

Kubernetes服务质量

主讲人：宋小金





目录

1

QoS分类

2

资源需求与限制

3

资源回收策略

4

使用建议

预期收获

- 学习Kubernetes QoS配置
- 在资源利用率和业务稳定性之间如何做好权衡



背景说明

Kubernetes作为目前主流的容器集群管理平台，需要整体统筹平台资源使用情况、公平合理的将资源分配给相关pod容器使用，并且要保证容器生命周期内有足够的资源来保证其运行；与此同时，由于资源发放的独占性，即资源已经分配给了某容器，同样的资源不会在分配给其他容器，对于资源利用率相对较低的容器来说，占用资源却没有实际使用（比如CPU、内存）造成了严重的资源浪费，Kubernetes需从优先级与公平性等角度提高资源的利用率。



QoS分类

- **Guaranteed:** 独占分配的资源，不与其他Pod共享
- **Burstable:** 共享程度次之
- **Best-Effort** : 与其他Pod共享资源，共享资源程度最高

Best-Effort pods -> Burstable pods -> Guaranteed pods

优先级由低向高



Guaranteed

- 容器只指明了limits而未指明requests
- requests与limit均指定且值相等

举例1

```
containers:
  name: foo
  resources:
    limits:
      cpu: 10m
      memory: 1Gi
  name: bar
  resources:
    limits:
      cpu: 100m
      memory: 100Mi
```

举例2

```
containers:
  name: foo
  resources:
    limits:
      cpu: 10m
      memory: 1Gi
    requests:
      cpu: 10m
      memory: 1Gi
  name: bar
  resources:
    limits:
      cpu: 100m
      memory: 100Mi
    requests:
      cpu: 100m
      memory: 100Mi
```




资源回收策略

当kubernetes集群中某个节点上可用资源比较小时，kubernetes提供了[资源回收策略](#)保证被调度到该节点pod服务正常运行。当节点上的内存或者CPU资源耗尽时，可能会造成该节点上正在运行的pod服务不稳定。Kubernetes通过[kubelet来进行回收策略控制](#)，保证节点上pod在节点资源比较小时可以稳定运行。



QoS pods被kill掉场景与顺序

- Best-Effort 类型的pods：系统用完了全部内存时，该类型pods会最先被kill掉。
- Burstable类型pods：系统用完了全部内存，且没有Best-Effort container可以被kill时，该类型pods会被kill掉。
- Guaranteed pods：系统用完了全部内存、且没有Burstable与Best-Effort container可以被kill，该类型的pods会被kill掉。

注：如果pod进程因使用超过预先设定的**limites**而非**Node**资源紧张情况，系统倾向于在其原所在的机器上重启该container，或本机/其他重新创建一个pod。



QoS使用建议

- 如果资源充足，可将QoS pods类型均设置为Guaranteed。用计算资源换业务性能和稳定性，减少排查问题时间和成本。
- 如果想更好的提高资源利用率，业务服务可以设置为Guaranteed，而其他服务根据重要程度可分别设置为Burstable或Best-Effort，例如filebeat。



QoS分类

- Guaranteed: 优先级最高，独占分配的资源
- Burstable: 优先级次之
- Best-Effort : 优先级最低，共享资源程度最高



可压缩资源与不可压缩资源

Kubernetes根据资源能否伸缩进行分类，划分为可压缩资源和不可以压缩资源2种。CPU资源是目前支持一种可压缩资源，而内存资源和磁盘资源为目前所支持的不可压缩资源。

- 可压缩资源：**CPU**

在压缩资源部分已经提到CPU属于可压缩资源，当pod使用超过设置的limits值，pod中进程使用cpu会被限制，但不会被kill。

- 不可压缩资源：内存

Kubernetes通过cgroup给pod设置QoS级别，当资源不足时先kill优先级低的pod，在实际使用过程中，通过OOM分数值来实现，OOM分数值从0-1000。



课程回顾

已学知识要点

学习QoS配置，在资源利用率和业务稳定性之间做好权衡