

18.删除Pod的几种方式

[按名称删除Pod](#)

[使用标签选择器删除Pod](#)

[通过命名空间来删除Pod](#)

[删除命名空间下的所有资源](#)

通过前面一系列学习，我们已经在集群中创建了几个Pod，并都处于运行状态中。在default命名空间下有四个Pod，在my-namespace命名空间下有一个Pod：

```
[david@dhr-demo root]$ kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
test1         1/1     Running   0           13d
test1-gpu     1/1     Running   0           3d10h
test1-manual  1/1     Running   0           6d22h
test1-manual-v2 1/1     Running   0           4d21h
[david@dhr-demo root]$ kubectl get po -n my-namespace
NAME          READY   STATUS    RESTARTS   AGE
test1-manual  1/1     Running   0           13h
[david@dhr-demo root]$
```

现在来停止并删除这些Pod。

按名称删除Pod

首先来删除Pod test1-gpu：

`kubectl delete po test1-gpu`

```
[david@dhr-demo root]$ kubectl delete po test1-gpu
pod "test1-gpu" deleted
```

我们也可以通过在命名行中指定多个空格分隔的Pod名称来删除多个Pod，比如`kubectl delete po pod1 pod2 pod3`。

删除一个Pod的时候，Kubernetes会终止该Pod中的所有容器的运行。Kubernetes会发送一个SIGTERM信号到所有进程，然后等待一定的秒数（默认30s）让进程正常关闭。如果进程未正常关闭，则发送SIGKILL信号终止该进程。为了确保进程正常关闭，进程需要正确地处理SIGTERM信号。

使用标签选择器删除Pod

我们知道标签选择器可以筛选出一组符合条件的资源，因此可以基于它来删除多个资源。

如下图，test1-manual和test1-manual-v2这两个Pod具有共同的标签`creation_method=manual`：

```
[david@dhr-demo root]$ kubectl get pod --show-labels
NAME      READY   STATUS    RESTARTS   AGE   LABELS
test1     1/1     Running   0          13d   run=test1
test1-manual 1/1     Running   0          6d22h creation_method=manual
test1-manual-v2 1/1     Running   0          4d22h creation_method=manual,env=test
[david@dhr-demo root]$
```

因此我们可以使用标签选择器来删除它们：

```
kubectl delete po -l creation_method=manual
```

```
[david@dhr-demo root]$ kubectl delete po -l creation_method=manual
pod "test1-manual" deleted
pod "test1-manual-v2" deleted
[david@dhr-demo root]$
```

通过命名空间来删除Pod

在上一节我们创建了一个新的命名空间my-namespace。现在不需要这个命名空间以及其中的Pod了。

可以通过如下命令来删除整个命名空间，该空间下的所有Pod都会自动被删除：

```
kubectl delete ns my-namespace
```

```
[david@dhr-demo root]$ kubectl get po -n my-namespace
NAME      READY   STATUS    RESTARTS   AGE
test1-manual 1/1     Running   0          13h
[david@dhr-demo root]$ kubectl delete ns my-namespace
namespace "my-namespace" deleted
[david@dhr-demo root]$
```

如果只是想删除命名空间下的所有Pod并保留命名空间的话，可以执行如下命令：

```
kubectl delete po --all
```

上面的命令会删除当前命名空间下的所有Pod：

```
[david@dhr-demo root]$ kubectl get po
NAME      READY   STATUS    RESTARTS   AGE
test1     1/1     Running   0          13d
[david@dhr-demo root]$ kubectl delete po --all
pod "test1" deleted
[david@dhr-demo root]$
```

删除命名空间下的所有资源

需要注意的是，如果一个Pod不是由你直接创建的，而是由另外的资源对象（比如ReplicationController）负责创建的，那么即使删除了这个Pod，该资源对象可能还会自动创建一个新的Pod。要想完全删除这个Pod，就需要把这个资源对象也删除了。

通过如下命令，可以将当前命名空间下的所有资源（Pod、ReplicationContainer、Service等等）全部删除：

```
kubectl delete all --all
```

```
[david@dhr-demo root]$ kubectl delete all --all
service "kubernetes" deleted
service "test1-http" deleted
[david@dhr-demo root]$
```

可以看到上面的命名将kubernetes服务也一并删除了，不必担心，因为过会儿它就会被自动重建：

```
[david@dhr-demo root]$ kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP    11m
[david@dhr-demo root]$
```

需要注意的是all并不绝对就表示所有资源。某些资源（比如Secrets）仍然会保留，我们需要显示地删除它们。