

# Zabbix meets Cloud Deploy

—

## Intelligent Provisioning of multiple Zabbix Instances in Kubernetes



IntelliTrend GmbH



Contact: Wolfgang Alper

 [www.intellitrend.de](http://www.intellitrend.de)

[wolfgang.alper@intellitrend.de](mailto:wolfgang.alper@intellitrend.de)



**ZABBIX**  
PREMIUM PARTNER

# Zabbix meets Cloud Deploy

-

## Intelligent Provisioning of multiple Zabbix Instances in Kubernetes

With special thanks to

**Deutsche Telekom Technik GmbH**  
for a great project

# Zabbix meets Cloud Deploy

## Lets start simple Zabbix installation with Docker

# Zabbix installation with Docker

## Using Docker command line

```
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
```

## More optional Env-Vars

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

# Zabbix meets Cloud Deploy

## Zabbix installation with Kubernetes

# Installation options with Kubernetes

- Deploy Zabbix (similar to Docker) manually with YAML based configuration files using „kubectl“

or

- Deploy Zabbix using templates, configuration repositories and config synchronization using “Helm”

# Introducing Helm

- Helm is often called “The **package manager** for Kubernetes”
- Helm is both a K8s **config repository** and **templating system**
- Bundles multiple configuration files into a **release**
- Allows consistent upgrades with versioning
- Flexible configuration via a „values.yaml“ file
- Official 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>  
(Former 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-server
    app.kubernetes.io/managed-by: {{ .Release.Service }}-zabbix-server
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-server
        app.kubernetes.io/managed-by: {{ .Release.Service }}-zabbix-server
    spec:
      containers:
        - name: zabbix-server
          image: {{ .Values.zabbixserver.image.repository }}/{{ .Values.zabbixserver.image.tag }}
          ports:
            - containerPort: {{ .Values.zabbixserver.hostPort }}
          resources:
            limits:
              memory: 128Mi
            requests:
              memory: 64Mi
              cpu: 100m
          volumeMounts:
            - name: zabbix-data
              mountPath: /var/lib/zabbix
      volumes:
        - name: zabbix-data
          persistentVolumeClaim:
            claimName: zabbix-data
{{- end }}
```

```
# 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



# Zabbix meets Cloud Deploy

## Installing Zabbix in K8s using Helm

# Installing Zabbix using Helm

Install 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!*
```

Create values\_zabbix.yaml (skeleton with defaults)

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

# Installing Zabbix instance

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"
```



# Installing Zabbix instance

Create helm release „zabbix-1“ in 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 for „zabbix-1“ after installation

Overview Pods Deployments DaemonSets StatefulSets ReplicaSets Jobs CronJobs									
Pods									
3 items			Namespace: zabbix		Search Pods...				
<input type="checkbox"/>	Name	Namesp...	Contain...	Restarts	Controll...	Node	QoS	A...	Stat...
<input type="checkbox"/>	zabbix-1-zabbix-proxy-0	zabbix	■ ■	0	StatefulSet	worker3	BestEffort	45s	Running
<input type="checkbox"/>	zabbix-1-zabbix-server-0	zabbix	■ ■	0	StatefulSet	worker2	BestEffort	45s	Running
<input type="checkbox"/>	zabbix-1-zabbix-web-64d7...	zabbix	■	0	ReplicaSet	worker1	BestEffort	45s	Running



**ZABBIX**  
PREMIUM PARTNER

# Running Zabbix instance

Zabbix Server for „zabbix-1“ initializing database:

```
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
```

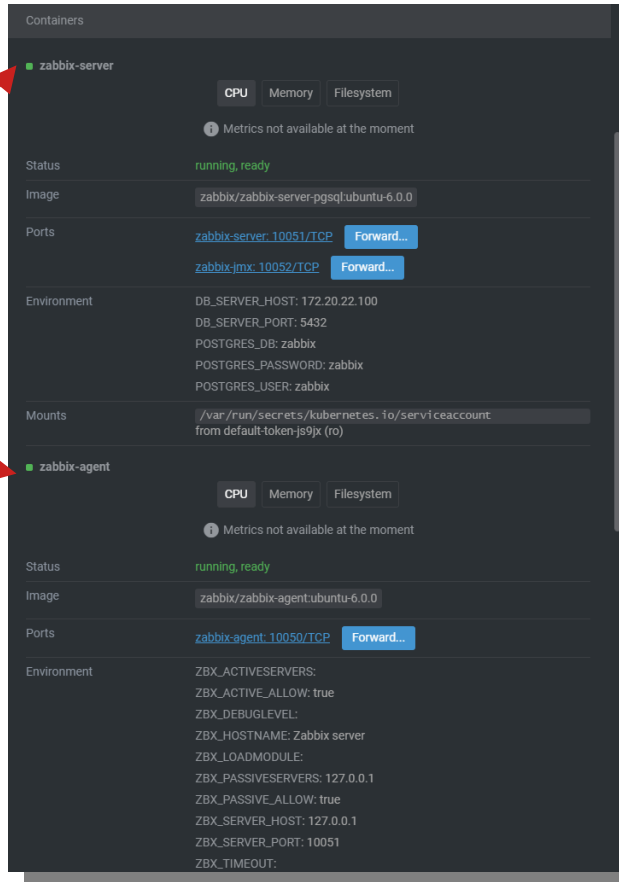
Zabbix Server for „zabbix-1“ starting:

```
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.
```



# Running Zabbix instance – 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/serviceaccount from default-token-js9jx (ro)

■ zabbix-agent

CPU Memory Filesystem

Metrics not available at the moment

Status: **running, ready**

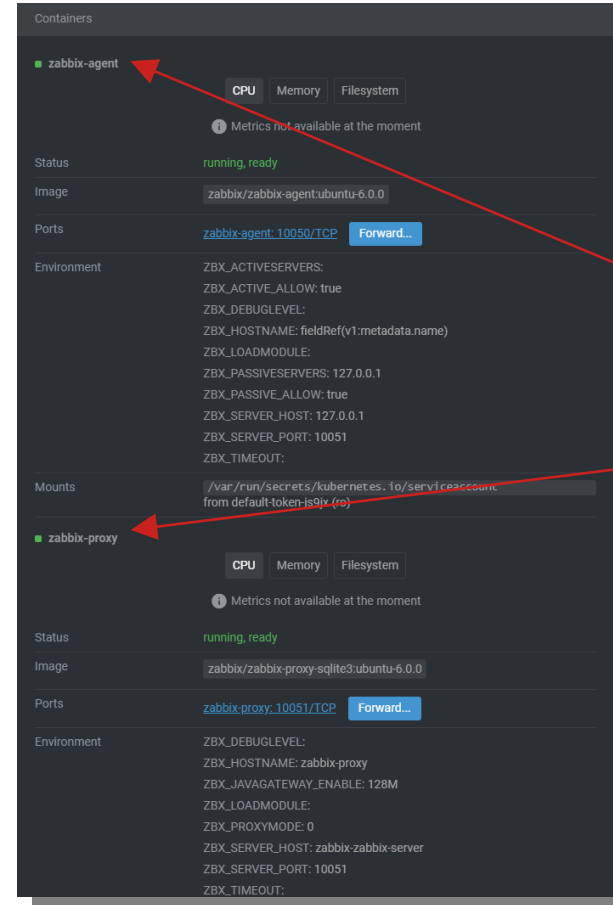
Image: zabbix/zabbix-agent:ubuntu-6.0.0

Ports: zabbix-agent:10050/TCP Forward...

Environment: ZBX\_ACTIVE\_SERVERS:  
ZBX\_ACTIVE\_ALLOW: true  
ZBX\_DEBUGLEVEL:  
ZBX\_HOSTNAME: fieldRef(v1.metadata.name)  
ZBX\_LOADMODULE:  
ZBX\_PASSIVE\_SERVERS: 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/serviceaccount 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\_ACTIVE\_SERVERS:  
ZBX\_ACTIVE\_ALLOW: true  
ZBX\_DEBUGLEVEL:  
ZBX\_HOSTNAME: fieldRef(v1.metadata.name)  
ZBX\_LOADMODULE:  
ZBX\_PASSIVE\_SERVERS: 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/serviceaccount 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

# Zabbix meets Cloud Deploy

Make services accessible  
from outside the cluster

# Make services accessible from outside

- HTTP based services can be routed using a reverse proxy like nginx or Traefik
- Non-HTTP based TCP / UDP services cannot be routed easily, they require a dedicated external IP address

Cloud services like AWS, Azure or GCS provide load balancers as a solutions to this

How to solve this in a self-hosted cloud environment?



# Make services accessible from outside

## Introducing MetalLB

- MetalLB provides a **layer 2 network load balancer** usable in self-hosted cloud environments
- Assigns external IP addresses to cluster services, does not rely on URLs or 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

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

# Reconfigure Zabbix for MetalLB

Adjust `zabbix_values.yaml` to use service type `LoadBalancer` instead of `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

Upgrade release „zabbix-1“ after the change to `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**  
PREMIUM PARTNER

# Zabbix Services via MetalLB

Service for Zabbix server now has the type „LoadBalancer“ and is accessible via an external IP address

Services

4 items

Namespace: zabbix

Search Services...

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

Zabbix server is accessible outside the cluster using:

- External IP 172.20.22.110 (assigned by MetalLB)
- Port 10051

# Zabbix meets Cloud Deploy

## Real world challenges

# Kubernetes - Real world challenges

When installing applications in a Kubernetes cluster you might face some challenges

- Not everyone is used to kubectl, Helm charts etc.
- How to ensure consistent application configurations?
- How to track configuration changes?
- How to manage passwords/secrets in configuration files?
- How to allow users/admins to simply deploy an application?

We need **building blocks** to create  
a more **user friendly** approach



# Zabbix meets Cloud Deploy

Building block: **Flux**  
(Manage and apply configurations)

# Building Block - Introducing Flux

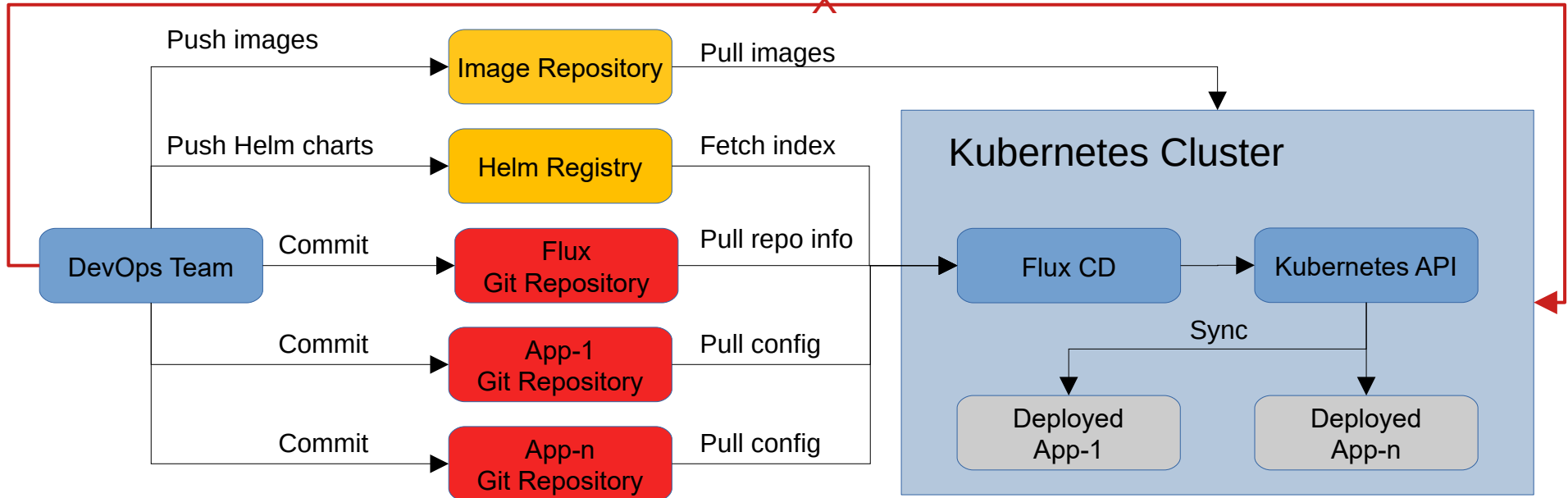
**Flux** offers a GitOps solution for Kubernetes

- K8s configuration is synchronized with Git repositories, which gives:
  - Proper versioning
  - Easy rollbacks
  - Consistent config states
- Flux is deployed on the cluster and controls it via the Kubernetes API
- Instead of „kubectl apply“ or „helm install“, changes are committed and pushed to Git repositories
- Flux polls these Git repositories for changes and performs automatic rollouts in the cluster when required



# Building Block - 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 our case apps are individual Zabbix instances

# Zabbix meets Cloud Deploy

Building block: **SOPS**  
(Manage and apply secrets)

# Building Block – Introducing SOPS

- Applications require credentials (databases, APIs, email, etc.) that are stored in config files for GitOps
- However: Clear-text credentials in Git is a bad idea!
- Use encryption?
  - Deployed applications still need to be able to read credentials
- Where to put the keys for the encrypted credentials?
  - Chicken-and-egg problem
- Multiple users may have to edit credentials
  - How do we share the keys?



# Building Block - SOPS

Mozilla **SOPS** (Secrets **OP**eration**S**) manages encrypted information in text files – supported by **Flux**

- Sensitive values are encrypted and encoded into special Base64 strings
- Uses asymmetric cryptography **with one or multiple** public/private key pairs
- Flux receives private key as a secret and automatically decrypts encrypted YAML files from Git repositories on the fly
- Only owners of the private keys can view and edit the encrypted values, multiple owners are possible
- Support for 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: 85D77543B3D624B63CEA9E6D8C17301B491B3F21
        created_at: 1441570391.930042
        enc: |
          -----BEGIN PGP MESSAGE-----
          hQIMA0t4uZHf19qAQ//UvGAwGePyHuf2/zayWc1oGaDs0MzI+zw6CmXvMRNPUsA
          ...=0JgS
          -----END PGP MESSAGE-----
```

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



# SOPS – Basic encryption example

```
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+1pY5wgp0dGY13NG4LjKGir0zk=,tag:UTTRhSgW4Wbm79boaefAxQ==,type:str]
password: ENC[AES256_GCM,data:+Fd9G1guMhB8o13s,iv:Yd47/6YPKNJwWl17sso7nymAeanQ3hiERXLw4Nrx/ts=,tag:cskv9EZBGw2Fn9zNrWg8Pw==,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+NdjQpV06Ie4jzC+XA71aJXj7wEInk4YLgq7yjhPaFoMv1yVn415cPb11EA2dR
  pgp:
    - created_at: "2022-04-26T14:50:20Z"
      enc: |
        -----BEGIN PGP MESSAGE-----

        hQEMAYUpShfNkFB/AQf/U16Wq+X6o00rI8b/Dpremc5Mcha/6Bz43V28PIhqS1IC
        xLPkR0yqXU48h0zp11bLsviDLz+msAny713T91y9thS/V0Sbwj6mhZhc2CvTAL7g
        8CuXcAhGk70vTh8DH7N/L3zE5514wLJ0+gnbn7CroxpoVPLbliite07gf3zIYBdv
        xGDBab0jUua/Y7Z/w0456IA6RmpB9vEe9wR1GE5iItt8a/BAsi9pWmG9vXN3UPUD
        Ub4phl+c3REqmxShVuzOsDQLCerBqk3F4PTIXozCuWkVz44kKwD4wIcgZC3dFFR
        Ot/JVkpDhHYjt0c1Q4AsIUayXVziKbwaNcTgN/ed9JeAYPKTdtGFUthRzro3rk9
        5jAtpDSV88EcXqLk36oYT/IU+bzPaFyWA8Gj+R5WPyPhXyWqNeLNVajwoPc4oVG
        aJ+hvthvLd/JNvpPaLYxvAmfTxdfoOP4Zz7Cupq9dw==
        =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 – More complex example

```
# data collection
StartPingers: 5
StartPollers: 100
# database section
DBName: ENC[AES256_GCM,data:hOPj6ocNN723iOEHWHLJzw==,iv:fYfP9eVMB1NyhasoLSfRI7nfwplhiUoY37ydLbKy5k=,tag:SXvjSKQQfnUahtTrdcHPBw==,type:str]
DBUser: ENC[AES256_GCM,data:22v5h0ouL04=,iv:VDq/MptR0K/4E7nlBnzKzCjvS3iU19yzjJgN0mwS2Ns=,tag:z6E+7gnjBstG+H4cGfW3Gw==,type:str]
DBPassword: ENC[AES256_GCM,data:+ljE8VGD/94JNm2oEEDjIB81,iv:1V1TzaKLC1I4xRkG1PHQ1EfafAh0Tk3PIxxyTtk5do=,tag:xYU00dYUW3cBrrzNpM7LRg==,type:str]
sops:
  kms: []
  gcp_kms: []
  azure_kv: []
  hc_vault: []
  age:
    - recipient: age1tqltjgmzlaw7eek27sdveygt6w2cer3zad6sn3eqtrjfrvrs5dtqwwq079
      enc: |
        -----BEGIN AGE ENCRYPTED FILE-----
        YWdlLWVuY3J5cHRpb24ub3JnL3YxYXciO+IFgyNTUxOSBOSjJtazhYNkJJQXgvaTZO
        V3F2bTduNDFOOTVFEYlNvZ1VqS3ZmMlE3RldBCj15Tm50N3BGbVAreXlwT2t0S3Vl
        dEJGWHJlWU1ZY2UxRlRXT0gzRFNza0EKL50tIHRpK1Q5UHpRV3FLcGpjZWpGSzZk
        Ykt0aXNkb0ozUWREZFZTWFB5NWYvMVkK0pHn13XD4Fy6xvD98Iq3HaSsnKUoEEtG
        d+iSaktouuXgB2lBie2ZW3d9Lb8xmXBAiWLvovQbcvE1S/Gk0/Lfhg==
        -----END AGE ENCRYPTED FILE-----
    - recipient: age1r3lnjrmymk2kawe59vwttn37czvgxmnd0eypfhrpcgck5fgxrufqnx0nn
      enc: |
        -----BEGIN AGE ENCRYPTED FILE-----
        YWdlLWVuY3J5cHRpb24ub3JnL3YxYXciO+IFgyNTUxOSB0IAEpRdmNHeksvVF4b3N4
        cH2nc3lGMS8ydnJsUWlqSkw2c2dKQ0QydkJrCl42S0VsN3h4TEpUTTdWY1ZzWGJK
        NXoyeVpHNXl14OWEzdFozR3hBbkptYVVKLS0tIDl1K2dBOHFxdzdVdkowMwtZcjVt
        MU5CeXBhdUFUGcTd1ZHBobEpDcFFOWUkKulBletkn4pLYnDeyz2ZWlekExWVtceVt
        mngvV5CbAwLIIs/Qm2eS2F/ahH8E7kiJ+JtVoRcIqIXOyrX3WpYk57g==
        -----END AGE ENCRYPTED FILE-----
  lastmodified: "2023-08-30T12:59:52Z"
  mac: ENC[AES256_GCM,data:z/liUEIQreDvKGWmqbgSRODCxDsUlFkxPbcjFUIh3OpYHnbGx7OrR/zvyDqsaT+uLzRJCMItyxTKUikPfePjisJrlBWm0Dq8Ib9nB/
  sPMQ3QiBizUxAmi46QzV4TYhdNrgAlE/A0g+NDGvOpIn2cQBE/iHagH9k89vtgK0d3gzc=,iv:bHVR0t9fgt4Q8mrE6EzrSpLjwiwt7Lxg4asf6oiNyI=,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

Different users/applications can update/read the file without sharing a common secret!

# Zabbix meets Cloud Deploy

Building block: **Cloud Deploy**  
(Manages cloud services)

# Building Block – Introducing Cloud Deploy

Lets **ask** ChatGPT: “What is cloud deploy?”



# Building Block – Introducing Cloud Deploy

Answer from ChatGPT:



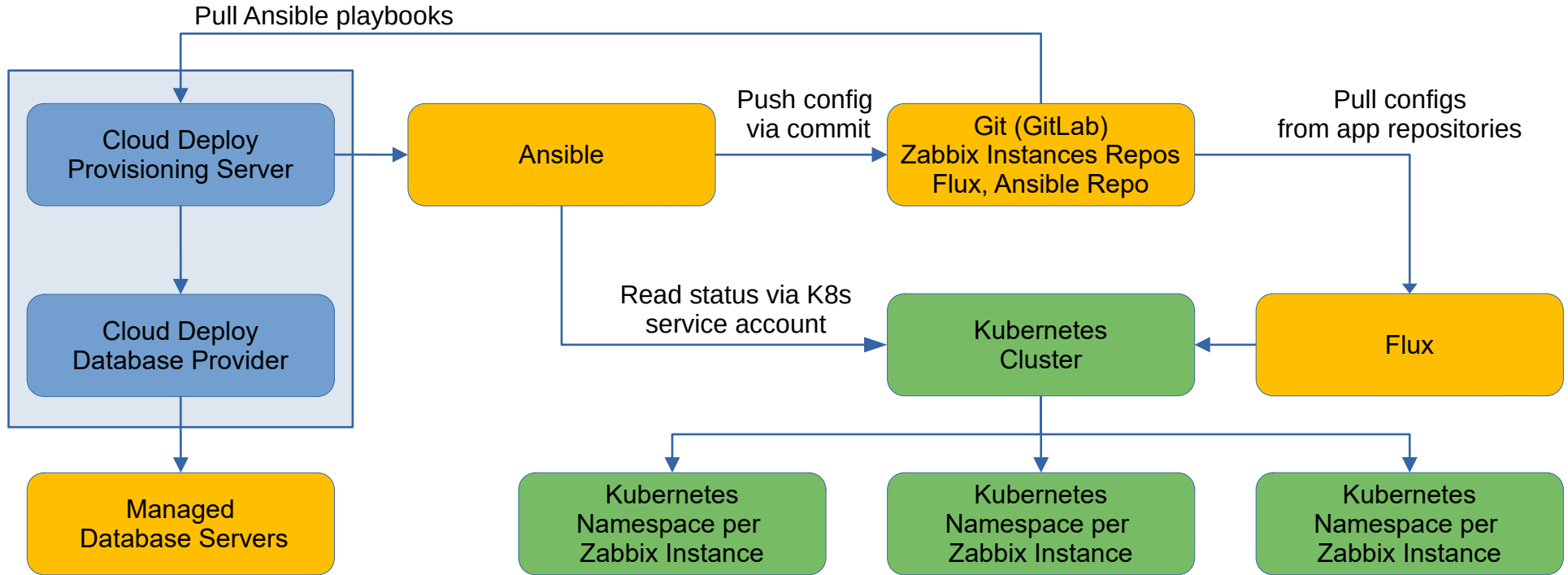
"Cloud deploy" is not a specific term or technology; rather, it is a broad concept that refers to the process of deploying applications or services on cloud infrastructure.

Cloud deployment involves taking your software application or system and making it available to users or clients over the internet using cloud resources and services.

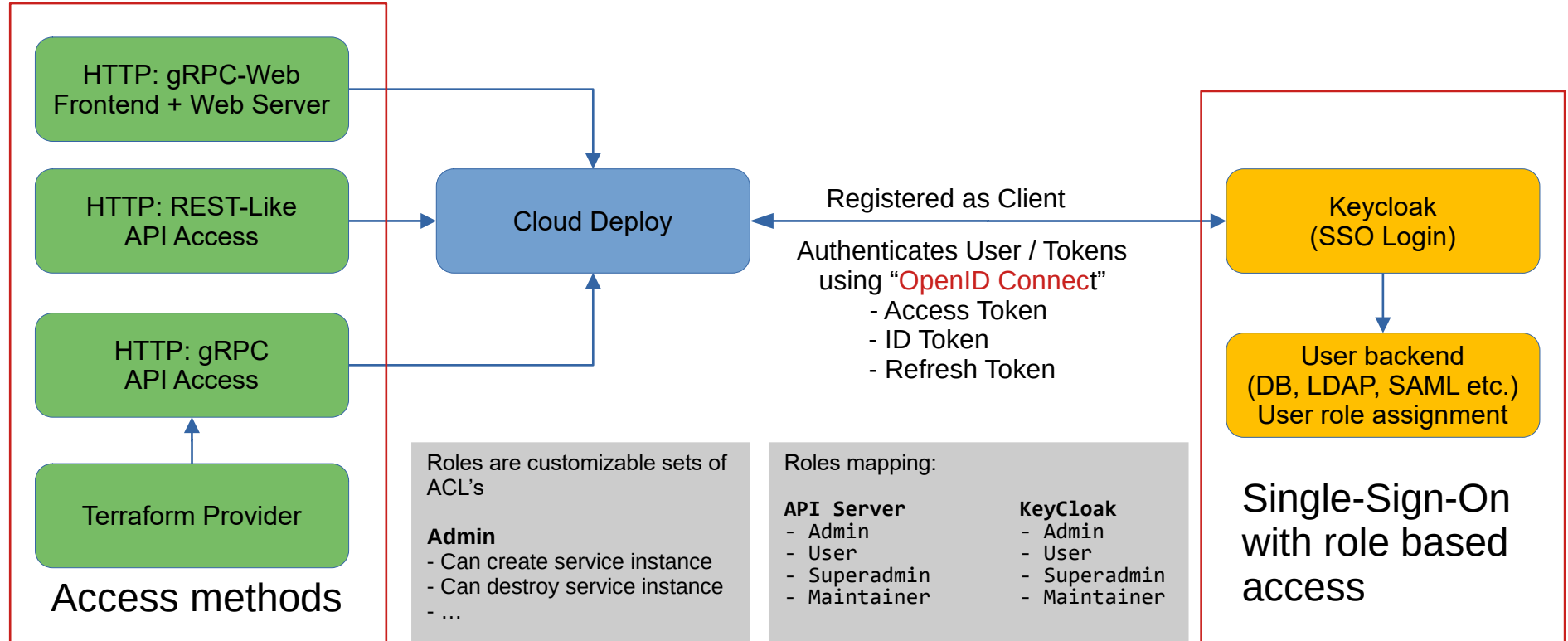
# Zabbix meets Cloud Deploy

## Cloud Deploy internals

# Cloud Deploy – Provisioning Design

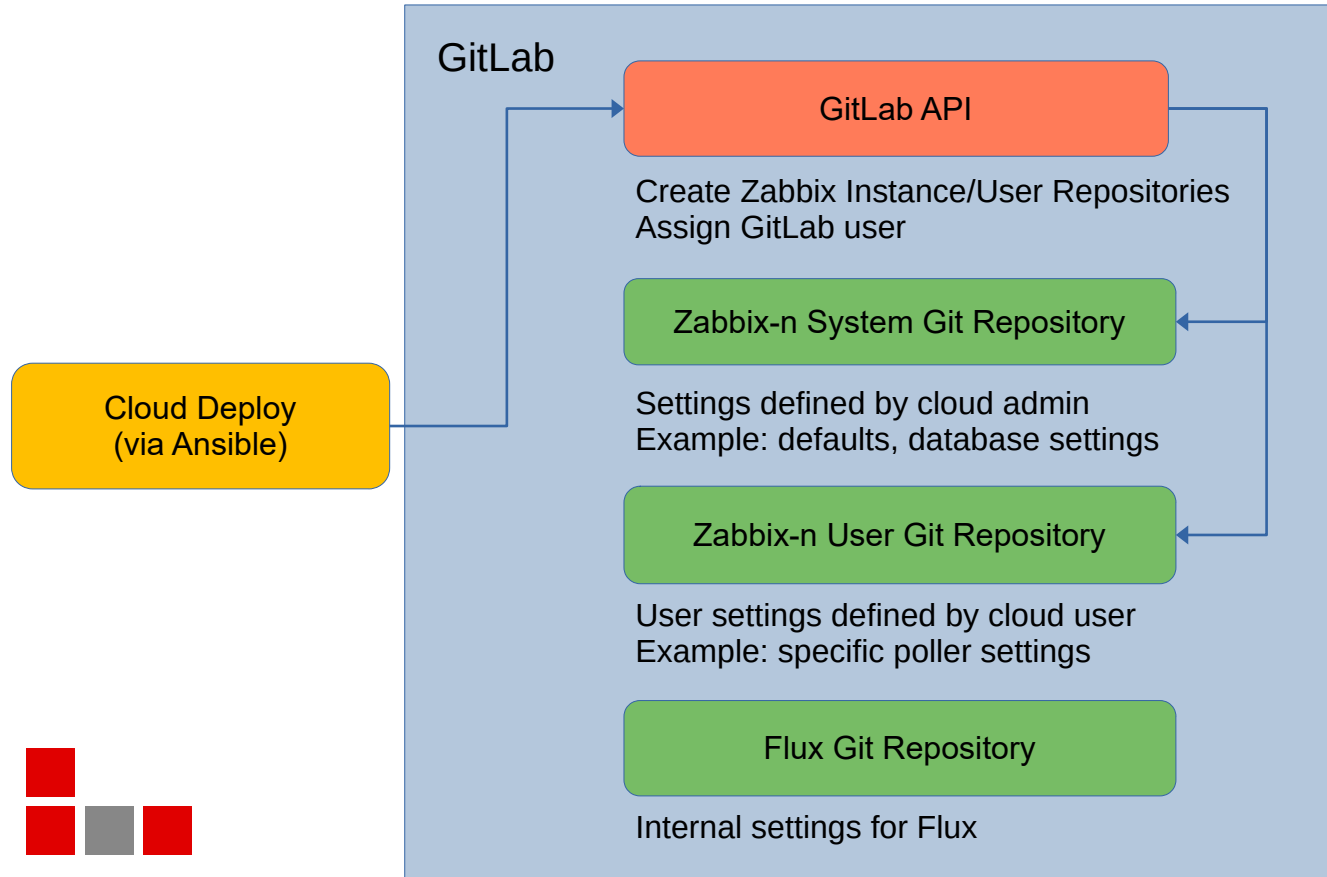


# Cloud Deploy – Auth Design and API Access



In our case services are individual Zabbix instances (including Frontends etc.)

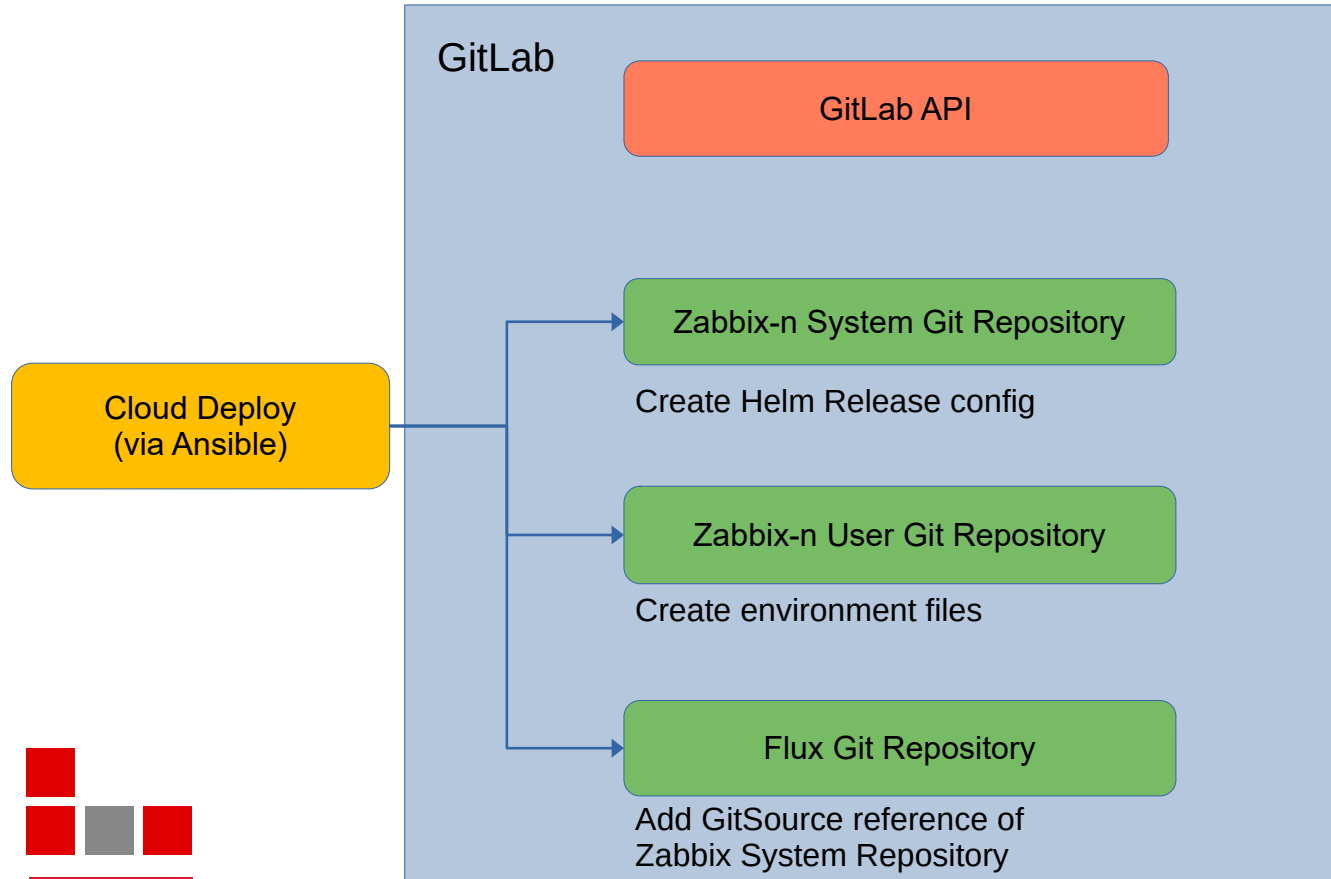
# Cloud Deploy - Git setup and workflow - Setup



Cloud Deploy creates its own set of Git repositories for each Zabbix installation

- System Git repository
- User Git repository

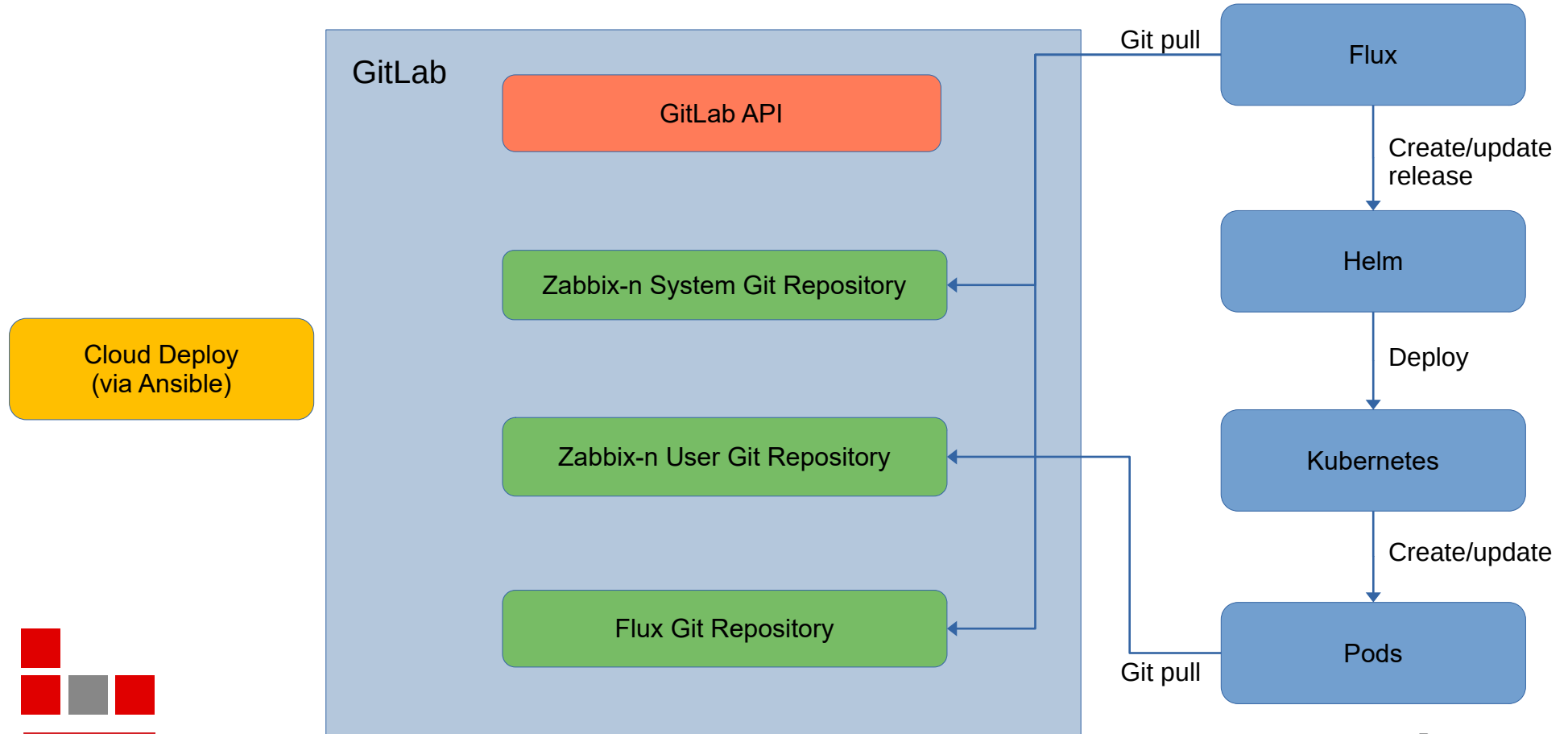
# Cloud Deploy - Git setup and workflow - Config



Cloud Deploy commits created files and configurations automatically

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



# Cloud Deploy - Git setup and workflow - Deploy








# Cloud Deploy – Created Flux repository files

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 – Created Flux repository files

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 to read Flux config from the previously registered Zabbix repository.

# Cloud Deploy - Zabbix repositories

Zabbix

**Zabbix** Group ID: 3 [Leave group](#)

[New subgroup](#) [New project](#)

[Subgroups and projects](#) [Shared projects](#) [Archived projects](#)  [Name](#)

	<b>F</b> <b>Factories</b> <a href="#">Owner</a>	0  2  1
<b>M</b> <b>My Zabbix</b>	0	1 month ago
<b>M</b> <b>My Zabbix - System</b>	0	1 month ago

New repositories created by Cloud Deploy via Ansible for each new Zabbix instance

# 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:
```

Helm Release config created by Cloud Deploy.  
Contains the values for the Helm Chart

SOPS

```
values:
  postgresql:
    database: zabbix
    host: 172.20.22.50
    password: ENC[AES256_GCM,data:NomliZ/1k9xbYw==,iv:mqG41l42+uFdQZwwxP0i0m2dwRnJh19mnefMe1LE8BI=,tag:2
    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:
```



**ZABBIX**  
PREMIUM PARTNER

# Cloud Deploy - Zabbix user repository

Zabbix > Factories > My Zabbix

**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

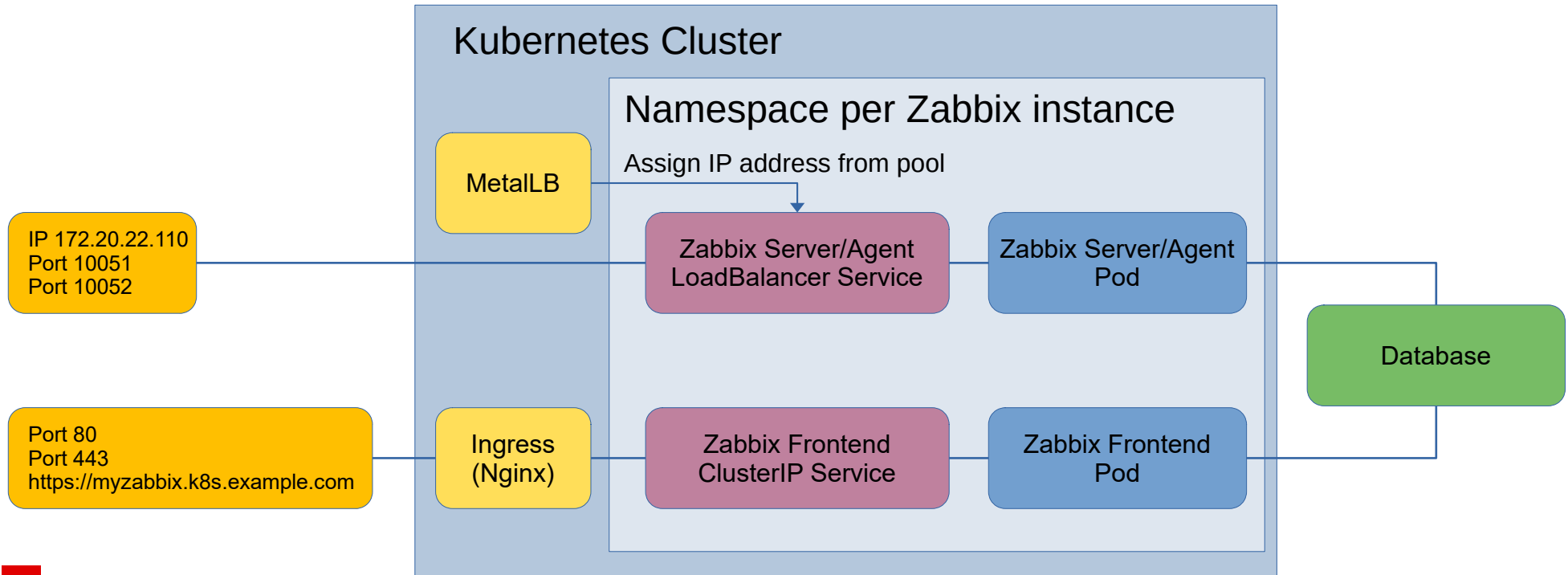
.env 976 bytes

```
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_SS0_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.
```

User repository contains environment files with the user specific configurations.



# Cloud Deploy - Cluster setup after reconciling



# Zabbix meets Cloud Deploy

## Cloud Deploy User Portal

# Cloud Deploy - User Portal

- Web based portal allows provisioning of new services in Kubernetes (here Zabbix)
- **Services** are based on **Service Templates**.
- Service Templates define **Provisioning Actions** to create, update, delete and migrate a service.
- Each Provisioning Action can have multiple **Tasks**.
- Supports SSO and RBAC.

# Cloud Deploy – Service Templates

←

Service Template Configuration

Name

Zabbix 6.0

Description

Template for a Zabbix 6.0 factory

Owner

LocalAdmin

Status

PUBLISHED

Service Actions

+ NEW ACTION

Update

Update

Create

Creation

Delete

Deletion

Cloud Deploy Template for a 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 configuration for  
Update, Create and Delete

Playbook with per task  
configuration

# Cloud Deploy – Create a new Service

Services

There are no services for you yet.  
Use the  in the bottom right corner to create a new Service.


Create a new Service

Choose a Service Template

Service Template  
Zabbix 6.0

NEXT

Services

There are no services for you yet.  
Use the  in the bottom right corner to create a new Service.

Create a new Service

Creation Parameters

Factory name

Name of the Zabbix factory.

Factory size

Size of the Zabbix factory, based on the expected new values per second (NVPS).

Factory slug


Optional short machine name of the Zabbix factory. Allowed characters: a-z, 0-9 and -. Will be generated from factory name if left empty.

GitLab user name

Name of your GitLab account.

BACK CREATE SERVICE

Services

There are no services for you yet.  
Use the  in the bottom right corner to create a new Service.

Create a new Service

Creation Parameters

Factory name

My Zabbix

Name of the Zabbix factory.

Factory size

Medium (100-1000 NVPS)

Size of the Zabbix factory, based on the expected new values per second (NVPS).

Factory slug

my-zabbix

Optional short machine name of the Zabbix factory. Allowed characters: a-z, 0-9 and -. Will be generated from factory name if left empty.

GitLab user name

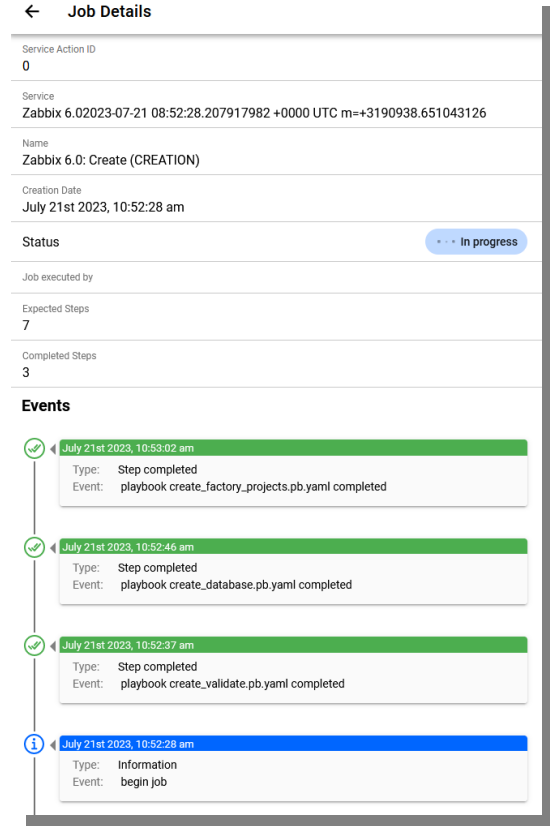
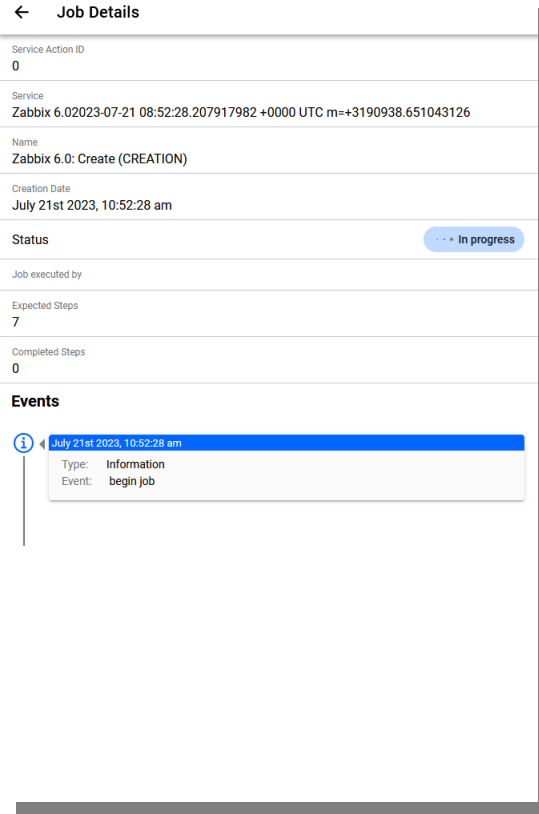
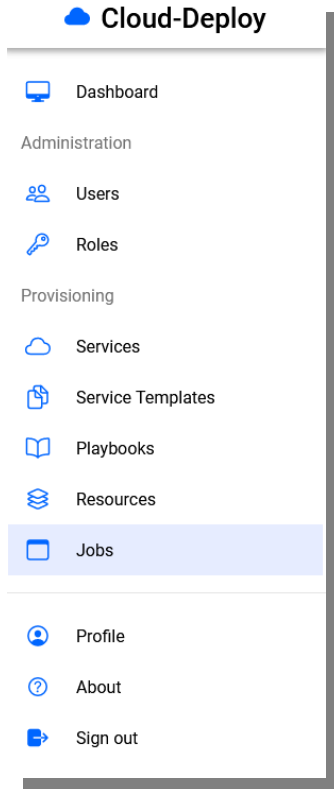
support@intellitrend.de

Name of your GitLab account.

BACK CREATE SERVICE



# Cloud Deploy – Job Status in real-time



# Cloud Deploy – Deployment Done

**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

Pods

2 items

Namespace: my-zabbix

Search Pods...

<input type="checkbox"/>	Name ^	Namespace	Containers	Restarts	Controlled By	Node	QoS	Age	Status	
<input type="checkbox"/>	my-zabbix-zabbix-server-67684bcf98...	my-zabbix	<div><div></div><div></div><div></div></div>	0	ReplicaSet	dev-k8s-node01	BestEffort	94m	Running	
<input type="checkbox"/>	my-zabbix-zabbix-web-5898d7f4b6-8...	my-zabbix	<div><div></div></div>	0	ReplicaSet	dev-k8s-node02	BestEffort	94m	Running	

Pods created on K8s cluster

**ZABBIX** My Zabbix

Global view

All dashboards / Global view

**System information**

Parameter	Value	Details
Zabbix server is running	Yes	my-zabbix-zabbix-server:10051
Number of hosts (enabled/disabled)	1	1 / 0
Number of templates	325	
Number of items (enabled/disabled/not supported)	120	112 / 0 / 8
Number of triggers	72	72 / 0 / 172

**Problems**

Time	Info	Host	Problem • Severity	Duration	Ack	Actions	Tags
No data found							

**System status summary:**

- 1 Available
- 0 Not available
- 0 Unknown
- 1 Total
- 0 Disaster
- 0 High
- 0 Average
- 0 Warning
- 0 Information
- 0 Not classified

**Favorite maps**

No maps added.

Zabbix frontend after login



**ZABBIX**  
PREMIUM PARTNER

# Zabbix meets Cloud Deploy

## Cloud Deploy APIs

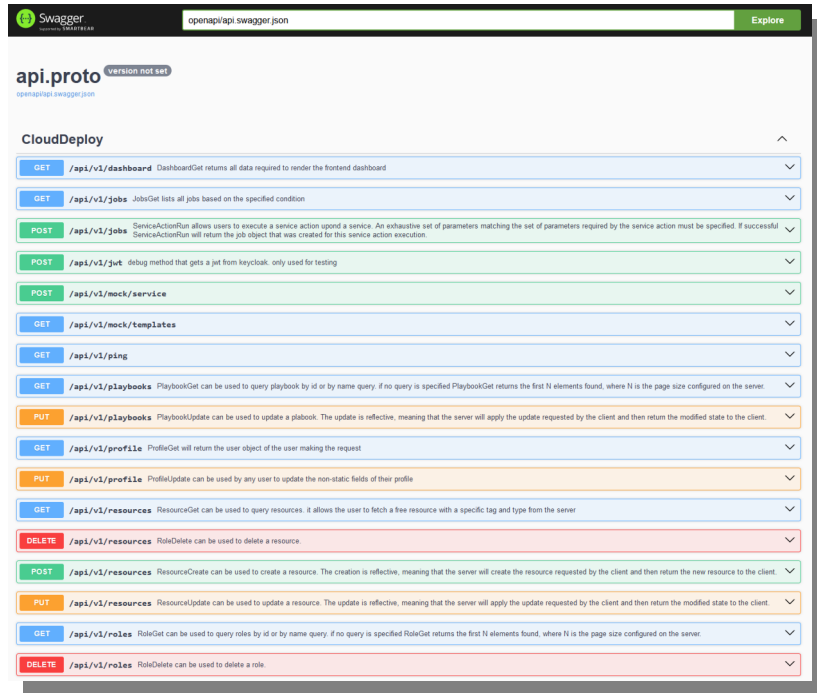
# Cloud Deploy - APIs

All operations performed in the User Portal (and more) can also be executed via APIs, making **automated provisioning** straightforward.

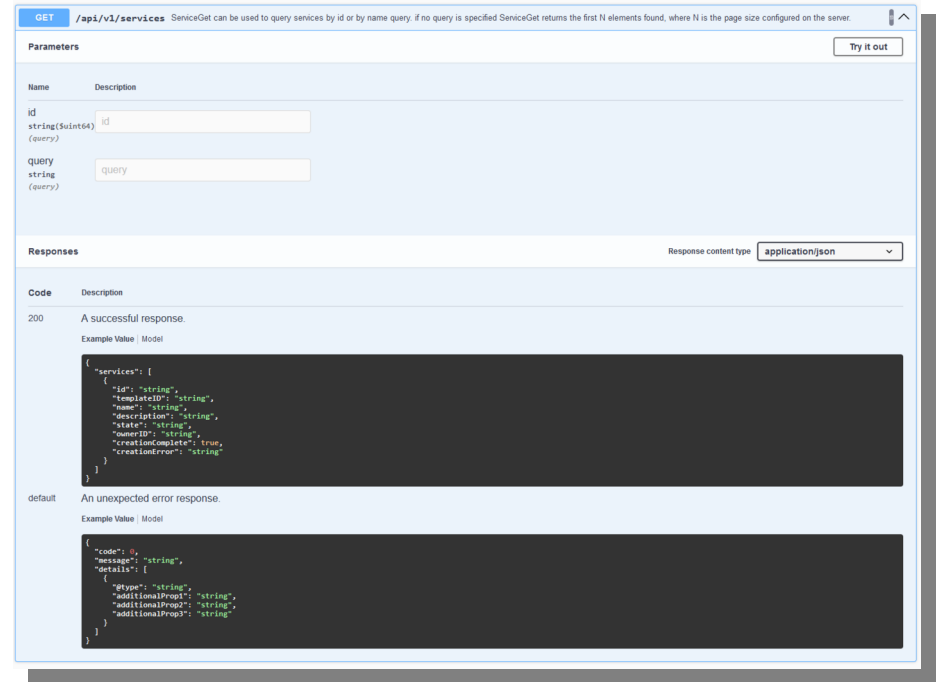
Available **APIs**:

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

# Cloud Deploy – REST-Like API



Documentation



Live testing with swagger

Based on OpenAPI specification, usable with Swagger - <https://swagger.io/>



**ZABBIX**  
PREMIUM PARTNER

# Cloud Deploy – Terraform Provider

```
terraform {  
  required_providers {  
    clouddeploy = {  
      source = "intellitrend.de/cloud-deploy/cloud-deploy"  
      version = "1.0.0"  
    }  
  }  
}  
  
# Configuration for the Cloud Deploy provider  
provider "clouddeploy" {  
  user      = "admin"  
  password  = "admin"  
  server_addr = "cloud-deploy.loc"  
  server_port = 8090  
}
```

Cloud Deploy can be used with Hashicorp Terraform – <https://www.terraform.io/>



# Cloud Deploy – Terraform Service Example

```
# 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"
```



## Check progress with “refresh”

# Zabbix meets Cloud Deploy

“One to rule them all”  
Monitoring of deployed  
Zabbix Instances

# Monitoring of deployed Zabbix Instances

- A central “Master” Zabbix Server monitors all deployed Zabbix Instances
- Each Zabbix Instance defines a service on the Master including Zabbix Server, Zabbix Agent and Frontend
- The Master Server has access to internal metrics of each Zabbix Instance
- Each Zabbix Instance should have a dedicated 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

Host IPMI Tags 2 Macros 4 Inventory Encryption

Discovered by Factory discovery

\* Host name my-zabbix - Zabbix Server

Visible name My Zabbix - Zabbix Server

Templates

Name

Remote Zabbix server health extended

Cloud Deploy Zabbix Server

\* Groups Cloud Deploy x Select

Interfaces

Type	IP address	DNS name	Connect to	Port	Default
Agent	{{SERVER_IP}}		IP DNS	10050	

Description

Monitored by proxy (no proxy)

Enabled

Update Clone Full clone Delete Cancel

Services provisioned by Cloud Deploy can be discovered via LLD and monitored by Zabbix using the Cloud Deploy API.

Host prototypes allow creation from LLD.



# Cloud Deploy – Service Definition per Instance

The screenshot shows the Zabbix web interface for 'IntelliTrend Advanced Services for Zabbix v6.1.0'. The left sidebar contains navigation links for Monitoring, Services, Inventory, Reports, Configuration, Administration, Support, Integrations, Help, User settings, and Sign out. The main content area displays a summary of services, triggers, and hosts, followed by a detailed view of the 'Zabbix Services' for 'My Zabbix'.

**Summary:**

- Services:** 1 (No Configuration Issues)
- Hosts:** 3 (No Configuration Issues)
- Triggers:** 3 (No Configuration Issues)
- Tag Names:** 3 (Host: SvcMonHostItem: SvcMonKey Limit: SvcMonVLimit)
- License:** Licensed services: unlimited (IntelliTrend GmbH)

**Services:**

1.) My Zabbix

Services: 1 Triggers: 3 Items: 0 Hosts: 3

Fit: 50% 75% 100% 125% 150% ☒ Show real host trigger expressions ☒ Show virtual host trigger expressions [Reload](#)

**Zabbix Services**

- My Zabbix - Zabbix Server**  
Zabbix server is down on My Zabbix - Zabbix Server  
max(/my-zabbix - Zabbix Server/net.tcp.service[tcp,,10051],#3)=0
- My Zabbix - Zabbix Frontend**  
Zabbix frontend is down on My Zabbix - Zabbix Frontend  
max(/my-zabbix - Zabbix Frontend/net.tcp.service[http,#3]=0
- My Zabbix - Zabbix Agent**  
Zabbix agent is down on My Zabbix - Zabbix Agent  
max(/my-zabbix - Zabbix Agent/net.tcp.service[tcp,,10050],#3)=0

Zabbix 6.0.17. © 2001–2023, Zabbix SIA

Service created on the Master for each Zabbix Instance with graphical presentation

# Cloud Deploy – Zabbix Instances on the Master

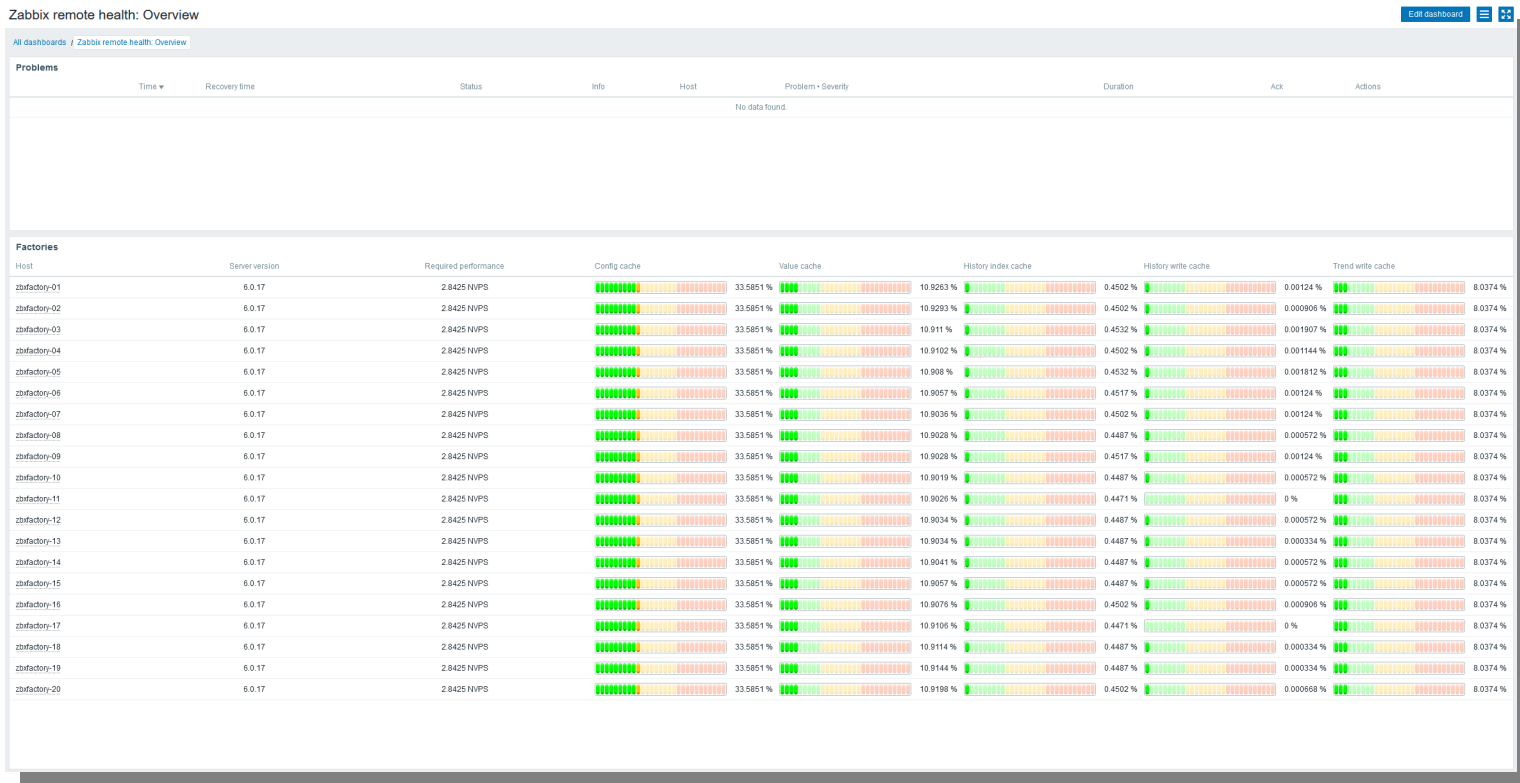
Hosts Create host Import

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"/> zbxfactory-01	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-02	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-03	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-04	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-05	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-06	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-07	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-08	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-09	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-10	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-11	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-12	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-13	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-14	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-15	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-16	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-17	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-18	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-19	Items 57	Triggers 42	Graphs 11	Discovery 1	Web			Remote Zabbix server health	Enabled		None		dashboard
<input type="checkbox"/> zbxfactory-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 Instances on the Master



Health status dashboard - Zabbix Instances on the Master



# Zabbix meets Cloud Deploy

—

## Intelligent Provisioning of multiple Zabbix Instances in Kubernetes



IntelliTrend GmbH

# Thank You!



Contact: Wolfgang Alper

 [www.intellitrend.de](http://www.intellitrend.de)



**ZABBIX**  
PREMIUM PARTNER

[wolfgang.alper@intellitrend.de](mailto:wolfgang.alper@intellitrend.de)