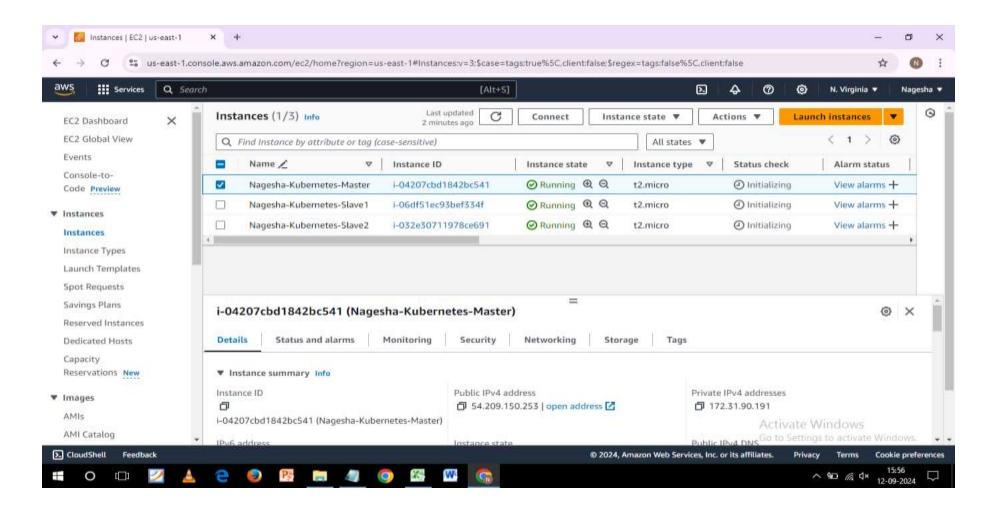
AWS Solutions Architect Certification Training

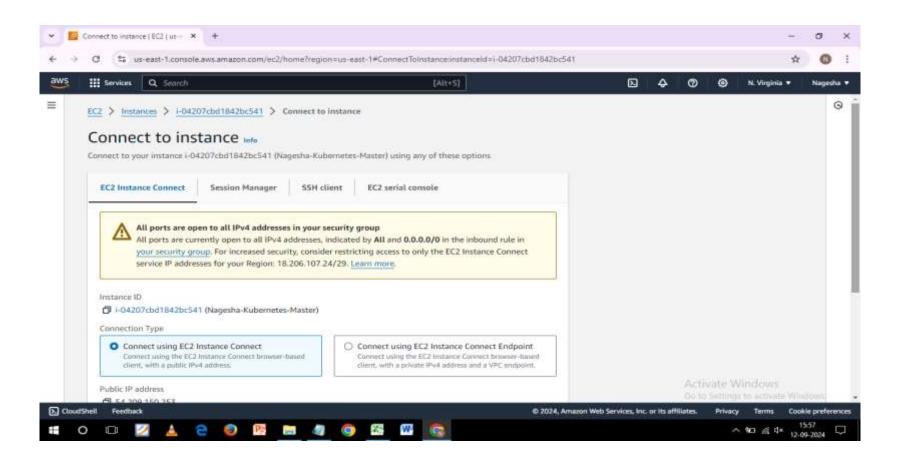
support@intellipaat.com - +91-7022374614 - US: 1-800-216-8930 (Toll Free)

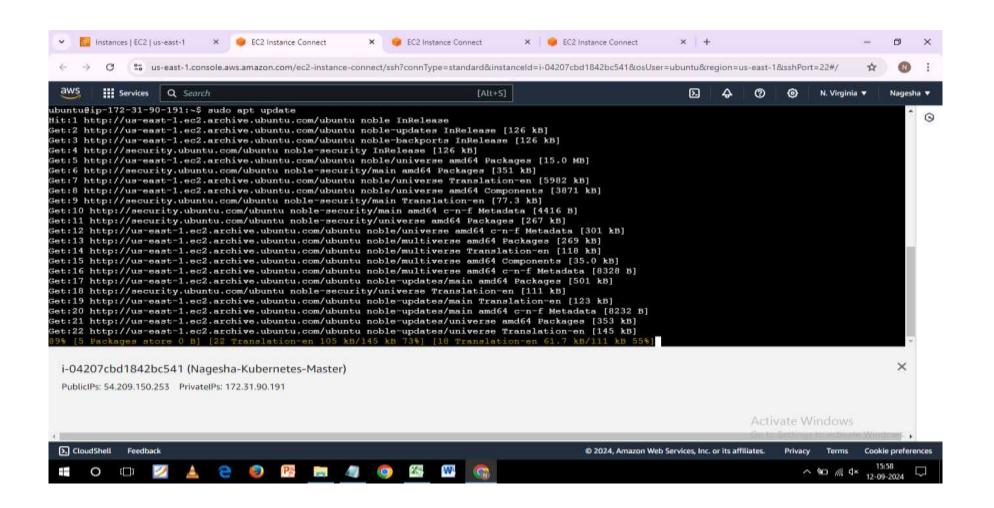
DevOps-Module7-Kubernetes-Assignment-1 COMPLETED by Nagesha KS Please check the following screenshots for each question.

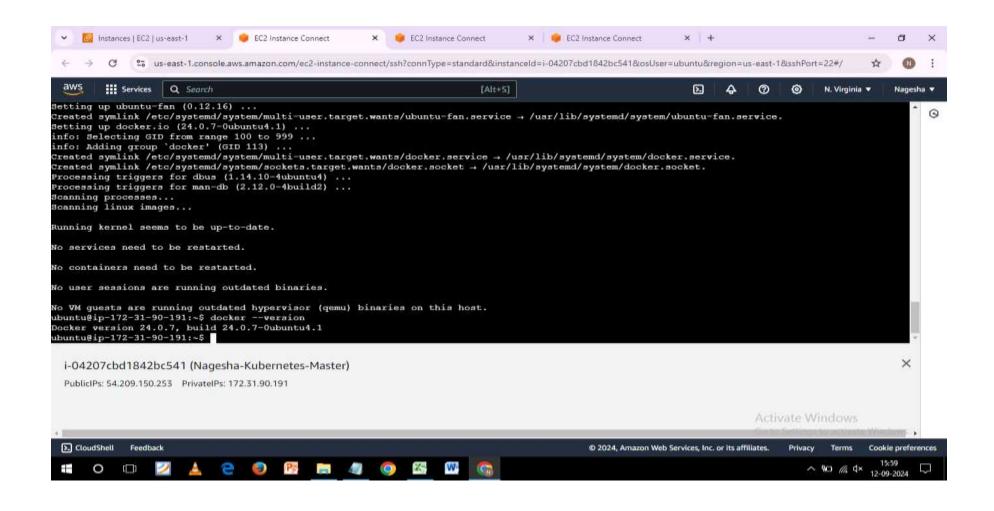
Tasks To Be Performed:

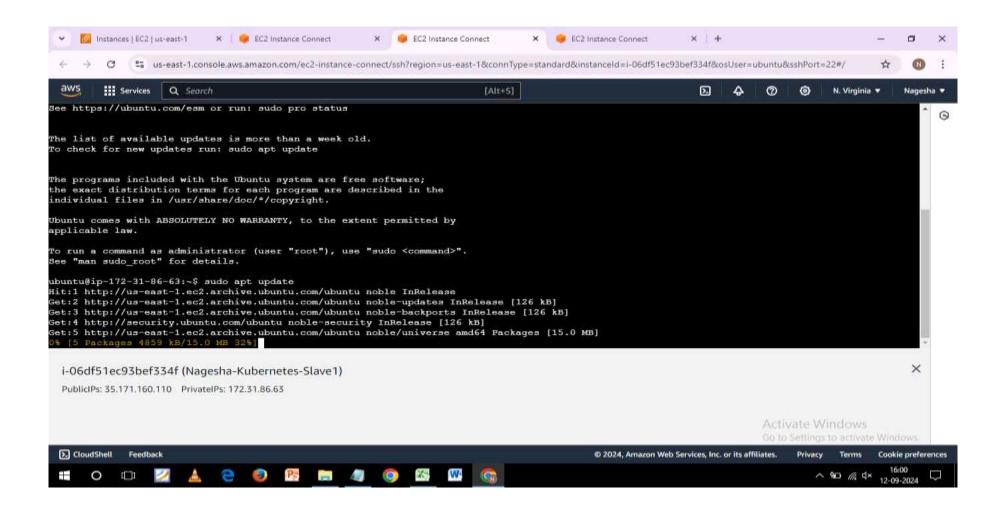
- 1. Deploy a Kubernetes cluster for 3 nodes
- 2. Create a NGINX deployment of 3 replicas

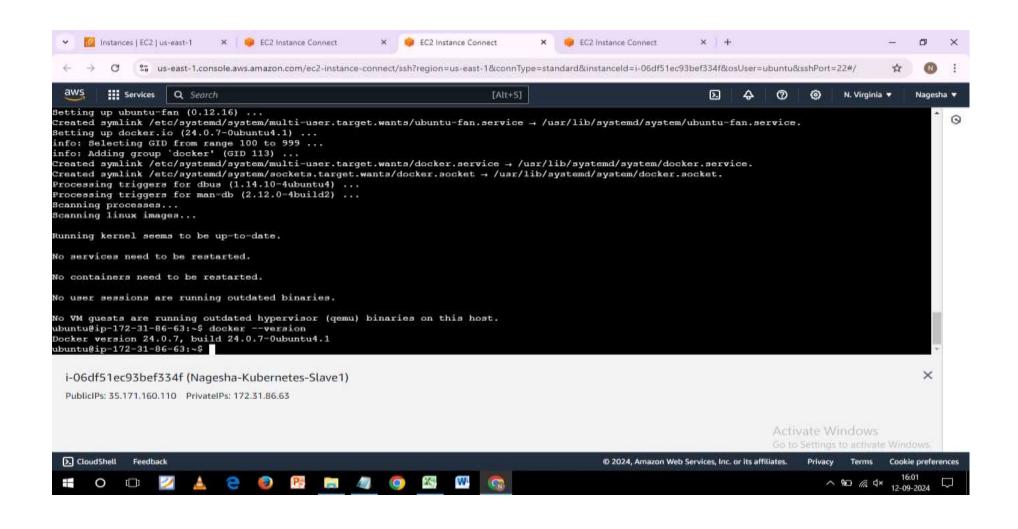


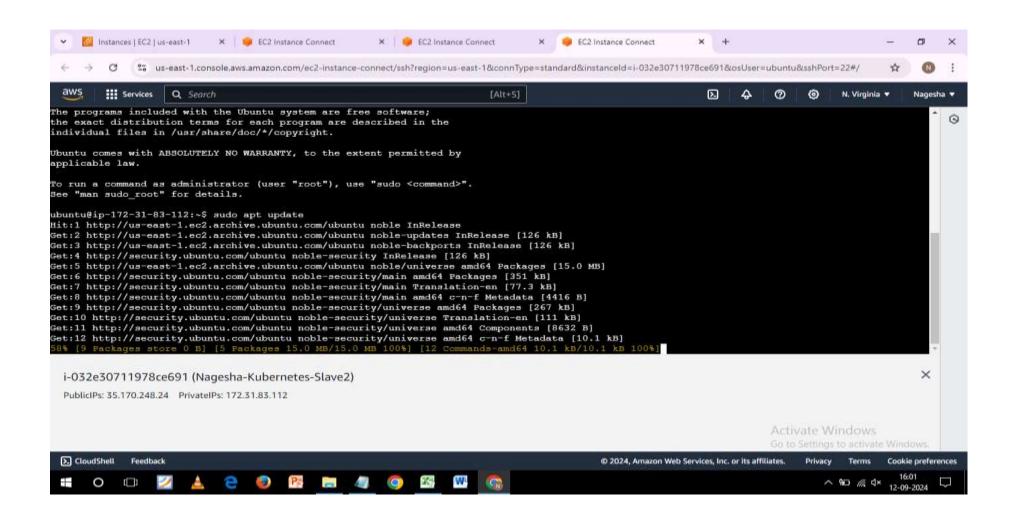


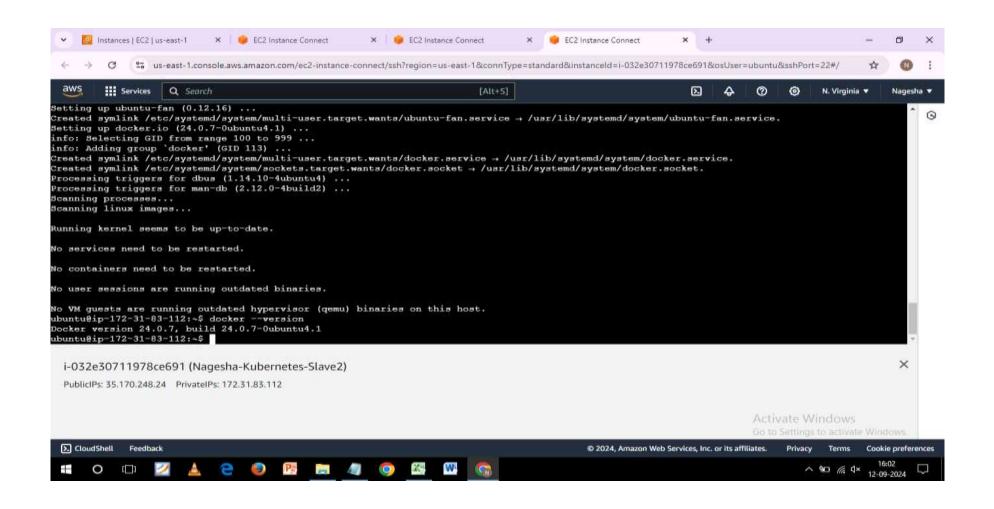


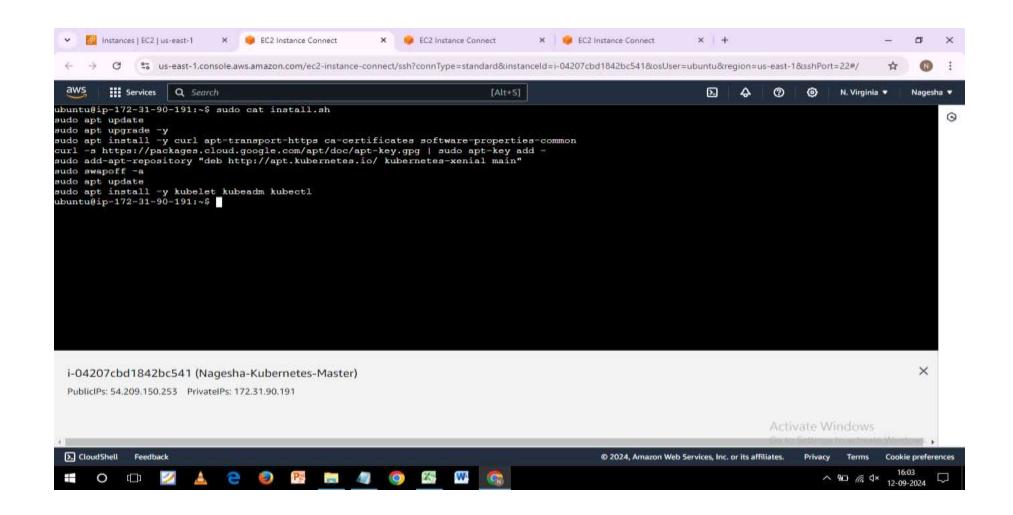


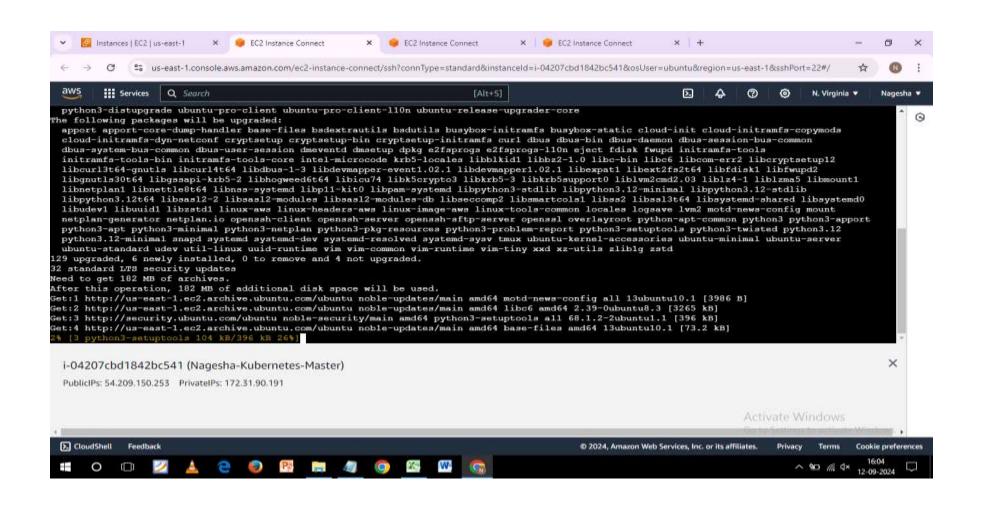


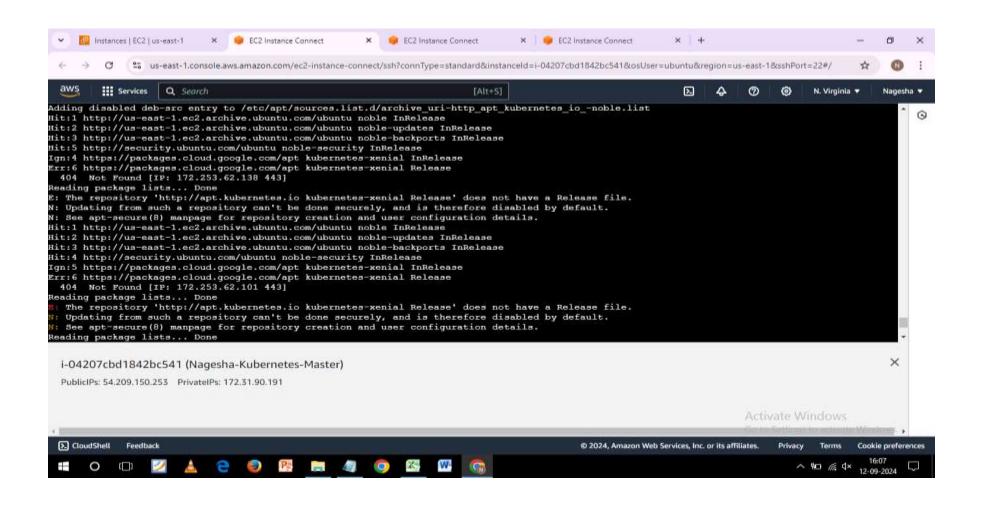


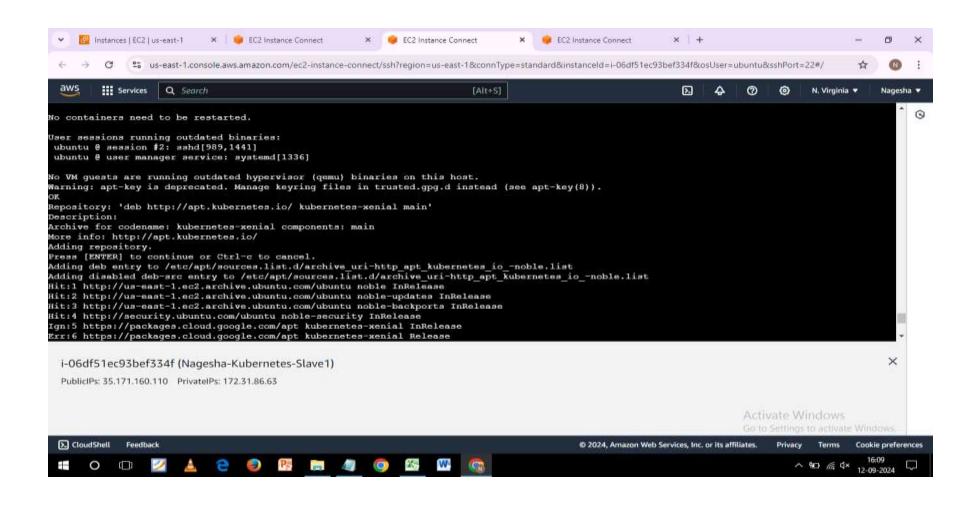


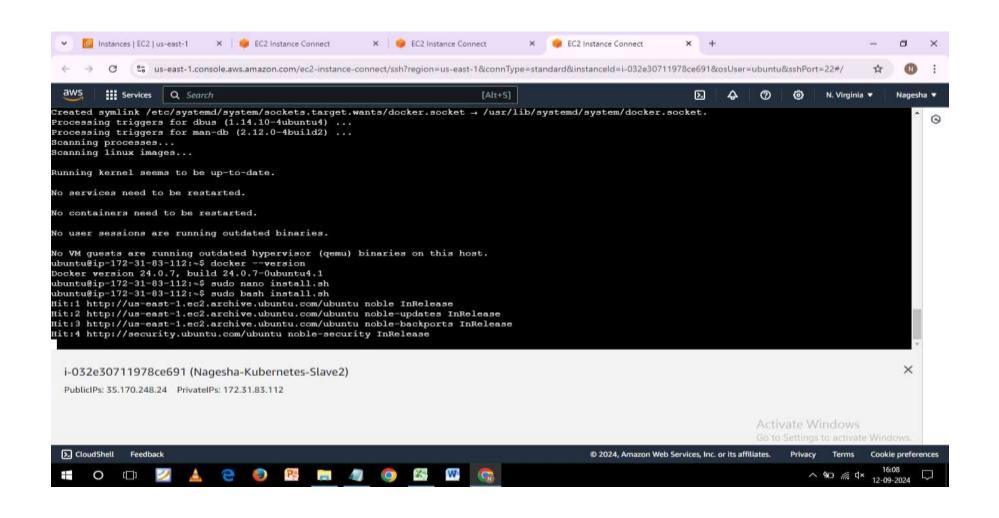


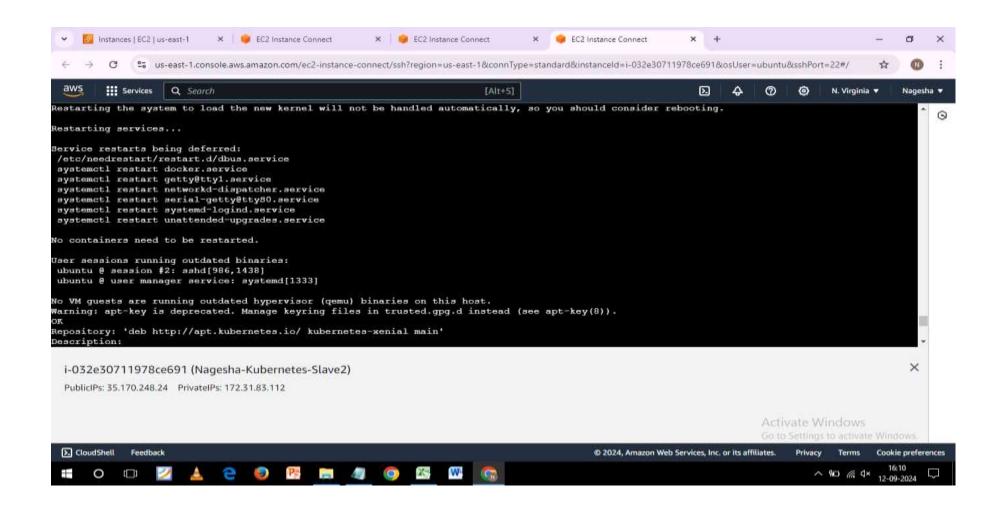












Kubernetes Commands

AWS Console

launch instance EC2 number of instances 3

Ubuntu

20.04 LTS

t2.medium is needed for K8S as t2.micro is not sufficient

security group: Add all traffic in Security group

rename instances as

Nagesha-Kubernetes-Master

Nagesha-Kubernetes-Slave1

Nagesha-Kubernetes-Slave2

connect through EC2 dashboard connection, all the instnaces one by one

Update and Install Docker on all Machines (Master, Slave1, Slave2)

sudo apt update -- master

sudo apt update -- slave1

sudo apt update -- slave2

sudo apt install docker.io -y -- do in all machines of master, slave1 and slave2

docker --version

Create a script file in all machines

goto master

sudo nano install.sh

sudo apt update

sudo apt upgrade -y

sudo apt install -y curl apt-transport-https ca-certificates software-properties-common

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

sudo add-apt-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"

sudo swapoff -a

sudo apt update

sudo apt install -y kubelet kubeadm kubectl

Run the install.sh file in all machine (master, slave1 and slave2) sudo bash install.sh

Verify if Kubeadm, Kubectl, Kubelet installed which Kubeadm && which Kubectl && which Kubelet

In Master machine -- to initialize k8s sudo kubeadm init

In Slave machines paste the token in a root user

In Master machines paste below command being a normal user

mkdir -p \$HOME/.kube sudo cp -i/etc/kubernetes/admin.conf \$HOME/. kube/config sudo chown \$(id -u):\$(id -g) \$HOME/.kube/config

Alternatively, if you are the root user, you can run: export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

https://kubernetes.io/docs/concepts/cluster-administration/addons/
kubectl apply -f https://github.com/weaverworks/releases/download/v2.8.1/weave-daemonset-k8s.yaml

goto slave1 and slave2 sudo kubeadm join 172.31.27.5.6443 --token xix122.z......

goto master kubectl get no it shows master and slave1 and slave2.... control.planne is master under ROLES --- <none> are slave1 and slave2 respectively Create a NGINX deployment of 3 replicas will be done as follows go to official document

goto google deployment k8s

the first search Deployments has to be clicked

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: nginx-deployment
 labels:
  app: nginx
spec:
 replicas: 3
 selector:
  matchLabels:
   app: nginx
 template:
  metadata:
   labels:
   app: nginx
  spec:
   containers:
   - name: nginx
   image: nginx:
    ports:
    - containerPort: 30008
```

go to master
sudo nano assign1.yaml
paste here
edit this file
check for replica 3 are there
next check for nginx and remove the version and retain only nginx
replicas: 3

image: nginx

goto master check if pods are preent **kubectl get po**

you will see no resoures found in default namespace.

kubectl get deploy

you will see no resources found in default namespace.

kubectl apply -f assign1.yaml

deployment.apps/nginx-depoyemtn created.
kubectl get deploy
nginx-deployment
kubectl get po
you will see pods now.
kubectl get po -o wide
you will see pods with IP and NODE etc.