Kubernetes Commands :

Namespace : Logical grouping of resources.

systemctl status kubeadm

systemctl status kubelet

Check Version:

root@master:~# kubelet --version

Kubernetes v1.20.4

root@master:~#

How to check API version.

kubectl api-resporces

kubectl explain pods | less

kubectl explain deployment | less

root@master:~/YAML# kubectl explain pods | less

root@master:~/YAML#

root@master:~/YAML# kubectl explain deployments | less

root@master:~/YAML# kubectl api-resources

NAME SHORTNAMES APIVERSION NAMESPACED KIND

bindings v1 true Binding

componentstatuses cs v1 false ComponentStatus

configmaps cm v1 true ConfigMap

endpoints ep v1 true Endpoints

events ev v1 true Event

limitranges limits v1 true LimitRange

Steps to forcefully delete a namespace:

**kubectl get ns developer -o json > tmp.json**

vi tmp.json —> remove the Kubernetes from the finalizer , after removing it should look like below.

"name": "dev2",

"resourceVersion": "56683",

"uid": "054727c3-e4da-422a-b641-dec5f84294b4"

},

"spec": {

"finalizers": []

Open another terminal and run below

root@master:~# kubectl proxy

Starting to serve on 127.0.0.1:8001

Then, go to other terminal and run below command :

curl -k -H "Content-Type: application/json" -X PUT --data-binary @tmp.json http://127.0.0.1:8001/api/v1/namespaces/{Your namespace name}/finalize

curl -k -H "Content-Type: application/json" -X PUT --data-binary @tmp.json http://127.0.0.1:8001/api/v1/namespaces/dev2/finalize

curl -k -H "Content-Type: application/json" -X PUT --data-binary @tmp.json http://127.0.0.1:8001/api/v1/namespaces/dev2/finalize

kubectl get pods —all-namespace -o wide

kubectl config set-context --current --namespace=<insert-namespace-name-here>

*# Validate it*

kubectl config view --minify | grep namespace:

kubectl get nodes

kubectl create namespace <name-space>

root@master:~# kubectl create namespace dev2

namespace/dev2 created

root@master:~#

kubectl delete ns <namespace> : kubectl delete ns dev-reg10

root@master:~# kubectl delete ns dev-reg10

namespace "dev-reg10" deleted

root@master:~#

kubectl get ns — to list the namespaces.

root@master:~# kubectl get ns

NAME STATUS AGE

default Active 2d21h

dev2 Active 30s

kube-node-lease Active 2d21h

kube-public Active 2d21h

kube-system Active 2d21h

root@master:~#

kubectl describe ns <name-space>

PODS can be created in two ways.

1. Command Line - Imperative

2. Yaml - Declarative

kubectl run <podname> —image=<image-name> -n <namespace>

root@master:~# kubectl run pod1 --image=httpd -n dev2

pod/pod1 created

root@master:~#

root@master:~# kubectl get pods -n dev2

NAME READY STATUS RESTARTS AGE

pod1 1/1 Running 0 46s

root@master:~#

root@master:~# kubectl get pods -n dev2 -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

pod1 1/1 Running 0 84s 10.46.0.1 worker1 <none> <none>

root@master:~#

kubectl describe pod/<podname> -n <namespace> - to get complete information about the POD.

kubectl apply -f filename.yaml

kubectl edit pod <pod-name> -n <namespace>

kubectl describe pods/<podname> -n <namespace>

kubectl describe pods/myfirst-pod -n dev2

YAML file consists of 4 components:

apiVersion:

kind:

metadata:

spec:

root@master:/yamls# cat first.yaml

apiVersion: v1

kind: Pod

metadata:

name: myfirst-pod

namespace: dev2

labels:

env: dev

customer: GOOGLE

region: US-WEST-COAST

sev: P1

disk: ssd

annotations:

image: httpd-latest

service: web-service

App: dev

spec:

containers:

- name: pscsdev

image: httpd

ports:

- containerPort: 80

root@master:/yamls#

Sample file for mysql database:

root@master:/yamls# cat mysql.yaml

apiVersion: v1

kind: Pod

metadata:

name: mydb-pod

namespace: dev2

labels:

env: devdb

customer: mysql

region: us-west

disk: ssd

annotations:

image: mysql\_db

app: database

spec:

containers:

- name: testsqldb

image: mysql

env:

- name: "MYSQL\_DATABASE"

value: "TESTDB"

- name: "MYSQL\_USER"

value: "mysql"

- name: "MYSQL\_PASSWORD"

value: "sample"

- name: "MYSQL\_ROOT\_PASSWORD"

value: "sample123"

ports:

- containerPort: 3306

root@master:/yamls#

Steps to delete pods.

kubectl delete pods/<podname> -n <namespace>

root@master:/yamls# kubectl delete pods/new-os-pod -n dev2

pod "new-os-pod" deleted

Delete pod using label : kubectl delete pod -l label

kubectl delete pod -l disk=ssd

How to set your namespace as default namespace.

kubectl config set-context --current --namespace <namespace>

Check labels of a POD — kubectl get pods —show-labels

Label pods : kubectl label pod <podname> label

kubectl label pod pod1 disk=ssd

Remove label : kubectl label pod <podname> label -

kubectl label pod pod1 disk-

Create a yaml file using dry run.

kubectl create deployment <pod-name> -n <namepsace> --image=<image-name> —dry-run -o yaml

root@master:/yamls# kubectl create deployment nginx-cont --image=nginx -n dev2 --dry-run -o yaml

W0319 23:01:54.926679 19906 helpers.go:553] --dry-run is deprecated and can be replaced with --dry-run=client.

apiVersion: apps/v1

kind: Deployment

metadata:

creationTimestamp: null

labels:

app: nginx-cont

name: nginx-cont

namespace: dev2

spec:

replicas: 1

selector:

matchLabels:

app: nginx-cont

strategy: {}

template:

metadata:

creationTimestamp: null

labels:

app: nginx-cont

spec:

containers:

- image: nginx

name: nginx

resources: {}

status: {}

root@master:/yamls#

Scale up and scale down the pods.

kubectl get rs n <namespace>

root@master:/yamls# kubectl get rs -n dev2

NAME DESIRED CURRENT READY AGE

pod-tru-dep-7968b885f7 2 2 2 20m

root@master:/yamls#

kubectl scale deployment/<dep-name> —replicas=<number> -n <namespace>

root@master:/yamls# kubectl scale deployment/pod-tru-dep --replicas=10 -n dev2

Scale up/down using yaml file.

After updating the replicas in yaml file, then issue below command

kubectl apply -f <filename> -n <namespace>

root@master:/yamls# kubectl apply -f pods\_deploymet.yaml -n dev2

deployment.apps/pod-tru-dep configured

root@master:/yamls#