sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key

amazon-linux-extras install epel

amazon-linux-extras install java-openjdk11

yum install jenkins

service jenkins start

service jenkins status

Open Jenkins publiIp:8080

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

Install Git On Jenkins Instace

install GitHub Plugin on Jenkins GUI

Configure Git On Jenkins GUI

yum install git

git --version

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

In Jenkins we we crete FreshStyle --> we give name PullFromGitHub --> we provide git Repositories URl.

In Jenkins , When we build pipline

/var/lib/jenkins/workspace/ --> in that place our code is present.

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

Integrate Maven with Jenkins

1) SetUp Maven On Jenkins server

2) SetUp Enviroment Variable

--> JAVA\_HOME, M3,M3 ,M2\_HOME

3) Install Maven Plugin

4) Configure Mavenand Java

i)

Binary tar.gz archive --->

wget https://dlcdn.apache.org/maven/maven-3/3.9.8/binaries/apache-maven-3.9.8-bin.tar.gz

ii) tar -xvzf apache-maven-3.9.8-bin.tar.gz

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

Moving this Dirt--->

mv apache-maven-3.9.8 maven

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

for serching Command

--->

for find

find / -name jvm

cd /usr/lib/jvm

ls

[root@ip-172-31-23-7 jvm]# find / -name java-11\*

/usr/lib/jvm/java-11-openjdk-11.0.23.0.9-2.amzn2.0.1.x86\_64

/usr/share/doc/java-11-openjdk-11.0.23.0.9-2.amzn2.0.1.x86\_64

[root@ip-172-31-23-7 ~]# ll -la

vi .bash\_profile

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

# .bash\_profile

# Get the aliases and functions

if [ -f ~/.bashrc ]; then

. ~/.bashrc

fi

M2\_HOME=/opt/maven

M2=/opt/maven/bin

JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-11.0.23.0.9-2.amzn2.0.1.x86\_64

# User specific environment and startup programs

PATH=$PATH:$HOME/bin:$JAVA\_HOME:$M2\_HOME:$M2

export PATH

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

echo $PATH

now you can check version --> mvn -v

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

In Jenkins , you can download Plugin

-->

Maven IntegrationVersion

--> In Jenkins goto Tool

1) In JDK add

Name--->java-11

JAVA\_HOME

/usr/lib/jvm/java-11-openjdk-11.0.23.0.9-2.amzn2.0.1.x86\_64

2) In Add Maven

name-->maven

MAVEN\_HOME

/opt/maven

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

Build a java project Using Jenkins

In New Items ,

Create Maven Project In Jenkins

FirstMavenProject

1) Give Git Repositories

2) In Build --> In Goals and options --> clen install

------------------------------------Section 3

Deploy Artifacts on a Tomacat Server

we create new EC2 - Tomcat-server

In that we download JAVA

amazon-linux-extras install java-openjdk11

install tomcat

1)

wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.91/bin/apache-tomcat-9.0.91.tar.gz

2) tar -xvzf apache-tomcat-9.0.91.tar.gz

3)mv apache-tomcat-9.0.91 tomcat

goto bin and

cd /opt/tomcat/bin

start tomcat --> ./startup.sh

Got Broswer and 52.91.169.122:8080

when you click on Manage App ---> 403 Access Denied

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

got to tomcat Folder

and find context.xml file

[root@ip-172-31-85-62 tomcat]# find / -name context.xml

in that we need to update this file

1) vi /opt/tomcat/webapps/host-manager/META-INF/context.xml

we need to comment out this line

---> <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"

allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->

2)vi /opt/tomcat/webapps/manager/META-INF/context.xml

---> <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"

allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->

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

after restart server

./shutdown.sh

./startup.sh

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

open tomcat-users.xml file

vi /opt/tomcat/conf/tomcat-users.xml

In that file add

last

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

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>

<user username="deployer" password="deployer" roles="manager-script"/>

<user username="tomcat" password="s3cret" roles="manager-gui"/>

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

ln -s /opt/tomcat/bin/startup.sh /usr/local/bin/tomcatup

---> we change startup too tomcatup

ln -s /opt/tomcat/bin/shutdown.sh /usr/local/bin/tomcatdown

--> we change to shotdown to tomcatdown

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

then Login Tomcat ------>

username="tomcat" password="s3cret"

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

Integrate Tomcat with Jenkins

1) Install "Deploy to container

2) Configure tomcat server with Creatials

In Jenkins --> download plugin --> Deploy to container

In Jenkins -->

Dashboard/Manage Jenkins/Credentials/System/Global credentials (unrestricted)

add username and password --> deployer

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

Create New Item --> BuildtoDeploy --> mavenproject

provide git Repositories

provide Build clean install

provide Post-build Actions -->

1) WAR/EAR files --> \*\*/\*.war

2) Containers --> Tomcat 9.x Remote --> provide credentials

after build goto tomcat ther is your code in deploy ...

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

22

SetUp Docker Host

1) SetUp a Linux EC2 Instace

2) Install Docker

3) start docker service

4) Basic docker commands

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

yum install docker -y

service docker start

service docker statup

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

Create tomcat Container

--> search tomcat image on dockerHub

--> change hostname

vi /etc/hostname

init 6 --> restart server

after that you start the service

1)service docker start

2)service docker statup

3)we pull our image in docker by using this commands

--> docker pull tomcat

4) docker images

5) run docker

--> docker run -d --name tomcat-container -p 8081:8080 tomcat

6) docker ps -a

afrer open public:8081 in Browser we get error 404 Not Found

4) you want to stope the container

--> docker stop tomcat-container

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

Fixing Tomcat container issue.

Login to Docker tomcat-container

1) docker exec -it tomcat-container /bin/bash

after that we jump into docker container.

2) do ls --> then goto webapps.dist

3) copy all the content from webapps.dist to webapps

---> cp -R \* ../webapps/

4) then refresh the Browser --> publicIp:8081

5) exit from container --> exit

------>

whenever we create new container this issue we goes on for that reason we create dockerFile

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

Create a first Docker file

1)From : To Pull the Base image

2)Run: To execute commnads

3)CMD :To provid defaukt for an executing container

4)ENTRYPOINT

5)WORKDIR

6)Add

7)EXPOSE

8)ENV

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

Install tomcat on Centos

1) Pull centos from dockerHub

2) Install java

3) Create /opt/tomcat directory

4) Change work directory to /opt/tomcat

5) Download tomcat packages

6) Extract tar.gz file

7) Rename to tomcat directory

8) tell to docker that it runs o port 8080

9) Start tomcat services

this all staps we do in DockerFile

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

got to root --> create dockerFile

1) vi DockerFile

-->

FROM centos:latest

RUN sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-\*

RUN sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-\*

RUN yum update -y

RUN yum install java -y

RUN mkdir /opt/tomcat

WORKDIR /opt/tomcat

ADD https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.91/bin/apache-tomcat-9.0.91.tar.gz .

RUN tar -xvzf apache-tomcat-9.0.91.tar.gz

RUN mv apache-tomcat-9.0.89/\* /opt/tomcat

EXPOSE 8080

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

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

1) [root@docker-host ~]# docker build -t mytomcat .

2) docker images

3) docker run -d --name mytomcat-server -p 8082:8080 mytomcat

3) docker ps -a

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

create a customized DockerFile for tomcat

we remove first Dokerfile

and create new DockerFile

>>

vi Dokerfile

FROM tomcat:latest

RUN cp -R /usr/local/tomcat/webapps.dist/\* /usr/local/tomcat/webapps

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

docker run -d --name demotomcat-server2 -p 8083:8080 mytomcat2

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

Integrate Docker with Jenkins

1)Create a dockeradmin user

2)Install "publish Over SSh" plugin

3)Add Dockerhost to Jenkins "configure systems"

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

>> useradd dockeradmin

passwd dockeradmin

Aksh@y06

[root@docker-host ~]# id dockeradmin

[root@docker-host ~]# usermod -aG docker dockeradmin

---->

EC2 instance doesn't allow password base athentication

for that reason we update code in sshd\_config file

>> vi /etc/ssh/sshd\_config

and also provid SSH Key

>> service sshd reload

and login different account

----> dockeradmin@IP

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

In Jenkins , add plugin

--> Publish Over SSH

then next

>> Manage Jenkins/System

Inside -> systems ,

Inside -> Publish over SSH

add -> SSH Servers

(

Name

dockerhost

Hostname

172.31.83.43 --> Docker privateId

Username

dockeradmin

click on --> Advance.

and add password here Passphrase / Password

)

In docker-server ,

>> sudo su dockeradmin

>> ssh-keygen

it will show path where we crete key

--> cd /home/dockeradmin/.ssh

--------> No need to use this one bez we use password base authentication.

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

Deploy on a Container

In jenkins,

create new item,

>>BuildAndDeployeOnDockerContainer

choose -> copy from

>>BuildAndDeploye

and after that

Post-build Actions-> choose

Send build artifacts over SSH

(

Source files

webapp/target/\*.war

Remove prefix

webapp/target

Remote directory

/home/dockeradmin

)

after build the pipline.

and you can check in server

[dockeradmin@docker-host ~]$ ll

there is file present (dockeradmin)

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

Update Tomcat dockerfile to automate deployment process

1) create folder in /opt

>> mkdir docker

i am giving ounership of this folder ad dockeradmin

>> chown -R dockeradmin:dockeradmin docker

Moving DockerFile inside docker Folder

>> mv Dockerfile /opt/docker/

i also provide owner shi for this DockerFile

>> chown -R dockeradmin:dockeradmin /opt/docker

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

In Jenkins , Now you goto that BuildAndDeployeOnDockerContainer and

in Remote directory you going to change PATH.

--->/home/dockeradmin Instat of you write.

---> //opt//docker

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

Now you can Check ,

[root@docker-host docker]# ll

.war file is present

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

We add some line in DockerFile

[root@docker-host docker]# vi Dockerfile

-->

FROM tomcat:latest

RUN cp -R /usr/local/tomcat/webapps.dist/\* /usr/local/tomcat/webapps

COPY ./\*.war /usr/local/tomcat/webapps

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

then we build this file

[root@docker-host docker]# docker build -t tomcat:v1 .

check Images,

>> docker images

then you run this image.

>> docker run -d --name tomcatv1 -p 8085:8080 tomcat:v1

you go to browser

--> http://ip:8085/webapp/

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

Automate Build and Deployment on Docker container

and

Jenkins Job to automate CI/CD to deploy application on docker container.

n Jenkins , Now you can Update BuildAndDeployeOnDockerContainer In

SSH Server

--> you add Exec command

-->

cd /opt/docker

docker build -t regapp:v1 .;

docker stop registerapp;

docker rm registerapp;

docker run -d --name registerapp -p 8086:8080 regapp:v1

In Docker-server,

i stop all container

>> docker stop c895b148b728 26f8b09e3691

or

>> docker container prune

--> stope all docker images

>> docker image prune -a

then Build ...

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

Ansible installation.

1) SetUp EC2 instance

2) setup hostname

3) Crate ansadmin file

4) Add user to sudoers file

5) Generate ssh keys

6) Enable password based login

7) inatll ansible

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

Create Ec2 ansible-server

1) create user

>> useradd ansadmin

>> passwd ansadmin

Aksh@y06

we going to add this command in visudo file

[root@ansible-server ~]# visudo

we add this line

-->

## Same thing without a password

# %wheel ALL=(ALL) NOPASSWD: ALL

ansadmin ALL=(ALL) NOPASSWD: ALL

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

after that

open

[root@ansible-server ~]# vi /etc/ssh/sshd\_config

inside you update this command

# To disable tunneled clear text passwords, change to no here!

PasswordAuthentication yes

#PermitEmptyPasswords no

#PasswordAuthentication no

>> service sshd reload

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

after Login

>> sudo su ansadmin

>> ssh-keygen

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

install ansible

---

1)>> amazon-linux-extras install epel

2)>> yum install ansible

3)>> python --version

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

Manage DockerHost with Ansible 34.

On Docker Host On Ansible Node

1)Crate ansadmin add to hosts file

2)add ansadmin to sudoers file Copy ssh keys

3)Enable password based login test the connection

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

Same think you do in docker-server

--->

>> useradd ansadmin

>> passwd ansadmin

Aksh@y06

we going to add this command in visudo file

[root@ansible-server ~]# visudo

we add this line

-->

## Same thing without a password

# %wheel ALL=(ALL) NOPASSWD: ALL

ansadmin ALL=(ALL) NOPASSWD: ALL

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

thins command check --> pass is yes or #

[root@docker-host ~]# grep Password /etc/ssh/sshd\_config

and copy Ip addresh of docker host

>> [root@docker-host ~]# ifconfig

(

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001

inet 172.31.83.43

)

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

Goto Ansible-server

[root@ansible-server ~]# vi /etc/ansible/hosts

--->

172.31.83.43

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

>> sudo su ansadmin

>> ssh-copy-id dockerPrivateIp

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

Inside you will see authorized\_keys

[root@ansible-server ~]# cd .ssh

[root@ansible-server .ssh]# ls

authorized\_keys id\_rsa id\_rsa.pub

-->

authorized\_keys this key we copy to our systems..

so let's check our system is connected or Not

>> sudo su ansadmin

>> cd home

[ansadmin@ansible-server home]$

--> ansible all -m ping

>> ansible all -m command -a uptime

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

Integrate Ansible with Jenkins

In Jenkins,

Inside -> systems ,

Inside -> Publish over SSH

add -> SSH Servers

(

Name

ansible-server

Hostname

172.31.83.43 --> ansible privateId

Username

ansadmin

click on --> Advance.

and add password here Passphrase / Password

)

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

In ansible-server , create folder docker

[ansadmin@ansible-server opt]$ sudo mkdir docker

[ansadmin@ansible-server opt]$ sudo chown ansadmin:ansadmin docker

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

In Jenkins ,

create new job

--> Copy\_Artifacts\_onto\_Ansible

and use copy artifacts use -->

BuildAndDeployeOnDockerContainer

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

and inside SSH Publishers

and name use --> ansible-server

we use this dirctory(//opt//docker) in that job that why we create this folder in ansible-server..

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

Bild an image and create container On Ansible

1) we download docker in Ansible-server

[ansadmin@ansible-server ~]$ sudo yum install docker -y

2)start docker

-->

[ansadmin@ansible-server docker]$ sudo service docker start

[ansadmin@ansible-server docker]$ sudo service docker status

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

we create new docker file -->

[ansadmin@ansible-server docker]$ vi Dockerfile

-->

FROM tomcat:latest

RUN cp -R /usr/local/tomcat/webapps.dist/\* /usr/local/tomcat/webapps

COPY ./\*.war /usr/local/tomcat/webapps

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

we give the permition to our .sock file or we use sudo befor each command

>> sudo chmod 777 /var/run/docker.sock

then we build -->

>> docker build -t regapp:v1 .

check Images,

>> docker images

then you run this image.

>> docker run -d --name regapp-server -p 8081:8080 regapp:v1

you go to browser

--> http://ansibleip:8081/webapp/

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

Ansible playbook to create images and conntainer

we want run ansible command in ansible-server for that reasone we need to add ansible-server ip in hosts

--> we alreday add docker Ip for that reason we need to add.

ansadmin@ansible-server docker]$ -->

>>ifconfig

>> vi /etc/ansible/hosts

[dockerhost]

172.31.83.43

[ansible]

172.31.95.232

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

after that we perform 2 commands in ansible-server

>> ssh-copy-id 172.31.95.232

>> ssh-copy-id localhost

---> now going to check is succefuly connect by using this command.

>> ansible all -a uptime

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

I am going to create ansible playbook file

[ansadmin@ansible-server docker]$ vi regapp.yml

---

- hosts: ansible

tasks:

- name: create docker image

command: docker build -t regapp:latest .

args:

chdir: /opt/docker

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

first we check this file

>> ansible-playbook regapp.yml --check

--> then you excute

>> ansible-playbook regapp.yml

then check docker images

>> docker images

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

Copy image on to dockerHub

--> first we create dockerHub account

--> after that we came to ansible-server

>> [ansadmin@ansible-server docker]$ docker login

username : foodorderingapp12

password : Aksh@y061999

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

then we push over images in docker Hub we use this command.

>> docker tag 1492446588e3 foodorderingapp12/regapp:latest

>> docker images

>> docker push foodorderingapp12/regapp

--> the you check in your docker Hub there Repository is created

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

\*) Jenkins Job to build an image onto ansible

[ansadmin@ansible-server docker]$

we going to update our ansible playbook

[ansadmin@ansible-server docker]$ vi regapp.yml

- hosts: ansible <<--- this is ansible host name

------->

---

- hosts: ansible

tasks:

- name: create docker image

command: docker build -t regapp:latest .

args:

chdir: /opt/docker

- name: create tag to push image onto dockerhub

command: docker tag regapp:latest foodorderingapp12/regapp:latest

- name: push docker image

command: docker push foodorderingapp12/regapp:latest

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

>> ansible-playbook regapp.yml --check

----> this playbook is run only in ansible host bez we use hosts: ansible.

or we want to excute this ansiboleIp addresh playbook by using commnads.

>> ansible-playbook regapp.yml --limit 172.31.95.232

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

after that goto Jenkins ,

Open Copy\_Artifacts\_onto\_Ansible jobs,

In that SSH Server

In Exec command you add this command

-->

ansible-playbook /opt/docker/regapp.yml

--> you can check in

[ansadmin@ansible-server docker]$ ll

there is webapp.war is update

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

How to create container on docker-host using ansible-playbook

-->

[ansadmin@ansible-server docker]$ vi deploy\_regapp.yml

-->

---

- hosts: dockerhost

tasks:

- name: create container

command: docker run -d --name regapp-server -p 8082:8080 foodorderingapp12/regapp:latest

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

but first in Docker server give the permision .sock file

[root@docker-host ~]# chmod 777 /var/run/docker.sock

-->

>> [ansadmin@ansible-server docker]$ ansible-playbook deploy\_regapp.yml --check

>> [ansadmin@ansible-server docker]$ ansible-playbook deploy\_regapp.yml

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

Now you can check in Docker server

there images will be creating

[root@docker-host ~]# docker images

[root@docker-host ~]# docker ps -a

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

Now you can check in Broswer

http://dockerIp:8082/webapp/

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

Continuous deployment of docker container using ansible playbook.

1) Remove exiting container

2) remove exiting image

3) Crate new container

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

[ansadmin@ansible-server docker]$ vi deploy\_regapp.yml

--->

---

- hosts: dockerhost

tasks:

- name: stop existing container

command: docker stop regapp-server

ignore\_errors: yes

- name: remove the container

command: docker rm regapp-server

ignore\_errors: yes

- name: remove image

command: docker rmi foodorderingapp12/regapp:latesh

ignore\_errors: yes

- name: create container

command: docker run -d --name regapp-server -p 8082:8080 foodorderingapp12/regapp:latest

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

>> ansible-playbook deploy\_regapp.yml --check

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

Jenkins CI/CD to deploy on container using Ansible

goto Copy\_Artifacts\_onto\_Ansible

Exec command

--> add this

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

ansible-playbook /opt/docker/regapp.yml;

sleep 10;

ansible-playbook /opt/docker/deploy\_regapp.yml;

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

1) first deploy in dockerHub

2) use for run in port

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

45.. section 6

EKS install procedure

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

Setup bootstrap server for eksctl

Create EC2 linux inside you

-->

1) download Linux x86 (https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html)

-->

>> curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

unzip awscliv2.zip

sudo ./aws/install

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

check aws version

>> [root@kuberhost ~]# aws --version

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

download Kubernetes 1.21

-->

curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.21.14/2024-04-19/bin/linux/amd64/kubectl

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

we give excute permission for kubectl

>> chmod +x kubectl

>> echo $PATH

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin

[root@kuberhost ~]# mv kubectl /usr/local/bin

[root@kuberhost ~]# kubectl version

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

dounload eksctl (https://docs.aws.amazon.com/emr/latest/EMR-on-EKS-DevelopmentGuide/setting-up-eksctl.html)

>>

curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" | tar xz -C /tmp

>> cd /tmp

>> ll

>> mv eksctl /usr/local/bin

>> eksctl version

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

After that create Roll in IAM service -->> EC2FullAccess

--> provide Policy

1)AdministratorAccess

2)AmazonEC2FullAccess

3)AWSCloudFormationFullAccess

4)IAMFullAccess

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

i am going to add this roll for my kubernet server

>> goto action -> modify roll

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

Setup kubernets using eksctl

[root@kuberhost tmp]#

>>

eksctl create cluster --name EKS-Cluster \

--region us-east-1 \

--node-type t2.micro

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

In AWS you open you CloudFormation and check there ..

your EKS cluster is created..

why bez he going to create enveroment using cloudformation

then In EC2 , EKS cluster is aslo creted

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

>> cat /root/.kube/config

in that file we will see all All imformation reageding our ec2 cluster..

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

we check nodes

>> kubectl get nodes

>> kubectl get all

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

now we create pod

>> kubectl run webapp --image=httpd

>> kubectl get all

--> wew want see only pod

>> kubectl get po

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

Run Kubernetes Basic Command

Create Pods

>> kubectl create deployment demo-nginx --image=nginx --port=80 --replicas=2

--> From this commnads WE PULL nginx images from docker hub and we create deployment called demo-nginx.

--> and in backend it's create replica set respect of this flag (--replicas=2)

--> we are not give this means it's create 1 pod but we provied 2 this is means creat 2 pod.

---> Now to check out our deployments,

we can use kubectl get deployments.

>> kubectl get deployment

or

>> kubectl get deploy

--->In the backend,

it is going to create a replica set

So you can see here it is created a replica set

>> kubectl get replicaset

--> also we can check the kubectl get pod

>> kubectl get po

>> kubectl get pod

both are same

>> kubectl get all ---> you see all resources in one

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

>> kubectl expose deployment demo-nginx --port=80 --type=LoadBalancer

service/demo-nginx exposed

>> kubectl get all

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

After that go to Aws in LoadBalancer

it's create LoadBalancer hereh

--> also our two instances should be attached

to this Load Balancer and it should be in service.

--> it is listening

on port number 80 and it forwarded to the TCP:31426

--> let's access it from the browser,

from kubernetserver copy EXTERNAL-IP

>> aed03d84dfc8a49bdaa96b132ab5468d-662566343.us-east-1.elb.amazonaws.com

--> we could able to access our application.

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

>> kubectl expose deployment demo-nginx --port=8080 --type=LoadBalancer

service/demo-nginx exposed

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

how we can create a Pod and Service,

by using the manifest files.

--> delete the existing configurations

>> kubectl delete deployment demo-nginx

--> we need to delete our Load Balancer as well

>> kubectl delete service/demo-nginx

--> I'm going to create a file called pod.yml.

we write this file in yml format

>>vi pod.yml

apiVersion: v1

kind: Pod

metadata:

name: demo-pod

labels:

app: demo-app

spec:

containers:

- name: demo-nginx

image: nginx

ports:

- name: demo-nginx

containerPort: 80

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

Create a service manifest file

>> vi service.yml

apiVersion: v1

kind: Service

metadata:

name: demo-service

spec:

ports:

- name: nginx-port

port: 80

targetPort: 80

selector:

app: demo-app

type: LoadBalancer

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

first we create pod then we create service

-->

>> kubectl apply -f pod.yml

>> kubectl apply -f service.yml

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

It's created a cluster.

And you can see here instances it is OutOfService.

Why? Because, why because we have created a pod.

We have created a service,

and while creating a service, we got a LoadBalancer.

That LoadBalancer we are accessing

from the external network.

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

Using Labels and selector

when we get all details when we are forwarded this request.

>> kubectl get all

>> kubectl describe service/demo-service

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

this command gives that what is the pod name,

what is the pod ip and on which server it is running.

All this information.

>> kubectl get pod -o wide

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

Write a deployment file

first we delete our pod

>> kubectl get all

>> kubectl delete pod demo-pod

and also delele our services

>> kubectl delete service/demo-service

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

If we delete or if pod get terminated,

there is no way it is going to recreate by default.

how we can overcome this problem?

we need to do some changes

to the way of using our manifest files.

we have tried to deploy a few applications

by using the "KubeCTL" command with the Deployment option.

Same thing, we can use it as a manifest file,

so we can create manifest files for Deployments.

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

>> vi regapp-deployment.yml

spec:

replicas: 3

selector:

matchLabels:

app: regapp

template:

metadata:

labels:

app: regapp

spec:

containers:

- name: regapp

image: foodorderingapp12/regapp

imagePullPolicy: Always

ports:

- containerPort: 8080

strategy:

type: RollingUpdate

rollingUpdate:

maxSurge: 1

maxUnavailable: 1

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

>> kubectl apply -f regapp-deployment.yml

Get More impormation about pod

>> kubectl get pod -o wide

after that 3 pod is creted ..

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

apiVersion: v1

kind: Service

metadata:

name: valaxy-service

labels:

app: regapp

spec:

selector:

app: regapp

ports:

- port: 8080

targetPort: 8080

type: LoadBalancer

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

Now you can excute you service file

>> kubectl apply -f regapp-service.yml

>> kubectl get all

this command show you your services details.

>> kubectl describe service/valaxy-service

>> kubectl get pod -o wide

--> this file going to crete you LoadBalancer

--> goto you AWS and see LoadBalancer

and copy you DNS name :8080 --> it's our liceners

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

>> kubectl delete pod akshay-regapp-8b96f895f-l5jpb

or whenever we going to delele pod it's going to create new one

>> kubectl get pod

--> it we see 3 pods

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

Integrate Kubernetes bootstrap server with Ansible

[root@kuberhost ~]# useradd ansadmin

[root@kuberhost ~]# passwd ansadmin

Aksh@y06

[root@kuberhost ~]# visudo

## Allow root to run any commands anywhere

root ALL=(ALL) ALL

ansadmin ALL=(ALL) NOPASSWD: ALL

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

[root@kuberhost ~]# vi /etc/ssh/sshd\_config

# To disable tunneled clear text passwords, change to no here!

PasswordAuthentication yes

#PermitEmptyPasswords no

#PasswordAuthentication no

--------->

[root@kuberhost ~]# service sshd reload

[root@kuberhost ~]# ifconfig

copy id -->

172.31.35.203

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

Goto our Ansible server

In Ansible goto Docker Folder in that you create hosts file

and there you going to add kubernets and Ansible private id

-->

[ansadmin@ansible-server docker]$ vi hosts

localhost

[kubernet]

172.31.35.203

[ansible]

172.31.95.232

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

[ansadmin@ansible-server docker]$ ssh-copy-id 172.31.35.203

we are going to add kubernest server with our ansible

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

then we are going to check our hosts file is connect or no by using this command

[ansadmin@ansible-server docker]$ ansible -i hosts all -a uptime

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

Create ansible playbook for deploy and service file

we ceate file in Ansible-server

[ansadmin@ansible-server docker]$ vi kube\_deploy.yml

-->

---

- hosts: kubernet

# become: true

user: root

tasks:

- name: deploy regapp on kubernet

command: kubectl apply -f regapp-deployment.yml

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

[ansadmin@ansible-server docker]$ vi kube\_service.yml

-->

---

- hosts: kubernet

user: root

tasks:

- name: deploy regapp on kubernetes

command: kubectl apply -f regapp-service.yml

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

But first we delete services in Kubernetes-server

[root@kuberhost ~]# ls

[root@kuberhost ~]# kubectl delete -f regapp-service.yml

[root@kuberhost ~]# kubectl delete -f regapp-deployment.yml

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

executing our ansible-playbook

But before you copy ssh id as root

and set password in kubernes-server

>>

[root@kuberhost ~]# passwd root

Aksh@y06

------------->

172.31.35.203 <<<----- this Ip is Kubernetes Ip priv ate

[ansadmin@ansible-server docker]$ ssh-copy-id root@172.31.35.203

[ansadmin@ansible-server docker]$ ansible-playbook -i /opt/docker/hosts kube\_deploy.yml

[ansadmin@ansible-server docker]$ ansible-playbook -i /opt/docker/hosts kube\_service.yml

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

then you check in Kubernetes pods ans service will be create

[root@kuberhost ~]# kubectl get pods

[root@kuberhost ~]# kubectl get all

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

We are going to excute this fils using Jenkis

Create Jenkins deployment job for Kubernetes

Now create new Item

Give Name as

Deploy\_On\_Kubernets

uses freedtyle Project

---->

we are going to deploy On Kubernetes

choose

-->Send build artifacts over SSH

Name

ansible --> this is our server

Exec command

-->

ansible-playbook -i /opt/docker/hosts /opt/docker/kube\_deploy.yml;

ansible-playbook -i /opt/docker/hosts /opt/docker/kube\_service.yml

-------------------------Save------

In Kubernetes-server you delele privius pod and service

[root@kuberhost ~]# kubectl get all

[root@kuberhost ~]# kubectl delete deployment.apps/akshay-regapp

[root@kuberhost ~]# kubectl delete service/valaxy-service

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

57...........

