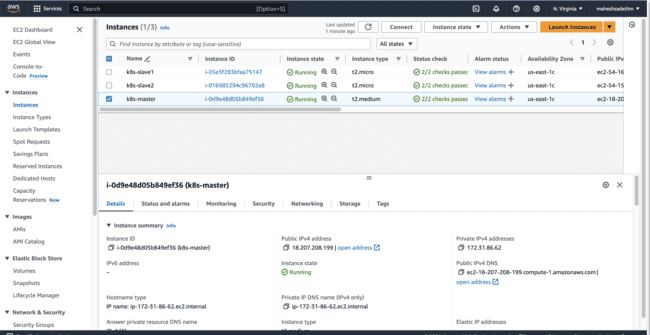


Go to AWS CONSOLE .

Launch t2.medium instance for Kubernetes master cluster and 2 slave t2.micro

Select default subnet



Update all the 3 EC2 Instances

By sudo apt update

Install docker on all 3 instances by the below script.

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

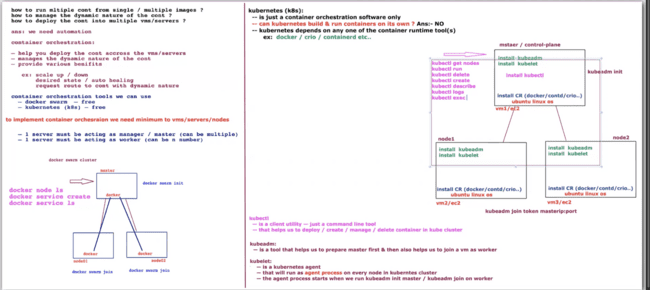
sudo chmod 755 /tmp/installDocker.sh

sudo bash /tmp/installDocker.sh

check docker by the command docker -v ( installed in all 3 machines )

ubuntu@ip-172-31-94-196:~$ docker -v

docker version 27.2.1, build 9e34c9b



Run the command to join the slave nodes

kubeadm join token masterip:port

Now go to the github link of the trainer naresh and install Kubernetes

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

**Install Kubernetes Using Script**

**Step1: On Master Node Only**

## Install Docker

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

sudo chmod 755 /tmp/installDocker.sh

sudo bash /tmp/installDocker.sh

sudo systemctl restart docker.service

## Install CRI-Docker

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

sudo chmod 755 /tmp/installCRIDockerd.sh

sudo bash /tmp/installCRIDockerd.sh

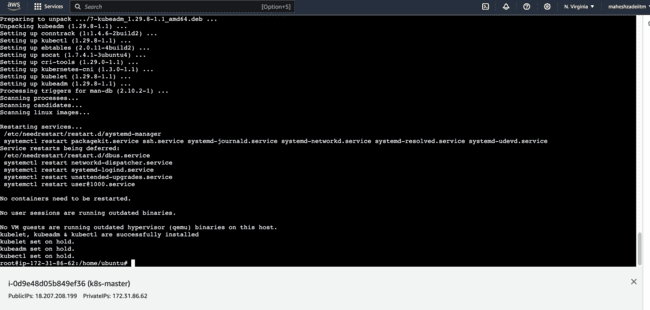
sudo systemctl restart cri-docker.service

## Install kubeadm,kubelet,kubectl

sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installK8S.sh -P /tmp

sudo chmod 755 /tmp/installK8S.sh

sudo bash /tmp/installK8S.sh



Installed kubeadm,kubelet,kubectl successfully

Now pls verify all by below command

# Validate

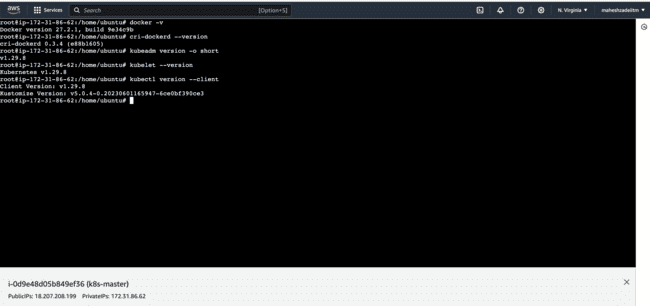
docker -v

cri-dockerd --version

kubeadm version -o short

kubelet --version

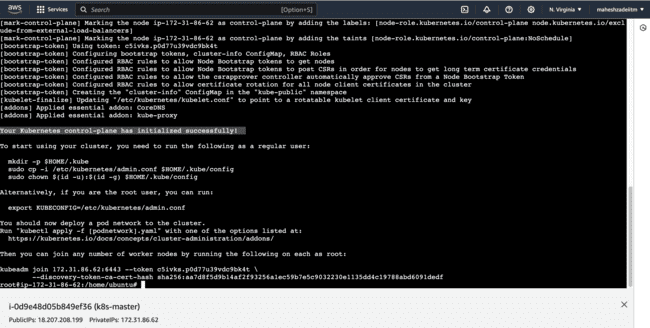
kubectl version --client



## Initialize kubernetes Master Node

sudo kubeadm init --cri-socket unix:///var/run/cri-dockerd.sock --ignore-preflight-errors=all ( ignore if the core the server is 1 core type full command or if 2 core then type the command

sudo kubeadm init --cri-socket unix:///var/run/cri-dockerd.sock)

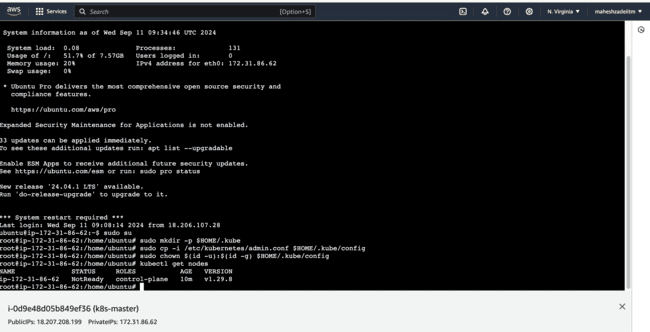


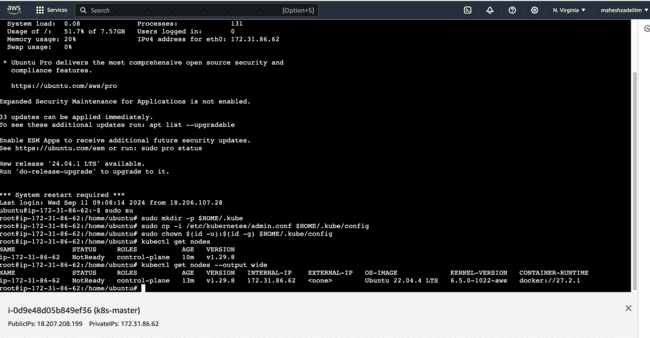
Pls run the below commands as mandatory to run kubectl

sudo mkdir -p $HOME/.kube

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

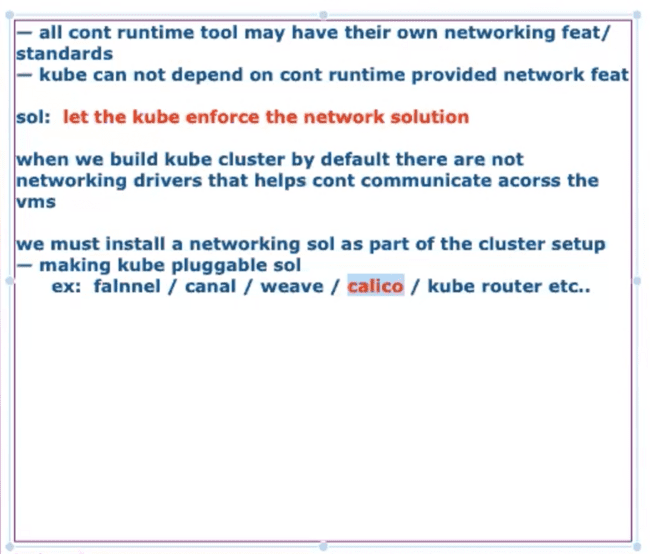
sudo chown $(id -u):$(id -g) $HOME/.kube/config



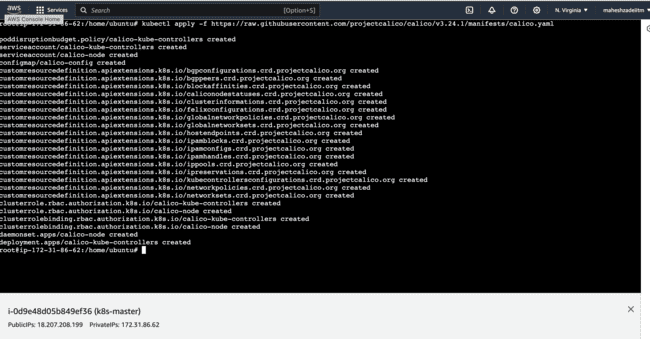


## install networking driver -- Weave/flannel/canal/calico etc...

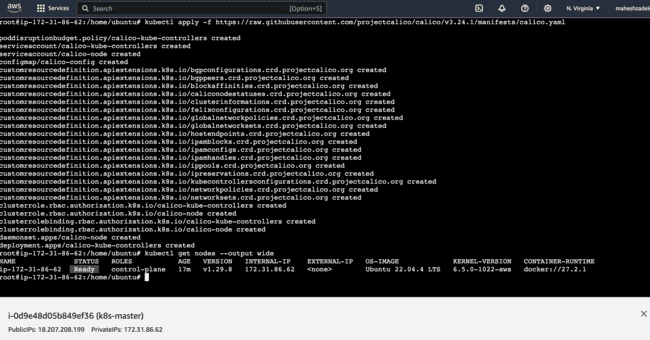
## below installs calico networking driver



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



Now status is ready



**Master setup is completed**

# Validate: kubectl get nodes –output wide

**Step2: On All Worker Nodes**

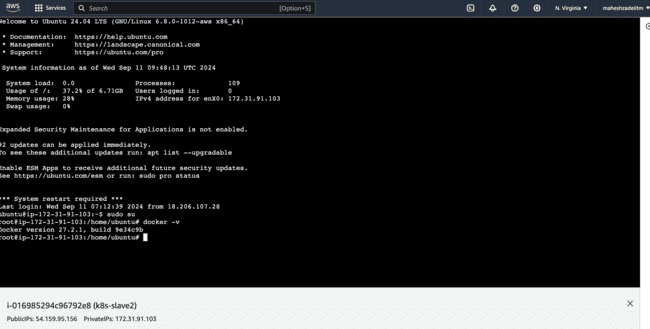
## Install Docker

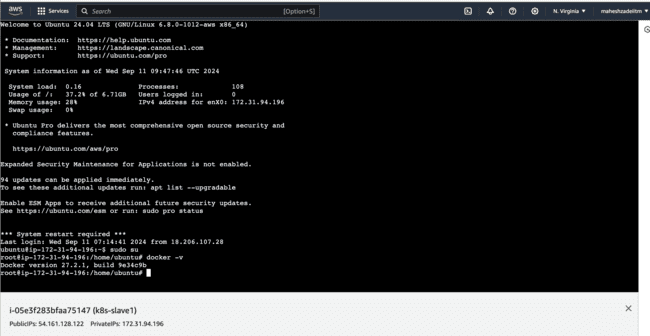
sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installDocker.sh -P /tmp

sudo chmod 755 /tmp/installDocker.sh

sudo bash /tmp/installDocker.sh

sudo systemctl restart docker.service





## Install CRI-Docker

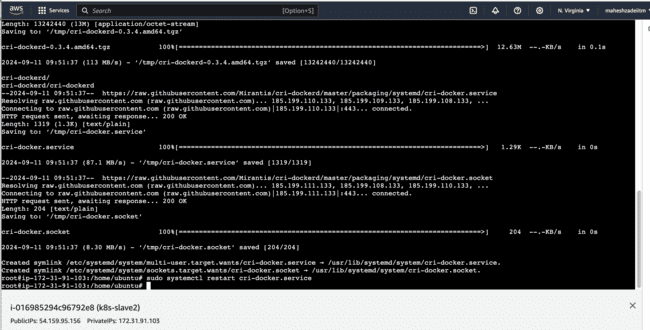
sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installCRIDockerd.sh -P /tmp

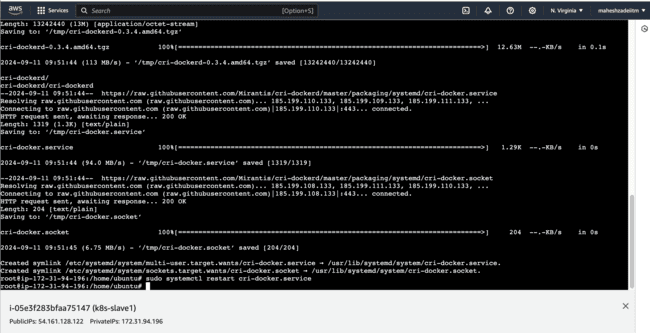
sudo chmod 755 /tmp/installCRIDockerd.sh

sudo bash /tmp/installCRIDockerd.sh

sudo systemctl restart cri-docker.service

All comnads run in both the slave nodes





## Install kubeadm,kubelet,kubectl

sudo wget https://raw.githubusercontent.com/lerndevops/labs/master/scripts/installK8S.sh -P /tmp

sudo chmod 755 /tmp/installK8S.sh

sudo bash /tmp/installK8S.sh

# Validate

docker -v

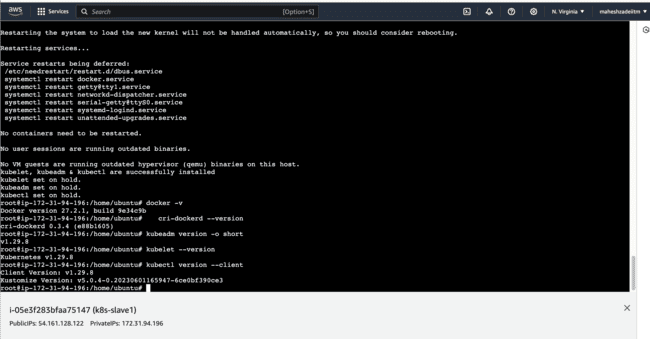
cri-dockerd --version

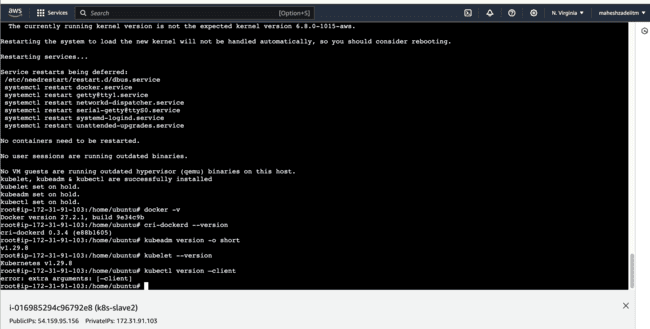
kubeadm version -o short

kubelet --version

kubectl version –client

run all above command in both slave nodes



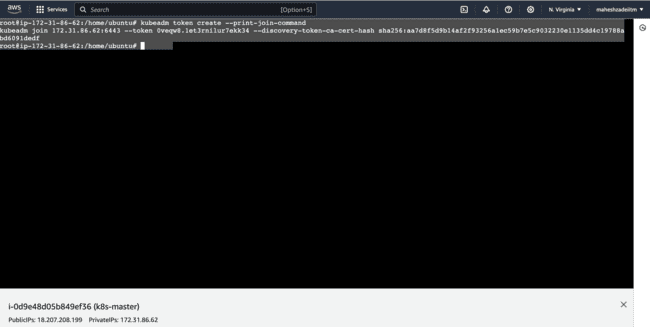


## Run Below on Master Node to get join token

kubeadm token create --print-join-command

root@ip-172-31-86-62:/home/ubuntu# kubeadm token create --print-join-command

kubeadm join 172.31.86.62:6443 --token 0veqw8.1et3rnilur7ekk34 --discovery-token-ca-cert-hash sha256:aa7d8f5d9b14af2f93256a1ec59b7e5c9032230e1135dd4c19788abd6091dedf



copy the kubeadm join token from master &

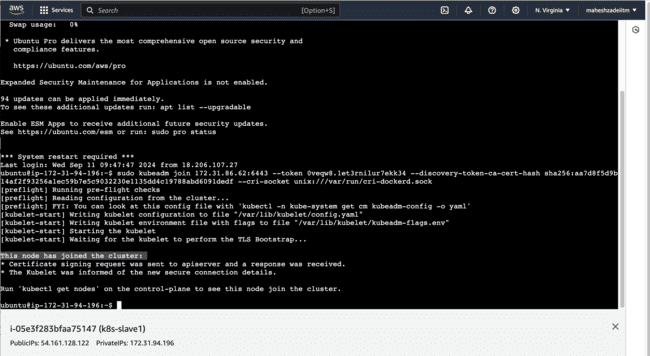
ensure to add --cri-socket unix:///var/run/cri-dockerd.sock as below &

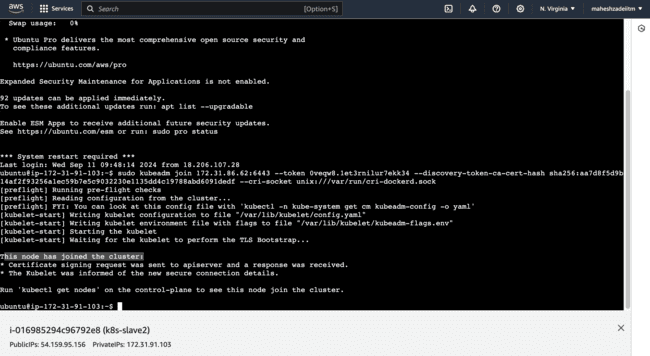
ensure to add sudo

then run on worker nodes

type full command on both the server

ubuntu@ip-172-31-91-103:~$ sudo kubeadm join 172.31.86.62:6443 --token 0veqw8.1et3rnilur7ekk34 --discovery-token-ca-cert-hash sha256:aa7d8f5d9b14af2f93256a1ec59b7e5c9032230e1135dd4c19788abd6091dedf --cri-socket unix:///var/run/cri-dockerd.sock





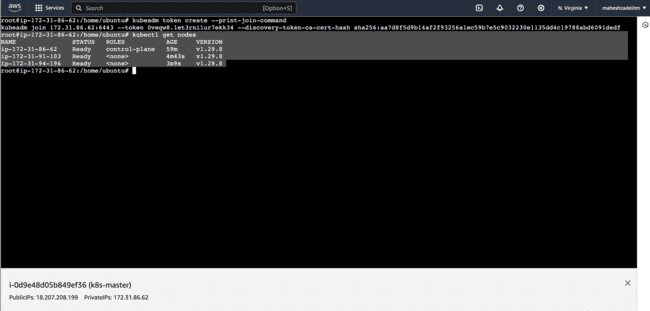
root@ip-172-31-86-62:/home/ubuntu# kubectl get nodes

NAME STATUS ROLES AGE VERSION

ip-172-31-86-62 Ready control-plane 59m v1.29.8

ip-172-31-91-103 Ready <none> 4m43s v1.29.8

ip-172-31-94-196 Ready <none> 3m9s v1.29.8



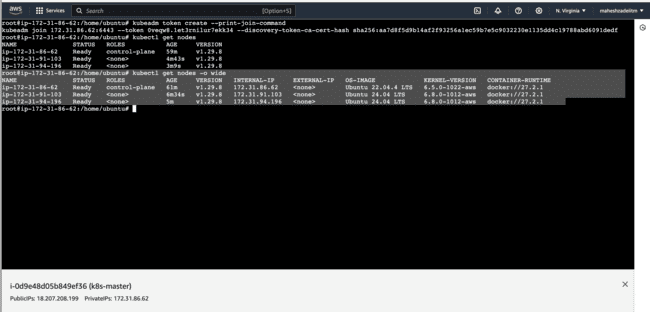
root@ip-172-31-86-62:/home/ubuntu# kubectl get nodes -o wide

NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE KERNEL-VERSION CONTAINER-RUNTIME

ip-172-31-86-62 Ready control-plane 61m v1.29.8 172.31.86.62 <none> Ubuntu 22.04.4 LTS 6.5.0-1022-aws docker://27.2.1

ip-172-31-91-103 Ready <none> 6m34s v1.29.8 172.31.91.103 <none> Ubuntu 24.04 LTS 6.8.0-1012-aws docker://27.2.1

ip-172-31-94-196 Ready <none> 5m v1.29.8 172.31.94.196 <none> Ubuntu 24.04 LTS 6.8.0-1012-aws docker://27.2.1

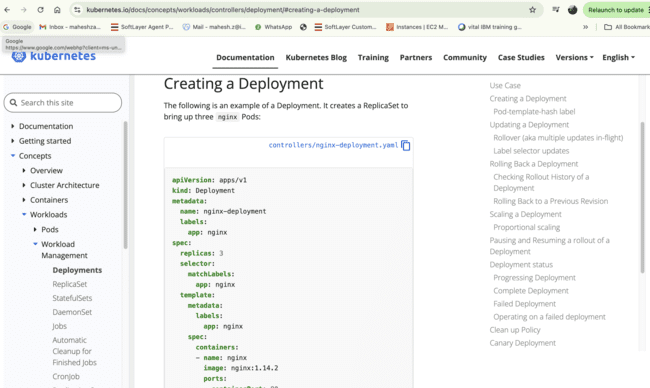


**I have successfully deploy a Kubernetes cluster for 3 nodes .**

Create Nginx deployment for 3 replicas

Go to this Kubernetes official website for the replicas

<https://kubernetes.io/docs/concepts/workloads/controllers/deployment/#creating-a-deployment>



Copy the code and crerate the yaml file in the master node

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:

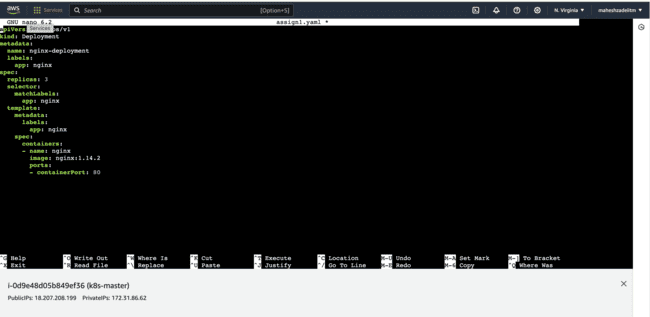
containers:

- name: nginx

image: nginx:1.14.2

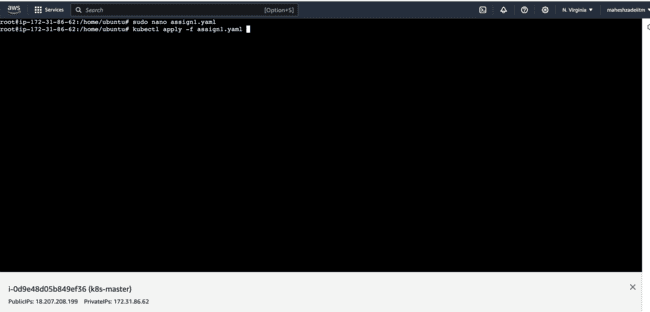
ports:

- containerPort: 80



root@ip-172-31-86-62:/home/ubuntu# sudo nano assign1.yaml

root@ip-172-31-86-62:/home/ubuntu# kubectl apply -f assign1.yaml



root@ip-172-31-86-62:/home/ubuntu# sudo nano assign1.yaml

root@ip-172-31-86-62:/home/ubuntu# kubectl apply -f assign1.yaml

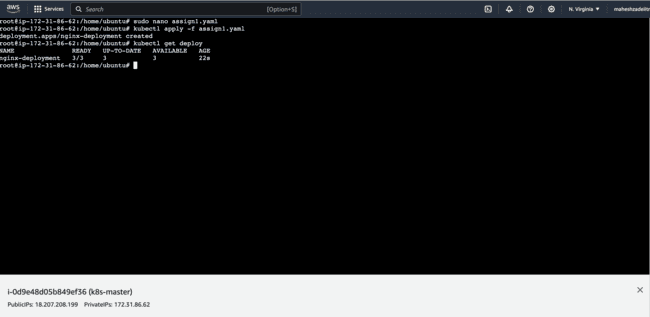
deployment.apps/nginx-deployment created

root@ip-172-31-86-62:/home/ubuntu# kubectl get deploy

NAME READY UP-TO-DATE AVAILABLE AGE

nginx-deployment 3/3 3 3 22s

root@ip-172-31-86-62:/home/ubuntu#



root@ip-172-31-86-62:/home/ubuntu# kubectl get pod

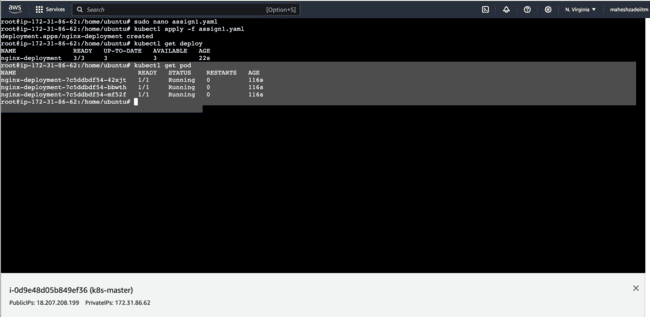
NAME READY STATUS RESTARTS AGE

nginx-deployment-7c5ddbdf54-42xjt 1/1 Running 0 116s

nginx-deployment-7c5ddbdf54-bbwth 1/1 Running 0 116s

nginx-deployment-7c5ddbdf54-mf52f 1/1 Running 0 116s

root@ip-172-31-86-62:/home/ubuntu#



root@ip-172-31-86-62:/home/ubuntu# kubectl get pod -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

nginx-deployment-7c5ddbdf54-42xjt 1/1 Running 0 3m57s 192.168.126.65 ip-172-31-94-196 <none> <none>

nginx-deployment-7c5ddbdf54-bbwth 1/1 Running 0 3m57s 192.168.225.1 ip-172-31-91-103 <none> <none>

nginx-deployment-7c5ddbdf54-mf52f 1/1 Running 0 3m57s 192.168.126.66 ip-172-31-94-196 <none> <none>

root@ip-172-31-86-62:/home/ubuntu#