**CKAD Lab Guide**

Day – 01

TASK - 01

#kubectl get nodes

#kubectl get nodes -o wide

#kubectl run nginx --image=nginx:1.22.0

#kubectl get pods

#kubectl get pods -A

#kubectl get pods -o wide

#kubectl exec -it nginx -- sh

#kubectl describe pod <pod-name> | less

#kubectl explain pod

#kubectl describe node <node-name> | less

#kubectl cluster-info

TASK - 02

#vim multicon.yml

apiVersion: v1

kind: Pod

metadata:

name: multicon

spec:

containers:

- name: cont1

image: quay.io/gauravkumar9130/mywebapp

imagePullPolicy: IfNotPresent

- name: cont2

image: redis

imagePullPolicy: IfNotPresent

#kubectl create -f multicon.yml

#kubectl get pods -o wide

#kubectl exec -it multicon -- sh

#kubectl exec -it multicon -c cont2 -- sh

#kubectl delete -f multicon.yml

#kubectl get nodes -o wide

#kubectl debug node/<node-name> -it --image=mcr.microsoft.com/dotnet/runtime-deps:6.0

#exit

#kubectl get pods -o wide

#kubectl delete pod nginx

#kubectl get pods -o wide

TASK - 03

#vim label.yml

apiVersion: v1

kind: Pod

metadata:

name: dev-pod

labels:

env: dev

manager: gaurav

spec:

containers:

- name: abc

image: quay.io/gauravkumar9130/nginxdemo

#kubectl create -f label.yml

#kubectl get pods -o wide

#kubectl get pods --show-labels

#kubectl run pod1 --image=quay.io/gauravkumar9130/mywebapp

#kubectl get pods --show-labels

#kubectl label pod pod1 env=dev

#kubectl get pods --show-labels

#kubectl label pod pod1 run-

#kubectl get pods --show-labels

#kubectl run pod2 --image=quay.io/gauravkumar9130/mywebapp

#kubectl run pod3 --image=quay.io/gauravkumar9130/mywebapp -l env=prod

#kubectl get pods --show-labels

#kubectl label --overwrite pod pod2 env=test

#kubectl get pods --show-labels

#kubectl label --overwrite pod pod2 env=test2

#kubectl get pods --show-labels

#kubectl label --overwrite pod pod2 env=test

#kubectl get pods --show-labels

#kubectl get pods --selector env=dev

#kubectl get pods --selector env!=dev

#kubectl get pods --selector 'env in (test,dev)'

#kubectl get pods --selector 'env notin (test,dev)'

#kubectl delete pod --all

#kubectl get pods

TASK - 04A

#vim rs.yml

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: rs-web

spec:

replicas: 5

selector:

matchLabels:

app: web

template:

metadata:

labels:

app: web

spec:

containers:

- name: mycontainer

image: quay.io/gauravkumar9130/nginxdemo

#kubectl create -f rs.yml

#kubectl get pods -o wide

#kubectl get rs

#kubectl scale rs rs-web --replicas=3

#kubectl get rs

#kubectl get pods -o wide

#kubectl scale rs rs-web --replicas=5

#kubectl get rs

#kubectl get pods -o wide

#kubectl delete -f rs.yml

TASK - 04B

#vim rs-setbased.yml

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: rs-web

spec:

replicas: 5

selector:

matchExpressions:

- key: "app"

operator: "In"

values:

- "nginx"

- "web"

template:

metadata:

labels:

app: web

spec:

containers:

- name: mycontainer

image: quay.io/gauravkumar9130/nginxdemo

#kubectl create -f rs-setbased.yml

#kubectl get pods -o wide

#kubectl get rs

#kubectl get pods --show-labels

#kubectl delete -f rs-setbased.yml

#kubectl get pods -o wide

TASK - 05

#vim ecom.yml

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: ecommerce

spec:

replicas: 5

selector:

matchLabels:

app: ecommerce

template:

metadata:

labels:

app: ecommerce

spec:

containers:

- name: abc

image: quay.io/gauravkumar9130/mywebapp

#kubectl create -f ecom.yml

#kubectl get pods -o wide

#kubectl get pods --show-labels

#kubectl get svc

PART1 - SERVICE TYPE CLUSTER IP

#vim cip-ecom.yml

apiVersion: v1

kind: Service

metadata:

name: cip-ecommerce

spec:

type: ClusterIP

ports:

- targetPort: 80 ##container port no

port: 5000 ##clusterip port no

selector:

app: ecommerce

#kubectl create -f cip-ecom.yml

#kubectl get svc

#kubectl get nodes

#kubectl debug node/<node-name> -it --image=mcr.microsoft.com/dotnet/runtime-deps:6.0

#apt-get update

#apt-get install curl

#curl 10.0.131.58:5000

#exit

#kubectl delete -f cip-ecom.yml

#kubectl get svc

PART2 - SERVICE TYPE NODEPORT

#vim nodep.yml

apiVersion: v1

kind: Service

metadata:

name: ecommerce-outside-app

spec:

type: NodePort

ports:

- targetPort: 80 ##container port no

port: 80 ##cluster ip port

nodePort: 30003 ####range between 30000-32767 only allowed

selector:

app: ecommerce

#kubectl create -f nodep.yml

#kubectl get svc

#kubectl get nodes -o wide

#kubectl debug node/<node-name> -it --image=mcr.microsoft.com/dotnet/runtime-deps:6.0

#apt-get update

#apt-get install curl

#curl <node-IP>:30003

#exit

#kubectl delete -f nodep.yml

#kubectl get svc

PART3 - SERVICE TYPE LOAD BALANCE

#vim lb.yml

apiVersion: v1

kind: Service

metadata:

name: cip-ecommerce

spec:

type: LoadBalancer

ports:

- targetPort: 80

port: 80

selector:

app: ecommerce

#kubectl create -f lb.yml

#kubectl get svc

Note - Open a new tab in chrome browser and access the Load Balancer public IP

#kubectl delete -f lb.yml

#kubectl get rs

#kubectl delete -f ecom.yml

#kubectl get rs

TASK - 05

#kubectl get nodes -o wide

#vim manual.yml8

apiVersion: v1

kind: Pod

metadata:

name: my-custom-pod

spec:

containers:

- name: abc

image: quay.io/gauravkumar9130/nginxdemo

nodeName: aks-agentpool-97484218-vmss000002

#kubectl create -f manual.yml

#kubectl get pods -o wide

#kubectl delete -f manual.yml

#kubectl get pods -o wide

Day-02

ASK - 01

#kubectl get nodes -o wide

#kubectl describe node <node name> | grep -i taint

#kubectl taint nodes <node name> app=blue:NoSchedule

#kubectl describe node <node name> | grep -i taint

#vim toleration.yml

apiVersion: v1

kind: Pod

metadata:

name: mypod

spec:

containers:

- name: mycontainer

image: quay.io/gauravkumar9130/nginxdemo

tolerations:

- key: "app"

operator: "Equal"

value: "blue"

effect: "NoSchedule"

#kubectl create -f toleration.yml

#kubectl get pods -o wide

#kubectl delete -f toleration.yml

#kubectl taint nodes <node name> app-

#kubectl describe node <node name> | grep -i taint

TASK - 02

#kubectl get nodes -o wide

#kubectl label node <node name> size=large

#kubectl get nodes --show-labels

#kubectl describe node <node name> | grep -i size

#vim selector.yml

apiVersion: v1

kind: Pod

metadata:

name: newpod

spec:

containers:

- name: newcontainer

image: quay.io/gauravkumar9130/nginxdemo

nodeSelector:

size: large

#kubectl create -f selector.yml

#kubectl get pods -o wide

#kubectl delete -f selector.yml

#kubectl label node <nodename> size-

TASK - 03

#kubectl get nodes -o wide

#kubectl label node <node1 name> size=large

#kubectl label node <node2 name> size=medium

#kubectl describe node <node1 name> | grep -i size

#kubectl describe node <node2 name> | grep -i size

#vim affinity.yml

apiVersion: v1

kind: Pod

metadata:

name: mypod

spec:

containers:

- name: abc

image: quay.io/gauravkumar9130/nginxdemo

affinity:

nodeAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:

- key: size

operator: In

values:

- large

- medium

#kubectl create -f affinity.yml

#kubectl get pods -o wide

#kubectl delete -f affinity.yml

#kubectl label node <node1-name> size-

#kubectl label node <node2-name> size-

#kubectl describe node <node1 name> | grep -i size

#kubectl describe node <node2 name> | grep -i size

TASK - 04

#vim dep.yml

apiVersion: apps/v1

kind: Deployment

metadata:

name: myapp

spec:

replicas: 5

selector:

matchLabels:

app: myapp

template:

metadata:

labels:

app: myapp

spec:

containers:

- name: c1

image: quay.io/gauravkumar9130/production:v1

#kubectl create -f dep.yml

#kubectl get deployment

#kubectl expose deployment myapp --target-port=80 --port=80 --type=LoadBalancer

#kubectl get svc

#kubectl describe deployment myapp | less

#kubectl set image deployment myapp c1=quay.io/gauravkumar9130/production:v2

#kubectl rollout history deployment myapp

#kubectl set image deployment myapp c1=quay.io/gauravkumar9130/production:v3 --record

#kubectl rollout history deployment myapp

#kubectl rollout undo deployment myapp

#kubectl rollout history deployment myapp

#kubectl rollout undo deployment myapp

#kubectl rollout undo deployment myapp --to-revision=1

TASK - 05

#vim blue.yml

apiVersion: apps/v1

kind: Deployment

metadata:

name: blue-deployment

spec:

replicas: 5

selector:

matchLabels:

app: nginx

version: blue ##it can be anything

template:

metadata:

labels:

app: nginx

version: blue

spec:

containers:

- name: abc

image: quay.io/gauravkumar9130/production:v1

#vim bgsvc.yml

apiVersion: v1

kind: Service

metadata:

name: bgsvc

spec:

type: LoadBalancer

ports:

- port: 80

targetPort: 80

selector:

version: blue

#kubectl create -f blue.yml

#kubectl create -f bgsvc.yml

Note - Open new tab in web browser and paste the load balancer IP address

#vim green.yml

apiVersion: apps/v1

kind: Deployment

metadata:

name: green-deployment

spec:

replicas: 5

selector:

matchLabels:

app: nginx

version: green ##it can be anything

template:

metadata:

labels:

app: nginx

version: green

spec:

containers:

- name: abc

image: quay.io/gauravkumar9130/production:v2

#kubectl create -f green.yml

#kubectl edit svc bgsvc

Note - change under selector version: green

Note - refresh the web browser tab from where you access application earlier

TASK - 06

#kubectl get ns

#kubectl get sa

#kubectl describe sa default -n default | less

#kubectl get secret

#kubectl create ns myns

#kubectl get ns

#kubectl create sa sam -n myns

#kubectl get sa -n myns

#kubectl get secret -n myns

#vim podsa.yml

apiVersion: v1

kind: Pod

metadata:

name: podsa

namespace: myns

spec:

containers:

- name: c1

image: quay.io/gauravkumar9130/nginx

serviceAccountName: sam

#kubectl create -f podsa.yml

#kubectl delete -f podsa.yml

TASK - 07

- PLAIN KEY

#vim plainenv.yml

apiVersion: v1

kind: Pod

metadata:

name: plain

spec:

containers:

- name: c1

image: quay.io/gauravkumar9130/mysql

env:

- name: MYSQL\_ROOT\_PASSWORD

value: myroot

- name: MYSQL\_USER

value: sam

- name: MYSQL\_PASSWORD

value: sam12345

#kubectl create -f plainenv.yml

#kubectl exec -it plain -- bash

#env

#exit

#kubectl delete -f plainenv.yml

- CONFIG MAP

#kubectl create cm conf --from-literal=MYSQL\_ROOT\_PASSWORD=root1234 --from-literal=MYSQL\_USER=sam --from-literal=MYSQL\_PASSWORD=sam12345

#kubectl get cm

#kubectl describe cm conf | less

#vim configmapenv.yml

apiVersion: v1

kind: Pod

metadata:

name: cm

spec:

containers:

- name: c1

image: quay.io/gauravkumar9130/mysql

envFrom:

- configMapRef:

name: conf

#kubectl create -f configmapenv.yml

#kubectl exec -it cm -- bash

#env

#exit

#kubectl delete -f configmapenv.yml

#kubectl delete cm conf

- SECRET

#kubectl create secret generic sec --from-literal=MYSQL\_ROOT\_PASSWORD=root1234 --from-literal=MYSQL\_USER=sam --from-literal=MYSQL\_PASSWORD=sam12345

#kubectl get secret

#kubectl describe secret sec

#vim secevn.yml

apiVersion: v1

kind: Pod

metadata:

name: sec

spec:

containers:

- name: c1

image: quay.io/gauravkumar9130/mysql

envFrom:

- secretRef:

name: sec

#kubectl create -f secevn.yml

#kubectl exec -it sec -- bash

#env

#exit

#kubectl delete -f secevn.yml

#kubectl delete secret sec

- CONFIG MAP AS A VOLUME

#kubectl create cm db-config-vol --from-file=multicon.yml

#kubectl get cm

#vim cmvol.yml

apiVersion: v1

kind: Pod

metadata:

name: abc

spec:

volumes:

- name: myvol

configMap:

name: db-config-vol

containers:

- name: db

image: quay.io/gauravkumar9130/nginxdemo

volumeMounts:

- name: myvol ##same name as volume name

mountPath: /data ##data folder will be created automatically

#kubectl create -f cmvol.yml

#kubectl exec -it abc -- sh

#pwd

#cd data

#ls

#exit

#kubectl delete -f cmvol.yml

#kubectl delete cm db-config-vol

TASK - 08

- EMPTYDIR

#vim emptydir.yml

apiVersion: v1

kind: Pod

metadata:

name: mypod

spec:

volumes:

- name: cache

emptyDir: {}

containers:

- name: c1

image: quay.io/gauravkumar9130/nginx

volumeMounts:

- name: cache

mountPath: /mydata

#kubectl create -f emptydir.yml

#kubectl get pods -o wide

#kubectl exec -it mypod -- sh

#pwd

#cd mydata

#touch file1

#exit

#kubectl debug node/aks-agentpool-28081813-vmss000000 -it --image=mcr.microsoft.com/dotnet/runtime-deps:6.0

#find / -name file1

#cd /host/var/lib/kubelet/pods/<string>/volumes/kubernetes.io~empty-dir/cache

#ls

#exit

#kubectl delete -f emptydir.yml

#kubectl debug node/aks-agentpool-28081813-vmss000000 -it --image=mcr.microsoft.com/dotnet/runtime-deps:6.0

#cd /host/var/lib/kubelet/pods [no <string> will be found in pods dir]

#exit

- PV AND PVC

#vim pvol.yml

apiVersion: v1

kind: PersistentVolume

metadata:

name: pv

spec:

storageClassName: hdd

accessModes:

- ReadWriteMany

capacity:

storage: "3Gi"

hostPath:

path: "/insidenode"

#kubectl create -f pvol.yml

#kubectl get pv

#vim pvclaim.yml

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: pvc

spec:

storageClassName: hdd

accessModes:

- ReadWriteMany

resources:

requests:

storage: "2Gi"

#kubectl create -f pvclaim.yml

#kubectl get pvc

#vim podpv.yml

apiVersion: v1

kind: Pod

metadata:

name: vpod

spec:

volumes:

- name: pv

persistentVolumeClaim:

claimName: pvc

containers:

- name: c1

image: quay.io/gauravkumar9130/nginx

volumeMounts:

- name: pv

mountPath: /usr/share/nginx/html

#kubectl create -f podpv.yml

#kubectl get pods -o wide

#kubectl exec -it vpod -- sh

#pwd

#cd usr/share/nginx/html

#touch file1

#ls

#exit

#kubectl debug node/<node-name> -it --image=mcr.microsoft.com/dotnet/runtime-deps:6.0

#ls host/insidenode

#exit

#kubectl delete -f podpv.yml

#kubectl debug node/<node-name> -it --image=mcr.microsoft.com/dotnet/runtime-deps:6.0

#ls host/insidenode

#exit

#kubectl delete -f pvclaim.yml

#kubectl delete -f pvol.yml

- PV AS AZURE DISK

#vim azpvc.yml

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: azpvc

spec:

storageClassName: default

accessModes:

- ReadWriteOnce

resources:

requests:

storage: "2Gi"

#kubectl create -f azpvc.yml

#kubectl get pvc

#vim azpod.yml

apiVersion: v1

kind: Pod

metadata:

name: azpod

spec:

volumes:

- name: vol

persistentVolumeClaim:

claimName: azpvc

containers:

- name: c1

image: quay.io/gauravkumar9130/nginx

volumeMounts:

- name: vol

mountPath: /usr/share/nginx/html

#kubectl create -f azpod.yml

#kubectl exec -it azpod -- sh

#df -hT | grep /usr/share/nginx/html

#cd /usr/share/nginx/html

#touch file1

#ls

#exit

#kubectl get pods -o wide

#kubectl debug node/aks-agentpool-18777746-vmss000000 -it --image=mcr.microsoft.com/dotnet/runtime-deps:6.0

#df -hT | grep /dev/sdc

#cd /host/var/lib/kubelet/plugins/kubernetes.io/csi/pv/pvc-b73f8711-f40f-42ef-b94a-9574f3821dd5/globalmount/

#ls

#exit

#kubectl delete -f azpod.yml

#kubectl debug node/aks-agentpool-18777746-vmss000000 -it --image=mcr.microsoft.com/dotnet/runtime-deps:6.0

#df -hT | grep /dev/sdc

#exit

TASK - 09

kubectl run web --image=quay.io/gauravkumar9130/nginxdemo -l app=web

kubectl run db --image=quay.io/gauravkumar9130/nginxdemo -l app=db

kubectl run test --image=quay.io/gauravkumar9130/nginxdemo -l app=test

kubectl get pods --show-labels

kubectl get pods -o wide

kubectl exec -it web -- sh

kubectl exec -it test -- sh

vim deny.yml

apiVersion: networking.k8s.io/v1

kind: NetworkPolicy

metadata:

name: deny-policy

spec:

podSelector:

matchLabels: {}

policyTypes:

- Ingress

kubectl create -f deny.yml

kubectl get netpol

kubectl get pods -o wide

kubectl exec -it web -- sh

kubectl exec -it test -- sh

kubectl get pods --show-labels

vim db-web-netpol.yml

apiVersion: networking.k8s.io/v1

kind: NetworkPolicy

metadata:

name: db-policy

spec:

podSelector:

matchLabels:

app: db

policyTypes:

- Ingress

ingress:

- from:

- podSelector:

matchLabels:

app: web

kubectl create -f db-web-netpol.yml

kubectl get pods -o wide

kubectl exec -it web -- sh

kubectl exec -it test -- sh

kubectl delete -f db-web-netpol.yml

DB to WEB and WEB to DB

apiVersion: networking.k8s.io/v1

kind: NetworkPolicy

metadata:

name: db-policy

spec:

podSelector:

matchLabels:

app: db

policyTypes:

- Ingress

- Egress

ingress:

- from:

- podSelector:

matchLabels:

app: web

egress:

- to:

- podSelector:

matchLabels:

app: web

TASK - 10

git clone https://github.com/kubernetes/ingress-nginx

cd ingress-nginx/deploy/static/provider/cloud

kubectl create -f deploy.yaml

kubectl get ns

kubectl get pods -n ingress-nginx

kubectl get svc -n ingress-nginx

kubectl create deployment hotel --image=quay.io/gauravkumar9130/hotel --replicas=4

kubectl create deployment tea --image=quay.io/gauravkumar9130/tea --replicas=4

kubectl create deployment coffee --image=quay.io/gauravkumar9130/coffee --replicas=4

kubectl get pods

kubectl expose deployment hotel --target-port=80 --port=80

kubectl expose deployment tea --target-port=80 --port=80

kubectl expose deployment coffee --target-port=80 --port=80

kubectl get deploy

kubectl get svc

cd

cd myks

vim ingress.yml

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: hotel-app

annotations:

kubernetes.io/ingress.class: nginx

spec:

rules:

- http:

paths:

- path: /hotel

pathType: Prefix

backend:

service:

name: hotel

port:

number: 80

- path: /tea

pathType: Prefix

backend:

service:

name: tea

port:

number: 80

- path: /coffee

pathType: Prefix

backend:

service:

name: coffee

port:

number: 80

kubectl create -f ingress.yml

kubectl get ingress

kubectl get svc -n ingress-nginx

kubectl delete -f ingress.yml

kubectl get ingress

kubectl get svc

kubectl delete svc hotel

kubectl delete svc tea

kubectl delete svc coffee

kubectl get deployment

kubectl delete deployment hotel

kubectl delete deployment tea

kubectl delete deployment coffee

kubectl get pods

Day – 03

STATEFULSETS [TASK - 1]

kubectl create -f rs.yml

kubectl get pods -o wide

kubectl delete -f rs.yml

vim sf.yml

apiVersion: apps/v1

kind: StatefulSet

metadata:

name: myweb

spec:

serviceName: websvc

selector:

matchLabels:

app: web

replicas: 4

template:

metadata:

labels:

app: web

spec:

containers:

- name: abc

image: quay.io/gauravkumar9130/nginxdemo

kubectl create -f sf.yml

kubectl get pods -o wide

kubectl delete pod myweb-1

kubectl get pods

kubectl scale statefulset myweb --replicas=6

kubectl get pods

kubectl scale statefulset myweb --replicas=3

kubectl get pods

vim hdsvc.yml

apiVersion: v1

kind: Service

metadata:

name: websvc

spec:

ports:

- targetPort: 80

port: 80

selector:

app: web

clusterIP: None

kubectl create -f hdsvc.yml

kubectl get pods

kubectl get svc

#myweb-2.websvc.default.svc.cluster.local

kubectl run test-dns -it --rm --image=centos

curl http://myweb-0.websvc.default.svc.cluster.local

curl http://myweb-1.websvc.default.svc.cluster.local

kubectl get pods

kubectl get pods -o wide

cp sf.yml sfst.yml

vim sfst.yml

apiVersion: apps/v1

kind: StatefulSet

metadata:

name: myweb-statefulset

spec:

serviceName: websvc-hsvc

replicas: 3

selector:

matchLabels:

app: web

template:

metadata:

labels:

app: web

spec:

containers:

- name: abc

image: quay.io/gauravkumar9130/nginxdemo

volumeMounts:

- mountPath: /mydata

name: data-volume

volumeClaimTemplates:

- metadata:

name: data-volume

spec:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 500Mi

kubectl create -f sfst.yml

kubectl get pvc

kubectl get pv

READINESS LIVENESS [TASK - 2]

#vim nordsvc.yml

apiVersion: v1

kind: Service

metadata:

name: nordsvc

spec:

ports:

- targetPort: 8080

port: 8080

selector:

app: jenkins

type: LoadBalancer

kubectl create -f nordsvc.yml

#vim rddep.yml

apiVersion: apps/v1

kind: Deployment

metadata:

name: rddep

spec:

selector:

matchLabels:

app: jenkins

replicas: 5

template:

metadata:

labels:

app: jenkins

spec:

containers:

- name: c1

image: quay.io/gauravkumar9130/jenkins

#kubectl get svc

#kubectl create -f rddep.yml

Note - copy LB-IP:8080 and paste in a new web browser tab, you should see JENKINS IS INITIALIZING page

kubectl delete -f rddep.yml

kubectl create deployment rdnessdep --image=quay.io/gauravkumar9130/jenkins --replicas=5 --dry-run=client -o yaml > rdnessdep.yml

vim rdnessdep.yml

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

app: rdnessdep

name: rdnessdep

spec:

replicas: 5

selector:

matchLabels:

app: jenkins

strategy: {}

template:

metadata:

labels:

app: jenkins

spec:

containers:

- image: quay.io/gauravkumar9130/jenkins

name: jenkins

readinessProbe:

httpGet:

path: /login

port: 8080

initialDelaySeconds: 5

periodSeconds: 5

kubectl create -f rdnessdep.yml

kubectl get pods

Note - copy LB-IP:8080 and paste in a new web browser tab, you should see JENKINS LOGIN page

kubectl delete -f rdnessdep.yml

cp rdnessdep.yml rdlvnessdep.yml

vim rdlvnessdep.yml

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

app: rdnessdep

name: rdnessdep

spec:

replicas: 5

selector:

matchLabels:

app: jenkins

strategy: {}

template:

metadata:

labels:

app: jenkins

spec:

containers:

- image: quay.io/gauravkumar9130/jenkins

name: jenkins

readinessProbe:

httpGet:

path: /login

port: 8080

initialDelaySeconds: 5

periodSeconds: 5

livenessProbe:

exec:

command: ["ls","/usr/share/jenkins/jenkins.war"]

initialDelaySeconds: 5

periodSeconds: 5

kubectl create -f rdlvnessdep.yml

kubectl get svc

kubectl get pods

Note - copy LB-IP:8080 and paste in a new web browser tab, you should see JENKINS LOGIN page

LOGGING AND MONITORING [TASK - 3]

vim logging.yml

apiVersion: v1

kind: Pod

metadata:

name: mypod

spec:

containers:

- name: c1

image: quay.io/gauravkumar9130/mywebapp

kubectl create -f logging.yml

kubectl get pods

kubectl describe pod mypod

kubectl logs mypod

kubectl logs mypod -c c1

git clone https://github.com/gauravkumar9130/grafana

cd grafana/

ls

kubectl create -f 1-prometheus/.

kubectl create -f 2-grafana/.

kubectl get ns

kubectl get pods -n monitoring

kubectl get svc -n monitoring

In web browser copy PUBLIC IP:3000

Note - https://grafana.com/grafana/dashboards/6417

git clone https://github.com/gauravkumar9130/kube-elk

cd kube-elk/

cat Instructions

kubectl create -f 1-elastic.yml

kubectl get pods

kubectl create -f 2-kibana.yml

kubectl get pods

kubectl create -f 3-filebeat.yml >> server system

git clone https://github.com/kubernetes/kube-state-metrics.git

kubectl apply -f kube-state-metrics/examples/standard/.

kubectl create -f 4-metricbeat.yml

kubectl get pods

kubectl get svc

In web browser copy KIBANA PUBLIC IP:5601

Day – 04

SIDECAR [TASK - 1]

vim sidecar.yml

apiVersion: v1

kind: Pod

metadata:

name: nginx-app

spec:

volumes:

- name: myvol

emptyDir: {}

containers:

- name: nginx-main-app

image: quay.io/gauravkumar9130/nginx

volumeMounts:

- name: myvol

mountPath: /usr/share/nginx/html

- name: side-car

image: quay.io/gauravkumar9130/ubuntu-git

command: ["/bin/sh"]

args: ["-c","while true; do git clone https://github.com/gauravkumar9130/webpage; cd webpage; mv \* /website/; sleep 10; done"]

volumeMounts:

- name: myvol

mountPath: /website

kubectl create -f sidecar.yml

kubectl get pods

vim sidecar.yml

kubectl label pod nginx-app app=nginx

kubectl expose pod nginx-app --target-port=80 --port=80 --type=LoadBalancer

kubectl get svc

kubectl delete -f sidecar.yml

CRONJOB AND JOB [TASK - 2]

vim cronjob.yml

apiVersion: batch/v1

kind: CronJob

metadata:

name: cronjob

spec:

schedule: "\*/1 \* \* \* \*"

jobTemplate:

spec:

template:

spec:

containers:

- name: pod

image: quay.io/gauravkumar9130/busybox

command:

- /bin/sh

- -c

- date; echo hello from the Kubernetes cluster

restartPolicy: OnFailure

kubectl create -f cronjob.yml

kubectl get cronjobs

kubectl get pods

kubectl logs pod-name

kubectl get job -w

kubectl delete -f cronjob.yml

vim job.yml

apiVersion: batch/v1

kind: Job

metadata:

name: first-job

spec:

template:

spec:

containers:

- name: c1

image: quay.io/gauravkumar9130/busybox

command: ["/bin/sh"]

args: ["-c","echo Hello World"]

restartPolicy: Never

kubectl create -f job.yml

kubectl get pods

kubectl get jobs

kubectl logs pod-name

kubectl delete -f job.yml

Day – 05

Helm [Task - 01]

--- Using Bitnami Repo ---

helm repo add bitnami https://charts.bitnami.com/bitnami

helm repo list

helm search repo bitnami | less

helm install apacheapp bitnami/tomcat

kubectl get pods

kubectl get svc

helm list

helm uninstall apacheapp

helm repo remove bitnami

helm repo list

--- Using Custom Charts ---

mkdir charts

cd charts

helm create mycharts

ls

cd mycharts

ls

cp Chart.yaml Chartbk.yaml

ls

vim Chart.yaml

apiVersion: v2

name: mycharts

description: My Helm chart for Kubernetes

type: application

version: 0.1.0

appVersion: "1.16.0"

Note - The Charts.yaml file should look like above

cd templates

cp vim deployment.yaml vim deploymentbk.yaml

vim deployment.yaml

line 34: image: "{{ .Values.image.repository }}"

delete line 40 - 47 [remove readiness and liveness block]

Note - The deployment.yaml file should have above changes

cd ..

cp values.yaml valuesbk.yaml

vim values.yaml

line 5 - replicaCount: 3

line 8 - repository: quay.io/gauravkumar9130/production:v1

line 40 - type: LoadBalancer

Note - The values.yaml file should have above changes

cd ..

helm install myapp mycharts

kubectl get pods

kubectl get svc

helm uninstall myapp

Kubernetes Installation [Task - 02]

- Installation environment has 3 nodes

1. Configure Local DNS (On Master)

#vim /etc/hosts

MASTERMACHINEIP master

WORKER1MACHINEIP worker1

WORKER2CHINEIP worker2

#scp /etc/hosts worker1:/etc/

#scp /etc/hosts worker2:/etc/

2. Install Container Runtime: (ON ALL)

#yum install cri-o -y

#systemctl start crio

#systemctl enable crio

3. Install Kubernetes Packages: (ON ALL)

#yum install kubectl kubelet kubeadm -y

#systemctl daemon-reload

#systemctl start kubelet

#systemctl enable kubelet

4. Initialize Kubernetes Cluster: (ON MASTER)

#kubeadm init --pod-network-cidr=10.244.0.0/16

NOTE: ABOVE COMMAND WILL GIVE USER CONFIGURATION SO COPY THREE LINES from mkdir COMMAND AND PASTE ON MASTER, COPY JOIN COMMAND AND PASTE ON ALL THE WORKER NODES.

5. Install Calico: (ON MASTER)

#wget https://projectcalico.docs.tigera.io/manifests/calico.yaml --no-check-certficate

#gedit calico.yaml

press control+h (for replace):

in find: docker.io

in replace with: quay.io

save file

#kubectl apply/create -f calico.yaml

6. Verify: (ON MASTER)

#kubectl get nodes

#kubectl get pods -A