. Create a PostgreSQL Instance on Google Cloud
    4. Create a PostgreSQL Database as a Stand-alone Installation
2. SSL connections are **not** required but recommended. If you want to setup SSL connections for communications between Tableau Server and the External Repository, do the following:
    - **Amazon RDS:** See Using SSL to Encrypt the Connection to a DB Instance.
    - **Azure Database:** See Configure TLS connectivity for Azure Database for PostgreSQL.

- **Google Cloud Instance:** See Configuring SSL/TLS certificates
- **Stand-alone PostgreSQL Instance:** See Configure SSL .

3. Create a configuration file for the new instance you created in step 1.

   Create a json file with the following configuration settings:

```
{
 "flavor":"<flavor name>",
 "masterUsername":"<admin user name>",
 "masterPassword":"<password>",
 "host":"<instance host name>",
 "port":5432
}
```

   - **flavor:** This is the type of external service you are going to use for Tableau Server repository.
     - Amazon RDS: use “rds”
     - Azure Database: use "azure"
     - Google Cloud Database: use "gcp"
     - Stand-alone PostgreSQL database: use "generic"
   - **masterUsername:**
     - **Amazon RDS:** Use "rails" for the user name. This is the user that you specified when creating the RDS instance.

       You must use "rails" as the masterUsername. This is required for the external repository to work with Tableau Server properly.
     - **Azure Database, Google Cloud Instance or Stand-alone PostgreSQL Instance:** Choose a user name that meets your requirements. We recommend using **postgres** as the Administrator user name. If you choose to use a different user name, make sure that the user name does not start with **pg**, or **azure**. The user name also cannot be **rails**, **tblwgadmin**, **tableau**, **readonly**, or **tbladminviews**.

- **masterPassword:** This is the same password you specified when creating the PostgreSQL database instance.
- **host:** This is the endpoint of your PostgreSQL database instance.
- **port:** The database port you specified when creating the PostgreSQL DB instance. Default port for PostgreSQL is 5432.

Tableau Server Upgrade

> **Note:** If you are using Tableau Server External Repository, you must upgrade Tableau server using the command line option.

The following are the high level steps of how to specify the external repository parameters when running the upgrade script.

For a full walk-through of Tableau Server upgrade process, see Upgrading from earlier versions

1. On the Tableau Server, open a command prompt as administrator.

   > **Note:** You must open a new command window because the Setup program updates the path for the new installation.

2. Navigate to the scripts folder for your new installation.

   By default:

   ```
   /opt/tableau/tableau_server/packages/scripts.<version_code>/
   ```

3. Run the upgrade script and specify the configuration file and the SSL certificate:

   ```
   upgrade-tsm --external-repository-config-file=<json config
   file> --external-repository-cert-file=<SSL certificate file>
   ```

Product Compatibility

The table below lists the version of PostgreSQL that is supported with Tableau Server. Use this table to determine the version of PostgreSQL to install for your External Repository.

Amazon RDS support: External Repository using Amazon RDS is supported on Tableau Server versions 2019.3 and later.
Azure Database support: External Repository using Azure Database instance is supported on Tableau Server versions 2020.4 and later.

Google Cloud support: External Repository using Google Cloud SQL intance is supported on Tableau Server versions 2021.4 and later.
Stand alone PostgreSQL instance support: External Repository using a stand alone installation of PostgreSQL database is supported on Tableau Server versions 2021.2 and later.

Google Cloud Platform support: External Repository using a PostgreSQL instance on Google Cloud Platform is supported on Tableau Server versions 2021.4 and later.

Tableau Server versions	PostgreSQL minimum compatible versions
2021.2.3 - 2021.2.8 2021.3.0 - 2021.3.7 2021.4.0 - 2021.4.3	12.6
2021.2.10 - 2021.2.14 2021.3.8 - 2021.3.13 2021.4.4 - 2021.4.8	12.8
2021.2.15 - 2021.2.16 2021.3.14 - 2021.3.15	12.10

2021.4.9 - 2021.4.10	
2021.2.17 - 2021.2.18 2021.3.16 - 2021.3.17 2021.4.11 - 2021.4.12	12.11
2021.3.26 2021.4.23	12.15
2022.1.0	13.3
2022.1.1 - 2022.1.3	13.4
2022.1.4 - 2022.1.6	13.6
2022.1.7 - 2022.1.16 2022.3.0 - 2022.3.7 2023.1.0 - 2023.1.4	13.7
2022.1.17 - 2022.1.19 2022.3.8 - 2022.3.19 2023.1.5 - 2023.1.15 2023.3.0 - 2023.3.8	13.11
2022.3.20 - 2022.3.x 2023.1.16 - 2023.1.x 2023.3.9 - 2023.3.x	13.14
2024.0 - 2024.x	15.6

Legacy compatibility

The table below includes two columns:

1. PostgreSQL version shipped with Tableau Server, which is the version that is installed with Tableau Server for local Repositories.
2. All PostgreSQL versions supported for External Repository.

Notes:
- For PostgreSQL versions earlier than version 10, the first two digits indicate the major version, and the minor version is represented by the last digit. For example, in the version 9.4.1, 9.4 indicates the major version and the .1 indicates the minor version.
- For PostgreSQL versions 10 or later, the first digit indicates the major version, and the minor version is the represented by the last digit. For example, in the version 11.1, 11 is the major version and .1 is the minor version.
- A PostgreSQL version with a major version equal to, and a minor version greater than, the Postgres version shipped with Tableau server is always acceptable for use with the external repository, along with any explicitly allowed older versions.

Tableau Server Version	PostgreSQL Version (Shipped with Tableau Server)	Alternate PostgreSQL Versions supported for External Repository
2019.3 - 2019.3.3	9.6.11	9.6.x, where x is greater than 11
2019.3.4 - 2019.3.10	9.6.15	9.6.x, where x is greater than 15
2019.3.11 - 2019.3.14	9.6.17	9.6.15, or 9.6.x, where x greater than 17
2019.4 - 2019.4.1	9.6.14	9.6.x, where x is greater than 14
2019.4.2 - 2019.4.6	9.6.15	9.6.x, where x is greater than 15

2019.4.7 - 2019.4.13	9.6.17	9.6.15, or 9.6.x, where x is greater than 17
2020.1 - 2020.1.6	9.6.15	9.6.x, where x is greater than 15
2020.1.7 - 2020.3.2	9.6.17	9.6.15, or 9.6.x, where x is greater than 17
2020.4 - 2021.4	12.8	12.8, or 12.x, where x is greater than 8
2022.1 and later	13.3	13.3 or 13.x, where x is greater than 3

Who can do this

Tableau Server Administrators can upgrade and configure Tableau Server.

Upgrading your RDS Instance

If you find that the current RDS instance you are using to host Tableau repository is a performance bottleneck, you can upgrade your RDS instance to a larger size. This topic describes the steps that you can use to upgrade your RDS instance.

1. Back up Tableau Server Data.
2. Stop Tableau Server:

   ```
   tsm stop
   ```

3. After confirming that the server has shutdown, sign in to the AWS Management Console and open the Amazon RDS console at

   https://console.aws.amazon.com/rds/.

4. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to modify.
5. Choose **Modify**. The Modify DB Instance page appears.

6. Modify your RDS instance by setting the DB instance class to the one you want.
7. Choose **Apply immediately** to make sure that the changes are applied right away. For more detailed information, see Modifying a DB Instance Running the PostgreSQL Database Engine on AWS documentation site.
8. Monitor the status of the RDS instance in the AWS console. It may take a while, but when the status shows as Available, you can start Tableau Server and resume normal operations:

   ```
   tsm start
   ```

Who can do this

Tableau Server Administrator who has have access to the Amazon RDS account can perform all the steps required to upgrade the Amazon RDS instance.

Workload Management through Node Roles

Using node roles, you can configure where certain types of workloads are processed on your Tableau Server installation. The node roles features allows you to dedicate and scale resources to specific workloads. You can configure node roles for Backgrounder and File Store.

The Backgrounder node role specifies the type of background tasks that should run on a node, whereas the File Store node role specifies the type extract workload that should run on a node. Both node roles are specified at a node level. Although these node roles can work independently to optimize selected workload, the two node roles in combination can be used to specialize server nodes to preferentially execute selected workloads to optimize performance extract heavy workloads. This combination is discussed in more detail later in the File Store node roles section.

Backgrounder node roles

The Backgrounder process runs Tableau Server tasks, including extract refreshes, subscriptions, flow tasks, ‘Run Now’ tasks, and tasks initiated from *tabcmd*. Running all these tasks can use a lot of machine resources. If you have more than one Backgrounder node in

your cluster, you can manage your Backgrounder workload by specifying the type of tasks a Backgrounder can run on a node using the Backgrounder node role feature.

This configuration option is currently available only through TSM CLI commands and is only useful on multi-node clusters. If you have only one node, the Backgrounder is set to run all tasks by default and that cannot be changed.

Using Backgrounder node roles

The Backgrounder node role feature is intended to give you more control and governance over where certain type of Backgrounder workloads are processed in your Tableau Server installation and allows you to dedicate and scale resources to specific workloads.

For example, if your deployment is heavy on extract and users are running a lot of extract refreshes or encryption jobs, it could be beneficial to dedicate a node to extract refreshes. Similarly, in the case of subscriptions, if your Tableau Server installation processes a lot of subscriptions and you want to ensure that other jobs do not take resources from subscriptions, then you can dedicate a node to subscriptions. In these cases, you would also want to dedicate other backgrounder nodes to workloads other than extract refreshes or subscriptions.

To support high availability, Tableau recommends having multiple nodes that are dedicated towards a specific workload. For example, if you dedicate a node to extract refreshes, you should also configure a second node to process extract refresh workload. This way if a node dedicated to extract refreshes becomes unavailable, extract refreshes can still be processed by the other node.

Configuration options

Configuration	**Jobs**
all-jobs (default)	All Tableau Server jobs
flows	Flow run jobs.
no-flows	All jobs except flows.
extract-refreshes	Jobs that are created for:

	Incremental refreshes, full refreshes, encryption and decryption of all extracts including extracts that flow outputs create.
subscriptions	Subscription jobs
system	System maintenance jobs that interact with other Tableau Server processes. For example, cleaning crashed jobs, reaping database events, and synching Active Directory.
extract-refreshes-and-subscriptions	Extract-refreshes, encryption and decryption of all extracts including extracts that flow outputs generate, and subscription jobs.
no-extract-refreshes	All jobs except extract-refreshes, extract encryption and decryption of all extracts including extracts created from flow outputs.
no-subscriptions	All jobs except subscriptions.
no-extract-refreshes-and-subscriptions	All jobs except extract-refreshes, encryption and decryption of all extracts including extracts created from flow outputs, and subscriptions.
no-system	All jobs except system maintenance jobs.

For more information on how to use the tsm commands to set the node role, see tsm topology.

Note: Making configurations to node roles require a restart of the server and will require some downtime. For more information, see tsm pending-changes.

License requirements

Configuring a node to do only a specific type of tasks, like, flows, extract refreshes, and subscriptions, you must have one of the following licenses activated on your Tableau Server:

- To configure a node to run flows, you must have a valid Data Management license activated on your server, and have Tableau Prep Conductor running on that node. To learn more about Tableau Prep Conductor, see Tableau Prep Conductor.
- To configure a node to run extract refreshes, subscriptions, and any combination related to extract refreshes and subscriptions you must haveAdvanced Management capabilities enabled on your Tableau Server. If the license expires or is deactivated, you will see an error any time you make a change to the Server configuration. For more information on Advanced Management. see About Tableau Advanced Management on Tableau Server.

Important!
While flows, extract refreshes, and subscriptions can be expensive and resource heavy, they are not the only jobs that may require dedicated resources. In the **all jobs** group, there are a variety of System jobs that the Backgrounder executes, such as thumbnail generation for workbooks. Make sure that the nodes that run jobs other than extract refreshes, subscriptions, or flows have enough machine resources.

For more information on configuring node roles using TSM commands, see tsm topology set-node-role.

Considerations

There are some rules you must consider when configuring Backgrounder node roles, which are listed below:

- Only one node role configuration can be set for a node at a time. You cannot configure multiple node roles on a node.
- To configure a node role, there must be at least one Backgrounder process on that node.
- If you have only one Backgrounder node, you must configure this node to run all jobs. This is the default configuration and does not require additional licensing.

- If you have more than one Backgrounder node, combined, they must be configured to handle all jobs. This can be achieved in the following ways:
    - Configure one of the nodes to run all jobs using the all jobs option. This is the easiest and most straightforward way.
    - Using one of the exception configurations on one of the nodes:
        - no-flows
        - no-subscriptions
        - no-extract-refreshes
        - no-extract-refreshes-and-subscriptions

For example, in a cluster where there are three backgrounders, you could have one node configured to run flows, one to run subscriptions and extract refreshes, and one to run all jobs except flows, subscription and extract refreshes.

Note: The ability to specify node roles to run flows, or run all jobs except flows, or run all jobs was introduced in 2019.1.

File Store node roles

The Tableau Server File Store controls the storage of extracts. There are three broad categories of workloads that are extract dependent.

Extract Workload	**Execution Service**
Refresh	Backgrounder
Query	Data Engine
Backup/Restore	Backup/Restore

File Store node role management in combination with Backgrounder node role management gives server admins the ability to specialize server nodes to preferentially execute selected workloads to optimize performance of all categories of extract heavy workloads.

It is possible to specialize a node to execute extract query workloads through a topology that has only stand-alone Data Engine nodes. For more information, see Optimize for Extract Query-Heavy Environments. However, this is at the expense of extract refresh workloads, which are executed by Backgrounder nodes. With the topology-based isolation approach, extract refresh heavy Backgrounder workloads can get slower as none of the Backgrounder nodes have a File Store and thus all extract refresh traffic goes over the network.

With the File Store Node Role configuration option, it is possible to designate certain server nodes that process extract queries to be preferentially selected from the list of server nodes that can do so. This helps speed up workloads such as backup and extract refreshes by allowing server admins to enable File Store on Backgrounder server nodes, which prevents extract queries from running on these nodes. This feature is useful if you have an extract-heavy query workload and an extract-heavy refresh workload and want to achieve optimal extract query and refresh performance.

Guidelines to optimize for extract refresh and backup or restore workloads.

Start from a topology with specialized Data Engine nodes (see Optimize for Extract Query-Heavy Environments).

Note: In the below diagram and procedure, node 1 is Initial Node, node 2 is Additional Node 1, node 3 is Additional Node 2, and node 4 is Additional Node 3.

Process	Initial Node	Additional Node 1	Additional Node 2	Additional Node 3
Cluster Controller	✓	✓	✓	✓
Gateway	✓	✓		
Application Server	✓	✓		
VizQL Server	✓ ✓	✓ ✓		
Cache Server	✓ ✓	✓ ✓		
Search & Browse	✓	✓		
Backgrounder	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓		
Data Server	✓ ✓	✓ ✓		
Data Engine	✓	✓	✓	✓
File Store			✓	✓
Repository	✓	✓		

Topology 1 - Dedicated Data Engine Nodes

1. Add File Store to Node 1.

   ```
   tsm topology set-process -n node1 -pr filestore -c 1
   ```

2. Designate Node 3 and Node 4 to preferentially execute extract-query workloads

   ```
   tsm topology set-node-role -n node3, node4 -r extract-queries
   ```

3. Designate Node 1 to preferentially execute extract-refresh workloads.

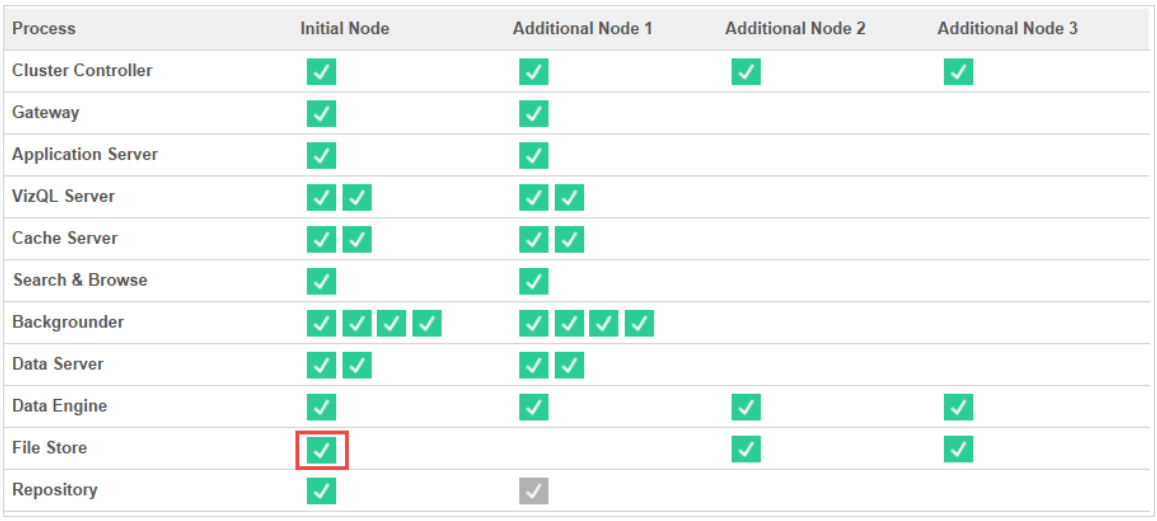
   ```
   tsm topology set-node-role -n node1 -r extract-refreshes
   ```

4. Designate Node 2 to preferentially execute non-extract-refresh workloads.

   ```
   tsm topology set-node-role -n node2 -r no-extract-refreshes
   ```

5. Apply pending changes.

   ```
   tsm pending-changes apply
   ```



Process	Initial Node	Additional Node 1	Additional Node 2	Additional Node 3
Cluster Controller	✓	✓	✓	✓
Gateway	✓	✓		
Application Server	✓	✓		
VizQL Server	✓ ✓	✓ ✓		
Cache Server	✓ ✓	✓ ✓		
Search & Browse	✓	✓		
Backgrounder	✓ ✓ ✓ ✓	✓ ✓ ✓ ✓		
Data Server	✓ ✓	✓ ✓		
Data Engine	✓	✓	✓	✓
File Store	✓		✓	✓
Repository	✓	✓		

Topology 2 - Extra File Store Node

Note: In your Tableau Server deployment, adding File Store roles to existing nodes will temporarily increase network I/O between all File Store nodes while the new File Store is being synchronized. The duration of this operation is dependent on the volume of data on the File Store and the network bandwidth capacity. The status of synchronization can be monitored using the TSM Web GUI. If you are adding more than one File Store to your deployment, it is recommended to add them consecutively and wait for the initial synchronization to complete in between each File Store addition.

Fine tune extract query workload management

When extract queries for email subscriptions and metric alerts are running at the same time that users are interactively viewing extract-based visualizations, users may experience slower than normal viz load times. Use the following node roles to fine tune how these workloads are prioritized.

Node role to use	Type of extract query workload	Example
`extract-queries`	scheduled	email subscriptions and metric alerts

`extract-queries-interactive`	interactive	users viewing an extract-based visualization

If your server deployment is seeing growth in email subscriptions and metric alerts, you can add nodes and assign the `extract-queries` node role, which makes them more available to handle subscriptions and alerts.

If your server deployment is seeing growth in users viewing extract-based visualizations, you can add nodes and assign the `extract-queries-interactive` node role, which makes them prioritize interactive extract queries to reduce extract-based viz load times. The `extract-queries-interactive` node role is a preference and not strict isolation. This means that queries will be routed to nodes that have the `extract-queries-interactive` node role assigned. If you have multiple nodes with the `extract-queries-interactive` role, queries will be routed based on node health.

For example, add a node and designate it to preferentially execute `extract-queries-interactive` workloads.

- `tsm topology set-node-role -n node4 -r extract-queries-interactive`

Configuration options

Configuration	**Jobs**
all-jobs (default)	All Tableau Server jobs
extract-queries	Jobs that are created for extract queries. The nodes selected will run as all-jobs and will prioritize the processing of extract queries.
extract-queries-interactive	Jobs that are created for extract queries. The nodes selected will run as all-jobs and will prioritize the processing of interactive extract queries, such as those that run when a user is looking at their screen and waiting for an extract-based dashboard to load. This is an advanced setting and it should only be used if the cluster has a heavy subscription and alert job workload that causes users to experience

	degraded performance on viz load times that run around the same time as scheduled loads.

For more information on configuring node roles using TSM commands, see tsm topology set-node-role.

License requirements

To configure a node to run extract queries you must have a valid Advanced Management license activated on your Tableau Server.

How to see node roles

Use the following command to see what node roles are currently configured on Tableau Server:

```
tsm topology list-nodes -v
```

Who can do this

Tableau Server Administrators can configure node roles and activate any required product keys.

Tableau Server Independent Gateway

This topic provides an overview of Tableau Server Independent Gateway.

Independent Gateway is a reverse proxy server and load balancer based on Apache httpd. It uses the same Apache httpd version as the Tableau Server Gateway process internal to Tableau Server cluster, but it is suitable for deployment in a network DMZ. Because Independent Gateway is a part of Tableau Server, it is managed by TSM and does not need separate configuration.

Its configuration has full knowledge of the topology of externally accessible Tableau Server components, and is updated when the cluster topology changes. After a simple Independent

Gateway installation process, configuration choices are made centrally using Tableau Services Manager (TSM) configuration items.

With this feature, Tableau Server can now be configured in two ways:

- Install Tableau Server in a self-contained installation. Any reverse proxy needs to be installed and managed separately.
- Install Tableau Server and install Independent Gateway as a reverse proxy that is managed by Tableau Server (version 2022.1 and later).

Why use Independent Gateway?

Using Independent Gateway has the following benefits over installing a separate reverse proxy:

- **Fully supported:** The Independent Gateway is part of a Tableau Server installation, and is fully supported by Tableau.
- **Tableau Server aware:** When you use a separate reverse proxy, it needs to be updated when the topology of Tableau Server changes. Independent Gateway is fully aware of all externally-callable Tableau Server components, and is updated when these change.

Managing Independent Gateway

License Management

To use Independent Gateway you must have Advanced Management capabilities in Tableau Server. There is no licensing done on the Independent Gateway node. For more information, see About Tableau Advanced Management on Tableau Server. If you don’t have an Advanced Management key activated or if the license is expired, you will see the following behavior:

- If you try to configure Tableau Server with Independent Gateway it will fail.
- If you are already using Independent Gateway, and the Advanced Management license expires, the server will fail on restart.

Backup and Restore

There is no impact to backup or restore with Independent Gateway. A backup or restore of Tableau Server will not include any information or configuration for Independent Gateway. If you use the backup to create a new installation of Tableau, you will need to separately install, configure, and enable Independent Gateway for the new Tableau Server installation.

High Availability Considerations

You can install multiple instances of Independent Gateway to provide robust high availability in your reverse proxy. You might also want to increase the number of Independent Gateway nodes if you have large numbers of client sessions accessing Tableau.

Topology

When you configure Tableau Server with Independent Gateway, you no longer need to set up and configure a separate reverse proxy. Independent Gateway will appear on the TSM status page as an external service.

> **Note:** The status page and the status output on the CLI will only show a single instance of Independent Gateway, even if you have installed multiple Independent Gateway nodes.

Next

Install Tableau Server with Independent Gateway

Install Tableau Server with Independent Gateway

This topic walks you through the process of installing Tableau Server Independent Gateway.

Following this process will result in a Independent Gateway configuration with a direct connection to the backend Tableau Sever deployment. You can learn more about *direct* vs *relay* connection modes in the topic, Configure Tableau Server with Independent Gateway.

Prerequisites

- You must have a dedicated server with at least 2 cores (4 vCPUs), 8 GB of RAM, and 100 GB free disk space.
- You must use Tableau Server 2022.1 or later.
- You must use an installer for Tableau Server Independent Gateway with a major version (**2022.1** for example) that matches the version of Tableau Server. We recommend maintenance versions (2022.1**.1** or 2022.1**.5** for example) match as well, but this is not a requirement. If "static assets" change between versions and versions do not match, you may see some unexpected image impact. For example, maps may not be up-to-date if Independent Gateway is an earlier version than Tableau Server.
- You must have Advanced Management enabled in Tableau Server. To learn more about Advanced Management, see About Tableau Advanced Management on Tableau Server.
- By default, the Independent Gateway must be able to communicate with the backend Tableau Server deployment on ports 80 and 21319 during installation. You can change these default ports during initialization as described later in this topic.
- Verify that your Tableau Server deployment is complete and healthy before installing and configuring Independent Gateway.
- Verify that you are not running any other web-aware applications on the computer where you are installing Independent Gateway. For example if Apache httpd is installed on the computer, uninstall it, or configure `httpd` so that it is not actively listening on port 80.

Install Tableau Server and Independent Gateway

Installing Independent Gateway is done using a standalone installation package that includes "tsig" as part of the file name to distinguish it from the full Tableau Server installer. We strongly recommend you install Independent Gateway after installing Tableau Server and confirming that it is functioning as expected. To install you must be the root user (or be able to sudo to root). You can install one or more instances of Independent Gateway, but each instance of Independent Gateway must be installed separately. If you are installing multiple instances for high availability or to distribute a heavy client load, repeat the installation steps for each instance.

After installation, you are prompted to run a script called `initialize-tsig` to complete the installation. The script takes information your provide using parameters, and configures

Independent Gateway. Once Independent Gateway is fully installed, you need to run a TSM command on the initial Tableau Server node to configure the server with details about the instance of Independent Gateway.

The IG installation consists of these steps:

- Run the platform-specific installer.
- Run the post-install script.
- Enable the Independent Gateway instance using TSM.

Step 1: Download and install Tableau Server

1. Install and Initialize TSM: Follow the instructions provided in this topic and complete steps 1-5 which runs the setup program and installs TSM.
2. Activate and Register Tableau Server: Provide the Tableau Server Key and the Advanced Management key in the activate step. You will need to run the following command twice, first with the Tableau Server product key and then with the Advanced Management product key:

   ```
   tsm licenses activate -k <product key>
   ```

3. Configure Initial Node Settings: Follow the instructions provided in the topic to configure the initial node settings.

Step 2: Download and install Independent Gateway

Install Tableau Server Independent Gateway with your distribution's package manager, then run a script to initialize Independent Gateway. The script is included with the installed package.

Independent Gateway is installed in the `/opt` directory.

1. Log on as a user with `sudo` access to the computer where you want to install Independent Gateway.

> **Note:** To avoid possible complications, we recommend a user account that does not include any special characters (for example, non-ASCII, "+", "-"). These may cause problems, including a failure to fully install Independent Gateway, depending on how your environment is configured.

2. Download the `.rpm` or `.deb` installer package from the Tableau Server Downloads and Release Notes page.
3. Navigate to the directory where you copied the `.rpm` or `.deb` package.
4. Use the package manager to install the Independent Gateway package.

   Do not install to a location using a symbolic link or to a directory on a Network File System (NFS) volume. Run the following commands to install Independent Gateway, where `<version>` is formatted as major-minor-maintenance (example: `2022-1-0`).

   - On RHEL-like distributions, including CentOS:

     ```
     sudo yum update

     sudo yum install tableau-tsig-<version>.x86_64.rpm
     ```

   - On Ubuntu:

     ```
     sudo apt-get update
     sudo apt-get upgrade
     sudo apt-get -y install gdebi-core
     sudo gdebi -n tableau-tsig-<version>_amd64.deb
     ```

Initialize Tableau Server Independent Gateway

The next step is to run the `initialize-tsig` script.

1. Navigate to the `scripts` directory:

   ```
   cd /opt/tableau/tableau_tsig/packages/scripts.<version_code>/
   ```

2. Run the following script to initialize and start Independent Gateway:

   ```
   sudo ./initialize-tsig --accepteula -c <ts_cluster_location> --
   <optional_parameters>
   ```

   The only required parameters for the `initialize-tsig` script are `--accepteula` and `-c`.

   - **--accepteula** - You must include this parameter to accept the Tableau End User License Agreement (EULA). A link to the EULA is available in the following location:

     ```
     /opt/tableau/tableau_tsig/packages/docs.<version_code>/
     ```

   - **-c** - You must include this parameter to specify the network location of all the nodes in the Tableau Server cluster. These nodes may be sending "housekeeping" requests to the Independent Gateway. Wild cards and subnet masks can be used to specify multiple nodes. To specify multiple addresses, separate addresses by spaces and use quotes around the complete set. Values must be provided in one of the forms acceptable to the Apache `httpd mod_authz_host` "Require" directive. For more information, see https://httpd.apache.org/docs/2.4/mod/mod_authz_host.html.

   All other parameters, for example default ports used by HTTP and the housekeeping process, are optional and if left off will be given default values. Review the parameters and their default values before running the script: Help Output for initialize-tsig Script.

Step 3: Enable Independent Gateway in Tableau Server

The last step of installing and configuring Independent Gateway is to enable Independent Gateway in Tableau Server. To do this, use the TSM command `tsm topology external-services gateway enable -c <file>` with a JSON file that identifies the Independent Gateway instance or instances, and provides Tableau Server with the details necessary for communication between the server and the Independent Gateway.

Tableau Server must be in a stopped state to enable Independent Gateway.

The Independent Gateway JSON file contents

The JSON file that you use to enable the instances of Independent Gateway on Tableau Server needs to contain the following:

- **id**—The id value must match the `tsig_instance_id` for the particular instance. If you did not provide this, the default is the fully qualified domain name of the Independent Gateway computer, in lowercase. The value in the json file must match the output of the `hostname` command.
- **host**—The host value must be a DNS-resolvable name for the Independent Gateway computer that can be resolved by the Tableau Server nodes using DNS.
- **port**—The port must match the housekeeping port (`tsig_housekeeping_port`) specified on the Independent Gateway instance. If you did not provide this during initialization, the default is "21319".
- **protocol**—The protocol must be the same as the housekeeping protocol (`tsig_housekeeping_port_protocol`) specified on the Independent Gateway instance. If you did not provide this during initialization, the default is "http".
- **authsecret**—The authsecret must match the secret created by the initialization script on the Independent Gateway instance.

The Independent Gateway auth secret

The initialization script creates a unique, shared secret on each Independent Gateway computer. You need this secret to enable Independent Gateway in Tableau Server. Copy the secret and include it in your JSON file as the "authsecret".

The shared secret is located in the `tsighk-auth.conf` file here:

```
/var/opt/tableau/tableau_tsig/config/tsighk-auth.conf
```

Independent Gateway JSON file example

The JSON file should be in the format below. This example JSON file shows default values where there are defaults. Your file should have use the actual values that match your installation of Independent Gateway and your organization.

```
{
 "independentGateways": [
```

```
    {
      "id": "<mycomputer.example.com>",
      "host": "<DNS name of Independent Gateway computer>",
      "port": "21319",
      "protocol": "http",
      "authsecret": "<shared-secret01>"
    },
    {
      "id": "<mycomputer2.example.com>",
      "host": "<DNS name of second Independent Gateway computer>",
      "port": "21319",
      "protocol": "http",
      "authsecret": "<shared-secret02>"
    }
  ]
}
```

Enabling Independent Gateway in Tableau Server

To complete the installation of Independent Gateway , you need to enable it using TSM.

1. Copy the JSON configuration file to the initial node of Tableau Server.
2. On the initial node, open a command prompt using an account hat is a member of the `tsmadmin` group.
3. Run the following commands to stop Tableau Server, enable Independent Gateway using the json configuration file, and restart the server:

```
tsm stop
tsm topology external-services gateway enable -c tsig.json
tsm start
```

Step 4: Verify Independent Gateway in Tableau Server

You should be able to navigate to the Tableau Server sign-in page by entering the address of the Independent Gateway in a browser.

If there is a firewall between the Independent Gateway and the backend Tableau Server deployment, then you will need to open the ports for the Tableau Server processes for direct connection. See Direct connection for more information.

Alternatively, you may choose to minimize port requirements by configuring Independent Gateway for a relay connection. See Relay connection for more information.

Configure Tableau Server with Independent Gateway

This topic describes how to configure Tableau Server with Independent Gateway for different connection scenarios and for a custom authentication module.

For installation procedure, see Install Tableau Server with Independent Gateway.

For an end-to-end deployment example running on Tableau Server for Linux in AWS, see Configuring Web Tier in the Enterprise Deployment Guide.

Direct vs relay connection

The Independent Gateway can communicate directly with the back-end Tableau Server processes over multiple ports. We refer to this communication as *direct* connection.

Alternatively. you can configure Independent Gateway to relay client communication over a single port to the gateway process on Tableau Server. We refer to this as a *relay* connection.

The TSM configuration key that sets the connection type is `gateway.tsig.proxy_tls_optional`.

The following sections describe how these connections differ and how to set them.

Direct connection

In this configuration, the Independent Gateway communicates directly with the backend processes on Tableau Server over multiple ports. This requires that you open the ports between the firewall that separates Independent Gateway from the Tableau Server back end deployment.

The current implementation of Independent Gateway does not support TLS connections on these processes.

A direct connection allows Independent Gateway to communicate with the backend Tableau Server processes without proxying through the Gateway process. Direct connection provides better performance than the alternative relay connection.

Configuration

Direct connection is the default configuration. As such you do not need to run a command to set it. However, should you need to reset to the default direct connection, run the following commands:

```
tsm configuration set -k gateway.tsig.proxy_tls_optional -v all --
force-keys
tsm pending-changes apply
```

Manage port ingress

After installation, Independent Gateway must be able to communicate with Tableau Server over multiple ports. These ports are dynamically assigned during setup and are in the range, TCP 8000-9000. The specific ports and corresponding processes used to communicate to Tableau Server are written to a CSV file on the computer running Independent Gateway at `TSIG_DATA/config/httpd/proxy_targets.csv`.

In a default installation: `/var/opt/tableau/tableau_tsig/config/httpd/proxy_targets.csv`.

Use `proxy_targets.csv` to set or automate port ingress configuration through your network to Tableau Server. We recommend automating port ingress configuration since the ports may change if the topology Tableau Server deployment changes. Adding nodes or reconfiguring processes on the Tableau Server deployment will trigger changes to the port access required by Independent Gateway.

Relay connection

In a relay connection configuration, the Independent Gateway does not connect directly to the backend processes. Instead, the Independent Gateway relays communication to the Gateway process on the backend Tableau Server deployment over HTTP. This relay process results in an extra hop and therefore degrades performance as compared to the direct connection configuration.

One benefit of configuring Independent Gateway as a relay connection is to secure traffic with TLS. See Configure TLS on Independent Gateway.

Configuration

To configure Independent Gateway for relay connection to Tableau Server, run the following commands:

```
tsm configuration set -k gateway.tsig.proxy_tls_optional -v none --
force-keys
tsm pending-changes apply
```

Housekeeping protocol

Both direct and relay connections require communication with the Tableau Server housekeeping (HK) protocol. The HK process maintains configuration state between the backend Tableau Server deployment and the Independent Gateway. During installation the Tableau Server must be able to communicate with Independent Gateway over port 21319.

Housekeeping protocol communication details:

- The HK requests check Independent Gateway status and update configuration as needed. There is no customer data in these requests. The configurations do not include passwords or other secrets.
- The configuration files do contain details about the Tableau Server cluster topology so that Independent Gateway can perform reverse proxy functions. Cluster topology configuration can be considered sensitive because the configuration could provide targeting information to an attacker. Note that such configuration data would only be useful to attackers who could then access the Tableau Server cluster.

- The configuration update files include a check of the hashed contents. This provides an extra layer of security to validate the integrity of the configuration files that are used to update Independent Gateway.

By default, the HK process uses TCP 21319.

Beginning with Tableau Server 2022.1.2, TLS is supported on HK connection. See Configure TLS on Independent Gateway.

Change the HK port

You can change the port used by the HK protocol as part of the Independent Gateway initialization. See Help Output for initialize-tsig Script.

If you have already installed Independent Gateway, you can change the port by updating the `TSIG_HK_PORT` value in `environment.bash`.

By default, `environment.bash` is located at `/etc/opt/tableau/tableau_tsig`.

After you have updated the file you must restart tsig-httpd:

```
sudo su - tableau-tsig
systemctl --user restart tsig-httpd
exit
```

Log file locations

The most useful log entries on Tableau Server are in the `tabadminagent` log file directory. However, if you are running Tableau Server in a cluster, you must look on each instance to locate the latest tabadminagent logs.

On the Independent Gateway, the following log files are written to the `TSIG_DATA/logs/` directory.

By default, this is at the path `/var/opt/tableau/tableau_tsig/logs`:

- `access.log`: Independent Gateway will write to `access.log` for logging that is generated by the httpd.conf.stub configuration. Timestamped log files (e.g. `access_date.log`) are generated by httpd.conf configuration.

- `error.log`
- `startup.log`

These logs are also relayed verbatim to the Tableau Server deployment and stored in sub-directories of the Cluster Controller logs directory. As such, the Independent Gateway logs are included in the ziplog file generated by `tsm maintenance ziplogs` command.

Troubleshooting

For troubleshooting tips, see Troubleshooting Tableau Server Independent Gateway in the Enterprise Deployment Guide (EDG). The EDG provides example deployment of Tableau Server on Linux. The troubleshooting steps are useful for Windows or Linux versions of Tableau Server.

Configure Authentication Module with Independent Gateway

A common security practice is to allow only authenticated requests to pass through the inner firewall of the DMZ servers. The Independent Gateway supports the traditional Tableau Server authentication methods, but it also includes configuration properties that allow integrating an Apache httpd loadable module for custom authentication.

For example, by configuring SAML on Tableau Server and configuring a custom authentication module, you can require all users to authenticate with your IdP at the Independent Gateway. Only those users who are authenticated will then be able to access Tableau Server, which can then authenticate and authorize user access.

For a more detailed explanation of this authentication scheme, see Pre-authentication with an AuthN module in the Enterprise Deployment Guide.

To configure the authentication module, you must complete the following steps:

1. Generate authentication module configuration files. When setup is complete, each module and its configuration directives will be treated as Include options making the included files logically part of the overall httpd configuration.
2. Copy the configuration files to each computer running Independent Gateway. All files must be copied to the same locations on each Independent Gateway computer. Each file maps to a configuration property that is managed by Tableau Server.

3. Set the configuration properties with the `tsm configuration set` command on Tableau Server.

Do not edit the httpd configuration file (httpd.conf) on the Independent Gateway, since Independent Gateway includes logic to update httpd configuration based on changes made with TSM commands on Tableau Server.

Example authentication module configuration

For an end-to-end authentication module configuration example, see Example authentication configuration: SAML with external IdP in the Enterprise Deployment Guide. The example describes how to setup and configure SAML with Okta IdP and Mellon authentication module for a Tableau Server on Linux deployment running in AWS. Although the example describes the process for Linux, the configuration example is also useful for Tableau Server on Windows.

Configuration properties

The following table describes the various configuration files that you may reference. Each file maps to a configuration property that is set onTableau Server. You only need to define the properties necessary to formulate your custom authentication configuration. Skip any configuration properties that are not needed.

Configuration property	Description
gateway.tsig.authn_module_block	Appears at the end of the set of normally loaded Apache httpd modules. The intent is that the file includes one or more `LoadModule` directives. The modules themselves should be identified with full paths.
gateway.tsig.authn_global_block	Appears after all of the `LoadModule` references and before most other Apache httpd directives. The intent is to provide a place for any configuration directives needed by the custom module.
gateway.tsig.authn_globalbottom_block	Appears at the very bottom of the configuration file, again at the global level. The intent is to provide a

	place for any configuration directives needed by the custom module that must specifically come after various other directives. (You are unlikely to need this.)
gateway.tsig.authn_location_block	Appears inside a `<Location "/">` block, covering all URL paths.
gateway.tsig.authn_directory_block	This appears inside a `<Directory "/">` block, covering all paths to files served by the Independent Gateway. (You are unlikely to need this. Most files served by Independent Gateway are non-sensitive static assets, like images and JavaScript files.)
gateway.tsig.authn_virtualhost_block	Appears inside one or two `<VirtualHost>` blocks: one for TLS (if enabled) and one for non-TLS. If configured, the same file is included in both places. If you need to distinguish the two cases, you can use the standard Apache httpd HTTPS environment variable.

The `<Location "/tsighk">` block

In addition to the expected `<Location "/">` block for normal request traffic, there is also a `<Location "/tsighk">` block used to service internal Independent Gateway housekeeping (HK) requests. These HK requests have their own authentication guards and will not work with typical custom SSO solutions.

You may need to explicitly exclude your custom module from attempting authentication for the HK URL path.

To determine whether you need to exclude your module, first configure the module. Then look for HK requests in the Independent Gateway access log. You should see at least a status check once or twice a minute. If those requests are receiving a 200 response code, things are probably OK. On the other hand, if those requests receive a 3xx response code (redirecting to your custom authentication provider), then you need to do something about it.

Possible solutions include:

- The `<Location "/tsighk">` block contains the directive `AuthType None`, and that may be sufficient.
- The Independent Gateway httpd.conf has the standard Apache httpd directive `ProxyPreserveHost On`. If there is an unusual circumstance that requires it to be Off or some other value, that value can be set with the TSM configuration item `gateway.tsig.proxypreservehost`.
- You might need some module-specific directives to disable your authentication module for `<Location "/tsighk">`. You can't directly modify that block in the httpd.conf file. Instead, you can make another `<Location "/tsighk">` block in your `gateway.tsig.authn_global_block` file and let Apache httpd logically merge them. For example, some versions of mod_auth_mellon, a popular open source authentication module, requires `MellonEnable Off` for sections where it doesn't apply, even if `AuthType None` is set in those sections.
- When creating an additional `<Location "/tsighk">` section, as described in the previous bullet, you may find that the order of appearance of the various sections in the httpd.conf file makes a difference in how they affect each other. The standard `<Location "/tsighk">` section appears before the standard `<Location "/">` section. If your experimentation shows that some different order is needed, you might have to define another `<Location "/">` section in your `gateway.tsig.authn_global_block` block in addition to another `<Location "/tsighk">` section, in which case you might not need anything in a `gateway.tsig.authn_location_block` block.

Troubleshoot custom authentication module configuration

A handy way to understand how the Independent Gateway will compose the httpd.conf file is to set the TSM configuration items with values that point to empty files on your Independent Gateway computers. (The files must exist, but they can be empty.) You can then look at the Independent Gateway's httpd.conf file to get a concrete understanding of where the `Include` directives for the various configuration files will actually appear.

Configuration problems in the Independent Gateway httpd.conf can result in the `tsig-httpd` service being unable to start. Other configuration problems may interfere with receiving configuration updates from the Independent Gateway companion service on the Tableau Server cluster. One way to recover, after you've fixed whatever caused the problem, is to copy `TSIG_DATA/config/httpd.conf.stub` to `TSIG_DATA/config/httpd.conf`, and then restart the `tsig-httpd` service.

For more troubleshooting tips, see Troubleshooting Tableau Server Independent Gateway in the Enterprise Deployment Guide (EDG). The EDG provides example deployment of Tableau Server on Linux. The troubleshooting steps are useful for Windows or Linux versions of Tableau Server.

Configure TLS on Independent Gateway

TLS support for Independent Gateway is in Tableau Server 2022.1.2 and later.

Both Tableau Server and Tableau Server Independent Gateway use the SSL module (mod_ssl) built with OpenSSL to implement Transport Layer Security (TLS) features.

Because of its complexity and security-sensitive nature, we recommend that TLS configuration is planned and implemented by an IT professional who is familiar TLS on Apache httpd.

In many cases, we use "SSL" in the names of things for compatibility with existing TSM or Apache httpd configuration properties or concepts. "SSL" actually refers to protocol versions now considered insecure and obsolete. However, the legacy name persists and is often used interchangeably with TLS as convention. Tableau Server and Independent Gateway do not support SSL-era protocols.

TLS configuration example

For an end-to-end TLS configuration example, see Configure SSL/TLS from load balancer to Tableau Server in the Enterprise Deployment Guide. The topic shows a step-by-step example of configuring TLS on Tableau Server on Linux in an AWS deployment. Although the example describes the process for Linux, the configuration example is also useful for Tableau Server on Windows.

TLS configuration overview

You can configure TLS for HTTPS on any of the following sections of the internet-to-Tableau Server path:

- From the external network (internet or front-end load balancer) to Independent Gateway

- From Independent Gateway to Tableau Server
- For housekeeping (HK) process from Tableau Server to Independent Gateway

This topic provides procedures to configure each of these hops.

You will need to make configuration changes to Independent Gateway computers and to the Tableau Server cluster.

Certificate requirements and considerations

The certificate requirements for Independent Gateway are the same as those specified for Tableau Server "external SSL." See SSL certificate requirements.

Other considerations:

- To simplify certificate management and deployment, and as a security best practice, we recommend using certificates generated by a major trusted third party certificate authority (CA). Alternatively, you may generate self-signed certificates or use certificates from a PKI for TLS. In this case, pay attention to the configuration options for trusting CA certificates and validating certificates.
- If your implementation requires the use of a certificate chain file, see the Knowledge Base article, Configure TLS on Independent Gateway when using a certificate that has a certificate chain.
- If you are running multiple instances of Independent Gateway, then you must distribute certificates to each computer in the same location (file path).
- If you are running a Tableau Server deployment with more than one node, certificates that you upload with TSM commands are automatically distributed across the nodes. Run all TSM commands on the initial node.

Global TLS configurations

The following configurations are global. The configuration options below refer to configuration keys that must set with the `tsm configuration set` command. The commands must include the `--force-keys` option.

It's unlikely that you will need to change these values.

Note that each pair of keys shares the same naming format, where the string, `tsig`, sets the value for the Independent Gateway. The key that does *not* include the string, `tsig`, sets the value for the gateway process on the Tableau Server cluster.

If you do not set a value for the `tsig` key, then the default Tableau Server gateway value will be used.

`gateway.tsig.httpd.socache or gateway.httpd.socache`

Default: `shmcb`

Alternate value: `dbm`

The storage type of the inter-process SSL Session Cache. For more information about the shmcb and dbm storage types, see SSLSessionCache Directive on Apache website.

`gateway.tsig.httpd.shmcb.size or gateway.httpd.shmcb.size`

Default: `2048000`

Amount of memory, in bytes, to use for the circular buffer when using `shmcb` storage type.

Note: Another global key is `gateway.tsig.ssl.key.passphrase.dialog`. If applicable, there is only a single configuration for `gateway.tsig.ssl.key.passphrase.dialog`. By design, it collects passphrases for all encrypted private key files in the configuration. The applicable sections later in this topic describe the use of this key.

External TLS to Independent Gateway

The process of configuring external connections to terminate TLS on the Independent Gateway servers is conceptually similar to how "external SSL" is configured for a Tableau Server cluster. The mechanics are different. TSM does not automatically distribute certificate and

key material to Independent Gateway nodes. Additionally, Independent Gateway does not automatically provide a way to supply the optional TLS key passphrase on startup.

The following steps describe how to configure TLS from external source to Independent Gateway computers.

Step 1: Distribute files to Independent Gateway computers

1. Place certificates and related files in a location and with permissions that allow the Independent Gateway service (tsig-httpd) to read them. We recommend restricting access to the key files such that only the Independent Gateway service can read them.
2. Place all files, certificates, and keys in exactly the same locations on all Independent Gateway computers. Place the files outside of the TSIG_INSTALL and TSIG_DATA paths so that they do not get removed if you reinstall or upgrade Independent Gateway.

Step 2: Update environment variables on Independent Gateway computers

On each Independent Gateway computer, set the `TSIG_PORT` and `TSIG_PROTOCOL` environment variables to `443` (by convention, but any unused TCP port number is supported) and `https` respectively.

Change these values by updating the `TSIG_PORT` and `TSIG_PROTOCOL` environment variables in `environment.bash`.

By default, `environment.bash` is located at `/etc/opt/tableau/tableau_tsig`.

After you have updated the file you must restart tsig-httpd:

```
sudo su - tableau-tsig
systemctl --user restart tsig-httpd
exit
```

Step 3: Set TLS configuration properties on Tableau Server

Most of the TSM configuration keys in the following table are derived from Apache httpd directives. As such, the configuration values for these TSM configuration keys map directly to the valid values for the corresponding Apache directive. Links to corresponding directives are included in the following table.

In some cases. the configuration will use fallback configurations if a particular key is not set. These are called out in the table below.

The configuration options in the following table refer to configuration keys that you must set with the `tsm configuration set` command. All commands must include the `--force-keys` option. For example:

```
tsm configuration set -k gateway.tsig.ssl.enabled -v true --force-keys
```

After you have set the configuration keys, you must run `tsm pending-changes apply`.

Configuration property	Description	Corresponding Apache directive
gateway.tsig.ssl.enabled	Required. Enables TLS. Must be set to `true`.	N/A
gateway.tsig.ssl.cert.file_name	Required. Path + file name of the certificate file for Independent Gateway. For example, `/etc/ssl/certs/tsig-ssl.crt`.	SSLCertificateFile
gateway.tsig.ssl.key.file_name	Required.	SSLCertificateKeyFile

	Path + file name of the certificate key file for Independent Gateway. For example, `/etc/ssl/keys/tsig-ssl.key`.	
gateway.tsig.ssl.key.passphrase.dialog	If your key requires a passphrase, then you must configure this key with the correct string expected by the Apache httpd SSLPassPhraseDialog directive. Do not enter the literal passphrase for this key. Refer to the Apache documentation for information about how to configure this key. This configuration is global for Independent Gateway.	SSLPassPhraseDialog
gateway.tsig.ssl.protocols Fallback: ssl.protocols	Specify supported versions of SSL/TLS. See Security Hardening Checklist for more information about default configuration.	SSLProtocols
gateway.tsig.ssl.ciphersuite Fallback: ssl.ciphersuite	Specifies ciphers that the client is permitted to negotiate for SSL connection.	SSLCipherSuite
gateway.tsig.ssl.client_certificate_login.required	Set this value to `true` to enable mutual TLS on this connection. You must also set the `gateway.tsig.ssl.cacert.file`	N/A

	property as specified below.	
gateway.tsig.ssl.cacert.file	Specifies the file containing the concatenated CA certificates for client authentication process.	SSLCACer-tificateFile
gateway.tsig.ssl.revocation.file	Specifies the file containing the concatenated CA revocation lists for clients that connect to Independent Gateway.	SSLCARe-vocationFile
gateway.tsig.ssl.redirect	When Independent Gateway has been configured for TLS, this option forces client requests from port 80 (default) to redirect to TLS. Default: `true`.	N/A
gateway.tsig.ssl.redirect_from_port	When `gate-way.tsig.ssl.redirect` is set to `true`, this option allows you to specify the port from which traffic is redirected. Default: `80`.	N/A

Independent Gateway to Tableau Server

This section describes how to encrypt the connection between the Independent Gateway and Tableau Server.

Step 1: Configure and enable TLS on Tableau Server

See Configure SSL for External HTTP Traffic to and from Tableau Server.

Note that "SSL" is actually a TLS implementation, and "external" refers to an external connection to Tableau Server. In this scenario, the Independent Gateway is the "external" connection.

We recommend enabling and verifying that clients can connect with TLS directly to Tableau Server before configuring Independent Gateway.

Step 2: Distribute certificate files on Independent Gateway computers

You will need to distribute certificate files on the Independent Gateway computers if either of the following are true:

- You are using self-signed or PKI certificates for the TLS certificates on the Tableau Server deployment.
- You are enabling mutual TLS on the connection from Independent Gateway to Tableau Server.

As with all TLS-related files on Independent Gateway computers, you must put the files in the same paths on each computer. All file names for TLS shared files must also be the same.

Step 3: Set TLS configuration properties on Tableau Server

Most of the TSM configuration keys in the following table are derived from Apache httpd directives. As such, the configuration values for these TSM configuration keys map directly to the valid values for the corresponding Apache directive. Links to corresponding directives are included in the following table.

In some cases. the configuration will use fallback configurations if a particular key is not set. These are called out in the table below.

The configuration options in the following table refer to configuration keys that you must set with the `tsm configuration set` command. All commands must include the `--force-keys` option. For example:

```
tsm configuration set -k gateway.tsig.ssl.enabled -v true --force-keys
```

After you have set the configuration keys, you must run `tsm pending-changes apply`.

Configuration property	Description	Corresponding Apache directive
gateway.tsig.ssl.proxy.cacertificatefile	If your organization uses a self-singed or PKI-generated TLS certificate for Tableau Server, then you must specify a path to the root CA certificate file. This root CA certificate file must be stored on the Independent Gateway computers.	SSLProxyCACertificateFile
gateway.tsig.ssl.proxy.protocols Fallback: ssl.protocols	Specify supported versions of SSL/TLS. See Security Hardening Checklist for more information about default configuraiton.	SSLProtocols

gateway.tsig.ssl.proxy.ciphersuite Fallback: ssl.ciphersuite	Specifies ciphers that the client is permitted to negotiate for SSL connection.	SSLCipherSuite
gateway.tsig.ssl.proxy.machinecertificatefile	For mutual TLS. Specifies the file containing concatenated certificate-key pairs for authentication of the Independent Gateway to Tableau Server.	SSLProxyMachineCertificateFile
gateway.tsig.ssl.proxy.verify	Specify whether Independent Gateway should verify the certificate presented by Tableau Server. Defaults to `require`.	SSLProxyVerify

gateway.tsig.ssl.proxy.checkpeername	Specify whether Independent Gateway inspects Tableau Server certificate to verify that subject name matches server name. Defaults to `off`.	SSLProxyCheckPeerName
gateway.tsig.ssl.proxy.checkpeerexpire	Specify whether Independent Gateway inspects Tableau Server certificate to verify expiry: Defaults to `off`.	SSLProxyCheckPeerExpire

Step 4: Upload root CA certificate to Tableau Server

If the TLS certificate that you are using on the Independent Gateway computers is a self-signed or PKI-generated certificate then you must perform this additional step. If the TLS certificate that you are using on the Independent Gateway computer is a certificate from a trusted third-party certificate authority, then you can skip this step.

Copy the root CA certificate used for the Independent Gateway computers to the initial node of Tableau Sever, and then run the following commands:

```
tsm security custom-cert add -c <root-certificate-file-name>.pem
tsm pending-changes apply
```

Housekeeping connection between Tableau Server and Independent Gateway

The housekeeping (HK) process maintains configuration state between the backend Tableau Server deployment and the Independent Gateway.

When Independent Gateway is installed, the default configuration provides an unencrypted HTTP connection. Independent Gateway listens for housekeeping requests originating in the Tableau Server cluster (as you defined it during installation).

If you are running multiple instances of Independent Gateway, then all servers must accept housekeeping requests with TLS or all without TLS. This section describes how to configure HK connection for TLS. This process requires restarting Tableau Server and will result in down-time.

As with the previous TLS scenarios described above, many of the configuration changes for HK connection are set on configuration properties managed by the Tableau Server cluster. However, HK TLS configuration requires additional steps on Independent Gateway.

Step 1: Distribute files to Independent Gateway computers

If you have enabled TLS with external network and Independent Gateway, you may use the same certificate and key files for the HK connection.

If you are using the same assets, then the only other certificate file that you need to distribute is the root CA certificate for the certificate used by Tableau Server. If the TLS certificate presented by Tableau Server is generated by a trusted third-party CA, then you do not need to copy a root CA certificate to the Independent Gateway computers.

1. Place certificates and related files in a location and with permissions that allow the Independent Gateway service (tsig-httpd) to read them. We recommend restricting access to the key files such that only the Independent Gateway service can read them.

2. Place all files, certificates, and keys in exactly the same locations on all Independent Gateway computers.

Step 2: Import Independent Gateway root CA certificate into Tableau Server trust store

If the TLS certificate that you are using on the Independent Gateway computers is a self-signed or PKI-generated certificate then you must perform this additional step. If the TLS certificate that you are using on the Independent Gateway computer is a certificate from a trusted third-party certificate authority, then you can skip this step.

You may only upload one root CA certificate to Tableau Server. Therefore, if you have already uploaded a root CA certificate, then the same root CA certificate must sign the certificate that you will be using for HK connection.

Copy the root CA certificate used for the Independent Gateway computers to the initial node of Tableau Sever, and then run the following commands:

```
tsm security custom-cert add -c <root-certificate-file-name>.pem
tsm pending-changes apply
```

Step 3: Update environment variables on Independent Gateway computers

On each Independent Gateway computer, set the `TSIG_HK_PROTOCOL` environmental variable to `https`. You may specify an alternative port for HK (default is 21319) by setting the `TSIG_HK_PORT` environment variable as well.

Change these values by updating the `TSIG_HK_PORT` and `TSIG_HK_PROTOCOL` environment variables in `environment.bash`.

By default, `environment.bash` is located at `/etc/opt/tableau/tableau_tsig`.

After you have updated the file you must restart tsig-httpd:

```
sudo su - tableau-tsig
systemctl --user restart tsig-httpd
exit
```

Step 4: Update httpd.conf.stub on Independent Gateway

You must update the `httpd.conf.stub` file on each Independent Gateway server. The `httpd.conf.stub` file is used to seed the global httpd configuration.

The file is located at `TSIG_DATA/config/httpd.conf.stub`.

In a default installation: `/var/opt/tableau/tableau_tsig/-config/httpd.conf.stub`.

1. Open the `httpd.conf.stub` file in a text editor. You must update the `<VirtualHost *:${TSIG_HK_PORT}>` block with HK configuration details. The following example shows the required changes:

   ```
   <VirtualHost *:${TSIG_HK_PORT}>
    SSLEngine on
    #TLS# SSLHonorCipherOrder on
    #TLS# SSLCompression off
    SSLCertificateFile /etc/ssl/certs/tsig-ssl.crt
    SSLCertificateKeyFile /etc/ssl/private/tsig-ssl.key
    SSLCACertificateFile /etc/ssl/certs/rootTS-CACert.pem
   #TLS# SSLCARevocationFile /path/to/file
   </VirtualHost>
   ```

   Notes:

   - By default, each line in the `<VirtualHost *:${TSIG_HK_PORT}>` block is commented out by the string, `#TLS#`. To "enable" a line in the block, delete the `#TLS#` string at the beginning of the line.
   - As with all httpd configurations, each referenced file requires an absolute path to the file.
   - `SSLCACertificateFile` specifies the root CA certificate for the CA that generates the certificate presented by Tableau Server. You only need to set this if the TLS certificate used by Tableau Server is self-signed or generated by a PKI.

2. Stop the tsig-httpd service.

   ```
   sudo su - tableau-tsig

   systemctl --user stop tsig-httpd

   exit
   ```

   You will start receiving failed status checks at this point, indicating in TSM that your Independent Gateway component is degraded.

3. Copy `httpd.conf.stub` to `httpd.conf`.

   The `httpd.conf` file is in the same directory. Overwrite the `httpd.conf` file with the `httpd.conf.stub` file.

   ```
   cp httpd.conf.stub httpd.conf
   ```

4. Start the tsig-httpd service.

   ```
   sudo su - tableau-tsig
   systemctl --user start tsig-httpd
   exit
   ```

   You will continue receiving failed status checks at this point, indicating in TSM that your Independent Gateway component is degraded. These status checks will fail until you have completed the configuration as described in the following steps.

Step 5: Set TLS configuration properties on Tableau Server

Applying the configuration changes requires a restart of the server. To avoid long timout times, we recommend stopping the server before applying changes that you set here. In Step 6, you will run an update command and then restart TSM. Stopping TSM at this phase of the configuration results in a shorter downtime.

1. Stop TSM. Run the following command:

   ```
   tsm stop
   ```

2. Most of the TSM configuration keys in the following table are derived from Apache httpd directives. As such, the configuration values for these TSM configuration keys map directly to the valid values for the corresponding Apache directive. Links to corresponding directives are included in the following table.

   There are TSM configuration property names that include the `hk` node in the prefix: `gateway.tsig.hk.xyz.abc`. If set, these values are used for the HK TLS configuration. If not set, many configuration properties will use the fallback to `gateway.tsig.xyz.abc`, which themselves may or may not fall back to `gateway.xyz.abc`. The fallback configuration property is listed when relevant.

   The configuration options in the following table refer to configuration keys that you must set with the `tsm configuration set` command. All commands must include the `--force-keys` option. For example:

   ```
   tsm configuration set -k gateway.tsig.hk.ssl.enabled -v true --force-keys
   ```

Configuration property	Description	Corresponding Apache directive
gateway.tsig.hk.ssl.enabled (No fallback)	Required. Enables TLS. Must be set to `true`.	N/A
gateway.tsig.hk.ssl.cert.file_name Fallback: gateway.tsig.ssl.cert.file_name	Path + file name of the certificate file for Independent Gateway. For example, `/etc/ssl/certs/tsig-ssl.crt`.	SSLCertificateFile
gateway.tsig.hk.ssl.key.file_name	Path + file name of the certificate key file for Independent	SSLCertificateKeyFile

Fallback: gateway.tsig.ssl.key.file_name	Gateway. For example, `/etc/ssl/keys/tsig-ssl.key`.	
gateway.tsig.ssl.key.passphrase.dialog (Global property)	If your key requires a passphrase, then you must configure this key with the correct string expected by the Apache httpd SSLPassPhraseDialog directive. This configuration is global for Independent Gateway.	SSLPassPhraseDialog
gateway.tsig.hk.ssl.protocols Fallbacks: gateway.tsig.ssl.protocols ssl.protocols	Specify supported versions of SSL/TLS. See Security Hardening Checklist for more information about default configuration.	SSLProtocols
gateway.tsig.hk.ssl.ciphersuite Fallbacks: gateway.tsig.ssl.ciphersuite ssl.ciphersuite	Specifies ciphers that the client is permitted to negotiate for SSL connection.	SSLCipherSuite
gateway.tsig.hk.ssl.client_certificate_login.required (No fallback)	Set this value to `true` to enable mutual TLS on this connection. You must also set the `gateway.tsig.hk.ssl.cacert.file`	N/A

	property as specified below.	
gateway.tsig.hk.ssl.cacert.file Fallback: gateway.tsig.ssl.cacert.file	Specifies the file containing the concatenated CA certificates for client authentication process.	SSLCACertificateFile
gateway.tsig.hk.ssl.revocation.file Fallback: gateway.tsig.hk.ssl.revocation.file	Specifies the file containing the concatenated CA revocation lists for clients that connect to Independent Gateway.	SSLCARevocationFile

3. Apply changes. Run the following command:

```
tsm pending-changes apply.
```

Step 6 Update Independent Gateway JSON configuration file

The final step is to update the Independent Gateway configuration with a JSON file reflecting the switch to `https` and, if applicable, other port numbers.

Refer to the installation topic for more information about editing this file. See Step 3: Enable Independent Gateway in Tableau Server.

After you have updated the JSON file, run the following commands:

```
tsm topology external-services gateway update -c tsig.json
tsm start
```

Troubleshooting

For troubleshooting tips, see Troubleshooting Tableau Server Independent Gateway in the Enterprise Deployment Guide (EDG). The EDG provides example deployment of Tableau

Server on Linux. The troubleshooting steps are useful for Windows or Linux versions of Tableau Server.

Upgrade Tableau Server Independent Gateway

This topic walks you through the process of upgrading Tableau Server Independent Gateway. The process described in this topic is the same for all version upgrades. This means you can use this process for major version (for example, 2022.1 to 2023.1) and maintenance version (2022.1.1 to 2023.1.1) upgrades.

You must use an installer for Tableau Server Independent Gateway with a major version that matches the version of Tableau Server. We recommend maintenance versions (2022.1.1 or 2022.1.2 for example) match as well, but this is not a requirement. If "static assets" change between versions and versions do not match, you may see some unexpected image impact. For example, maps may not be up-to-date if Independent Gateway is an earlier version than Tableau Server.-

Overview

The upgrade process for Independent Gateway is actually a process of uninstalling then reinstalling the software. However, you can minimize configuration and overall downtime by following the process as described in this topic.

The process described in this topic assumes that you have a functional deployment of Tableau Server and Tableau Server Independent Gateway running. If you are upgrading the backend Tableau Server deployment as part of your upgrade, we recommend upgrading the Independent Gateway server(s) first, validating basic connectivity with a Tableau client, then proceeding with Tableau Server upgrade.

The steps detailed in this topic are summarized here:

1. Copy some existing files for use during upgrade.
2. Run the obliterate script to remove Independent Gateway.
3. Install new version of Independent Gateway.
4. Optional: Overwrite tsighk-auth file with original copy.
5. Update TLS settings.

6. Optional: Update back end Tableau Server
7. Restart Independent Gateway service.

Step 1: Copy files for reference

You may need to reference the configuration settings stored in the following files. Copy these files to a secure and accessible location (file paths are default location):

- /var/opt/tableau/tableau_tsig/config/tsighk-auth.conf
- /var/opt/tableau/tableau_tsig/config/httpd.conf.stub
- /etc/opt/tableau/tableau_tsig/environment.bash

Step 2: Obliterate Independent Gateway

To remove Independent Gateway from the server(s), run the `tableau-tsig-obliterate` script:

1. On the initial node, open a terminal session.
2. Run the `tableau-tsig-obliterate` script:

   ```
   sudo /opt/tableau/tableau_tsig/packages/scripts.<version_
   code>/tableau-tsig-obliterate -y -y -y
   ```

3. Restart the computer.

Step 3: Install Independent Gateway

Download the version of Independent Gateway that you want to upgrade to and then follow the same installation procedure as described in the topic, Install Tableau Server with Independent Gateway.

As with the original installation, you will need to specify initialization settings as part of set up. To use the same values, reference the `enviornment.bash` file that you copied in step 1. Do not overwrite the new bash file with the original.

Step 4: (Optional) Overwrite tsighk-auth file with original copy

The `tsighk-auth` file is a configuration file that stores a unique string called the authsecret. The authsecret is used to verify that the back end Tableau Server is communicating with a trusted instance of Independent Gateway. When you set up the original instance of Independent Gateway, you had to update a configuration file on the back end Tableau Server with the authsecret.

If your security policy allows, you can continue to use the original authsecret on Independent Gateway. Doing so avoids the process of updating and restarting the back end Tableau Server with a new authsecret.

To maintain the original authsecret, overwrite the `tsighk-auth.conf` file (located at `/var/opt/tableau/tableau_tsig/config/tsighk-auth.conf`) with the copy that you saved in step 1.

If your security policy requires that you refresh the authsecret, then make note of the new authsecret that was generated by the installation process. You will update the back end Tableau Server with the new authsecret later in the process.

Step 5: Update housekeeping TLS settings

If you did not configure TLS for housekeeping (HK) communication between Independent Gateway instance(s) and the back end Tableau Server deployment, then you can skip this step.

If you configured HK TLS, then you must manually copy the configuration from the original `httpd.conf.stub` file into the file located at `/var/opt/tableau/tableau_tsig/-config/httpd.conf.stub`) . Do not overwrite the new file with original file, as the newer file may include other settings.

1. Update the relevant TLS configuration settings in the following block:

```
<VirtualHost *:${TSIG_HK_PORT}>
SSLEngine on
#TLS# SSLHonorCipherOrder on
```

```
#TLS# SSLCompression off
SSLCertificateFile /etc/ssl/certs/tsig-ssl.crt
SSLCertificateKeyFile /etc/ssl/private/tsig-ssl.key
SSLCACertificateFile /etc/ssl/certs/rootTS-CACert.pem
#TLS# SSLCARevocationFile /path/to/file
</VirtualHost>
```

For more information about these settings, see Configure TLS on Independent Gateway.

2. When you have finished updating `httpd.conf.stub`, save it.

3. Copy `httpd.conf.stub` and then save to overwrite `httpd.conf` in the same directory.

Step 6: (Optional) Update back-end Tableau Server deployment

If you copied the original authsecret file (`tsighk-auth.conf`) to the new instance of Independent Gateway as described in step 4, then you may skip this step.

If you are refreshing the authsecret in the back end Tableau Server deployment, then you must update the `tsig.json` file on the initial node with the new authsecret. See Install Tableau Server with Independent Gateway. When you are finished, run the following TSM commands:

```
tsm stop

tsm topology external-services gateway update -c tsig.json

tsm start
```

Step 7: Restart the tsig-httpd service

When you have finished with the configuration. restart the `tsig-httpd` service.

```
sudo su - tableau-tsig

systemctl --user restart tsig-httpd

exit
```

Uninstall Tableau Server Independent Gateway

This topic walks you through the process of uninstalling Tableau Server Independent Gateway.

Uninstalling Independent Gateway

Uninstalling Independent Gateway is a two step process:

1. Disable the Independent Gateway instance in Tableau Server using TSM.

   To disable Independent Gateway in Tableau Server, run this command at the command prompt of the initial Tableau Server node:

   ```
   tsm topology external-services gateway disable
   ```

2. Uninstall the Independent Gateway instance you are upgrading from the computer it is running on.

   To uninstall Independent Gateway from a Linux computer, run the `tableau-tsig-obliterate` script in the `/opt/tableau/tableau_tsig/packaged/scripts.<version_code>` directory. This will entirely remove Independent Gateway from the computer.

   ```
   sudo /opt/tableau/tableau_tsig/packages/scripts.<version_
   code>/tableau-tsig-obliterate -y -y -y
   ```

To uninstall an instance of Independent Gateway, remove it from the server cluster using TSM, and then, after Tableau Server is fully reconfigured, uninstall Independent Gateway from the computer where it was installed. When you use TSM to remove the instance from Tableau Server, TSM no longer sends any communication to the Independent Gateway instance, so Independent Gateway is unaware of any configuration changes. but the Independent Gateway continues to respond based on the last known configuration and Tableau Server continues to accept those responses. You also need to remove the Independent Gateway instance from the computer it is running on.

Help Output for initialize-tsig Script

The following help content is the output when you run the following command:

```
sudo ./initialize-tsig -h
```

The initialize-tsig script is installed to `/opt/tableau/tableau_tsig/packages/scripts.<version_code>/`.

Output

```
REQUIRED
--accepteula            Indicate that you have accepted the End User
License Agreement (EULA).
                        You can find a link to the EULA in /opt/tableau/tableau_tsig
ages/docs.<version_code>

-c <ts_cluster_location>
                        The network location of all nodes in the Tableau
Server cluster. These may send
                        "housekeeping" requests to the TSIG node. The
locations must be one of the forms
                        that are acceptable to Apache httpd mod_
authz_host "Require" directive as described
                        at https://httpd.apache.org/docs/2.4/mod/mod_
authz_host.html. Use quotes if there are
                        embedded spaces.

OPTIONAL

-i <tsig_instance_id>   A unique identifier for the TSIG instance.
                        Default: The computer name.

-p <tsig_external_port>
                        Port listening for external requests.
                        Default: 80.
```

```
-t <tsig_external_port_protocol>
                        Protocol used for external requests. Options are "http" or
"https".
                        Default: "http".

-k <tsig_housekeeping_port>
                        Port listening for housekeeping requests from Tableau Server
                        Default: 21319.

-s <tsig_housekeeping_port_protocol>
                        Protocol used for housekeeping requests. Options are "http"
"https".
                        Default: "http".

-d data-dir             Set a custom location for the data directory
                        if it's not already set. If not set, the default is
                        "/var/opt/tableau/tableau_tsig".

-f                      Bypass warning messages and distribution version checks.

-g                      Do not add the current user to the "tableau-tsig" group.
                        Use this for easier access to log files and runtime files.

-a <username>           Name of the user to be added to the appropriate
groups instead of
                        the current user running the script. You cannot use both -g
-a.

-q                      Quiet, suppress output except for errors and warn-
ings.

--unprivileged-user=<name>
                        Name of the unprivileged account to run Tableau Server Indep
ent Gateway.
                        Default: "tableau-tsig".
```

```
--disable-account-creation
                        Do not create groups or user accounts for Tableau Server Ind
ent Gateway.
                        However, the values in: unprivileged-user will still be used
TSIG configuration.
```

Related topics

- Install Tableau Server with Independent Gateway

Tableau Server Backgrounder Resource Limits

Tableau Server Backgrounder resource limits feature was introduced in Tableau Server 2022.1.

Overview and Concepts

What it is

The Backgrounder resource limits feature gives you the ability to manage Backgrounder resources and control how they are used. Starting in 2022.1, you can set limits on the number of background jobs that can run at the same time on a site. You can also specify the number of Backgrounder hours that a site can use to run jobs per day.

Default limits can be applied to all sites, but you can also set custom limits per site. This gives you the ability to manage the Backgrounder resources based on the specific requirements for a site.

When to use it

Use this feature when you want to make sure the resources are used where it is most needed. More importantly, you can prevent a single site from consuming a lot of backgrounder resources thus impacting job completion or job queue time on other sites.

If you are currently experiencing unbalanced Backgrounder resource usage or delays in background job completions, use this feature to optimize resource usage that is right for your organizational needs and content priorities.

Requirements and recommendations

1. This feature requires that Tableau Server be enabled with a Advanced Management license.
2. A new Tableau Server process called **Resource Limits Manager** is required to enforce the set resource limits. When you install or upgrade to Tableau Server 2022.1 or later, this process is automatically configured on the initial node.
    - We recommend having at least 5 Backgrounder processes in total running on Tableau Server for Backgrounder resource limits feature to run optimally. The default configuration and topology recommendations are described in detail in the Tableau Server Resource Limits Manager.
    - We do not recommend adding more Resource Limits Manager processes to your Tableau Server. The automatically installed process on the initial node is sufficient.

Terminology and concepts

- **Site job limits:** The background jobs concurrency and run time limits for a site.
- **Default site limits:** The default jobs concurrency and run time limits for a site.
- **Custom site limits:** Site limits specific to that site.
- **Job type:** Same as the task type. Includes extract refreshes, subscriptions, and flows.
- **Concurrent jobs limits:** The maximum number of jobs of a specific type that can run at the same time. The maximum number is equal to the total number of Backgrounder processes deployed on the Server.
- **Daily limit:** The daily limit includes run time limits and the reset time.
- **Runtime limit:** This is the total number of Backgrounder job hours allotted for a site per day. The maximum number is equal to the total number of Backgrounder processes deployed on the Server multiplied by 24, 24 being the number of hours in a full day.
- **Reset time:** The time at which the daily limits are reset. This is automatically set to midnight UTC and cannot be changed.
- **No limit:** This is the same as if there are no resource limits configured. In this case, the jobs on a site that can run in parallel could use all the available backgrounder processes on Tableau Server.

What you can do

As a Tableau Server administrator,

1. You can specify concurrency jobs limit for each task type - extract refreshes, subscriptions, and flows. This limit controls the number of background jobs on a site that can be run at the same time, for a given task type. The limits are unique to each task type, meaning you can set different limits for extract refreshes, subscriptions, and flows.
2. You can specify the number of Backgrounder hours a site can use to run jobs in a 24 hour period - the daily job run time limit. Unlike the concurrency jobs limit, the daily job run time limit is cumulative across all task types.
3. You can find out how many Backgrounder hours have been used by a site in the last 24 hour time period.
4. You can see when a job is delayed due to set concurrency limits.
5. You can identify when a job is canceled due to lack of available Backgrounder resources.

You can see the job status and details on the Managing Background Jobs in Tableau Server page.

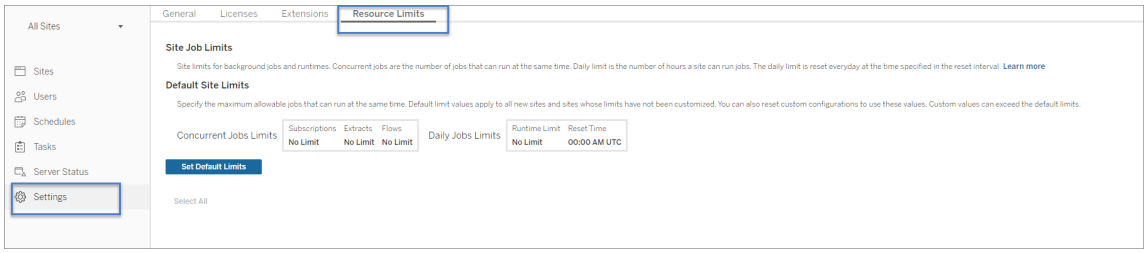
As a task owner, you will receive an e-mail when your background job is canceled because the site reached its daily run time limit.

How to set Backgrounder resource limits

Tableau Server does not automatically set any resource limits. Until you configure resource limits, there is no resource limitation on backgrounder resource usage.

To set resource limits for the first time or make changes thereafter, navigate to the **Settings** page, and select the **Resource Limits** tab.

You can only configure custom limits once you have configured the default limits.

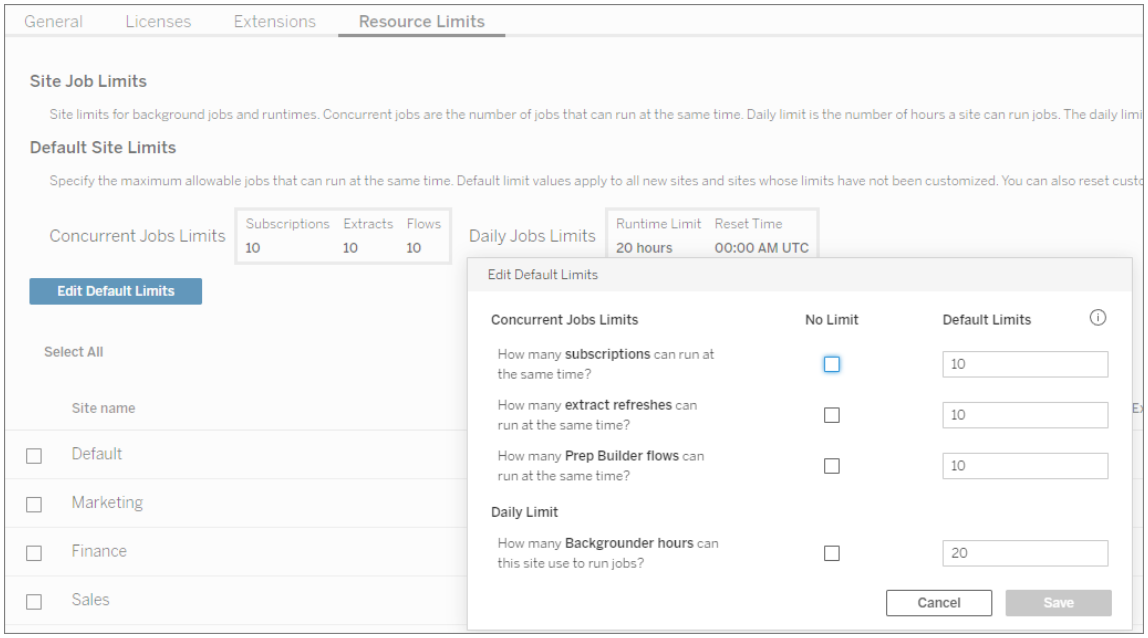


Default site limits

Default site limits are set at the server level so they can be applied to all sites. Changes to default site limits apply to all new sites and any existing sites that are set to use default limits. Sites that have custom limits will not be affected by this change.

- **Concurrent Jobs Limits** and the **Daily Runtime Limits** must be specified in whole numbers.
- Changes to default site limits are effective immediately and do not require a server restart.

To set default limits for the first time, on the **Resource Limits** tab, select **Set Default Limits**. If you are making updates to existing default limits, select **Edit Default Limits**.

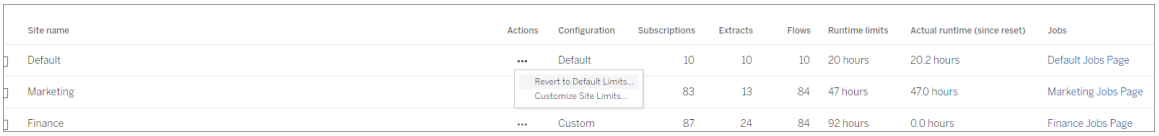


Custom site limits

For certain sites, you may need more or fewer resources than the default limits. This depends on the amount of content you have on the site and their importance to your business operations. When the default values are not the right capacity for a site, you can set custom resource limits for that site. Custom site limits can exceed default limits.

- **Concurrenct Jobs Limits** and the **Daily Runtime Limits** must be specified in whole numbers.
- Changes to custom site limits are effective immediately and do not require a server restart.

To configure custom limits for a site, on the **Resource limits tab**, under **Actions**, click the ellipses and select **Customize Site Limits**.



What happens after you configure resource limits

Once you set the resource limits for your sites, Tableau Server monitors and keeps track of the backgrounder resource usage, and makes sure the appropriate limits are enforced.

Jobs are placed in a queue when the site is at its maximum concurrency limit and will not run until a a job that is currently running on that site is complete, and Backgrounder capacity becomes available.

If after 12 hours there is still no concurrency resource available, then jobs that are still pending will be removed from the queue. At the end of the daily run time limit, **any pending jobs will be canceled for the day** and the task owner will receive a notification.

This feature requires that Tableau Server is activated with a valid Server Management license without which the limits are not enforced. If the license is invalid or deactivated for any reason, previously configured limits are saved and will be enforced once the license issue has been resolved.

When to make adjustments to the resource limits

Before making configuration changes, make sure that Tableau Server is running and is in a good state.

You can use the jobs page to identify jobs that are pending or canceled, and then determine which of those were due to resource limits.

Here are some patterns to monitor and make adjustments to the resource limits:

- If you are seeing that one or more jobs on a site are consistently getting canceled, you may need to increase the daily run time limits.
- If you are seeing that jobs on a site are consistently getting canceled early in the day, it probably means that the daily job run time is not enough for that site.
- If you are consistently seeing several jobs of a particular type stay pending over a period of time, you may want to consider increasing the concurrent jobs limits for that task type. Alternately consider scheduling the tasks over different time periods so everything is not running at the same time. Also, you may want to consider creating schedules that are spread over a longer period, so tasks are not all scheduled to run too closely to one another.
- If you don't want to change the resource limits in the situations described above, you can choose to adjust the frequency of when the jobs are scheduled to run. For example, if the job is scheduled to run every hour, adjust it to run on a less frequent schedule.

Who can do this

Tableau Server Administrators can configure resource limits.

A Tableau Server user who owns extract refresh, subscription, or flow run tasks will receive email notifications when their jobs are canceled.

Dynamic Scaling in a Container - Tableau Server Backgrounders

Introduction

The dynamic scaling of Backgrounder in a container enables various scaling strategies to be applied to backgrounder and scheduled jobs in Tableau Server. Autoscaling in this context means services can be independently scaled to handle variable task loads without requiring human intervention or impacting uptime of other server systems. The Tableau Server Containers which contain complete nodes of Tableau Server processes will continue to run as monolithic systems. Instead a smaller set of decoupled independent container services which comprise the "backgrounder" service role will be dynamicly scalable and handle computational load that would normally be handled by Tableau Server Containers. Backgrounder services are responsible for processing system tasks that include refreshing/creating extracts, sending subscriptions, checking data alerts, and many maintenance jobs. If there are times, for example, where it would be advantageous to refresh a large number of datasets or compute a swath of computationally expensive data alerts, you can now leverage Kubernetes to scale up compute power to complete those tasks efficiently. This guide is covers the configuration and deployment requirements for Autoscaling Backgrounders in Kubernetes. This document is a supplement to the Tableau Server in a Container documentation.

Prerequisites

Autoscaling Backgrounders is only available in Kubernetes and is based on Tableau Server in Containers. In order to use the autoscaling backgrounders feature, you must satisfy certain prerequisite:

- Tableau Server must have Advanced Management capabilities. For more information, see About Tableau Advanced Management on Tableau Server.
- You must be using Role-based licensing. Core licenses are not supported. For details on licensing, see Understanding License Models and Product Keys.
- The following Advanced Management features must be enabled:
    - Tableau Server External File Store
    - Tableau Server External Repository

- You must have a Kubernetes cluster version 1.20 or later and understand how to use and manage it.

Limitations

- This feature only works as a part of a Linux-based deployment.
- Flow jobs are not supported on the autoscaling backgrounders. Flow jobs will be handled by the backgrounder services that continue to run in the Tableau Server container.

Creating Tableau Server and Backgrounder Pod Images

The first step for using autoscaling backgrounders in containers is to create the service images that comprise the Tableau Server installation. These images will include the Tableau Server image, as well as images for the separate backgrounder and supporting services. You use the same build tool that is used to create the comprehensive all-in-one Tableau Server container image, but the tool must be version 2022.3.0 or later, you must have an Advanced Management license, and you need to use a special flag when building the images.

1. To create the service images, run this command:

   ```
   build-image --accepteula -i <installer> --backgrounder-images
   ```

   This creates the Tableau Server and four new images. These additional images contain individual services that comprise the new autoscalable backgrounder pod.

   The `docker images` command lists the images created in the local docker repository:

   ```
   hyper                          20214.21.1117.1006
   52fd9843c178   10 seconds ago
   gateway                        20214.21.1117.1006
   2308e602a2a3   11 seconds ago
   backgrounder                   20214.21.1117.1006
   4540e459cf23   12 seconds ago
   dataserver                     20214.21.1117.1006
   c5345ed47f51   12 seconds ago
   ```

```
tableau_server_image           20214.21.1117.1006
b27817047dd1   7 minutes ago
```

The hyper, gateway, backgrounder, and dataserver images comprise the new Backgrounder pod. Custom drivers, installation scripts, and properties will be shared across all five of these images. For more information, see Customizing the image.

2. Publish all of these images to your internal image repository for deployment.

Deployment Guide

The following information provides context for how to deploy Tableau Server in a Container and with Autoscaling Backgrounders. This information assumes you already understand and know how to deploy Tableau Server in a self-container container. For more information, see Tableau Server in a Container. The three Kubernetes configuration files in the Kubernetes Configuration section are templates that can be used to set up the deployment. The other sections in this guide cover the requirements and details of the deployment.

Deployment of Tableau Server and Autoscaling Backgrounders should be as simple as deploying the populated Kubernetes Configuration files at the bottom of the guide:

```
kubectl apply -f <tableau-kubeconfig-dir>
```

Backgrounder Jobs

Backgrounder pods assist the Tableau Server in a Container compute additional scheduled workloads in parallel. Backgrounder handles extract refresh, subscription, alert, flow, and system workloads. Distributing jobs among backgrounder pods means there will be more compute resources available for Tableau Server to handle other tasks, like interactive user activities like rendering workbooks and dashboards. Flow jobs are the only type of backgrounder job that does not run in the backgrounderFor details about Backgrounder jobs, see Managing Background Jobs in Tableau Server.

Backgrounder pods can handle any kind of load, except flow jobs, which must be run in the main Tableau Server containers, which continue to run the backgrounder service.

The Node Roles feature give users the flexibility to dedicate backgrounder pods for specific type of jobs. This feature is an extension of the Node Roles Feature on Tableau server. The detailed description about different node roles can be found here. Note, by default flow jobs are disabled on the backgrounder pods (i.e. the role is set to "no-flows") since the backgrounder pods are unable to run flow jobs.

To setup up node roles for backgrounder, you need to set the environment variable `NODE_ROLES` as part of kubeconfig for the container running backgrounder service. For example to set backgrounder to run only extract-refresh jobs, set up `NODE_ROLES` environment variable to extract-refreshes as shown below:

NODE_ROLE_CONFIG

```
containers:
      - name: backgrounder
        image: <backgrounder_image> # Backgrounder Image
        ports:
        - containerPort: 8600
        volumeMounts:
        - name: logmount
          mountPath: /var/log/tableau
        - name: clone-volume
          mountPath: /docker/clone
        - name: dataengine-volume
          mountPath: /docker/dataengine
        - name: temp
          mountPath: /var/opt/tableau/tableau_server-
/data/tabsvc/temp
        env:
          - name: ROOT_LOG_DIR
            value: /var/log/tableau
          - name: CLONE_ARTIFACT_DIR
            value: /docker/clone/clone-artifacts
          - name: FILESTORE_MOUNT_PATH
            value: /docker/dataengine
          - name: NODE_ROLES
            value: extract-refreshes
```

The Tableau Server pods will have at least one backgrounder configured in their topology, which is required to ensure there is always a place to run backgrounder jobs. By default, TSM will require that there must be a backgrounder able to handle every role of background job. In some scenarios, you may want to have backgrounder pods handle all jobs of a particular type. To do so, you must set the Server configuration key `topology.roles_handle_all_jobs_constraint_disabled` to `true`, which will disable the requirement that the TSM topology handle all job types. With this parameter set, the backgrounder role for the Tableau Server backgrounder instance could be set to `no-extract-refreshes` and the role for the backgrounder pods could be set to `extract-refreshes`, which would ensure that all extract refresh jobs run only on the backgrounder pods.

Note: Disabling this constraint allows you to configure roles such that some job types are never scheduled. The role configuration of TSM backgrounders and backgrounder jobs must be set carefully because TSM will no longer be verifying that all backgrounder job types can be scheduled.

Tableau Server in a Container Pods

The containers with Tableau Server as part of the autoscaling backgrounder pods deploy in almost the same way that our existing Tableau Server in a Container does. There are a few key requirement changes:

- A network file share is *required* to transfer configuration between the Tableau Server container and the backgrounder pods.
- You must enable and use the External Filestore feature. This also requires a dedicated network file share.

Backgrounder Pods

Backgrounder pods consist of four independent service containers working together: **gateway**, **hyper**, **dataserver**, and **backgrounder**. You may deploy these pods like typical independent Kubernetes container pods. The pods have the following requirements:

- Backgrounder pods must be able to reach the Tableau Server node using hostname DNS resolution.
- External Filestore and Clone network file shares must be provided.

Note: Backgrounder pods are configured with an init-container to wait until the Tableau Server container has successfully produced clone configuration output before proceeding to run.

Logs

Backgrounder pod services (like Tableau Server) still write logs predominately to disk. Because the backgrounder pods can be scaled in and out, they are ephemeral so it is essential to ensure that logs are stored off of the pods. Many customers with existing K8s environments will already be using a log aggregation service of some type to collect the logs from the pods they deploy. Examples of log aggregation services are Splunk, and fluentd. We strongly recommend that customers use some sort of log aggregation service to collect the logs from their backgrounder pods. To make log management easier, the kubeconfig we provide configures each service in the pod to write to a shared log volume. The path of the directory in each service container is specified by the `ROOT_LOG_DIR` environment variable.

If you need to open a support case and provide logs, you will provide two sets of logs: ziplogs gathered from the main Server containers, and logs from the backgrounder pods (either retrieved from your log aggregation service, or using the manual process below).

For customers that are unable to use a log aggregation service, logs can be retrieved manually from the pods.

Note: Logs from any pod that did not use a Persistent Volume Claim for the volume containing the logs will be lost when the pod is scaled down!

All the relevant logs are available at the `/var/log/tableau` directory (configurable via the `ROOT_LOG_DIR` environment variable) inside the backgrounder pod. We highly recommend you mount a PersistantVolumeClaim at this location so that logs are available when the pod dies.

Collecting logs when the backgrounder pod is running:

Create a tar file of the logs inside the container:

```
kubectl exec -it <backgrounder-pod-name> -- bash -c "tar -czf /dock-
er/user/backgrounder-pod-logs.tar.gz /var/log/tableau"
```

Copy the tarfile to outside the container:

```
kubectl cp <backgrounder-pod-name>:docker/user/backgrounder-pod-log-
s.tar.gz ./backgrounder-pod-logs.tar.gz
```

Collecting logs when the backgrounder pod has exited (or failed to start)

Attach any long-running pod with PersistantVolumeClaim mount which is used for backgrounder pod deployment logs. One example configuration:

```
apiVersion: v1
kind: Pod
metadata:
  name: <name>
  namespace: <namespace>
spec:
  containers:
  - name: get-logs-pod
    image: busybox:1.28
    securityContext:
      runAsUser: 0
      allowPrivilegeEscalation: true
    command: ['sh', '-c', "while :; do sleep 5; done"]
    volumeMounts:
    - name: logmount
      mountPath: /var/log/tableau
restartPolicy: Never
volumes:
  - name: logmount
    persistentVolumeClaim:
      claimName: logvolume
```

Create a tar file of the logs inside container:

```
kubectl exec -it <backgrounder-pod-name> -- bash -c "tar -czf /back-
grounder-pod-logs.tar.gz /var/log/tableau"
```

Copy the tarfile to outside the container:

```
kubectl cp <backgrounder-pod-name>:/backgrounder-pod-logs.tar.gz
./backgrounder-pod-logs.tar.gz
```

Live Configuration Changes

If you make configuration changes in Tableau Server in a Container (for example using the tsm command line) and want those configuration changes to be represented in Backgrounder pods, you need to run the `tsm settings clone` command to produce a new set of clone configuration files ("clone payload").

1. Use TSM to make configuration changes in the Tableau Server in a Container pod and apply the configuration changes to the server.
2. Run the following command in the Tableau Server in a Container pod:

   ```
   ## Run this command in the Tableau Server in a Container pod.
   tsm settings clone -d $CLONE_ARTIFACT_DIR
   ```

   This command creates a new set of configuration files and writes it to the Clone NFS drive location.
3. Redeploy your backgrounder pods. The pods should be configured to use the Clone NFS drive and will pick up the new configuration.

Scaling Strategies

Backgrounder pods can be scaled in Kubernetes using a variety of techniques and strategies. We provide an example scaling strategy that changes Backgrounder pod pool size based on a time schedule.

Note that CPU and memory utilization are **not** good metrics for scaling Backgrounder pods. Memory and CPU utilization do not accurately portray the overall load demand on the cluster. For example, a Backgrounder pod could be at max utilization to refresh an extract, but there

are no additional jobs waiting in the Backgrounder job queue. In this case autoscaling would not improve job throughput.

Scheduled Scaling

Standard Kubernetes mechanisms using cron jobs allow you to schedule a scaling solution.

An example Kubernetes configuration for this is provided in the Kubernetes Configuration section below.

Kubernetes Configuration

New Environment Variables

In addition to the standard Tableau Server container environment variables (see Initial Configuration Options), there are some new required environment variables that must be set in the Kubernetes Configuration.

Environment Variable	Recommended Value	Description
`FILESTORE_MOUNT_PATH`	`/docker/dataengine`	External Filestore directory mount location. This directory should point to the dataengine NFS directory mounted inside each deployed Tableau container. For more information about External Filestore, see Tableau Server External File Store. The value should be the same for the Tableau Server in a Container pod and the Backgrounder pod.
`CLONE_ARTIFACT_DIR`	`/docker/clone/clone-artifacts`	Clone configuration directory mount location. This directory should point to an NFS directory mounted inside each Tableau container. Tableau Server will ouput configuration data that Backgrounder Pods consume to

Environment Variable	Recommended Value	Description
		become members of the cluster.
ROOT_LOG_DIR	/var/log/tableau	(Backgrounder pods only) Common log directory location for all services running in a backgrounder pod.

Backgrounder Pod Ports

Backgrounder pods consist of four services, each by default is set to run on a specified port. If you want to change the port the service attaches to inside the container, you need to supply the key that corresponds to the service's port assignment. This kind of configuration should not be necessary in most cases, unless there is a sidecar container or some other additional component that is being added to the pod and conflicts with a service's port.

Port Environment Variable	Default
BACKGROUNDER_PORT	8600
DATASERVER_PORT	8400
HYPER_PORT	8200
GATEWAY_PORT	8080

Dataserver also uses port 8300, which cannot be reconfigured.

Shared Network Directory

This deployment of Tableau Server requires two network shares to function properly. Note in all Tableau Server and Backgrounder od Kubernetes configuration templates these network directories are present:

- **Dataengine Directory** (`FILESTORE_MOUNT_PATH`): Backgrounder Pods require the External Filestore feature. This network share contains extracts and other file-based artifacts that will be shared between Tableau Server and Backgrounder Pods.
- **Clone Directory** (`CLONE_ARTIFACT_DIR`): Tableau Server writes static connection and configuration information to a network share. Backgrounder pods will consume this information to become members of the Tableau Server cluster. In future pre-releases this configuration will be incorporated into the standard life cycle of Kubernetes configuration.

Important: If you want to redeploy the cluster entirely (including a new Tableau Server Container), you must purge the contents of the clone NFS mount (otherwise backgrounder pods will attempt to attach to the old server).

Kubernetes Configuration Examples

Note: The configuration examples include use of a readiness probe. You can use the readiness probe when your Tableau Server Container deployment is a single-node TSM deployment (the deployment can include multiple backgrounder pods). You cannot use a readiness probe with multi-node Tableau Server in a Container deployments.

Tableau Server Container Config

```
---
apiVersion: v1
kind: Service
metadata:
  name: <service_name>
  namespace: <namespace>
spec:
  selector:
    app: <service_name>
  ports:
    - protocol: TCP
      port: 8080
```

```
      nodePort: <nodeport-number>
      name: http
    - protocol: TCP
      port: 8443
      nodePort: <nodeport-number>
      name: https
  type: NodePort
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: configfile
  namespace: <namespace>
data:
  config.json: |-
    {
      "configEntities": {
      "identityStore": {
        "_type": "identityStoreType",
        "type": "local"
      }
    },
    "configKeys" : {
      "tabadmincontroller.init.smart_defaults.enable" : "false",
      "wgserver.domain.ldap.starttls.enabled" : "false"
    },
    "daysLeftForMaintenanceExpiring" : 0
  }
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: extrepojsonfile
  namespace: <namespace>
data:
  config.json: |-
    {
```

```
        "flavor":"generic",
        "masterUsername":"<admin-name>",
        "masterPassword":"<password>",
        "host":"<hostname>",
        "port":5432,
        "prerequisiteCheckEnabled":false,
        "requireSsl":false
      }
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: datadir1
  namespace: <namespace>
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
    storage: 50Gi
---
# This is required for multi-node tableau server in container
apiVersion: v1
kind: PersistentVolume
metadata:
  name: bootstrapnfs
  namespace: <namespace>
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 1Gi
  nfs:
    server: <nfs-ip>
    path: <mount-path>
---
# This is required for multi-node tableau server in container
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: bootstrapnfs
  namespace: <namespace>
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: ""
  resources:
    requests:
      storage: 1Mi
---
apiVersion: v1
kind: PersistentVolumn
metadata:
  name: clonenfs
  namespace: <namespace>
spec:
  accessModes:
    - ReadWriteMany
capacity:
  storage: 1Gi
nfs:
  server: <nfs-ip>
  path: <mount-path>
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: clonenfs
  namespace: <namespace>
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: ""
  resources:
```

```
      requests:
        storage: 1Mi
---
apiVersion: v1
kind: PersistentVolume
metadata:
    name: dataenginenfs
    namespace: <namespace>
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 1Gi
  nfs:
    server: <nfs-ip>
    path: <namespace>
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: dataenginenfs
  namespace: <namespace>
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: ""
  resources:
    requests:
      storage: 1Mi
---
apiVersion: v1
kind: Secret
type: Opaque
metadata:
  name: tableau-server-in-a-container-secrets
  namespace: <namespace>
stringData:
```

```
  license_key: <license_key> # Tableau License Key String
  tableau_username: <tableau_username> # Initial admin username in
Tableau Server
  tableau_password: <tableau_password> # Initial admin password
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: tableau-server
  namespace: <namespace>
  labels:
    app: <service_name>
spec:
  selector:
    matchLabels:
      app: <service_name>
  replicas: 1
  serviceName: <service_name>
  template:
    metadata:
      labels:
        app: <service_name>
    spec:
      securityContext:
        runAsUser: 999
        fsGroup: 998
        fsGroupChangePolicy: "OnRootMismatch"
      terminationGracePeriodSeconds: 120
      dnsPolicy: "None"
      dnsConfig:
        nameservers:
          - <dns_ip> # DNS IP for resolving container hostnames
        searches:
          - <service_name>.<namespace>.svc.<cluster_
domain>.<example> # SRV Record
          - <namespace>.svc.<cluster_domain>.<example> # SRV Record
          - svc.<cluster_domain>.<example> # SRV Record
```

```
          - <cluster_domain>.<example> # SRV Record
        options:
          - name: ndots
            value: "5"
      initContainers: # init containers are optional, to clear dir-
ectory content if already exists
        - name: clean-bootstrap-dir
          image: busybox:1.28
          securityContext:
            runAsUser: 0
            allowPrivilegeEscalation: true
          volumeMounts:
          - name: bootstrap
            mountPath: /docker/config/bootstrap
          command: ['sh', '-c', 'rm -rf /docker/config/bootstrap/*
|| true']
        - name: clean-clone-artifacts-dir
          image: busybox:1.28
          securityContext:
            runAsUser: 0
            allowPrivilegeEscalation: true
          volumeMounts:
          - name: clone
            mountPath: /docker/clone
          command: ['sh', '-c', 'rm -rf /docker/clone/clone-arti-
facts || true']
      containers:
      - name: <container_name> # Name of container
        image: <tableau_server_image> # Tableau Server in Container
Image
        env:
        - name: LICENSE_KEY
          valueFrom:
            secretKeyRef:
              name: tableau-server-in-a-container-secrets
              key: license_key
        - name: FILESTORE_MOUNT_PATH
```

```
    value: /docker/dataengine
  - name: CLONE_ARTIFACT_DIR_FOR_INDEPENDENT_CONTAINERS
    value: /docker/clone/clone-artifacts
  - name: SERVER_FOR_INDEPENDENT_SERVICE_CONTAINERS
    value: "1"
  - name: EXT_REP_JSON_FILE
    value: /docker/external-repository/config.json
  - name: TABLEAU_USERNAME
    valueFrom:
      secretKeyRef:
        name: tableau-server-in-a-container-secrets
        key: tableau_username
  - name: TABLEAU_PASSWORD
    valueFrom:
      secretKeyRef:
        name: tableau-server-in-a-container-secrets
        key: tableau_password
  resources:
    requests:
      memory: 40Gi
      cpu: 15
    limits:
      memory: 40Gi
      cpu: 15
  ports:
  - containerPort: 8080
  volumeMounts:
  - name: configmount
    mountPath: /docker/config/config.json
    subPath: config.json
  - name: externalrepomount
    mountPath: /docker/external-repository
  - name: datadir1
    mountPath: /var/opt/tableau
  - name: bootstrap
    mountPath: /docker/config/bootstrap
  - name: clone
```

```
        mountPath: /docker/clone
      - name: dataengine
        mountPath: /docker/dataengine
      imagePullPolicy: IfNotPresent
      startupProbe:
        exec:
          command:
          - /bin/sh
          - -c
          - /docker/server-ready-check
        initialDelaySeconds: 300
        periodSeconds: 15
        timeoutSeconds: 30
        failureThreshold: 200
      readinessProbe:
        exec:
          command:
          - /bin/sh
          - -c
          - /docker/server-ready-check
        periodSeconds: 30
        timeoutSeconds: 60
      livenessProbe:
        exec:
          command:
          - /bin/sh
          - -c
          - /docker/alive-check
        initialDelaySeconds: 600
        periodSeconds: 60
        timeoutSeconds: 60
    volumes:
    - name: configmount
      configMap:
        name: configfile
    - name: externalrepomount
      configMap:
```

```
    name: extrepojsonfile
- name: datadir1
  persistentVolumeClaim:
    claimName: datadir1
- name: bootstrap
  persistentVolumeClaim:
    claimName: bootstrapnfs
- name: clone
  persistentVolumeClaim:
    claimName: clonenfs
- name: dataengine
  persistentVolumeClaim:
    claimName: dataenginenfs
```

Backgrounder Pod Config

```
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: logvolume
  namespace: <namespace>
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 20Gi
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backgrounder
  labels:
    app: backgrounder
  namespace: <namespace>
spec:
  replicas: 2
  selector:
    matchLabels:
      app: backgrounder
  template:
    metadata:
      labels:
        app: backgrounder
    spec:
      securityContext:
        runAsUser: 999
        runAsGroup: 998
        fsGroup: 998
        fsGroupChangePolicy: "OnRootMismatch"
      hostname: backgrounder
```

```
      dnsPolicy: "None"
      dnsConfig:
        nameservers:
          - <dns_ip> # DNS IP for resolving container hostnames
        searches:
          - <service_name>.<namespace>.svc.<cluster_
domain>.<example> # SRV Record
          - <namespace>.svc.<cluster_domain>.<example> # SRV Record
          - svc.<cluster_domain>.<example> # SRV Record
          - <cluster_domain>.<example> # SRV Record
        options:
          - name: ndots
            value: "5"
      initContainers:
        - name: init-myservice
          image: busybox # This init-container is optional (as long
as there is a mechanism to set the log volume directory permissions
and the pod waits for clone artifacts)
          securityContext:
            runAsUser: 0
            allowPrivilegeEscalation: true
          env:
            - name: CLONE_ARTIFACT_DIR_FOR_INDEPENDENT_CONTAINERS
            value: /docker/clone/clone-artifacts
          volumeMounts:
          - name: logmount
            mountPath: /var/log/tableau
          - name: clone-volume
            mountPath: /docker/clone
          command: ['sh', '-c', "chmod 777 /var/log/tableau &&
while [ ! -d ${CLONE_ARTIFACT_DIR_FOR_INDEPENDENT_CONTAINERS} ]; do
sleep 5; done"]
      containers:
      - name: backgrounder
        image: <backgrounder_image> # Backgrounder Image
        ports:
        - containerPort: 8600
```

```
          imagePullPolicy: IfNotPresent
          readinessProbe:
            exec:
              command:
                - /bin/sh
                - -c
                - /tsm_docker_utils/status_check.sh | grep -E
  'ACTIVE|BUSY'
            periodSeconds: 30
            timeoutSeconds: 60
          livenessProbe:
            exec:
              command:
                - /bin/sh
                - -c
                - /tsm_docker_utils/status_check.sh | grep -E
  'ACTIVE|BUSY'
            initialDelaySeconds: 600
            periodSeconds: 60
            timeoutSeconds: 60
          volumeMounts:
          - name: logmount
            mountPath: /var/log/tableau
          - name: clone-volume
            mountPath: /docker/clone
          - name: dataengine-volume
            mountPath: /docker/dataengine
          - name: temp
            mountPath: /var/opt/tableau/tableau_server-
  /data/tabsvc/temp
          env:
            - name: ROOT_LOG_DIR
              value: /var/log/tableau
            - name: CLONE_ARTIFACT_DIR_FOR_INDEPENDENT_CONTAINERS
              value: /docker/clone/clone-artifacts
            - name: FILESTORE_MOUNT_PATH
              value: /docker/dataengine
```

```
      - name: dataserver
        image: <dataserver_image> # Dataserver Image
        ports:
        - containerPort: 8400
        imagePullPolicy: IfNotPresent
        readinessProbe:
          exec:
            command:
              - /bin/sh
              - -c
              - /tsm_docker_utils/status_check.sh | grep -E
'ACTIVE|BUSY'
          periodSeconds: 30
          timeoutSeconds: 60
        livenessProbe:
          exec:
            command:
              - /bin/sh
              - -c
              - /tsm_docker_utils/status_check.sh | grep -E
'ACTIVE|BUSY'
          initialDelaySeconds: 600
          periodSeconds: 60
          timeoutSeconds: 60
        volumeMounts:
        - name: logmount
          mountPath: /var/log/tableau
        - name: clone-volume
          mountPath: /docker/clone
        - name: dataengine-volume
          mountPath: /docker/dataengine
        - name: temp
          mountPath: /var/opt/tableau/tableau_server-
/data/tabsvc/temp
        env:
          - name: ROOT_LOG_DIR
            value: /var/log/tableau
```

```
            - name: CLONE_ARTIFACT_DIR_FOR_INDEPENDENT_CONTAINERS
              value: /docker/clone/clone-artifacts
            - name: FILESTORE_MOUNT_PATH
              value: /docker/dataengine
        - name: gateway
          image: <gateway_image> # Gateway Image
          ports:
          - containerPort: 8080
          imagePullPolicy: IfNotPresent
          readinessProbe:
            exec:
              command:
                - /bin/sh
                - -c
                - /tsm_docker_utils/status_check.sh | grep -E
'ACTIVE|BUSY'
            periodSeconds: 30
            timeoutSeconds: 60
          livenessProbe:
            exec:
              command:
                - /bin/sh
                - -c
                - /tsm_docker_utils/status_check.sh | grep -E
'ACTIVE|BUSY'
            initialDelaySeconds: 600
            periodSeconds: 60
            timeoutSeconds: 60
          volumeMounts:
          - name: logmount
            mountPath: /var/log/tableau
          - name: clone-volume
            mountPath: /docker/clone
          - name: dataengine-volume
            mountPath: /docker/dataengine
          - name: temp
            mountPath: /var/opt/tableau/tableau_
```

```
server/data/tabsvc/temp
        env:
          - name: ROOT_LOG_DIR
            value: /var/log/tableau
          - name: CLONE_ARTIFACT_DIR_FOR_INDEPENDENT_CONTAINERS
            value: /docker/clone/clone-artifacts
          - name: FILESTORE_MOUNT_PATH
            value: /docker/dataengine
    - name: hyper
      image: <hyper_image> # Hyper Image
      ports:
      - containerPort: 8200
      imagePullPolicy: IfNotPresent
      readinessProbe:
        exec:
          command:
            - /bin/sh
            - -c
            - /tsm_docker_utils/status_check.sh | grep -E
'ACTIVE|BUSY'
        periodSeconds: 30
        timeoutSeconds: 60
      livenessProbe:
        exec:
          command:
          - /bin/sh
          - -c
          - /tsm_docker_utils/status_check.sh | grep -E
'ACTIVE|BUSY'
        initialDelaySeconds: 600
        periodSeconds: 60
        timeoutSeconds: 60
      volumeMounts:
        - name: logmount
          mountPath: /var/log/tableau
        - name: clone-volume
          mountPath: /docker/clone
```

```
          - name: dataengine-volume
            mountPath: /docker/dataengine
          - name: temp
            mountPath: /var/opt/tableau/tableau_server-
  /data/tabsvc/temp
        env:
          - name: ROOT_LOG_DIR
            value: /var/log/tableau
          - name: CLONE_ARTIFACT_DIR_FOR_INDEPENDENT_CONTAINERS
            value: /docker/clone/clone-artifacts
          - name: FILESTORE_MOUNT_PATH
            value: /docker/dataengine
      volumes:
        - name: clone-volume
          nfs:
            server: <nfs_ip>
            path: <mount_path>
        - name: dataengine-volume
          nfs:
            server: <nfs_ip>
            path: /dataengine
        - name: logmount
          persistentVolumeClaim:
            claimName: logvolume
        - name: temp
          emptyDir: {}
```

Scheduled Scaling Config

```
apiVersion: v1
kind: ServiceAccount
metadata:
name: backgrounder-scaler-service-account
namespace: <namespace> # Namespace
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
name: scale-backgrounder-pods
namespace: <namespace> # Namespace
subjects:
- kind: ServiceAccount
name: backgrounder-scaler-service-account
roleRef:
kind: ClusterRole
name: cluster-admin
apiGroup: rbac.authorization.k8s.io
---
apiVersion: batch/v1beta1
kind: CronJob
metadata:
name: scale-up-job
namespace: <namespace> # Namespace
spec:
schedule: "0 7 * * *" # Cron Job timing to scale up deployment rep-
licas
jobTemplate:
spec:
template:
spec:
serviceAccountName: backgrounder-scaler-service-account
restartPolicy: OnFailure
containers:
- name: scale
image: bitnami/kubectl:1.21
```

```
imagePullPolicy: IfNotPresent
args:
- scale
- --replicas=4
- deployment/backgrounder
---
apiVersion: batch/v1beta1
kind: CronJob
metadata:
name: scale-down-job
namespace: <namespace>
spec:
schedule: "0 9 * * *" # Cron Job timing to scale down deployment rep-
licas
jobTemplate:
spec:
template:
spec:
serviceAccountName: backgrounder-scaler-service-account
restartPolicy: OnFailure
containers:
- name: scale
image: bitnami/kubectl:1.21
imagePullPolicy: IfNotPresent
args:
- scale
- --replicas=2
- deployment/backgrounder
```

Kubernetes Job to clean Clone Configuration (Optional)

This is a convenience Kubernetes job you might use during testing. If you want to clear out the clone configuration produced by Tableau Server in a Container between distinct deployment runs, you can run a job like this to clean the NFS.

```
apiVersion: batch/v1
kind: Job
metadata:
```

```
name: delete-clone-artifacts-job
namespace: manatee-cluster
spec:
template:
spec:
containers:
- name: delete-clone-artifacts
image: busybox:1.28
command: ['sh', '-c', "rm -rf ${CLONE_ARTIFACT_DIR}"]
env:
- name: CLONE_ARTIFACT_DIR
value: /docker/clone/clone-artifacts
securityContext:
runAsUser: 0
allowPrivilegeEscalation: true
volumeMounts:
- name: clone-volume
mountPath: /docker/clone
restartPolicy: Never
volumes:
- name: clone-volume
nfs:
server: <nfs_ip> # IP for shared NFS directory for clone output
path: /clone
```

About Data Management

Important: As of September 16, 2024, Data Management is no longer available as an independent add-on option. Data Management capabilities are only available if you previously purchased Data Management, or if you purchase certain license editions - either

> Tableau Enterprise (for Tableau Server or Tableau Cloud) or Tableau+ (for Tableau Cloud).

Data Management is a collection of features and capabilities that helps customers manage Tableau content and data assets in their Tableau Server or Tableau Cloud environment.

Starting in Tableau Server 2019.1, Tableau Prep Conductor is available for on-premise Tableau Server deployments, and in version 2019.3, Tableau Prep Conductor is available for Tableau Cloud deployments. You can use Tableau Prep Conductor to schedule and monitor flows.

Starting in Tableau 2019.3, Tableau Catalog is included in Data Management, making a variety of additional features available to you in the data management space. You can use Tableau Catalog to discover data, curate data assets, communicate data quality, perform impact analysis, and trace the lineage of data used in Tableau content.

Starting in Tableau 2021.4, more governance and security features are added to Data Management: virtual connections and data policies. Using the virtual connection editor, you can create:

- Virtual connections that provide a sharable central access point to data.
- Data policies to apply row-level security at the connection level.

Data Management Features

The following table lists the features for Data Management, which include:

- Tableau Catalog
- Tableau Prep Conductor
- Virtual connections
- Data policies

Tableau Catalog

Feature	Description	
Permissions on metadata	Tableau Catalog enables you to control who can see and manage external assets and what metadata is shown through lineage by setting permissions.	
Expanded connect experience - data discovery	Whether you author in the web or in Tableau Desktop, you can now search for and connect to the specific databases and tables used by published data sources and workbooks on your Tableau Server or Tableau Cloud site.	
Expanded search	Tableau Catalog expands search to include results based on columns, databases, and tables.	
Tag external assets	You can categorize items on Tableau Server and Tableau Cloud with tags, helping users to filter external assets (databases, files, tables, and columns).	
Certify databases and tables	Help users find trusted data that meets the standards you set by certifying databases and tables.	
Set data quality warnings	You can set warnings to alert users to data quality issues, such as stale or deprecated data.	
Lineage and impact analysis	The Lineage tool traces the source of your data. You can use it to analyze the impact of changes to your data, identify which users might be impacted, and email owners of a workbook, data source, or flow, or contacts for a database or table, about data-related updates.	
Data Details	Enable users to better understand a published visualization by seeing information about the data used.	
Add descriptions to assets	Help users find the data they're looking for by adding descriptions to databases, tables, and columns.	
Developer resources	Tableau REST API - metadata methods	Programmatically add, update, and remove external assets; and add additional

Feature	Description	
		metadata to Tableau content and external assets like descriptions.
	Tableau Metadata API	Programmatically query metadata from the content published to Tableau Server or Tableau Cloud. Programmatically update certain metadata using the metadata methods in the Tableau Server REST API. **Note:** The Metadata API does not require Data Management.
	GraphiQL	Explore and test queries against the Metadata API schema using an interactive in-browser tool called GraphiQL. **Note:** GraphiQL does not require Data Management.

Tableau Prep Conductor

Feature	Description
Schedule Flow Tasks in the Tableau Cloud or Tableau Server help.	You can create scheduled flow tasks to run a flow at a specific time or on a recurring basis.
Monitor Flow Health and Performance	Set up email notifications at the site or server level when flows fail, view and resume suspended flow tasks, and view errors and alerts.
Administrative Views for Flows	Use Administrative Views to monitor the activities related to flows, performance history, and the disk space used at the server or site level.
Tableau REST API - flow methods	Programmatically schedule flows.

Virtual connections and data policies

Feature	Description
Create a Virtual Connection	A Tableau content type that enables you to create a shareable re-usable connection to curated data.
Create a Data Policy for Row-Level Security	Use the virtual connection editor to create data policies with policy conditions that apply row-level security to the data at the connection level.
Test Row-Level Security with Preview as User	Test the data policy with Preview as user to ensure that users can see only their data.
Schedule Extract Refreshes for a Virtual Connection	Create an extract refresh schedule for the tables in your connection, ensuring that the data is fresh for any content that uses that virtual connection.

License Data Management

Important: As of September 16, 2024, Data Management is no longer available as an independent add-on option. Data Management capabilities are only available if you previously purchased Data Management, or if you purchase certain license editions - either Tableau Enterprise (for Tableau Server or Tableau Cloud) or Tableau+ (for Tableau Cloud).

Data Management includes Tableau Catalog, Tableau Prep Conductor, virtual connections, and data policies and is available when you purchase Tableau Enterprise. Contact your account manager (or go to the Tableau pricing page) for more information.

Data Management can only be activated on a licensed Tableau Server Deployment. A Deployment includes a licensed production Tableau Server installation and licensed non-production

Tableau Server installations that support the production installation. For more information on Deployment, see the EULA Documentation.

Tableau Prep Conductor

When you purchase Tableau Enterprise, you must enable Prep Conductor on Tableau Server. For more information, see Enable and Configure Tableau Prep Conductor on Tableau Server.

- When Data Management is active and enabled, you can schedule flows in Tableau Server or Tableau Cloud and monitor flows.
- When Data Management is removed or deactivated, or if the Data Management license expires, then the ability to schedule flows is disabled.
- If your Tableau Server or Tableau Cloud license is still active and valid, you can download the flows using the Tableau Server REST API. For more information, see Flow Methods.

Tableau Catalog

When you purchase Tableau Enterprise, you must enable Catalog on Tableau Server. For more information, see Enable Tableau Catalog.

- When Data Management is active and enabled, you can use Tableau Catalog to discover data, curate data assets, perform impact analysis, and trace the lineage of data used in Tableau content.
- When Data Management is removed, deactivated, or the license expires, the information remains on the server. The Tableau Catalog-specific information is then only accessible using the Tableau Metadata API; it no longer appears in the product. For more information, see the Metadata API.
- When Data Management is removed, deactivated, or the license expires, the write APIs for all new Tableau Catalog information (for example, table descriptions, data quality warnings, column descriptions) are disabled. You can still read information using the Metadata API, however permissions on tables and databases can't be explicitly managed in the product.

Virtual connections and data policies

When you purchase Tableau Enterprise, and enable Data Management for Tableau, virtual connections and data policies are automatically enabled on Tableau Server.

After you purchase and license Data Management for Tableau, virtual connections and data policies are automatically enabled.

- When Data Management is active and enabled, you can use virtual connections to create sharable resources that provide a central access point to data. You can also create data policies that enable you to filter data for users using centralized row-level security.
- When Data Management is removed, deactivated, or the license expires, the information remains on the server but is not accessible.
- When Data Management is reactivated, the information is restored on the server and becomes accessible.

How Data Management licensing works

A Tableau Server Deployment may be user-based or core-based, depending on which license you purchase.

User-Based

A User-Based license metric allows you to deploy Tableau Server on a single computer or on multiple computers in a cluster. Each user that accesses Tableau Server must be licensed. Administrators add users and license them. The first Creator or Explorer product key that you add to Tableau Server activates Tableau Server and will be used by a Server Administrator.

The Data Management product key enables the included features at the deployment level, and those features are licensed for all the users that are already licensed for Tableau Server.

Core-Based

A Core-Based license metric imposes no constraints on the number of user accounts in Tableau Server. Instead, the license specifies the maximum number of computer cores on which you can run Tableau Server.

Note: When you purchase and use core-based licensing, you must apply both the Data Management product key and the Resource Core product key to your Tableau Deployment. The first key allows flows to be run on Tableau Server though the Tableau Prep

Conductor and the second key adds the additional cores for the Tableau Prep Conductor nodes. All product keys are available through the Customer Portal.

Typically, the total number of cores in all the computers should not exceed the total number that the Tableau Server license allows. When you activate the Data Management product key on your Tableau Server deployment, it includes a specific number of Tableau Prep Conductor cores. In this scenario, the total number of cores on all computers should not exceed the total number that the Tableau Server license and the Data Management license together allow.

In this topic we will refer to the cores licensed through Tableau Server license as Tableau Server cores, and the cores licensed through Data Management as Tableau Prep Conductor cores.

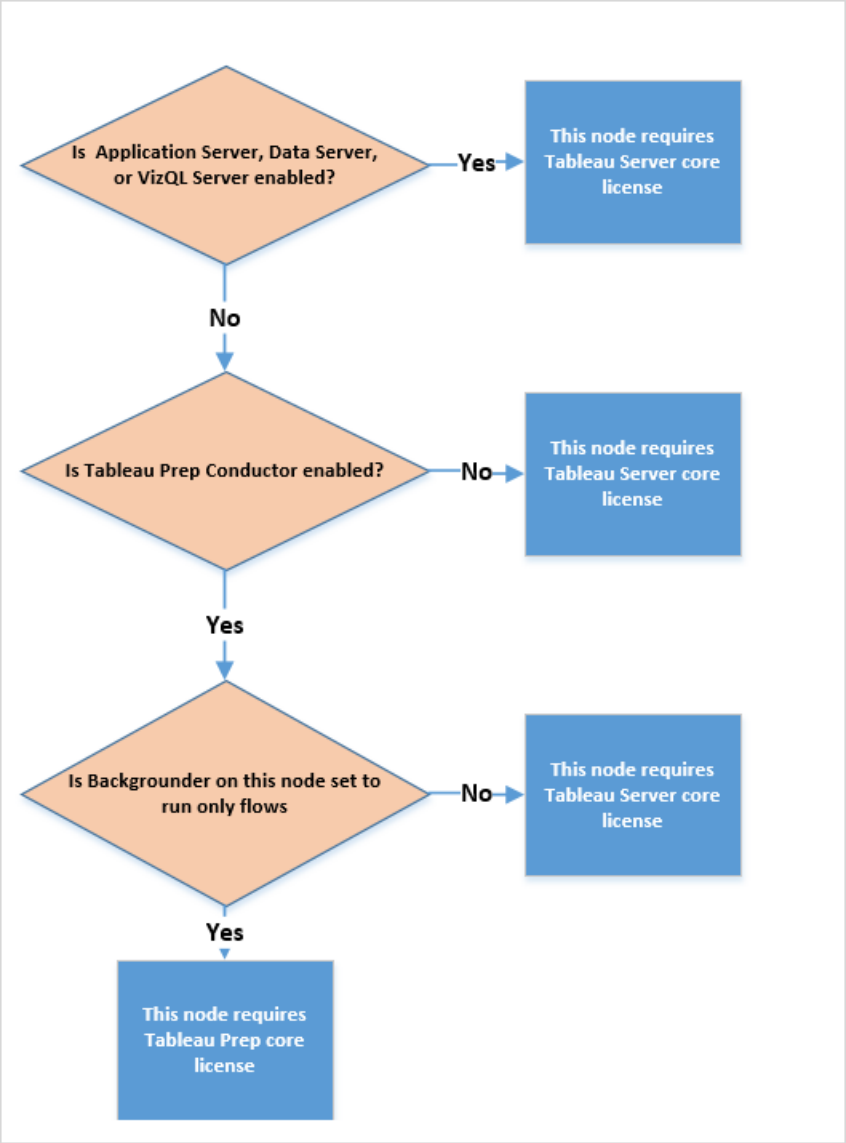
Here are some concepts that apply to how licensing is applied in a Core-Based metric:

- A node can be licensed by only one of type—Tableau Server Cores or Tableau Prep Conductor cores.
- The Tableau Prep Conductor cores are applied to any node that is dedicated to running Tableau Prep Conductor and when the Backgrounder on that node is set to run only flow background jobs. In this case, the total number of cores on this node cannot exceed the number of cores that the Data Management license allows. If this node has any other licensed process besides Tableau Prep Conductor, Backgrounder and Data Engine enabled, then this node will require and use a Tableau Server core license.
- As mentioned above, the Backgrounder node role also affects which license is used by a node. For example, if the Backgrounder node role is set to run jobs of all types (this is the default), then this node will be licensed through the Tableau Server cores. For more information on node roles, see Node Roles in Tableau Server.

  See the following table and decision flow to understand how a node is licensed:

If a node has...	**the core on the node is counted towards...**	**the node is licensed using...**
one of the following	Total count of Tableau	Tableau Server cores.

processes enabled: • Application Server • Backgrounder (node role is set to run all jobs) • File Store • Data Server • VizQL Server	Server cores.	
only the following processes enabled: • Tableau Prep Conductor • Backgrounder (node role is set to run only flows) • Data Engine	Total number of Tableau Prep Cores purchased through Data Management. **Note:** If there are no Tableau Prep Conductor cores available, but Tableau Server cores are available, then Tableau Server cores are used.	Tableau Prep Cores included in Data Management. **Note:** If there are no Tableau Prep Conductor cores available, but Tableau Server cores are available, then Tableau Server cores are used.



For more information about licensed processes, see Tableau Server Processes.

To learn more about Tableau Server licensing, see Tableau Server Licensing Overview.

Tableau Prep Conductor

Tableau Prep Conductor enables you to leverage the scheduling and tracking functionality available in Tableau Server to run your flows automatically to update the flow output. Tableau

Prep Conductor is part of Tableau Data Management introduced in Tableau Server version 2019.1 and must be enabled to schedule your flows to run.

For more information about the Tableau Prep Conductor process on Tableau Server, see Tableau Prep Conductor

> **Note**: Starting in version 2020.4, the Data Management license is required to run flows on a schedule and when using REST API to run flows.
> Data Management isn't required to publish flows and manually run them on the web, and as a Creator, create and edit flows directly on your server.

Flows created in Tableau Prep Builder must be published to Tableau Server before they can be scheduled to run.

Publishing flows is similar to publishing data sources and workbooks with Tableau Desktop. You can package files with the flow or specify a direct connection to data sources to update the flow input as data changes. If your flow connects to databases, specify the authentication type and set credentials to access the data.

You can also publish a flow to share it with others or to continue editing it on the web. For example, publish an incomplete flow to Tableau Server and then open the flow on the web in Edit mode to continue working on it. You could also create a flow with only Input steps (that are properly configured) and share it with co-workers who can then download the flow to their computers and create and publish their own flows.

For flows to run they must include output steps and have no errors or incompatible features. For more information about publishing a flow, see Publish a Flow to Tableau Server or Tableau Cloud. For more information about incompatibility, see Version Compatibility with Tableau Prep.

Keeping track of the health of your flows is easy. If a flow fails to run due to errors, such as a calculation that isn’t valid or a connection failed, you can fix the error right in Tableau Server.

You can edit the connection or edit the flow to fix the error, then republish it to pick up where you left off.

The following table shows the flow management features that are available with and without the Data Management and Tableau Prep Conductor enabled.

Data Management with Tableau Prep Conductor enabled	**No Data Management**
• View and monitor the details about your flow, including recent activity in the **Content** pages. • Edit your flow (starting in version 2020.4). • View the results of the flow runs and any errors in the **Run History** tab. • Use **Administrative Views** to monitor server and site activity including a new view that tracks flow performance history. • Run flows using REST API. • View detailed alerts for failed flow runs. • Set up email notification alerts to send emails to flow owners notifying them when the flow failed to run and why. For more information about setting up alerts, Monitor Flow Health and Performance.	• View the details about your flow, including recent activity in the **Content** pages. • Edit your flow (starting in version 2020.4). • View and edit your connections on the **Connections** tab.

Enabling Tableau Prep Conductor on Tableau Server

Before you can start publishing flows to your Tableau Server, there are server-level and site-level settings you must configure or verify to prepare your Tableau Server to allow publishing, scheduling and monitoring flows.

Review the following topics to understand Tableau Prep Conductor licensing, and learn how to enable Tableau Prep Conductor:

- Enable and Configure Tableau Prep Conductor on Tableau Server: This topic provides step-by-step instructions on how to enable and configure Tableau Prep Conductor, and preparing your Tableau Server to allow publishing, scheduling, managing, and maintaining flows.
- License Data Management

About the Flow Workspace

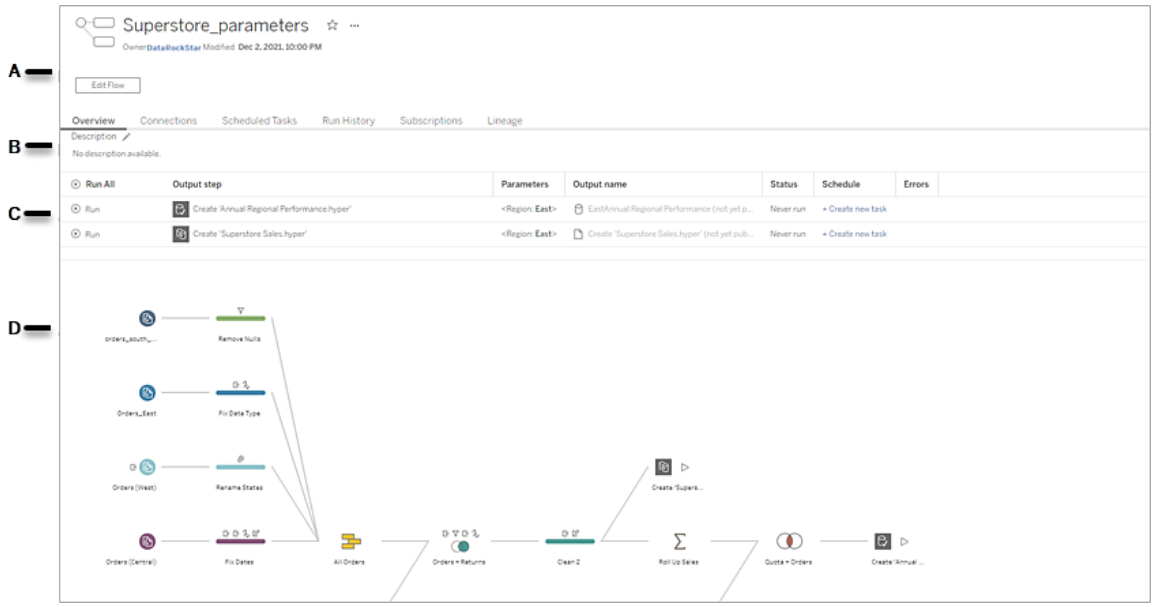
After you publish your flow you can schedule tasks or linked tasks (version 2021.3 and later) in Tableau Server to automatically run your flows on a regular basis to keep your output data fresh. You can also run your flows manually at any time (no Data Management required).

Tableau Prep Conductor leverages much of the same functionality for managing flows that you might see when managing workbooks or data sources from Tableau Desktop in Tableau Server . For example, just like extract refreshes, scheduled flow tasks and on-demand flow runs are queued as background tasks. But when it comes to working with flows, there are a few differences.

Flow Overview page

The flow **Overview** page is the main landing page where you can view data about your flow and schedule, monitor, and maintain the flow. If you don't have the Data Management, you will have different options.

Open the flow **Overview** page by clicking on a flow in your list. You can navigate there from **Content** > **Explore** > **All Flows** or by opening the project that contains your flows.



A. The header lists the name of the flow, the flow owner and the date that the flow was last modified. Starting in version 2020.4, click **Edit** to edit existing flows.

   Add a flow to your favorites, or from the **More actions** ••• menu you can also edit, run, download the flow, set permissions, change the flow owner, restore previous flow versions, and more.

B. View and edit the flow description and set tags to help others find the flows they are looking for.

C. View the output steps for a flow along with any parameters applied to the flow (version 2021.4 and later), the status of the last update, any schedule the output is assigned to, and any errors from the last flow run. You can also click the **Run** button to run all output steps or individual output steps on-demand.

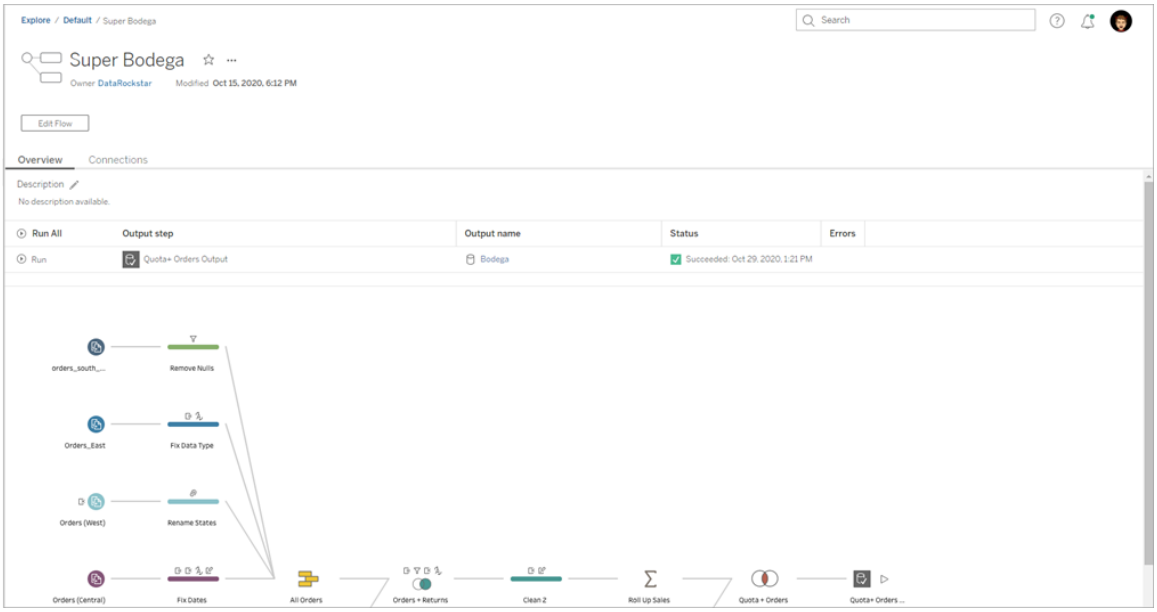
Parameters	If the flow includes user parameters, the parameter value last run in the flow is shown and you can see all generated outputs in the **Output** column. When the flow is run, you'll be prompted to enter the parameter values.

	System parameters (version 2023.2 and later) are automatically generated at flow run time and the type of system parameter is shown in the **Parameters** column. To see the last system parameter value applied to the flow, edit the flow. For more information about using parameters in flows, see Create and Use Parameters in Flows in the Tableau Prep help.
Status	After a flow has run successfully, outputs that are data sources become links that you can click to open the **Data Source** page to view more information about the data source or edit the flow input connection.
Schedule	In the **Schedule** field, view the scheduled tasks that the output step is assigned to. A flow output can be assigned to one or more tasks. If no schedule has been assigned yet, click **Create new task** to add the output step to a schedule. To immediately run the flow to update a specific output step, click the **Run** button on the left-hand side of the row.
Errors	If the flow has errors, the flow run will fail. Connectivity errors can be resolved directly by navigating to the **Connections** tab for the flow and editing the input connections. To resolve any other flow errors, edit the flow then republish it and try running the flow again. If you are using an earlier version of Tableau Prep Builder, from the **More actions** ... menu, you can also download and open the flow in Tableau Prep Builder, then republish it and try running the flow again.

D. View an image of the flow.

Flow Overview page without the Data Management

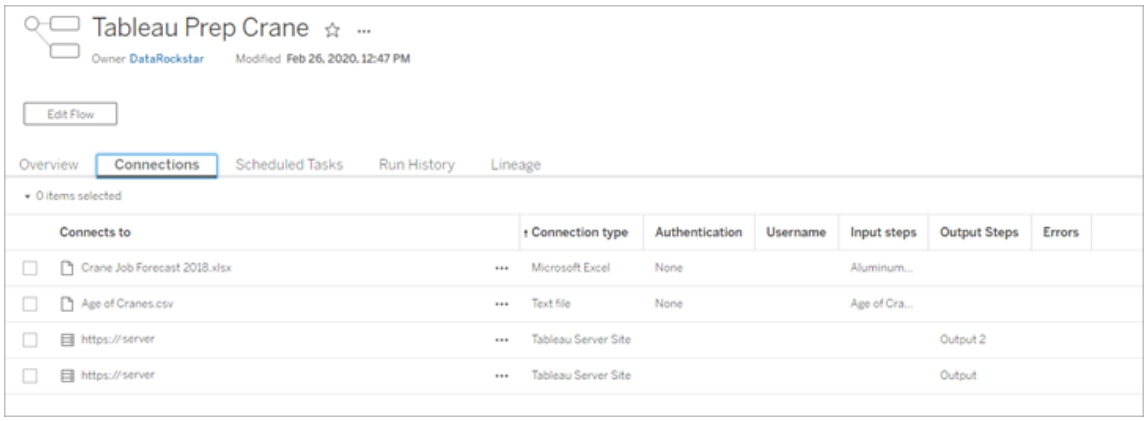
If you don't have the Data Management installed on your server, you can still publish flows to Tableau Server , but you will see fewer options to manage your flow.



Flow Connections page

View both the input and output locations for a flow, connection types, authentication settings, input and output steps and any connectivity errors. You can set authentication settings when publishing a flow. For more information, see Publish a Flow.

For database input types, click the **More actions** ••• menu for an input connection to edit the connection and change the server name, port, user name and password.



Flow Scheduled Tasks page (Data Management required)

View any schedules that the flow is assigned to, the outputs that are included in those schedules, and any parameters applied to the flow (version 2021.4 and later). As an administrator, you can click the schedule link to open the **Schedules** page and see a list of flows that are assigned to that schedule. For more information about assigning flows to a schedule, see *Schedule Flow Tasks* in the Tableau Cloud or Tableau Server help.

To view the outputs on a schedule or the tasks assigned to a linked task (version 2021.3 and later), click the links in the **Schedule type** column.

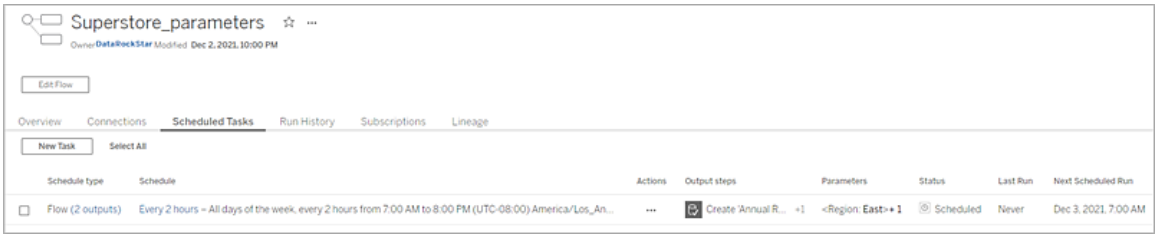
You can also add new tasks or manage existing ones from this page. To take action on an existing task, select the check box on a task card then click the **Actions** drop-down menu to run, edit, or delete the task.

If the flow fails to run after a configured number of consecutive attempts, the flow is automatically suspended.

You can see that status on the **Overview** tab as well as this tab. You can resume suspended tasks from this menu.

For information about how to set the threshold for suspended flow tasks, see Step 5: Optional Server Configurations. For more information about suspended flow tasks, see View and resolve errors.

> **Note**: The Scheduled Tasks page for flows was redesigned in version 2021.3. Your view may look different depending on your server version.

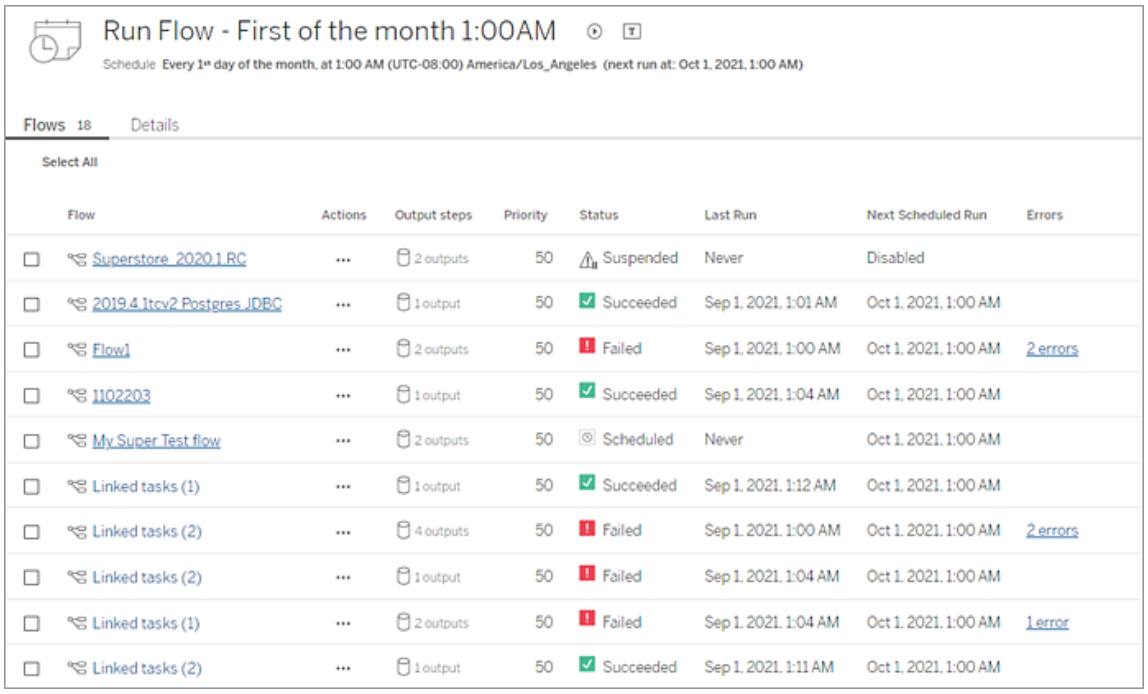


Schedules page

On the **Schedules** page, you can view the flows assigned to a schedule and the details about the flow runs. If the schedule includes linked tasks (version 2021.3 and later) the number of flows included in the linked tasks is shown.

You can run the schedule on-demand and run all flows assigned to it. You can also select one or more flows, then use the **Actions** menu to change the flow schedule or priority, delete selected flows from the schedule or resume suspended flows.

For information about how to set up a schedule, see Step 3: Create Schedules for Flow Tasks.

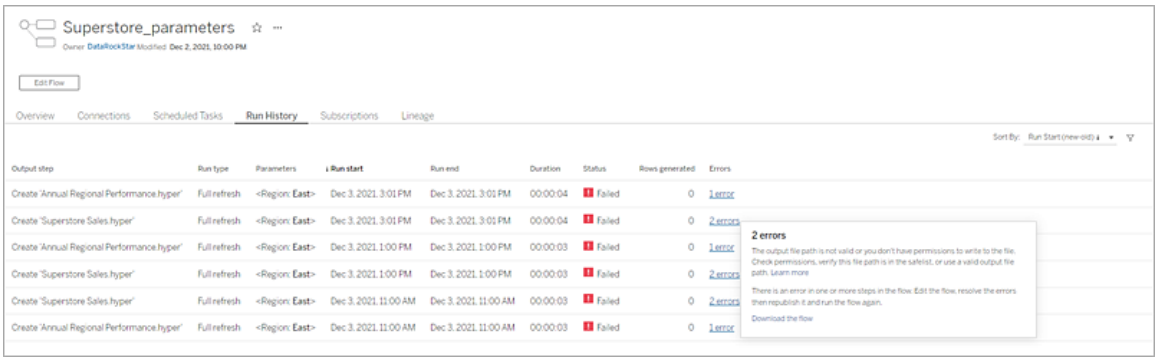


Flow Run History (Data Management required)

See, search, and sort through a list of historical runs for a flow. This page also includes details about the flow run such as run type, parameter values applied to flows included in each flow run (version 2021.4 and later), duration and number of rows that were generated.

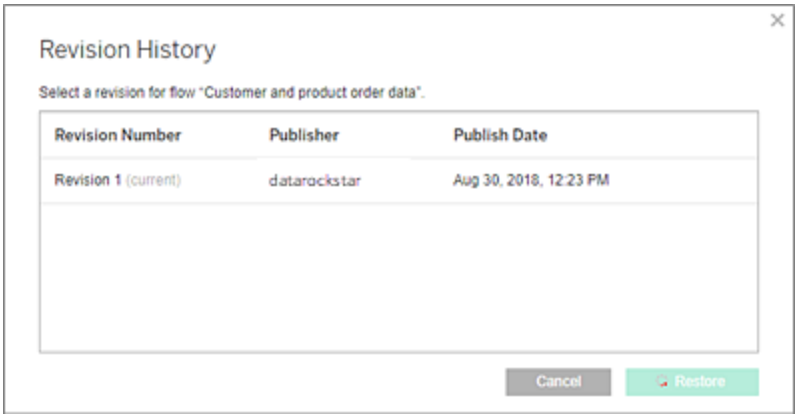
If the flow output has an error, hover over the error to view the messages. If applicable, click the **Go to Connections** link in the error message to navigate to the **Connections** page to fix connectivity errors. You can also edit the flow directly to fix any errors, or click **Download the flow** to download and fix flow errors in Tableau Prep Builder, then republish the flow to continue to manage it using Tableau Prep Conductor.

Note: The run history for a flow will persist unless the flow is deleted.



Flow Revision History

If you need to revert a flow to a previous version, from the **More actions** ••• menu for the flow, select **Revision History**. On the **Revision History** dialog, select the flow version from the list that you want to revert to.



Who can do this

Server Administrators can activate Data Management license keys.

Server administrators can enable Tableau Prep Conductor.

Creators can create, edit, and run flows manually. If the Data Management is installed, creators can run flows on a schedule.

Enable and Configure Tableau Prep Conductor on Tableau Server

Supported in Tableau Server version 2019.1 and later.

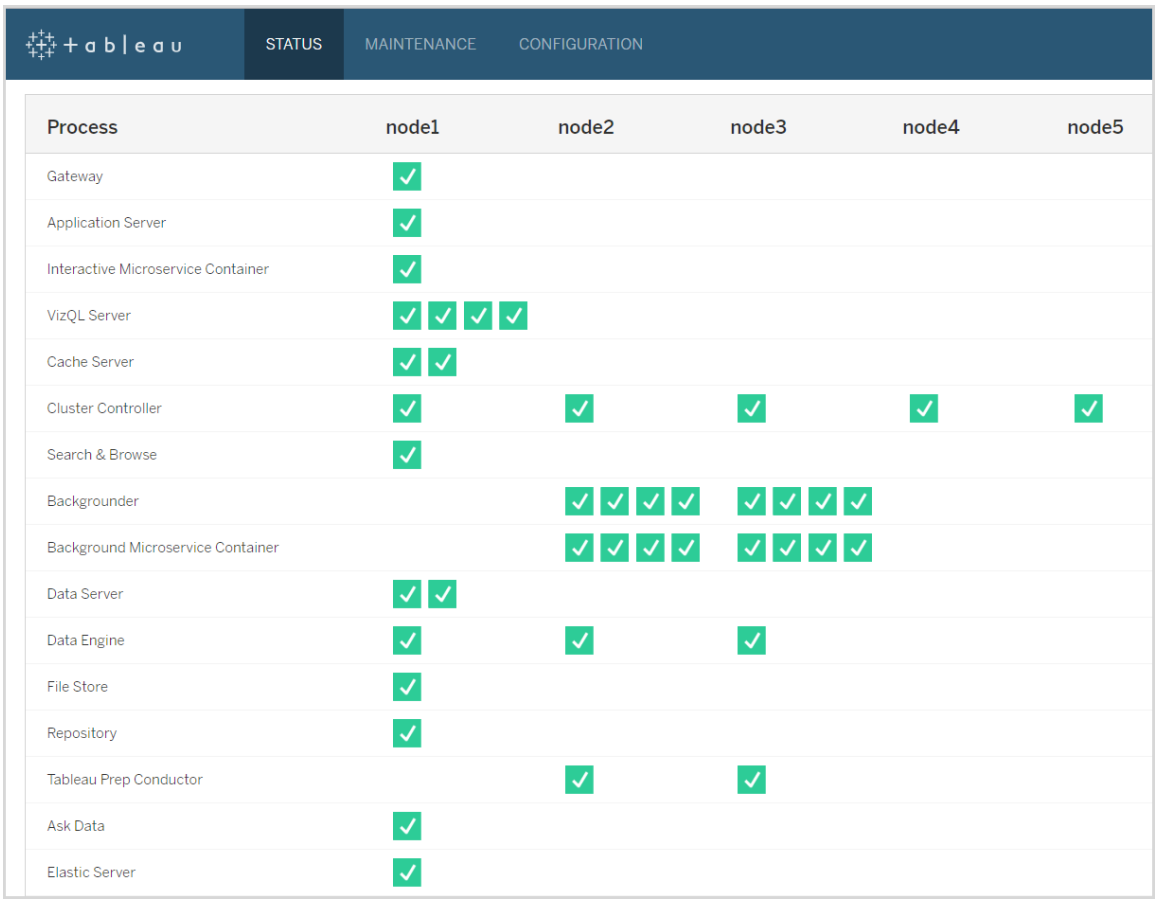
Tableau Prep Conductor is licensed through Data Management, on a per Deployment basis, which is User-Based or Core-Based. A Deployment includes a licensed production Tableau Server installation and licensed non-production Tableau Server installations that support the production installation. For more information on Deployment, see the Tableau Deployment Guide.

For more information on how Tableau Prep Conductor licensing works, see License Data Management.

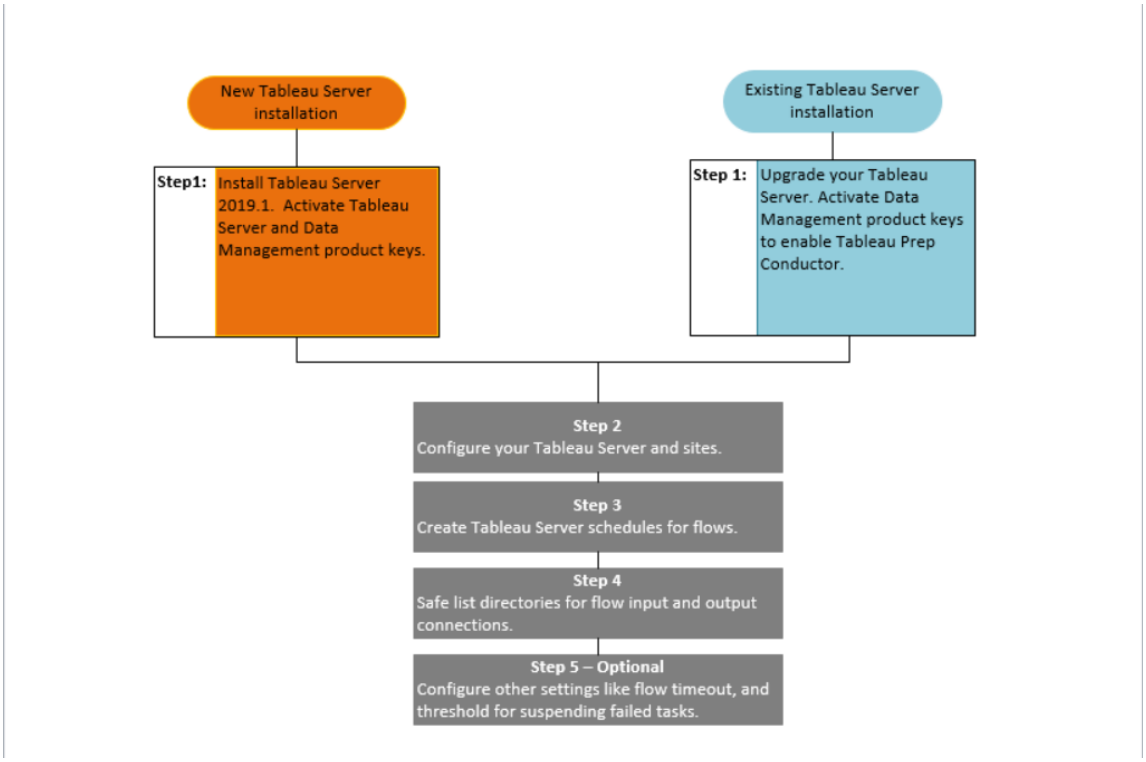
Server Topology

When you install Tableau Server and enable Tableau Prep Conductor, using the Data Management product key, Tableau Prep Conductor is automatically enabled by default by the setup program.

For multi-node installations, by default, one instance of Tableau Prep Conductor is enabled on any node that has backgrounder installed. In the example below, Tableau Prep Conductor is enabled on node 2 and 3 where the backgrounders are also enabled, but not on node 1,4, and 5.



Below is a visual representation of that work-flow:



Next step:

New Tableau Server Installations: Step 1 (New Install): Install Tableau Server with Tableau Prep Conductor

Existing Tableau Server Installations: Step 1 (Existing Install): Enable Tableau Prep Conductor

Who can do this

Server administrators can install Tableau Server and enable Tableau Prep Conductor.

Server-level settings can be configured by Tableau Server administrators, and site-level settings can be configured by Tableau Server and Site administrators.

Step 1 (New Install): Install Tableau Server with Tableau Prep Conductor

This topic describes how to Tableau Prep conductor on a new installation of Tableau Server.

Tableau Prep Conductor is supported only on Tableau Server versions 2019.1 or later.

Tableau Prep Conductor is licensed through Data Management, on a per Deployment basis. A Deployment includes a licensed production Tableau Server installation and licensed non-production Tableau Server installations that support the production installation. For more information on Deployment, see the Tableau Deployment Guide.

Before you install

The recommended topology for a production Tableau Server installation is a dedicated node for running flows. If you are currently planning to have a single node Tableau Server installation it is recommend that you add a second node and dedicate it to run flows.

- Review the hardware recommendations for Tableau Server and Tableau Prep conductor.
    - Minimum Hardware Requirements and Recommendations for Tableau Server installation on Windows.
    - Minimum Hardware Requirements and Recommendations for Tableau Server installation on Linux.

Install Tableau Server and enable Tableau Prep Conductor

Use the instructions provided in the following topics to install Tableau Server.

Windows: Install Tableau Server

Linux: Install Tableau Server topic

When you get to the **Activate** step, use the Tableau Server product keys to activate Tableau Server.

All product keys are available through the Customer Portal.

Configure public gateway settings

If your Tableau Server is set up with one of the following:

- Load balancer to distribute requests across gateways.
- Reverse proxy to authenticate external (internet) client requests and offloading SSL-based encryption.

You must configure the following public gateway settings:

`tsm configuration set -k gateway.public.host -v <name>` (This should be the URL that your users are using to access Tableau Server)

```
tsm configuration set -k gateway.public.port -v 443
```

For more information on configuring gateway settings, see Configuring Proxies for Tableau Server.

Enable Tableau Prep Conductor

Use the following steps to add the Data Management product key to your Tableau Server:

Note: This process requires a restart of the Tableau Server.

Note: If you are using core-based licensing, you must apply both the Data Management product key and the Resource Core product key to your Tableau Deployment. The first key allows flows to be run on Tableau Server though the Tableau Prep Conductor and the second key adds the additional cores for the Tableau Prep Conductor nodes. All product keys are available through the Customer Portal.

1. If the computer where you are running Tableau Server has been configured to connect to the internet through a forward proxy, follow the procedure in the topic, Configure Product Key Operations with Forward Proxy, before continuing.
2. Open TSM in a browser:

```
https://<tsm-computer-name>:8850
```

3. Click **Licensing** on the **Configuration** tab and click **Activate License**.
4. Enter or paste your **Data Management product key** and click **Activate**.
5. On the Register page, enter your information into the fields and click **Register**.
6. Follow the prompts and restart Tableau Server after registration is complete.

Verify Tableau Prep Conductor is enabled and running

When you activate the Data Management product key, a single instance of Tableau Prep Conductor is automatically enabled on any node that has Backgrounder enabled.

Use the following steps to verify that it is enabled and running:

1. Open a browser and enter the Tableau Server URL, and append the dedicated TSM web UI port. Here are some examples of what the URL might look like:

   https://localhost:8850/ (if you're working directly on the server computer)

   https://MarketingServer:8850/ (if you know the server's name)

   https://10.0.0.2:8850/ (if you know the server's IP address)

   In the sign-in page that appears, enter your administrator user name and password.

   > **Note:** Tableau Server creates and configures a self-signed certificate during the installation process. This certificate is used to encrypt traffic to the TSM Web UI. Because it's a self-signed certificate, your browser will not trust it by default. Therefore, your browser will display a warning about the trustworthiness of the certificate before allowing you to connect.

2. In the Tableau Services Manager web interface, click the **Status** tab to see the status.

- If Tableau Prep Conductor is enabled and running, you should see Tableau Prep Conductor in the list of processes as **Active** on at least on one node. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor in the list of processes, but with no status information for any of the nodes.

  Tableau Prep Conductor not enabled:

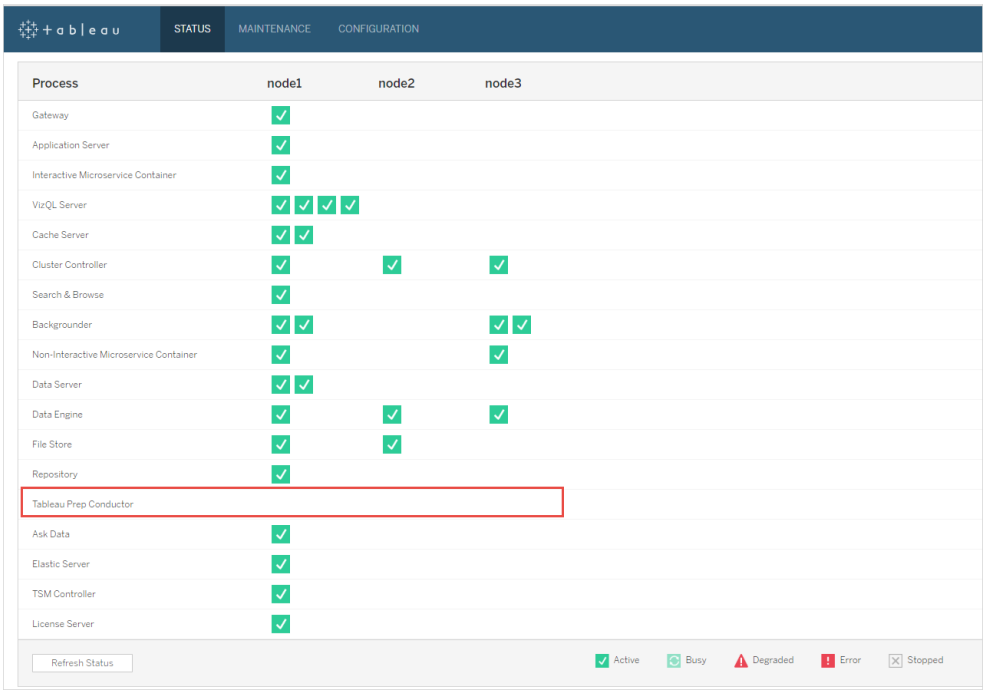
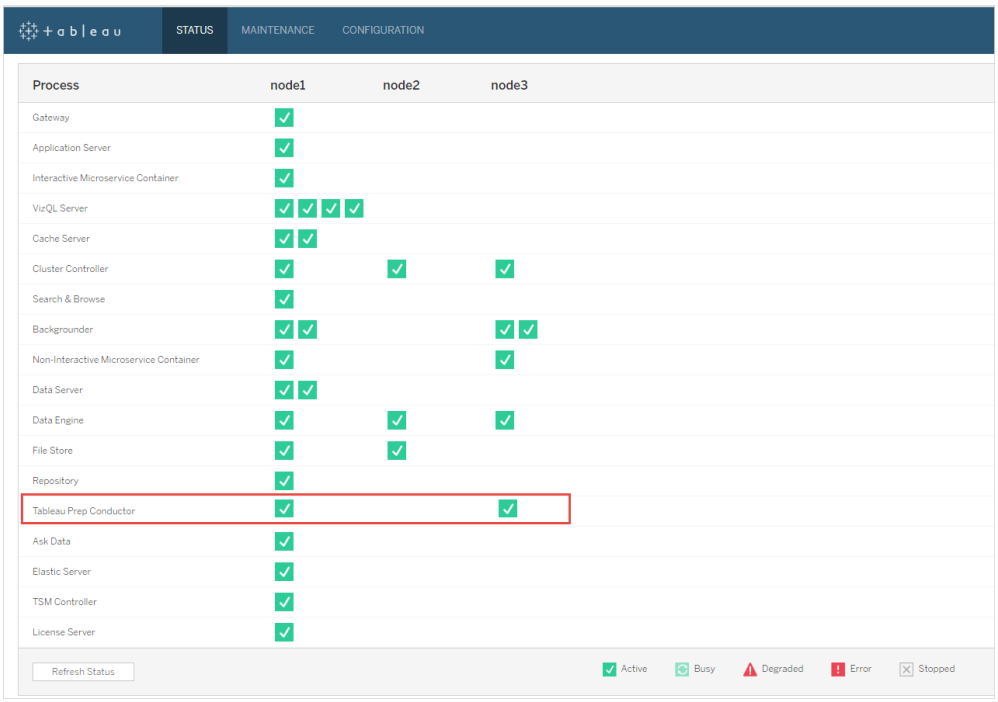


  Tableau Prep Conductor enabled and running. In the image below Tableau Prep Conductor is enabled on node1 and node3:



Dedicate a node for Tableau Prep Conductor

On the node you are planning to dedicate to running flows, enable Backgrounder process if it is not already enabled. It is recommended that you do not run other processes like VizQL server on this node.

Because you are dedicating this node to running flows, you must configure Backgrounder to run only flow tasks. By default, the Backgrounder process runs tasks of all types, including flows, extract refreshes, and subscriptions. For more information, see Node Roles in Tableau Server.

Run the following tsm commands on that dedicated node to run only flow tasks:

1. Run the following command to allow Backgrounders on this node to run only flow tasks.

```
tsm topology set-node-role -n node1 -r flows
```

2. Set the node role on the initial node to no flows. The backgrounder on this node will run all jobs except flows:

   ```
   tsm topology set-node-role -n node1 -r no-flows
   ```

3. Apply the changes and restart Tableau Server:

   ```
   tsm pending-changes apply
   ```

Multi-node installations

If you have more than 2 nodes in your Tableau Server installation, you can choose to configure other nodes to run all tasks other than flows:

1. Restrict a node to not allow flows. This command removes Tableau Prep Conductor from this node and Backgrounders on this node will not run flow tasks.

   ```
   tsm topology set-node-role -n node1 -r no-flows
   ```

2. Apply the changes and restart Tableau Server:

   ```
   tsm pending-changes apply
   ```

Next step

Step 2: Configure Flow Settings for your Tableau Server

Who can do this

Server administrators can install Tableau Server and enable Tableau Prep Conductor.

Step 1 (Existing Install): Enable Tableau Prep Conductor

This topic describes how to enable Tableau Prep conductor on your existing installation of Tableau Server.

Tableau Prep Conductor is supported only on Tableau Server versions 2019.1 or later. If you are using Tableau Server 2018.3 or earlier, you must first upgrade your Tableau Server to 2019.1 before enabling Tableau Prep Conductor on your Tableau Server installation.

Tableau Prep Conductor is licensed through Data Management, on a per Deployment basis, which is User-Based or Core-Based. A Deployment includes a licensed production Tableau Server installation and licensed non-production Tableau Server installations that support the production installation. For more information on Deployment, see the Tableau Deployment Guide.

This topic describes how to enable Tableau Prep conductor on your existing installation of Tableau Server.

Before you upgrade

Prepare for upgrade:

- Know before you upgrade
- Licensing Tableau Prep Conductor
- Tableau Server Hardware Requirements and Recommendations

Configure public gateway settings

If your Tableau Server is set up with one of the following:

- Load balancer to distribute requests across gateways.
- Reverse proxy to authenticate external (internet) client requests and offloading SSL-based encryption.

You must configure the following public gateway settings:

`tsm configuration set -k gateway.public.host -v <name>` (This should be the URL that your users are using to access Tableau Server)

```
tsm configuration set -k gateway.public.port -v 443
```

For more information on configuring gateway settings, see Configuring Proxies for Tableau Server.

Tableau Server Installations using User-Based licenses

The recommended topology for a production Tableau Server installation is a dedicated node for running flows. For more information, see Minimum Hardware Requirements and Recommendations for Tableau Server.

Tableau Server single-node installations

If you currently have a single node Tableau Server installation, it is recommended that you add a second node and dedicate it to running flows.

1. Run upgrade on your current Tableau Server installation using the information in the topics below:

    - Windows
    - Linux

    When you get to the **Activate** step, use the Tableau Server product keys to activate Tableau Server.

    All product keys are available through the Customer Portal.

2. After completing the installation, add the Data Management product key to enable Tableau Prep Conductor on your node. The Data Management product key, like your other server keys, are available through the Customer Portal.

    - In the Tableau Services Manager web interface, click **Licensing** on the **Configuration** tab and click **Activate License**.
    - Enter or paste your new product key and click **Activate**.
    - On the **Register** page, enter your information into the fields and click **Register**.

3. You will be prompted to restart the server. Restart the server and verify that Tableau Prep Conductor is enabled and running.

    - In the Tableau Services Manager web interface, click the **Status** tab to see the status. If Tableau Prep Conductor is enabled and running, you should see Tableau Prep Conductor in the list of processes as **Active**. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor in the list of processes, but with no status information.

      Tableau Prep Conductor not enabled:

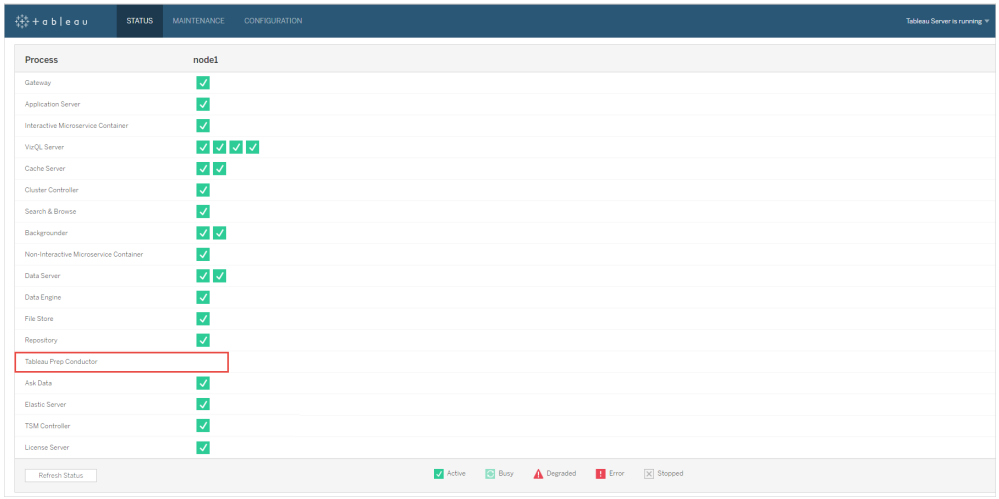
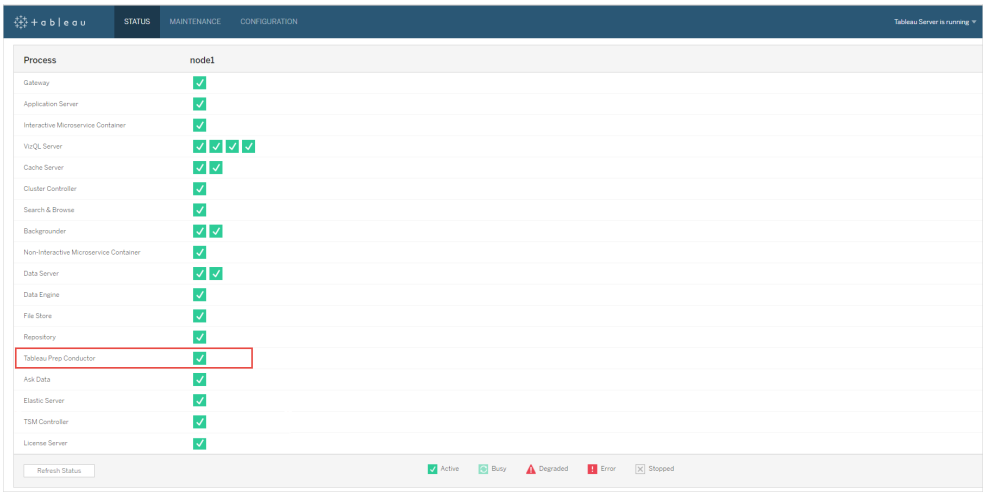


Tableau Prep Conductor enabled and running:



4. Add a second node to your Tableau Server installation. The installer will enable certain required processes like the Cluster Controller. Enable Backgrounder process on it as it is required to run scheduled flow tasks. When you enable the Backgrounder process, the installer automatically enables a single instance of Data Engine and Tableau Prep Conductor on the node. Do not add any other processes on this node.

5. Run the following commands to dedicate this node to do only flow tasks. For more information on node roles, see Node Roles in Tableau Server.

    - Get the nodeID for your dedicated node to see the list of services on each node:

      ```
      tsm topology list-nodes -v.
      ```

    - Set the node role for the dedicated node using the nodeID that you got from running the command described above:

      ```
      tsm topology set-node-role -n <nodeID> -r flows.
      ```

    - Apply the changes, and restart the server:

      ```
      tsm pending-changes apply.
      ```

- Review the status to ensure that all the processes are up and running and configured correctly:

  ```
  tsm status -v.
  ```

You have successfully added Tableau Prep Conductor to your Tableau Server installation.

Tableau Server multi-node installations

1. Run upgrade on your current Tableau Server Installation using the information in the topics below:

   - Windows
   - Linux

     When you get to the **Activate** step, use the Tableau Server product keys to activate Tableau Server.

     All product keys are available through the Customer Portal.

2. After completing the installation, add the Data Management product key to enable Tableau Prep Conductor. Tableau Prep Conductor is automatically enabled on the nodes where you already have the Backgrounder process enabled. The Data Management product key, like your other server keys, are available through the Customer Portal.

   - In the Tableau Services Manager web interface, click **Licensing** on the **Configuration** tab and click **Activate License**.
   - Enter or paste your new product key and click **Activate**.
   - On the **Register** page, enter your information into the fields and click **Register**.

3. You will be prompted to restart the server. Restart the server and verify that Tableau Prep Conductor is enabled and is running.

- In the Tableau Services Manager web interface, click the **Status** tab to see the status of all the processes. If Tableau Prep Conductor is enabled and running, you should see Tableau Prep Conductor in the list of processes as **Active**. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor in the list of processes, but with no status information.

Tableau Prep Conductor not enabled:

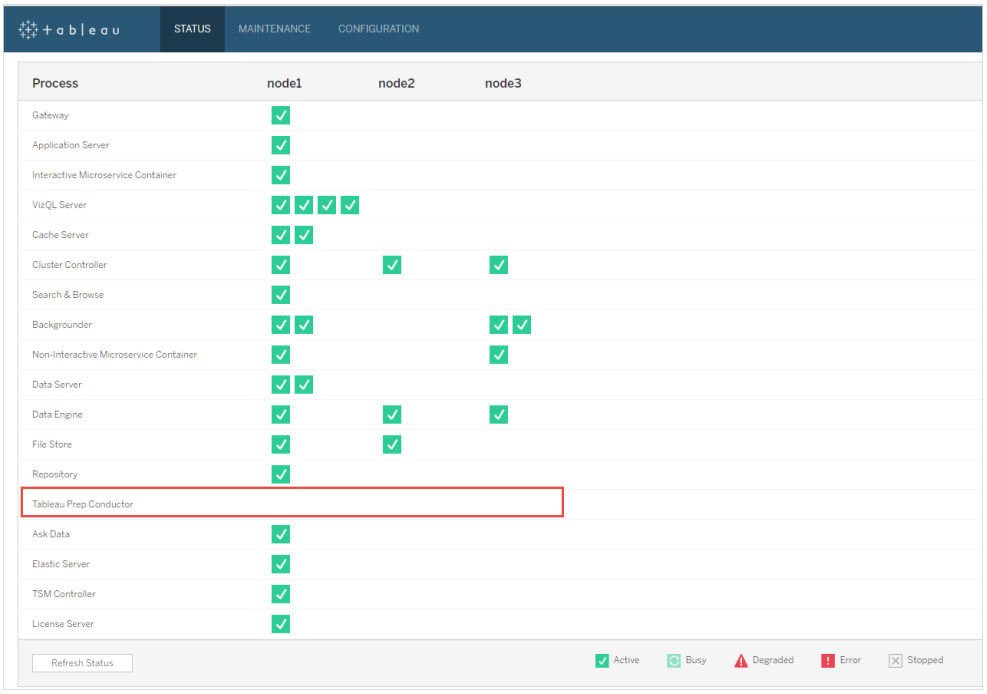
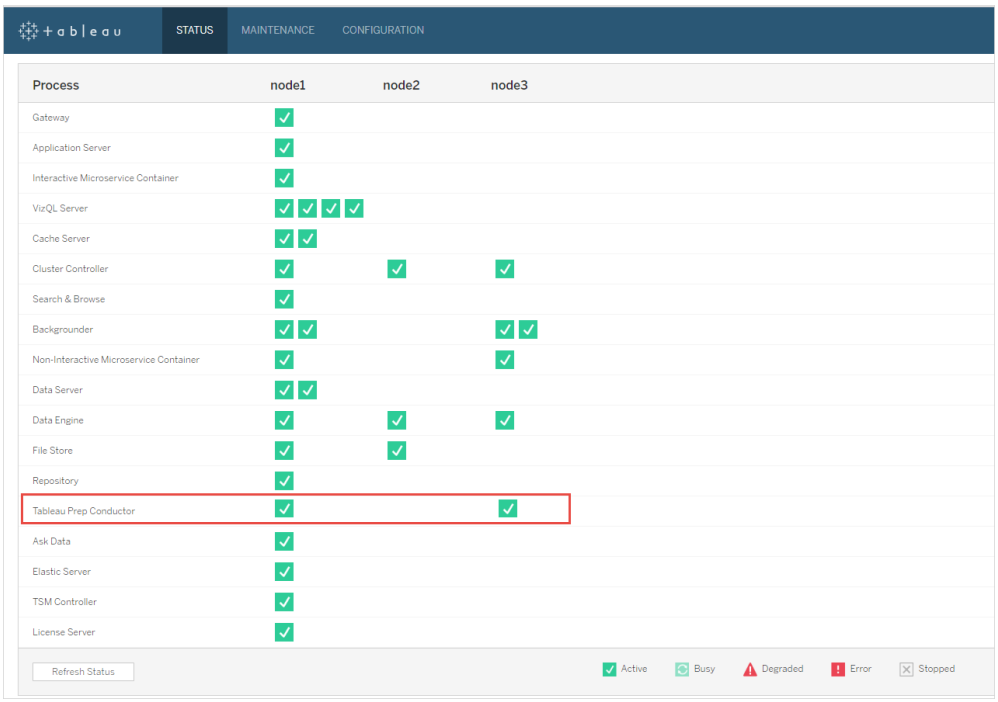


Tableau Prep Conductor enabled and running:



4. Add a new node to your Tableau Server installation. The installer will enable certain required processes like the Cluster Controller. Enable Backgrounder process on it as it is required to run scheduled flow tasks. When you enable the Backgrounder process, the installer automatically enables a single instance of Data Engine and Tableau Prep Conductor on the node. Do not add any other processes on this node.

   Note: The dedicated node counts towards the total count of the Coordination Service ensemble. You may need to deploy a Coordination Service on the new node depending on the total number of nodes you have in your cluster including the new dedicated node. For more information, see Deploy a Coordination Service Ensemble.

5. Run the following command to dedicate this node to only doing flow related operations. For more information on node roles, see Node Roles in Tableau Server.

- Get the nodeID for your dedicated node to see the list of services on each node:
- `tsm topology list-nodes -v`.
- Set the node role for the dedicated node using the nodeID that you got from running the command described above:
- `tsm topology set-node-role -n <nodeID> -r flows`.
- Apply the changes and restart the server:
- `tsm pending-changes apply`.
- Review the status to ensure that all the processes are up and running and configured correctly:
- `tsm status -v`.

6. At this stage, you may have Tableau Prep Conductor enabled on other nodes. By default, the Backgrounder process on a node performs all tasks of all types including flow tasks. To isolate Tableau Prep Conductor and flow tasks to only certain nodes, you can configure the Backgrounders to do one of the following:

    - To run only flow tasks: `tsm topology set-node-role -n <nodeID> -r flows`.
    - To run all other tasks except flows: `tsm topology set-node-role -n <nodeID> -r no-flows`.

You have successfully added Tableau Prep Conductor to your Tableau Server installation.

Tableau Server Installations using Core-Based licenses

The recommended topology for a production Tableau Server installation is a dedicated node for running flows. For more information, see Minimum Hardware Requirements and Recommendations for Tableau Server.

Data Management for Core-Based licenses includes product keys that enable Tableau Prep Conductor for your Tableau Server, and Tableau Prep Conductor cores that comes in units of four. The Tableau Prep Conductor cores should be applied to the node dedicated to running the flows. These product keys, like your other server keys, are available through the Customer Portal.

To learn more about Tableau Prep Conductor licensing, see Licensing Tableau Prep Conductor for Tableau Server.

Tableau Server single-node installations

If you currently have a single node Tableau Server installation, it is recommended that you add a second node and dedicate it to running flows.

1. Run upgrade on your current Tableau Server Installation using the information in the topics below:
    - Windows
    - Linux
2. Activate the product keys. This will enable Tableau Prep Conductor on the nodes where you already have the Backgrounder process enabled. When you are using core-based licensing, you must apply both the Data Management product key and the Resource Core product key to your Tableau Deployment. The first key allows flows to be run on Tableau Server though the Tableau Prep Conductor and the second key adds the additional cores for the Tableau Prep Conductor nodes. All product keys are available through the Customer Portal.
    - In the Tableau Services Manager web interface, click **Licensing** on the **Configuration** tab and click **Activate License**.
    - Enter or paste your new product key and click **Activate**.
    - On the **Register** page, enter your information into the fields and click **Register**.
3. You will be prompted to restart the server. Restart the server and verify that Tableau Prep Conductor is enabled and is running.
    - In the Tableau Services Manager web interface, click the **Status** tab to see the status. If Tableau Prep Conductor is enabled and running, you should see Tableau Prep Conductor in the list of processes as **Active**. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor in the list of processes, but with no status information.

Tableau Prep Conductor not enabled:

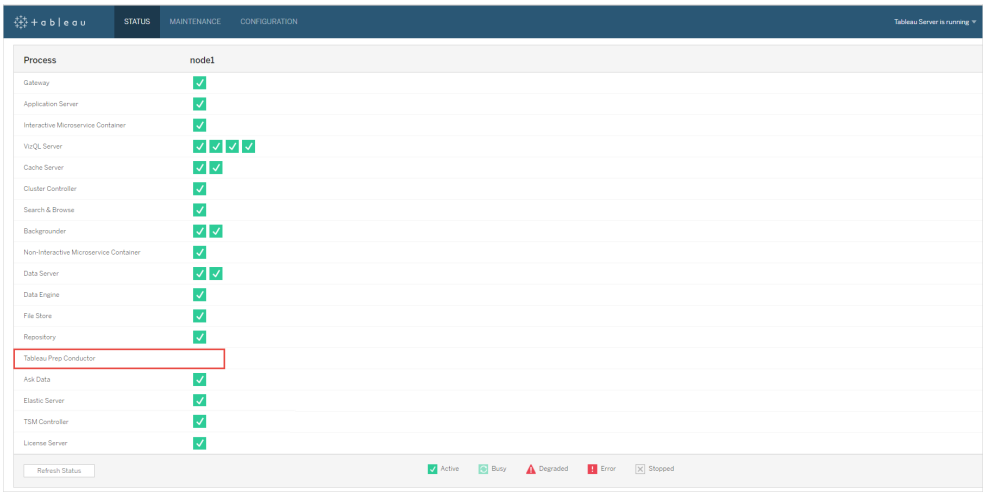
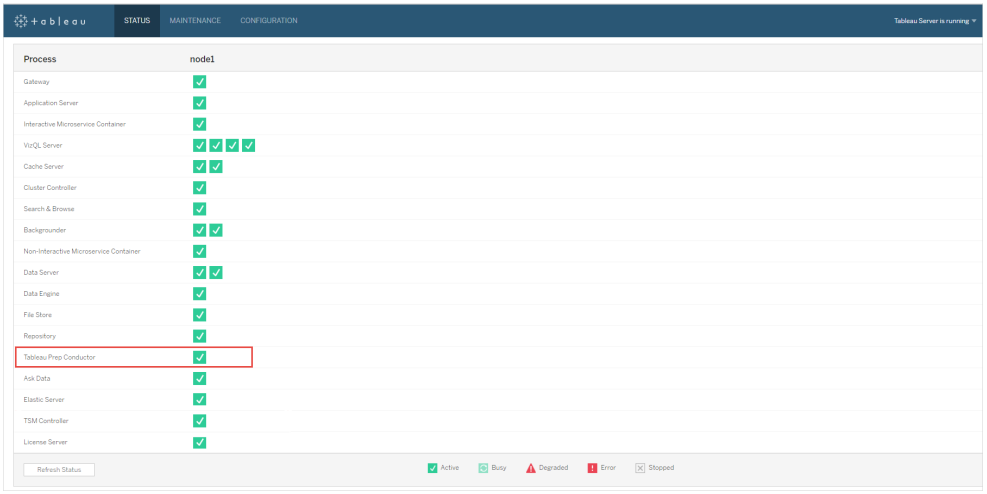


Tableau Prep Conductor enabled and running:



4. Add a second node to your Tableau Server installation. The installer will enable certain required processes like the Cluster Controller. Enable Backgrounder process on it as it is required to run scheduled flow tasks. When you enable the Backgrounder process, the installer automatically enables a single instance of Data Engine and Tableau Prep Conductor on the node. Do not add any other processes on this node.

Important: The number of physical cores on this machine must be equal to, or less than the Tableau Prep Conductor cores you purchased. For example, if you purchased four Tableau Prep Conductor cores, your node can only have up to four physical cores. To understand about how Tableau Prep Conductor licensing works, see Licensing Tableau Prep Conductor for Tableau Server.

5. Run the following commands to dedicate this node to only doing flow tasks. For more information on node roles, see Node Roles in Tableau Server.

    - Get the nodeID for your dedicated node to see the list of services on each node:

      `tsm topology list-nodes -v`.

    - Set the node role for the dedicated node using the nodeID that you got from running the command described above:

      `tsm topology set-node-role -n <nodeID> -r flows`.

    - Apply the changes and restart the server: `tsm pending-changes apply`.

    - Review the status to ensure that all the processes are up and running and configured correctly:

      `tsm status -v`.

You have successfully added Tableau Prep Conductor to your Tableau Server installation.

Tableau Server multi-node installations

1. Run upgrade on your current Tableau Server installation using the information in the topics below:

    - Windows
    - Linux

2. Activate the product keys. This will enable Tableau Prep Conductor on the nodes where you already have the Backgrounder process enabled. When you are using core-based licensing, you must apply both the Data Management product key and the Resource Core product key to your Tableau Deployment. The first key allows flows to be run on Tableau Server though the Tableau Prep Conductor and the second key adds the additional cores for the Tableau Prep Conductor nodes. All product keys are available through the Customer Portal.

    - In the Tableau Services Manager web interface, click **Licensing** on the **Configuration** tab and click **Activate License**.
    - Enter or paste your new product key and click **Activate**.
    - On the **Register** page, enter your information into the fields and click **Register**.

3. You will be prompted to restart the server. Restart the server and verify that Tableau Prep Conductor is enabled and is running.

    - In the Tableau Services Manager web interface, click the **Status** tab to see the status. If Tableau Prep Conductor is enabled and running, you should see Tableau Prep Conductor in the list of processes as **Active**. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor in the list of processes, but with no status information.

      Tableau Prep Conductor not enabled:

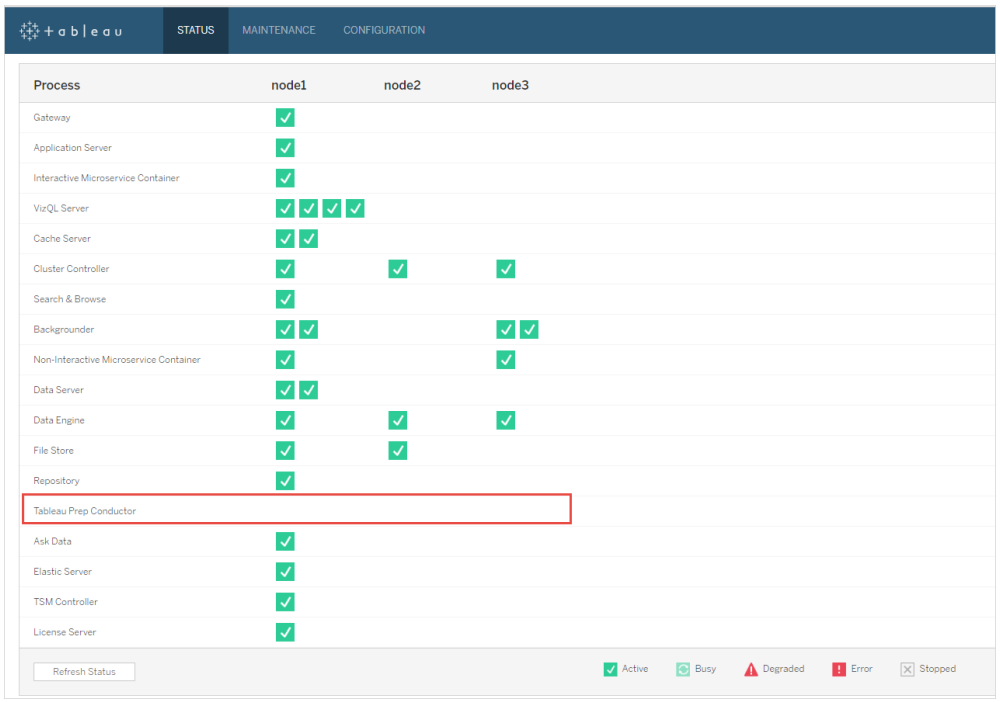
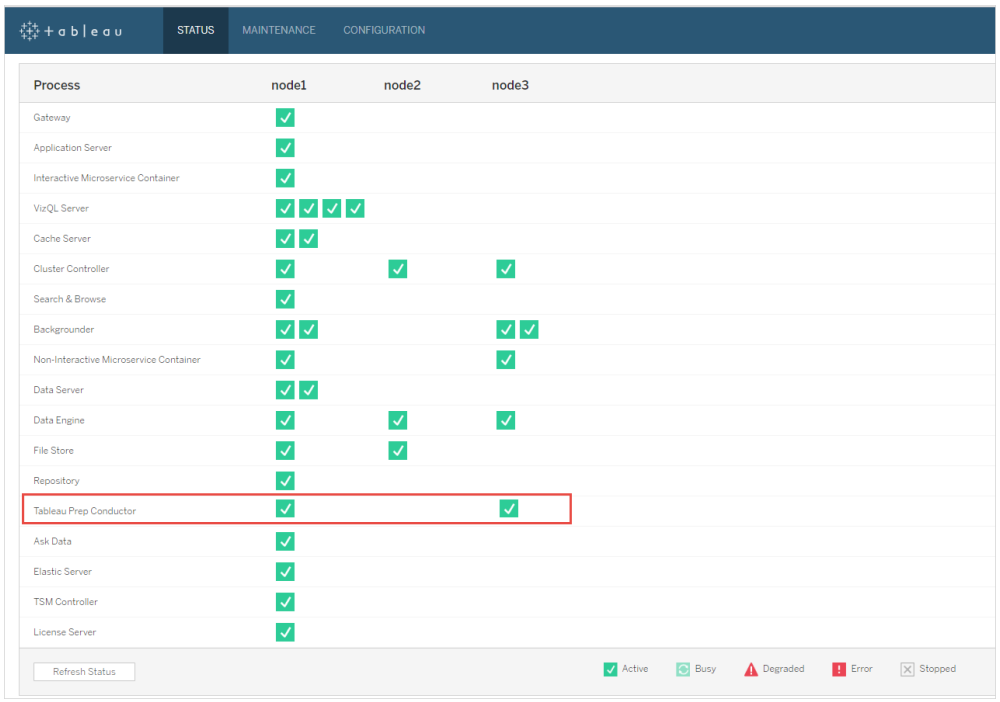


Tableau Prep Conductor enabled and running:



4. Add a new node to your Tableau Server installation. A dedicated node to run flow related operations is recommended for production Tableau Server installations. The installer will enable certain required processes like the Cluster Controller. Enable Backgrounder process on it as it is required to run scheduled flow tasks. When you enable the Backgrounder process, the installer automatically enables a single instance of Data Engine on the node. Do not add any other processes on this node.

   Note: The dedicated node counts towards the total count of the Coordination Service ensemble. You may need to deploy a Coordination Service on the new node depending on the total number of nodes you have in your cluster including the new dedicated node. For more information, see Deploy a Coordination Service Ensemble.

   Important:
   The number of physical cores on this machine must be equal to, or less than the Tableau Prep Conductor cores you purchased. For example, if you purchased four Tableau Prep Conductor cores, your node can only have up to four physical cores. To understand about how Tableau Prep Conductor licensing works, see Licensing Tableau Prep Conductor for Tableau Server.

5. Run the following commands to dedicate this node to only doing flow tasks. This will enable Tableau Prep Conductor on your new node. For more information, see Node Roles in Tableau Server.

   - Get the nodeID for your dedicated node to see the list of services on each node:

     ```
     tsm topology list-nodes -v.
     ```

   - Set the node role for the dedicated node using the nodeID that you got from running the command described above:

```
tsm topology set-node-role -n nodeID -r flows.
```

- Apply the changes and restart the server:

  ```
  tsm pending-changes apply.
  ```

- Review the status to ensure that all the processes are up and running and configured correctly:

  ```
  tsm status -v.
  ```

6. At this stage, you may have Tableau Prep Conductor enabled on other nodes that have the Backgrounder process. By default, the Backgrounder process on a node performs all tasks of all types including flow tasks. To isolate Tableau Prep Conductor and flow operations to only certain nodes, you can configure the backgrounders to do one of the following:

   - To run only flow tasks:

     ```
     tsm topology set-node-role -n <nodeID> -r flows.
     ```

   - To run all other tasks except flows:

     ```
     tsm topology set-node-role -n <nodeID> -r no-flows.
     ```

Next step

Step 2: Configure Flow Settings for your Tableau Server.

Who can do this

Tableau Server Administrators can install or upgrade Tableau Server, and enable Tableau Prep Conductor on Tableau Server.

Step 2: Configure Flow Settings for your Tableau Server

This topic describes the various flow settings that you can configure for your Tableau Server. For more information about the different settings needed to enable web authoring for flows,

see Create and Interact with Flows on the Web.

Publishing, Scheduling, and Credential Settings

When you activate Tableau Prep Conductor using the Data Management product key, Tableau Prep Conductor is enabled for the entire Tableau Server installation. You can further modify and customize the setting for sites.

Use the following instructions to configure settings related to flows for all your sites or for individual sites:

Use the following instructions to sign in to Tableau Server Admin pages:

- Windows: Tableau Server Admin Pages.
- Linux: Tableau Server Admin Pages.

Configure whether publishing and scheduling flow should be allowed for a site:

1. **Enable users to publish and schedule flows:** This setting is enabled by default when you enable Tableau Prep Conductor. If you have multiple sites, you can selectively turn off Tableau Prep for Server for individual sites. If you disable this setting for a site that once allowed flows, see Implication of disabling Tableau Prep Conductor for more information.

   On the **General** page under **Settings**, scroll to the **Tableau Prep Conductor** section and clear the **Allow users to schedule and monitor flows** check box.

2. **Enable users to link flow runs together using Linked Tasks (version 2021.3 and later):** Enable users to schedule flow tasks to run one after the other. Starting in version 2022.1, this option is enabled by default. In prior versions, administrators needed to enable this feature first.

   If you have multiple sites, you can selectively turn off **Linked Tasks** for individual sites, but the option must first be enabled at the **Server Settings** level.

If the setting is turned off after linked tasks are scheduled, any tasks that are running will complete and the scheduled linked tasks are hidden and no longer show on the **Scheduled Tasks** tab.

3. **Embed Credentials**

   -Allow publishers to embed credentials in a data source, flow or workbook: This setting allows publishers to attach passwords to published flows that will automatically authenticate web users.

   -Allow publishers to schedule flow runs and data extract refreshes: This option is only available if setting above is enabled. When this setting is enabled, publishers will see scheduling options in the Publish dialog box.

Implication of disabling Tableau Prep Conductor

If you disable Tableau Prep Conductor after using it for a while, you will not be able to see the flows, schedules, tasks, and other things related to flows. The following table gives you more information on what you can and cannot see when you disable Tableau Prep for Server completely or only for specific sites:

	Prep not enabled at Server level	**Prep enabled at Server level, but disabled for a site**	**Prep enabled for both Server and site**
Show flows	Yes	Yes	Yes
Show tasks/schedules in Server view	No	Yes	Yes
Show tasks/schedules in Site view	No	No	Yes
Show Site setting (only for Server Admins)	Yes (disabled)	Yes	Yes

Show TSM status	Yes (Tableau Prep Conductor is not shown)	Yes	Yes
Show TSM settings	Yes (disabled)	Yes	Yes

Important: Scheduled tasks will continue to run even when Tableau Prep Conductor is disabled for that site, but will fail.

Configure notifications for flow failures

You can configure Tableau Server to send email notifications for flow run failures. The notifications are sent for failures that occur when running the flows through either a scheduled task, linked task, or a manual run using the **Run now** menu option. You must first enable the server-wide setting, and then configure at the site level.

To enable the server-wide email notification

You can either use the Tableau Services Manager (TSM) web interface or TSM CLI as described below:

Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850.

2. Click **Notifications** on the **Configuration** tab and click **Email Server**.

3. Enter the email server information.

4. Click the **Events** tab.

5. Under **Content Updates**, select **Send emails when flow runs, encryption jobs, or scheduled refreshes fail** if not already turned on by default.
6. Click **Save Pending Changes** after you've entered your configuration information.
7. Click **Apply Changes and Restart**.

Use the TSM CLI

The notification values can be set individually with the tsm configuration set command:

Windows: tsm configuration.

Linux: tsm configuration.

Set notification values

Use the tsm configuration set command with the following syntax to

to enable flow failure notifications, run the following command:

```
tsm configuration set -k backgrounder.notifications_enabled -v true
```

> **Note:** This will enable email notification for both extract refresh failures and flow failures.

After you are done setting values, you must run the following command:

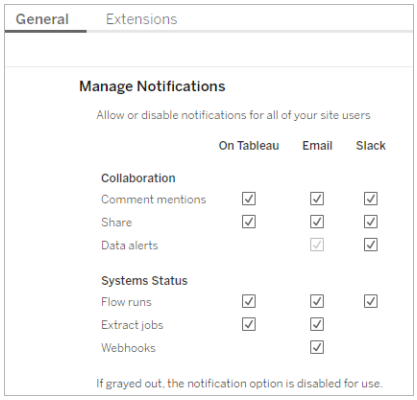
```
tsm pending-changes apply
```

The pending-changes apply command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in the case there is no restart. You can suppress the prompt using the --ignore-prompt option, but this does not change the restart behavior.

To configure notification for a site:

On the **General** page under **Settings**, scroll to the **Manage Notifications** setting and select the notification types that you want site users to receive.

You can receive notifications as an email, on the Tableau site, or in your Slack workspace if your administrator has connected your site to Slack. For more information, see Site Settings Reference.



Note: Recreate your notifications settings when upgrading from 2020.4 or earlier to 2021.1 and later. Older notifications settings are not automatically moved to the Manage Notifications setting.

Next step

Step 3: Create Schedules for Flow Tasks

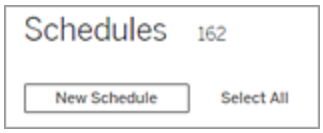
Who can do this

Tableau Server Administrators can configure server and site level settings. Tableau Site Administrators can configure site level settings.

Step 3: Create Schedules for Flow Tasks

Create a new schedule:

1. On the **Schedules** tab, click **New Schedule**.

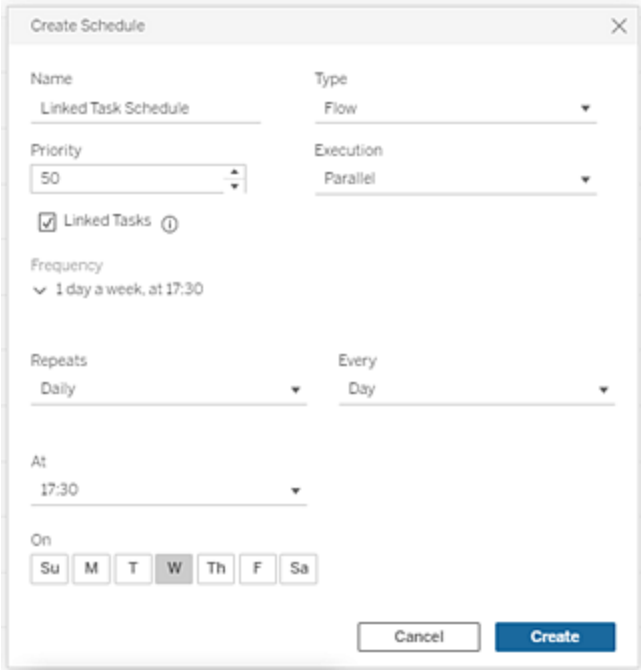


2. Enter the following information in the **New Schedule** dialog box and click **Create**.

   - **Name:** Enter a descriptive name for the schedule. Typically, this includes the description of the schedule frequency.
   - **Type:** Select Flow as the task type.
   - **Priority:** You can define a default priority from 1 to 100, where 1 is the highest priority. This value will be assigned to the tasks by default. If two tasks are pending in the queue, the one with the higher priority runs first.
   - **Execution:** Choose whether a schedule will run in parallel or serially. Schedules that run in parallel run on all available backgrounder processes so that they can complete faster.
   - **Linked Tasks (version 2021.3 and later):** Select the check box if the schedule can be used to schedule flows to run one after the other. Linked tasks require a **Parallel** execution method.

     Starting in version 2022.1, this option is enabled by default. In prior versions, the Server Administrator must first enable linked tasks for the server before you can configure schedules to support linked tasks. For more information, see Step 2: Configure Flow Settings for your Tableau Server

- **Frequency:** You can define an hourly, daily, weekly, or monthly schedule.



Next step

Step 4: Safe list Input and Output locations

Who can do this

Tableau Server Administrators can create and modify schedules. Schedules are created at the server level and apply across all the sites on a server.

Step 4: Safe list Input and Output locations

This topic describes the rules that apply to this feature and how to safe list the directories on your network.

Flow input and output connections may need to connect to databases or files in the directories on your network. You must safe list the directories you want to allow access to. Input and Output connections will only be allowed to connect to data in the safe listed locations. By default, no connections are allowed.

> **Note**: You can still publish the flows and any data that is embedded in the flow file (tflx) to Tableau Server, but the flow will fail to run if the directories aren't included in your organization's safe list.

How to safe list input and output locations

The following rules apply and must be considered when configuring this setting:

- The directory paths should be accessible by Tableau Server. These paths are verified during server startup and at flow run time and are **not** verified at the time of publishing the flow to Tableau Server.
- Network directory paths have to be absolute and cannot contain wildcards or other path traversing symbols. For example, `\\myhost\myShare\*` **or** `\\myhost\myShare*` are invalid paths and would result in all the paths as disallowed. The correct way to safelist any folder under *myShare* would be `\\myhost\myShare` **or** `\\myhost\\myShare\`.

  > **Note:** The `\\myhost\myShare` configuration will not allow `\\myhost\myShare1`. In order to safe list both of these folders safe list them as \\myhost\myShare; \\myhost\myShare1.

- **Windows:**
  - The value can be either `*`, (for example, `tsm configuration set -k maestro.input.allowed_paths -v "*"`) to allow any network directory, or a specified list of network directory paths, delimited by a semicolon (**;**). If you specify a list of directory paths, be sure to specify particular directories rather than the root of the file share.

- If the path contains spaces or special characters you will have to either use single or double quotes. Whether you use single or double quotes depends on the shell that you are using.
- No local directory paths are allowed even when the value is set to `*`.
- To save flow output to a network share, you must first configure a Run As user service account on Tableau Server. You cannot save flows to a network share using the default system account. Then configure the target directory on the network share for Full Control permissions for the Run As user account you created.

  Depending on how your organization manages nested folder permissions, you may need to grant additional permissions in the folder hierarchy, with a minimum of Read, Write, Execute, Delete, and List Folder permission, to allow the Run As user account access to the target folder.

- **Linux:**
  - The value can be either `*`, (for example, `tsm configuration set -k maestro.input.allowed_paths -v "*"`) meaning that any path, including local (with the exception of some system paths configured using “native_api.internal_disallowed_paths”), or a list of paths, delimited by a semicolon (;).
  - You must be using a kernel version equal to or later than 4.7. Safe listing to or from a network share is not supported on kernel versions earlier than 4.7. On earlier versions, when the output is written to a network share, hyper fails to output files, resulting in flows failing at runtime. When reading input files from a network share on earlier versions, flow executions fail. To check the kernel version, in the Linux terminal, type the command `uname -r`. This will display the full version of the kernel you are running on the Linux machine. Note that for Red Hat Enterprise Linux, kernel version 4.7 and later is only available with Red Hat Enterprise Linux version 8.

- To save flow output to a network share, the local Linux account that has access to Tableau Server resources must be given Full Control permissions to the target directory on the network share. If a path is both on the flows allowed list and internal_disallowed list, internal_disallowed takes precedence.
  The mount points for both input and output paths used by flows must to be configured using the `native_api.unc_mountpoints` configuration key. For example:
  `tsm configuration set -k native_api.unc_mountpoints -v 'mountpoints'`
  For information about configuring this, see this Tableau Knowledge Base article: Tableau Server on Linux - How to Connect to a Windows Shared Directory.

Use the following commands to create a list of allowed network directory paths:

For input connections:

```
tsm configuration set -k maestro.input.allowed_paths -v your_net-
workdirectory_path_1;your_networkdirectory_path_2

tsm pending-changes apply
```

For output connections:

```
tsm configuration set -k maestro.output.allowed_paths -v your_net-
workdirectory_path_1;your_networkdirectory_path_2

tsm pending-changes apply
```

Important:
These commands overwrite existing information and replace it with the new information you provided. If you want to add a new location to an existing list, you must provide a list of all the locations, existing, and the new one you want to add. Use the following commands to see the current list of input and output locations:

```
tsm configuration get -k maestro.input.allowed_paths
tsm configuration get -k maestro.output.allowed_paths
```

Next step

Step 5: Optional Server Configurations

Who can do this

On Windows, members of the local computer Administrators group can run *tsm* commands.

On Linux, members of the **tsmadmin** group can run *tsm*commands. The **tsmadmin** group can be configured using the *tsm.authorized.groups* setting.

Step 5: Optional Server Configurations

The options described in this topic are not required to enable flow publishing and scheduling flows on Tableau Server. They can be used to customizing your environment according to your requirements.

Set the timeout period for flows

You can set time limits for how long a flow can run to make sure that subsequent tasks are not held up due to stalled tasks. The following two *tsm* command options determine how long a flow task can run before the flow background task is canceled. These two commands together determine the total timeout value for flow tasks.

The `backgrounder.default_timeout.run_flow` sets the number of seconds before a flow run task is canceled.

For example:

```
tsm configuration set -k backgrounder.default_timeout.run_flow -v
<new value>

tsm pending-changes apply
```

(Default value: 14400 seconds or 4 hours)

The `backgrounder.extra_timeout_in_seconds` command sets the number of seconds beyond the setting in `backgrounder.querylimit` before a background job is

canceled. This setting makes sure that a stalled job does not hold up subsequent jobs. The setting applies to processes listed in `backgrounder.timeout_tasks`.

For example:

```
tsm configuration set -k backgrounder.extra_timeout_in_seconds -v
<value>
```

(Default value: 1800 seconds or 30 minutes)

Check the available resources on the server running flows. It's recommended that you have a dedicated node for Tableau Prep Conductor.

Set the threshold for suspended flow tasks

By default, a flow task is suspended after 5 consecutive flow task failures. To change the threshold number of flow task failures that can occur before they are suspended, use the following tsm configuration set command:

```
tsm configuration set -k backgrounder.flow_failure_threshold_for_
run_prevention -v <number>
```

This sets the threshold for the number of consecutive failed flow tasks necessary before suspending the tasks. This is a server-wide setting.

Who can do this

Tableau Server administrators can make changes to server configurations.

Schedule Flow Tasks

Note:Flows can be scheduled to run on Tableau Cloud or Tableau Server using Tableau Prep Conductor. Prep Conductor is licensed through Data Management on a per deployment basis. After you purchase and license Data Management, you must enable Prep Conductor.
Starting in version 2020.4.1, you no longer need the Data Management license to publish flows to the web. As a Creator, you can also create and edit flows directly on your server. For more information authoring flows on the web, see Tableau Prep on the Web.

Note: With the 2024.1 release, changes were introduced for flow schedules on Tableau Cloud. The Schedules tab on the left navigation pane has been removed and you can now create custom schedules directly from your flows Scheduled Tasks tab.

To schedule flows to run at a specific time or on a recurring basis you can create scheduled tasks. Scheduled tasks rely on pre-configured schedules. Schedules are created by the System Administrator. For information about how to create schedules on Tableau Server, see Step 3: Create Schedules for Flow Tasks in the Tableau Server help.

Starting in version 2021.3, you can link flow runs together when scheduling tasks to run flows one after the other. For more information, see Schedule linked tasks in this topic.

Starting in version 2022.1, you can also run linked tasks via the REST API using the new flow methods **Query Linked Tasks**, **Query Single Linked Tasks**, and **Run Linked Task Now**. For more information, see Flow Methods in the Tableau REST API help.

Running flows that include parameters

Starting in version 2021.4, you can include user parameters in your flows to make flows more dynamic. When the flow runs, you are prompted to enter your parameter values. When setting up flows on a schedule, you specify parameter values at that time.

You must specify the parameter values for any required parameters. For optional parameters, you can enter those values as well, or accept the current (default) value for the parameter. For

more information about running flows with parameters, see Run flows on a schedule in the Tableau Prep help.

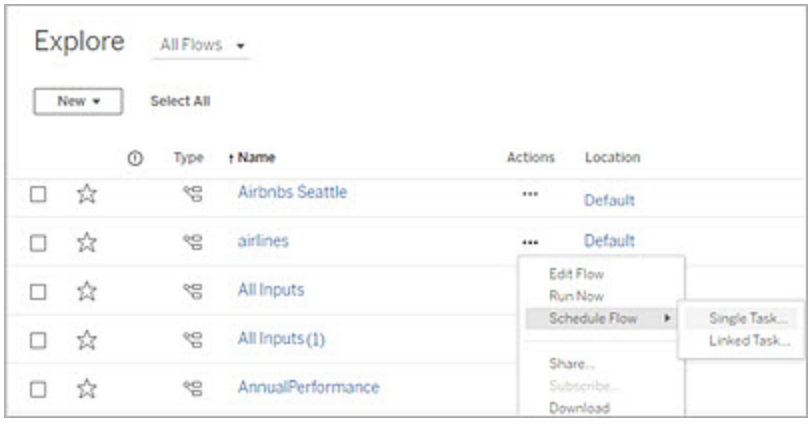
Starting in Tableau Prep Builder and Tableau Cloud version 2023.2, you can apply date or time system parameters to flow output names for file and published data source output types. The start time is automatically added to the flow output name.

> **Note**: Your administrator must enable the **Flow Parameter** server and site settings on your server before you can run flows that include parameters in Tableau Server or Tableau Cloud. For more information, see **Create and Interact with Flows on the Web** in the Tableau Server or Tableau Cloud help.

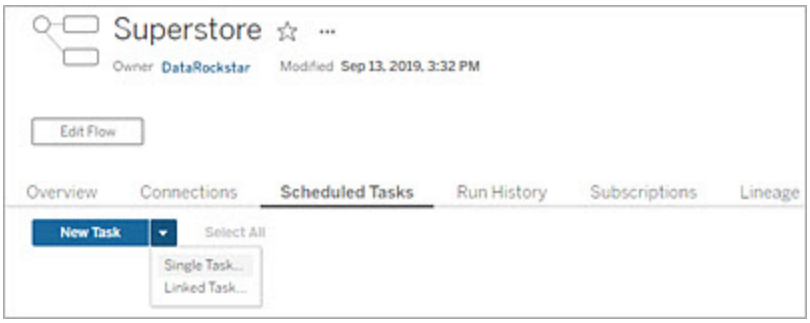
Schedule a flow task

1. Do one of the following:

   - (version 2022.1 and later) From the **Explore** page, in **List** view, in the **Actions** menu, select **Schedule Flow** > **Single Task**. If you select a flow in the list, you can also use the top **Actions** menu.

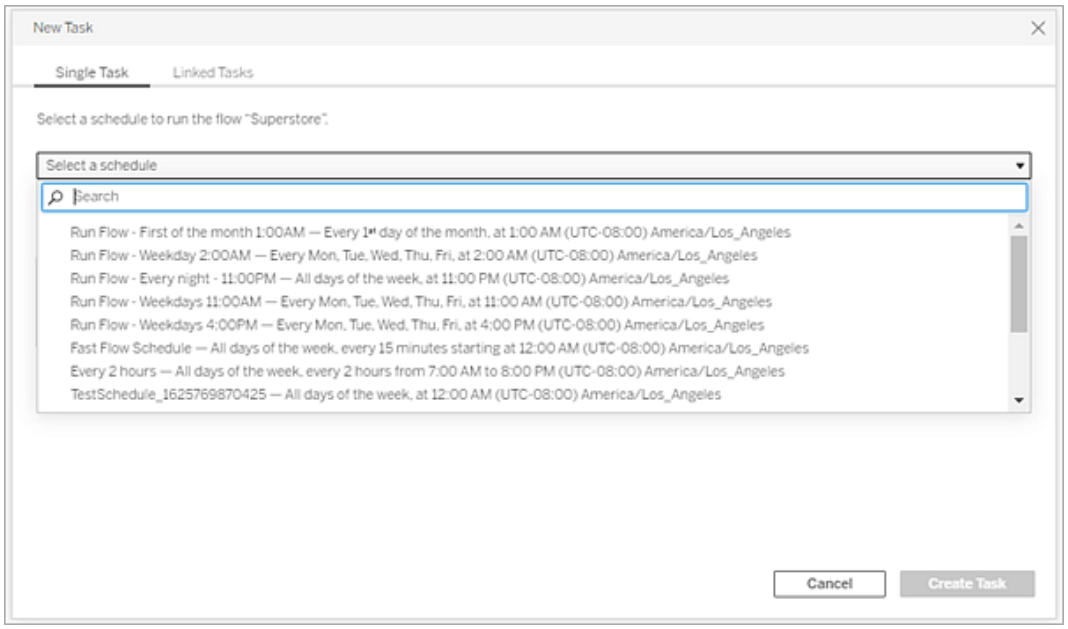


   - From the **Overview** page for the flow, **Scheduled Tasks** tab, click **New Task** or click the drop-down and select **Single Task**.



If the output step isn't assigned to a task, you can also create a new task from the **Overview** page. On that page, in the **Schedules** field, click **Create new task**.

2. In the **New Task** dialog, on the **Single Task** (**New Task** in prior releases) tab, select a schedule from the drop-down list.

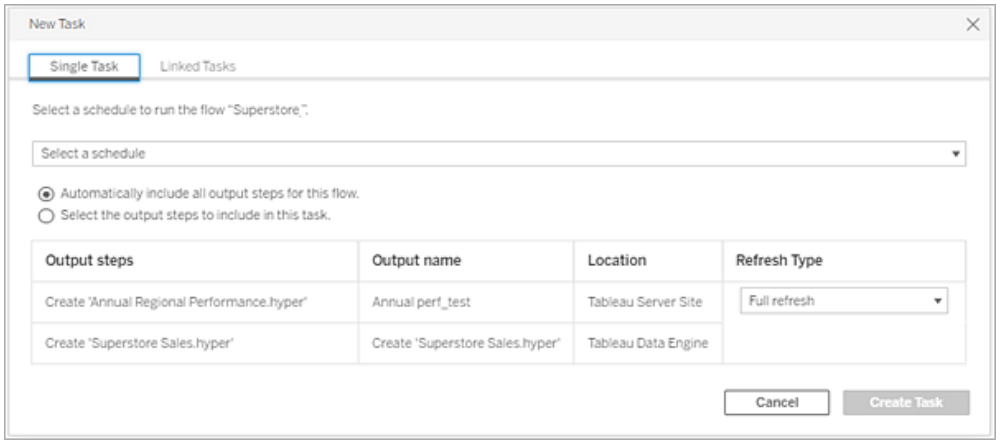


3. Select one of the following options:

- **Automatically include all output steps for this flow:**(default) Select this option to include all current and future output steps for this flow in the scheduled task. As new output steps are added to the flow over time, they are automatically included in the schedule when it runs.

- **Select the output steps to include in this task**: Select this option and manually select the output steps to include in this scheduled task.

  To include all output steps in the flow task, select the check box next to **Output Steps**. This area can't be edited if the **Automatically include all output steps for this flow** radio button is selected. Select the other radio button to enable this section.

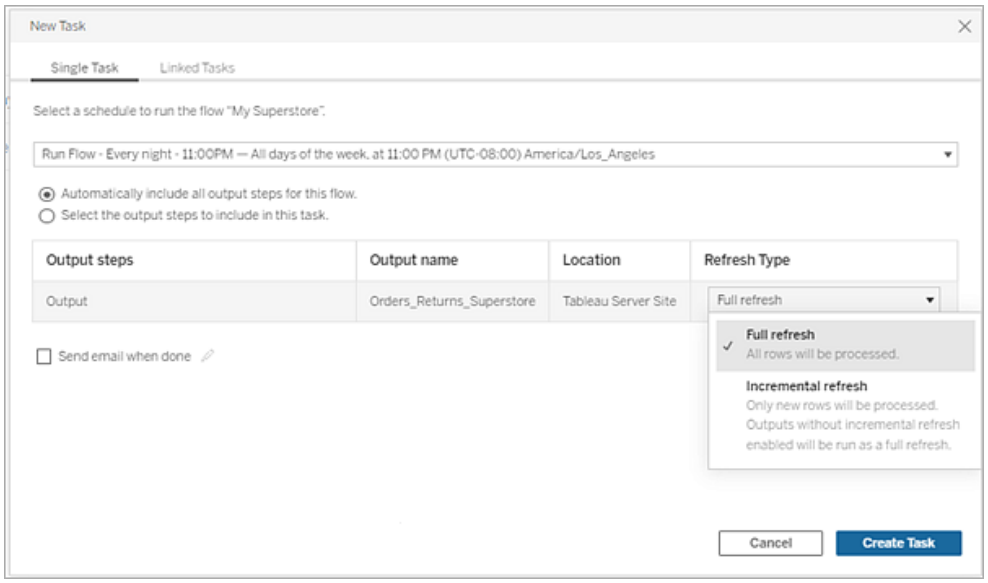


4. (version 2020.2.1 and later) Select a **Refresh Type**. For more information about these settings, see Refresh Flow Data Using Incremental Refresh.

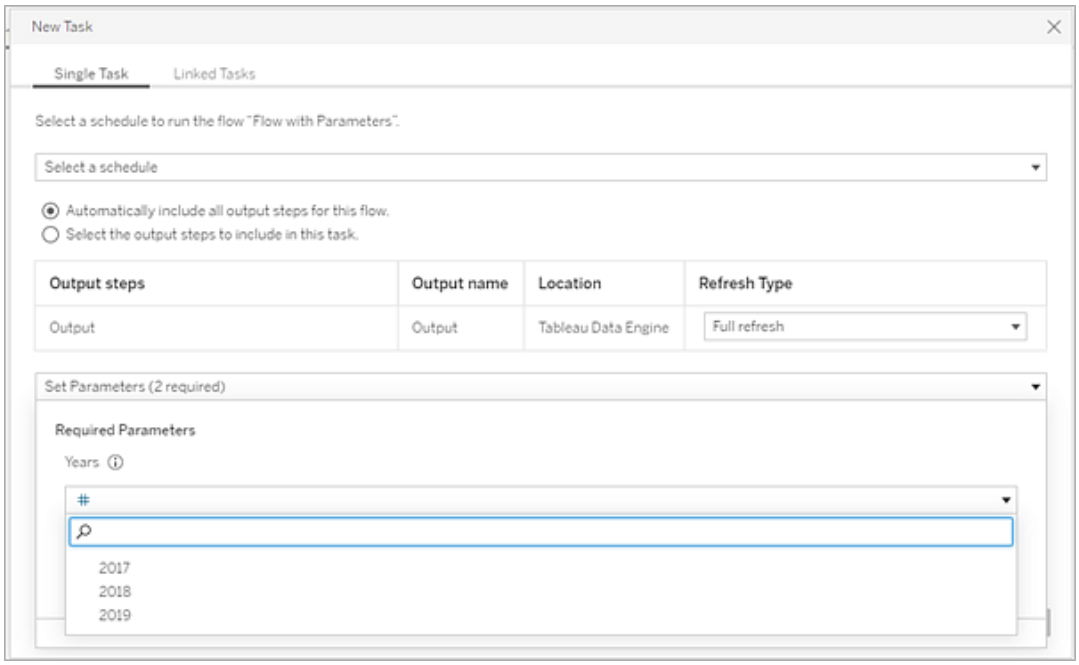
> **Note**: Starting in version 2020.2.1 and later, if one input is configured to use incremental refresh and it is associated with multiple outputs, those outputs must be run together and must use the same refresh type. Otherwise the flow will fail.

- **Full refresh** (default): Refresh all data and create or append data to your table based on the flow output setting.
- **Incremental refresh**: Refresh only the new rows and create or append data to your table based on the flow output setting. The incremental refresh option is only available when the flow is configured to use this refresh type.

> **Note**: Tableau Prep Conductor runs a full refresh for all outputs regardless of the run option you select if no existing output is found. Subsequent flow runs will use the incremental refresh process and retrieve and process only your new rows unless incremental refresh configuration data is missing or the existing output is removed.



5. (optional) If you are the flow owner, select **Send email when done** to notify users when the flow is successful. For more information about how to send email notifications on flow runs, see Notify Users of Successful Flow Runs.
6. (version 2021.4 and later) If your flows include parameters, enter any required or optional parameter values. You must enter required values for the flow to run.



7. Click **Create Task** to create the scheduled task.

Schedule linked tasks

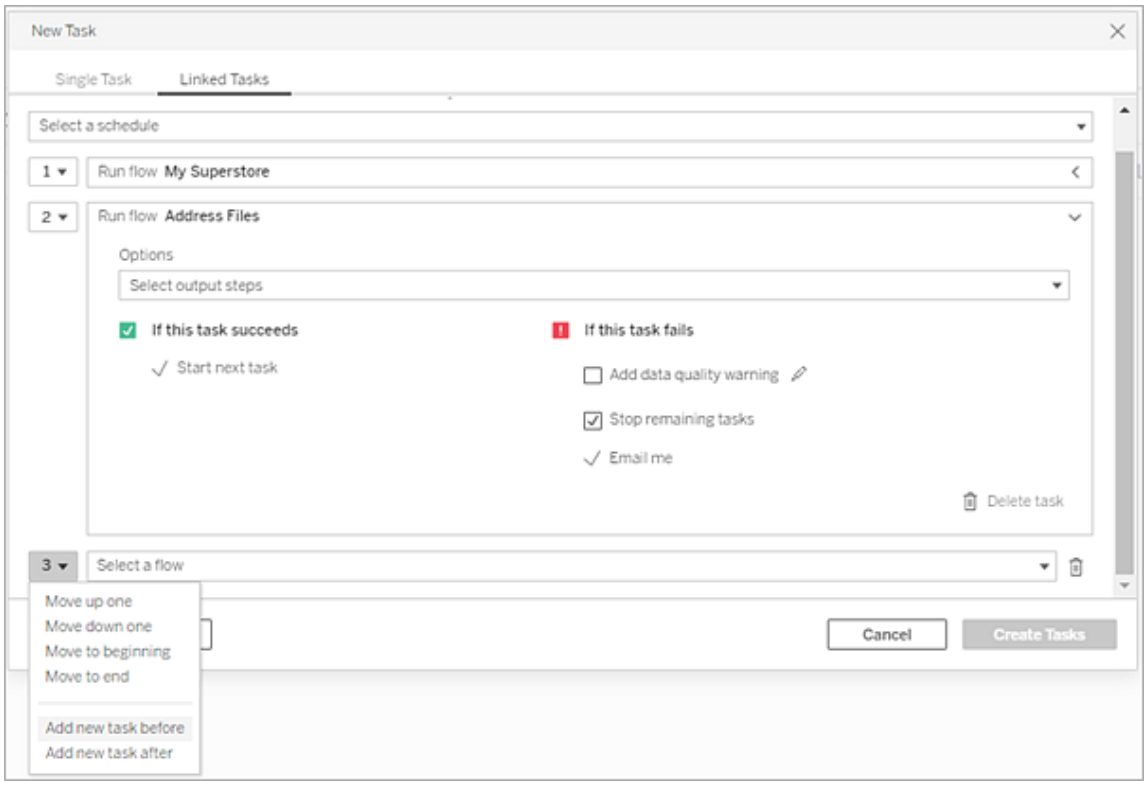
Supported in Tableau Server and Tableau Cloud version 2021.3.0 and later.

> **Note**: Starting in version 2022.1, Linked tasks functionality is enabled by default. Server and Site Administrators can turn off this functionality on the **Settings** page and on flow schedules in the **Schedules** dialog. In previous versions, Server Administrators must first enable this functionality to use and manage it. For more information, see Step 2: Configure Flow Settings for your Tableau Server and Step 3: Create Schedules for Flow Tasks.

Use the **Linked Tasks** option to schedule up to 20 flows to run sequentially, one after the other. Easily set up your flow list by selecting your schedule, then select downstream flows to run in the order you choose.

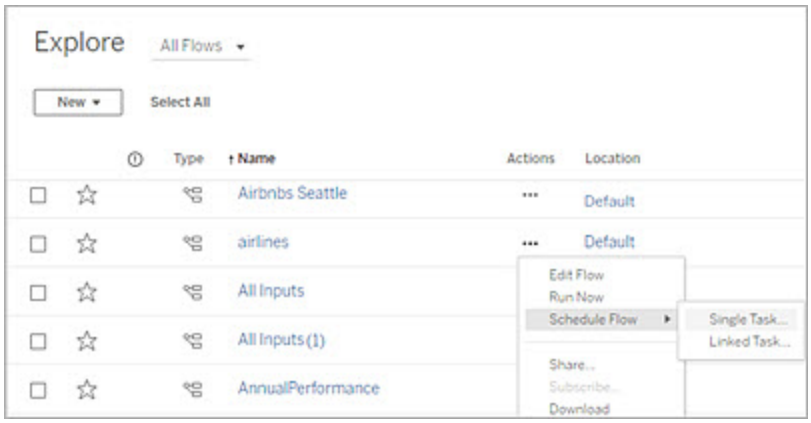
Flows run in the order specified in the list. Use the menu to move flows around in your list or add new upstream or downstream flows to the list at any time.

Select the outputs that you want to include in the flow run and configure the settings to tell Tableau what to do with remaining flows in the schedule when the previous flow run fails.

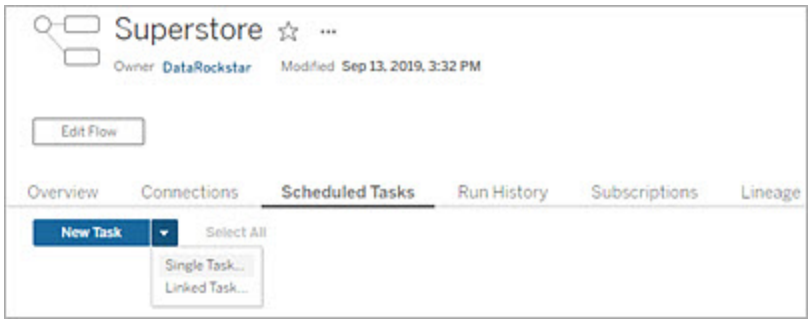


1. Do one of the following:

    - (version 2022.1 and later) From the **Explore** page, in **List** view, in the **Actions** menu, select **Schedule Flow** > **Linked Task**. If you select multiple flows in the list, you can also use the top **Actions** menu.

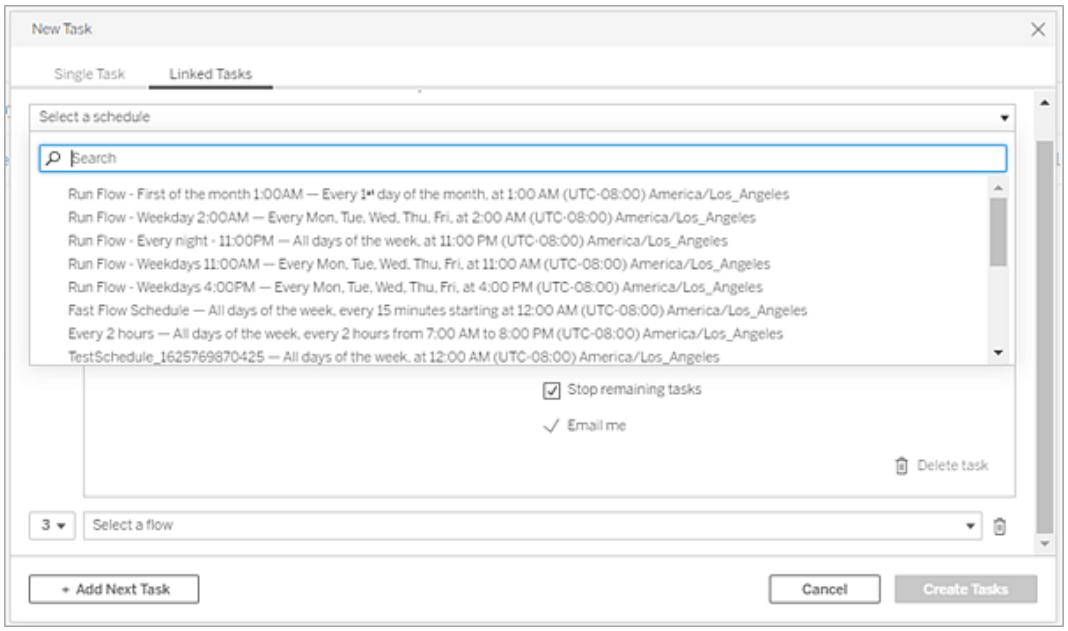


- From the **Overview** page for the flow, **Scheduled Tasks** tab, click **New Task** and select the **Linked Task** tab, or click the drop-down and select **Linked Task**.



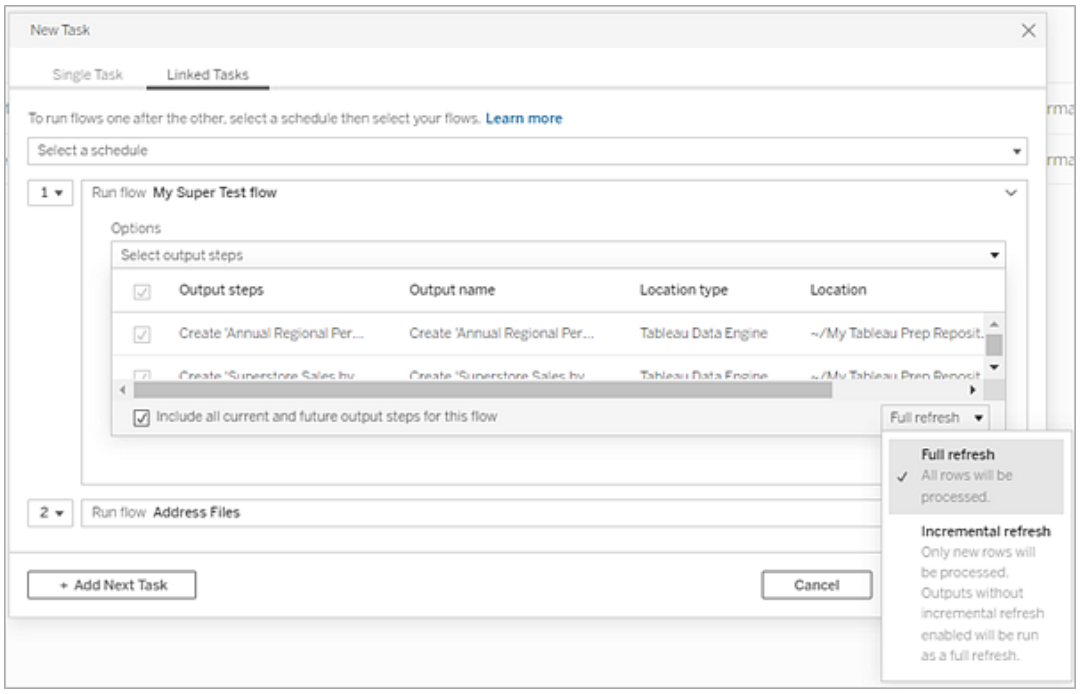
If the output step isn't assigned to a task, you can also create a new task from the **Overview** page. On that page, in the **Schedules** field, click **Create new task**.

2. In the **New Task** dialog, on the **Linked Tasks** tab, select a schedule from the drop-down list. Only schedules that are enabled for linked tasks are shown.



3. Click the **Select output steps** drop-down to select the flow outputs to run. By default, all flow outputs are included. To select specific outputs, clear the **Include all current and future output steps for this flow** check box.

   The flow where the task is initiated is automatically set as the first flow to run, but you can use the menu to change the run order after you add other flows to your list.



4. Select your refresh type from the following options:

> **Note**: If one input is configured to use incremental refresh and it is associated with multiple outputs, those outputs must be run together and must use the same refresh type. Otherwise the flow will fail.

- **Full refresh** (default): Refresh all data and create or append data to your table based on the flow output setting.
- **Incremental refresh**: Refresh only the new rows and create or append data to your table based on the flow output setting. The incremental refresh option is only available when the flow is configured to use this refresh type. For more information, see Refresh Flow Data Using Incremental Refresh.

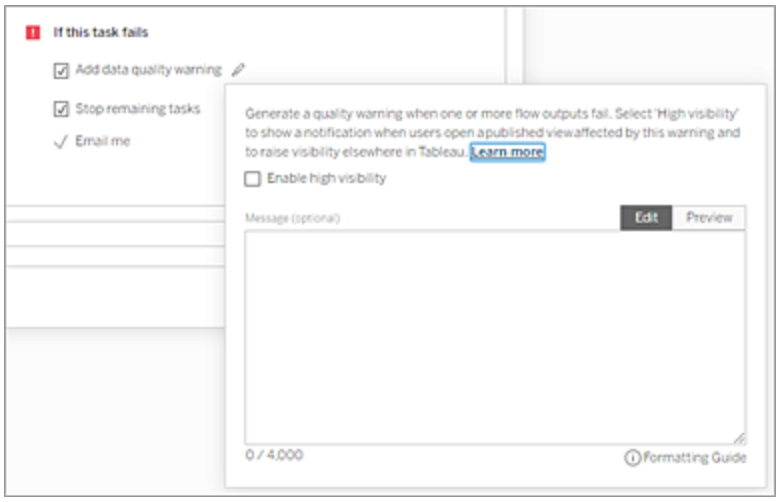
> **Note**: Tableau Prep Conductor runs a full refresh for all outputs regardless of the run option you select if no existing output is found. Subsequent flow runs will use the incremental refresh process and retrieve and process only your new rows unless incremental refresh configuration data is missing or the existing output is removed.

5. (optional) If you are the flow owner, select **Send email when done** to notify users when the flow is successful. For more information about how to send email notifications on flow runs, see Notify Users of Successful Flow Runs.

6. Set your flow failure options:

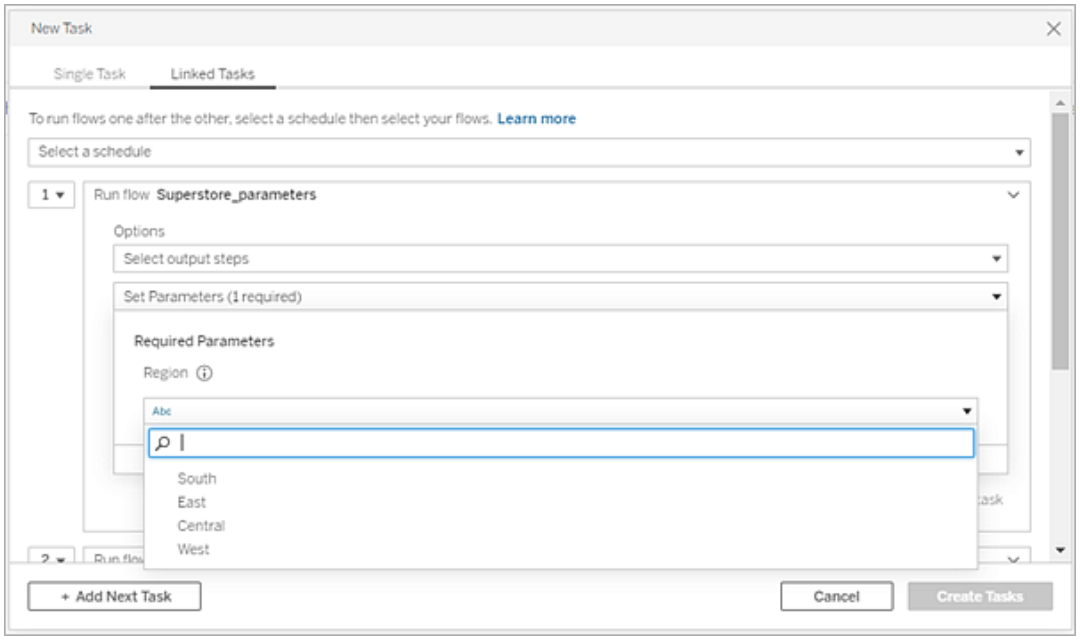
   - **Add data quality warning**: Select the check box to set a warning message on the flow so that users of the data are aware of issues. The message remains until the flow runs successfully. If the flow already has a data quality warning, this option shows selected and can't be turned off.

     > **Note**: In version 2021.4 and earlier, click the **Flow run monitoring** icon to open the **Data Quality Warning** dialog.



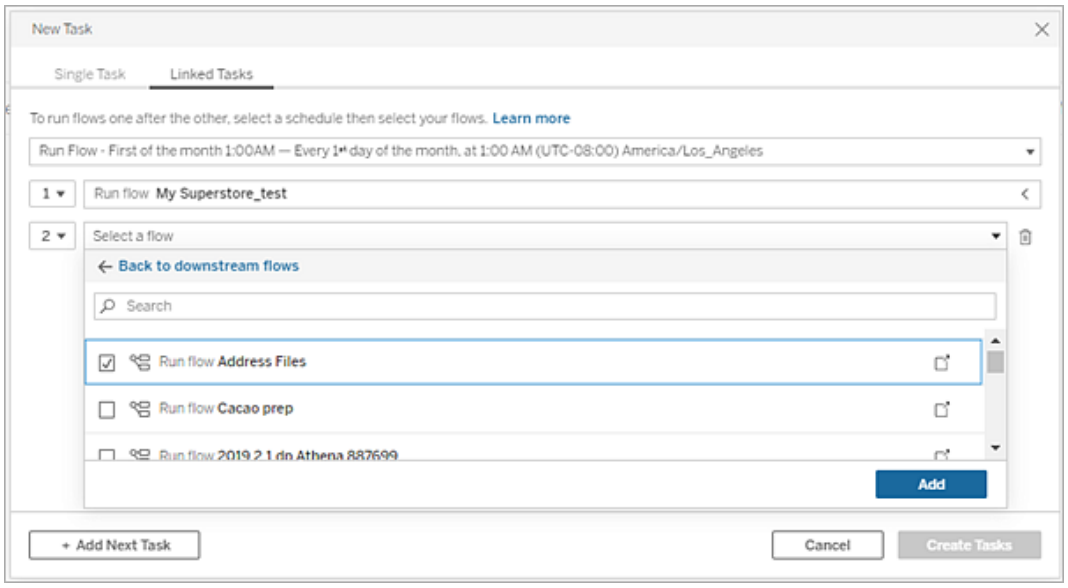
- **Stop remaining tasks**: Select this option to prevent the downstream tasks in the list from being queued to run.
- **Email me**:Email notifications are automatically sent to the flow owner and the linked task creator when the flow fails, is suspended, or is canceled.

7. (version 2021.4 and later) If your flows include parameters, enter any required or optional parameter values. You must enter required values for the flow to run.

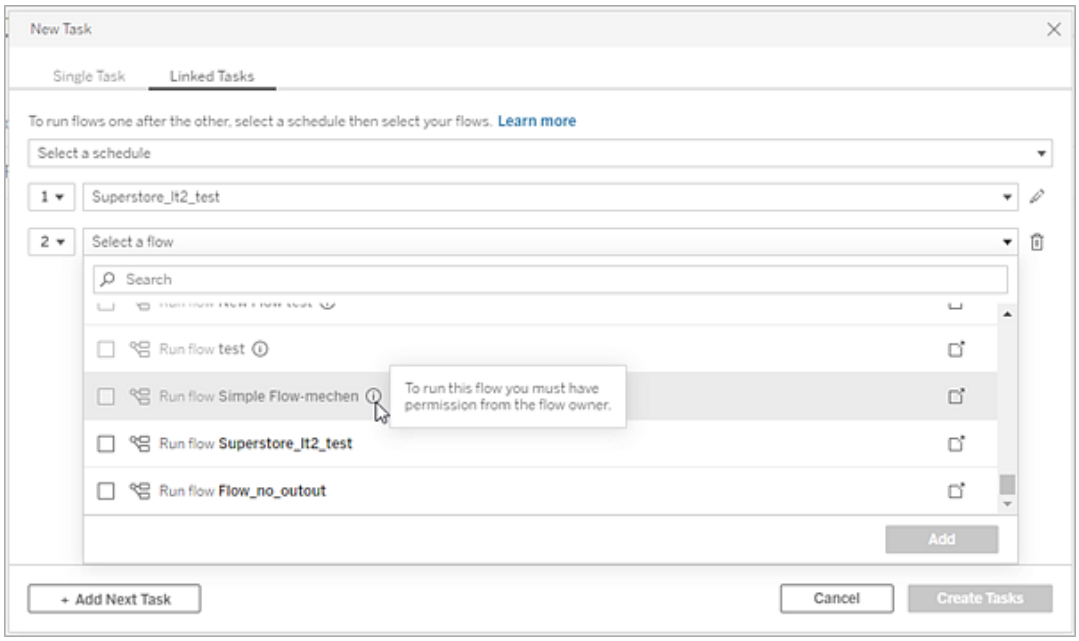


8. Click the drop-down for the 2nd task to add your next flow. Flows that use the previous flow's outputs are shown automatically or click **View all flows** to see all available flows.

   Select one or more flows and click **Add** or click **Add Next Task** to add more flow run tasks to your list.



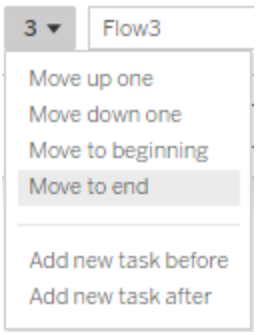
   If you don't have permission to run the flow, you must contact the flow owner to grant permissions before adding the flow to your list.



9. Repeat steps 3-5 to configure your flow run options.

> **Note**: In version 2021.4 and earlier, click the **Edit** icon next to your new flow to expand the Options pane.

10. (Optional) Click the drop-down next to any numbered task to open the menu to change the order of your flow tasks or insert new tasks between existing tasks.



11. Click **Create Tasks** to create the scheduled linked tasks.

Who can do this

- The Server administrator can do this on all sites on the server. The Site administrator can do this on sites they have access to if the site settings to allow users to publish and schedule tasks is enabled.
- For linked tasks, the Server administrator can do this on all sites where the server settings to allow users to schedule linked tasks is enabled. The Site administrator can do this on sites they have access to if the site settings to allow users to schedule linked tasks is enabled.
- Flow owners and project leaders with the Creator site role can create flow tasks for flows or projects that they own respectively.

Site administrators, flow owners, and project leaders can create flow tasks for the flows and projects that they own respectively. Only the Creator site role and above can create or edit a flow task.

For more information, see Set Users' Site Roles and Content Ownership and Permissions

Notify Users of Successful Flow Runs

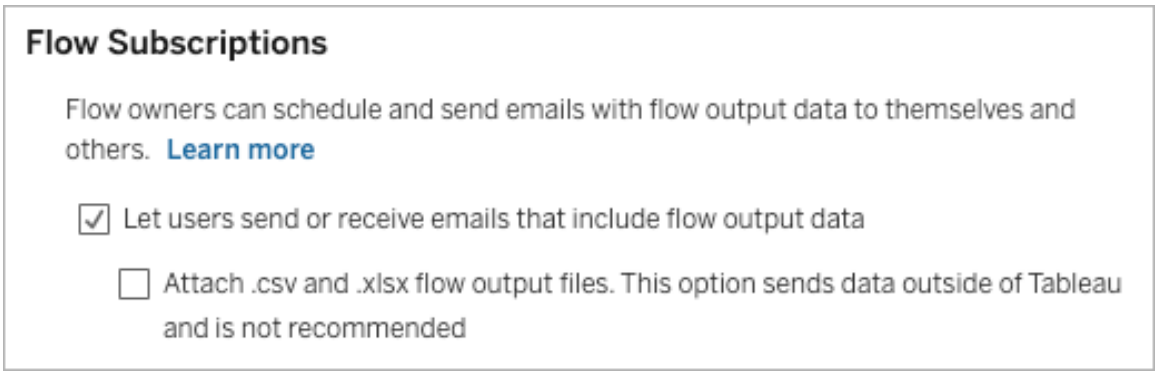
Supported in Tableau Prep Builder version 2021.4.1 and later and in Tableau Server and Tableau Cloud version 2021.4 and later. Data Management is required to use this feature.

Flow owners can subscribe themselves, individual users, and groups to email notifications for information about scheduled tasks for successful flow runs. The email includes links to data within the Tableau environment, or you can optionally include the details of the flow run in attached Excel and CSV files.

Flow subscriptions are added to scheduled tasks for flows. You can add flow subscriptions when you create a new flow task or to an existing flow task. Email notifications are sent when the scheduled task is completed successfully.

Configure the site settings for flow subscriptions

By default, the **Flow Subscriptions** site setting for sending and receiving email notifications is enabled.



- The **Let users send or receive emails that include flow output data** option allows the flow owner to receive, and subscribe users and groups to successful flow run notifications. From the notification email, users can access the full data source or view the flow details from within Tableau.
- (Not recommended) The **Attach .csv and .xlsx flow output files** option lets the flow owner attach files to notification emails. The email recipients must be added to the Tableau server or site, however, the files contain the data source and can be exposed outside of the Tableau system. This option is available only for on-premise environments.

Publish the Flow

Publish the flow output as either a file, database table, or data source. Consider the following when saving the flow:

- (On-premise only) When publishing you can save the output as a file or as a database table and choose to attach either a .csv or .xlsx file type to the email.
- When publishing and saving the output as a published data source, the email notification provides a link to the flow in Tableau. Files cannot be attached to the email.
- When choosing to save as a file output, you must use a network share and the output and input location must be included in a safe list. For more information, see Step 4: Safe list Input and Output locations.
- Flow subscriptions are supported on Windows and Linux. The following restrictions apply to flow subscriptions on Linux:

  - File outputs must be output to a Windows server.
  - For flows that output to a file, use the UNC format for the path: *server**path**file-name*. Do not use a local drive letter.
  - The mounted path must be safe listed.

- When attaching files to an email, the file limit is approximately 25 MB for Tableau Cloud. When using an on-premise Tableau Server, you configure the size of attachment files.

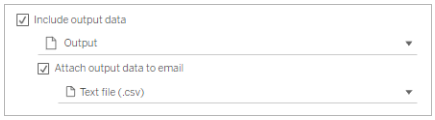
Add a flow subscription

1. As the owner of the flow, select a published flow in Tableau Server or Tableau Cloud that you want to add subscriptions to for email notifications.
2. You can add subscriptions to a new or existing task:
   - If you are adding a subscription to a new task:
     1. Click **Scheduled Tasks** > **New Task**.
     2. In the New Task dialog select a schedule to run the flow from the **Select a schedule** drop-down list.
     3. Enable **Send email when done**
   - If you are adding a subscription to an existing task:
     1. Click **Subscriptions** > **Subscribe**.
     2. In the **Add Flow Subscriptions** dialog, select a schedule from the **Frequency** drop-down list.
3. In the **Send to** field, start typing the name of the user or group to populate the field. Select the users and groups that you want to send a notification to.

   Users and groups must be added to the Tableau environment by the administrator.
4. (Optional) To be included in the notification, check the **Send to me** box.
5. (Optional) In the **Subject** field, customize the default email subject line for the flow run notification.
6. (Optional) Add information about the flow run in the **Email message** text box.

7. Click **Include output data** and select the type of output that you want to include in the email.

    - If you published your flow as a file or database table output you can choose to attach .csv or .xlsx files containing the data source to the email. This is not recommended because data can be exposed outside of the Tableau system.

    - If you published your flow as a data source, you can choose to include the link to the data source. Attaching files to the email is not supported.

8. If you are adding a subscription to an existing task, click Subscribe.

Unsubscribe from a flow subscription

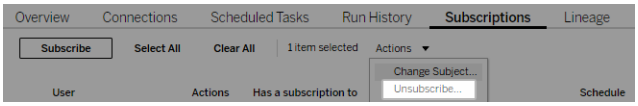
To unsubscribe from notifications from the email, follow these steps.

1. Click **Unsubscribe** from the bottom of a subscription email.
2. As the flow owner, Sign in to Tableau Server or Tableau Cloud. At the top of the page, click the **Notifications** icon.
3. Expand the **…** menu, then select **Remove notification**.

To unsubscribe and remove the subscription as the flow owner, follow these steps.

1. Click **Subscriptions**.
2. Open the published flow in Tableau Server or Tableau Cloud.
3. From the list of flow subscriptions, click the selection box for the flow you want to unsubscribe from.

4. Select **Actions** > **Unsubscribe**.



View Subscriptions

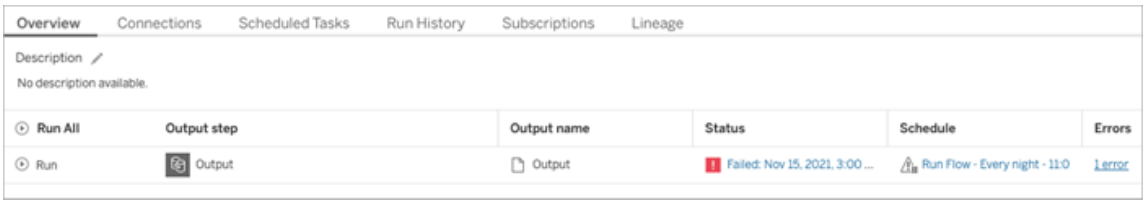
You can view your current flow subscriptions in Tableau Server or Tableau Cloud.

- From the **Subscriptions** tab on the **Overview** page of the flow, you can see the list of current subscriptions.
- From the **Subscriptions** tab on the **Tasks** page, you can see the list of subscriptions along with subscriptions to workbooks.

Resume suspended flow subscriptions

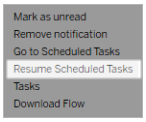
Sometimes, subscriptions fail because of an issue with the flow. If a subscription fails more than five times, you'll receive a notification email thatyour scheduled flow task has been suspended.

From the flow Overview page, you can see when a scheduled flow task fails.

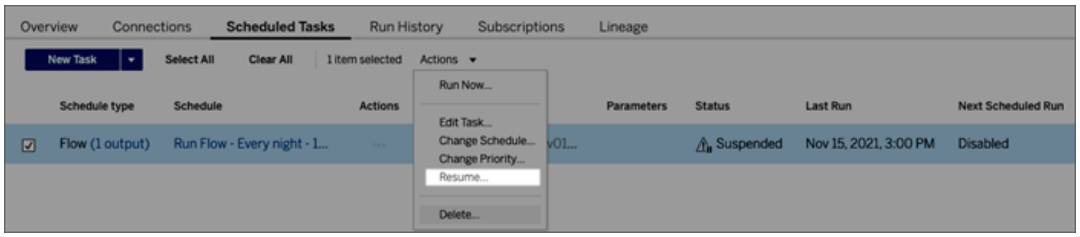


There are a few ways to resume a suspended flow task. If you're flow owner:

- From the My Content area of Tableau web pages, an icon appears in the Last update column to indicate that the subscription is suspended. Select **...** > **Resume Scheduled Tasks** to resume.



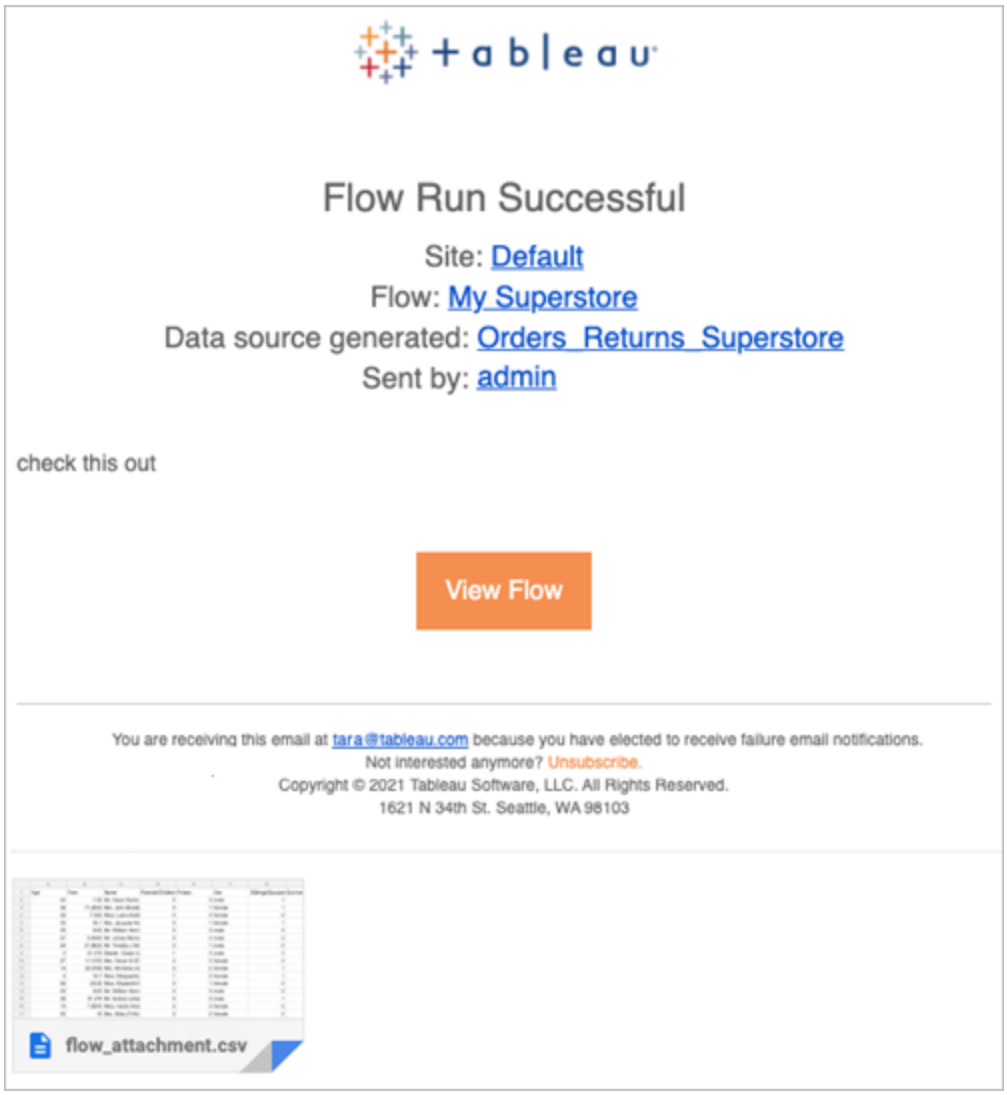
- From the Scheduled Tasks page, an icon appears in the last update column to indicate that the subscription is suspended. Select the flow, then click **Actions** > **Resume**.



Access the flow data from a notification email

Depending on how the flow notification was configured, you can access the data source and file attachments from the notification email.

- Click View Flow to open the flow in Tableau Server or Tableau Cloud.
- Click the attachment file to view the flow data.



Who can do this

- Flow owners can create flow notification subscriptions for flows that they own.
- To receive notifications, users and groups must be added to the Tableau environment by the administrator.

For more information, review the following articles:

- Windows: Set Users' Site Roles and Content Permissions and Ownership
- Linux: Set Users' Site Roles and Content Permissions and Ownership

Manage a Flow

Once you publish a flow to Tableau Server or Tableau Cloud, you can manage your flows and make changes to them as necessary. This topic describes the various actions you can take to manage your flows.

> **Note**: The content in this topic applies to both Tableau Server and Tableau Cloud, exceptions are called out specifically.

Managing your flows

Following is a list of actions you can take to manage your flows:

- **Create flows:** Starting in version 2020.4, as a Creator you can create flows directly on the web. From the **Home** page, click **Create** > **Flow** or from the **Explore** page, click **New** > **Flow**. For more information, see Tableau Prep on the Web.
- **Edit flows:** Starting in version 2020.4, as a Creator you can edit flows directly on the web. Navigate to the list of flows, select **Actions** and click **Edit Flow**, or open a flow and click the **Edit** button.

  When you edit a flow, your changes are moved to a draft state. When you're finished, publish your flow to commit your changes and create a new version of the flow. For more information, see Autosave and working with drafts.
- **Run flows:** You can manually run a flow in addition to creating scheduled flows tasks that run at a specific time.
- > **Note**: The Data Management is not required to manually run flows, but is required to schedule flows to run.
- Navigate to the list of flows, select one or more flows you want to run, select **Actions** and click **Run Now**.

- **Tag:** Tags are keywords you can create for flows to help you find, filter, and categorize content. Authors can add tags to flows when they publish it. But you can also add tags to any workbook, view, or data source that you are allowed to access and you can delete any tags you have added. You can add a tag to a list of flows.

  Navigate to the list of flows, select one or more items you want to tag, select **Actions** and click **Tag**. To add a tag to a specific flow you can do it from the list of flows as described above. Open the Flow, from the Overview tab, select **Actions**, and click **Tag**.

- **Change Owner:** Administrators and flow owners can change owners, and only to themselves.

- **Permissions:** You can set permissions for users and specify if they can perform edit actions like save, download, move to a different project and delete. In addition, you can specify who can view and run the flow.

- **Download:** You can download a flow to view or modify it using Tableau Prep Builder. To download a flow, you need download permissions. You'll have that by default if you are the owner, but you might need to add it for other users.

- **Revision History:** When you make a change to the flow, and republish it to the same project with the same name, a new version of the flow is created. You can view the revision history by selecting revision history from the actions menu. Flow owners have permissions to restore a previous version of a flow.

- **Move:** You can move flows between projects. To move a flow, users need Move permission. You'll have that by default for flows you own, but you might need to add it for other users.

- **Rename:** You can rename a flow. To rename a flow, users need the Save permission. You'll have that by default for flows you own, but you might need to add it for other users.

- **Delete:** You can delete a flow. To delete a flow, users need the Delete permission. You'll have that by default for flows you own, but you might need to add it for other users.

Who can do this

Tableau Server Administrator

Can do the following tasks across all the sites:

- Create flows
- Edit published flows
- View a list of all draft flows
- Run flows
- Delete flows
- Download flows
- Change Owner
- Change Permissions
- Change Project
- Add/Remove Tags
- Change Description
- Change Name
- Update Flow Task
- Delete Flow Task

- Create a Flow Task*
- Version Management*

*Some additional conditions apply to these actions:

- To create a flow task:
    - A flow schedule must be available. Only Server Administrators can create a schedule.
    - The flow must have at least one output step.
    - The flow version must be compatible with the Tableau Server version.
- Version management:
    - Revision history must be enabled on the site.
    - User role is allowed to publish to the project.

Tableau Site Administrator

Can do the following tasks on flows published to the sites that they are site administrator for:

- Create flows
- Edit published flows
- View a list of all draft flows
- Run flows
- Delete flows
- Download flows
- Change Owner
- Change Permissions

- Change Project
- Add/Remove Tags
- Change Description
- Change Name
- Update Flow Task
- Delete Flow Task
- Create a Flow Task*
- Version Management*

Some additional conditions apply to these actions:

- To create a flow task:
  - A flow schedule must be available. Only Server Administrators can create a schedule.
  - The flow must have at least one output step.
  - The flow version must be compatible with the Tableau Server version.
- Version management:
  - Revision history must be enabled on the site.
  - User role is allowed to publish to the project.

Project Leader

Can do the following tasks on flows published to the projects where they have project leader permissions:

- Create flows
- Edit published flows
- Run Flows
- Delete
- Download
- Change Permissions
- Change Project
- Add/Remove Tags
- Change Description
- Change Name
- Update Flow Task
- Delete Flow Task
- Create a Flow Task*
- Version Management*

*Some additional conditions apply to these actions:

- To create a flow task:
    - A flow schedule must be available. Only Server Administrators can create a schedule.
    - The flow must have at least one output step.
    - The flow version must be compatible with the Tableau Server version.
- Version management:

- Revision history must be enabled on the site.
- User role is allowed to publish to the project.

Project Owner

Can do the following tasks on flows published to the projects that they own:

- Create Flows
- Edit published flows
- Run Flows
- Delete
- Download
- Change Permissions
- Change Project
- Add/Remove Tags
- Change Description
- Change Name
- Update Flow Task
- Delete Flow Task
- Create a Flow Task*
- Version Management*

* Some additional conditions apply to these actions:

- To create a flow task:
    - A flow schedule must be available. Only Server Administrators can create a schedule.
    - The flow must have at least one output step.
    - The flow version must be compatible with the Tableau Server version.
- Version management:
    - Revision history must be enabled on the site.
    - User role is allowed to publish to the project.

Flow Owner

Can do the following tasks on flows that they own:

- Create Flows
- Edit Draft (flows they own) and Published Flows
- Run flows
- Delete flows
- Download flows
- Change Owner
- Change Permissions
- Change Project
- Add/Remove Tags
- Change Description
- Change Name

- Update Flow Task
- Delete Flow Task
- Create a Flow Task*
- Version Management*

*Some additional conditions apply to these actions:

- To create a flow task:
  - A flow schedule must be available. Only Server Administrators can create a schedule.
  - The flow must have at least one output step.
  - The flow version must be compatible with the Tableau Server version.
- Version management:
  - Revision history must be enabled on the site.
  - User role is allowed to publish to the project.

User with Creator License

Can do the following tasks:

- Create Flows
- Edit Draft (flows they own) and Published Flows
- Run Flows (with Run flow permissions)
- Delete (with delete permissions)
- Download (with download or Save as, and read permissions)
- Change Permissions (with Change Permissions)

- Change Project (with move permissions, and write permissions on the destination project)
- Add/Remove Tags (with read permissions)
- Change Description (with Save permissions)
- Change Name (with Save permissions)
- Update Flow task (with Execute permissions)
- Delete Flow Task (with Execute permissions)
- Create a Flow Task* (with Execute permissions)
- Version Management* (with view, read, save as, download permissions)

* Some additional conditions apply to these actions:

- To create a flow task:
  - A flow schedule must be available. Only Server Administrators can create a schedule.
  - The flow must have at least one output step.
  - The flow version must be compatible with the Tableau Server version.
- Version management:
  - Revision history must be enabled on the site.
  - User role is allowed to publish to the project.

User with Explorer License

Can do the following tasks:

> **Note**: Starting in version 2020.4, Explorer license users can no longer run flows on Tableau Server.

- Explorer license users (with Run flow permissions) can run flows on Tableau Cloud.
- Delete (with delete permissions)
- Download (with download or Save as, and read permissions)
- Change Permissions (with Change Permissions)
- Change Project (with move permissions and write permissions on the destination project)
- Add/Remove Tags (with read permissions)
- Change Description (with Save permissions)
- Change Name (with Save permissions)
- Update Flow task (with Execute permissions)
- Delete Flow Task (with Execute permissions)
- Create a Flow Task* (with Execute permissions)
- Version Management* (with view, read, save as, download permissions)

*Some additional conditions apply to these actions:

- To create a flow task:
  - A flow schedule must be available. Only Server Administrators can create a schedule.
  - The flow must have at least one output step.
  - The flow version must be compatible with the Tableau Server version.
- Version management:

- Revision history must be enabled on the site.
- User role is allowed to publish to the project.

User with Viewer License

Viewers cannot manage flows, they can however view the flow and the different versions of the flow.

For more information about the full capabilities you can set on flows, see Permission capabilities.

Monitor Flow Health and Performance

After you publish flows and schedule them to run periodically, you want to know that they are running as expected and resolve any issues as they occur. You will also want to monitor and understand the performance of your flows.

This topic describes the various methods that Tableau Server provide to help you monitor your flows.

Detect issues as they occur and resolve them

You can set up Tableau Server to send email notifications when flows fail, or find and review errors on Tableau Server using the Alerts menu or by reviewing the flow pages for the flows that you are interested in. This type of monitoring allows you to detect problems as they occur.

Get notifications when a flow fails:

You can configure Tableau Server to send notifications about flow failure through email, on the Tableau site, or Slack. To set up notifications for your Tableau Server and sites, follow the instructions in Step 2: Configure Flow Settings for your Tableau Server.

View and resolve errors

> **Note**: Starting in version 2020.4.1, you can now create and edit flows directly in Tableau Server and Tableau Cloud. The content in this section applies to all platforms, unless specifically noted. For more information about authoring flows on the web, see Tableau Prep on the Web.

The following errors can happen when running a flow:

- **Connection errors:** Connection errors generally happen when Tableau Server is unable to connect to one or more data inputs or is unable to make a connection in one or more output steps.
    - For Input connection errors, use the **Edit connections** option on the **Connections** tab to make changes to connection details, then run the flow again.
    - For output connection errors, check the output location for the flow output steps. If the flow output is going to a network share, make sure the output steps are pointing to a safe listed location. After you make any changes republish the flow and try running it again.

      > **Note**: To fix output connection errors for flows that output to a file or network share, download the flow to Tableau Prep Builder, then republish the flow to your server. Flows that output to a published data source or database can be edited directly on the web.

- **Errors in the flow:** If there are errors in one or more steps in the flow, you will see an error message. You can edit the flow directly on the web and republish it. You can also download the flow to Tableau Prep Builder, resolve the errors, republish the flow to the server and then run the flow again.

- **Suspended flow tasks:** When a scheduled flow task fails to run after a configured number of attempts, the flow task is suspended. By default, a flow task is suspended after 5 consecutive flow tasks failures.

  A flow can have multiple scheduled tasks assigned to it, but only the failed tasks are suspended. All other flow tasks will continue to run unless they have errors. To resolve a suspended task, review and resolve the errors then run the flow on-demand or let the flow run automatically based on the assigned schedule.

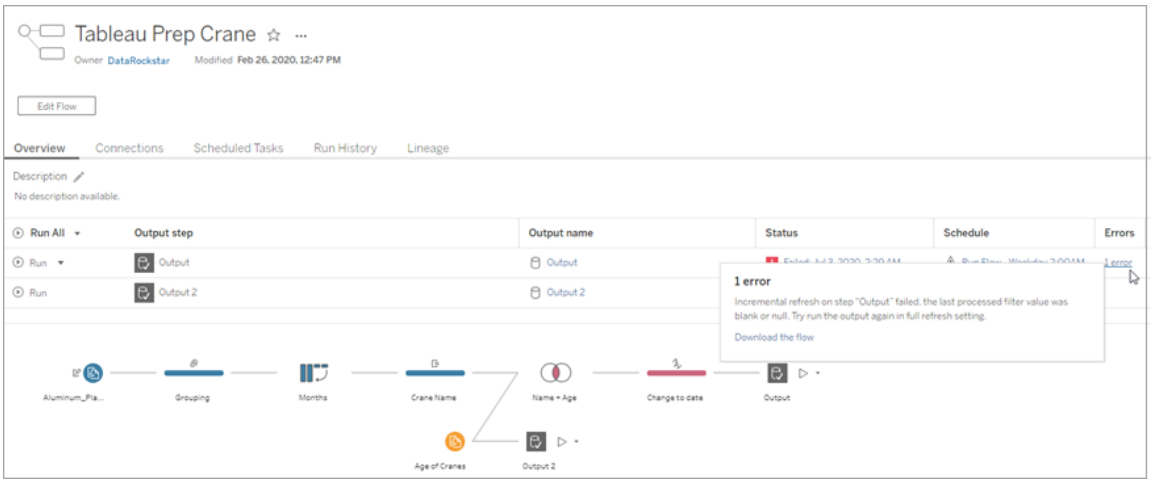
  > **Note:** Server administrators can configure the number of attempts before a flow run is suspended using the tsm configuration set option. For more details, see Step 5: Optional Server Configurations.

You can view errors on the following pages:

Flow Overview page

On this page you can see the status of the most recent flow run and any errors. Hover on the error text to review the error details. If a scheduled task is suspended, a warning icon shows next to the schedule. Hover on the icon to view the status.

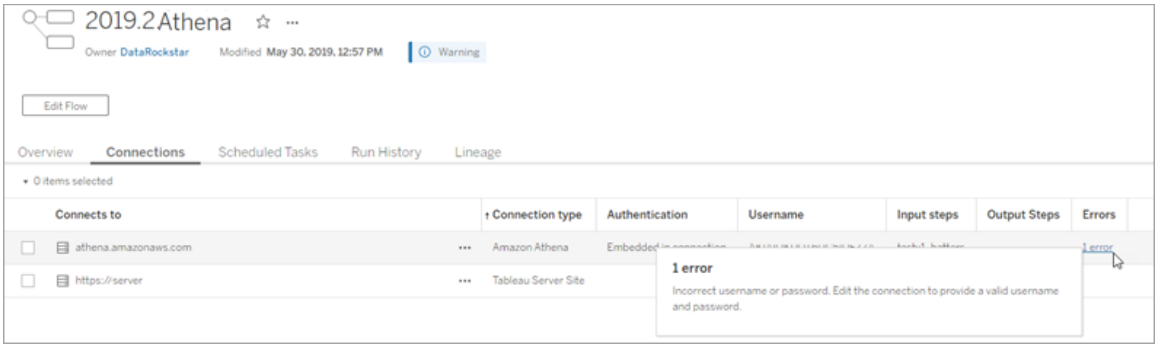
After you resolve the error that caused the flow to fail or the task to be suspended, you can run the flow manually or let the flow run based on the assigned schedule. For suspended flow tasks, click the **Go to Scheduled Task** link on the tooltip for the suspended task to navigate to the **Scheduled Tasks** page and click the **Resume Scheduled Tasks** button to resume the suspended tasks.



Connections page

The **Connections** page shows the most recent status and any related connectivity errors. To correct input errors, click the **More actions** ••• menu for an input connection to edit the connection and change the server name, port, user name and password.

To fix output connection errors, edit the flow directly or download the flow in Tableau Prep Builder, correct the file path, then republish the flow to continue running it.



Scheduled Tasks page

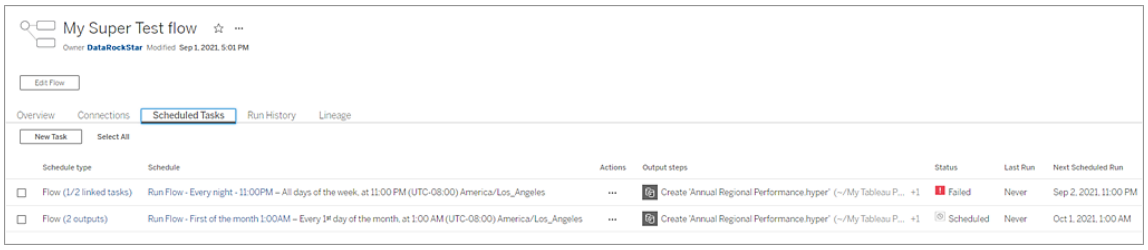
Note: The Data Management is required to see this tab.

View the scheduled tasks assigned to a flow. If a scheduled task is suspended, you can see the status of that tasks here and you can manually resume the flow tasks from this page. Before resuming a suspended task, resolve any errors in the flow.

Error details are not shown on this page, but you can review them on the **Overview** or **Run History** pages. You can also click the links in the **Schedule type** column to view the details of what was scheduled and to edit the tasks.

A suspended scheduled task will automatically resume when the flow is republished, if you edit a connection for the flow or manually run the flow tasks. To manually resume a suspended tasks, on the **Scheduled Tasks** page, click **Resume Scheduled Tasks**. This resumes all suspended tasks for the flow.

To resume individual tasks, click the **More actions** ••• menu for a scheduled task and select **Resume**. You can also click **Run Now** to run all tasks for the flow immediately.



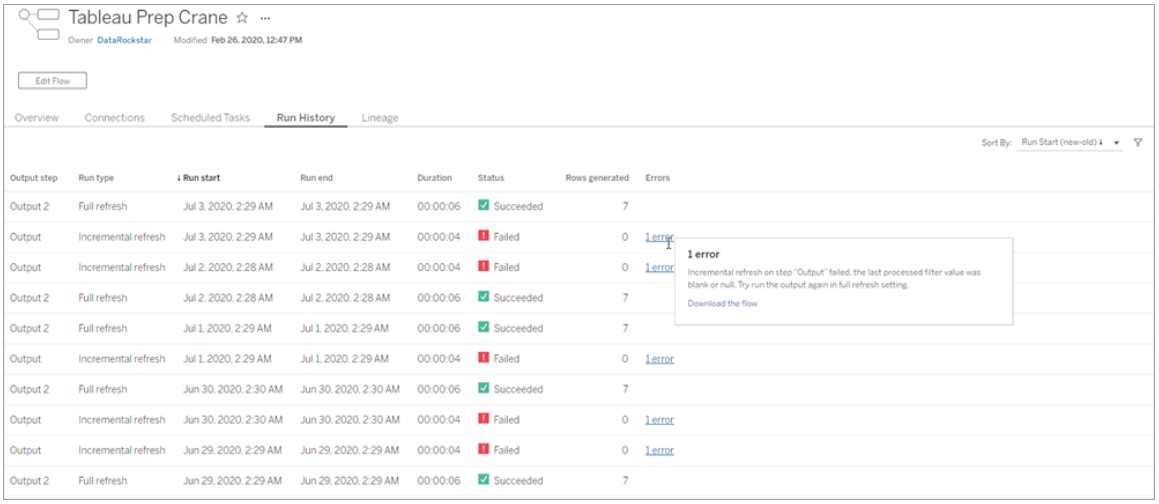
Run History page

Note: The Data Management is required to see this tab.

The **Run History** page shows the details of all the flow runs that have either completed or are in progress for each output. View any error details by hovering over the errors in the **Errors** column. The duration column shows you the run time of the flow.

Note: Starting in version 2020.2.1, the **Run Type** field shows the refresh type for the output. In prior releases this field showed whether the output was run on a schedule or on-

demand. For more information about setting up output refresh types, see Refresh Flow Data Using Incremental Refresh.



Alerts

When a flow fails, the alerts menu is populated with the error details with the option to re-run the flow, or download the flow to troubleshoot.

Note:Flow owners, Server or Site Administrators can see this menu.

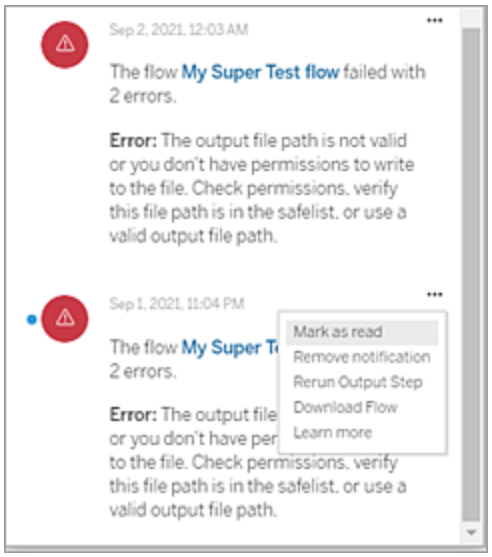
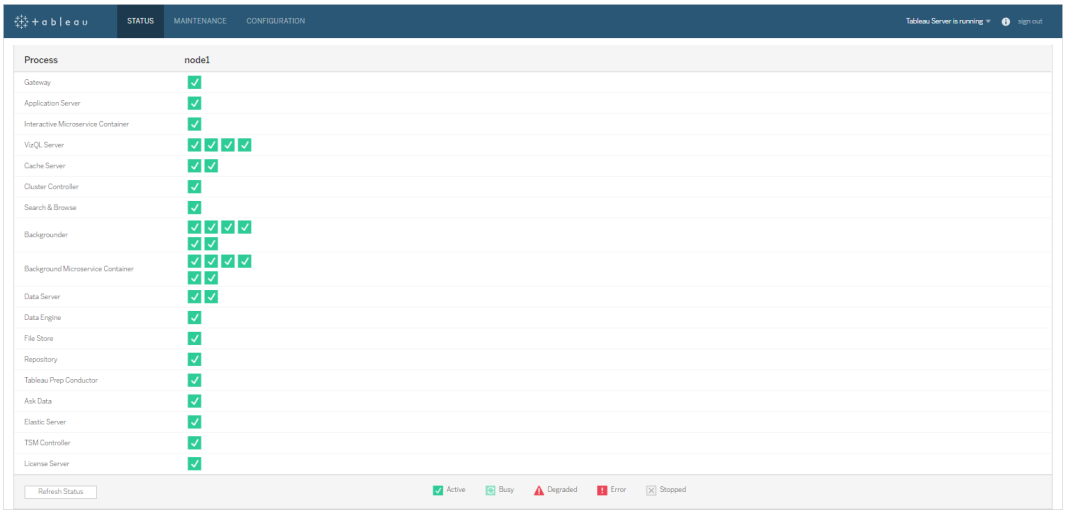


Tableau Prep Conductor process status

- The Tableau Services Manager (TSM) status page is accessible in TSM and can be viewed by TSM administrators. You must be able to log into TSM to see this page.
- The Tableau Server status page appears in the Tableau Server web UI and is accessible by Tableau Server administrators. If you hover your mouse pointer over the status indicator for a process, a tooltip shows the node name and the port the process is running on. The Tableau Server status page does not show TSM processes.



When Tableau Server is functioning properly, Tableau Prep Conductor will show as Active or Busy:

- **Active**—The process is functioning as intended.
- **Busy**—The process is completing some task.
- **Down**—The process is down. The implications of this differ depending on the process.
- **Status unavailable**—Tableau Server is unable to determine the status of the process.

Who can do this

- **Tableau Server Administrators**:
  - Setup email notifications at the server level
  - Set up email notifications for a site
  - View errors
  - Resume suspended tasks
  - View alerts
  - view process status
- **Tableau Site Administrators**:
  - Set up email notifications at the site level
  - View errors
  - Resume suspended tasks
  - View alerts
- **Flow owners, project leaders and any user who is granted permissions to view the flow**:
  - View errors
  - Resume suspended tasks
  - View alerts (Flow owners)

Administrative Views for Flows

Administrative views can be used to monitor the activities related to flows, performance history, and the disk space used. The **Status** page contains an embedded Tableau workbook

with various administrative views that can be used to monitor different types of server or site activity.

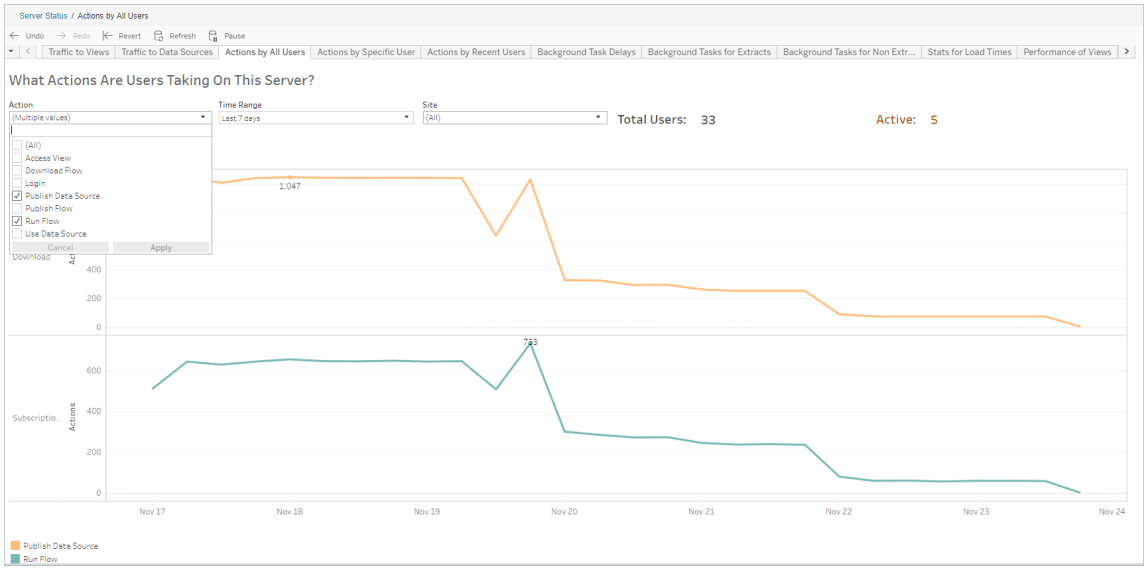
For **Tableau Server** installations, you must install PostgreSQL drivers before you can see Administrative views. For more information, see Database Drivers (Linux), Database Drivers (Windows). Server administrators can use these views to see activity both at the server level (aggregated for all sites) or for a specific site. Only server administrators can filter by site.

Who can do this?

Tableau Server administrators and Tableau Site administrators can both view and work with Administrative Views. Only Server administrators can filter by site.

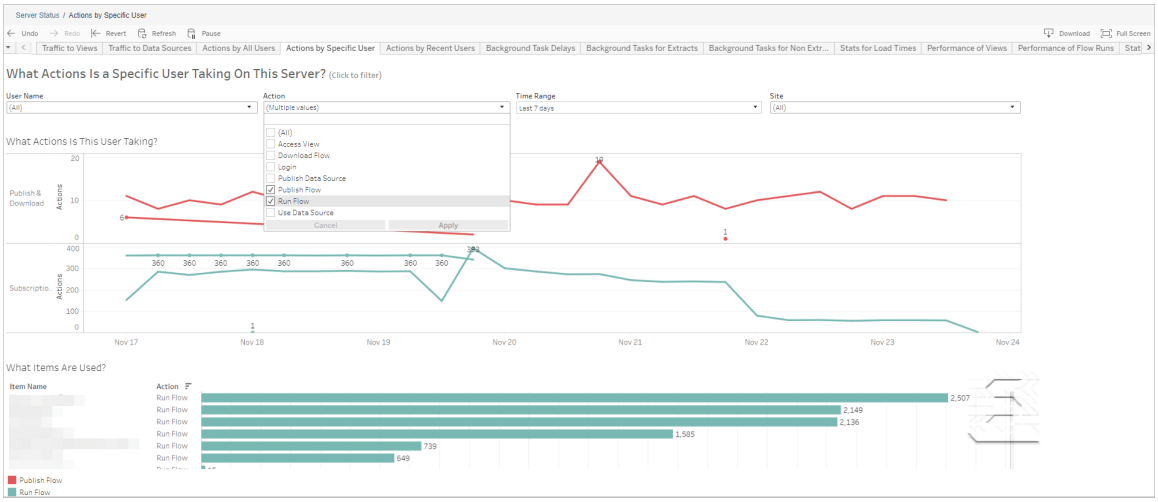
Action by all users

Use this view to gather insight into how flows are being used. This includes actions like publish, download, and flow runs. You can filter the view by actions, by site, and by time range. The Total Users count shows the number of users who have performed an action. This value is not affected by any filtering. The Active user count shows the number of users who have been active during the selected time period and performed one of the selected actions.



Action by Specific User

Use this view to gather insights about how an individual user is working with flows. You can filter the view by user name, the type of action, time range, and by site.



Action by Recent Users

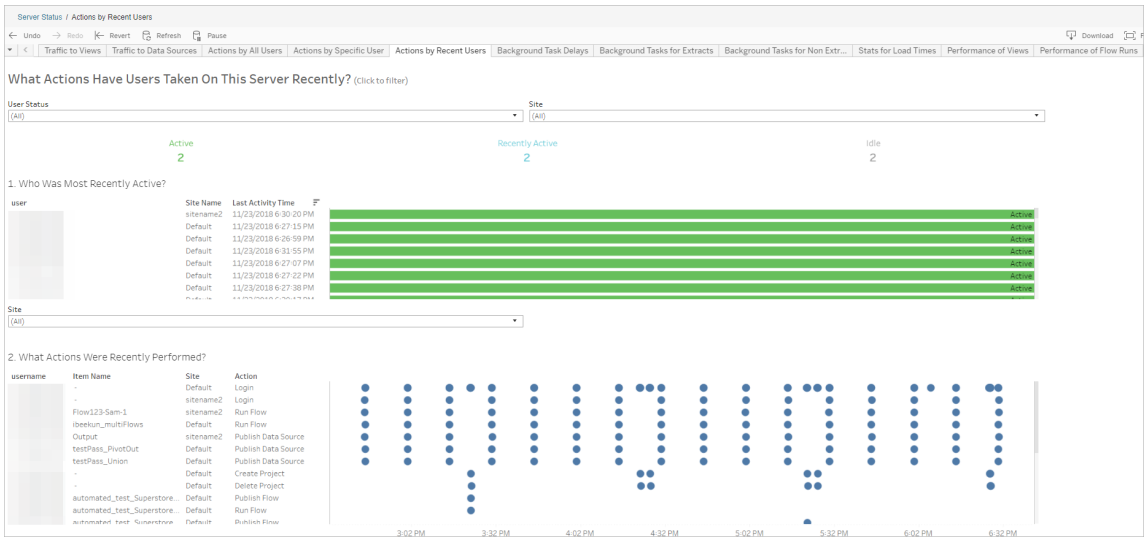
This view shows you which users have been active on Tableau Server over the past 24 hours.

This can be useful if you need to do some maintenance activity on the server and want to know which users and how many this will affect, and what they're doing.

The view shows **Active**, **Recently Active**, and **Idle** users that are currently signed in to Tableau Server.

For this view, an active user is one who took an action in the last 5 minutes, a recently active user is one who last took an action within 30 minutes, and an idle user is one who last took an action more than 30 minutes ago.

Select a user to see only the actions that user performed recently. Hover over an action to see details of the action.



Backgrounder Task Delays

This view shows the delay for extract refresh tasks, subscription, and flow tasks—that is, the amount of time between when they are scheduled to run and when they actually run. You can use the view to help identify places you can improve server performance by distributing your task schedules and optimizing tasks.

Possible reasons for the delays and ways to reduce the delays include the following:

- Many tasks are scheduled for the same time.

  In the example view, tasks that show long delays are clustered at the same time every day, which creates spikes in the wait time. You can set the Timeline filter to a single day to view task delays by hour and identify the hours of the day when many tasks are scheduled at the same time. One solution is to distribute the tasks to off-peak hours to reduce load on the server.

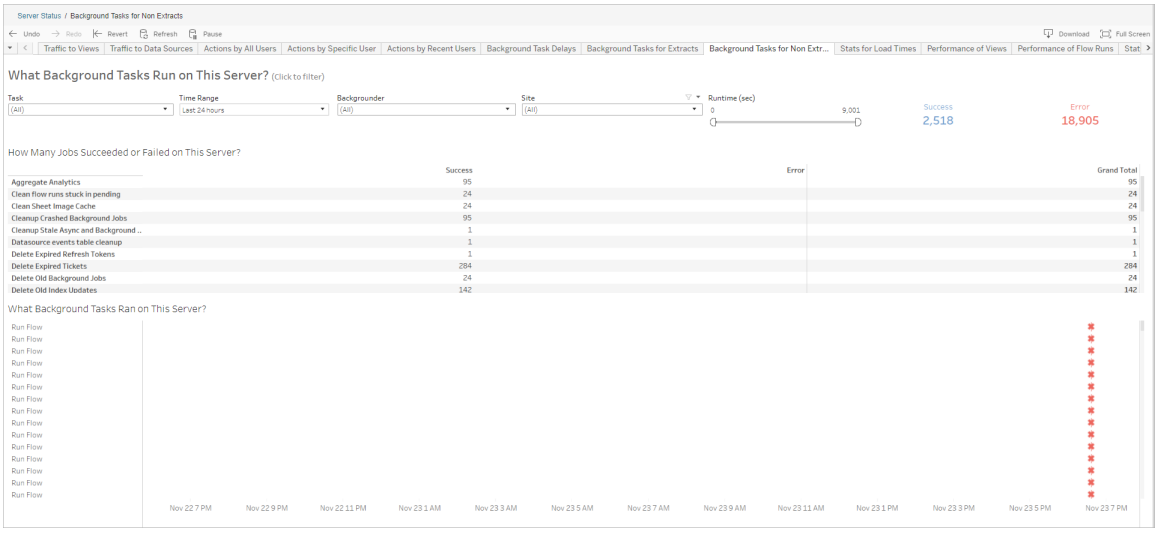
- Other server processes running at the same time are consuming server resources and slowing down performance.

  Monitor the CPU and memory usage of server processes to identify processes that consume the most resources, then adjust the configuration of processes on your server.

  For more information on monitoring processes, see Collect Data with Windows Performance Monitor.

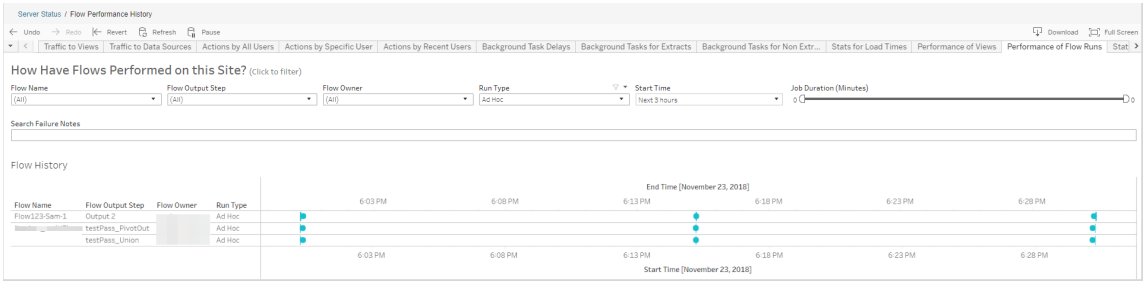
Background Tasks for Non Extracts

Background Tasks are created to run flows (scheduled and ad hoc). You can use this view to see how many flow tasks succeeded or failed on this site. For details on a task, hover over its icon.



Performance of Flow Runs

Use this view to see the performance history for all the flows on a site. You can filter by Flow Name, Output Step Name, Flow Owner, Run Type (Scheduled or Ad Hoc), and the time the flow runs were started.



Questions you can answer using this view include:

- **What flow tasks are currently scheduled?** – To do this, use the Start Time filter and select the time frame you want to look at. For example, to see flow tasks that are scheduled in the next 3 hours, select **Hours** -> **Next** -> and enter **3**.
- **What is the duration of flow tasks?** - To answer this, click on a mark in the view to see details, including the task duration.
- **How many flows were run ad hoc, and how many were scheduled runs?** - To answer this, use the **Run Type** filter and select **Ad hoc** or **Scheduled**.

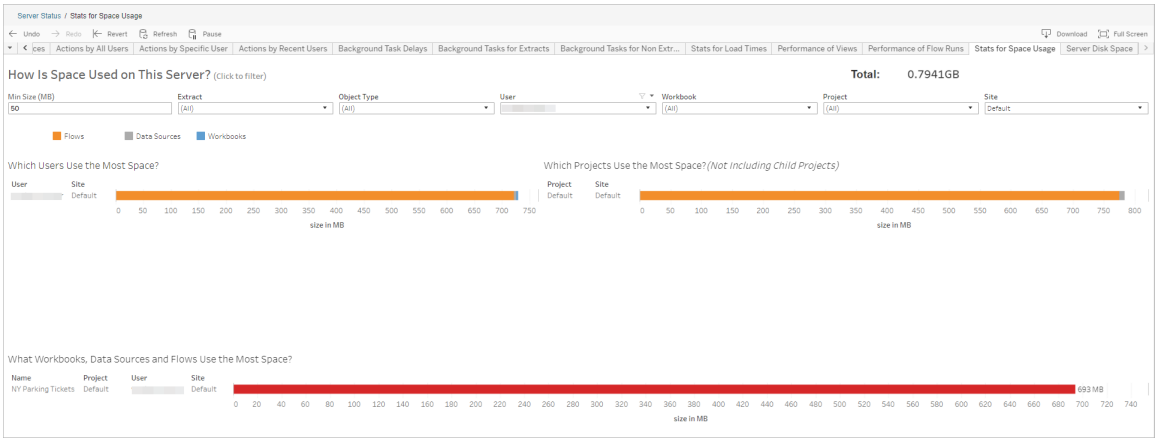
This view can also show you the following information:

- Flows with the highest run frequency have the most marks.
- To see flows that are currently running at the same time, hover over a mark that shows "**In Progress**" or "**Pending and select** "**Keep Only**" to filter all flow runs that are currently running.
- To see flows that are running at the same time during a specific time range, select a

range for the **Start Time** filter. For example, select "**Next three hours**" to see which flows will be running in the next three hours.

Stats for Space Usage

Use this view to identify which flow outputs are taking up the most disk space on the server. Disk space usage is displayed by user, project, and by the size of flow output and is rounded down to the nearest number.



Use the Min Size filter to control which flow outputs are displayed, based on the amount of space they take up. Use the object type filter for flows.

- **What Users Use the Most Space** – This section shows the users who own flows (when filtered for flows) that are taking up the most space. Click a user name to filter the next two graphs for that user.
- **What Projects Use the Most Space** – This section shows the projects with flows (when filtered for flows) that are using the most space.
- **What Workbooks, Data Source and Flows Use the Most Space** – This section shows the flows (when filtered for flows) that take up the most space.

Who can do this

- **Tableau Server Administrators**:
  - Setup email notifications at the server level
  - Set up email notifications for a site
  - View errors
  - Resume suspended tasks
  - View alerts
  - view process status
- **Tableau Site Administrators**:
  - Set up email notifications at the site level
  - View errors
  - Resume suspended tasks
  - View alerts
- **Flow owners, project leaders and any user who is granted permissions to view the flow**:
  - View errors
  - Resume suspended tasks
  - View alerts (Flow owners)

Developer Resources - REST APIs

Use Tableau Server REST APIs to automate and seamlessly integrate Tableau into your existing workflows. Tableau Server REST API gives you programmatic access to work with your content, users, sites, and now flows. Manage provisioning, permissions, and publishing on Tableau Server or Tableau Cloud via HTTP. The REST API gives you access to the functionality behind the data sources, projects, workbooks, site users, sites, and flows. You can use this access to create custom applications or to script interactions with server resources.

Tableau REST API supports the following flow functionality:

New flow endpoints have been added to support publishing flows, scheduling flows, running flows on demand, managing permissions, downloading flows and such. For a full list of all the new REST API endpoints for flows, see Flow Methods.

In addition, existing endpoints have been updated to support flow functionality such as Creating New Schedules for Flows, Creating new sites, updating existing sites, and managing default permissions.

Note: The Data Management license is required when using REST API to run flows.

About Tableau Catalog

Data is increasing in volume, formats, and importance leading to more complex environments. With the rapid pace that data changes, it can be hard to keep track of that data and how it's being used in such complex environments. At the same time, more users need to access more of that data in more places, and it’s difficult for users to find the right data. Ultimately, this causes a lack of trust in the data because people question whether they’re using the right source or if the source is up to date.

Tableau Catalog integrates features like lineage, impact analysis, data dictionary, data quality warnings, and search into your Tableau applications, helping solve these problems differently from a stand-alone catalog. It focuses on both IT and the end user so that everyone using Tableau Server or Tableau Cloud has more trust in and visibility into the data, while also enabling more discoverability. Tableau Catalog builds a catalog out of the Tableau content being used by your organization, enabling comprehensive functionality like the following:

- **Impact analysis and lineage**.
  - You can see the workbooks and other Tableau content that depend on particular columns or fields from tables or data sources you manage. When you need to make changes to your data, you can notify the impacted Tableau authors using email.
  - As a workbook author, you can use lineage to trace the fields that your workbook depends on.
  - As a user, when you use a Tableau visualization, you can see where the data came from that was used to create the view.
- **Curation and trust**. As a data steward, you can add helpful metadata, like descriptions and certification, so that users find the right data. You can set data quality warnings,

view data details on the Data Details pane, certify assets, and remove assets from the catalog.
- **Data discovery**. In Tableau Desktop or Tableau web authoring, you can use Tableau Catalog to search for databases, tables, data sources, and virtual connections to analyze in Tableau and connect to them from the search results.

Starting in 2019.3, Tableau Catalog is available as part of Data Management for Tableau Server and Tableau Cloud. When the product key is active and enabled, the catalog features described above are integrated into the product you're using, so you can work with the data where you find it.

How Tableau Catalog works

Tableau Catalog discovers and indexes all the content on your site—workbooks, data sources, sheets, virtual connections, and flows—to gather metadata about the content. From the metadata, external assets (databases, tables, and other objects) are identified. Knowing the relationships between the content and the external assets enables Tableau to display the lineage of the content and external assets. Tableau Catalog also enables users to connect to external assets using Tableau Server or Tableau Cloud.

Users on your site can publish or delete content, can attach data quality warnings or certifications, or do anything else that changes the content or its metadata on the site, and Tableau Catalog will update its information accordingly.

For information about how you can use Tableau Catalog to support data governance in your organization, see Governance in Tableau in the Tableau Blueprint Help.

Key Tableau Catalog terms

- Metadata. Information about the data.
- Tableau content. Content created in Tableau such as workbooks, data sources, virtual connections, and flows.
- External assets. The metadata about the databases and tables used by Tableau content that's published to Tableau Server or Tableau Cloud.

License Tableau Catalog

Tableau Catalog is licensed through Data Management. For information about how Data Management licensing works, see License Data Management.

Enable Tableau Catalog

After Tableau Server or Tableau Cloud is licensed with Data Management capabilities, you can enable Tableau Catalog by doing one of the following tasks:

- **For Tableau Cloud**, no action is necessary. Tableau Catalog is on by default, configured to use derived permissions, and ready to use. For more information about derived permissions, see the Permissions on metadata topic.
- **For Tableau Server**, the Server admin must first enable the Tableau Metadata API using the `tsm maintenance metadata-services` command. For more information, see Enable Tableau Catalog.

  After the Metadata API is enabled, Tableau Catalog is on by default, configured to use derived permissions, and ready to use. For more information about derived permissions, see the Permissions on metadata topic.

Features and functionality

To learn more about the features you can use with Tableau Catalog, see the following Help articles:

Data discovery

- In the **Connect** pane on Tableau Desktop, under **Search for Data** select **Tableau Server** to connect to data using Tableau Server or Tableau Cloud. When Tableau Catalog is enabled, in addition to searching for published data sources to connect to, you can now search for and connect to the specific databases, tables, and objects used by published data sources and workbooks on your Tableau Server or your Tableau Cloud site.
- Search is expanded to include results based on columns, databases, tables, and other objects when Tableau Catalog is enabled.
- If you author in the web, in addition to published data sources, you can also connect to databases and tables.

- If you use Tableau Prep on the web, you can create new flows based on external assets, such as databases and tables.
- If you connect to Salesforce Data Cloud, you'll see support for native Data Cloud objects built into Tableau Catalog. Data Lake Objects (DLOs), Data Model Objects (DMOs), and calculated insights appear distinct in search, connect, and lineage pages, which makes discovering, connecting to, and reusing them simpler.

Curation and trust

- Certify your data assets to help users find trusted and recommended data.
- Set data quality warnings to alert users to data quality issues, such as stale or deprecated data.
- Add sensitivity labels to warn users about data that needs to be handled with care.
- Add custom labels to classify data in ways that suit the needs of your organization
- Manage data labels to extend the label names and categories available to users.
- Categorize items on Tableau Server and Tableau Cloud using tags to help users filter external assets.
- Better understand published visualizations by using the Data Details tab to see information about the data used.
- Add descriptions to databases, tables, and columns to help users find the data they're looking for.

Lineage and impact analysis

- Use lineage to trace the source of your data and to analyze the impact of changes to your data and identify which users might be impacted.
- Email owners of a workbook, data source, or flow about data-related updates. Do the same with database, table, or object contacts.

Developer resources

You can use metadata methods in the Tableau REST API to programmatically update certain metadata. For more information about the metadata methods, see Metadata Methods in the Tableau Server REST API.

In addition to the REST API, you can use the Tableau Metadata API to programmatically query metadata from the content published to Tableau Server or Tableau Cloud. The Metadata API is fast and flexible and is best when you are looking to find out specific inform-

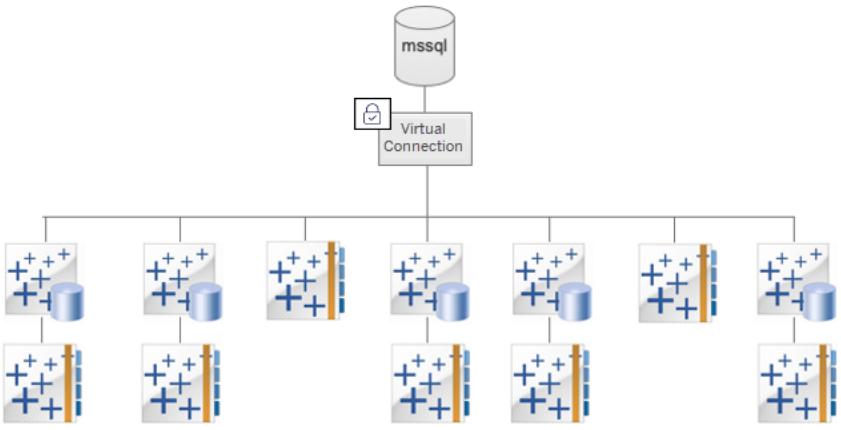
ation about the relationship between metadata and its structures. Explore and test queries against the Metadata API using an interactive in-browser tool called GraphiQL.

Note: Data Management is not required to use the Metadata API or GraphiQL.

About Virtual Connections and Data Policies

Virtual connections are a Tableau content type, along with data sources, workbooks, and flows, to help you see and understand your data. Virtual connections provide a central access point to data. Another key feature introduced with virtual connections is data policies, which support row-level security at the connection level, rather than the workbook or data source level. Row-level security data policies are applied to any workbook, data source, or flow that uses the virtual connection.

A virtual connection can access multiple tables across several databases. Virtual connections let you manage extracting the data and the security in one place, at the connection level.



For information about row-level security options, see an Overview of Row-Level Security Options in Tableau.

Not every virtual connection has an associated data policy. You can also use a virtual connection simply as a central place to manage connection credentials.

Key terms

- Virtual connection. A sharable resource that provides a central access point to data.
- Connection. The server name, database, and credentials you use to access data. A virtual connection has one or more connections. Each connection accesses one database or file.
- Virtual connection table. A table in a virtual connection.
- Data policy. A policy that's applied to one or more tables in a virtual connection to filter data for users. For example, use a data policy to apply row-level security to tables in a virtual connection.
- Policy table. A fact or data table in a data policy that is filtered.
- Policy column. A column that's used to filter the data in the policy tables. A policy column can be in a policy table or in an entitlement table.
- Entitlement table. A table that includes both a policy column you can use to filter policy tables and another column you can relate (map) to a column in a policy table.
- Policy condition. An expression or calculation that is evaluated for every row at query time. If the policy condition is TRUE, then the row is shown in the query.

License virtual connections and data policies

Virtual connections and data policies are licensed through Data Management. For information about how Data Management licensing works, see License Data Management.

Enable virtual connections and data policies

Virtual connections and data policies are automatically enabled on Tableau Server and Tableau Cloud with Data Management.

Permissions

Permissions for virtual connections work much like the permissions for other Tableau content. After you publish a virtual connection, anyone can view the connection. However, only the connection creator and administrators can access data using the connection, until the connection creator explicitly grants more permissions.

When you create a virtual connection, you must set the permissions for the Connect capability to enable other users to connect to data using the virtual connection. The Connect capability

allows you to share a virtual connection and allows users to query it. With connect permissions, a user can view the tables in a virtual connection and create content using the tables. For more information, see Set permissions on a virtual connection.

Permissions vs. data policies

Permissions define what a person can or can't do with a piece of content in Tableau. Permissions are made up of capabilities—the ability to do things like view content, web edit, download data sources, or delete content. Permission rules define which capabilities are allowed or denied for a user or group on a piece of content. The interplay between license level, site role, and potentially multiple permission rules factor into the final determination of what a person can or can’t do—their effective permissions. See Permissions for details.

Data policies filter the data in a virtual connection, making sure that people see only the data they're supposed to see. A data policy is applied and filters the data when it's viewed in the Tableau content (for example, a workbook or flow). The policy condition in a data policy is a calculation or expression that defines access to the data. User functions are often used to limit access to users or groups. Access can be based on the user name, the group a user belongs to, or a region value. See Create a Data Policy for Row-Level Security for details.

Both permissions and data policies govern access. Simply put, permissions determine which *content* you can see, access, use, or create; data policies determine which *data* you can see.

How permissions and data policies work together

Tableau permissions are applied to Tableau content first. People can only do the things they have the capabilities to do with Tableau content—data policies don't override Tableau permissions. After permissions are evaluated, the data policy is applied to determine which data in the virtual connection the person can see based on the policy condition.

The following example describes the effects of permissions and data policies on a virtual connection that contains salary data:

- The virtual connection is in the HR project, which is restricted to Tableau users in the HR group. Anyone outside the HR group can't see content in the HR project, which means they can't browse to, connect to, or view the virtual connection.

- The virtual connection has Connect permissions granted only to members of the HR Business Partners group. All others in the HR group can see that the virtual connection exists, but they can't view the data it contains. When they view a workbook that uses that virtual connection, they can't see any data.
- The virtual connection also contains a data policy that filters the salary data based on the individual user, so HR Business Partners can see only rows that pertain to employees in their business unit. When they view a workbook that uses that virtual connection, they see data only for their business unit.

Features and functionality

For the manager of data, virtual connections provide:

- **Securely managed service accounts.** If you use a 'service account' model, now instead of having to share that service account information with any user who wants to access that data, you can give the service account credentials to the few analysts who are empowered to create virtual connections.
- **Agile physical database management.** You must make database changes (for example, a field is added or table name is changed) only one time in the virtual connection, rather than in every piece of content where the data is used.
- **Reduced data proliferation.** By centrally managing extract refresh schedules, refreshes are scheduled once, ensuring that anyone who accesses the data from that virtual connection is seeing fresh data.
- **Centralized row-level security.** You can create data policies that apply row-level security to both Tableau extracts and live queries at the connection level. The data policies are applied to any workbook, data source, or flow that uses the virtual connection.

Note: Data policies are valid for flow input data, but not for flow output data. Users with access to flow output data will see all of the data, and not only a subset of it that pertains only to them.

As the user of data, you benefit from virtual connections knowing that you have:

- **Appropriate access** to only the data you should see, because row-level security is already applied to the data.

- **Flexibility** to use data that's been curated and secured. The virtual connection stores and shares the connection information. All you have to do is create a data source with a data model specific to your needs.
- **Trust** that data is fresh because the extract refresh schedule has already been set.
- The ability to **share** content freely, assured that you won't put security at risk because data policies are always enforced.

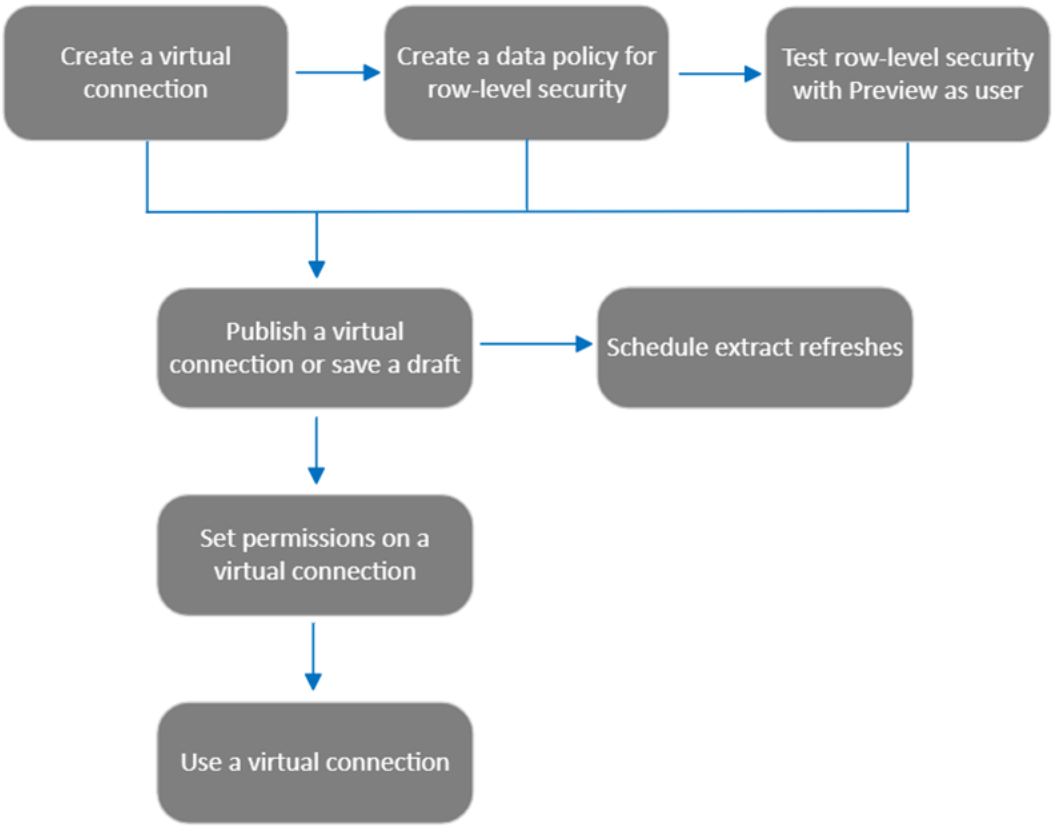
Virtual connection editor workflow

The virtual connection editor enables you to create:

- Virtual connections, which are a Tableau content type that provides a sharable central access point to data.
- Data policies that support row-level security at the connection level.

After you create a virtual connection and its associated data policies, you can publish it and set the permissions to share with other users. You can also schedule extract refreshes so that all content that uses the virtual connection is accessing fresh data.

The following diagram shows the workflow to create a virtual connection. At any time during the process, you can publish or save a draft of your connection, but the connection must be published before you can schedule extract refreshes or use (or edit) a virtual connection. You must also set permissions before others can use the connection.



Click a step in the process to go to that help topic.

Next step

The first step is to Create a Virtual Connection.

Create a Virtual Connection

A virtual connection is a Tableau content type that provides a sharable central access point to data, and supports row-level security at the connection level. Creating a virtual connection is a multi-step process. This topic covers connecting to the data you want to share and working in the Tables tab of the virtual connection editor.

Connect to data

To create a virtual connection in Tableau Cloud or Tableau Server:

1. From the Home or Explore page, click **New** > **Virtual Connection**.
2. In the Connect to Data dialog box, select the connector for your data. For a list of supported connectors for virtual connections, see Creators: Connect to Data in the Tableau Desktop and Web Authoring help.
3. Enter the information you're prompted for. The credentials you enter are saved in the virtual connection, so connection users don't have to enter credentials to connect to the data.
4. Click **Sign In** if prompted. To add another connection, click  and select a connector, enter credentials, and sign in.

A virtual connection can have multiple connections. Each connection accesses one database or file.

> **Note:** For Tableau Cloud, virtual connections that connect to private network data use Tableau Bridge to keep data fresh. For information about configuring Tableau Bridge, see Configure and Manage the Bridge Client Pool. For information about supported connections, see Connectivity with Bridge.

Add another connection

As needed, add another connection to a virtual connection and connect to more than one database by clicking  next to **Connections**. You can add a connection to a different server or database, or to the same server or database.

With multiple connections, you can:

- Use a table from any connection or database as an entitlement table in a data policy that secures tables from other connections and databases.
- Add or replace tables in a virtual connection with tables from a different database. For example, say you migrate data from one database to another. In the virtual connection editor, you can add a connection to the second database and replace the existing tables from the first database with tables from the second one.
- Add multiple connections to the same server or database. This can be helpful when you need to for example access data from the same database but with different credentials.

- Share a group of tables that are related or meant to be used together, no matter where they're physically located. For example, from multiple databases you can group tables related to employee information.

When you open a virtual connection to edit it, if prompted you must authenticate connections in sequence. If any connections fail to authenticate, you can't edit the virtual connection.

Select tables to include in the connection

If necessary, select a database to view the tables in it.

1. On the left, under **Tables**, select the tables and click or drag them to the Tables tab on the right. You can include tables from different connections. Include an entitlement table, if you're using one.
2. (Optional) Click **New Custom SQL** to create a custom table schema.

Note: Virtual connections don't support tables with a spatial data type.

Select live or extract mode for tables

You can set individual tables—whether they're from multiple connections or not—to either live or extract mode in the same virtual connection.

- **Live**- Tables are queried directly from the database. (Live is the default.)
- **Extracts**- Tables are extracted and saved to Tableau.

As an example, you can set some tables to extract mode so that they're not impacted by report generation or heavy customer traffic.

Under Tables, select the table or tables you want to change the mode for and select **Actions**, and **Change to Live** or **Change to Extract**. Alternatively, select the Actions Menu (...) in the table's row and select **Live** or **Extract**.

Incremental Extracts

Starting in Tableau Cloud June 2024 and Tableau Server 2024.2, you can configure table extracts for incremental refresh.

When configuring an incremental extract, you specify a key column that is used to identify new rows. When the incremental extract is refreshed, only the rows where the key column has increased will be added to the extract. Fewer rows processed means a faster extract refresh job and less load on the database.

For example, suppose we have an extract for the **Batters** table, and the data in the extract is:

Year	Team
1978	Lions
1979	Tigers

The **Batters** table is configured for incremental extract refresh, and the key column is **Year**.

The live table is updated with a new row for **1980**:

Year	Team
1978	Lions
1979	Tigers
1980	Bears

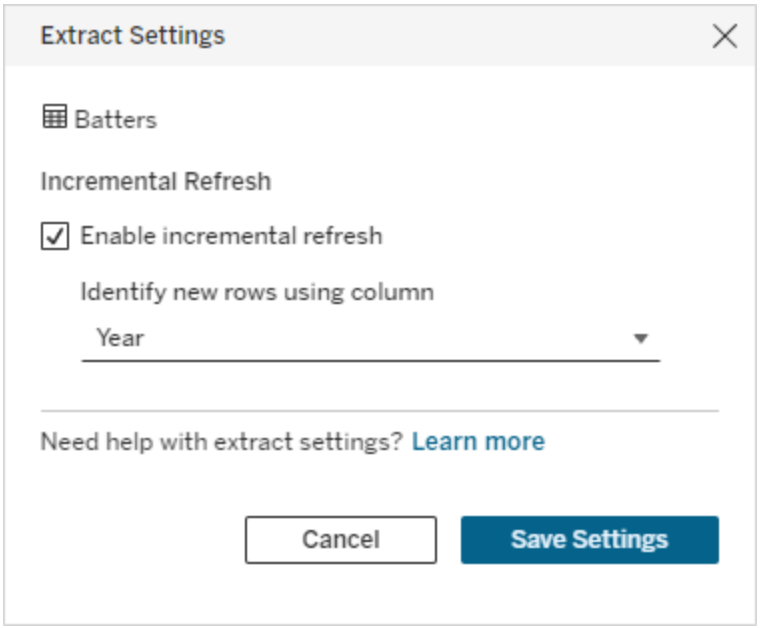
When the **Batters** table's extract is refreshed incrementally, only rows that exceed the greatest value in the extract's key column are added. In this case, that means that the **1980** row is added to the extract. Instead of refreshing the entire extract file, only 1 row is processed and appended.

You can still do a full refresh on an extract configured for incremental refresh if you want to refresh the entire extract.

To configure incremental refresh for a table extract:

1. Change the table from **Live** to **Extract.**
2. Select the Actions Menu (...) in the table's row and select **Extract Settings...**.
3. Check **Enable incremental refresh**.

4. Select an incrementing column to use when determining which rows to add.
5. Select **Save Settings**.



Convert to Custom SQL

Starting in Tableau Cloud June 2024 and Tableau Server 2024.2, you can approximate the SQL used to connect to a table and use that as a starting point for your own custom SQL. Custom SQL allows you to filter or make other query changes that can modify the result set. Creating custom SQL this way instead of using **New Custom SQL** in the data pane is less impactful to existing virtual connections. Downstream assets see the table as the same table instead of a new one.

To convert a table to custom SQL:

1. Select the Actions menu (...) in the row for the table.
2. Select **Convert to Custom SQL**.
3. In the **Edit Custom SQL** dialog, edit the SQL as needed.
4. Select **Generate Table**.

Note: The SQL that first appears in the dialog should be considered a starting point, and may not work without modification. The virtual connection editor lacks nuanced

> information about the specific SQL syntax used in the connection. If you encounter errors when selecting the Generate Table button, try removing or changing single quotes, double quotes, back quotes, and square brackets to make the SQL compliant with the database you're using.

To edit the custom SQL:

1. Select the Actions menu (...) in the row for the table.
2. Select **Edit Custom SQL**.
3. In the **Edit Custom SQL** dialog, edit the SQL.
4. Select **Generate Table**.

To return the table to its default state, without custom SQL:

1. Select the Actions menu (...) in the row for the table.
2. Select **Replace**.
3. In the replace table dialog, select the original table name.
4. Select **OK**.

Extract table data

After a table or tables are changed from live to extract but haven’t been extracted yet, click **Create Pending Extracts** (or **Run Pending Extracts** in Tableau Server 2023.3 and earlier) to run the pending extracts. After all pending extracts are run, click **Refresh All Extracts** (or **Run All Extracts** in Tableau Server 2023.3 and earlier) to extract all table data at that time.

Alternatively, in Tableau Cloud June 2024 or Tableau Server 2024.2 and later, select the Actions Menu (...) in the table's row and select **Refresh Extract...**. If incremental refresh is not configured for the table, you can only select **Refresh (Full)**. If incremental refresh is configured, you can choose either **Refresh (Full)** to refresh the extract completely, or **Refresh (Incremental)** to incrementally refresh the extract.

You must run any pending extracts before you publish the virtual connection. You can't edit the connection while extracts are generated.

Schedule extract refreshes of the tables in your virtual connection on the virtual connection page after you publish the connection. See Schedule Extract Refreshes for a Virtual Connection.

Set the table visibility state

Use the Visibility toggle on the Tables tab to show or hide tables and their data from users.

 Users can see table data. You can create a data policy to govern which data users can see. (Visible is the default.)

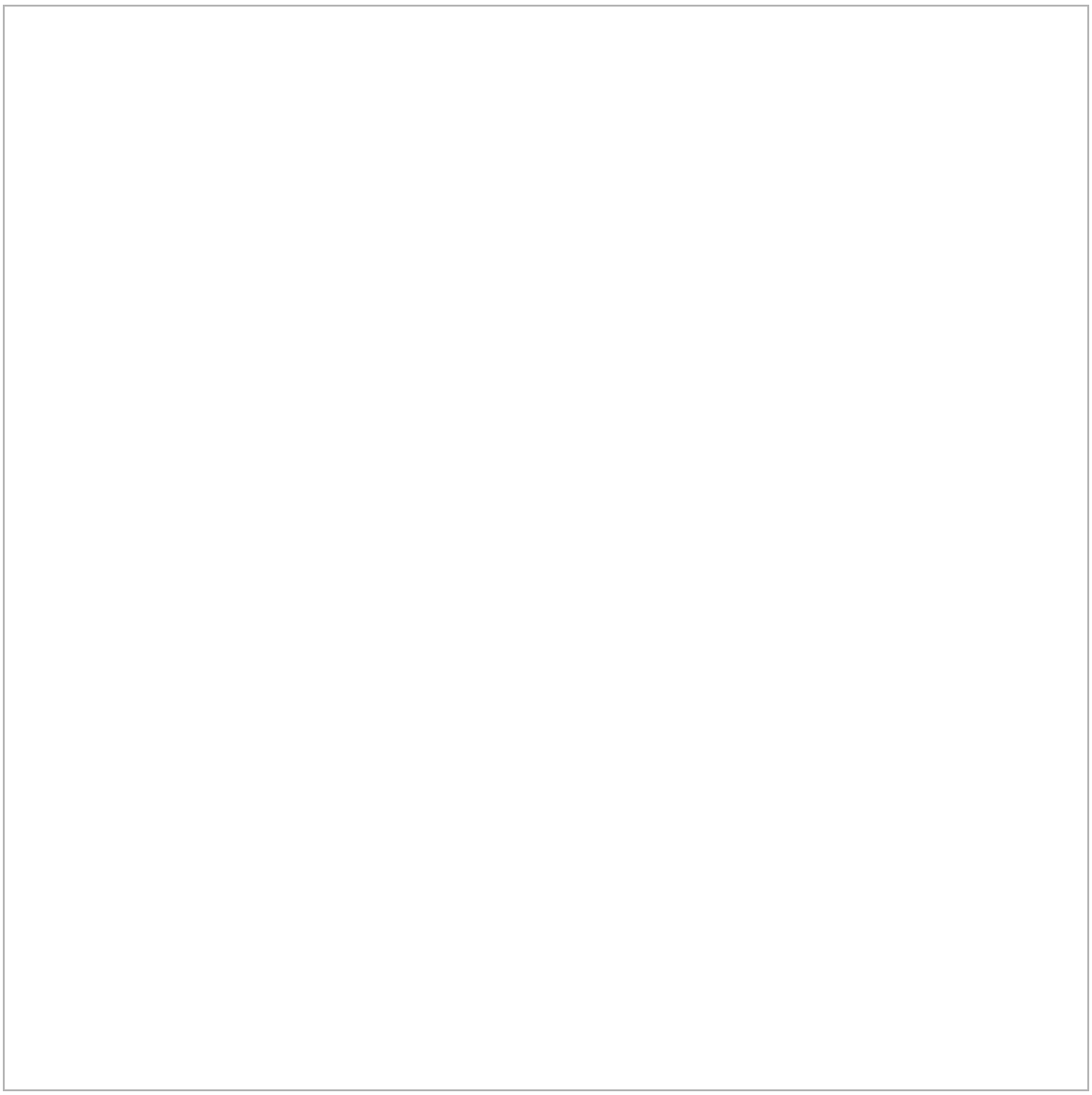
 Users can't see table data. You can use hidden tables in a data policy and as an entitlement table.

See table details

Click a table at the top of the Tables tab to see its details. You can make simple edits in the Table Details section, such as change a table name, hide or rename a column, or change a data type.

Switch the table information you see using these icons:

- A list of columns in the table and each column's data type.
- Sample data for each column and linked keys, if available. Linked keys show which columns link to other tables. They're visible only when databases have primary and foreign key information.
- The range of values in a histogram for each column selected.



Refresh data from the database

Click the refresh icon in the toolbar to get the latest data from the database for all the connections in a virtual connection, including:

- The lists of databases, tables, and columns. Both the tables included in a virtual connection and not included are refreshed.
- Table and histogram data.

For tables in live mode, refreshing retrieves the latest list of databases, tables, and columns and the most recent table and histogram data. For tables in extract mode, refreshing retrieves the updated list of tables and columns. But to see the most recent table and histogram data, you must start a new extract. For example, when there's a new column in a database table and you click the refresh icon, the new column appears in the editor but its data does not. To see the most current data, you must start a new extract.

Refreshing data invalidates any currently cached data. Closing and reopening the editor, switching tables from extract to live mode, and changing a connection credential like user-name or password also refreshes data.

Who can do this

To create a virtual connection, you must

- have credentials to the database that the virtual connection connects to, and
- be a server or site administrator, or a Creator.

Next steps

After tables have been added and configured on the Tables tab, you can choose to Create a Data Policy for Row-Level Security or Publish a Virtual Connection and Set Permissions.

See also

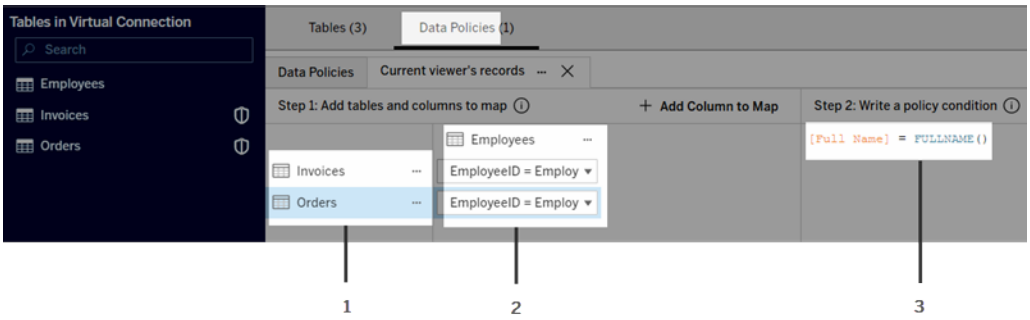
Use a .properties file to customize a JDBC connection — If you're customizing a JDBC-based connection, you can also make customizations in a .properties file

Create a Data Policy for Row-Level Security

Use a data policy to apply row-level security to one or more tables in a virtual connection. A data policy filters the data, ensuring that users see only the data they're supposed to see. Data policies apply to both live and extract connections.

About data policies

A data policy has three main components:



1. The tables it applies to, called policy tables. These are the tables that are filtered.
2. The mapped columns that define the relationships between tables (for example, between entitlement and fact tables) and between table columns and policy columns. A policy column is the column used to filter data.
3. The policy condition, which is an expression or calculation that is evaluated for every row at query time. If the policy condition is TRUE, then the row is shown in the query.

When you create a data policy, you need a column you can use to filter the data. This column is called a policy column. Data is filtered by the policy condition, usually using a user function, such as USERNAME() or FULLNAME().

If your policy table includes a column that you can filter on, then use that column as your policy column.

When a policy table doesn't include such a column, use an entitlement table with a column you can use to filter the data. An entitlement table is a table that includes both a policy column you can use to filter policy tables and another column you can relate (map) to a column in a policy table (as shown in the data policy example image above).

Filter with a policy column from a policy table

The most common way to filter data is to use a column in the table that has the data that you want to filter on. Use that column as a policy column and then map the appropriate table columns to the policy column.

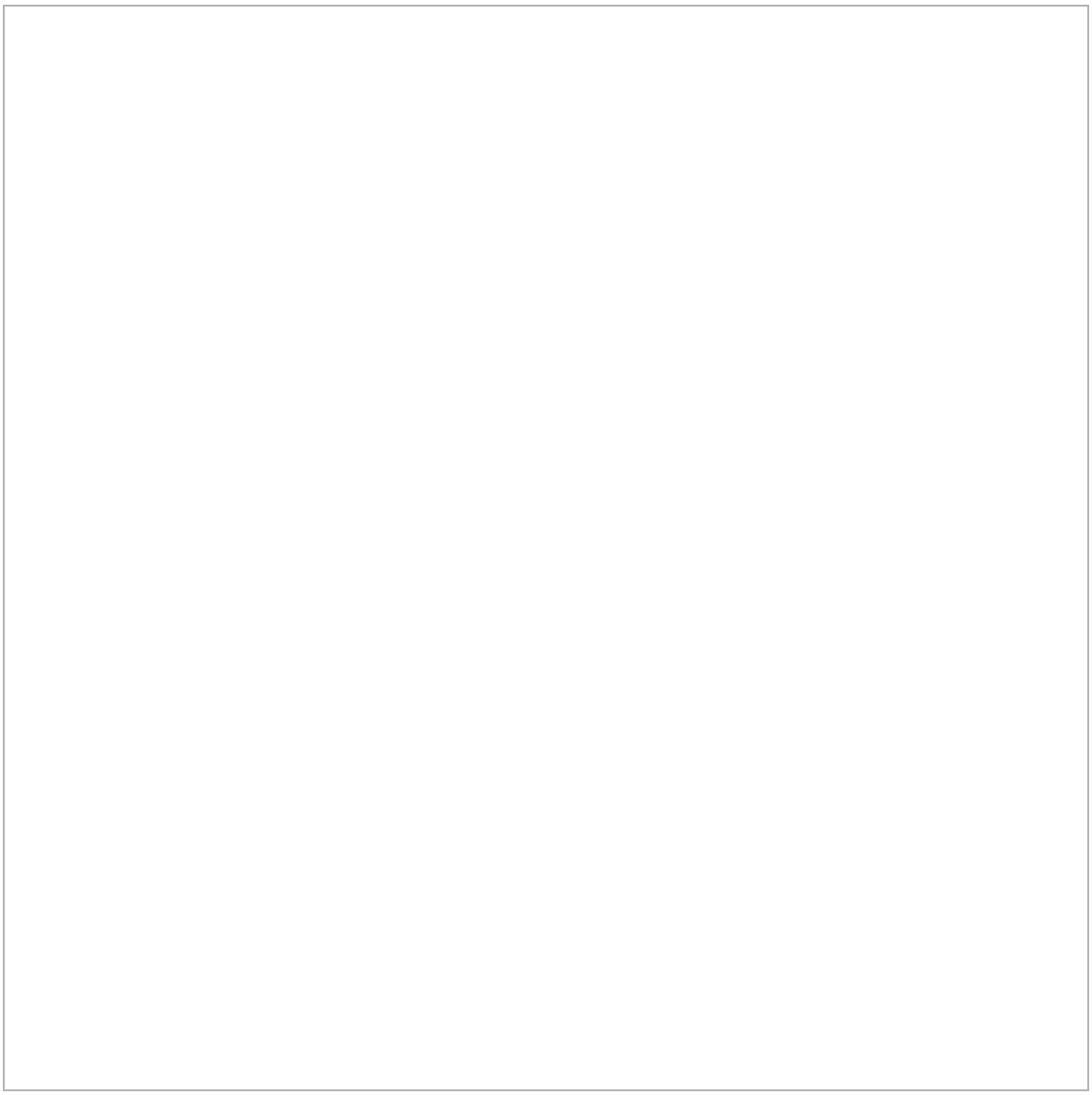
To use a policy column to filter your data, first, add tables to the policy from the left pane. To add a table, do one of the following:

- Double-click the table name.
- Click the drop-down arrow near the table name and select **Manage table with policy**.
- Or, drag the table to the right and drop it on **Add as Policy Table**.

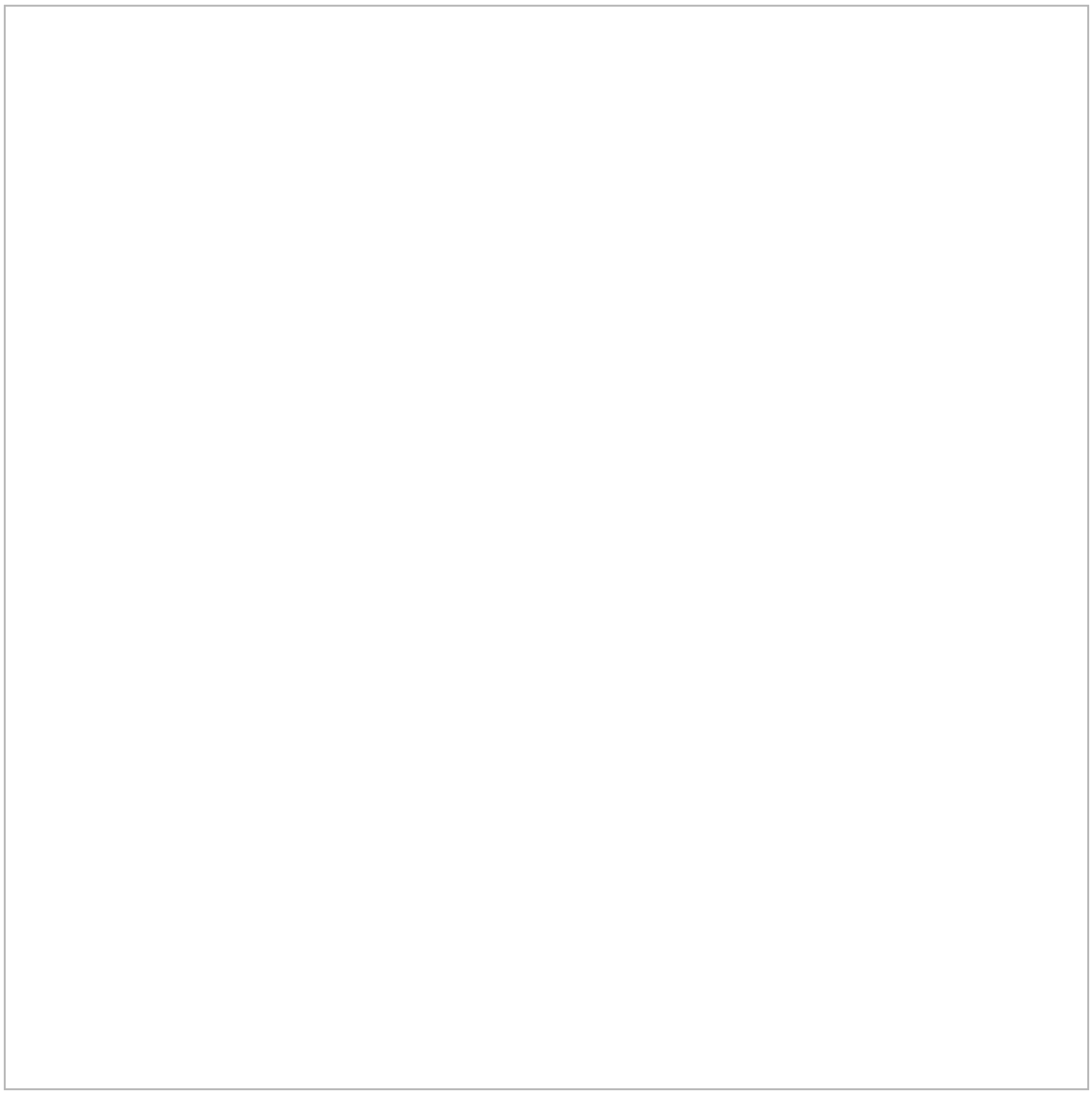
After a table is added to a policy, a shield icon appears to the right of the table name in the left pane indicating that it's a policy table.

Next, map columns to create a relationship between the column name in the table and the policy column name. Use the policy column name in the data policy condition to control row-level data access for users:

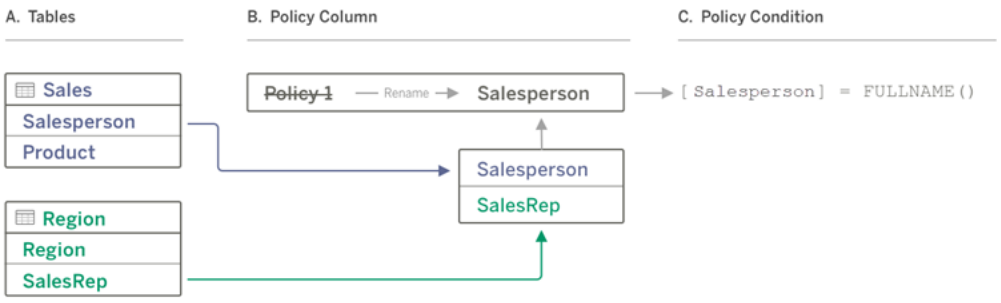
1. Click **+Add Column to Map** to add one or more columns you'll use to filter data.
2. Name the policy column. You'll use this name in the policy condition.
3. For each table the policy applies to, use the drop-down menu to select the table column that maps to the policy column.
4. Repeat this process for as many policy columns as you want to use in the policy condition.



Tip: Instead of using the +Add Column to Map button, you can start typing the calculation in the policy condition area and use auto-complete to choose the column name, which will then populate the policy column information under Step 1.



An example using a policy column from a policy table



A. The Sales table has a [Salesperson] column, and the Region table has a [SalesRep] column. The Salesperson and SalesRep data matches the full name of Tableau users on your site.
B. You want to filter the Sales and Region data by Salesperson, so you name the policy column "Salesperson" and then map the Salesperson column from Sales and the SalesRep column from Region to the Salesperson policy column.
C. Then write the policy condition to filter both tables. Use the [Salesperson] policy column and the FULLNAME() user function so that each user can see only their own data.

Filter with policy column from an entitlement table

Entitlement tables are used when your policy table doesn't contain a column you can filter on. You can use the entitlement table to map a column in the data table to a column in the entitlement table. Note the following:

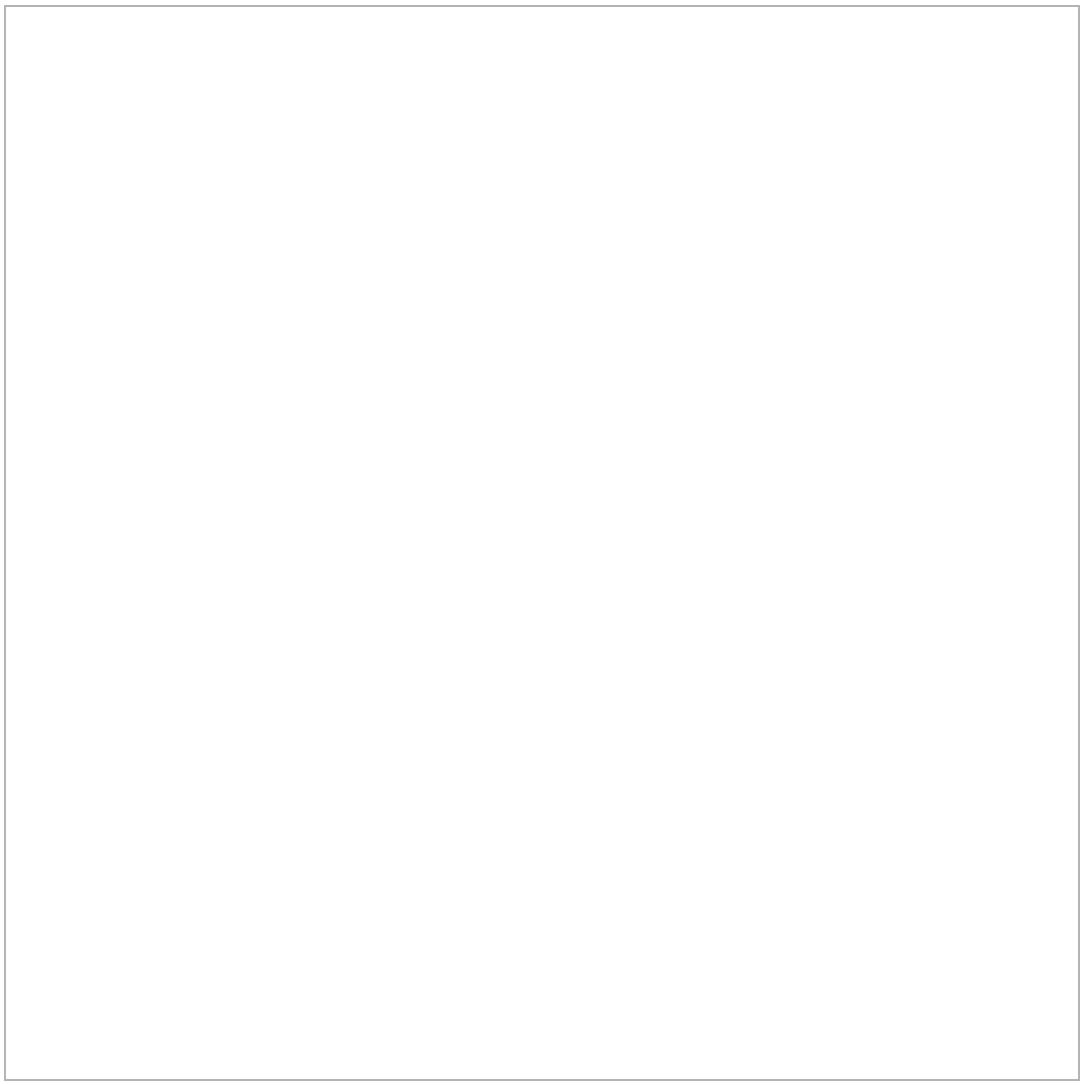
- Be sure to include the entitlement table as a table in the virtual connection. You can use a table from any connection or database as a central entitlement table that secures tables across many other databases. In some cases, an entitlement table that's in the same database as the tables you're securing can be a potential security risk because of the potential for exposing employee data. And having an entitlement table in a different database can make it easier to control permissions, for example, to grant someone access to a database.
- If you don't want virtual connection users to see the entitlement table, you can toggle the setting in the Visibility column on the Tables tab to hide it. Once hidden, the entitlement table is still available for policy filtering but can't be used in vizzes or workbook data sources.

Note: Connecting directly to a flow output (.hyper file) is not supported for the entitlement table. The flow output must write directly to the database.

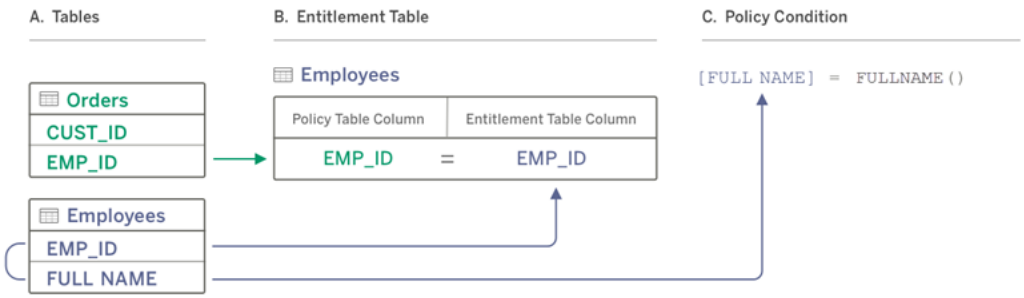
To use an entitlement table to filter your data:

1. Add the data tables that you want the data policy to apply to. Do one of the following:
   - Double-click the table name.
   - Click the drop-down arrow near the table name and select **Manage table with policy**.
   - Or, drag the table to the right and drop it on **Add as Policy Table**.
2. After a table is added to a policy, a shield icon appears to the right of the table name in the left pane indicating that it's a policy table.
3. Select the entitlement table, then either:
   - Click the drop-down arrow and select **Use as entitlement table**.
   - Or, drag the table to the right and drop it on **Add as Entitlement Table**.
4. For each table that the policy applies to, click the drop-down menu and select the

column to map the policy table to the entitlement table.



An example using a policy column from an entitlement table



A. The data you want to filter has an EMP_ID column, but not an employee name column. However, you have a second table that includes columns for both EMP_ID and the employee's FULL NAME. And, the values in the employee FULL NAME column match the full name of Tableau users on your site.
B. You can add Employees table to the policy as an entitlement table, and then map the policy table column name EMP_ID to the entitlement column name EMP_ID for each table.
C. Then use the FULLNAME() function in your policy condition to match the Tableau Server user's full name with the entitlement table's [FULL NAME] column (which is the policy column) so that each user can see only their own data.

Write a policy condition

The last step in creating a data policy is to write a policy condition, which is a calculation or expression used to define row-level access. Policy conditions are often used to limit access to users or groups through user functions.

A policy condition:

- Is required in a data policy.
- Must evaluate to true or false.
- Shows rows when the policy condition is true.

When you close a policy tab, it doesn't discard your work.

Policy condition examples

Shows only rows where the Region column value is North:

```
[Region] = "North"
```

Enables a signed-in user to see the rows where the user's name matches the value in EmployeeName:

```
FULLNAME() = [EmployeeName]
```

Enables members of the Managers group to see all rows, while users can see only the rows where their username matches the value in the employee_name column:

```
ISMEMBEROF('Managers') OR USERNAME() = [employee_name]
```

Supported Tableau functions in policy conditions

Policy conditions support a subset of Tableau functions:

- Logical (except null-related)
- String
- User
- Date
- Number: MIN, MID, MAX

To see which specific functions are supported, in the virtual connection editor, on the Data Policies tab, see the **Reference** panel on the right.

> **Note**: If the virtual connection has a data policy that contains user functions (for example, `USERNAME()`) and you connect to it from a workbook or data source and create an extract there, the extract will contain only the rows that match the virtual connection data policy at the time the extract is created. To take advantage of a virtual connection with user functions in the data policy, use a live connection from the workbook or data source to the virtual connection instead of an extract.

Who can do this

To create a data policy, you must

- have credentials to the database that the virtual connection connects to, and
- be a server or site administrator, or a Creator.

Next steps

After you create a data policy, the next step is to verify that it works as you expect it to. See Test Row-Level Security with Preview as User. Or, if you're ready to share the virtual connection and its data policies with others, see Publish a Virtual Connection and Set Permissions.

Resources

For detailed information about calculations, see Understanding Calculations in Tableau in the Tableau Desktop and Web Authoring help.

For information about user functions, see User Functions in the Tableau Desktop and Web Authoring help.

For information about other row-level security options in Tableau, see Overview of Row-Level Security Options in Tableau in the Tableau Server help.

Test Row-Level Security with Preview as User

Use **Preview as user** to test your data policy. You can see the data as the user sees it and ensure that row-level security is working as expected. This helps when the data policy keeps you from seeing the rows in the table (for example, if only salespeople can see rows, and you're not a salesperson).

To preview the data when the data policy is applied:

1. Select a table.
2. In the Table Details section, select the **With policy applied** check box.
3. Click **Preview as user**, select a **Group** (optional) and a **User**.
4. Verify that the policy shows the correct data for that user in the table details.
5. Repeat for other users as needed.

Tip: In Table Details, click to show the range of values for a column, including which values show and which are filtered out by the data policy. Select one or two columns that are good indicators that the policy is correctly filtering the data.

Who can do this

To test a virtual connection, you must

- have credentials to the database that the virtual connection connects to, and
- be a server or site administrator, or a Creator.

Next step

After you test your data policy, when you're ready to share the virtual connection with others, see Publish a Virtual Connection and Set Permissions.

Publish a Virtual Connection and Set Permissions

When you work in the virtual connection editor, your changes are automatically saved as a draft while you work. To share a new virtual connection with other users, you need to publish it.

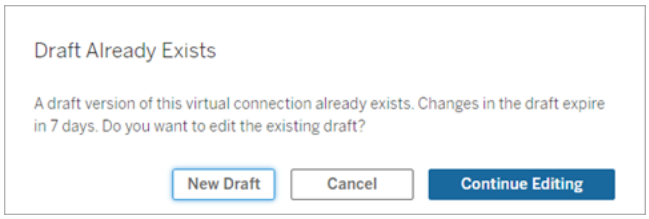
Save a draft

You can manually save a draft of the connection by clicking the save icon in the toolbar or by selecting **File** > **Save Draft** from the menu.

When editing a published virtual connection, the connection stays available to users in its current published state. You can save your updates as a draft while you work on the connection in the editor. To share the updates to the virtual connection with other users, you need to publish it.

Draft in progress

If you close the editor while updating a published virtual connection, the next time you open the connection in the editor within seven days, you have the option of continuing to make edits to the existing draft, starting a new draft, or opening the connection in its current published state by clicking **Cancel**.



To return to a draft version of an unpublished virtual connection, you need to manually save the URL of the draft **before** you close the editor. You can use the URL to open the draft in the editor the next time you want to work on the connection within seven days. For example:

```
https://yourserver.test.com/published-connection-edit-
or/?draft=d1789edc-5d9f-40ae-988d-9fc879f37a98
```

Publish the connection

To publish a new connection:

1. Click the **Publish** button in the upper right corner of the editor or select **File** > **Publish** from the menu.
2. In the Publish dialog box:
    a. Type a name in the **Name** field.
    b. Select a project to save the connection to.
3. Click **Publish**.

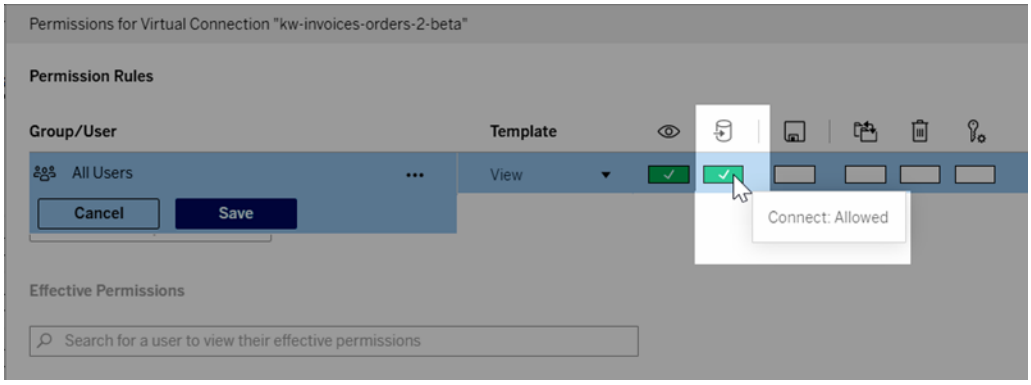
To publish an updated connection, click the **Publish** button in the upper right corner of the editor or select **File** > **Publish** from the menu.

Set permissions on a virtual connection

After you publish a virtual connection, you need to set the permissions so that others can use it. By default, all users can **View** the connection, in other words, see it listed under Virtual Connections in Tableau, but unless you set the **Connect** capability to Allowed, only you and administrators can use the virtual connection. For more information on the Connect capability, see Permissions.

To set permissions:

1. Navigate to the virtual connection.
2. Open the Actions menu (...) and click **Permissions**.
3. Check the box under the Connect icon so that connect is allowed for all users.



Tip: You can add additional rules if you want to grant the permission only to certain users or groups.

4. Click **Save**.

For more information about permissions on Tableau content, see Permissions. For information on embedding passwords when you publish Tableau content such as a data source or workbook that uses a virtual connection, see Virtual connections in the Tableau Server help.

Who can do this

To publish a virtual connection or set permissions, you must

- have credentials to the database that the virtual connection connects to, and
- be a server or site administrator, or a Creator.

Next step

After you publish a virtual connection and set its permissions, you can Use a Virtual Connection .

Schedule Extract Refreshes for a Virtual Connection

One of the benefits of virtual connections is that you can reuse the same extract multiple times, reducing data proliferation and removing redundant extract refresh jobs. To ensure that extract

data is fresh for any content that uses a virtual connection, you can create an extract refresh schedule for the tables in your connection after you publish the connection.

You can also schedule extract refreshes of data sources and workbooks that use virtual connections. See Schedule Refreshes on Tableau Cloud and Refresh Data on a Schedule (Tableau Server).

Extract tables

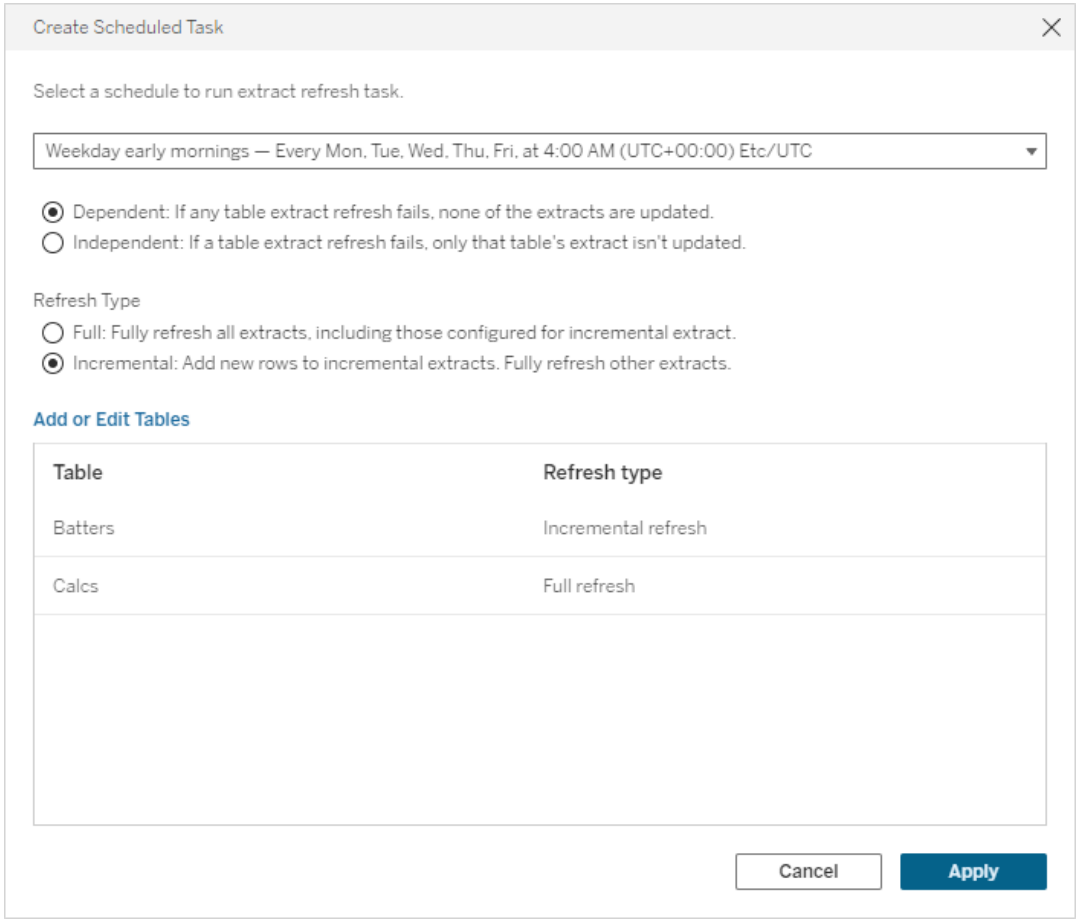
See Extract table data.

Schedule extract refreshes on Tableau Server

1. Navigate to the virtual connection page. (From the **Home** or **Explore** page, click **Virtual Connections** from the dropdown menu, then select your virtual connection.)
2. At the top of the page, a heading should say **Data is Extract**. If it says **Data is Live**, refresh your browser.
3. Select the **Scheduled Tasks** tab and click **+New Task**.
4. The **Create Scheduled Task** dialog box opens. The **Create Scheduled Task** dialog box looks different in Tableau Server and Tableau Cloud. If you're using Tableau Cloud, see Schedule Extract Refreshes for a Virtual Connection in the Tableau Cloud Product Help.
5. Select a schedule from the dropdown menu.
6. For cases when multiple tables use extracts, select **Dependent** or **Independent**. (In Tableau Server 2023.3 and earlier, these options were "Keep tables in sync" and "Refresh tables independently", but the underlying meaning is the same.)
    - **Dependent** means that none of the extracts will be updated if one or more tables' extract refresh jobs fails.
    - **Independent** means that the success or failure of a table's extract job doesn't affect whether or not other tables' extracts are updated.
7. Select **Refresh Type**. You can configure the type of extract in the virtual connection editor. For more information, see the Incremental Extracts section of the Create a Virtual Connection page. (In Tableau Server 2023.3 and earlier, virtual connections don't support incremental extracts, so you won't see these options.)
    - **Full** means that full extract refresh jobs will be run on all extracts in the virtual connection, regardless of whether they are configured for full extract refresh or incremental extract refresh.

- **Incremental** means that incremental extract refresh jobs will be run on all incremental extracts in the virtual connection. For all other extracts in the virtual connection, full extract refresh jobs will be run.

8. Select **Add or Edit Tables** and select the tables you want to refresh.
9. Select **OK**.
10. Select **Apply**.



Time limit for extract refreshes

To ensure that long running refresh tasks don't take up all system resources and don't prevent refreshes of other extracts on your site, extract refreshes for a virtual connection are subject to a two-hour time limit. For more information about the timeout limit for refresh tasks and suggestions for resolving these errors, see Time limit for extract refreshes. However note that virtual connections support only full and not incremental refreshes.

Who can do this

To publish a virtual connection or set permissions, you must

- have credentials to the database that the virtual connection connects to, and
- be a server or site administrator, or a Creator.

Next step

After you schedule extract refreshes for a virtual connection, you can Use a Virtual Connection

Use a Virtual Connection

After a virtual connection is published and permissions are set, it's available to users to connect to data in the same ways that users access all data in Tableau. When you must edit a virtual connection or the data policy in the connection—for example, when the underlying schema changes—simply open the connection in the virtual connection editor, make your changes, and either save or publish the updates. You can also replace an existing data source in a workbook with a virtual connection.

Connect to a virtual connection

For web authoring in Tableau Cloud or Tableau Server:

1. On the Home or Explore page, click **New**.
2. Select the type of content you want to create: workbook, flow, or published data source.
3. In Connect to Data > On This Site > Content Type dropdown menu, select **Virtual Connections**.
4. Select the name of the connection and click **Connect**.

For Tableau Desktop and Tableau Prep:

1. On the Connect pane, under Search for Data, click **Tableau Server**.
2. Enter the server name and click **Connect**, or click **Tableau Cloud**.
3. Enter the information prompted for.
4. On the Search for Data dialog box, from the Content Type dropdown menu, select **Vir-**

tual Connections.

5. Select the name of the connection and click **Connect**.

> **Note:** There's no need to enter credentials when you connect using a virtual connection. The credentials to access the data are embedded in the connection.

Edit a virtual connection or data policy

When editing a published virtual connection, the connection stays available to users in its current published state. For more information, see Publish a Virtual Connection and Set Permissions.

To edit a connection, navigate to it from the Explore page. Note that even though database credentials are embedded in the connection, only those with the database credentials can make any changes to a virtual connection.

1. From the dropdown menu, select **All Virtual Connections**, then select the connection you want to edit.
2. Click **Edit Virtual Connection**.
3. Enter the information prompted for to connect. To edit a connection, you must enter the credentials required to access the data.
4. Click **Sign In**.
5. In the virtual connection editor, make your changes and then either save a draft or publish the connection.

Respond to underlying schema changes

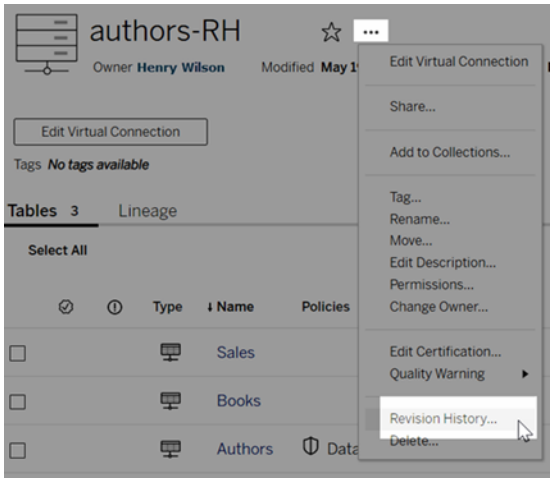
When the underlying schema in a virtual connection changes—for example, a table is added or deleted, or a column is added or renamed—you must edit the virtual connection to reflect the schema changes and then republish the connection. (If the connection has extracts, remember to refresh the extracts.) This way, you can add or edit the tables, columns, and policies in the connection before new data is exposed to everyone.

Work with virtual connection revision history

When you publish a virtual connection, a version is saved in the revision history for Tableau Cloud or Tableau Server. You can revert to a previous version at any time.

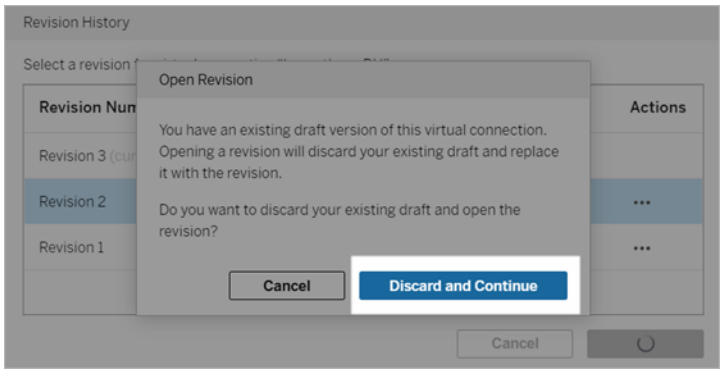
To access revision history, you must have a **Creator** site role and the **View** and **Overwrite** permissions.

To see the virtual connection revision history, click the actions menu (. . .) for the virtual connection, then click **Revision History**.

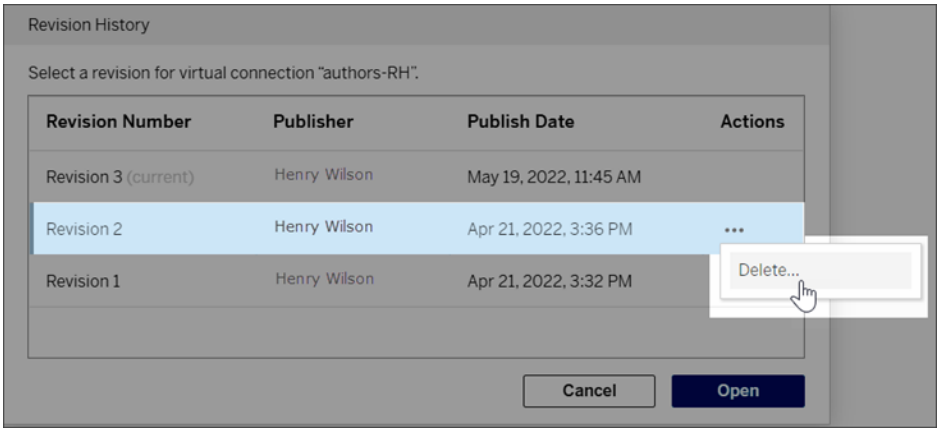


Restore or delete a virtual connection revision

To restore a virtual connection revision, select a revision, and then click **Open**. You are then prompted to discard the existing version of the connection. When you click **Discard and Continue**, the revision you selected becomes the current version of the connection.



To delete a revision, from the revision's actions menu (. . .), click **Delete**.



Replace an existing data source in a workbook with a virtual connection

For web authoring in Tableau Cloud or Tableau Server:

1. Download the workbook. For more information, see Download Views and Workbooks in Tableau Desktop Help.
2. In Tableau Desktop, open the workbook and replace its existing data source with a virtual connection. For more information, see Replace Data Sources in Tableau Desktop Help.
3. In Tableau Desktop, upload the workbook to your Tableau Cloud or Tableau Server site. For more information, see Upload Workbooks to a Tableau Site in Tableau Desktop Help.
4. In Tableau Cloud or Tableau Server, click **Publish** to save your changes to the server.

For Tableau Desktop:

1. Open the workbook and replace its existing data source with a virtual connection. For more information, see Replace Data Sources in Tableau Desktop Help.
2. Republish the workbook. For more information, see Simple Steps to Publish a Workbook in Tableau Desktop Help.

Who can do this

To use a virtual connection, you must be a Server Administrator, Site Administrator Creator, or Creator.

To edit a virtual connection or data policy, you must

- have credentials to the database that the virtual connection connects to, and
- be a server or site administrator, or a Creator.

To migrate existing content to use a virtual connection, you must

- be a server or site administrator, or
- be a Creator who is also the data source owner.