

# kubectl命令使用

笔记本：新课程笔记

创建时间：2018/9/8 星期六 上午 8:56

更新时间：2018/10/11 星期四 下午 10:28

作者：306798658@qq.com

语法：

kubectl [command] [TYPE] [NAME] [flags]

1 command：子命令，用于操作Kubernetes集群资源对象的命令，如create, delete, describe, get, apply等

2 TYPE：资源对象的类型，如pod, service, rc, deployment, node等，可以单数、复数以及简写（pod, pods, po/service, services, svc）

3 NAME：资源对象的名称，不指定则返回所有，如get pod 会返回所有pod，get pod nginx，只返回nginx这个pod

4 flags：kubectl子命令的可选参数，例如-n 指定namespace，-s 指定apiserver的URL

资源对象类型列表  
可以用这个命令获取到：

kubectl explain

或

kubectl api-resources

名称	简写
componentsstatuses	cs
daemonsets	ds
deployment	deploy
events	ev
endpoints	ep
horizontalpodautoscalers	hpa
ingresses	ing
jobs	
limitranges	limits
nodes	no
namespaces	ns
pods	po
persistentvolumes	pv
persistentvolumeclaims	pvc
resourcequotas	quota

replicationcontrollers	rc
secrets	
serviceaccounts	sa
services	svc

特殊用法：

```
kubect1 get pods pod1 pod2
kubect1 get pod/pod1 rc/rc1
kubect1 create -f pod1.yaml -f rc1.yaml -f service1.yaml
```

kubect1子命令

主要包括对资源的创建、删除、查看、修改、配置、运行等

kubect1 --help 可以查看所有子命令

kubect1参数

kubect1 options 可以查看支持的参数，例如--namespace指定所在namespace

kubect1输出格式

kubect1命令可以用多种格式对结果进行显示，输出格式通过-o参数指定：

-o支持的格式有

输出格式	说明
custom-columns= <spec>	根据自定义列名进行输出，逗号分隔
custom-columns-file= <filename>	从文件中获取自定义列名进行输出
json	以JSON格式显示结果
jsonpath= <template>	输出jsonpath表达式定义的字段信息
jsonpath-file= <filename>	输出jsonpath表达式定义的字段信息，来源于文件
name	仅输出资源对象的名称
wide	输出更多信息，比如会输出node名
yaml	以yaml格式输出

举例：

```
kubect1 get pod -o wide
kubect1 get pod -o yaml
kubect1 get pod -o custom-columns=NAME:.metadata.name,RESC:.metadata.resourceVersion
kubect1 get pod --sort-by=.metadata.name //按name排序
```

kubect1命令示例：

1) 创建资源对象

根据yaml文件创建service和deployment

```
kubect1 create -f my-service.yaml -f my-deploy.yaml
```

也可以指定一个目录，这样可以一次性根据该目录下所有yaml或json文件定义资源

```
kubectl create -f <directory>
```

## 2) 查看资源对象

查看所有pod

```
kubectl get pods
```

查看deployment和service

```
kubectl get deploy,svc
```

## 3) 描述资源对象

显示node的详细信息

```
kubectl describe nodes <node-name>
```

显示pod的详细信息

```
kubectl describe pods/<pod-name>
```

显示deployment管理的pod信息

```
kubectl describe pods <deployment-name>
```

## 4) 删除资源对象

基于yaml文件删除

```
kubectl delete -f pod.yaml
```

删除所有包含某个label的pod和service

```
kubectl delete po,svc -l name=<lable-name>
```

删除所有pod

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

## 5) 执行容器的命令

在pod中执行某个命令，如date

```
kubectl exec <pod-name> date //pod-name如果不加，默认会选择第一个pod
```

指定pod的某个容器执行命令

```
kubectl exec <pod-name> date
```

进入到pod的容器里

```
kubectl exec -it <pod-name> bash
```

## 6) 查看容器日志

```
kubectl logs <pod-name>
```

可以动态查看，类似于tail -f

```
kubectl logs -f <pod-name> -c <container-name>
```