

Individuelle Zabbix Installationen in On-Premises Kubernetes Umgebungen



IntelliTrend GmbH

 www.intellitrend.de



ZABBIX
PREMIUM PARTNER



Kontakt: Wolfgang Alper

wolfgang.alper@intellitrend.de

Beginnen wir "einfach"

-

Zabbix Installation mit Docker

Zabbix Installation mit Docker

Docker Kommandozeile

```
docker run --name my-zabbix-server-mysql \  
-e DB_SERVER_HOST="my-mysql-server" \  
-e MYSQL_USER="mydbuser" \  
-e MYSQL_PASSWORD="mydbpass" \  
-d zabbix/zabbix-server-mysql:ubuntu-6.4-latest
```

Weitere optionale Umgebungsvariablen

```
...  
ZBX_LISTENPORT=10051  
ZBX_STARTREPORTWRITERS=1  
ZBX_WEBSERVICEURL=http://zabbix-web-service:10053/report  
...
```

Zabbix Installation in Kubernetes



Installationsoptionen in Kubernetes

- Installation von Zabbix (ähnlich wie mit Docker) manuell mit YAML basierenden Konfiguration Dateien unter Verwendung von „kubectl“

Oder

- Installation von Zabbix unter Verwendung von Templates, Konfigurations-Repositories und synchronisierten Konfigurationen mit “Helm”

Was ist Helm

- Helm wird häufig “The **package manager** for Kubernetes” genannt
- Helm ist ein K8s **Konfigurations-Repository** und **Template System**
- Helm bündelt verschiedene Konfigurationsdateien in einem **Release**
- Helm erlaubt konsistente Upgrades mit Versionierung
- Helm erlaubt flexible Konfigurationen über die „values.yaml“ Datei

- Offizielles Zabbix K8s **Monitoring** Helm Chart:
<https://git.zabbix.com/projects/ZT/repos/kubernetes-helm/browse>

- Community Zabbix K8s **Deployment** Helm Chart:
<https://github.com/zabbix-community/helm-zabbix>
(Ehemals repo: <https://github.com/cetic/helm-zabbix>)

Helm – Templates and Values

Helm Values

```
{{- if .Values.zabbixserver.enabled }}
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: {{ template "zabbix.fullname" . }}-zabbix-server
  labels:
    app: {{ template "zabbix.fullname" . }}-zabbix-server
    app.kubernetes.io/name: zabbix-server
    helm.sh/chart: {{ include "zabbix.chart" . }}
    app.kubernetes.io/instance: {{ .Release.Name }}-zabbix-se
    app.kubernetes.io/managed-by: {{ .Release.Service }}-zabb
spec:
  replicas: {{ .Values.zabbixserver.replicaCount }}
  serviceName: {{ template "zabbix.fullname" . }}
  selector:
    matchLabels:
      app: {{ template "zabbix.fullname" . }}-zabbix-server
  template:
    metadata:
      labels:
        app: {{ template "zabbix.fullname" . }}-zabbix-server
        app.kubernetes.io/name: zabbix-server
        helm.sh/chart: {{ include "zabbix.chart" . }}
        app.kubernetes.io/instance: {{ .Release.Name }}-zabbix
        app.kubernetes.io/managed-by: {{ .Release.Service }}-
```

```
# Default values for zabbix.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

# **Zabbix Server** configurations
zabbixserver:
  # -- Enables use of **Zabbix Server**
  enabled: true
  # -- Number of replicas of ``zabbixserver`` module
  replicaCount: 1
  # -- optional set true open a port direct on node where zabbix server runs
  hostPort: false
  # -- optional set hostIP different from 0.0.0.0 to open port only on this IP
  hostIP: 0.0.0.0
  resources: {}
  image:
    # -- Zabbix server Docker image name
    repository: zabbix/zabbix-server-pgsql
    # -- Tag of Docker image of Zabbix server
    tag: ubuntu-6.0.0
    # -- Pull policy of Docker image
    pullPolicy: IfNotPresent
    # -- List of dockerconfig secrets names to use when pulling images
    pullSecrets: []
  # -- Address of database host
  DB_SERVER_HOST: "172.20.22.100"
  # -- Port of database host
  DB_SERVER_PORT: "5432"
  # -- User of database
  POSTGRES_USER: "zabbix"
  # -- Password of database
  POSTGRES_PASSWORD: "zabbix"
```



Helm Template

ZABBIX
PREMIUM PARTNER

Installation von Zabbix in K8s unter Verwendung von Helm

Installation von Zabbix mit Helm

Installation des Zabbix Helm Repository

```
[serveradmin@k8s-master demo]$ helm repo add cetic https://cetic.github.io/helm-charts  
"cetic" has been added to your repositories
```

```
[serveradmin@k8s-master demo]$ helm repo update  
Hang tight while we grab the latest from your chart repositories...  
...Successfully got an update from the "metallb" chart repository  
...Successfully got an update from the "nfs-subdir-external-provisioner" chart repository  
...Successfully got an update from the "cetic" chart repository  
...Successfully got an update from the "prometheus-community" chart repository  
Update Complete. *Happy Helming!*
```

Erstellung der "values_zabbix.yaml" (skeleton mit Defaults)

```
[serveradmin@k8s-master demo]$ helm show values cetic/zabbix > zabbix_values.yaml
```

Installation von Zabbix mit Helm

Edit `zabbix_values.yaml`:

- Disable Postgres deployment since we use an external DB
- Set `DB_SERVER_HOST` to address of external DB
- Set username, password and database name as required

```
# **PostgreSQL** configurations
postgresql:
# -- Create a database using Postgresql
  enabled: false
  auth:
# -- Enable remote access to "postgres" user
  enablePostgresUser: true
# -- Password of "postgres" user
  postgresPassword: "zabbix_pwd"
# -- User of database
  username: "zabbix"
# Password of database
  password: "zabbix"
# -- Name of database
  database: "zabbix"
```

```
# **Zabbix Server** configurations
zabbixserver:
# -- Enables use of **Zabbix Server**
  enabled: true
# -- Number of replicas of "zabbixserver"
  replicaCount: 1
# -- optional set true open a port direct
  hostPort: false
# -- optional set hostIP
  hostIP: 0.0.0.0
  resources: {}
  image:
# Zabbix server Docker image name
  repository: zabbix/zabbix-server-pgsql
# -- Tag of Docker image of Zabbix server
  tag: ubuntu-6.0.0
# -- Pull policy of Docker image
  pullPolicy: IfNotPresent
# -- List of dockerconfig secrets
  pullSecrets: []
# -- database
  DB_SERVER_HOST: "172.20.20.100"
  DB_SERVER_PORT: "5432"
  POSTGRES_USER: "zabbix"
  POSTGRES_PASSWORD: "zabbix"
```

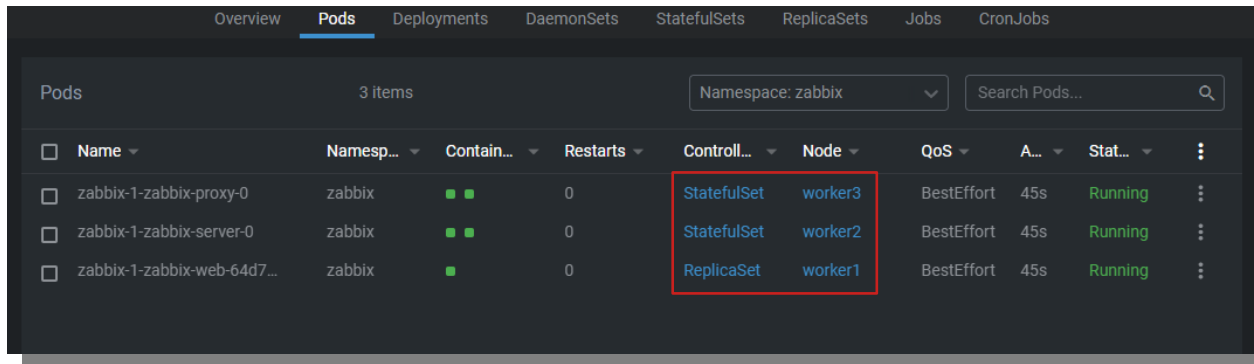


Installation von Zabbix mit Helm

Erstellung eines Helm Releases „zabbix-1“ im Namespace „zabbix“:

```
[serveradmin@k8s-master demo]$ helm install -n zabbix --create-namespace -f zabbix_values.yaml zabbix-1 cetic/zabbix
NAME: zabbix-1
LAST DEPLOYED: Thu Apr 14 10:07:32 2022
NAMESPACE: zabbix
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Get the application URL by running these commands:
  export POD_NAME=$(kubectl get pods --namespace zabbix -l "app=zabbix-1-zabbix-web" -o jsonpath="{.items[0].metadata.name}")
  export CONTAINER_PORT=$(kubectl get pod --namespace zabbix $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
  echo "Visit http://127.0.0.1:8888 to use your application"
  kubectl --namespace zabbix port-forward $POD_NAME 8888:$CONTAINER_PORT
```

Pods vom Release „zabbix-1“ nach Installation



Name	Namespace	Container	Restarts	Controller	Node	QoS	Age	Status
zabbix-1-zabbix-proxy-0	zabbix		0	StatefulSet	worker3	BestEffort	45s	Running
zabbix-1-zabbix-server-0	zabbix		0	StatefulSet	worker2	BestEffort	45s	Running
zabbix-1-zabbix-web-64d7...	zabbix		0	ReplicaSet	worker1	BestEffort	45s	Running

Start der neuen Zabbix Instanz

Zabbix Server in „zabbix-1“, Initialisierung der DB:

```
Pod zabbix-1-zabbix-server-0 X +
Namespace zabbix Owner StatefulSet zabbix-1-zabbix-server Pod zabbix-1-zabbix-... Container zabbix-server
** Preparing Zabbix server
** Using POSTGRES_USER variable from ENV
** Using POSTGRES_PASSWORD variable from ENV
*****
* DB_SERVER_HOST: 172.20.22.100
* DB_SERVER_PORT: 5432
* DB_SERVER_DBNAME: zabbix
* DB_SERVER_SCHEMA: public
*****
** Database 'zabbix' already exists. Please be careful
** Creating 'zabbix' schema in PostgreSQL
```

Zabbix Server in „zabbix-1“, Start:

```
Pod zabbix-1-zabbix-server-0 X +
Namespace zabbix Owner StatefulSet zabbix-1-zabbix-server Pod zabbix-1-zabbix-... Container zabbix-server
255:20220412:120633.263 server #34 started [trapper #3]
256:20220412:120633.265 server #35 started [trapper #4]
257:20220412:120633.267 server #36 started [trapper #5]
258:20220412:120633.268 server #37 started [icmp pinger #1]
264:20220412:120633.269 server #43 started [history poller #5]
226:20220412:120633.270 server #5 started [alerter #2]
265:20220412:120633.271 server #44 started [availability manager #1]
266:20220412:120633.271 server #45 started [trigger housekeeper #1]
267:20220412:120633.271 server #46 started [odbc poller #1]
260:20220412:120633.271 server #39 started [history poller #1]
259:20220412:120633.273 server #38 started [alert syncer #1]
263:20220412:120633.273 server #42 started [history poller #4]
261:20220412:120633.274 server #40 started [history poller #2]
262:20220412:120633.276 server #41 started [history poller #3]
239:20220412:120641.351 item "Zabbix server:zabbix[process,ipmi poller,avg,busy]" became not supported: No "ipmi poller" processes started.
239:20220412:120642.360 item "Zabbix server:zabbix[process,java poller,avg,busy]" became not supported: No "java poller" processes started.
239:20220412:120647.406 item "Zabbix server:zabbix[process,ipmi manager,avg,busy]" became not supported: No "ipmi manager" processes started.
239:20220412:120647.406 item "Zabbix server:zabbix[process,snmp trapper,avg,busy]" became not supported: No "snmp trapper" processes started.
247:20220412:120648.270 enabling Zabbix agent checks on host "Zabbix server": interface became available
239:20220412:120648.417 item "Zabbix server:zabbix[process,vmware collector,avg,busy]" became not supported: No "vmware collector" processes started.
239:20220412:120655.457 item "Zabbix server:zabbix[vmware,buffer,pused]" became not supported: No "vmware collector" processes started.
239:20220412:120658.479 item "Zabbix server:zabbix[process,report writer,avg,busy]" became not supported: No "report writer" processes started.
239:20220412:120659.485 item "Zabbix server:zabbix[process,report manager,avg,busy]" became not supported: No "report manager" processes started.
```



Zabbix Instanz – Pods

Pod „zabbix-1-zabbix-server-0“
- Container zabbix-server
- Container zabbix-agent

Containers

- zabbix-server
 - CPU Memory Filesystem
 - Metrics not available at the moment
 - Status: **running, ready**
 - Image: zabbix/zabbix-server-pgsql:ubuntu-6.0.0
 - Ports: zabbix-server_10051/TCP Forward...; zabbix-jmx_10052/TCP Forward...
 - Environment: DB_SERVER_HOST: 172.20.22.100; DB_SERVER_PORT: 5432; POSTGRES_DB: zabbix; POSTGRES_PASSWORD: zabbix; POSTGRES_USER: zabbix
 - Mounts: /var/run/secrets/kubernetes.io/servic... from default-token-js9jx (ro)
- zabbix-agent
 - CPU Memory Filesystem
 - Metrics not available at the moment
 - Status: **running, ready**
 - Image: zabbix/zabbix-agent:ubuntu-6.0.0
 - Ports: zabbix-agent_10050/TCP Forward...
 - Environment: ZBX_ACTIVESEVERERS:; ZBX_ACTIVE_ALLOW: true; ZBX_DEBUGLEVEL:; ZBX_HOSTNAME: fieldRef(v1.metadata.name); ZBX_LOADMODULE:; ZBX_PASSIVESEVERERS: 127.0.0.1; ZBX_PASSIVE_ALLOW: true; ZBX_SERVER_HOST: 127.0.0.1; ZBX_SERVER_PORT: 10051; ZBX_TIMEOUT:;
 - Mounts: /var/run/secrets/kubernetes.io/servic... from default-token-js9jx (ro)

Pod „zabbix-1-zabbix-proxy-0“
- Container zabbix-agent
- Container zabbix-proxy

Containers

- zabbix-agent
 - CPU Memory Filesystem
 - Metrics not available at the moment
 - Status: **running, ready**
 - Image: zabbix/zabbix-agent:ubuntu-6.0.0
 - Ports: zabbix-agent_10050/TCP Forward...
 - Environment: ZBX_ACTIVESEVERERS:; ZBX_ACTIVE_ALLOW: true; ZBX_DEBUGLEVEL:; ZBX_HOSTNAME: fieldRef(v1.metadata.name); ZBX_LOADMODULE:; ZBX_PASSIVESEVERERS: 127.0.0.1; ZBX_PASSIVE_ALLOW: true; ZBX_SERVER_HOST: 127.0.0.1; ZBX_SERVER_PORT: 10051; ZBX_TIMEOUT:;
 - Mounts: /var/run/secrets/kubernetes.io/servic... from default-token-js9jx (ro)
- zabbix-proxy
 - CPU Memory Filesystem
 - Metrics not available at the moment
 - Status: **running, ready**
 - Image: zabbix/zabbix-proxy-sqlite3:ubuntu-6.0.0
 - Ports: zabbix-proxy_10051/TCP Forward...
 - Environment: ZBX_DEBUGLEVEL:; ZBX_HOSTNAME: zabbix-proxy; ZBX_JAVAGATEWAY_ENABLE: 128M; ZBX_LOADMODULE:; ZBX_PROXYMODE: 0; ZBX_SERVER_HOST: zabbix-zabbix-server; ZBX_SERVER_PORT: 10051; ZBX_TIMEOUT:;



ZABBIX
PREMIUM PARTNER

Services

Erreichbarkeit außerhalb des Clusters

Services – Erreichbarkeit außerhalb des Clusters

- HTTP-basierte Dienste können über einen Reverse-Proxy wie nginx oder Traefik geroutet werden.
- Nicht-HTTP-basierte TCP-/UDP-Dienste können nicht einfach geroutet werden, sie benötigen eine eigene externe IP-Adresse

Cloud-Dienste wie AWS, Azure oder GCS bieten Load Balancer als Lösung für dieses Problem

Wie lässt sich dieses Problem in einer selbst gehosteten Cloud-Umgebung lösen?

Services – Erreichbarkeit außerhalb des Clusters

Lösung: MetalLB

- MetalLB ist ein **Layer-2-Netzwerk-Load-Balancer**, der in selbst gehosteten Cloud-Umgebungen eingesetzt werden kann
- MetalLB weist den Cluster-Diensten externe IP-Adressen zu und verlässt sich nicht auf URLs oder Reverse Proxies

```
[serveradmin@k8s-master demo]$ helm repo add metallb https://metallb.github.io/metallb  
"metallb" has been added to your repositories
```

← Add repo

```
[serveradmin@k8s-master demo]$ helm repo update
```

← Update repo

```
[serveradmin@k8s-master demo]$ helm install metallb metallb/metallb --namespace metallb \  
--create-namespace -f metallb_values.yaml --version "0.11.0"
```

← Install

```
NAME: metallb  
LAST DEPLOYED: Thu Apr 14 12:08:35 2022  
NAMESPACE: metallb  
STATUS: deployed  
REVISION: 1  
TEST SUITE: None  
NOTES: MetalLB is now running in the cluster.  
LoadBalancer Services in your cluster are now available on the IPs you  
defined in MetalLB's configuration:
```

```
config:  
  address-pools:  
  - addresses:  
    - 172.20.22.110-172.20.22.120  
    name: default  
    protocol: layer2
```

← Address pool

<https://github.com/metallb/metallb>



ZABBIX
PREMIUM PARTNER

Reconfiguration des Deployments mit MetalLB

Editiere `zabbix_values.yaml`, verwende service/type "LoadBalancer" anstatt "ClusterIP"

```
# -- Password of database
POSTGRES_PASSWORD: "zabbix"
# -- Name of database
POSTGRES_DB: "zabbix"
service:
  # -- Type of service in Kubernetes cluster
  type: LoadBalancer
  # -- Cluster IP for Zabbix server
  clusterIP:
  # -- Port of service in Kubernetes cluster
  port: 10051
  # NodePort of service on each node
  nodePort: 31051
  # -- Annotations for the zabbix-server service
  annotations: {}
  # metallb.universe.tf/address-pool: production-public-ips
# -- Extra environment variables. A list of additional environment variables.
extraEnv: {}
```



Zabbix Services via MetalLB

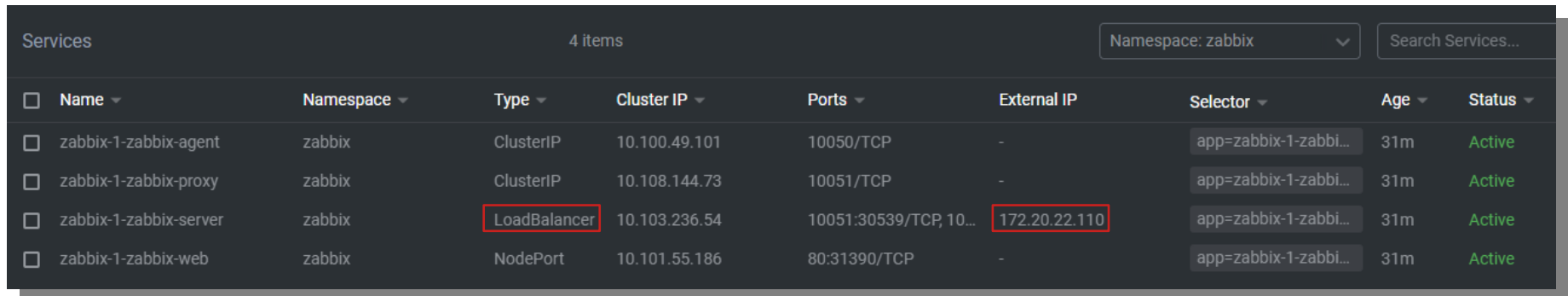
Aktualisiere das Release „zabbix-1“ nach der Änderung in `zabbix_values.yaml`

```
[serveradmin@k8s-master demo]$ helm upgrade -n zabbix -f zabbix_values.yaml zabbix-1 cetic/zabbix
Release "zabbix-1" has been upgraded. Happy Helming!
NAME: zabbix-1
LAST DEPLOYED: Tue Apr 26 15:18:37 2022
NAMESPACE: zabbix
STATUS: deployed
REVISION: 11
TEST SUITE: None
NOTES:

Get the application URL by running these commands:
  export POD_NAME=$(kubectl get pods --namespace zabbix -l "app=zabbix-1-zabbix-web" -o jsonpath="{.items[0].metadata.name}")
  export CONTAINER_PORT=$(kubectl get pod --namespace zabbix $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
  echo "Visit http://127.0.0.1:8888 to use your application"
  kubectl --namespace zabbix port-forward $POD_NAME 8888:$CONTAINER_PORT
```

Zabbix Services via MetalLB

Der Zabbix Server verwendet nun den Type „LoadBalancer“ und ist über eine externe IP-Adresse erreichbar



Name	Namespace	Type	Cluster IP	Ports	External IP	Selector	Age	Status
zabbix-1-zabbix-agent	zabbix	ClusterIP	10.100.49.101	10050/TCP	-	app=zabbix-1-zabbi...	31m	Active
zabbix-1-zabbix-proxy	zabbix	ClusterIP	10.108.144.73	10051/TCP	-	app=zabbix-1-zabbi...	31m	Active
zabbix-1-zabbix-server	zabbix	LoadBalancer	10.103.236.54	10051:30539/TCP, 10...	172.20.22.110	app=zabbix-1-zabbi...	31m	Active
zabbix-1-zabbix-web	zabbix	NodePort	10.101.55.186	80:31390/TCP	-	app=zabbix-1-zabbi...	31m	Active

Zabbix Server ist außerhalb des Clusters zugänglich:

- External IP 172.20.22.110 (zugewiesen von MetalLB)
- Port 10051

Real world challenges

Kubernetes - Real world challenges

Bei dem Wunsch, einfach Anwendungen in einem Kubernetes-Cluster bereitzustellen, können in der Praxis einige Herausforderungen auftreten:

- Nicht jeder ist mit kubectl, Helm-Charts usw. vertraut.
- Wie kann man konsistente Anwendungskonfigurationen sicherstellen?
- Wie kann man Konfigurationsänderungen verfolgen?
- Wie verwaltet man Passwörter/Geheimnisse in Konfigurationsdateien?
- Wie kann man Nutzern/Admins erlauben, eine Anwendung einfach bereitzustellen?

Wir brauchen **Bausteine** zur Schaffung eines **benutzerfreundlicheren** Ansatzes



Baustein: Flux

(Verwaltung und Anwendung von Konfigurationen)

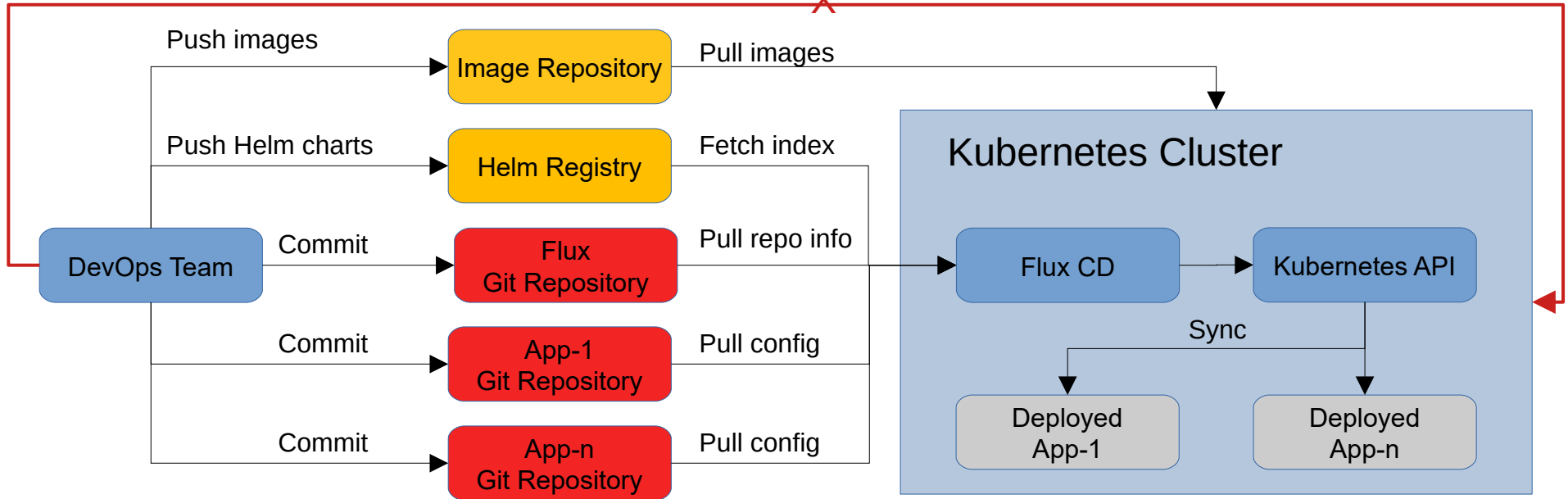
Baustein - Flux

Flux bietet eine GitOps-Lösung für Kubernetes

- Die K8-Konfigurationen werden mit Git-Repositories synchronisiert. Dies erlaubt:
 - Versionierung
 - Einfache Rollbacks
 - Konsistente Konfigurationszustände
- Flux wird im Cluster bereitgestellt und steuert den Cluster über die Kubernetes-API
- Anstelle von "kubectl apply" oder "helm install" werden Änderungen festgeschrieben und in Git-Repositories übertragen
- Flux fragt diese Git-Repositories nach Änderungen ab und führt bei Bedarf automatische Rollouts im Cluster durch

Baustein - Flux/Git

NO kubectl X NO Helm



Flux Git Repository

- Flux base configuration
- References to App repositories

App Git Repositories

- App Helm release config
- Other K8s YAML config files

In unserem Fall sind die Anwendungen einzelne Zabbix-Instanzen

Baustein: SOPS

(Verwaltung and Verwendung von “Secrets”)

Warum SOPS

- Anwendungen erfordern Anmeldeinformationen (Datenbanken, API Tokens, E-Mail usw.), die in Konfigurationsdateien für GitOps gespeichert sind.
- Allerdings: Anmeldedaten im Klartext in Git sind keine gute Idee!
- Verschlüsselung verwenden?
 - Ja: Aber, eingesetzte Anwendungen müssen weiterhin in der Lage sein, Anmeldedaten zu lesen
- Wohin mit den Schlüsseln für die verschlüsselten Anmeldeinformationen?
 - Huhn-und-Ei-Problem
- Mehrere Benutzer müssen möglicherweise Anmeldedaten bearbeiten
 - Wie können die Schlüssel sicher geteilt werden?



Baustein - SOPS

Mozilla **SOPS** (**S**ecrets **OP**eration**S**) verwaltet verschlüsselte Informationen in Textdateien

- Sensible Werte werden verschlüsselt und in spezielle Base64-Strings kodiert
- Verwendung von asymmetrischer Kryptographie mit **einem oder mehreren** öffentlichen/privaten Schlüsselpaaren verschiedener Benutzer
- Flux erhält den privaten Schlüssel und dekodiert verschlüsselte YAML-Dateien “on the fly”
- Nur Besitzer der privaten Schlüssel können die verschlüsselten Werte einsehen und bearbeiten
- Support für **Flux**, Hashicorp Vault, AWS KMS, Azure Key Vault, GCP KMS, Age, PGP etc.

```
myapp1: ENC[AES256_GCM,data:Tr7o=,iv:1=,aad:No=,tag:k=]
app2:
  db:
    user: ENC[AES256_GCM,data:CwE401s=,iv:2k=,aad:o=,tag:w=]
    password: ENC[AES256_GCM,data:p673w=,iv:YY=,aad:UQ=,tag:A=]
    # private key for secret operations in app2
    key: |-
      ENC[AES256_GCM,data:Ea3kL505U8=,iv:DM=,aad:FKA=,tag:EA=]
  an_array:
    - ENC[AES256_GCM,data:v8jQ=,iv:HBE=,aad:21c=,tag:gA=]
    - ENC[AES256_GCM,data:X10=,iv:o8=,aad:CQ=,tag:Hw=]
    - ENC[AES256_GCM,data:KN=,iv:160=,aad:fI4=,tag:tNw=]
  sops:
    kms:
      - created_at: 1441570389.775376
        enc: CiC...Pm1Hm
        arn: arn:aws:kms:us-east-1:656532927350:key/920aff2e-c5f1-4040-943a-047fa387b27e
      - created_at: 1441570391.925734
        enc: Ci...awNx
        arn: arn:aws:kms:ap-southeast-1:656532927350:key/9006a8aa-0fa6-4c14-930e-a2dfb916de1d
    pgp:
      - fp: 85D77543B3D624B63CEA9E6DBC17301B491B3F21
        created_at: 1441570391.930042
        enc: |
          -----BEGIN PGP MESSAGE-----
          hQIMA0t4uZHf19qgAQ//UvGAWGePyHuf2/zayWc1oGaDs0MzI+zw6CmXvMRNPUSa
          ...oJgS
          -----END PGP MESSAGE-----
```

See <https://github.com/mozilla/sops>



SOPS – Einfaches Beispiel

```
serveradmin@dev-k8s-master:~/sops$ cat credentials.yaml
username: "Admin"
password: "Very secret!"
serveradmin@dev-k8s-master:~/sops$ sops -e credentials.yaml
[PGP] WARN[0000] Deprecation Warning: GPG key fetching from a keyserver within sops will be removed in a future version of sops.
username: ENC[AES256_GCM,data:amp5nwE=,iv:hVVYu22D4m/0HAccJ+1pY5wgp0dGY13NG4LjK6ir0zk=,tag:UTTRhSgW4wBm79boafAxQ==,type:str]
password: ENC[AES256_GCM,data:+Fd9G1guMhB8o13s,iv:Yd47/6YPKNJwWl17sso7nymAeanQ3hiERXLw4Nrx/ts=,tag:cksv9EZBGw2Fn9zNrWG8Pw==,type:st]
sops:
  kms: []
  gcp_kms: []
  azure_kv: []
  hc_vault: []
  age: []
  lastmodified: "2022-04-26T14:50:21Z"
  mac: ENC[AES256_GCM,data:9zI2f6pIPH22+nSgN0rJphiTrv10Gx/RG4grDzbC+NdjQpV06Te4jzC+XA71aJXj7wEInk4YLgq7yjhPaFoMv1yVn415cPb11EA2dR]
  pgp:
    - created_at: "2022-04-26T14:50:20Z"
      enc: |
        -----BEGIN PGP MESSAGE-----

        hQEMAyUpShfNkFB/AQf/U16Wq+X6o00rI8b/Dpremc5Mcha/6Bz43V28PIhqS1IC
        xLPkR0yqXU4Bh0zp11bLsviDLz+msAny713T91y9th5/VOSbwj6mhzHc2CvTAL7g
        8CuXcAhGk70vTh8DH7N/L3zE5514wLJ0+gnbn7CroxpoVPLbliite07f3zIYBdv
        xGDBab0jUua/Y7Z/w0456IA6RmpB9vEe9WR1GE5iItt8a/BAsi9pWmg9vXN3UPUD
        Ub4pWl+c3REqmxShVzuz0sDQLCerBqk3F4PTIXozCuWkVz44kKND4WIcgZC3dFFR
        Ot/JVkpDhHYjt0c1Q4AsIUayXvziKbwxNcTgN/ed9JeAYPKTdtGFUthRzro3rk9
        5jAtpDSV88EcXqLk36oYT/IU+bzPaFyWA8Gj+R5WPyPhXyWqNeLNVajwoPc4oVG
        aJ+hvthvLd/JNvpPaLYxvAmfTxdfoP4Zz7Cupq9dw==
        =j9Xw
        -----END PGP MESSAGE-----

      fp: FBC7B9E2A4F9289AC0C1D4843D16CEE4A27381B4
    unencrypted_suffix: _unencrypted
    version: 3.7.1
serveradmin@dev-k8s-master:~/sops$
```

—▶ Plaintext input file

—▶ Encrypted master key to decrypt credentials



SOPS ermöglicht die Verwaltung und Speicherung von "Secrets" in Text Dateien

SOPS – Komplexes Beispiel

```
# data collection
StartPingers: 5
StartPollers: 100
# database section
DBName: ENC[AES256_GCM,data:hOPj6ocNN723iOEHWHLJzw==,iv:fYfp9eVMB1NyhasoLSfRI7nfwplhiUoJY37ydLbKy5k=,tag:SXvjSKQofnUahtTrdcHPBw==,type:str]
DBUser: ENC[AES256_GCM,data:22v5h0ouL04=,iv:VDq/MptROK/4E7nlBnzKzCjvS3iU19yzjJgN0mwS2Ns=,tag:z6E+7gnjBsTG+H4cGfW3Gw==,type:str]
DBPassword: ENC[AES256_GCM,data:+1jE8VGD/94JNm2oEEDjIB81,iv:1V1TzaKlc1I4xRkG1PHQLfEfafAh0Tkh3PIxxyTtk5do=,tag:xYUO0dYUW3cBrrzNpM7LRg==,type:str]
sops:
  kms: []
  gcp_kms: []
  azure_kv: []
  hc_vault: []
  age:
    - recipient: age1tqltjgmzlaw7eek27sdveygt6w2cer3zad6sn3eqtrjfrvrs5dtqwwq079
      enc: |
        -----BEGIN AGE ENCRYPTED FILE-----
        YWdlLWVuY3J5cHRpb24ub3JnL3Yxci0+IFgyNTUxOSB0SjJtazhYnkJJQXGvaTZO
        V3FZbTduNDFOUVFEYlNvZ1VqS3ZmMlE3RldBCj15Tm50N3BGbVAREx1wT2t0S3Vl
        dEUGWHJ1WU1ZY2UxRlRXT0gzRFNza0EKLs0tIHRpK1Q5UHprV3FLcGpjZWpGSzZk
        Ykt0aXNkb00zUWREZFZTFWFB5NWYvMVk0pHn13XD4Fy6xvD98Iq3HaSsnKUoEeTG
        d+iSaktouuXgB2lBie2ZW3d9Lb8xmXBAiWLvvoXqbcvE1S/Gk0/Lfhg==
        -----END AGE ENCRYPTED FILE-----
    - recipient: age1r3lnjrmymk2kawe59vwttn37czvgxmnd0eypfhrpcgck5fgxrufqnx0nn
      enc: |
        -----BEGIN AGE ENCRYPTED FILE-----
        YWdlLWVuY3J5cHRpb24ub3JnL3Yxci0+IFgyNTUxOSB0SjJtazhYnkJJQXGvaTZO
        V3FZbTduNDFOUVFEYlNvZ1VqS3ZmMlE3RldBCj15Tm50N3BGbVAREx1wT2t0S3Vl
        dEUGWHJ1WU1ZY2UxRlRXT0gzRFNza0EKLs0tIHRpK1Q5UHprV3FLcGpjZWpGSzZk
        Ykt0aXNkb00zUWREZFZTFWFB5NWYvMVk0pHn13XD4Fy6xvD98Iq3HaSsnKUoEeTG
        d+iSaktouuXgB2lBie2ZW3d9Lb8xmXBAiWLvvoXqbcvE1S/Gk0/Lfhg==
        -----END AGE ENCRYPTED FILE-----
  lastmodified: "2023-08-30T11z:59:52Z"
  mac: ENC[AES256_GCM,data:z/liUEIQreDvKGWmqbgSRODCxDsUlFkxPbcjFUIh3OpYHnbGx7OrR/zvyDqsaT+uLzRJCMItyxTKUikPfePjisJrlBWm0Dq8Ib9nB/
  sPMQ3QiBzUxAmi46QzV4TYhdNrgAlE/A0g+NDGvOpIn2cQBE/iHagH9k89vtgK0d3gzc=,iv:bHVR0t9ftg4Q8mrE6EzRSpLjwiwxt7Lxg4asf6oiNyI=,tag:bgCn8CxcS+KYAP8PBMLRZQ==,type:str]
  pgp: []
  encrypted_regex: ^DB.+
  version: 3.7.3
```

Left unencrypted

Encrypted by SOPS

Two different recipients with their own public keys and encrypted masterkeys



Verschiedene Benutzer/Anwendungen können die Datei mit Ihrem Key verwalten

Baustein: **Cloud Deploy**

(Verwaltung von nativen Cloud-Diensten)

Baustein Cloud Deploy

Fragen wir **ChatGPT**: "Was ist Cloud Deploy?"

Baustein Cloud Deploy

Antwort von ChatGPT:

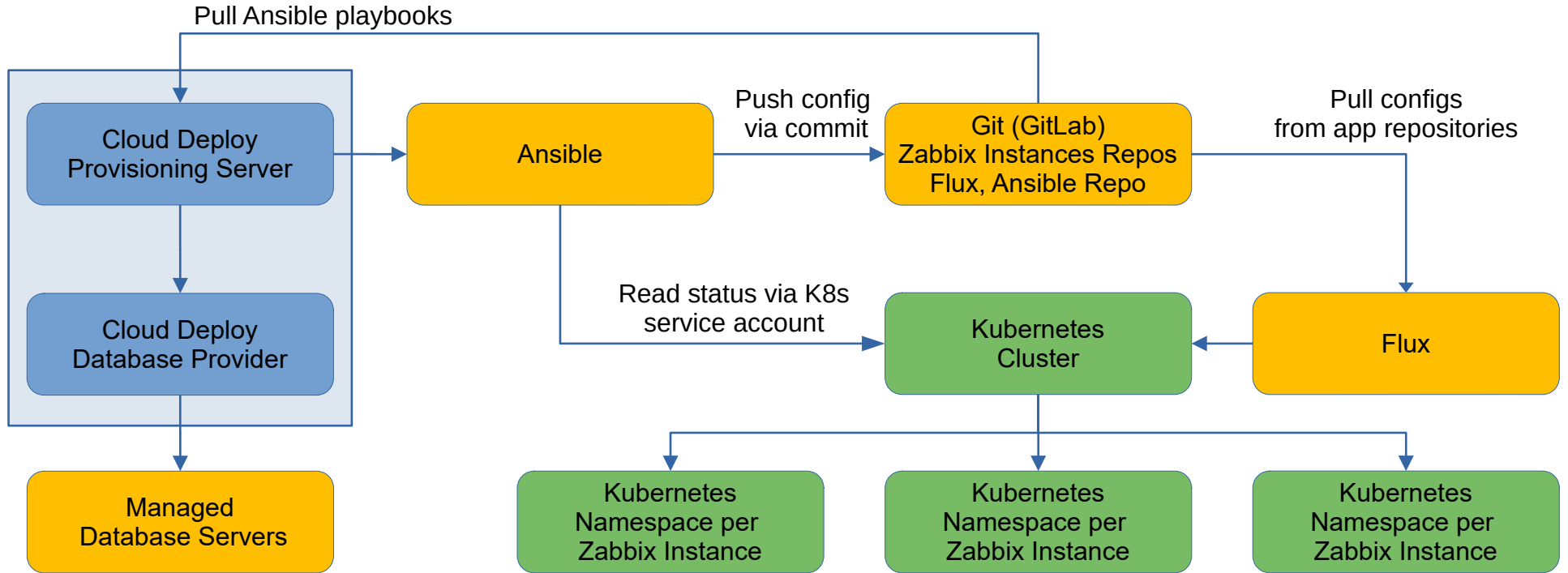


“Cloud-Deploy” ist kein spezifischer Begriff oder eine Technologie, sondern ein weit gefasster Begriff, der sich auf den Prozess der Bereitstellung von Anwendungen oder Diensten auf einer Cloud-Infrastruktur bezieht.

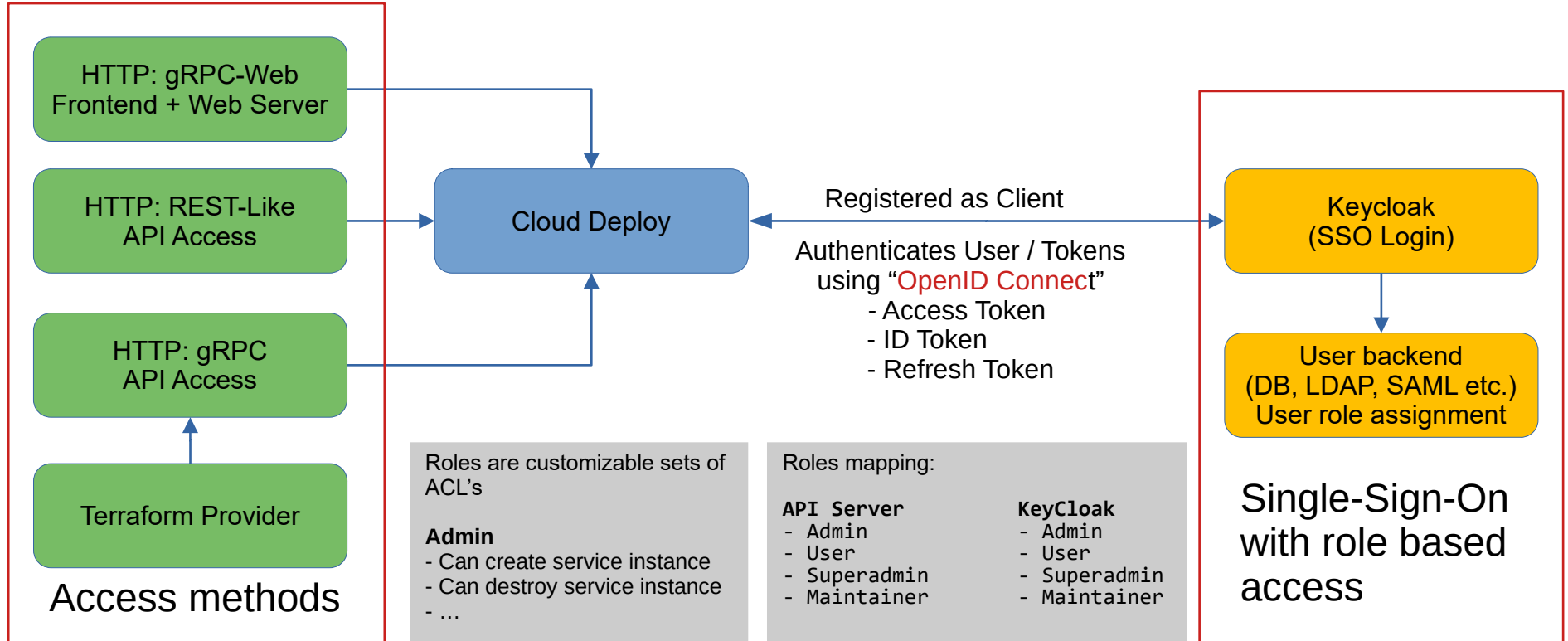
Bei der Cloud-Bereitstellung wird Ihre Softwareanwendung oder Ihr System mit Hilfe von Cloud-Ressourcen und -Diensten über das Internet / lokale Netz für Benutzer oder Kunden verfügbar gemacht.

Cloud Deploy Interna

Cloud Deploy – Provisioning Design

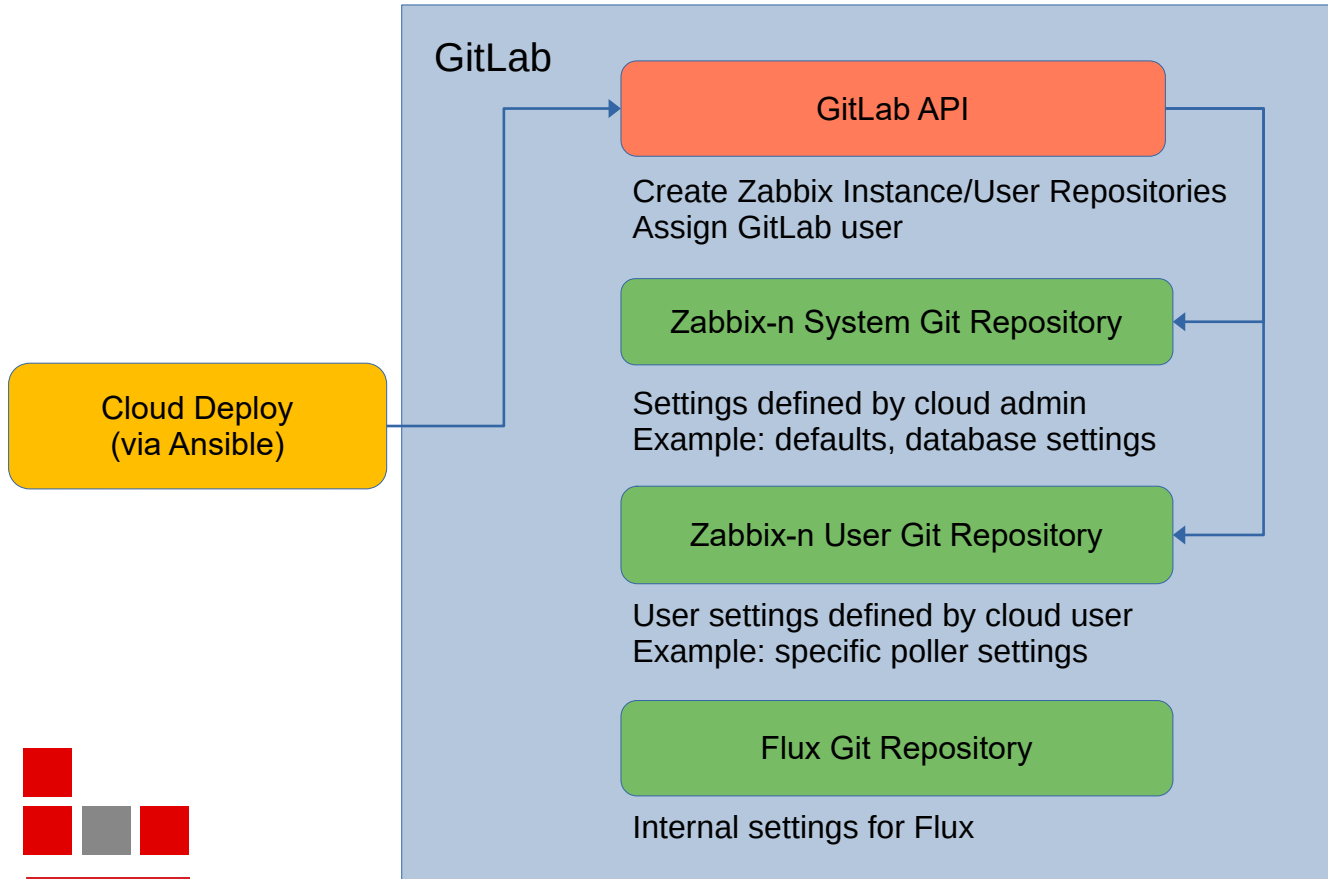


Cloud Deploy – Auth Design und API Zugriff



In unserem Fall sind Dienste einzelne Zabbix-Instanzen (einschließlich Frontend)

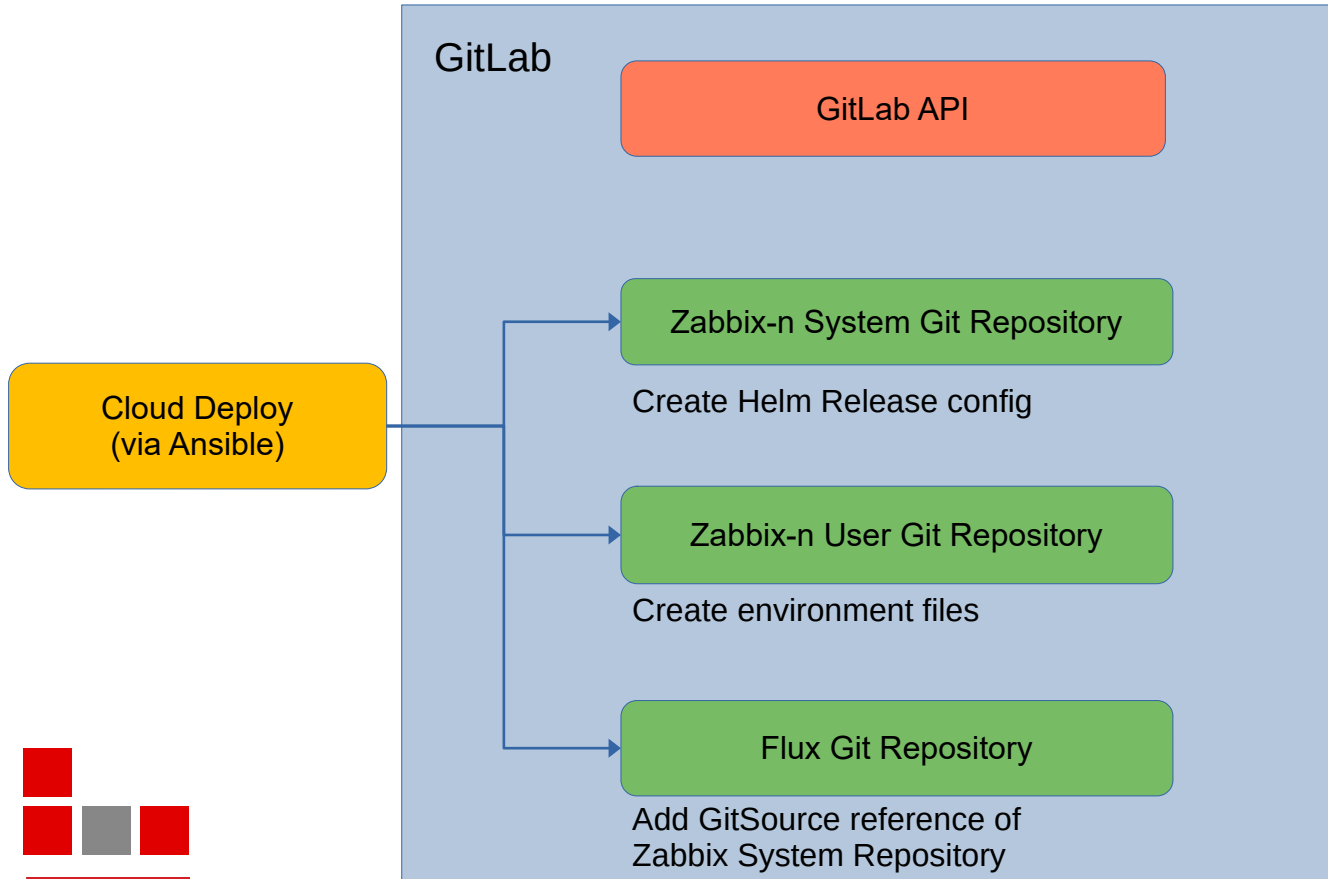
Cloud Deploy - Git Setup und Workflow - Setup



Cloud Deploy erstellt einen eigenen Satz Git-Repositories für jede Zabbix-Installation

- System Git repository
- User Git repository

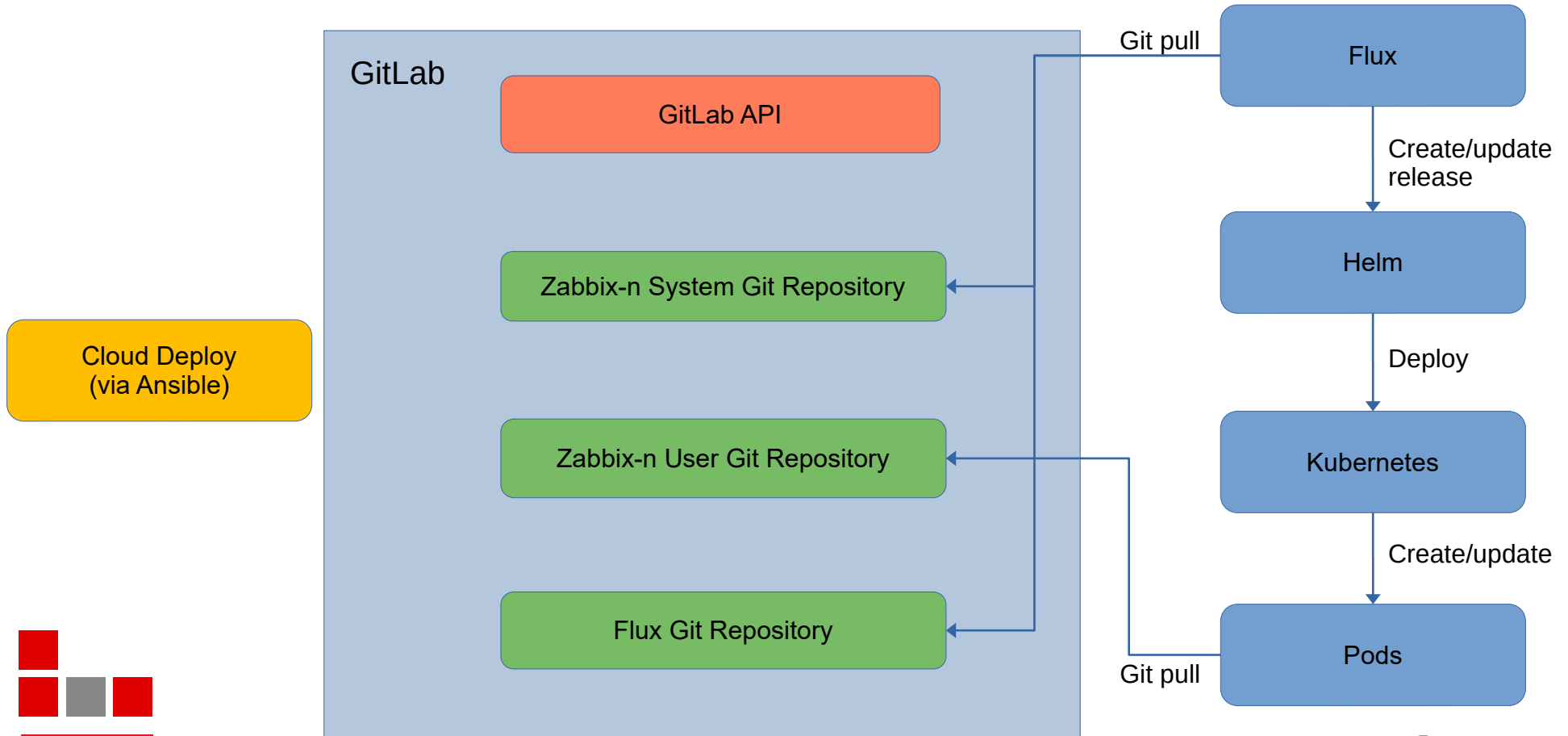
Cloud Deploy - Git Setup und Workflow - Config



Cloud Deploy überträgt erstellte Dateien und Konfigurationen automatisch an Git

- System Git repository
- User Git repository
- Flux Git repository



Cloud Deploy - Git Setup und Workflow - Deploy









Cloud Deploy – Erstellte Flux Repository Dateien

Zabbix > Flux

master flux / zabbix / my-zabbix / source.yaml Find file Blame History Permalink

 Factory 'my-zabbix' install
Zabbix Provider Bot authored 1 month ago 9ec2d3f4 

 source.yaml  305 bytes Open in Web IDE  Replace Delete   

```
1 apiVersion: source.toolkit.fluxcd.io/v1
2 kind: GitRepository
3 metadata:
4   name: my-zabbix-sysrepo
5   namespace: flux-system
6 spec:
7   interval: 1m0s
8   ref:
9     branch: master
10  secretRef:
11    name: zabbix-selfprov-git-auth
12  url: ssh://git@gitlab.loc:8022/zbx/factories/my-zabbix-system.git
```


App-Repo →





Config: Register the Zabbix System Git repository in Flux

Cloud Deploy – Erstellte Flux Repository Dateien

Zabbix > Flux

master flux / zabbix / my-zabbix / sync.yaml Find file Blame History Permalink

 Factory 'my-zabbix' install
Zabbix Provider Bot authored 1 month ago 9ec2d3f4

 sync.yaml 356 bytes Open in Web IDE Replace Delete   

```
1 apiVersion: kustomize.toolkit.fluxcd.io/v1
2 kind: Kustomization
3 metadata:
4   name: my-zabbix-kustomization
5   namespace: flux-system
6 spec:
7   decryption:
8     provider: sops
9     secretRef:
10      name: zabbix-selfprov-sops-secret
11   interval: 1m0s
12   path: ./
13   prune: true
14   serviceAccountName: ''
15   sourceRef:
16     kind: GitRepository
17     name: my-zabbix-sysrepo
```

SOPS →

Config: Kustomization Object um die Flux-Konfiguration aus dem zuvor registrierten Zabbix-Repository zu lesen

Cloud Deploy - Zabbix Instanz Repositories

The screenshot displays the Zabbix Cloud Deploy interface. At the top, the group name 'Zabbix' is shown with a lock icon, 'Group ID: 3', and a 'Leave group' link. Action buttons include 'New subgroup' and 'New project'. Below this, there are tabs for 'Subgroups and projects', 'Shared projects', and 'Archived projects'. A search bar is present with the text 'Search by name' and a dropdown menu set to 'Name'. The main content area shows a subgroup named 'Factories' with an 'Owner' tag. Underneath, two subgroups are listed: 'My Zabbix' and 'My Zabbix - System', both with a lock icon, 0 stars, and a creation date of '1 month ago'. A red box highlights these two subgroups.

Subgroup Name	Lock	Stars	Created
My Zabbix	🔒	★ 0	1 month ago
My Zabbix - System	🔒	★ 0	1 month ago

Neue Repositories, die von Cloud Deploy über Ansible für jede neue Zabbix-Instanz erstellt werden

Cloud Deploy - Zabbix System Repository

```
Zabbix > Factories > My Zabbix - System  
  
master my-zabbix-system / zabbix.yaml  
  
Added load balancer IP address  
Zabbix Provider Bot authored 1 month ago  
  
zabbix.yaml 4.11 KiB  
1 apiVersion: helm.toolkit.fluxcd.io/v2beta1  
2 kind: HelmRelease  
3 metadata:  
4   name: my-zabbix  
5   namespace: flux-system  
6 spec:  
7   chart:  
8     spec:  
9     chart: ./  
10    sourceRef:  
11      kind: GitRepository  
12      name: zabbix-helm-chart  
13      version: 1.0.1  
14    interval: 1m0s  
15    releaseName: my-zabbix  
16    serviceAccountName: ""  
17    targetNamespace: my-zabbix  
18    values:
```

SOPS

```
values:  
  postgresql:  
    database: zabbix  
    host: 172.20.22.50  
    password: ENC[AES256 GCM,data:NomliZ/1k9xbYw==,iv:mqG41142+uFdQZwwxP0i0m2dwRnJh19mnefMe1LE8BI=,tag:3  
    port: "5431"  
    username: postgres  
  secretMount:  
    - defaultMode: "0444"  
      mountPath: /tmp/git-config  
      name: git-auth-config  
  zabbixAgent:  
    env:  
      GIT_CONFIG_REPO: ssh://git@gitlab.loc:8022/zbx/factories/my-zabbix.git  
      GIT_CONFIG_UPDATE: "1689929585"  
    image:  
      pullPolicy: Always  
      pullSecrets:  
        - regcred  
      registry: registry.loc  
      repository: zabbix/zabbix-agent2  
      tag: alpine-6.0.17  
  zabbixServer:  
    env:  
      GIT_CONFIG_REPO: ssh://git@gitlab.loc:8022/zbx/factories/my-zabbix.git  
      GIT_CONFIG_UPDATE: "1689929585"  
    image:  
      pullPolicy: Always  
      pullSecrets:  
        - regcred  
      registry: registry.loc  
      repository: zabbix/zabbix-server-pgsql  
      tag: alpine-6.0.17  
    service:  
      loadBalancerIP: 172.20.22.65  
      type: LoadBalancer  
  zabbixWeb:
```

Von Cloud Deploy erstellte Helm Release-Konfiguration
Enthält die Werte für den Helm-Chart



Cloud Deploy - Zabbix User Repository

Zabbix > Factories > My Zabbix

M My Zabbix Project ID: 415

1 Commit 1 Branch 0 Tags 266 KB Files 266 KB Storage

master my-zabbix / + History Find file Web IDE Clone

Factory 'my-zabbix' install
Zabbix Provider Bot authored 1 month ago f172be3b

Upload File README Auto DevOps enabled Add LICENSE Add CHANGELOG Add CONTRIBUTING

Add Kubernetes cluster Configure Integrations

Name	Last commit	Last update
zabbix-agent	Factory 'my-zabbix' install	1 month ago
zabbix-server	Factory 'my-zabbix' install	1 month ago
zabbix-web	Factory 'my-zabbix' install	1 month ago
README.md	Factory 'my-zabbix' install	1 month ago

README.md

Zabbix User Configuration

Zabbix > Factories > My Zabbix

master my-zabbix / zabbix-web / .env

Factory 'my-zabbix' install
Zabbix Provider Bot authored 1 month ago

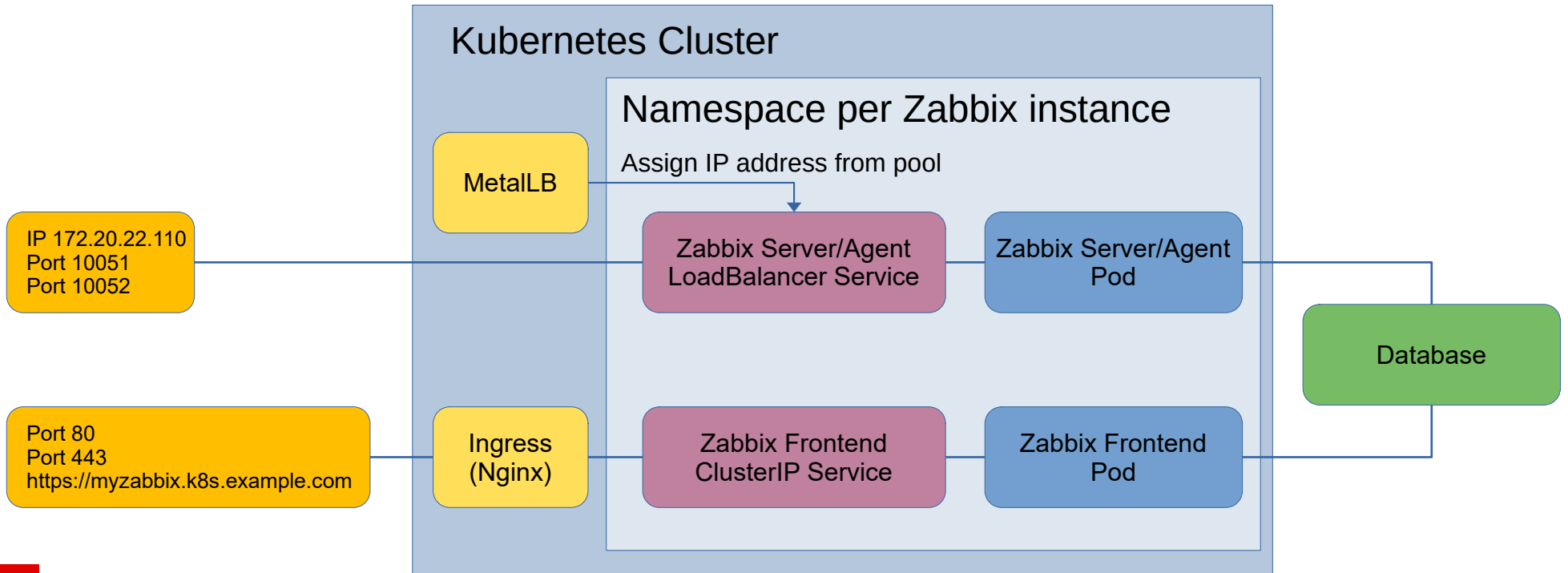
.env 976 bytes

```
1 # ZBX_SERVER_HOST=zabbix-server
2 # ZBX_SERVER_PORT=10051
3 ZBX_SERVER_NAME=My Zabbix
4 # ZBX_DB_ENCRYPTION=true # Available since 5.0.0
5 # ZBX_DB_KEY_FILE=/run/secrets/client-key.pem # Available since 5.0.0
6 # ZBX_DB_CERT_FILE=/run/secrets/client-cert.pem # Available since 5.0.0
7 # ZBX_DB_CA_FILE=/run/secrets/root-ca.pem # Available since 5.0.0
8 # ZBX_DB_VERIFY_HOST=false # Available since 5.0.0
9 # ZBX_DB_CIPHER_LIST= # Available since 5.0.0
10 # ZBX_VAULTDBPATH=
11 # ZBX_VAULTURL=https://127.0.0.1:8200
12 # ZBX_HISTORYSTORAGEURL=http://elasticsearch:9200/ # Available since 3.4.5
13 # ZBX_HISTORYSTORAGETYPES=['uint', 'dbl', 'str', 'text', 'log'] # Available since 3.4.5
14 # ZBX_SSO_SETTINGS=[] # Available since 5.0.0
15 # ZBX_MAXEXECUTIONTIME=600
16 # ZBX_MEMORYLIMIT=128M
17 # ZBX_POSTMAXSIZE=16M
18 # ZBX_UPLOADMAXFILESIZE=2M
19 # ZBX_MAXINPUTTIME=300
20 # ZBX_SESSION_NAME=zbx_sessionid
21 # ZBX_DENY_GUI_ACCESS=false
22 # ZBX_GUI_ACCESS_IP_RANGE=['127.0.0.1']
23 # ZBX_GUI_WARNING_MSG=Zabbix is under maintenance.
```

Das Benutzer-Repository enthält Umgebungsdateien mit den benutzerspezifischen Konfigurationen



Cloud Deploy - Cluster-Einrichtung nach Abgleich

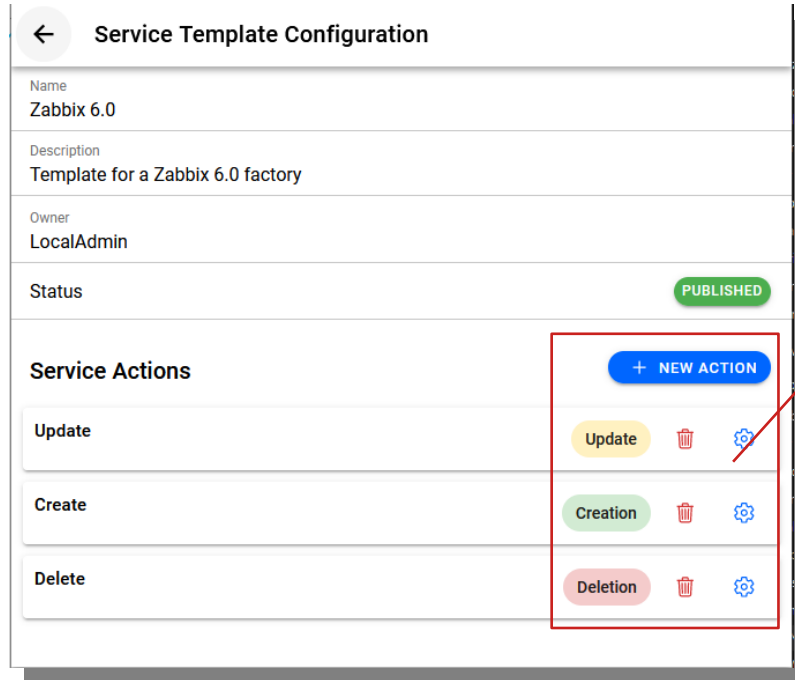


Cloud Deploy User Portal

Cloud Deploy - User Portal

- Webbasiertes Portal ermöglicht die Bereitstellung neuer Dienste in Kubernetes (hier Zabbix)
- **Services** basieren auf **Service Templates**
- Service Templates definieren **Provisioning Actions** zum Erstellen, Aktualisieren, Löschen und Migrieren eines Services
- Jede **Provisioning Action** kann mehrere **Tasks haben**
- Unterstützt SSO and RBAC

Cloud Deploy – Service Templates



Aktionen werden Service spezifisch definiert und stehen dem Benutzer für die Provisionierung zur Verfügung

Cloud Deploy Template für ein Zabbix 6.0 Deployment

Cloud Deploy – Service Templates / Tasks

← Service Template Configuration

Action Type
3

Name
Update

Description

Tasks + NEW TASK

- Playbook update_factory_config.pb.yaml
Description: Update factory config
- Playbook update_send_report.pb.yaml
Description: Send report

← Service Template Configuration

Action Type
0

Name
Create

Description

Tasks + NEW TASK

- Playbook create_validate.pb.yaml
Description: Validate input
- Playbook create_database.pb.yaml
Description: Assign database
- Playbook create_factory_projects.pb.yaml
Description: Create GitLab projects
- Playbook create_factory_config.pb.yaml
Description: Create Factory config
- Playbook create_zabbix_setup.pb.yaml
Description: Setup Zabbix
- Playbook create_factory_ip.pb.yaml
Description: Save load balancer IP address
- Playbook create_send_report.pb.yaml
Description: Send report

← Service Template Configuration

Action Type
1

Name
Delete

Description

Tasks + NEW TASK

- Playbook delete_factory_config.pb.yaml
Description: Delete factory config
- Playbook delete_factory_projects.pb.yaml
Description: Delete GitLab projects
- Playbook delete_send_report.pb.yaml
Description: Send report

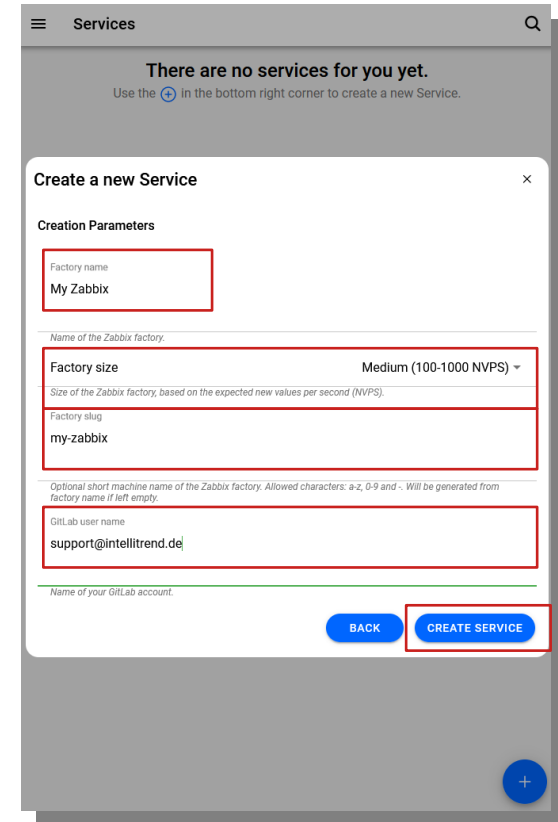
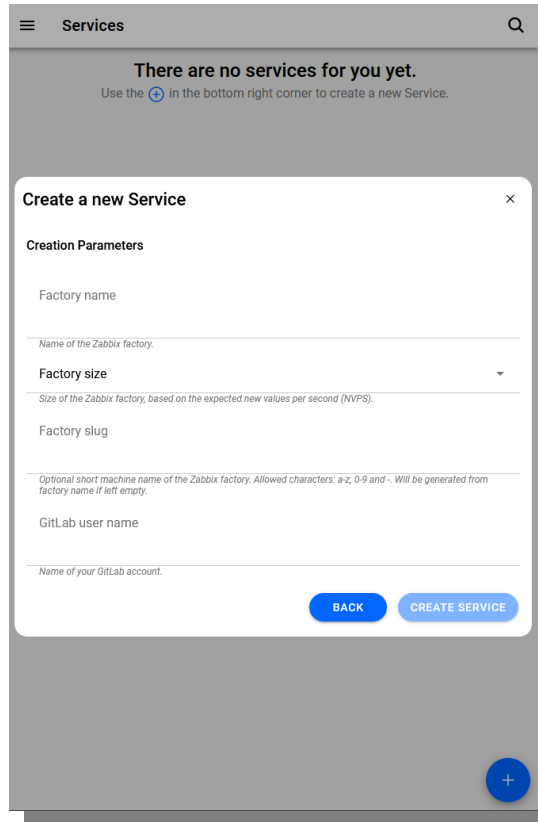
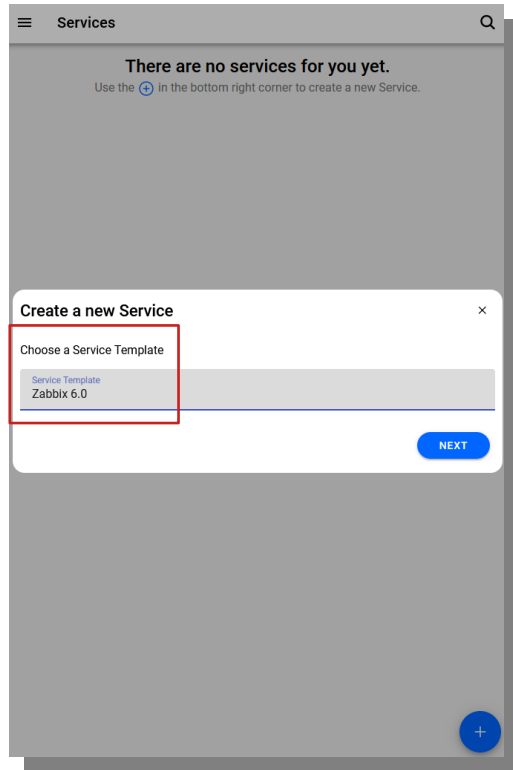
Service Konfiguration für:
Aktualisieren, Erstellen
und Löschen eines Services

Ansible Playbook mit Task
spezifischer Konfiguration



ZABBIX
PREMIUM PARTNER

Cloud Deploy – Erstellung eines neuen Service



Cloud Deploy – Job Status in Real-Time

Cloud-Deploy

- Dashboard
- Administration
 - Users
 - Roles
- Provisioning
 - Services
 - Service Templates
 - Playbooks
 - Resources
 - Jobs**
- Profile
- About
- Sign out

Job Details

Service Action ID: 0

Service: Zabbix 6.02023-07-21 08:52:28.207917982 +0000 UTC m=+3190938.651043126

Name: Zabbix 6.0: Create (CREATION)

Creation Date: July 21st 2023, 10:52:28 am

Status: In progress

Job executed by:

Expected Steps: 7

Completed Steps: 0

Events

- July 21st 2023, 10:52:28 am
 - Type: Information
 - Event: begin job

Job Details

Service Action ID: 0

Service: Zabbix 6.02023-07-21 08:52:28.207917982 +0000 UTC m=+3190938.651043126

Name: Zabbix 6.0: Create (CREATION)

Creation Date: July 21st 2023, 10:52:28 am

Status: In progress

Job executed by:

Expected Steps: 7

Completed Steps: 3

Events

- July 21st 2023, 10:53:02 am
 - Type: Step completed
 - Event: playbook create_factory_projects.pb.yaml completed
- July 21st 2023, 10:52:46 am
 - Type: Step completed
 - Event: playbook create_database.pb.yaml completed
- July 21st 2023, 10:52:37 am
 - Type: Step completed
 - Event: playbook create_validate.pb.yaml completed
- July 21st 2023, 10:52:28 am
 - Type: Information
 - Event: begin job



Cloud Deploy – Deployment abgeschlossen

Job Details

Service Action ID: 0

Service: Zabbix 6.02023-07-21 08:52:28.207917982 +0000 UTC m+=+3190938.651043126

Name: Zabbix 6.0: Create (CREATION)

Creation Date: July 21st 2023, 10:52:28 am

Status: Success

Job executed by:

Expected Steps: 7

Completed Steps: 7

Events

- July 21st 2023, 11:00:40 am
Type: Information
Event: job completed
- July 21st 2023, 11:00:40 am
Type: Step completed
Event: playbook create_send_report.pb.yaml completed
- July 21st 2023, 11:00:33 am
Type: Step completed
Event: playbook create_factory_ip.pb.yaml completed
- July 21st 2023, 11:00:15 am
Type: Step completed
Event: playbook create_zabbix_setup.pb.yaml completed

Job successful

Name	Namespace	Containers	Restarts	Controlled By	Node	QoS	Age	Status
my-zabbix-zabbix-server-67684bcf98...	my-zabbix	3	0	ReplicaSet	dev-k8s-node01	BestEffort	94m	Running
my-zabbix-zabbix-web-5898d7f4b6-8...	my-zabbix	1	0	ReplicaSet	dev-k8s-node02	BestEffort	94m	Running

Pods erstellt im K8s Cluster

ZABBIX Global view

System information

Parameter	Value	Details
Zabbix server is running	Yes	my-zabbix-zabbix-server:10051
Number of hosts (enabled/disabled)	1 / 0	
Number of templates	325	
Number of items (enabled/disabled/not supported)	120 / 112 / 0 / 8	
Number of triggers	72 / 0 / 0 / 72	

Health status: 1 Available, 0 Not available, 0 Unknown, 1 Total

Alerts: 0 Disaster, 0 High, 0 Average, 0 Warning, 0 Information, 0 Not classified

Problems: No data found.

Zabbix Frontend nach Login



ZABBIX
PREMIUM PARTNER

Cloud Deploy APIs

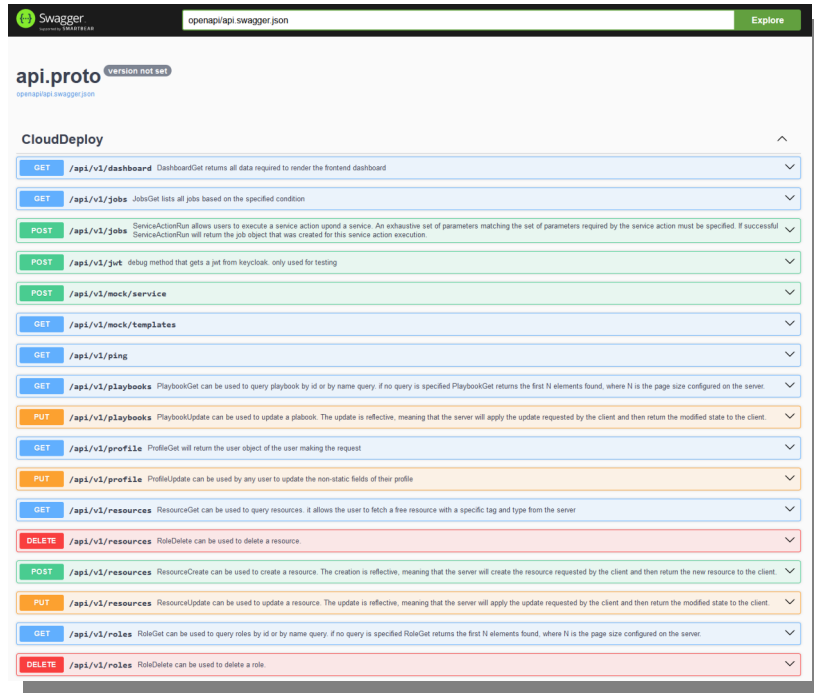
Cloud Deploy - APIs

Alle Vorgänge, die im Benutzerportal ausgeführt werden (und mehr), können auch über APIs ausgeführt werden, was die **automatische Bereitstellung** signifikant vereinfacht.

Available **APIs**:

- REST-Like HTTP API
- gRPC HTTP API
- Terraform

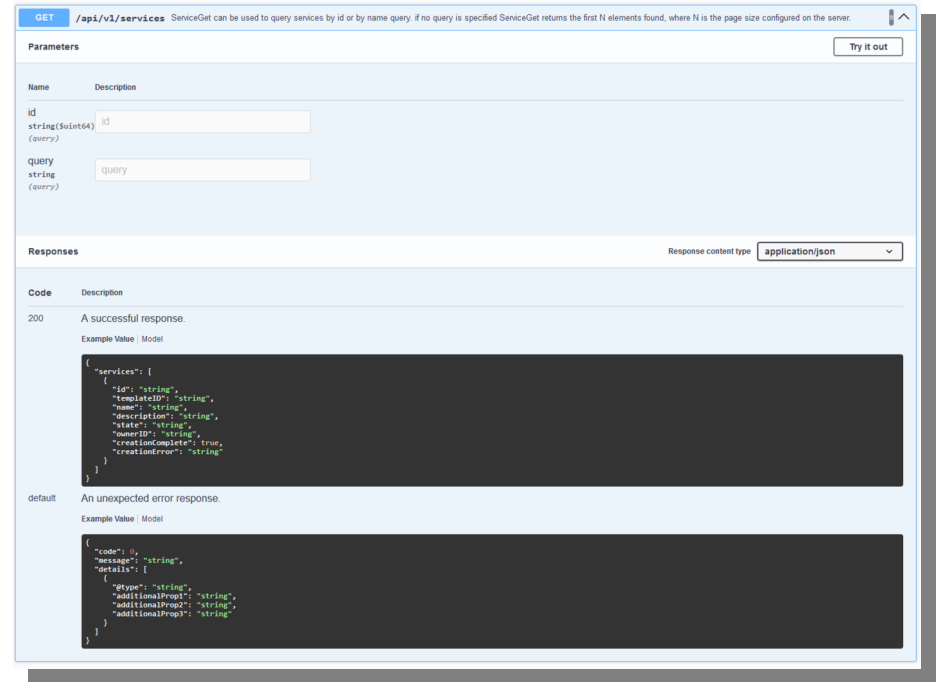
Cloud Deploy – REST-Like API



The image shows the Swagger UI interface for an API. The top bar includes the Swagger logo, the URL 'openapi/api.swagger.json', and an 'Explore' button. The main content area is titled 'api.proto' and 'CloudDeploy'. It lists various endpoints with their methods and descriptions:

- GET /api/v1/dashboard: DashboardGet returns all data required to render the frontend dashboard
- GET /api/v1/jobs: JobsGet lists all jobs based on the specified condition
- POST /api/v1/jobs: ServiceActionRun allows users to execute a service action upon a service. An exhaustive set of parameters matching the set of parameters required by the service action must be specified. If successful ServiceActionRun will return the job object that was created for this service action execution.
- POST /api/v1/jwt: debug method that gets a jet from keycloak: only used for testing
- POST /api/v1/mock/service
- GET /api/v1/mock/templates
- GET /api/v1/ping
- GET /api/v1/playbooks: PlaybookGet can be used to query playbook by id or by name query. if no query is specified PlaybookGet returns the first N elements found, where N is the page size configured on the server.
- PUT /api/v1/playbooks: PlaybookUpdate can be used to update a playbook. The update is reflective, meaning that the server will apply the update requested by the client and then return the modified state to the client.
- GET /api/v1/profile: ProfileGet will return the user object of the user making the request
- PUT /api/v1/profile: ProfileUpdate can be used by any user to update the non-static fields of their profile
- GET /api/v1/resources: ResourceGet can be used to query resources. it allows the user to fetch a free resource with a specific tag and type from the server
- DELETE /api/v1/resources: RoleDelete can be used to delete a resource.
- POST /api/v1/resources: ResourceCreate can be used to create a resource. The creation is reflective, meaning that the server will create the resource requested by the client and then return the new resource to the client.
- PUT /api/v1/resources: ResourceUpdate can be used to update a resource. The update is reflective, meaning that the server will apply the update requested by the client and then return the modified state to the client.
- GET /api/v1/roles: RoleGet can be used to query roles by id or by name query. if no query is specified RoleGet returns the first N elements found, where N is the page size configured on the server.
- DELETE /api/v1/roles: RoleDelete can be used to delete a role

Dokumentation



The image shows the Swagger UI interface for live testing an API endpoint. The endpoint is 'GET /api/v1/services'. The 'Parameters' section shows a table with columns 'Name' and 'Description'. The 'id' parameter is a string (required) with a text input field. The 'query' parameter is a string (required) with a text input field. The 'Responses' section shows a table with columns 'Code' and 'Description'. The '200' response is a successful response with an example value:

```
{ "services": [ { "id": "string", "templateID": "string", "name": "string", "description": "string", "state": "string", "ownerID": "string", "creationComplete": true, "creationError": "string" } ] }
```

. The 'default' response is an unexpected error response with an example value:

```
{ "code": 0, "message": "string", "details": [ { "type": "string", "additionalProp1": "string", "additionalProp2": "string", "additionalProp3": "string" } ] }
```

Live testing mit swagger



ZABBIX
PREMIUM PARTNER

Basiert auf OpenAPI Spezifikation, unterstützt Swagger - <https://swagger.io/>

Cloud Deploy APIs - Terraform

Cloud Deploy – Terraform Provider

```
terraform {
  required_providers {
    cloudeploy = {
      source = "intellitrend.de/cloud-deploy/cloud-deploy"
      version = "1.0.0"
    }
  }
}

# Configuration for the Cloud Deploy provider
provider "cloudeploy" {
  user      = "admin"
  password  = "admin"
  server_addr = "cloud-deploy.loc"
  server_port = 8090
}
```

Cloud Deploy kann mit Terraform verwendet werden – <https://www.terraform.io/>

Cloud Deploy – Terraform Service Beispiel

```
# Configuration for a service
resource "clouddeploy_service" "zabbix-60-small" {
  # name of the service template this service is based on,
  # changes after creation cause a re-deployment of the service
  template = "Zabbix 6.0"
  # optional name of the service. if not specified, the name is derived
  # from the template name and the creation time
  name = "Zabbix Small Instance"
  # optional description details for the service
  description = "A small Zabbix instance"
  # optional flag to wait for service jobs to finish
  # note: may cause timeouts on long running tasks
  block = false
  # parameters that are passed to action playbooks,
  # changes will cause service update actions to run on this service
  parameters = {
    "factory_name" = "Zabbix small instance"
    "factory_slug" = "zabbix-small"
    "factory_size" = "small"
    "gitlab_user"  = "admin"
    "email"       = "support@intellitrend.de"
  }
}
```



Cloud Deploy – Terraform Apply

```
$ terraform apply
```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

```
# clouddeploy_service.zabbix-60-small will be created
+ resource "clouddeploy_service" "zabbix-60-small" {
  + block      = false
  + description = "A small Zabbix instance"
  + id         = (known after apply)
  + name       = "Zabbix Small Instance"
  + parameters = {
    + "email"       = "support@intellitrend.de"
    + "factory_name" = "Zabbix small instance"
    + "factory_size" = "small"
    + "factory_slug" = "zabbix-small"
    + "gitlab_user"  = "admin"
  }
  + status      = (known after apply)
  + template    = "Zabbix 6.0"
}
```

Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ zabbix-60-small-status = (known after apply)

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

```
clouddeploy_service.zabbix-60-small: Creating...
clouddeploy_service.zabbix-60-small: Creation complete after 1s [id=180]
```

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

```
Outputs:
zabbix-60-small-status = "Job Zabbix 6.0: Create (CREATION): begin job"
```

Apply Terraform state
(returns immediately)



ZABBIX
PREMIUM PARTNER

Cloud Deploy – Terraform Refresh

```
$ terraform refresh
clouddeploy_service.zabbix-60-small: Refreshing state... [id=180]
```

Outputs:

```
zabbix-60-small-status = "Job Zabbix 6.0: Create (CREATION): playbook create_validate.pb.yaml completed"
```

```
$ terraform refresh
clouddeploy_service.zabbix-60-small: Refreshing state... [id=180]
```

Outputs:

```
zabbix-60-small-status = "Job Zabbix 6.0: Create (CREATION): playbook create_database.pb.yaml completed"
```

```
$ terraform refresh
clouddeploy_service.zabbix-60-small: Refreshing state... [id=180]
```

Outputs:

```
zabbix-60-small-status = "Job Zabbix 6.0: Create (CREATION): playbook create_factory_config.pb.yaml completed"
```

```
$ terraform refresh
clouddeploy_service.zabbix-60-small: Refreshing state... [id=180]
```

Outputs:

```
zabbix-60-small-status = "Ready"
```



ZABBIX
PREMIUM PARTNER

Überprüfung des Fortschritt mit “refresh”

“One to rule them all” Monitoring der installierten Zabbix Instanzen

Monitoring of deployed Zabbix Instances

- Ein zentraler "Master"-Zabbix-Server überwacht alle installierten Zabbix-Instanzen
- Jede Zabbix-Instanz definiert einen Dienst auf dem Master, einschließlich Zabbix Server, Zabbix Agent und Frontend
- Der Master Server hat Zugriff auf die internen Metriken jeder Zabbix-Instanz
- Jede Zabbix-Instanz hat ein eigenes Dashboard



Cloud Deploy – LLD Support

Hosts Create host Import

Filter

Name	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
Cloud Deploy	Items	Triggers	Graphs	Discovery 1	Web			Cloud Deploy Zabbix services via HTTP IAS	Enabled		None		
Factory discovery: My Zabbix - Zabbix Agent	Items 1	Triggers 1	Graphs	Discovery	Web	`\${SERVER_IP}:10050		Cloud Deploy Zabbix Agent	Enabled	ZBX	None		SvcMonHost
Factory discovery: My Zabbix - Zabbix Frontend	Items 1	Triggers 1	Graphs	Discovery	Web	`\${FRONTEND_URL}:10050		Cloud Deploy Zabbix Frontend	Enabled	ZBX	None		SvcMonHost
Factory discovery: My Zabbix - Zabbix Server	Items 65	Triggers 43	Graphs 11	Discovery 1	Web	`\${SERVER_IP}:10050		Cloud Deploy Zabbix Server, Remote Zabbix server health extended (Remote Zabbix server health)	Enabled	ZBX	None		dashboard SvcMonHost

Host x

Host IPMI Tags 2 Macros 4 Inventory Encryption

Discovered by [Factory discovery](#)

* Host name

Visible name

Templates Name

- Remote Zabbix server health extended
- Cloud Deploy Zabbix Server

* Groups Select

Interfaces	Type	IP address	DNS name	Connect to	Port	Default
Agent		<input type="text" value="`\${SERVER_IP}`"/>		IP DNS	10050	<input type="checkbox"/>

Description

Monitored by proxy

Enabled

Update Clone Full clone Delete Cancel

Von Cloud Deploy bereitgestellte Dienste können über LLD “discovered” und von Zabbix überwacht werden.



Cloud Deploy – Service Definition pro Instanz

The screenshot displays the Zabbix web interface for 'IntelliTrend Advanced Services for Zabbix v6.1.0'. The left sidebar shows navigation options like Monitoring, Services, Inventory, Reports, Configuration, Administration, Support, Integrations, Help, User settings, and Sign out. The main content area shows a summary of system status: 1 Service, 3 Hosts, 3 Triggers, 3 Tag Names, and an unlimited License. Below this, the 'Services' section is active, showing a list of services for '1) My Zabbix'. A red box highlights a flowchart for 'My Zabbix' services, which includes three items: 'My Zabbix - Zabbix Server', 'My Zabbix - Zabbix Frontend', and 'My Zabbix - Zabbix Agent'. Each item has a corresponding trigger definition, such as 'Zabbix server is down on My Zabbix - Zabbix Server' with the expression 'max(/my-zabbix - Zabbix Server/net.tcp.service[tcp,,10051],#3)=0'.

Pro Zabbix-Service wird automatisch ein Flowchart aus der jeweiligen Service Konfiguration erstellt.

Der Service Status und die zugehörigen Trigger-Status werden in Echtzeit angezeigt.

Auf dem Master für jede Zabbix-Instanz erstellter Service mit grafischer Darstellung



ZABBIX
PREMIUM PARTNER

Cloud Deploy – Zabbix Instanzen auf dem Master

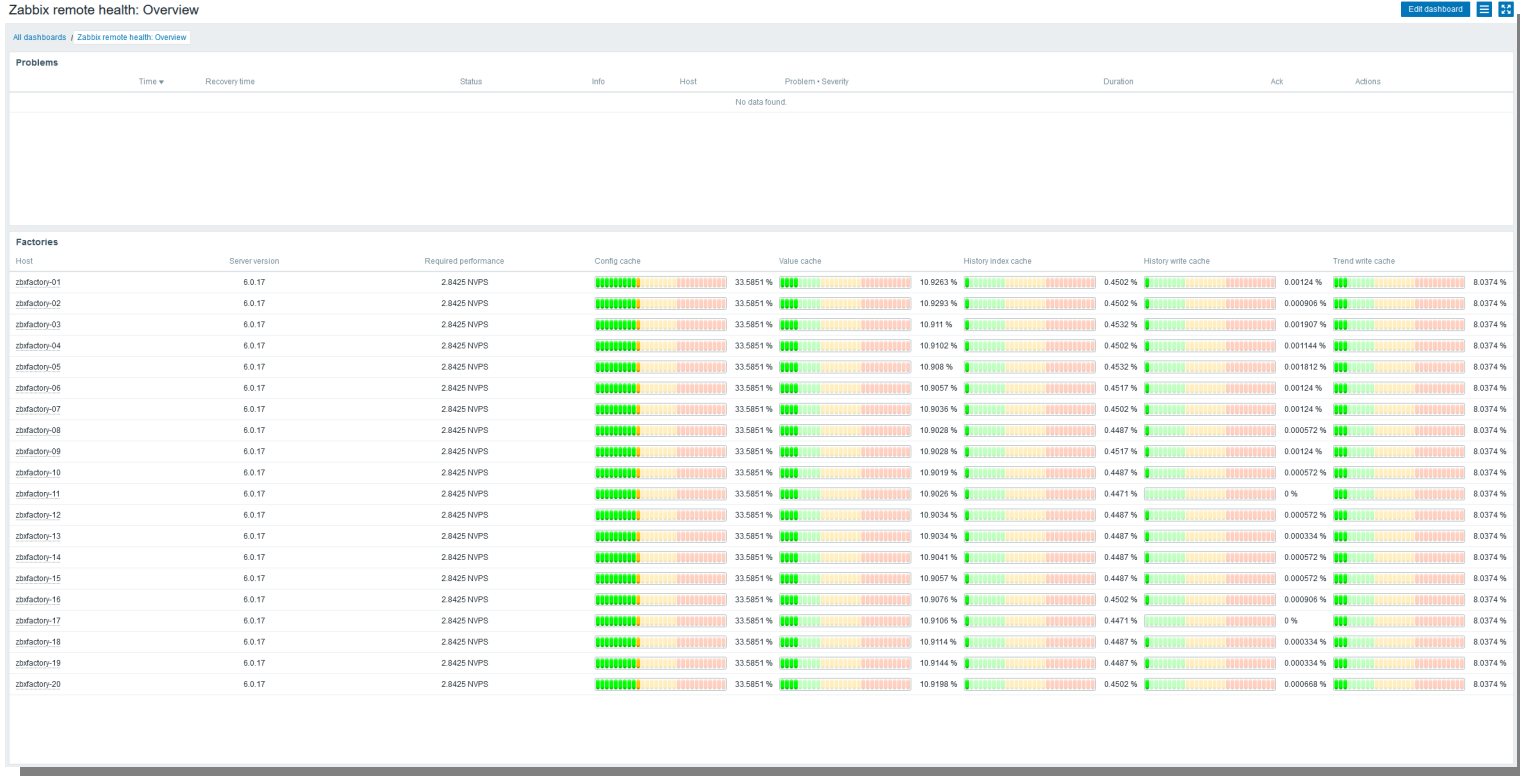
Hosts [Create host](#) [Import](#)

Name	Items	Triggers	Graphs	Discovery	Web interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
<input type="checkbox"/> Zabbix server	Items 128	Triggers 80	Graphs 25	Discovery 4	Web zabbix-agent 10050		Linux by Zabbix agent, Zabbix server health	Enabled	ZBX	None		
<input type="checkbox"/> zbfactory-01	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-02	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-03	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-04	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-05	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-06	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-07	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-08	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-09	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-10	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-11	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-12	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-13	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-14	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-15	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-16	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-17	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-18	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-19	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbfactory-20	Items 57	Triggers 42	Graphs 11	Discovery 1	Web		Remote Zabbix server health	Enabled		None		dashboard

Displaying 21 of 21 found



Cloud Deploy – Zabbix Instanzen auf dem Master



“Health Status Dashboard” aller Zabbix Instanzen auf dem Master



ZABBIX
PREMIUM PARTNER

Individuelle Zabbix Installationen in On-Premises Kubernetes Umgebungen



IntelliTrend GmbH

 www.intellitrend.de



ZABBIX
PREMIUM PARTNER

Danke!



Kontakt: Wolfgang Alper

wolfgang.alper@intellitrend.de