kubernetes常用命令

添加命令自动补全：
yum install -y bash-completion
vim ~/.bashrc
添加
source <(kubectl completion bash)

常用命令：

```
#####       ####
kubectl get deployments      ##
kubectl get pod         ## namespace pod
kubectl get pod --all-namespaces    ## namespace pod
kubectl get pod  -o wide      ## namespace pod
kubectl get pod weave-scope-agent-5kzl7 -n weave  -o yaml  ## yaml
-o
-n
--all-namespace:


# pspod
kubectl get pods

# pspod
kubectl get pods -o wide

# psNAME
kubectl get replicationcontroller web

# JSONpod
kubectl get -o json pod web-pod-13je7

# JSON“pod.yaml”pod
kubectl get -f pod.yaml -o json

# pod
kubectl get -o template pod/web-pod-13je7 --template={{.status.phase}}

# ps
kubectl get rc,services

#
kubectl get rc/web service/frontend pods/web-pod-13je7

#
kubectl get all
```

```
##### #        ########
#
kubectl describe nodes kubernetes-node-emt8.c.myproject.internal

# pod
kubectl describe pods/nginx

# “pod.json”pod
kubectl describe -f pod.json

#
kubectl describe pods

# = myLabel
kubectl describe po -l name=myLabel

# ''podrcpod
# rcpod
kubectl describe pods frontend









####        #####
# pod.json/pod.yamlpod
kubectl apply -f pod.json
kubectl apply -f pod.yaml


# nginx
kubectl run nginx --image=nginx

# hazelcast5701
kubectl run hazelcast --image=hazelcast --port=5701

# hazelcast“DNS_DOMAIN = cluster”“POD_NAMESPACE = default”
kubectl run hazelcast --image=hazelcast --env="DNS_DOMAIN = cluster" --env
="POD_NAMESPACE=default"

# hazelcast“app = hazelcast”“env = prod”
kubectl run hazelcast --image=nginx --labels="app=hazelcast,env=prod"

# nginx
kubectl run nginx --image = nginx --replicas=5

# API
kubectlnginx --image=nginx --dry-run
```

```
# nginxJSON
kubectl run nginx --image=nginx
--overrides='{“apiVersion”"v1","spec":{...}}'

# busyboxpod
kubectl run -i -t busybox --image=busybox --restart=Never

# defaultnginxarg1 .. argN
kubectl run nginx --image=nginx -- <arg1> <arg2> ... <argN>

# nginx
kubectl run nginx --image=nginx --command -- <cmd> <arg1> ... <argN>

# perl2000
kubectl run pi --image=perl --restart=OnFailure -- perl -Mbignum=bpi -wle
'print bpi(2000)'

# cron20005
kubectl run pi --schedule="0/5 * * * ?" --image=perl --restart=OnFailure --
perl -Mbignum=bpi -wle 'print bpi(2000)'




####        #####
# pod.json/pod.yamlpod
kubectl delete -f ./pod.json
kubectl delete -f ./pod.yaml

# stdinJSONpod
cat pod.json | kubectl delete -f -

# “baz”“foo”pod
kubectl delete pod,service baz foo

# = myLabelpod
kubectl delete pods,services -l name=myLabel

# pod
kubectl delete pod foo --now

# pod
kubectl delete pod foo --grace-period=0 --force

# pod
kubectl delete pods --all




#
kubectl run kubernetes-bootcamp
```

```
--image=10.22.60.169/test/kubernetes-bootcamp:v1 --port=8080

#
kubectl expose deployment/kubernetes-bootcamp --type="NodePort" --port 8080

#
kubectl scale deployments/kubernetes-bootcamp --replicas=3

#
kubectl scale deployments/kubernetes-bootcamp --replicas=2

#
kubectl set image deployments/kubernetes-bootcamp
kubernetes-bootcamp=docker.io/jocatalin/kubernetes-bootcamp:v2

#
kubectl rollout undo deployments/kubernetes-bootcamp

# label:
kubectl label node k8s-node01 disktype=ssd

# labels
kubectl get node --show-labels

# label
kubectl label node k8s-node01 disktype-

#
kubectl edit deployment nginx-deployment

# pod
kubectl exec -it -n weave weave-scope-agent-5kzl7 -- /bin/bash
-n: namespaces
weave-scope-agent-5kzl7pod
```

```
#
kubectl log --tail=110 -f etcd-odcbsck8s01 -n kube-system
kubectl log --tail=210 etcd-odcbsck8s01 -n kube-system
```