原文地址：https://blog.csdn.net/lxwh0323/article/details/121595105

**1 基本语法格式**

**1.1 语法格式**

kubectl action resource

语法帮助格式（查看可选参数）

kubectl action resource --help

**1.2 action**

常用的action有

get：列表查询

edit：编辑

create：创建

delete：删除

describe：单个详情查询

label：为资源打标签

**1.3 resource**

常用的资源有

services：服务

pods：pod

deployments：pod控制器

ns：命名空间

**1.4 常用的选项**

常用的选项有

-n xxx：指定命名空间

-l app=xxx：指定标签app=xxx的

-o wide：更详细的列表信息

**1.4 基本示例**

获取命名空间列表

kubectl get ns

创建命名空间

kubectl create ns/myns

查询命名空间

kubectl get ns/myns

查看命名空间详情

kubectl describe ns/myns

删除命名空间

kubectl delete ns/dev

**1.5 其他名**

查看集群信息

kubectl cluster-info

集群服务映射本机，如下将本机的ip:8080映射到服务kong-proxy:80端口

kubectl port-forward -n kong --address 0.0.0.0 service/kong-proxy 8080:80

**2 节点**

Node 是一个虚拟机或者物理机，它在 Kubernetes 集群中充当工作机器的角色

**2.1 查看集群节点**

root@k8s-master:~# kubectl get nodes

NAME STATUS ROLES AGE VERSION

k8s-master Ready control-plane,master 3h47m v1.23.5

k8s-node1 Ready <none> 90m v1.23.5

mirc4 Ready <none> 111s v1.23.5

**3 Namespace**

命名空间

Kubernetes的资源可以分配命名空间，不同命名空间下的资源相互隔离

**3.1 查看命名空间列表**

[root@master ~]# kubectl get ns

NAME STATUS AGE

default Active 45h # 所有未指定Namespace的对象都会被分配在default命名空间

kube-system Active 45h # 所有由Kubernetes系统创建的资源都处于这个命名空间

**3.2 创建命名空间**

# 创建namespace

[root@master ~]# kubectl create ns dev

namespace/dev created

**3.3 删除命名空间**

# 删除namespace

[root@master ~]# kubectl delete ns/dev

namespace "dev" deleted

**4 Deployment**

在kubernetes中，Pod是最小的控制单元，但是kubernetes很少直接控制Pod，一般都是通过Pod控制器来完成的。Deployment用于控制Pod的部署

**4.1 创建一个Deployment**

# kubectl create deployment nginx --image=nginx:latest --port=80 -n dev

deployment.apps/nginx created

--image：指定pod的镜像

--por：指定端口

--namespace ：指定namespace，如果不加这分配到default命名空间

**4.2 扩展Deploymeny的pod数量**

默认Deployment的pod数量为1，可以执行如下代码，让Deployment扩容pod的数量

# kubectl scale deployments/nginx --replicas=3 -n dev

deployment.apps/nginx scaled

--replicas：扩容至指定数量

扩容之后我们查看pod的列表

root@k8s-master:~# kubectl get pods -n dev

NAME READY STATUS RESTARTS AGE

nginx-8d545c96d-h95jb 1/1 Running 0 63s

nginx-8d545c96d-qz8gb 1/1 Running 0 3m57s

nginx-8d545c96d-vrprm 1/1 Running 0 63s

**4.3 列出创建的Deployment**

# UP-TO-DATE：成功升级的副本数量

# AVAILABLE：可用副本的数量

[root@master ~]# kubectl get deployments -n dev -o wide

NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

nginx 3/3 3 3 2m51s nginx nginx:latest run=nginx

-n：指定命名空间，如果不加，查询default命名空间

**4.4 查看Deployment详细信息**

# 查看deployment的详细信息

[root@master ~]# kubectl describe deployments/nginx -n dev

**4.5 删除Deployment**

# 删除

[root@master ~]# kubectl delete deployments/nginx -n dev

deployment.apps "nginx" deleted

**5 Pod**

Pod是kubernetes集群进行管理的最小单元，程序要运行必须部署在容器中，而容器必须存在于Pod中。Pod可以认为是容器的封装，一个Pod中可以存在一个或者多个容器。

**5.1 列出当前pod列表**

# kubectl get pods -n dev

**5.2 列出系统的pod**

[root@master ~]# kubectl get pods -n kube-system

NAMESPACE NAME READY STATUS RESTARTS AGE

kube-system coredns-6955765f44-68g6v 1/1 Running 0 2d1h

kube-system coredns-6955765f44-cs5r8 1/1 Running 0 2d1h

kube-system etcd-master 1/1 Running 0 2d1h

kube-system kube-apiserver-master 1/1 Running 0 2d1h

kube-system kube-controller-manager-master 1/1 Running 0 2d1h

kube-system kube-flannel-ds-amd64-47r25 1/1 Running 0 2d1h

kube-system kube-flannel-ds-amd64-ls5lh 1/1 Running 0 2d1h

kube-system kube-proxy-685tk 1/1 Running 0 2d1h

kube-system kube-proxy-87spt 1/1 Running 0 2d1h

kube-system kube-scheduler-master 1/1 Running 0 2d1h

**5.3 查询Deployment对应的pod**

kubectl get pods -l app=nginx -n dev

**5.4 查询pod日志**

kubectl logs $POD\_NAME

**5.5 删除pod**

如果我们的pod是由Deployment管理的，那么我们删除后Deployment会马上创建一个pod，这时要想删除pod，删除对应的Deployment即可

[root@master ~]# kubectl delete pods/nginx -n dev

pod "nginx" deleted

**5.6 进入pod的容器**

kubectl exec $pod\_name -n dev -c $container\_name -it /bin/sh

$pod\_name：pod名称

-c $container\_name：容器名称，如果pod只有一个容器，可以忽略

**6 Label**

Label的作用就是在资源上添加标识

Label以key/value键值对的形式附加到各种对象上，如Node、Pod、Service等等

一个资源对象可以定义任意数量的Label，一个Label也可以被添加到任意数量的资源对象上去

**6.1 为pod资源打标签**

[root@master ~]# kubectl label pods/nginx-pod version=1.0 -n dev

pod/nginx-pod labeled

**6.2 为pod资源更新标签**

[root@master ~]# kubectl label pods/nginx-pod version=2.0 -n dev --overwrite

pod/nginx-pod labeled

**6.3 查看资源的标签**

[root@master ~]# kubectl get pods/nginx-pod -n dev --show-labels

NAME READY STATUS RESTARTS AGE LABELS

nginx-pod 1/1 Running 0 10m version=2.0

**6.4 筛选标签**

[root@master ~]# kubectl get pods -n dev -l version=2.0 --show-labels

NAME READY STATUS RESTARTS AGE LABELS

nginx-pod 1/1 Running 0 17m version=2.0

[root@master ~]# kubectl get pod -n dev -l version!=2.0 --show-labels

No resources found in dev namespace.

**6.5 删除标签**

[root@master ~]# kubectl label pods/nginx-pod version- -n dev

pod/nginx-pod labeled

**7 Sevice**

Service可以看作是一组同类Pod对外的访问接口

**7.1 列出当前的服务**

kubectl get services -n dev

**7.2 创建服务（暴露服务）**

暴露之后可以通过外部url访问服务

kubectl expose deployments/nginx --name=nginx-service --type="NodePort" --port=80 --target-port=80 -n dev

deployments/nginx：要暴露的deployment

--name：服务的名称

--type：为NodePort表示暴露为外部服务，集群外可以访问，为ClusterIP则只能在集群内访问

--port：使用的端口

--target-port：deployment对应的端口

**7.3 查询Deployment对应的服务**

kubectl get services -l app=nginx -n dev

**7.4 删除服务**

[root@master ~]# kubectl delete services/nginx-service -n dev