

版权声明，本文档全部内容及版权归“张岩峰”老师所有，只可用于自己学习使用，**禁止私自传阅，违者依法追责。**

## 基于 kubernetes 部署 Metrics Server 度量指标组件

张岩峰老师微信，加我微信，邀请你加入 VIP 交流答疑群：

微信号：ZhangYanFeng0429

二维码：



### 1、Metrics Server 原理介绍

metrics-server 官网地址：

<https://github.com/kubernetes-sigs/metrics-server>

#### ● 概念

介绍 Metrics Server 前首先介绍下 Heapster，该工具是用于 Kubernetes 集群监控和性能分析工具，可以收集节点上的指标数据，例如：节点的 CPU、Memory、Network 和 Disk 的 Metric 数据。不过在 Kubernetes v1.11 版本后将被逐渐废弃。而 Metrics Server 正是 Heapster 的代替者。

Metrics Server 是 Kubernetes 集群核心监控数据的聚合器，Metrics Server 从 Kubelet 收集资源指标，并通过 Metrics API 在 Kubernetes API Server 中提供给缩放资源对象 HPA 使用。也可以通过 Metrics API 提供的 Kubectl top 查看 Pod 资源占用情况，从而实现对资源的自动缩放。

Metrics Server 不对旧值进行存储，且不负责将指标转发到第三方目标。

版权声明，本文档全部内容及版权归“张岩峰”老师所有，只可用于自己学习使用，**禁止私自传阅，违者依法追责。**



版权声明，本文档全部内容及版权归“张岩峰”老师所有，只可用于自己学习使用，**禁止私自传阅，违者依法追责。**

## 2、安装 Metrics Server

Metrics Server 与 kubernetes 版本对应表：

Compatibility Matrix		
Metrics Server	Metrics API group/version	Supported Kubernetes version
0.6.x	metrics.k8s.io/v1beta1	1.19+
0.5.x	metrics.k8s.io/v1beta1	*1.8+
0.4.x	metrics.k8s.io/v1beta1	*1.8+
0.3.x	metrics.k8s.io/v1beta1	1.8-1.21

基于 kubernetes 部署，官方提供了两个 yaml 文件，一个是以单机方式部署，一个是以高可用方式部署。

Feb 10, 2022

serathius

v0.6.1

9c9d712

Compare

### v0.6.1

#### Installation

```
kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/download/v0.6.1/components.yaml
```

#### Changes since v0.6.0


#### Deprecation

- Logging flags that are klog specific (`--log-dir`, `--log-file`, `--logtostderr`, `--alsologtostderr`, `--one-output`, `--stderrthreshold`, `--log-file-max-size`, `--skip-log-headers`, `--add-dir-header`, `--skip-headers`, `--log-backtrace-at`) are deprecated and will be removed following Kubernetes v1.25 release. More details are available in [KEP-2845](#).

#### Fixes

- Restore support for klog specific flags removed by mistake in v0.6.0 (@yangjunmyfm192085)

#### Contributors

 yangjunmyfm192085

#### Assets

<a href="#">components.yaml</a>	4.08 KB	Feb 10, 2022
<a href="#">high-availability-1.21+.yaml</a>	4.57 KB	Nov 26, 2022
<a href="#">high-availability.yaml</a>	4.57 KB	Feb 10, 2022
<a href="#">Source code (zip)</a>		Feb 9, 2022
<a href="#">Source code (tar.gz)</a>		Feb 9, 2022

4 people reacted

components.yaml 是以单机方式部署。  
high-availability-1.21+.yaml 和 high-availability.yaml 是以高可用方式部署。

版权声明，本文档全部内容及版权归“张岩峰”老师所有，只可用于自己学习使用，**禁止私自传阅，违者依法追责。**

版权声明，本文档全部内容及版权归“张岩峰”老师所有，只可用于自己学习使用，**禁止私自传阅，违者依法追责。**

## 2.1、单机版部署（v0.6.4）

部署操作如下：

### （1）下载官方 yaml 文件

```
[root@k8s-master01 ~]# wget https://github.com/kubernetes-sigs/metrics-server/releases/download/v0.6.4/components.yaml
```

### （2）修改 yaml 文件

```
# 查看当前所使用的镜像
[root@k8s-master01 ~]# cat components.yaml | grep image:
    image: k8s.gcr.io/metrics-server/metrics-server:v0.6.4
```

#### # 修改 yaml 文件

```
[root@k8s-master01 ~]# vi components.yaml
在 args 添加 “--kubelet-insecure-tls” 参数
    - --kubelet-insecure-tls
```

#### 修改 image 镜像

```
    image: bitnami/metrics-server:0.6.4
```

整体改动如下图：

```
spec:
  containers:
  - args:
    - --cert-dir=/tmp
    - --secure-port=4443
    - --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname
    - --kubelet-use-node-status-port
    - --metric-resolution=15s
    - --kubelet-insecure-tls
    image: bitnami/metrics-server:0.6.4
    imagePullPolicy: IfNotPresent
    livenessProbe:
```

为什么要添加--kubelet-insecure-tls 这条参数？

因为官方的 yaml 文件，做了就绪性探测，使用的是 https 方式做的探测，加上这条参数表示启用 https。如果不加，那么就绪性探测会一直失败。

### （3）安装

```
[root@k8s-master01 ~]# kubectl apply -f components.yaml
```

### （4）查看 pod 状态

```
[root@k8s-master01 ~]# kubectl get pods -n kube-system -l k8s-app=metrics-server
```

版权声明，本文档全部内容及版权归“张岩峰”老师所有，只可用于自己学习使用，**禁止私自传阅，违者依法追责。**

版权声明，本文档全部内容及版权归“张岩峰”老师所有，只可用于自己学习使用，**禁止私自传阅，违者依法追责。**

```
[root@k8s-master01 ~]# kubectl get pods -n kube-system -l k8s-app=metrics-server
```

NAME	READY	STATUS	RESTARTS	AGE
metrics-server-56ff899d9c-x7x4k	1/1	Running	0	109s

## (5) 测试

# 查看 pod cpu、内存使用率

```
[root@k8s-master01 ~]# kubectl top pods -n kube-system
```

```
[root@k8s-master01 ~]# kubectl top pods -n kube-system
```

NAME	CPU(cores)	MEMORY(bytes)
calico-kube-controllers-7b8458594b-gj5hj	4m	26Mi
calico-node-5t8qd	56m	147Mi
calico-node-7k9w2	55m	140Mi
calico-node-wt2h5	52m	142Mi
coredns-6d8c4cb4d-2qskw	2m	15Mi
coredns-6d8c4cb4d-cm74s	2m	18Mi
etcd-k8s-master01	21m	77Mi
kube-apiserver-k8s-master01	75m	351Mi
kube-controller-manager-k8s-master01	39m	55Mi
kube-proxy-6295r	1m	21Mi
kube-proxy-t7szc	12m	18Mi
kube-proxy-xlg7t	1m	20Mi
kube-scheduler-k8s-master01	5m	22Mi
metrics-server-56ff899d9c-x7x4k	5m	21Mi

# 查看 node 节点 cpu、内存使用率

```
[root@k8s-master01 ~]# kubectl top nodes
```

```
[root@k8s-master01 ~]# kubectl top nodes
```

NAME	CPU(cores)	CPU%	MEMORY(bytes)	MEMORY%
k8s-master01	258m	6%	1541Mi	41%
k8s-node01	114m	2%	767Mi	13%
k8s-node02	127m	3%	777Mi	13%

能够正常查看，就说明安装成功。

## (6) 清除环境

```
[root@k8s-master01 ~]# kubectl delete -f components.yaml
```

```
[root@k8s-master01 ~]# kubectl delete -f components.yaml
serviceaccount "metrics-server" deleted
clusterrole.rbac.authorization.k8s.io "system:aggregated-metrics-reader" deleted
clusterrole.rbac.authorization.k8s.io "system:metrics-server" deleted
rolebinding.rbac.authorization.k8s.io "metrics-server-auth-reader" deleted
clusterrolebinding.rbac.authorization.k8s.io "metrics-server:system:auth-delegator" deleted
clusterrolebinding.rbac.authorization.k8s.io "system:metrics-server" deleted
service "metrics-server" deleted
deployment.apps "metrics-server" deleted
apiservice.apiregistration.k8s.io "v1beta1.metrics.k8s.io" deleted
[root@k8s-master01 ~]#
```

## 2.2、高可用版部署（v0.6.4）

版权声明，本文档全部内容及版权归“张岩峰”老师所有，只可用于自己学习使用，**禁止私自传阅，违者依法追责。**



版权声明，本文档全部内容及版权归“张岩峰”老师所有，只可用于自己学习使用，**禁止私自传阅，违者依法追责。**

部署操作如下：

(1) 下载官方 yaml 文件

```
[root@k8s-master01 ~]# wget https://github.com/kubernetes-sigs/metrics-server/releases/download/v0.6.4/high-availability.yaml
```

(2) 修改 yaml 文件

```
# 查看当前所使用的镜像
[root@k8s-master01 ~]# cat high-availability.yaml | grep image:
    image: k8s.gcr.io/metrics-server/metrics-server:v0.6.4
```

# 修改 yaml 文件

```
[root@k8s-master01 ~]# vi high-availability.yaml
```

在 args 添加 “--kubelet-insecure-tls” 参数

```
    - --kubelet-insecure-tls
```

修改 image 镜像

```
    image: bitnami/metrics-server:0.6.4
```

整体改动如下图：

```
    - kube-system
      topologyKey: kubernetes.io/hostname
    containers:
      - args:
          - --cert-dir=/tmp
          - --secure-port=4443
          - --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname
          - --kubelet-use-node-status-port
          - --metric-resolution=15s
          - --kubelet-insecure-tls
        image: bitnami/metrics-server:0.6.4
        imagePullPolicy: IfNotPresent
        livenessProbe:
          failureThreshold: 3
          httpGet:
```

(3) 安装

```
[root@k8s-master01 ~]# kubectl apply -f high-availability.yaml
```

```
[root@k8s-master01 ~]# kubectl apply -f high-availability.yaml
serviceaccount/metrics-server unchanged
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader unchanged
clusterrole.rbac.authorization.k8s.io/system:metrics-server unchanged
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader unchanged
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator unchanged
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server unchanged
service/metrics-server unchanged
deployment.apps/metrics-server configured
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io unchanged
error: resource mapping not found for name: "metrics-server" namespace: "kube-system" from "high-availability.yaml": no matches for kind "PodDisruptionBudget" in version "policy/v1beta1"
ensure CRDs are installed first
```

部署报错，报错信息为：“PodDisruptionBudget 资源类型，api 版本 policy/v1beta1 没有被找到”

版权声明，本文档全部内容及版权归“张岩峰”老师所有，只可用于自己学习使用，**禁止私自传阅，违者依法追责。**

版权声明，本文档全部内容及版权归“张岩峰”老师所有，只可用于自己学习使用，**禁止私自传阅，违者依法追责。**

解决：

```
# 通过 kubectl explain 查看 PodDisruptionBudget 资源类型的 api 版本为
[root@k8s-master01 ~]# kubectl explain PodDisruptionBudget

[root@k8s-master01 ~]# kubectl explain PodDisruptionBudget
GROUP:          policy
KIND:            PodDisruptionBudget
VERSION:         v1

DESCRIPTION:
  PodDisruptionBudget is an object to define the max disruption allowed for pods.

# 修改资源清单文件
[root@k8s-master01 ~]# vi high-availability.yaml

---
apiVersion: policy/v1
kind: PodDisruptionBudget
metadata:
  name: metrics-server
  namespace: kube-system
spec:
  minAvailable: 1

# 再次部署
[root@k8s-master01 ~]# kubectl apply -f high-availability.yaml
```

(4) 查看 pod 状态

```
[root@k8s-master01 ~]# kubectl get pods -n kube-system -l k8s-app=metrics-server

[root@k8s-master01 ~]# kubectl get pods -n kube-system -l k8s-app=metrics-server
NAME                                READY   STATUS    RESTARTS   AGE
metrics-server-55fdd454d8-npg4p    1/1     Running   0           34s
metrics-server-55fdd454d8-qw7dw    1/1     Running   0           34s
```

(5) 测试

```
# 查看 pod cpu、内存使用率
[root@k8s-master01 ~]# kubectl top pods -n kube-system
```

版权声明，本文档全部内容及版权归“张岩峰”老师所有，只可用于自己学习使用，**禁止私自传阅，违者依法追责。**

版权声明，本文档全部内容及版权归“张岩峰”老师所有，只可用于自己学习使用，**禁止私自传阅，违者依法追责。**

```
[root@k8s-master01 ~]# kubectl top pods -n kube-system
```

NAME	CPU(cores)	MEMORY(bytes)
calico-kube-controllers-7b8458594b-gj5hj	4m	26Mi
calico-node-5t8qd	56m	147Mi
calico-node-7k9w2	55m	140Mi
calico-node-wt2h5	52m	142Mi
coredns-6d8c4cb4d-2qskw	2m	15Mi
coredns-6d8c4cb4d-cm74s	2m	18Mi
etcd-k8s-master01	21m	77Mi
kube-apiserver-k8s-master01	75m	351Mi
kube-controller-manager-k8s-master01	39m	55Mi
kube-proxy-6295r	1m	21Mi
kube-proxy-t7szc	12m	18Mi
kube-proxy-xlg7t	1m	20Mi
kube-scheduler-k8s-master01	5m	22Mi
metrics-server-56ff899d9c-x7x4k	5m	21Mi

# 查看 node 节点 cpu、内存使用率

```
[root@k8s-master01 ~]# kubectl top nodes
```

```
[root@k8s-master01 ~]# kubectl top nodes
```

NAME	CPU(cores)	CPU%	MEMORY(bytes)	MEMORY%
k8s-master01	258m	6%	1541Mi	41%
k8s-node01	114m	2%	767Mi	13%
k8s-node02	127m	3%	777Mi	13%

能够正常查看，就说明安装成功。

## (6) 清空环境

```
[root@k8s-master01 ~]# kubectl delete -f high-availability.yaml
```

版权声明，本文档全部内容及版权归“张岩峰”老师所有，只可用于自己学习使用，**禁止私自传阅，违者依法追责。**