kubectl get nodes

kubectl get pods

kubectl get pods --all-namespaces

kubectl get pods -o wide

kubectl get ns

Kubectl create namespace mynamespace

pod.yml

apiVersion: v1

kind: Pod

metadata:

name: tomcatpod

labels:

app: MyApp

spec:

containers:

- name: tomcat

image: tomcat

ports:

- containerPort: 8080

Multicontainer.yml

apiVersion: v1

kind: Pod

metadata:

name: examplepod

spec:

volumes:

- name: html

emptyDir: {}

containers:

- name: webcontainer

image: nginx

volumeMounts:

- name: html

mountPath: /usr/share/nginx/html

- name: filecontainer

image: debian

volumeMounts:

- name: html

mountPath: /html

command: ["/bin/sh", "-c"]

args:

- while true; do

date >> /html/index.html;

sleep 10;

done

kubectl describe pod <podname> # detailed output about a pod in current namespace

kubectl describe pod <podname> -n namespace # detailed output about a pod in current namespace

kubectl describe pod <podname> -o wide # detailed output about a pod wider output

kubectl describe pod <podname> -o yaml # detailed manifest file from apiserver yaml format

kubectl describe pod <podname> -o json

kubectl logs my-pod # dump pod logs (stdout)

kubectl logs -f my-pod # stream pod logs (stdout)

kubectl delete pods <podname>

Replicaset.yml

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: frontend

labels:

app: nginx

tier: frontend

spec:

# modify replicas according to your case

replicas: 3

selector:

matchLabels:

tier: frontend

template:

metadata:

labels:

app: nginx

tier: frontend

spec:

containers:

- name: nginx

image: darealmc/nginx-k8s:v1

ports:

- containerPort: 80

kubectl scale replicates.apps/frontend --replicas=10

apiVersion: v1

kind: Service

metadata:

name: my-internal-service

spec:

selector:

app: nginx

type: ClusterIP

ports:

- name: http

port: 2345

targetPort: 80

protocol: TCP

A picture containing text

Description automatically generated

apiVersion: v1  
kind: Service  
metadata:   
name: my-nodeport-service  
spec:  
selector:   
app: nginx

type: NodePort  
ports:   
- name: http  
port: 80  
targetPort: 80  
nodePort: 30036  
protocol: TCP

A close up of text on a white background

Description automatically generated

Deploy.yml

apiVersion: apps/v1

kind: Deployment

metadata:

name: example-deployment

labels:

app: nginx

spec:

replicas: 5

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: darealmc/nginx-k8s:v1

ports:

- containerPort: 80

Kubectl set image <deploymet>. nginx=darealmc/nginx-k8s:v2

Kubectl rollout undo deployment...

pv.yml

apiVersion: v1

kind: PersistentVolume

metadata:

name: task-pv-volume

labels:

type: local

spec:

storageClassName: manual

capacity:

storage: 10Gi

accessModes:

- ReadWriteOnce

hostPath:

path: "/mnt/data"

OR

apiVersion: v1

kind: PersistentVolume

metadata:

name: pv0001

spec:

capacity:

storage: 5Gi

accessModes:

- ReadWriteOnce

nfs:

path: /tmp

server: 172.17.0.2

persistentVolumeReclaimPolicy: Retain

Pvc.yml

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: tast-pv-claim

spec:

storageClassName: manual

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 3Gi

Multicontainer.yml

apiVersion: v1

kind: Pod

metadata:

name: examplepod

spec:

volumes:

- name: html

PersistentVolumeClaim:

claimName: tasl-pv-claim

containers:

- name: webcontainer

image: nginx

volumeMounts:

- name: html

mountPath: /usr/share/nginx/html

- name: filecontainer

image: debian

volumeMounts:

- name: html

mountPath: /html

command: ["/bin/sh", "-c"]

args:

- while true; do

date >> /html/index.html;

sleep 10;

done

Kubectl cordon kubenode2

Kubectl drain kubenode2

Kubectl uncordon kubenode2