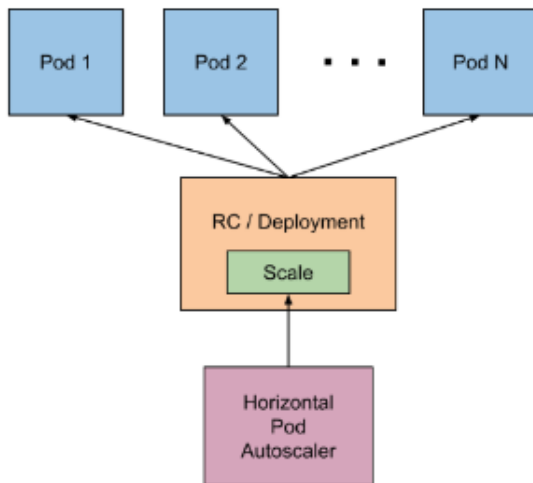


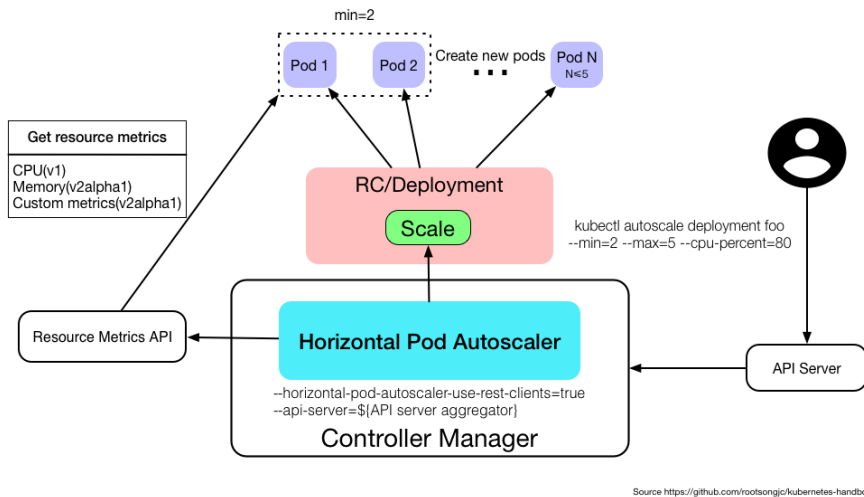
# K8S HPA介绍

- 1 HPA基本介绍
  - 1.1 hpa功能
  - 1.2 hpa意义
  - 1.3 hpa基本原理
  - 1.4 hpa使用场景
  - 1.5 hpa支持controller
  - 1.6 hpa获取metrics接口
  - 1.7 hpa算法
  - 1.8 hpa实现细节
- 2 HPA API支持
  - 2.1 autoscaling/v1
  - 2.2 autoscaling/v2beta2
  - 2.3 Pods和Object类型的差异
  - 2.4 Object和External类型的差异
- 3 HPA 配置
  - 3.1 获取默认配置
  - 3.2 功能调节配置
- 4 HPA使用
  - 4.1 命令行方式创建
  - 4.2 YAML方式创建
  - 4.3 状态查看
- 5 HPA-V2工作
- 6 参考

## 1 HPA基本介绍

HPA(Horizontal Pod Autoscaler)的工作原理图如下所示:





## 1.1 hpa功能

自动调整pod数量

## 1.2 hpa意义

使用户系统稳定运行，解决资源不足导致系统运行不稳定

## 1.3 hpa基本原理

基于当前metrics和目标metrics的设置，自动调整pod的数量

## 1.4 hpa使用场景

在pod增加(或减少)时，相应的metrics指标应该减少(或增加)

## 1.5 hpa支持controller

- ReplicationController
- ReplicaSet
- Deployment
- StatefulSets

## 1.6 hpa获取metrics接口

- [metrics.k8s.io](#): 该API由metrics-server提供
- [custom.metrics.k8s.io](#): 该API需要用户向APIServer注册
- [external.metrics.k8s.io](#): 该API需要用户向APIServer注册

## 1.7 hpa算法

```
desiredReplicas = ceil[currentReplicas * ( currentMetricValue / desiredMetricValue )]
```

## 1.8 hpa实现细节

- hpa必须设置最大pods，最小pods数可选设置
- 当设置了多个metrics指标，多个指标需要调整pod数时，选择最大的replica count
- hpa在调整pod数后，删除hpa controller，pod不会发生变化，因为hpa直接修改了replicas
- 如果pod中的任何一个container没有指定相关request资源(如CPU request)，则不会根据hpa设置的相关资源(如CPU request)进行pod autoscaler
- HPA不支持rc的rolling update，只支持deployment的rolling update
- HPA在计算AverageValue或AverageUtilization时，会先检查pod readiness和missing metrics:
  - 如果当前的Pod处于Terminating或者failed，这些Pods都不纳入计算
  - 如果当前的pod missing metrics，这些pods都不纳入计算

## 2 HPA API支持

### 2.1 autoscaling/v1

- 从[metrics.k8s.io](https://metrics.k8s.io)获取metrics
- 只支持CPU的平均百分比设置
- 根据request CPU判断，默认值为80%

### 2.2 autoscaling/v2beta2

- 支持Resource/Pods/Object/External四种类型
- Resource:
  - 从[metrics.k8s.io](https://metrics.k8s.io)获取metrics
  - 支持request CPU和MEM metrics
  - 支持平均百分比(如90%)和平均值(1Gi)
- Pods:
  - 从[custom.metrics.k8s.io](https://custom.metrics.k8s.io)获取metrics
  - 支持获取任意metrics(跟集群关联)
  - 只支持平均值设置
- Object:
  - 从[custom.metrics.k8s.io](https://custom.metrics.k8s.io)获取metrics
  - 支持获取任意metrics(跟集群关联)
  - 支持平均值设置和值设置(不用除以pod数)
- External:
  - 从[external.metrics.k8s.io](https://external.metrics.k8s.io)获取metrics
  - 支持获取任意metrics(跟集群无关联)
  - 支持平均值设置和值设置(不用除以pod数)

### 2.3 Pods和Object类型的差异

- Pods类型表示pod的metrics
- Object类型可以表示同一namespace下除pod外的所有对象(如Ingress, Service)

### 2.4 Object和External类型的差异

- Object类型表示K8S相关的对象
- External类型表示的对象与K8S对象无关，一般用在公有云上
- K8S推荐使用Object，不推荐使用External

## 3 HPA 配置

### 3.1 获取默认配置

获取apiserver:

- `kubectl get apiservice v1beta1.metrics.k8s.io -o yaml`
- `kubectl get apiservice v1beta1.custom.metrics.k8s.io -o yaml`

获取autoscaling:

- `kubectl get hpa.v1.autoscaling/php-apache -o yaml`
- `kubectl get hpa.v2beta2.autoscaling/php-apache -o yaml`

### 3.2 功能调节配置

通过kube-controller-manager进行配置:

- `--horizontal-pod-autoscaler-sync-period`: 获取metrics的时间间隔，默认值15秒
- `--horizontal-pod-autoscaler-tolerance`: 当前metrics和目标metrics之间的比列在tolerance之内，pod都不autoscale，默认值为0.1
- `--horizontal-pod-autoscaler-upscale-stabilization`: 每次调高pod数的最小间隔时间，默认为3分钟(从1.12版本之后都已移除了)
- `--horizontal-pod-autoscaler-downscale-stabilization`: 每次调低pod数的最小间隔时间，默认为5分钟
- `--horizontal-pod-autoscaler-initial-readiness-delay`: 默认值为30秒
- `--horizontal-pod-autoscaler-cpu-initialization-period`: 默认值5分钟

注: 最后两个选项官方文档解释不清晰，有人专门为此提了issue(<https://github.com/kubernetes/website/issues/12657>)

## 4 HPA使用

### 4.1 命令行方式创建

```
# kubectl autoscale deployment php-apache --cpu-percent=50 --min=1 --max=10
```

### 4.2 YAML方式创建

```
# cat hpa-php-apache.yaml
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: php-apache
  namespace: default
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: php-apache
  minReplicas: 1
  maxReplicas: 10
  targetCPUUtilizationPercentage: 50
# kubectl apply -f hpa-php-apache.yaml
```

### 4.3 状态查看

```
# kubectl get hpa/php-apache
NAME           REFERENCE              TARGETS  MINPODS  MAXPODS  REPLICAS  AGE
php-apache     Deployment/php-apache  1%/80%   1         10        1          6d5h
# kubectl describe hpa/php-apache
Name:
Namespace:
Labels:
Annotations:
CreationTimestamp:
Reference:
Metrics:
  resource cpu on pods (as a percentage of request): 1% (2m) / 80%
Min replicas:
Max replicas:
Deployment pods:
Conditions:
  Type           Status  Reason
  ----           -
  AbleToScale    True    ReadyForNewScale
  ScalingActive  True    ValidMetricFound
  ScalingLimited False   DesiredWithinRange
Events:
#
```

Type	Status	Reason	Message
AbleToScale	True	ReadyForNewScale	recommended size matches current size
ScalingActive	True	ValidMetricFound	the HPA was able to successfully calculate a replica count from cpu resource utilization (percentage of request)
ScalingLimited	False	DesiredWithinRange	the desired count is within the acceptable range

```
Events:
#
```

## 5 HPA-V2工作

- 支持基于IO、NET metrics来autoscale pod
- 支持基于limit resource来autoscale pod
- 支持基于namespace来autoscale pod
- 支持autoscale pod为0, 当前最小为1

## 6 参考

- <https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/>
- <https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale-walkthrough/>

- <https://github.com/kubernetes/community/blob/master/contributors/design-proposals/autoscaling/hpa-v2.md>
- <https://github.com/kubernetes/community/blob/master/contributors/design-proposals/autoscaling/horizontal-pod-autoscaler.md>
- <https://juejin.im/post/5d397288f265da1bce3e1585>
- <http://jira.iluvatar.ai:8080/browse/SDXSYS-321>