

Zabbix meets Kubernetes

Smart monitoring and self-deployment in cloud environments made easy



Contact: Wolfgang Alper

wolfgang.alper@intellitrend.de

www.intellitrend.de



 IntelliTrend IT-Services GmbH



ZABBIX
PREMIUM PARTNER

Zabbix meets Kubernetes

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-5.4-latest
```

More optional Env-Vars

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

Zabbix installation with Docker

Optional volumes to be mapped inside the container

```
/usr/lib/zabbix/alertscripts  
/usr/lib/zabbix/externalscripts  
/var/lib/zabbix/modules  
/var/lib/zabbix/enc  
/var/lib/zabbix/ssh_keys  
/var/lib/zabbix/ssl/certs  
/var/lib/zabbix/ssl/keys  
/var/lib/zabbix/ssl/ssl_ca  
/var/lib/zabbix/snmptraps  
/var/lib/zabbix/mibs  
/var/lib/zabbix/export
```

Zabbix installation with Docker

- Easy deployment
- Typically the Docker-Host is the machine to connect to for the service
- When using „Host-Volumes“, all the files are stored on that Docker-Host
- DB credentials like username and password must be available in cleartext
- Service URLs like
„ZBX_WEBSERVICEURL=http://zabbix-web-service:10053/report“
must be defined during provisioning

Zabbix meets Kubernetes

Zabbix installation with Kubernetes

Zabbix meets 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

Introducing Helm

- 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“
- Official Zabbix K8s Monitoring Helm Chart (no Zabbix-Server) as „Deployment“:
<https://git.zabbix.com/projects/ZT/repos/kubernetes-helm/browse>
- Inofficial Zabbix Deployment Helm Chart (with Zabbix-Server) as „StatefulSet“:
<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

Zabbix meets Kubernetes

Deployments vs StatefulSets

Deployments vs StatefulSets

Deployments

- Deployments are typically used for **stateless applications**
- Manages sets of identical pods (or ReplicaSets) using common configurations
- Ensures that managed pods meets the requirements set
- Provides granular control over the rollout of new pod versions and rollback to previous versions
- Deployments **can save a state** by **attaching a persistent volume** and making them stateful
- Deployed pods share the same volume, with the same data

```
{{- if and .Values.zabbixProxy.enabled -}}  
apiVersion: {{ template "zabbix.deployment.apiVersion" . }}  
kind: Deployment  
metadata:  
  name: zabbix-proxy  
  labels:  
    app: zabbix  
    tier: proxy
```



Deployments vs StatefulSets

StatefulSets

- StatefulSets are typically used for **stateful applications**
- Each pod is given a name and is treated individually
- Every pod has two unique stable identities: Network ID and Storage ID
 - Network ID: enables the pod's DNS name to persist across rescheduling (IP might differ)
 - Storage ID: retained regardless of the node the storage instance is rescheduled on
- Each replica has its **own state**, creating a **unique persistent volume claim** (PVC) for each pod
- A set requires a **headless service with an service IP** that returns the IPs of the associated pods

```
{{- 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
```



Deployment vs StatefulSets

```
{{- if and .Values.zabbixProxy.enabled -}}
apiVersion: {{ template "zabbix.deployment.apiVersion" . }}
kind: Deployment
metadata:
  name: zabbix-proxy
  labels:
    app: zabbix
    tier: proxy
spec:
  selector:
    matchLabels:
      name: zabbix-proxy
  replicas: 1
  template:
    metadata:
      labels:
        name: zabbix-proxy
        app: zabbix
    spec:
      containers:
        - name: zabbix-proxy
          image: "{{ .Values.zabbixProxy.image.repository }}"
          imagePullPolicy: {{ .Values.zabbixProxy.image.pullPolicy }}
          ports:
            - containerPort: {{ .Values.zabbixProxy.service.port }}
              protocol: TCP
              name: 10051-tcp
```

```
{{- 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
```

„Deployments“: stateless applications - „StatefulSets“: stateful applications

Zabbix meets Kubernetes

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!✨
[serveradmin@k8s-master demo]$
```

Create values_zabbix.yaml (skeleton with defaults):

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

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 in Postgresql
    postgresPassword: "zabbix_pwd"
    # -- User of database
    username: "zabbix"
    # -- Password of database
    password: "zabbix"
    # -- Name of database
    database: "zabbix"
```

```
GNU nano 2.9.8

# 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: "172.20.20.100"
  # -- Port of database host
  DB_SERVER_PORT: "5432"
  # -- User of database
  POSTGRES_USER: "zabbix"
  # -- Password of database
  POSTGRES_PASSWORD: "zabbix"
```

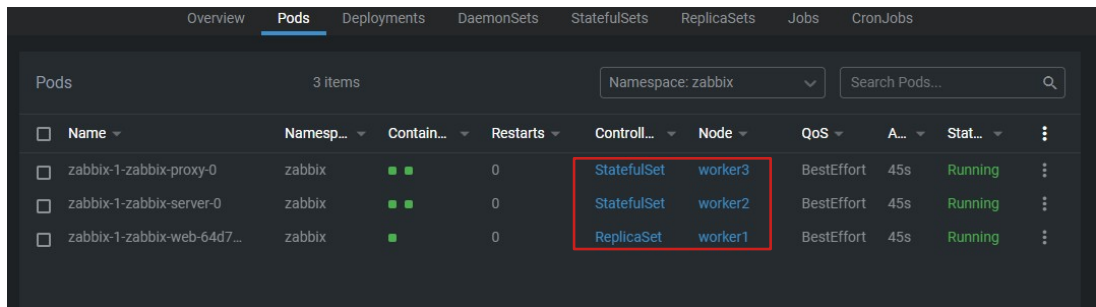


Installing Zabbix instance

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
[serveradmin@k8s-master demo]$
```

Pods for „zabbix-1“ after installation



Overview Pods Deployments DaemonSets StatefulSets ReplicaSets Jobs CronJobs									
Pods 3 items Namespace: zabbix Search Pods...									
☐	Name	Namesp...	Contain...	Restarts	Controll...	Node	QoS	A...	Stat...
☐	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

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 with database owner!
** 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

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-j99x (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-j99x (ro)

Name	Namespace	Containers
zabbix-1-zabbix-proxy-0	zabbix	■ ■
zabbix-1-zabbix-server-0	zabbix	■ ■
zabbix-1-zabbix-web-74678b8d45-qgbz5	zabbix	■

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

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-j99x (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:

Mounts: /var/run/secrets/kubernetes.io/serviceaccount from default-token-j99x (ro)

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



ZABBIX
PREMIUM PARTNER

Zabbix meets Kubernetes

Rescale a deployment

Rescaling Zabbix frontend deployment

Rescale zabbix-web deployment from 1 to 3 replicas:

```
[serveradmin@k8s-master demo]$ kubectl scale --replicas=3 -n zabbix deployment/zabbix-1-zabbix-web  
deployment.apps/zabbix-1-zabbix-web scaled  
[serveradmin@k8s-master demo]$
```

Additional replicas are being deployed:

☐	zabbix-1-zabbix-web-64d754d84c-7...	zabbix		0	ReplicaSet	worker3	BestEffort	8s	Pending
☐	zabbix-1-zabbix-web-64d754d84c-n...	zabbix		0	ReplicaSet	worker2	BestEffort	8s	Pending
☐	zabbix-1-zabbix-web-64d754d84c-v...	zabbix		0	ReplicaSet	worker1	BestEffort	18m	Running

Service for zabbix-web now picks a random available pod when accessing the web-frontend

☐	zabbix-1-zabbix-web-64d754d84c-7...	zabbix		0	ReplicaSet	worker3	BestEffort	91s	Running
☐	zabbix-1-zabbix-web-64d754d84c-n...	zabbix		0	ReplicaSet	worker2	BestEffort	91s	Running
☐	zabbix-1-zabbix-web-64d754d84c-v...	zabbix		0	ReplicaSet	worker1	BestEffort	19m	Running



ZABBIX
PREMIUM PARTNER

Because of its app design, Zabbix Web-Frontend runs great with replicas

Zabbix service accessibility

Services

4 items

Namespace: zabbix

Search Services...

☐	Name	Namespace	Type	Cluster IP	Ports	External IP	Selector	Age	Status
☐	zabbix-1-zabbix-agent	zabbix	ClusterIP	10.103.144.47	10050/TCP	-	app=zabbix-1-zab...	9m20s	Active
☐	zabbix-1-zabbix-proxy	zabbix	ClusterIP	10.100.118.101	10051/TCP	-	app=zabbix-1-zab...	9m20s	Active
☐	zabbix-1-zabbix-serv...	zabbix	ClusterIP	10.99.175.156	10051/TCP, 10052/...	-	app=zabbix-1-zab...	9m20s	Active
☐	zabbix-1-zabbix-web	zabbix	NodePort	10.106.242.110	80:32411/TCP	-	app=zabbix-1-zab...	9m20s	Active

Using the default Helm Chart service setup so far:

- Web frontend has assigned a **node port 32411** (random port between 30000 and 32768) and can be accessed via any node from the outside:
 - <http://172.20.22.100:32411/>
 - <http://172.20.22.101:32411/>
 - <http://172.20.22.102:32411/>
 - <http://172.20.22.103:32411/>
- Agent, proxy and server are ClusterIP services - only allow **cluster-internal communication**

How to make the Zabbix-Server accessible from outside?

Zabbix meets Kubernetes

Make services accessible
from outside the cluster

Make services accessible from outside

- Some services are only accessible inside the cluster
- Some services use „unusual“ ports per default
- Non-HTTP based TCP / UDP services cannot be easily routed

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

MetallLB

- Provides a load balancer usable in self-hosted cloud environments
- Assigns IP addresses and ports to services

Add Helm repo and install MetallLB to the cluster

```
[serveradmin@k8s-master demo]$ helm repo add metallb https://metallb.github.io/metallb
"metallb" 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!
[serveradmin@k8s-master demo]$ helm install metallb metallb/metallb --namespace metallb --create-namespace -f metallb_values.yaml --version "0.11.0"
NAME: metallb
LAST DEPLOYED: Thu Apr 14 12:08:35 2022
NAMESPACE: metallb
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
MetallLB is now running in the cluster.
LoadBalancer Services in your cluster are now available on the IPs you
defined in MetallLB's configuration:

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

To see IP assignments, try `kubectl get services`.
[serveradmin@k8s-master demo]$
```

Config with IP-range

```
[serveradmin@k8s-master demo]$ cat metallb_values.yaml
configInline:
  address-pools:
  - name: default
    protocol: layer2
  addresses:
  - 172.20.22.110-172.20.22.120
```



Reconfigure Zabbix for MetalLB

Adjust zabbix_values.yaml to use 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
[serveradmin@k8s-master demo]$
```



ZABBIX
PREMIUM PARTNER

Zabbix Services via MetalLB

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

Services									
4 items					Namespace: zabbix		Search Services...		
☐	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 is accessible outside the cluster using:

- External IP 172.20.22.110
- Port 10051

Zabbix meets Kubernetes

Real world challenges

Kubernetes - Real world challenges

When installing applications in a kubernetes cluster you might face some more challenges

- How to ensure a consistent application configuration?
- How to track configuration changes?
- How to manage passwords/secrets in configuration files?

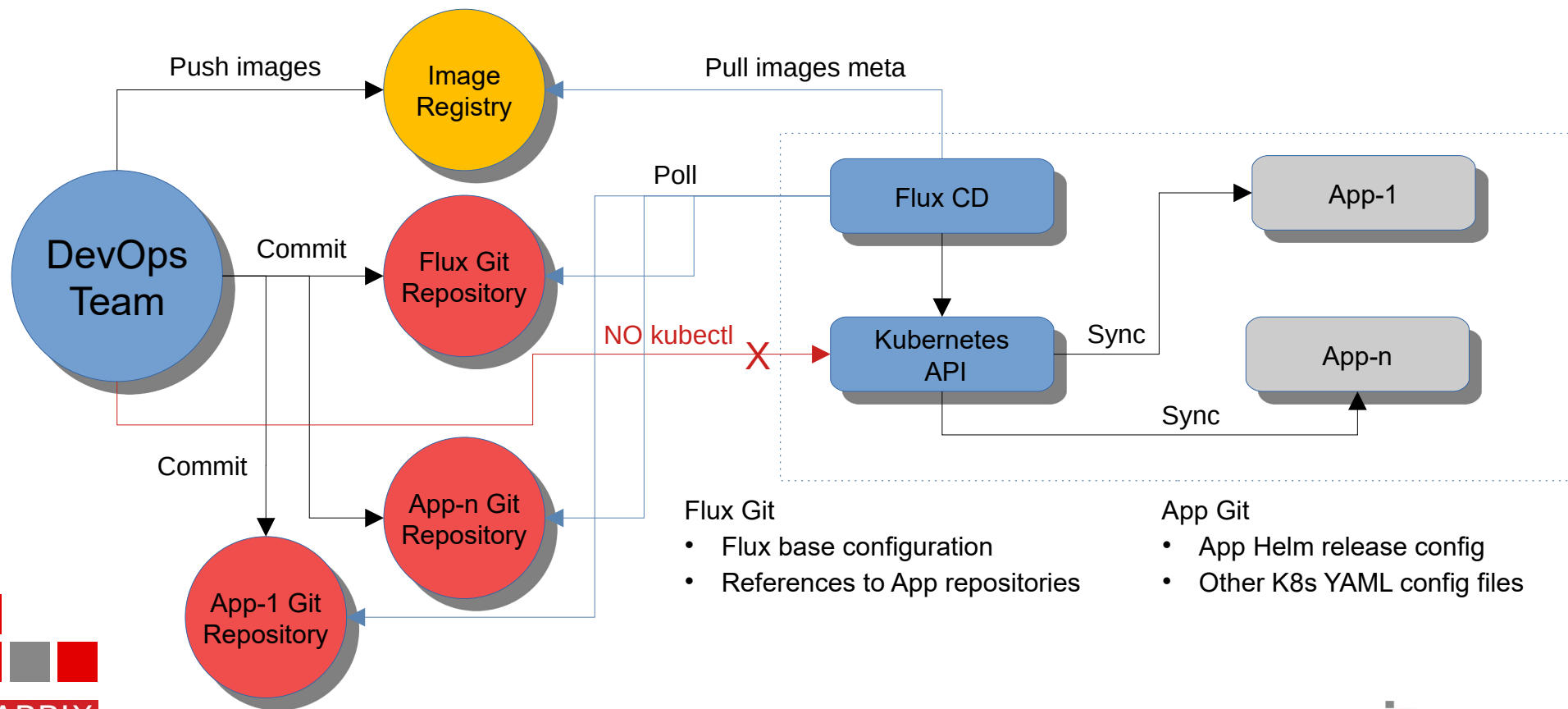


Introducing Flux

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 „kubectl create“, changes are committed and pushed to Git repositories
- Flux polls the Git repositories for changes and performs automatic rollouts in the cluster when required

Real world challenges - Flux/Git



Real world challenges - SOPS

Mozilla **SOPS** 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, Azure, GCS 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 – 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+1pY5wgp0dGYl3NG4LjKGirOzk=,tag:UTTRhSgW4Wbm79boaefAxQ==,type:str]
password: ENC[AES256_GCM,data:+Fd9GlgulMhB8o13s,iv:Yd47/6YPKNJwJl17sso7nymAeanQ3hiERXLw4Nrx/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/6Bz43V28PIhqSLIC
        xLPkR0yqXU4Bh0zp11bLsviDLz+msAny713T91y9thS/VOSbwj6mhZhc2CvTAL7g
        8CuXcAhGk70vTh8DH7N/L3zE5514wLJ0+gnbn7CroxpoVPLbliite07qf3zIYBdv
        xGDBab0jUua/Y7Z/w0456IA6RMpB9vEe9WR1GE5iItt8a/BAsi9pWmG9vXN3UPUD
        Ub4pWl+c3REqmxShVuz0sDQLCerBqk3F4PTIXozCuWkVz44kKwD4WicgZC3dFFR
        Ot/JVkpDhHYjt0c1Q4AsIUayXVziKbwxanCTgN/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$
```

SOPS allows to manage and store secrets in publicly available text files

Zabbix meets Kubernetes

Zabbix Kubernetes Monitoring Overview

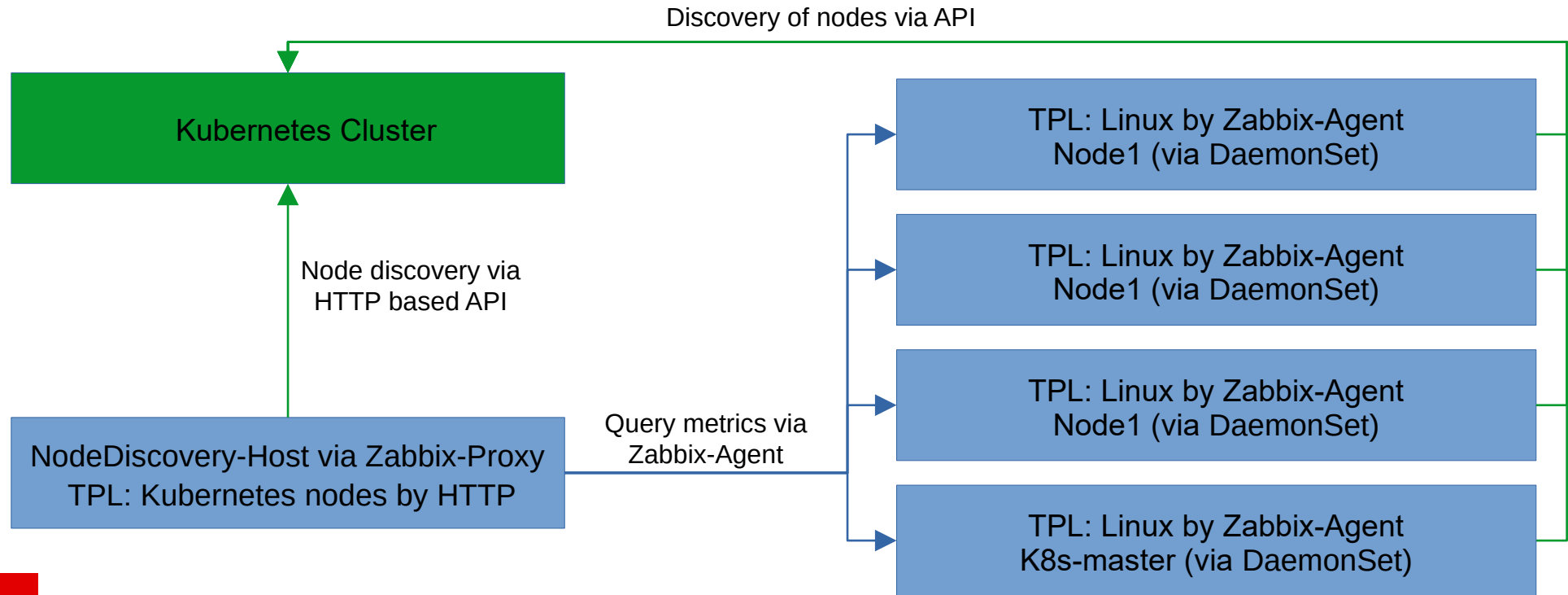
Kubernetes Monitoring

Zabbix 6.0 LTS added Kubernetes monitoring

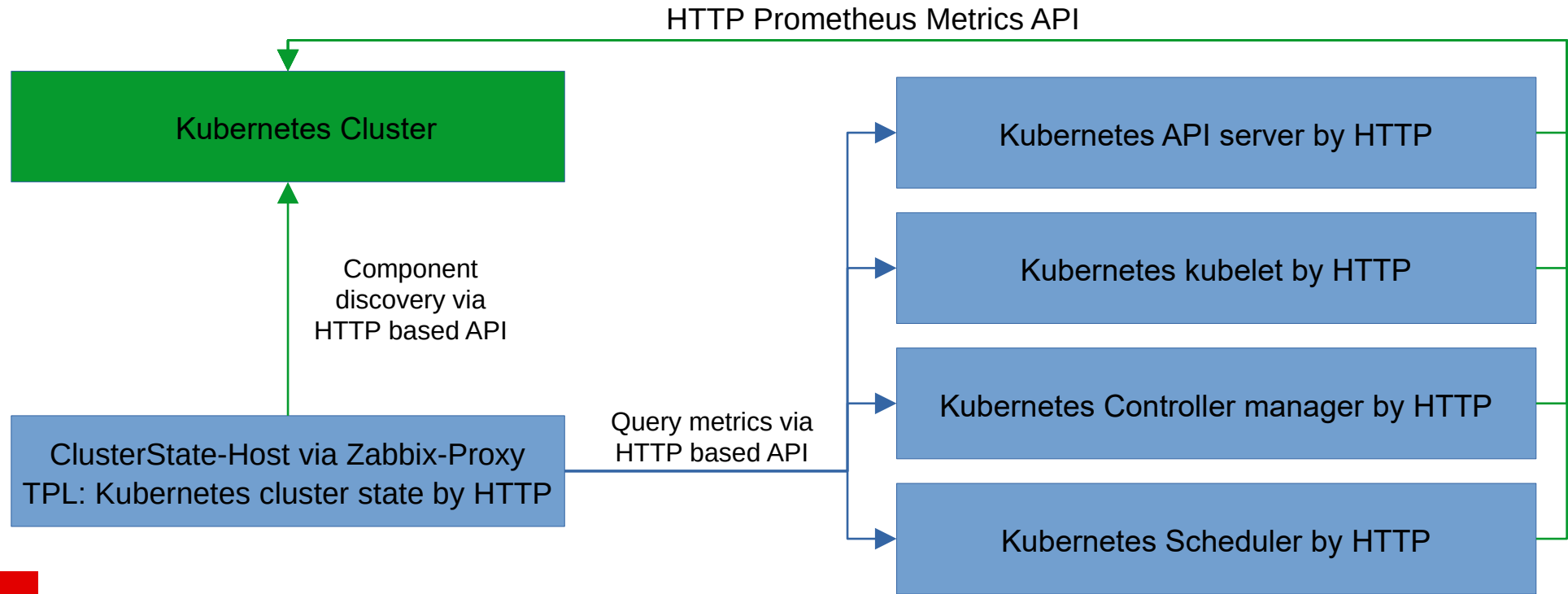
- Cluster node monitoring - **Kubernetes nodes by HTTP**
 - Item and host discovery via Kubernetes API
 - Monitors individual nodes through Zabbix Agent, deployed via a Kubernetes „DaemonSet“
 - Triggers for high resource utilization and resource limit violations
- Cluster component monitoring - **Kubernetes cluster state by HTTP**
 - Item discovery via Kubernetes API
 - Monitors K8s objects: deployments, daemonsets, replicaset, pods, services, endpoints etc.
 - Triggers for unhealthy pods
- More Kubernetes Templates
 - Kubernetes API server by HTTP, Kubernetes Controller manager by HTTP
 - Kubernetes Scheduler by HTTP, Kubernetes kubelet by HTTP



TPL - Kubernetes nodes by HTTP



TPL - Kubernetes cluster state by HTTP



Zabbix meets Kubernetes

Zabbix Kubernetes Monitoring Setup

Kubernetes Monitoring - Setup

Install the repo, install and configure the Zabbix Monitoring Helm Chart:

```
[serveradmin@k8s-master demo]$ helm repo add zabbix-chart-6.0 https://cdn.zabbix.com/zabbix/integrations/kubernetes-helm/6.0
"zabbix-chart-6.0" 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 "zabbix-chart-6.0" 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!✨
[serveradmin@k8s-master demo]$ helm show values zabbix-chart-6.0/zabbix-helm-chrt > ../
.ansible/      .bash_logout  .bashrc       .config/      .docker/      .gitconfig    .kube/        metallb/      .python_history .ssh/          .wget-hsts
.bash_history  .bash_profile .cache/       demo/         git/          hello-kubernetes/ .local/      .pylint.d/   shared/        .vscode-server/
[serveradmin@k8s-master demo]$ helm show values zabbix-chart-6.0/zabbix-helm-chrt > zabbix_monitoring_values.yaml
[serveradmin@k8s-master demo]$ helm install zabbix zabbix-chart-6.0/zabbix-helm-chrt -n monitoring --create-namespace --dependency-update -f zabbix_
zabbix_monitoring_values.yaml zabbix values original.yaml zabbix values.yaml
[serveradmin@k8s-master demo]$ helm install zabbix zabbix-chart-6.0/zabbix-helm-chrt -n monitoring --create-namespace --dependency-update -f zabbix_monitoring_values.yaml
NAME: zabbix
LAST DEPLOYED: Thu Apr 14 13:09:47 2022
NAMESPACE: monitoring
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Thank you for installing zabbix-helm-chrt.

Your release is named zabbix.
Zabbix agent installed: "zabbix/zabbix-agent2:alpine-6.0.2"
Zabbix proxy installed: "zabbix/zabbix-proxy-sqlite3:alpine-6.0.2"
```

```
## If ProxyMode is set to passive mode:
## List of comma delimited IP addresses, optionally in CIDR notation,
## Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.ex
- name: ZBX_SERVER_HOST
  value: "zabbix-1-zabbix-server.zabbix"
## Zabbix server port
- name: ZBX_SERVER_PORT
  value: 10051
## The variable is used to specify debug level. By default, value is 3
- name: ZBX_DEBUGLEVEL
  value: 3
## Cache size
- name: ZBX_CACHESIZE
  value: 128M
```

Before install, configure the Zabbix-Server



ZABBIX
PREMIUM PARTNER

Kubernetes Monitoring – Access API

Obtaining the token for the Zabbix service account via command line from the secret „zabbix-service-account“

```
[serveradmin@k8s-master demo]$ kubectl get secret zabbix-service-account -n monitoring -o jsonpath={.data.token} | base64 -d && echo
```

```
[serveradmin@k8s-master demo]$
```

- Inside the cluster, the Kubernetes API server is reachable via:
<https://kubernetes.default/api>
- DNS schema: `<service name>.<namespace>`
- Authorization via token in HTTP header example:
„Authorization: Bearer eyJhbGciOiJSUzI1NiIsI...

Kubernetes Monitoring – Register Proxy

Adding the K8s proxy to the Zabbix config.

Proxies

Proxy

Encryption

* Proxy name

zabbix-k8s-proxy

Proxy mode

Active

Passive

Proxy address

Description

Update

Clone

Delete

Cancel

Proxies

Create proxy

Name

Mode

Any

Active

Passive

Apply

Reset

☐	Name ▲	Mode	Encryption	Compression	Last seen (age)	Host count	Item count	Required performance (vps)	Hosts
☐	zabbix-k8s-proxy	Active	None	On	1s	0	0	0	

Displaying 1 of 1 found

0 selected

Enable hosts

Disable hosts

Delete

Kubernetes Monitoring – NodeDiscovery

Creating new host for Kubernetes nodes, setup template and macros

The image displays two screenshots of the Zabbix web interface, illustrating the steps to create a new host for Kubernetes nodes.

Left Screenshot: 'New host' form

- Host name:** Kubernetes Nodes
- Visible name:** Kubernetes Nodes
- Templates:** Kubernetes nodes by HTTP (highlighted with a red box)
- Groups:** Kubernetes
- Interfaces:** A table with columns: Type, IP address, DNS name, Connect to, Port. The 'Agent' interface is configured with IP address 127.0.0.1, Connect to IP, and Port 10050.
- Description:** (Empty text area)
- Monitored by proxy:** (no proxy)
- Enabled:** ☒

Right Screenshot: 'New host' - Macros tab

- Host macros:** Inherited and host macros
- Macro:** (\$KUBE.API.ENDPOINT)
- Effective value:** https://kubernetes.default/ap...
- Template value:** Kubernetes nodes by HTTP: *<PUT YOUR KUBERN...
- Macro:** (\$KUBE.API.TOKEN)
- Effective value:** (Empty text area)
- Template value:** Kubernetes nodes by HTTP: *
- Service account bearer token:** (Empty text area)



Kubernetes nodes by HTTP

☐ Name ▲	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
☐ Kubelet discovery: Kubelet k8s-master	Items 324	Triggers	Graphs 16	Discovery 4	Web	172.20.22.100:10050		Kubernetes kubelet by HTTP	Enabled	ZBX	None		addr: 172.20.22.100
☐ Kubelet discovery: Kubelet worker1	Items 357	Triggers	Graphs 18	Discovery 4	Web	172.20.22.101:10050		Kubernetes kubelet by HTTP	Enabled	ZBX	None		addr: 172.20.22.101
☐ Kubelet discovery: Kubelet worker2	Items 316	Triggers	Graphs 16	Discovery 4	Web	172.20.22.102:10050		Kubernetes kubelet by HTTP	Enabled	ZBX	None		addr: 172.20.22.102
☐ Kubelet discovery: Kubelet worker3	Items 332	Triggers	Graphs 17	Discovery 4	Web	172.20.22.103:10050		Kubernetes kubelet by HTTP	Enabled	ZBX	None		addr: 172.20.22.103
☐ Kubernetes Cluster	Items 889	Triggers 147	Graphs	Discovery 15	Web	127.0.0.1:10050		Kubernetes cluster state by HTTP	Enabled	ZBX	None		
☐ Kubernetes Nodes	Items 248	Triggers 71	Graphs	Discovery 3	Web	127.0.0.1:10050		Kubernetes nodes by HTTP	Enabled	ZBX	None		
☐ Zabbix server	Items 124	Triggers 73	Graphs 23	Discovery 4	Web	127.0.0.1:10050		Linux by Zabbix agent, Zabbix server health	Enabled	ZBX	None		

Displaying 7 of 7 found

Nodes are discovered and added automatically as hosts in Zabbix

That manually created host discovers additional items

Kubernetes Monitoring – ClusterState

Creating new host for Kubernetes cluster state, setup template and macros

New host

Host IPMI Tags Macros 3 Inventory Encryption Value mapping

* Host name: Kubernetes Cluster

Visible name: Kubernetes Cluster

Templates: **Kubernetes cluster state by HTTP** Select

* Groups: Kubernetes Select

Interfaces

Type	IP address	DNS name	Connect to	Port	Default
Agent	127.0.0.1		IP DNS	10050	☒

Add

Description

Monitored by proxy: (no proxy)

Enabled: ☒

New host

Host IPMI Tags Macros 3 Inventory Encryption Value mapping

Host macros Inherited and host macros

Macro	Effective value	Template value	Global value (configure)
{KUBE.API.HOST}	kubernetes.default	T Remove = Kubernetes cluster state by HTTP: "<PUT YOUR KU...	
Kubernetes API host			
{KUBE.API.PORT}	443	T Remove = Kubernetes cluster state by HTTP: "6443"	
Kubernetes API port			
{KUBE.API.TOKEN}	Service account bearer token	T Remove = Kubernetes cluster state by HTTP: ""	



Kubernetes cluster state by HTTP

	Name ▲	Triggers	Key	Interval	History	Trends	Type	Status	Tags
☐	... Replicaset discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [kube-system] Replicaset [coredns-f9fd979d6] Replicas	Triggers 1	kube.replicaset.replicas[kube-system/coredns-f9fd979d6]	7d	365d	Dependent item	Enabled	component: replicaset namespace: kube-system	
☐	... Daemonset discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Daemonset [metallb-speaker] Desired		kube.daemonset.desired[metallb/metallb-speaker]	7d	365d	Dependent item	Enabled	component: daemonset daemonset: metallb-speaker	
☐	... Daemonset discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Daemonset [metallb-speaker] Misscheduled		kube.daemonset.misschedule[metallb/metallb-speaker]	7d	365d	Dependent item	Enabled	component: daemonset daemonset: metallb-speaker	
☐	... Daemonset discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Daemonset [metallb-speaker] Ready		kube.daemonset.ready[metallb/metallb-speaker]	7d	365d	Dependent item	Enabled	component: daemonset daemonset: metallb-speaker	
☐	... Daemonset discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Daemonset [metallb-speaker] Scheduled		kube.daemonset.scheduled[metallb/metallb-speaker]	7d	365d	Dependent item	Enabled	component: daemonset daemonset: metallb-speaker	
☐	... Daemonset discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Daemonset [metallb-speaker] Updated number scheduled		kube.daemonset.update[metallb/metallb-speaker]	7d	365d	Dependent item	Enabled	component: daemonset daemonset: metallb-speaker	
☐	... Deployment discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Deployment [metallb-controller] Paused		kube.deployment.spec_paused[metallb/metallb-controller]	7d	365d	Dependent item	Enabled	component: deployment daemonset: metallb-controller	
☐	... Deployment discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Deployment [metallb-controller] Replicas	Triggers 1	kube.deployment.replicas[metallb/metallb-controller]	7d	365d	Dependent item	Enabled	component: deployment daemonset: metallb-controller	
☐	... Deployment discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Deployment [metallb-controller] Replicas available	Triggers 1	kube.deployment.replicas_available[metallb/metallb-controller]	7d	365d	Dependent item	Enabled	component: deployment daemonset: metallb-controller	
☐	... Deployment discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Deployment [metallb-controller] Replicas desired		kube.deployment.replicas_desired[metallb/metallb-controller]	7d	365d	Dependent item	Enabled	component: deployment daemonset: metallb-controller	
☐	... Deployment discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Deployment [metallb-controller] Replicas unavailable		kube.deployment.replicas_unavailable[metallb/metallb-controller]	7d	365d	Dependent item	Enabled	component: deployment daemonset: metallb-controller	
☐	... Deployment discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Deployment [metallb-controller] Replicas updated		kube.deployment.replicas_updated[metallb/metallb-controller]	7d	365d	Dependent item	Enabled	component: deployment daemonset: metallb-controller	
☐	... Deployment discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Deployment [metallb-controller] Rollingupdate max unavailable		kube.deployment.rollingupdate_max_unavailable[metallb/metallb-controller]	7d	365d	Dependent item	Enabled	component: deployment daemonset: metallb-controller	
☐	... Pod discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Pod [metallb-controller-5488f4c94b-1s7k9] Containers CPU limits		kube.pod.containers.limits.cpu[metallb/metallb-controller-5488f4c94b-1s7k9]	7d	365d	Dependent item	Enabled	component: pod namespace: metallb	
☐	... Pod discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Pod [metallb-controller-5488f4c94b-1s7k9] Containers CPU requests		kube.pod.containers.requests.cpu[metallb/metallb-controller-5488f4c94b-1s7k9]	7d	365d	Dependent item	Enabled	component: pod namespace: metallb	
☐	... Pod discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Pod [metallb-controller-5488f4c94b-1s7k9] Containers memory limits		kube.pod.containers.limits.memory[metallb/metallb-controller-5488f4c94b-1s7k9]	7d	365d	Dependent item	Enabled	component: pod namespace: metallb	
☐	... Pod discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Pod [metallb-controller-5488f4c94b-1s7k9] Containers memory requests		kube.pod.containers.requests.memory[metallb/metallb-controller-5488f4c94b-1s7k9]	7d	365d	Dependent item	Enabled	component: pod namespace: metallb	
☐	... Pod discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Pod [metallb-controller-5488f4c94b-1s7k9] Containers ready		kube.pod.containers_ready[metallb/metallb-controller-5488f4c94b-1s7k9]	7d	365d	Dependent item	Enabled	component: pod namespace: metallb	
☐	... Pod discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Pod [metallb-controller-5488f4c94b-1s7k9] Containers restarts	Triggers 1	kube.pod.containers_restarts[metallb/metallb-controller-5488f4c94b-1s7k9]	7d	365d	Dependent item	Enabled	component: pod namespace: metallb	
☐	... Pod discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Pod [metallb-controller-5488f4c94b-1s7k9] Containers running		kube.pod.containers_running[metallb/metallb-controller-5488f4c94b-1s7k9]	7d	365d	Dependent item	Enabled	component: pod namespace: metallb	
☐	... Pod discovery : Kubernetes: Get state metrics : Kubernetes: Namespace [metallb] Pod [metallb-controller-5488f4c94b-1s7k9] Containers terminated		kube.pod.containers_terminated[metallb/metallb-controller-5488f4c94b-1s7k9]	7d	365d	Dependent item	Enabled	component: pod namespace: metallb	

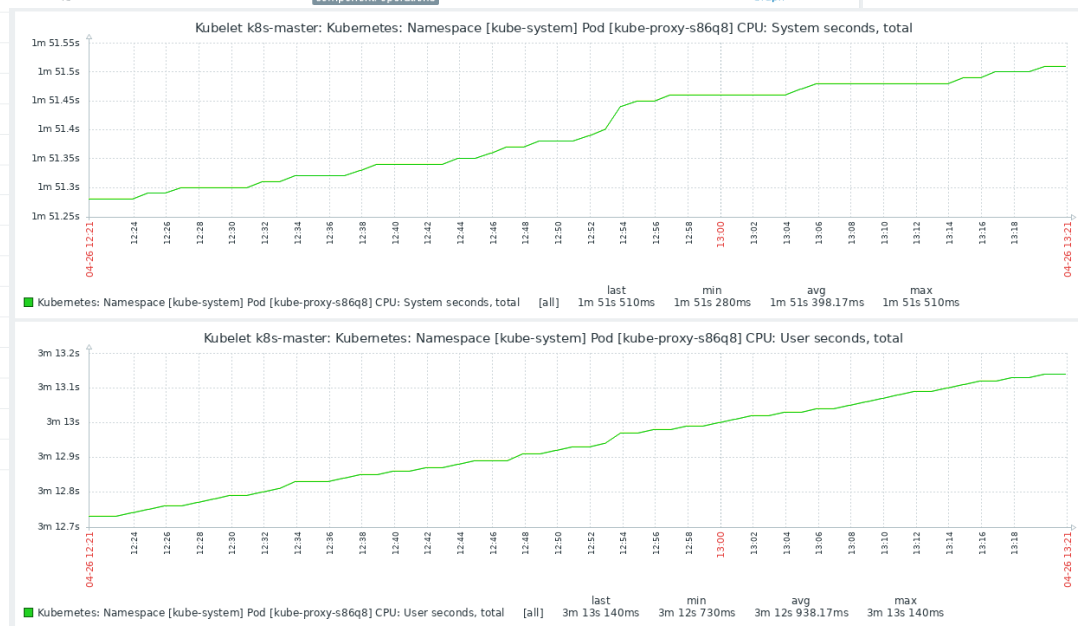
Depending on the complexity of the cluster, several thousand items may get discovered

Filtering via namespace macro is recommended!



Latest data: Kubelet k8s-master

☐ Host	Name ▲	Last check	Last value	Change	Tags	Info
☐ Kubelet k8s-master	Kubernetes: Pods running [?]	40s	10		component pods	Graph
☐ Kubelet k8s-master	Kubernetes: Virtual memory, bytes [?]	40s	1.77 GB		component memory	Graph
☐ Kubelet k8s-master	Kubernetes: [container_status] Operations, p50 [?]	22s	-1s		component operations	Graph
☐ Kubelet k8s-master	Kubernetes: [container_status] Operations, p90 [?]	6s	-1s		component operations	Graph
☐ Kubelet k8s-master	Kubernetes: [container_status] Operations, p95 [?]	50s				
☐ Kubelet k8s-master	Kubernetes: [container_status] Operations, p99 [?]	34s				
☐ Kubelet k8s-master	Kubernetes: [container_status] Runtime operations bucket +Inf [?]	40s				
☐ Kubelet k8s-master	Kubernetes: [container_status] Runtime operations bucket 0.005 [?]	40s				
☐ Kubelet k8s-master	Kubernetes: [container_status] Runtime operations bucket 0.01 [?]	40s				
☐ Kubelet k8s-master	Kubernetes: [container_status] Runtime operations bucket 0.025 [?]	40s				
☐ Kubelet k8s-master	Kubernetes: [container_status] Runtime operations bucket 0.05 [?]	40s				
☐ Kubelet k8s-master	Kubernetes: [container_status] Runtime operations bucket 0.1 [?]	40s				
☐ Kubelet k8s-master	Kubernetes: [container_status] Runtime operations bucket 0.5 [?]	40s				
☐ Kubelet k8s-master	Kubernetes: [container_status] Runtime operations bucket 0.25 [?]	40s				
☐ Kubelet k8s-master	Kubernetes: [container_status] Runtime operations bucket 1 [?]	40s				
☐ Kubelet k8s-master	Kubernetes: [container_status] Runtime operations bucket 2.5 [?]	40s				
☐ Kubelet k8s-master	Kubernetes: [container_status] Runtime operations bucket 5 [?]	40s				
☐ Kubelet k8s-master	Kubernetes: [container_status] Runtime operations bucket 10 [?]	40s				
☐ Kubelet k8s-master	Kubernetes: [container_status] Runtime operations total, rate [?]	40s				



Zabbix meets Kubernetes

Smart monitoring and self-deployment in cloud environments made easy



Questions?

Contact: Wolfgang Alper

wolfgang.alper@intellitrend.de

www.intellitrend.de



 IntelliTrend IT-Services GmbH



ZABBIX
PREMIUM PARTNER

Zabbix meets Kubernetes

Smart monitoring and self-deployment in cloud environments made easy

Thank you!



Contact: Wolfgang Alper

wolfgang.alper@intellitrend.de

www.intellitrend.de



 IntelliTrend IT-Services GmbH



ZABBIX
PREMIUM PARTNER