**KUBERNETES**

**Commands:**

**\*\*\*\*\*\*\*\*\*\*\***

**- minikube start**

- To start the Minikube.

**- kubectl apply -f manifest.yml**

- To Create Pod and Run a Container.

**- kubectl get pods**

- Get details of Pods.

- **kubectl get pods -o wide**

- Get exact details where or which Worker Node it got created.

**- kubectl create -f manifest.yml**

- To create a Pod.

**- kubectl describe pod “nameOfPod”**

(or) - To get extra info about Pod.

**- kubectl describe pod/nameOfPod**

**- kubectl logs -f “nameOfPod”**

- To get logs of Pod.

**- kubectl logs -f “nameOfPod” -c c01**

- To check specific container if we have multiple containers.

**- kubectl delete pod “nameOfPod”**

- To delete Pod.

**- kubectl delete manifest.yml**

(or) - To delete manifest.yml file.

**- kubectl delete -f manifest.yml**

**- kubectl delete pod --all**

- To delete all pods

**- kubectl delete -f repcont.yml**

- Delete only resources (pods) i.e replication controller inside the file, not the file.

**- kubectl exec -it “nameOfPod” -c c01 -- /*bin/bash***

*-* To go on to a running container inside Pod.

**Ad-hoc Commands:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**- kubectl create Pod new-nginx:latest** --- generally not recommended (Pods are usually created by deployment). ***#Not Working***

- **kubectl run “pod\_name”--image=nginx --port=8080 --restart=Never**

- To Create a Pod using Image externally in command line instead of manifest.yml.

**- kubectl create ReplicationController new-nginx –image=nginx:latest** --- generally not recommended.

**- Kubectl create deployment new-nginx –image=nginx:latest** --- We can create it (it will create deployment, pod).

**Labels:**

**\*\*\*\*\*\***

**- kubectl get pod --show-labels**

- To see labels of Pods.

**- kubetl get pods -l class=devops**

- To see Pod which use label class=devops (Filtering).

**- kubetl get pods -l class=devops --show-labels**

- To see Pod which use label class=devops (Filtering) with labels.

**- kubetl get pods -l class!=devops**

- To see Pod which use label class!=devops (Filtering).

**- kubetl get pods -l class==devops**

- To see Pod which use label class==devops (Filtering).

**- kubectl get pods -l status=online, status=offline //AND Operation**

- To see Pod which use label status = online and offline (Filtering).

**- kubectl get pod -l ‘env in (devops, development)’ //OR Operation**

- To see Pod which use label whose env = devops and development.

**- kubectl get pod -l ‘env notin (devops, development)’ //OR Operation**

- To see Pod which use label whose env != devops and development.

**Replication Controller:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**- kubectl scale --replicas=8 rc -l company=thbs**

- Pods will be replicated to 8 i.e, 8 pods created of replicationController name.

**- kubectl scale --replicas=2 rc -l company=thbs**

- Pods will be downed to 2 i.e, only 2 pods will be running in replicationController name.

**- kubectl scale --replicas=3 rc -l company=thbs**

- Pods will be replicated to 3 i.e, 3 pods created of replicationController name.

**- kubectl describe rc “replicationControllerName”**

**(or)  *#It is like inspect command***

**- kubectl describe rc/“replicationControllerName”**

**Replication Set:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**- kubectl create -f repSet.yml**

**(or)** - To create Replica Set

**- kubectl apply -f repSet.yml**

**- kubectl get rs [--watch]**

**(or)** -To check status

**- kubectl describe rs nginx**

**(or)**

**- kubectl describe rs/nginx**

**- kubectl scale rs nginx --replicas=3**

- To change No of Replicas

**- kubectl delete rs nginx**

- To delete Replica Set

**Deployment:**

**\*\*\*\*\*\*\*\*\*\*\***

**- kubectl run god --image=god --record**

**- kubectl create -f dep1.yml --record**

**- kubectl get deployment nginx [--watch]**

**- kubectl get deployments**

- Display no of deployments running

**- kubectl des**

**- kubectl scale deployment nginx --replicas=4**

- To change No of pods

------------------------------------------------------------------------------------------

**Manifest.yml file Example:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

---

kind: Pod

apiVersion: v1

metadata:

name: testpod

spec:

containers:

- name: c01

image: ubuntu

command: ["/bin/bash", "-c", "echo Hello-World"]

- name: c02

image: centos

command: ["/bin/bash", "-c", "echo bye-World"]

restartPolicy: Never # Default to always

------------------------------------------------------------------------------------------

**Create Pod with 1 Container and Run it continuously with pause of 5 Seconds:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

---

kind: Pod

apiVersion: v1

metadata:

name: testpod

spec:

containers:

- name: c01

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Hello-World; sleep 5; done"] (or) command: ['sh', '-c', 'echo "Hello, Kubernetes!" && sleep 3600']

restartPolicy: Never # Default to always

------------------------------------------------------------------------------------------

**Create Pod with 3 Container and Run it continuously with pause of 5 Seconds:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

---

kind: Pod

apiVersion: v1

metadata:

name: testpod

spec:

containers:

- name: c01

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Hello-World; sleep 5; done"]

- name: c02

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Hello-World; sleep 5; done"]

- name: c03

image: nigil1999/thbs\_docker:nig\_monday\_img #Our own image from Docker Hub Repository.

command: ["/bin/bash", "-c", "while true; do echo Hello-World; sleep 5; done"]

restartPolicy: Never

**Example:**

**\*\*\*\*\*\*\*\***

apiVersion: batch/v1

kind: Job

metadata:

name: hello

spec:

template: # This is the pod template

spec:

containers:

- name: hello

image: busybox

command: ['sh', '-c', 'echo "Hello, Kubernetes!" && sleep 3600']

restartPolicy: OnFailure # The pod template ends here

------------------------------------------------------------------------------------------

**To Create Pod with Variable:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#gedit manithird.yml

---

kind: Pod

apiVersion: v1

metadata:

name: myenv-pod

annotations:

description: This is pod with variable $MYNAME

spec:

containers:

- name: c01

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Hello-World; sleep 5; done"]

env:

- name: MYNAME

value: THBS

restartPolicy: Never

# kubectl apply -f maniThird.yml

# kubectl exec ~~it~~ myenvpod -c c01 --/bin/bash

/# echo $MYNAME – output: THBS

------------------------------------------------------------------------------------------

**To Create Pod to install Apache server with Port:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#gedit manithird.yml

---

kind: Pod

apiVersion: v1

metadata:

name: httpd-pod

spec:

containers:

- name: c01

image: httpd

ports:

- containerPort: 80

restartPolicy: Never

# kubectl apply -f maniFourth.yml

# kubectl exec it httpd-pod -c c01 --/bin/bash

# kubectl get pods o wide // Copy IpAddress of httpdpod

# curl IpAdress (or) curl IpAddress:80 //Show html file

------------------------------------------------------------------------------------------

**Create Pod with Label :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

---

kind: Pod

apiVersion: v1

metadata:

name: testpod

labels:

env: stage

class: devops

skill: cloud

spec:

containers:

- name: c01

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Hello-World; sleep 5; done"]

restartPolicy: Never

# kubectl apply -f labelPod.yml

# kubectl get pod –show-labels //Show labels of Pods

------------------------------------------------------------------------------------------

**Create Pod with NodeSelector :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

---

kind: Pod

apiVersion: v1

metadata:

name: tbs-pod

labels:

env: development

spec:

containers:

- name: c01

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Hello-World; sleep 5; done"]

nodeSelector:

cmpy: thbs

restartPolicy: Never

# kubectl apply -f nodeSelectorPod.yml

# kubectl get pods // Show pod is pending

# kubectl label nodes nameOfNode cmpy=thbs //to apply lable to node

# kubectl apply -f nodeSelectorPod.yml

# kubectl get pods // Show pod is running

------------------------------------------------------------------------------------------

**Create Pod with ReplicationController :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

---

kind: ReplicationController

apiVersion: v1

metadata:

name: replcntl ***#name of replication controller***

annotations:

description: this is replication controller

spec:

replicas: 3

selector:

company: thbs

template: #Pod Area

metadata:

name: tbs-pod ***#name of pod***

labels:

company: thbs

spec:

containers: ***#name of container***

- name: c01

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Hello-World; sleep 5; done"]

env:

- name: MYNAME

value: THBS

# kubectl apply -f replicationControllerPod.yml

# kubectl get pods

# kubectl scale --replicas=8 rc -l company=thbs // Pods will be replicated to 8 i.e, 8 pods created of replicationController name

# kubectl scale --replicas=2 rc -l company=thbs // Pods will be downed to 2 i.e, only 2 pods will be running in replicationController name

# kubectl scale --replicas=3 rc -l company=thbs // Pods will be replicated to 3 i.e, 3 pods created of replicationController name

------------------------------------------------------------------------------------------

**Create Pod with ReplicationSet :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

---

kind: ReplicaSet

apiVersion: apps/v1

metadata:

name: replica-set ***#name of replica set***

labels:

cmp: thbs

spec: ***#Specification of replica set***

replicas: 3

selector:

matchExpressions:

- {key: env, operator: In, values: [prod, stage]}

template:  ***#Pod Area***

metadata:

name: rspod ***#name of pod***

labels:

env: stage

spec: ***#Specification of pod***

containers:

- name: rsc01 ***#name of container***

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Hello-World; sleep 5; done"]

# kubectl apply -f replicationSetPod.yml

# kubectl get pods

# kubectl scale --replicas=8 rs -l env=stage // Pods will be replicated to 8 i.e, 8 pods created of replicationSet name

------------------------------------------------------------------------------------------

**Create Pod with Deployment :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-dep ***#name of deployment***

labels:

app: web

spec:

replicas: 3

selector:

matchLabels:

mypod: nginxpod

template:  ***#Pod Area***

metadata:

name: deppod ***#name of pod***

labels:

mypod: nginxpod

spec: ***#Specification of pod***

containers:

- name: depc01 ***#name of container***

image: nginx

ports:

- containerPort: 80

# kubectl apply -f deploymentPod.yml

# kubectl get pods

# kubectl get deployments

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Create Pod with Volume :**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

---

kind: Pod

apiVersion: v1

metadata:

name: emptyvol

spec:

containers:

- name: c01

image: ubuntu

command: [“/bin/bash”, “-c”, “sleep 15000”]

volumeMounts:

- name: xchange

mountPath: /*tmp/*nighp

- name: c02

image: ubuntu

command: [“/bin/bash”, “-c”, “sleep 15000”]

volumeMounts:

- name: xchange

mountPath: /*tmp/*thbs

volumes:

- name: xchange

emptyDir: {}