1 clear

2 sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installDocker.sh -P /tmp

3 sudo chmod 755 /tmp/installDocker.sh

4 sudo bash /tmp/installDocker.sh

5 docker -v

6 docker pull jenkins:2.60.3-alpine

7 docker pull tomcat:8.5.40

9 docker image ls

10 docker pull mongo

11 docker image ls

12 docker image rm mongo:latest

13 docker image ls

18 docker image ls

19 docker run tomcat:8.5.40 **(use Ctrl + c to come out)**

20 docker run -d tomcat:8.5.40

21 docker run -d jenkins:2.60.3-alpine

22 docker ps -a # it shows running /stopped / killed containers on your vm

23 docker start 291e32f5a519

24 docker ps -a # it shows running /stopped / killed containers on your vm

25 docker stop 7eaf8619fcef f64c91c6a96e

26 docker ps -a # it shows running /stopped / killed containers on your vm

27 docker restart 0bf92448f977 7eaf8619fcef

28 docker ps -a # it shows running /stopped / killed containers on your vm

29 docker kill 0bf92448f977

30 docker ps -a # it shows running /stopped / killed containers on your vm

31 docker start 0bf92448f977 f64c91c6a96e

32 docker ps -a # it shows running /stopped / killed containers on your vm

33 docker pull tomcat:8.5.45

34 docker image ls

35 docker run -d tomcat:8.5.45

36 docker ps -a

37 docker rm e0b2d8cc1663

38 docker rm --force e0b2d8cc1663 0bf92448f977 27e9b0579332 291e32f5a519

39 docker ps -a

41 docker ps

42 docker logs f64c91c6a96e

43 clear

44 docker ps

45 docker inspect 01a35e4ad6dc

46 curl 172.17.0.9:8080

49 docker stats

55 docker run -d -P tomcat:8.5.40

56 docker run -d -P tomcat:8.5.45

57 docker run -d -P jenkins:2.60.3-alpine

58 docker ps

59 docker run -d -p 9080:8080 tomcat:8.5.45

62 docker ps

63 docker exec -it 01a35e4ad6dc /bin/bash

64 docker ps

65 docker exec 01a35e4ad6dc java -version

66 java -version

67 docker exec 01a35e4ad6dc pwd

68 pwd

69 docker exec 01a35e4ad6dc vi abct.xt

79 docker image inspect tomcat:8.5.45

80 clear

81 docker pull centos:latest

82 docker pull alpine:latest

83 docker image ls

84 docker run -d centos:latest

85 docker run -d alpine:latest

86 docker ps -a

87 ls -l

88 sleep 10

89 docker run -d centos:latest sleep 30

90 docker ps -a

91 docker run -d tomcat:8.5.45 sleep 20

97 docker rm -f `docker ps -qa`

98 clear

99 docker pull tomcat:8.5.40

100 docker run -d tomcat:8.5.40

101 docker ps

102 docker exec -it 8e09a49cb112 /bin/bash

root@8e09a49cb112:/usr/local/tomcat/webapps/bzcapp# history

1 ls -l

2 cd /usr/local/tomcat/webapps/

3 ls -l

4 mkdir bzcapp

5 cd bzcapp/

6 ls -l

7 vi index.html

8 cat /etc/os-release

9 apt-get update

10 apt-get install -y vim

11 vi index.html

12 exit

103 docker ps

104 docker commit -m "deployed bzcapp" <contid> bzcapp:v1

105 docker image ls

106 docker run -d -P bzcapp:v1

root@master-node:~# cat Dockerfile

FROM tomcat:8.5.45

# FROM is used to set base image in docker file, on top of which we make changes

RUN cd /usr/local/tomcat/webapps ; mkdir bzcapp2

RUN cd /usr/local/tomcat/webapps/bzcapp2 ; touch index.html

RUN apt-get update

RUN apt-get install -y vim

RUN echo "hi there bzcapp2" > /usr/local/tomcat/webapps/bzcapp2/index.html

# RUN is used to exectue commands & make changes during the image build process

112 vi Dockerfile # the file name can be anything

113 cat Dockerfile

114 pwd

115 ls -l

116 docker build --file Dockerfile --tag bzcapp:v2

117 docker build --file Dockerfile --tag bzcapp:v2 /root

118 docker image ls

119 docker run -d -P bzcapp:v2

120 docker ps

FROM ubuntu:18.04

#COPY src(file on vm) dest(path-on-cont)

ADD jdk-8u331-linux-x64.tar.gz /tmp

#RUN tar -xzf /tmp/jdk-8u331-linux-x64.tar.gz -C /opt

#RUN rm /tmp/jdk-8u331-linux-x64.tar.gz

RUN mv /tmp/jdk1.8.0\_331 /opt/java

ENV JAVA\_HOME /opt/java

ENV JAVA\_VERSION 1.8

ADD apache-tomcat-9.0.63.tar.gz /tmp

# ADD copies only the extracted content to cont -- it wont copy the tar

RUN mv /tmp/apache-tomcat-9.0.63 /opt/tomcat

ENV TOMCAT\_HOME /opt/tomcat

COPY sampleapp.war /opt/tomcat/webapps

#CMD ["executable", "arg1", "arg..n"]

#CMD ["executbale"]

CMD ["/opt/tomcat/bin/catalina.sh", "run"]

# CMD is used to set default command to run at runtime (when we run the cont)

EXPOSE 8080

131 wget https://github.com/lerndevops/code/raw/main/jdk-8u331-linux-x64.tar.gz

132 wget https://github.com/lerndevops/code/raw/main/apache-tomcat-9.0.63.tar.gz

133 wget https://github.com/lerndevops/code/raw/main/sampleapp.war

134 ls -l

135 docker pull ubuntu:18.04

136 docker image ls

137 tar -xzf jdk-8u331-linux-x64.tar.gz

138 ls -l

139 rm -r jdk1.8.0\_331/

140 ls -l

141 clear

142 docker image ls

143 docker run -d -P tomcat:8.5.45

144 docker ps

145 docker image inspect tomcat:8.5.45

146 clear

147 vi sampleapdfile

148 ls -l

149 pwd

150 cat sampleapdfile

151 docker build --file sampleapdfile --tag sampleapp:v1

152 docker build --file sampleapdfile --tag sampleapp:v1 /tmp

153 docker build --file sampleapdfile --tag sampleapp:v1 /root

**Prepare vms as below for tomorrow**

**3 vms each with ubuntu 22.04 -- 2cpu/4g ram**

**set the hostname to the vms**

**sudo hostnamectl set-hostname master-node**

**1 vm -- set as -- master-node**

**1vm -- set as -- worker-node001**

**1vm -- set as -- worker-node002**

Install Docker on all VMs

DAY2

186 docker login (to loging to docker hub)

187 docker login ecr.io ( to login to private registry)

188 docker pull alpine:latest

189 docker pull docker.io/library/alpine:latest

190 docker push bzcapp:v1

191 docker build --file sampleapdfile --tag docker.io/lerndevops/bzcapp:v1 /root

192 docker image ls

193 docker build --file sampleapdfile --tag ecr.io/lerndevops/bzcapp:v1 /root

194 docker image ls

195 docker tag ubuntu:18.04 docker.io/lerndevops/bzcapp:ubuntu-18.04

196 docker image ls

197 docker push docker.io/lerndevops/bzcapp:ubuntu-18.04

198 docker push lerndevops/bzcapp:v1

Setup Kube CLUSTER:

<https://github.com/lerndevops/educka/blob/master/1-intall/install-kubernetes-v1.24-ubuntu-debian.md>

246 docker -v

247 cri-dockerd --version

248 kubeadm version -o short

249 kubelet --version

250 kubectl version --short --client

252 sudo kubeadm init --cri-socket unix:///var/run/cri-dockerd.sock --ignore-preflight-errors=all

253 kubectl get nodes

254 sudo mkdir -p $HOME/.kube

255 sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

256 sudo chown $(id -u):$(id -g) $HOME/.kube/config

257 kubectl get nodes

258 kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.24.1/manifests/calico.yaml

259 kubectl get nodes

260 kubectl get pods --all-namespaces

277 kubectl get pods -o wide

278 kubectl run webapppod --image=nginx

279 kubectl run webapppod1 --image=nginx

280 kubectl run dbpod --image=mongo

281 kubectl run dbpod1 --image=mongo0

282 kubectl get pods -o wide

283 kubectl delete pod dbpod1

284 kubectl describe pod webapppod

285 kubectl get pods -o wide

286 kubectl logs webapppod

287 kubectl exec -it webapppod -- /bin/bash

288 kubectl exec webapppod -- nginx -v

294 kubectl get pods -o wide

295 kubectl get pods -o wide --all-namespaces

296 kubectl get namespaces

297 kubectl get pods -o wide --all-namespaces

298 kubectl create namespace teama

299 kubectl create namespace teamb

300 kubectl create namespace app1

301 kubectl create namespace app2

302 kubectl create namespace dev

303 kubectl create namespace qa

304 kubectl get namespaces

305 kubectl get pods -o wide

306 kubectl get pods -o wide --namespace default

307 kubectl get pods -o wide --namespace kube-system

308 kubectl get pods -o wide --namespace teama

309 kubectl get pods -o wide --namespace teamb

310 kubectl run pod1 --image=nginx

311 kubectl get pods -o wide --namespace default

312 kubectl run pod1 --image=nginx --namespace teama

313 kubectl run pod1 --image=nginx --namespace teamb

314 kubectl run pod1 --image=nginx --namespace dev

315 kubectl get pods -o wide --all-namespaces

316 kubectl describe pod pod1 --namespace teama

317 kubectl delete pod pod1 --namespace teama

318 kubectl get pods -o wide --all-namespaces

319 kubectl get pods -o wide -A

320 kubectl run pod1 --image=nginx -n teamb

321 kubectl run pod1 --image=nginx -n teama

325 kubectl api-resources

326 kubectl get pod -o wide --show-labels

327 kubectl get pod -o wide --show-labels -l run

328 kubectl get pod --show-labels -A

329 kubectl get pod --show-labels -A -l run

330 kubectl get pod --show-labels -A -l run=pod1

331 clear

332 vi podx.yaml

333 kubectl apply --filename podx.yaml

334 kubectl get pods -o wide -n default

335 vi multi.yaml

336 kubectl apply -f multi.yaml

337 kubectl get pods -o wide

338 curl 192.168.115.69:80

339 curl 192.168.115.69:8080

340 curl 192.168.115.69:27017

341 kubectl logs multi-cont-pod

342 kubectl logs multi-cont-pod -c cont1

343 kubectl logs multi-cont-pod -c cont2

344 kubectl logs multi-cont-pod -c cont3

345 kubectl exec -it multi-cont-pod -c cont1 -- /bin/bash

346 kubectl exec -it multi-cont-pod -c cont2 -- /bin/bash

347 kubectl exec -it multi-cont-pod -c cont3 -- /bin/bash

in any yaml 99.99% of the time keys are pre defined by the technology (kubernetes)

90% of the time you write your own values

10% chances where both key & value both are pre defined

================================================================================

kubectl run podx --image=httpd --namespace default -l "app=app1,env=dev"

kind: Pod # kubectl api-resources

apiVersion: v1

metadata:

name: multi-cont-pod

namespace: default

labels: # are mandatory # like tags # helps to identify resource as a group # if you donot define a label kube will assing one to pod always

app: app1 # here both key & value are your choice

env: dev

spec:

containers:

- name: abc

image: tomcat

- name: cont2

image: nginx

- name: cont3

image: mongo

**DAY3**

Create EKS Cluster from AWS Console

<https://github.com/lerndevops/eks/blob/main/01-setup/eks-cluster-setup-from-aws-console.pdf>

**once the cluster is active, open the cloud shell on your browser**

run the below command

aws eks --region us-east-1 update-kubeconfig --name naresh-eks

kubectl get nodes

expected output: No resource found

<https://github.com/lerndevops/eks/blob/main/01-setup/Connect-to-EKS-Cluster-Using-AWS-CloudShell.pdf>

**To provide the other IAM user access to your EKS Clsuter follow below steps on Cloud Shell**

kubectl get cm aws-auth -n kube-system -o yaml > aws-auth.yaml

vi aws-auth.yaml

apiVersion: v1

data:

mapRoles: |

- groups:

- system:bootstrappers

- system:nodes

rolearn: arn:aws:iam::390220825855:role/EKS-WORKER-NODE-ROLE

username: system:node:{{EC2PrivateDNSName}}

**# add below section to the file change username & userarn accordingly**

mapUsers: |

- username: eks-admin

userarn: arn:aws:iam::390220825855:user/eks-admin

groups:

- system:masters

**# ends here**

kind: ConfigMap

metadata:

creationTimestamp: "2023-02-08T05:07:03Z"

name: aws-auth

namespace: kube-system

resourceVersion: "6366"

uid: e1291883-ca98-4afd-812e-90a76e1d089e

save the file & close. then apply the file using the kubectl

kubectl apply -f aws-auth.yaml

after applying successfully

goto the vm where you configured the access keys & connect to cluster by running below

**aws eks --region us-east-1 update-kubeconfig --name naresh-eks**

and you can access the cluster

kubectl get nodes – to validate

77 echo "set paste" > $HOME/.vimrc

78 vi rs.yaml

79 kubectl apply -f rs.yaml

80 kubectl get replicaset -o wide

81 kubectl get pods -o wide

82 kubectl scale replicaset myapp --replicas 10

83 kubectl get replicaset -o wide

84 kubectl get pods -o wide

85 kubectl get pods -o wide --show-labels

86 kubectl scale replicaset myapp --replicas 3

87 kubectl get replicaset -o wide

88 kubectl get pods -o wide --show-labels

89 kubectl delete pod myapp-58fgv myapp-75gfd myapp-r8g4w --force

90 kubectl get replicaset -o wide

91 kubectl get pods -o wide --show-labels

92 kubectl delete replicaset myapp

93 kubectl get pods -o wide --show-labels

kind: ReplicaSet

apiVersion: apps/v1

metadata:

name: myapp

namespace: default

#labels: # are optional

spec:

replicas: 4 # number of pods to be created # if we do not mentioned the replicas value it will create 1 by default

template: ## which pods to be created

metadata:

#name: # kube will generate a random unique name for each pod automatically

labels:

app: myapp

spec: # what is inside pod

containers:

- name: cont1

image: lerndevops/samples:pyapp-v1

selector: # are manadtory # helps the replicaset object to identify the pods it created

matchLabels:

app: myapp

108 kubectl get pods -o wide

109 kubectl get rs -o wide

110 kubectl get pods --show-labels

111 kubectl run podx --image=lerndevops/samples:pyapp-v1 -l app=myapp

112 kubectl get pods --show-labels

113 kubectl run podx --image=lerndevops/samples:pyapp-v1 -l app=myapp

114 kubectl delete rs myapp

115 kubectl run podx --image=lerndevops/samples:pyapp-v1 -l app=myapp

116 kubectl apply -f rs.yaml

117 vi ds.yaml

118 kubectl apply -f ds.yaml

119 kubectl get daemonset -o wide

120 kubectl get pods -o wide

121 kubectl scale deamonset myds --replicas 8

kind: DaemonSet

apiVersion: apps/v1

metadata:

name: myds

namespace: default

#labels: # are optional

spec:

template: ## which pods to be created

metadata:

#name: # kube will generate a random unique name for each pod automatically

labels:

app: dsapp

spec: # what is inside pod

containers:

- name: cont1

image: nginx:latest

#- name: cont2

selector: # are manadtory # helps the replicaset object to identify the pods it created

matchLabels:

app: dsapp

kind: Service

apiVersion: v1

metadata:

name: myapp

namespace: default

#labels: # are optional

spec:

type: ClusterIP # it is an internal virtual LB (kube will generate a LB IP dynamically) that can forward the request into pods in backend

selector:

app: myapp

ports:

- name: http

port: 80 # this is the port we will have to use along with ClusterIP/ServiceIP

targetPort: 3000 # is always the port of the applicatioin/service/process inside the cont

137 kubectl get pod -o wide --show-labels

138 kubectl get pod -o wide --show-labels -l app=myapp

139 vi svc.yaml

140 kubectl apply -f svc.yaml

141 kubectl get services -o wide

142 kubectl get pod -o wide --show-labels -l app=myapp

143 kubectl describe service myapp

144 kubectl get nodes -o wide

145 curl 172.31.10.206:3000

146 curl 10.100.160.192:80

147 kubectl get pod -o wide --show-labels -l app=myapp

148 kubectl describe service myapp

149 kubectl scale replicaset myapp --replicas 8

150 kubectl get pod -o wide --show-labels -l app=myapp

151 kubectl describe service myapp

152 kubectl scale replicaset myapp --replicas 2

153 kubectl describe service myapp

kind: Service

apiVersion: v1

metadata:

name: myapp

namespace: default

#labels: # are optional

spec:

type: NodePort # it is going to publish a nodeport/hostport/vmport on every workernode in the cluster & also creates

# an internal virtual LB (kube will generate a LB IP dynamically) that can forward the request into pods in backend and

# maps the nodport to the clsutrip lb

selector:

app: myapp

ports:

- name: http

port: 80 # this is the port we will have to use along with ClusterIP/ServiceIP

targetPort: 3000 # is always the port of the applicatioin/service/process inside the cont

165 kubectl delete service myapp

166 vi np-svc.yaml

167 kubectl apply -f np-svc.yaml

168 kubectl get services -o wide

169 kubectl get pod -o wide --show-labels -l app=myapp

170 kubectl describe service myapp

root@jump-server:~# cat lb-svc.yaml

kind: Service

apiVersion: v1

metadata:

name: myapp

namespace: default

#labels: # are optional

spec:

type: LoadBalancer # kube will create an netwokr lb on AWS cloud & map all the worker nodes of eks cluster to it &

# it is going to publish a nodeport/hostport/vmport on every workernode in the cluster & also creates

# an internal virtual LB (kube will generate a LB IP dynamically) that can forward the request into pods in backend and

# maps the nodport to the clsutrip lb

selector:

app: myapp

ports:

- name: http

port: 80 # this is the port we will have to use along with ClusterIP/ServiceIP

targetPort: 3000 # is always the port of the applicatioin/service/process inside the cont

172 vi lb-svc.yaml

173 kubectl get services -o wide

174 kubectl apply -f lb-svc.yaml

175 kubectl get services

176 kubectl delete service myapp

**DAY4**

400 vi deployment.yaml

401 kubectl apply -f deployment.yaml

402 kubectl get deployment -o wide

403 kubectl get pods -o wide

404 kubectl get deployment -o wide

405 kubectl get repicaset -o wide

406 kubectl get replicaset -o wide

407 kubectl get pods -o wide

408 vi deployment.yaml

409 clear

410 cat deployment.yaml

411 kubectl get services -o wide

412 vi deployment.yaml

413 kubectl apply -f deployment.yaml

414 kubectl rollout status deployment kubeserve

415 vi deployment.yaml

416 kubectl apply -f deployment.yaml

417 kubectl rollout undo deployment kubeserve

418 kubectl rollout history deployment kubeserve

419 cat deployment.yaml

420 kubectl get deployment -o wide

421 kubectl get pods -o wide

422 kubectl get pods

423 cat deployment.yaml

<https://raw.githubusercontent.com/lerndevops/educka/master/3-controllers/deployments/deployment-ex3.yml>

<https://raw.githubusercontent.com/lerndevops/educka/master/3-controllers/jobs/etcd-bakup.yaml>

root@jump-server:~# cat pod.yaml

kind: Pod # kubectl api-resources

apiVersion: v1

metadata:

name: multi-cont-pod

namespace: default

labels: # are mandatory # like tags # helps to identify resource as a group # if you donot define a label kube will assing one to pod always

app: app1 # here both key & value are your choice

env: dev

spec:

#restartPolicy: Always

containers:

- name: abc

image: tomcat

- name: cont2

image: nginx

- name: cont3

image: ubuntu:latest # /bin/bash

396 vi pod.yaml

397 kubectl apply -f pod.yaml

398 kubectl get pods -o wide

399 cat pod.yaml

400 kubectl describe pod multi-cont-pod

401 vi pod.yaml

402 clear

403 kubectl apply -f https://raw.githubusercontent.com/lerndevops/educka/master/3-controllers/jobs/etcd-bakup.yaml

404 kubectl get pods -o wide

405 kubectl get services

406 kubectl get jobs

407 kubectl describe pod backup-etcd-job-pxrf6

411 vi cj.yaml

412 kubectl apply -f cj.yaml

413 vi cj.yaml

414 kubectl apply -f cj.yaml

415 kubectl get cronjob

416 kubectl get jobs -o wide

417 kubectl get pods -o wide

418 kubectl get cronjob

419 kubectl get jobs -o wide

420 kubectl get pods -o wide

421 cat cj.yaml

422 kubectl get pods -o wide

423 cat cj.yaml

kind: CronJob

apiVersion: batch/v1

metadata:

name: backup-etcd-cornjob

namespace: default

spec:

schedule: "\* \* \* \* \*"

jobTemplate:

spec:

backoffLimit: 4

template:

metadata:

labels:

app: etcd1

spec:

restartPolicy: Never

volumes:

- name: hpvol

hostPath:

path: /opt/etcd-backup

containers:

- name: etcd

image: lerndevops/samples:etcdctl

command: ["sh", "-c", 'ETCDCTL\_API=3 etcdctl --endpoints=etcdserver:2379 snapshot save "etcd-snapshot-latest-`date +"%d-%m-%Y-%H-%M"`.db"']

volumeMounts:

- name: hpvol

mountPath: /opt/etcd-backup

428 kubectl explain pod

429 kubectl explain deployment

430 kubectl explain deployment.spec

431 kubectl explain deployment.spec.template

432 kubectl explain deployment.spec.template.spec

433 kubectl explain deployment.spec.template.spec.containers

kind: Deployment

apiVersion: apps/v1

metadata:

name: cont-res-dep

namespace: default

spec:

replicas: 2

#minReadySeconds:

#strategy:

#revisionHistoryLimit: 15

selector:

matchLabels:

app: myapp

template:

metadata:

labels:

app: myapp

spec: # what is inside pod

restartPolicy: Always

containers:

- name: cont1

image: lerndevops/samples:pyapp-v1

env:

- name: JAVA\_HOME

value: /opt/java

- name: DBHOST

value: "10.5.6.7"

- name: DBPORT

value: "1543"

resources:

requests: # the min amount of resources that are guranteed for the cont on the worker node/vm where it run

cpu: 20m # 1cpu core on a vm = 1000 milli cpus == 2% of 1cpu core

memory: 256M

limits: # the max amount of resource that the cont can take from the worker/vm where it runs

cpu: 100m

memory: 1024M

441 vi cont-res-dep.yaml

442 kubectl apply -f cont-res-dep.yaml

443 kubectl get pods

444 kubectl top pods

445 kubectl describe pod cont-res-dep-5679f45598-h2pqj

kind: Deployment

apiVersion: apps/v1

metadata:

name: grace-period-dep

namespace: default

spec:

replicas: 2

#minReadySeconds:

#strategy:

#revisionHistoryLimit: 15

selector:

matchLabels:

app: myapp

template:

metadata:

labels:

app: myapp

spec: # what is inside pod

terminationGracePeriodSeconds: 0 # 0 is forceful delete

restartPolicy: Always

containers:

- name: cont1

image: lerndevops/samples:pyapp-v1

env:

- name: JAVA\_HOME

value: /opt/java

- name: DBHOST

value: "10.5.6.7"

- name: DBPORT

value: "1543"

resources:

requests: # the min amount of resources that are guranteed for the cont on the worker node/vm where it run

cpu: 20m # 1cpu core on a vm = 1000 milli cpus == 2% of 1cpu core

memory: 256M

limits: # the max amount of resource that the cont can take from the worker/vm where it runs

cpu: 100m

memory: 1024M

450 vi grace.yaml

451 kubectl apply -f grace.yaml

452 kubectl get pod -o wide

453 kubectl delete pod grace-period-dep-6d99b6db5f-fp8ml

454 kubectl delete pod kubeserve-8456b48664-j5qzm

456 kubectl delete deployment --all

457 kubectl get pods -o wide

458 kubectl delete jobs --all

459 kubectl delete cronjob --all

460 kubectl delete pods --all

kind: Deployment

apiVersion: apps/v1

metadata:

name: edvol-dep

namespace: default

spec:

replicas: 2

#minReadySeconds:

#strategy:

#revisionHistoryLimit: 15

selector:

matchLabels:

app: myapp

template:

metadata:

labels:

app: myapp

spec: # what is inside pod

volumes:

- name: edvol # name can be anything

emptyDir: {}

terminationGracePeriodSeconds: 0 # 0 is forceful delete

restartPolicy: Always

containers:

- name: cont1

image: tomcat:latest

#env:

#resources:

volumeMounts:

- name: edvol

mountPath: /usr/local/tomcat/logs

- name: cont2

image: nginx:latest

volumeMounts:

- name: edvol

mountPath: /usr/share/nginx/html

464 vi edvol-dep.yaml

465 clear

466 kubectl apply -f edvol-dep.yaml

467 kubectl get pods -o wide

468 kubectl exec -it edvol-dep-84cbc7667f-jhcx6 -c cont1 -- /bin/bash

469 kubectl exec -it edvol-dep-84cbc7667f-jhcx6 -c cont2 -- /bin/bash

470 kubectl exec -it edvol-dep-84cbc7667f-s2rxj -c cont1 -- /bin/bash

471 kubectl delete deployment --all

kind: Deployment

apiVersion: apps/v1

metadata:

name: hpvol-dep

namespace: default

spec:

replicas: 2

#minReadySeconds:

#strategy:

#revisionHistoryLimit: 15

selector:

matchLabels:

app: myapp

template:

metadata:

labels:

app: myapp

spec: # what is inside pod

volumes:

- name: edvol # name can be anything

emptyDir: {}

- name: hpvol

hostPath:

path: /appconfig ## this is the path on the worker vm/node where the pod will run

terminationGracePeriodSeconds: 0 # 0 is forceful delete

restartPolicy: Always

containers:

- name: cont1

image: tomcat:latest

#env:

#resources:

volumeMounts:

- name: edvol

mountPath: /usr/local/tomcat/logs

- name: cont2

image: nginx:latest

volumeMounts:

- name: edvol

mountPath: /usr/share/nginx/html

- name: hpvol

mountPath: /var/log/nginx # this is the path inside the cont always

475 vi hp-vol-dep.yaml

476 clear

477 kubectl apply -f hp-vol-dep.yaml

478 kubectl get pods -o wide

479 kubectl exec -it hpvol-dep-6d9fc94b77-z95xt -c cont2 -- /bin/bash

480 kubectl delete deployment --all

481 kubectl get pods

kind: PersistentVolume

apiVersion: v1

metadata:

name: hp-pv

spec:

hostPath:

path: /appconfig

capacity:

storage: 2Gi

accessModes:

- ReadWriteOnce # only one node can read/write the data -- ReadOnlyMany / ReadWriteMany

persistentVolumeReclaimPolicy: Retain # Delete # what to do with the data inside the volume when the pv object the deleted by kube

---

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

name: myapp-pvc

spec:

volumeName: hp-pv

resources:

requests:

storage: 2Gi

accessModes:

- ReadWriteOnce

---

kind: Deployment

apiVersion: apps/v1

metadata:

name: pv-hpvol-dep

namespace: default

spec:

replicas: 2

#minReadySeconds:

#strategy:

#revisionHistoryLimit: 15

selector:

matchLabels:

app: myapp

template:

metadata:

labels:

app: myapp

spec: # what is inside pod

volumes:

#- name: edvol # name can be anything

# emptyDir: {}

#- name: hpvol

# hostPath:

# path: /appconfig ## this is the path on the worker vm/node where the pod will run

- name: pvcvol

persistentVolumeClaim:

claimName: myapp-pvc

terminationGracePeriodSeconds: 0 # 0 is forceful delete

restartPolicy: Always

containers:

- name: cont1

image: tomcat:latest

#env:

#resources:

volumeMounts:

- name: pvcvol

mountPath: /usr/local/tomcat/logs

486 vi pv.yaml

487 kubectl apply -f pv.yaml

488 kubectl get pv

489 kubectl describe pv hp-pv

490 vi pvc.yaml

491 kubectl apply -f pvc.yaml

492 kubectl get pvc

493 kubectl get sc

494 kubectl delete sc gp2

495 kubectl get pvc

498 kubectl get pvc

499 kubectl get pv

500 vi pv-hpvol-dep.yaml

501 kubectl apply -v pv-hpvol-dep.yaml

502 kubectl apply -f pv-hpvol-dep.yaml

503 kubectl get pods -o wide

504 kubectl exec -it pv-hpvol-dep-69d5f99774-r9gnm -- /bin/bash

kind: PersistentVolume

apiVersion: v1

metadata:

name: nfs-pv

spec:

nfs:

server: 54.242.169.89

path: /mnt/appdata

#hostPath:

#path: /appconfig

capacity:

storage: 2Gi

accessModes:

- ReadWriteMany # only one node can read/write the data -- ReadOnlyMany / ReadWriteMany

persistentVolumeReclaimPolicy: Retain # Delete # what to do with the data inside the volume when the pv object the deleted by kube

---

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

name: myapp-nfs-pvc

spec:

volumeName: nfs-pv

resources:

requests:

storage: 2Gi

accessModes:

- ReadWriteMany

---

kind: Deployment

apiVersion: apps/v1

metadata:

name: pv-nfsvol-dep

namespace: default

spec:

replicas: 2

#minReadySeconds:

#strategy:

#revisionHistoryLimit: 15

selector:

matchLabels:

app: myapp

template:

metadata:

labels:

app: myapp

spec: # what is inside pod

volumes:

#- name: edvol # name can be anything

# emptyDir: {}

#- name: hpvol

# hostPath:

# path: /appconfig ## this is the path on the worker vm/node where the pod will run

- name: nfsvol

persistentVolumeClaim:

claimName: myapp-nfs-pvc

terminationGracePeriodSeconds: 0 # 0 is forceful delete

restartPolicy: Always

containers:

- name: cont1

image: nginx:latest

#env:

#resources:

volumeMounts:

- name: nfsvol

mountPath: /usr/share/nginx/html

549 vi nfs-dep.yaml

550 kubectl apply -f nfs-dep.yaml

551 kubectl get pv,pvc

552 kubectl get pods

553 kubectl get pods -o wide

Created below files on NFS Server, you all should be able to see them when you login to the pod

554 touch /mnt/appdata/index.html

556 touch /mnt/appdata/naresh.txt

559 kubectl exec -it pv-nfsvol-dep-84698757fd-cccw4 -- /bin/bash

574 kubectl get pods -o wide

575 kubectl delete deployment --all

576 kubectl get pods

577 kubectl get pvc

578 kubectl delete pvc --all

579 kubectl get pvc

580 kubectl get pv

581 kubectl delete pv --all

582 kubectl get pods

589 kubectl create configmap devdbconfig --from-literal=DBHOST="10.5.6.7" --from-literal=DBPORT="1521" --from-literal=DBNAME=devdb

591 kubectl create configmap filecm --from-file=/root/ds.yaml --from-file=/etc/resolv.conf

592 kubectl get configmap

593 kubectl get cm

594 kubectl describe cm filecm

595 clear

596 history

597 kubectl get cm

598 clear

599 vi cm-dep1.yaml

600 kubectl apply -f cm-dep1.yaml

601 kubectl get pods -o wide

602 kubectl exec -it cm-as-vol-data-dep-b4bc5bbdf-9hgxk -- /bin/bash

cd /dbconfig

ls -l ; exit

603 kubectl get cm

604 clear

605 vi cm-dep2.yaml

606 clear

607 kubectl apply -f cm-dep2.yaml

608 kubectl get pods

609 kubectl describe cm devdbconfig

610 kubectl exec -it cm-as-env-data-dep-b8c89754-nxhmb -- /bin/bash

kind: ConfigMap

apiVersion: v1

metadata:

name: qadbconfig

data:

DBHOST: "4.5.6.7"

DBPORT: "1531"

DBNAME: "qadb"

kind: Deployment

apiVersion: apps/v1

metadata:

name: cm-as-vol-data-dep

namespace: default

spec:

replicas: 2

#minReadySeconds:

#strategy:

#revisionHistoryLimit: 15

selector:

matchLabels:

app: myapp

template:

metadata:

labels:

app: myapp

spec: # what is inside pod

volumes:

#- name: edvol # name can be anything

# emptyDir: {}

#- name: hpvol

# hostPath:

# path: /appconfig ## this is the path on the worker vm/node where the pod will run

- name: cmvol

configMap:

name: filecm

terminationGracePeriodSeconds: 0 # 0 is forceful delete

restartPolicy: Always

containers:

- name: cont1

image: nginx:latest

#env:

#resources:

volumeMounts:

- name: cmvol

mountPath: /dbconfig

kind: Deployment

apiVersion: apps/v1

metadata:

name: cm-as-env-data-dep

namespace: default

spec:

replicas: 2

#minReadySeconds:

#strategy:

#revisionHistoryLimit: 15

selector:

matchLabels:

app: myapp

template:

metadata:

labels:

app: myapp

spec: # what is inside pod

volumes:

#- name: edvol # name can be anything

# emptyDir: {}

#- name: hpvol

# hostPath:

# path: /appconfig ## this is the path on the worker vm/node where the pod will run

#- name: cmvol

# configMap:

# name: filecm

terminationGracePeriodSeconds: 0 # 0 is forceful delete

restartPolicy: Always

containers:

- name: cont1

image: nginx:latest

envFrom:

- configMapRef:

name: devdbconfig

#env:

#resources:

#volumeMounts:

## mountPath: /dbconfig

**DAY5**

kind: Deployment

apiVersion: apps/v1

metadata:

name: sec-as-env-data-dep

namespace: default

spec:

replicas: 2

#minReadySeconds:

#strategy:

#revisionHistoryLimit: 15

selector:

matchLabels:

app: myapp

template:

metadata:

labels:

app: myapp

spec: # what is inside pod

volumes:

#- name: edvol # name can be anything

# emptyDir: {}

#- name: hpvol

# hostPath:

# path: /appconfig ## this is the path on the worker vm/node where the pod will run

#- name: cmvol

# configMap:

# name: filecm

terminationGracePeriodSeconds: 0 # 0 is forceful delete

restartPolicy: Always

containers:

- name: cont1

image: nginx:latest

envFrom:

- secretRef:

name: devdbcred

#env:

#resources:

#volumeMounts:

## mountPath: /dbconfig

---

kind: Deployment

apiVersion: apps/v1

metadata:

name: sec-as-vol-data-dep

namespace: default

spec:

replicas: 2

#minReadySeconds:

#strategy:

#revisionHistoryLimit: 15

selector:

matchLabels:

app: myapp

template:

metadata:

labels:

app: myapp

spec: # what is inside pod

volumes:

#- name: edvol # name can be anything

# emptyDir: {}

#- name: hpvol

# hostPath:

# path: /appconfig ## this is the path on the worker vm/node where the pod will run

- name: secvol

secret:

secretName: devdbcred

terminationGracePeriodSeconds: 0 # 0 is forceful delete

restartPolicy: Always

containers:

- name: cont1

image: nginx:latest

#envFrom:

# - secretRef:

# name: devdbcred

#env:

#resources:

volumeMounts:

- name: secvol

mountPath: /dbcred

740 base64 --help

741 echo -n "nareshdbadminuser"

742 echo -n "nareshdbadminuser" | base64

743 echo -n "alkdjf4lksdj3;lkadjt" | base64

744 ls

745 cat cj.yaml

746 cat cj.yaml | base64

747 cat cj.yaml | base64 -d "\n"

748 cat cj.yaml | base64 | tr -d "\n"

749 clear

750 vi secret.yaml

751 kubectl apply -f secret.yaml

752 kubectl get secrets

753 kubectl describe secret devdbcred

754 kubectl describe cm filecm

759 kubectl apply -f sec-dep.yaml

760 kubectl get pods -o wide

761 kubectl exec sec-as-vol-data-dep-7fdd799f88-k865b -- ls -l /dbcred

762 kubectl exec sec-as-vol-data-dep-7fdd799f88-k865b -- cat /dbcred/cj.yaml

763 kubectl exec sec-as-vol-data-dep-7fdd799f88-k865b -- cat /dbcred/password

764 kubectl exec sec-as-vol-data-dep-7fdd799f88-k865b -- cat /dbcred/password ;echo

765 kubectl exec sec-as-env-data-dep-77594668cd-2fxbp -- env

Service networking

<https://github.com/lerndevops/educka/blob/master/6-networking/core-dns/dns-test.md>

Deploy Calico for Network policies

839 kubectl apply -f https://raw.githubusercontent.com/aws/amazon-vpc-cni-k8s/master/config/master/calico-operator.yaml

840 kubectl apply -f https://raw.githubusercontent.com/aws/amazon-vpc-cni-k8s/master/config/master/calico-crs.yaml

Deploy app / db in default namespace

kubectl apply -f <https://raw.githubusercontent.com/lerndevops/educka/master/examples/springboot-mongo-app.yml>

Deploy another app with in the same namespace on in any other namespace

kubectl create namespace test

kubectl apply -f <https://raw.githubusercontent.com/lerndevops/educka/master/6-networking/core-dns/app-dnstest.yml> -n test

Deploy a network policy that deny the access to the db from everywhere in cluster

Vi mongodb-netpol.yaml

kind: NetworkPolicy

apiVersion: networking.k8s.io/v1

metadata:

name: deny-mongodb-ingress-from-all

namespace: default

spec:

podSelector: # this network policy will be applied on the pods witb label

matchLabels:

app: mongodb

policyTypes:

- Ingress

#ingress:

kubectl apply -f mongodb-netpol.yaml

Now test the connection to the db from an app in all namespaces

kubectl exec -it springboot-app-5699498d67-cx4rt -n default -- /bin/sh

curl dbpodip:27017

kubectl exec -it nettest-5dd944d4b-rfhz8 -n test -- /bin/bash

curl dbpodip:27017

Deploy a network policy that allows the access to the db from springboot app in cluster

---

kind: NetworkPolicy

apiVersion: networking.k8s.io/v1

metadata:

name: allow-mongodb-ingress-from-springboot-app-only

namespace: default

spec:

podSelector: # this network policy will be applied on the pods witb label

matchLabels:

app: mongodb

policyTypes:

- Ingress

ingress:

- from:

- podSelector:

matchLabels:

app: myapp

Now test the connection again to the db from an app in all namespaces

kubectl exec -it springboot-app-5699498d67-cx4rt -n default -- /bin/sh – this should respond

curl dbpodip:27017

kubectl exec -it nettest-5dd944d4b-rfhz8 -n test -- /bin/bash – this should stuck & no response

curl dbpodip:27017

865 kubectl delete all --all

866 clear

867 kubectl apply -f https://raw.githubusercontent.com/lerndevops/educka/master/6-networking/ingress/deploy-app.yml

868 kubectl get pods -o wide

869 kubectl get services -o wide

870 kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.3.0/deploy/static/provider/baremetal/deploy.yaml

871 kubectl get pods -n ingress-nginx

872 kubectl get services -n ingress-nginx

873 kubectl get services

874 kubectl apply -f https://raw.githubusercontent.com/lerndevops/educka/master/6-networking/ingress/ingress-rule1.yml

875 kubectl get ingress

876 kubectl describe ingress ingress-rule1

877 kubectl get pods -o wide

890 aws sts get-caller-identity

891 aws eks update-kubeconfig --name naresh-eks

892 kubectl get pods

893 mv config /tmp

894 kubectl get nodes

895 kubectl --kubeconfig=/tmp/config get pods

896 docker ps

897 mv /tmp/config .

898 ls -l

899 vi config

900 aws eks --cluster-name naresh-eks --region us-east-1 get-token

901 aws eks get-token --cluster-name naresh-eks --region us-east-1

902 echo "aHR0cHM6Ly9zdHMudXMtZWFzdC0xLmFtYXpvbmF3cy5jb20vP0FjdGlvbj1HZXRDYWxsZXJJZGVudGl0eSZWZXJzaW9uPTIwMTEtMDYtMTUmWC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBVlZXWFpJVDc3QlVPSVQ0VSUyRjIwMjMwMjEwJTJGdXMtZWFzdC0xJTJGc3RzJTJGYXdzNF9yZXF1ZXN0JlgtQW16LURhdGU9MjAyMzAyMTBUMTAyMjMxWiZYLUFtei1FeHBpcmVzPTYwJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCUzQngtazhzLWF3cy1pZCZYLUFtei1TaWduYXR1cmU9YzVkZDY5YzEyMWYxOTZiN2EyMTAwZTg2ZWFlMzQ0MTA1Mjc2MTlkMGYxOTE1ZjI0NmYyZjhkMGQxNWUxMzI2YQ" | base64 -d

903 aws sts get-caller-identity

904 cd

905 kubectl get ns

906 kubectl create role readonly --verb=get,list,watch --resource=pods,deployments --namespace=teama

907 kubectl create role admin --verb="\*" --resource="\*.\*" --namespace=teama

908 kubectl get roles -n teama

909 kubectl create rolebinding readonly-rb --namespace=teama --role=readonly --user=user1

910 kubectl get configmap aws-auth -o yaml

911 kubectl get configmap aws-auth -o yaml -n kube-system

912 kubectl get configmap aws-auth -o yaml -n kube-system > aws-auth.yaml

913 vi aws-auth.yaml

914 kubectl apply -f aws-auth.yaml

915 kubectl get configmap aws-auth -o yaml -n kube-system

916 vi aws-auth.yaml

917 clear

918 history

919 kubectl create rolebinding admin-rb --namespace=teama --role=admin --user=user1

920 vi aws-auth.yaml

921 kubectl apply -f aws-auth.yaml

922 vi aws-auth.yaml

923 kubectl apply -f aws-auth.yaml

924 history

925 kubectl delete rolebinding admin-rb readonly-rb

926 kubectl delete rolebinding admin-rb readonly-rb -n teama

927 kubectl create rolebinding admin-rb --namespace=teama --role=admin --group=devgroup

928 vi aws-auth.yaml

929 clear

930 kubectl apply -f aws-auth.yaml

931 vi aws-auth.yaml

932 history

933 kubectl api-resources

934 kubectl api-resources | grep true

935 kubectl api-resources | grep fasle

936 kubectl api-resources | grep false

937 kubectl create clusterrole readonly-cr --verb=get,list,watch --resource=nodes,namespaces,pods

938 kubectl create clusterrolebinding readonly-crb --user=user1 --clusterrole=readonly-cr

939 vi aws-auth.yaml

940 kubectl apply -f aws-auth.yaml

941\* kubectl get clusterroes

942 kubectl get clusterroles

943 kubectl describe clusterrole cluster-admin

944 kubectl describe clusterrole view

945 kubectl describe clusterrole admin

946 kubectl get clusterrolebindings

947 kubectl describe clusterrolebinding cluster-admin

<https://github.com/lerndevops/educka/blob/master/7-security/serviceaccouts/README.md>

<https://github.com/lerndevops/educka/blob/master/7-security/rbac/rbac-normalUser.md>

950 kubectl apply -f https://raw.githubusercontent.com/lerndevops/educka/master/dashboard/dashboard-insecure-v2.4.0.yml

951 kubectl get all -n kubernetes-dashboard

952 clear

953 kubectl get services

954 kubectl described svc kubernetes

955 kubectl describe svc kubernetes

956 curl https://10.100.0.1:443/version

957 history

958 kubectl create clusterrolebinding kube-dash --clusterrole=view --serviceaccount=kubernetes-dashboard:kubernetes-dashboard

959 kubectl top nodes

960 kubectl apply -f https://raw.githubusercontent.com/lerndevops/educka/master/9-monitoring/metrics-server/metrics-server-v0.5.yml