Install Mertrics Server on Kubernetes

> This guide shows how to install metrics server on kubernetes.
>
> Kubernetes Version: `1.23.1`
>
> Metrics Server Version: `v0.5.2`
>
> Reference:
>
> https://github.com/kubernetes-sigs/metrics-server

Metrics Server collects resource metrics from Kubelets and exposes them in Kubernetes apiserver through Metrics API , we can use it **monitor the workload of CPU and Memory of each Pod in Kubernetes cluster**.

1. Get the offical Yaml manifest

To install the latest Metrics Server release from the `components.yaml` manifest, run the following command.

```
# do not run this at the moment, since we nee to make some changes to the file
kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
```

We can download this `components.yaml` locally in the **master node**.

```
# create the folder for metrics-server
mkdir metrics-server
cd metrics-server

# download specific version (latest is v0.5.2 at 2021-12-20)
wget https://github.com/kubernetes-sigs/metrics-server/releases/download/v0.5.2/components.yaml
```

2. Update `components.yaml`

- In the `Deployment` section, add `--kubelet-insecure-tls` option
- Update this option to `--kubelet-preferred-address-types=InternalIP`
- The updated YAML manifest now looks like

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    k8s-app: metrics-server
  name: metrics-server
  namespace: kube-system
spec:
  selector:
    matchLabels:
      k8s-app: metrics-server
  strategy:
    rollingUpdate:
      maxUnavailable: 0
  template:
    metadata:
      labels:
        k8s-app: metrics-server
    spec:
      containers:
      - args:
        - --cert-dir=/tmp
        - --secure-port=4443
        - --kubelet-preferred-address-types=InternalIP # change this line
        - --kubelet-insecure-tls # add this line
        - --kubelet-use-node-status-port
        - --metric-resolution=15s
        image: k8s.gcr.io/metrics-server/metrics-server:v0.5.2
        imagePullPolicy: IfNotPresent
        livenessProbe:
          failureThreshold: 3
          httpGet:
            path: /livez
            port: https
            scheme: HTTPS
          periodSeconds: 10
        name: metrics-server
        ports:
        - containerPort: 4443
          name: https
          protocol: TCP
        readinessProbe:
          failureThreshold: 3
          httpGet:
            path: /readyz
            port: https
            scheme: HTTPS
          initialDelaySeconds: 20
          periodSeconds: 10
        resources:
          requests:
            cpu: 100m
            memory: 200Mi
        securityContext:
          readOnlyRootFilesystem: true
          runAsNonRoot: true
          runAsUser: 1000
        volumeMounts:
        - mountPath: /tmp
          name: tmp-dir
      nodeSelector:
        kubernetes.io/os: linux
      priorityClassName: system-cluster-critical
      serviceAccountName: metrics-server
      volumes:
      - emptyDir: {}
        name: tmp-dir
```

3. Pull Images [optional]

In case some images cannot be pulled during runtime, we can pull them at first.

*Make sure you pull images on **all the VM nodes**.*

```
# check images and we can see the image used in the spec
grep image components.yaml

        image: k8s.gcr.io/metrics-server/metrics-server:v0.5.2
        imagePullPolicy: IfNotPresent

# Pull the image
docker pull k8s.gcr.io/metrics-server/metrics-server:v0.5.2
```

If you cannot pull it from original repo, you can try below workaroud.

```
# The image is actually synced from k8s.gcr.io/metrics-server/metrics-server
# You can find other similar image repos, just choose the one you want.
docker pull k8simage/metrics-server:v0.5.2

# tag it to the exact image name k8s.gcr.io/metrics-server/metrics-server:v0.5.2
docker image tag k8simage/metrics-server:v0.5.2 k8s.gcr.io/metrics-server/metrics-
server:v0.5.2

# untag the original name
docker image rm -f k8simage/metrics-server:v0.5.2

# Now you should have the image pulled
docker images | grep metrics-server
```

4. Install metrics server and verify the installation

In the master node, run below commands

```
# 1. Install metrics-server
kubectl apply -f components.yaml

# 2. Check metrics-server pod
kubectl get pod -n kube-system | grep metrics-server

# 3. Check metrics-server service
kubectl describe svc metrics-server -n kube-system
```

5. Use metrics server

```
# 1. Check workload of nodes
kubectl top nodes

# 2. Check workload of pods
kubectl top pods -n kube-system
```