

自记小屋

## k8s基础维护命令

2019年4月2日

64浏览

Kubernetes

发表评论

### 文章目录

- 0.1 基础维护
  - 0.1.1 #强制删除已经Terminating pod
  - 0.1.2 删除Evicted（被驱逐的pod）
- 1 更新操作
  - 1.0.1 也可以进行编辑进行更改
- 2 回滚操作 < deployment|daemonset >
  - 2.0.1 查看版本发布历史
  - 2.0.2 回滚到上一个版本
  - 2.0.3 回滚到指定版本
  - 2.0.4 查看发布的状态
- 3 扩容操作
  - 3.0.1 设置自动扩容（水平拉伸）, 当pod cpu使用率超过10%, 最小创建pod数为1, 最大3
  - 3.0.2 指定扩容pod副本数量
- 3.1 滚动更新
- 3.2 暂停滚动更新
- 3.3 继续滚动更新
- 3.4 保留版本次数
- 3.5 常用命令

## 基础维护

# 当需要对主机进行维护升级时，首先将节点主机设置成不可调度模式：

```
kubect1 cordon [nodeid]
```

```
[root@ziji-k8s-master1-192-168-1-16 coredns]# kubectl cordon 192.168.1.17
```

```
node/192.168.1.17 cordoned
```

```
[root@ziji-k8s-master1-192-168-1-16 coredns]# kubectl get node
```

NAME	STATUS	ROLES	AGE	VERSION
192.168.1.16	Ready,SchedulingDisabled	master	25d	v1.11.4
192.168.1.17	Ready,SchedulingDisabled	node	27d	v1.11.4
192.168.1.18	Ready	node	27d	v1.11.4
192.168.1.19	Ready	node	27d	v1.11.4
192.168.1.20	Ready	dc1-db,node	15d	v1.11.4

# 然后将主机上正在运行的容器驱赶到其它可用节点：

```
kubectl drain [nodeid]
```

# 给予900秒宽限期优雅的调度

```
kubectl drain node1.k8s.novalocal --grace-period=120
```

# 当容器迁移完毕后，运维人员可以对该主机进行操作，配置升级性能参数调优等等。当对主机的维护操作完毕后，再将主机设置成可调度模式：

```
kubectl uncordon [nodeid]
```

## #强制删除已经Terminating pod

```
kubectl get pod -n zk-kafka
```

```
kubectl delete zookeeper-2-4n426 -n zk-kafka --grace-period=0 --force
```

## 删除Evicted（被驱逐的pod）

```
kubectl get pods -n apache-test | grep Evicted | awk '{print $1}' | xargs kubectl delete pod -n apache-test
```

## 更新操作

注意：当且仅当部署的pod模板（即`.spec.template`）更改时，才会触发Deployment的部署，例如，如果更新模板的标签或容器图像。其他更新（例如扩展部署）不会触发部署。

```
kubectl set image deployment/web-service web-service=web-service:v1.9.1 --record  
deployment "web-service" image updated
```

## 也可以进行编辑进行更改

```
$ kubectl edit deployment/web-service -n ziji  
deployment "nginx-deployment" edited
```

## 回滚操作 < deployment|daemonset >

### 查看版本发布历史

```
kubectl rollout history deployment/web-service -n ziji  
deployments "web-service"
```

REVISION	CHANGE-CAUSE
23	<none>
27	<none>
28	<none>

### 回滚到上一个版本

```
kubectl rollout undo deployment -n ziji web-service
```

### #查看deployment对象信息

```
kubectl describe deployment -n ziji web-service

Name:                web-service
Namespace:           ziji
CreationTimestamp:    Sun, 02 Dec 2018 18:31:43 +0800
Labels:              app=apache-v1
                    name=web-service
Annotations:         deployment.kubernetes.io/revision=28
```

### #查看replicasets对象信息

```
kubectl get rs -o wide -n ziji
```

## 回滚到指定版本

```
kubectl rollout undo deployment/web-service --to-revision=27
```

## 查看发布的状态

```
kubectl rollout status deployment/web-service -n ziji
deployment "web-service" successfully rolled out
```

## 扩容操作

设置自动扩容（水平拉伸），当pod cpu使用率超过10%, 最小创建pod数为1, 最大3

```
kubectl autoscale deployment -n ziji web-service --min=1 --max=3 --cpu-percent=10
```

## 指定扩容pod副本数量

```
kubectl scale deployment -n ziji web-service --replicas 3
```

## 滚动更新

```
spec:
  replicas: 10
  selector:
    matchLabels:
      service: http-server

  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
  minReadySeconds: 5
```

## 暂停滚动更新

```
kubectl rollout pause deployment <deployment>
```

## 继续滚动更新

```
kubectl rollout resume deployment <deployment>
```

## 保留版本次数

```
...

spec:
  replicas: 10
  selector:
    matchLabels:
      service: http-server

  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
  minReadySeconds: 5
  revisionHistoryLimit: 10

...
```

## 常用命令

- 创建: `kubectl run <name> --image=<image>` 或者 `kubectl create -f manifest.yaml`
- 查询: `kubectl get <resource>`
- 更新 `kubectl set` 或者 `kubectl patch`
- 删除: `kubectl delete <resource> <name>` 或者 `kubectl delete -f manifest.yaml`
- 查询 Pod IP: `kubectl get pod <pod-name> -o jsonpath='{.status.podIP}'`
- 容器内执行命令: `kubectl exec -ti <pod-name> sh`
- 容器日志: `kubectl logs [-f] <pod-name>`
- 导出服务: `kubectl expose deploy <name> --port=80`
- Base64 解码:

```
kubectl get secret SECRET -o go-template='{{ .data.KEY | base64decode }}'
```

## 查看endpoints

```
[root@feiba-k8s-master1-192-168-1-16 ~]# kubectl get ep -n ziji-test
```

NAME	ENDPOINTS	AGE
glusterfs-dynamic-pvc-zup-config	192.168.1.12:1,192.168.1.13:1,192.168.1.14:1 + 1 more...	34d
glusterfs-dynamic-pvc-zup-gateway	192.168.1.12:1,192.168.1.13:1,192.168.1.14:1 + 1 more...	34d
glusterfs-dynamic-pvc-zup-payment-web	192.168.1.12:1,192.168.1.13:1,192.168.1.14:1 + 1 more...	32d
glusterfs-dynamic-pvc-zup-task	192.168.1.12:1,192.168.1.13:1,192.168.1.14:1 + 1 more...	33d
games	10.96.20.3:8099	4d
gateway	10.96.20.10:6674	18d
gateway-admin	10.96.66.7:5674	4d
merchant	10.96.20.8:8080	33d
mining	10.96.66.5:8085	33d
payment-web	10.96.20.13:8084	32d
register	10.96.20.7:8888	18d
task	10.96.20.6:8083	33d

```
[root@ziji-k8s-master1-192-168-1-16 ~]# kubectl get endpoints mining -n test
```

NAME	ENDPOINTS	AGE
mining	10.96.66.5:8085	33d

## 日志查看

```
kubectl logs nginx
kubectl logs -f -c ruby web-1
```

## 连接到一个正在运行的容器

```
kubectl attach 用于连接到一个正在运行的容器
```

## 文件拷贝

```
kubectl cp /tmp/foo_dir <some-pod>:/tmp/bar_dir      #拷贝文件到容器中
kubectl cp <some-namespace>/<some-pod>:/tmp/foo /tmp/bar  #从容器拷贝文件到本地
```

标签: [k8s基础维护命令](#), [ks8代维](#), [kubernetes代维](#)

## 发表评论

电子邮件地址不会被公开。 必填项已用\*标注

评论 \*

昵称 \*

邮箱 \*

网站

☒ Notify me when new comments are added.

发表评论

搜索...

搜索

## 近期文章

[使用kubeadm部署Kubernetes1.14.3集群之dashboard ui界面 – 第二章](#)

[kubeadm快速安装kubernetes v1.14.3 – 第一章](#)