**Module-9: Kubernetes Assignment – 1**

● Deploy a Kubernetes Cluster for 3 nodes

Master –

sudo swapoff -a

# Create the .conf file to load the modules at bootup

cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf

overlay

br\_netfilter

EOF

sudo modprobe overlay

sudo modprobe br\_netfilter

# sysctl params required by setup, params persist across reboots

cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf

net.bridge.bridge-nf-call-iptables = 1

net.bridge.bridge-nf-call-ip6tables = 1

net.ipv4.ip\_forward = 1

EOF

# Apply sysctl params without reboot

sudo sysctl --system

## Install CRIO Runtime

sudo apt-get update -y

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

sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg

echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/ /" | sudo tee /etc/apt/sources.list.d/cri-o.list

sudo apt-get update -y

sudo apt-get install -y cri-o

sudo systemctl daemon-reload

sudo systemctl enable crio --now

sudo systemctl start crio.service

echo "CRI runtime installed successfully"

# Add Kubernetes APT repository and install required packages

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update -y

sudo apt-get install -y kubelet="1.29.0-\*" kubectl="1.29.0-\*" kubeadm="1.29.0-\*"

sudo apt-get update -y

sudo apt-get install -y jq

sudo systemctl enable --now kubelet

sudo systemctl start kubelet

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

## Execute ONLY on "Master Node"

sudo kubeadm config images pull

sudo kubeadm init

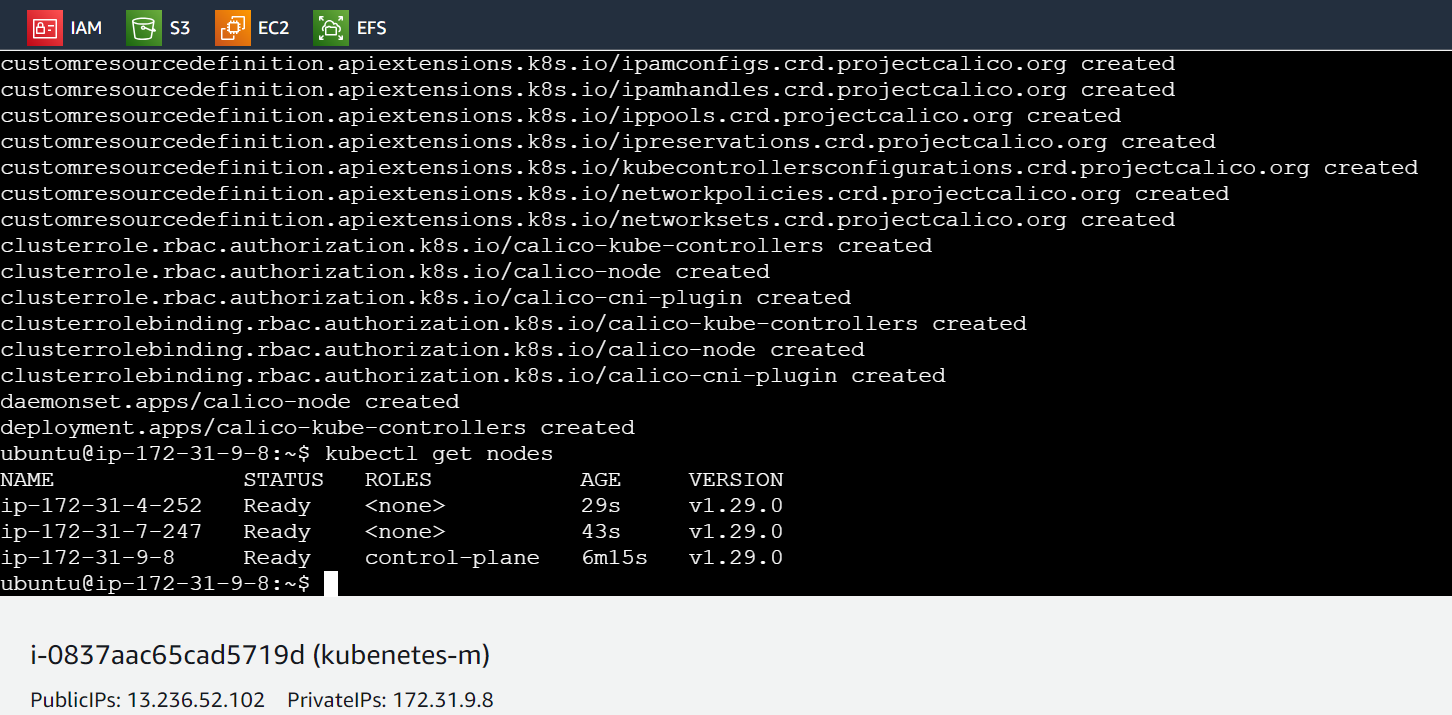
mkdir -p "$HOME"/.kube

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

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

# Network Plugin = calico

kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml



Nodes –

## Execute on ALL of your Worker Node's

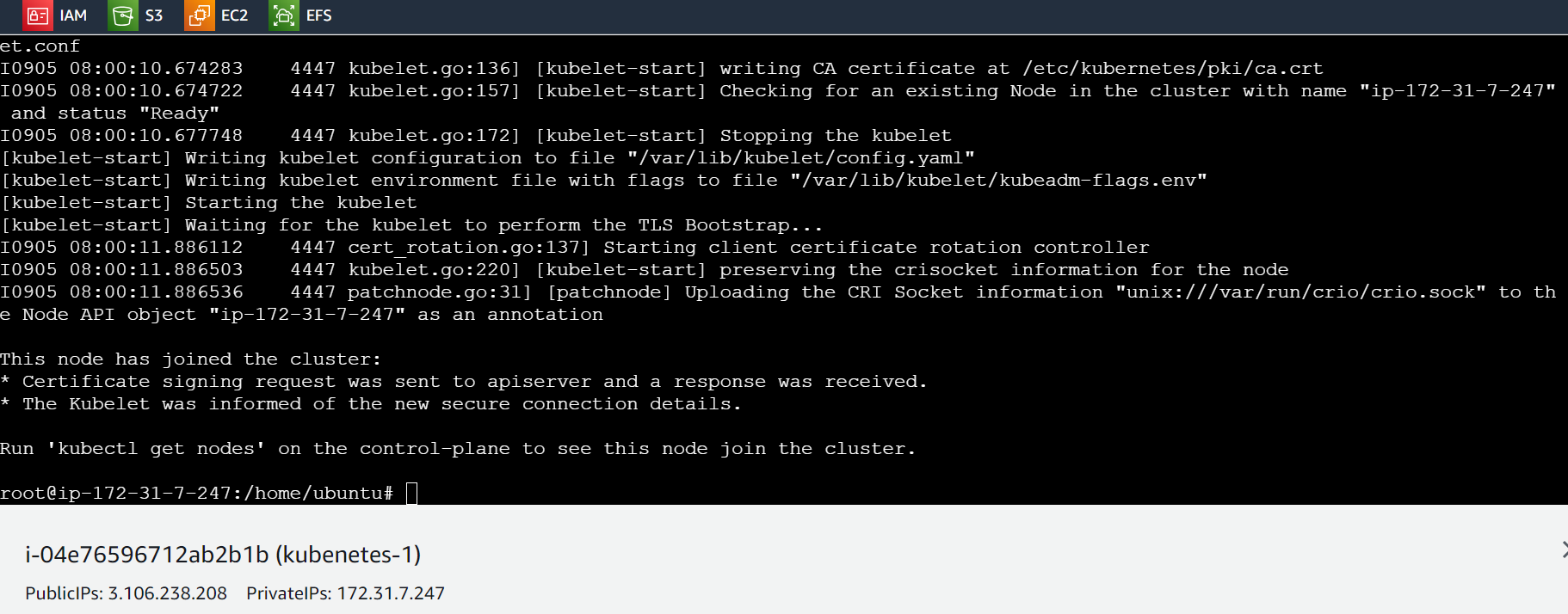
1. Perform pre-flight checks

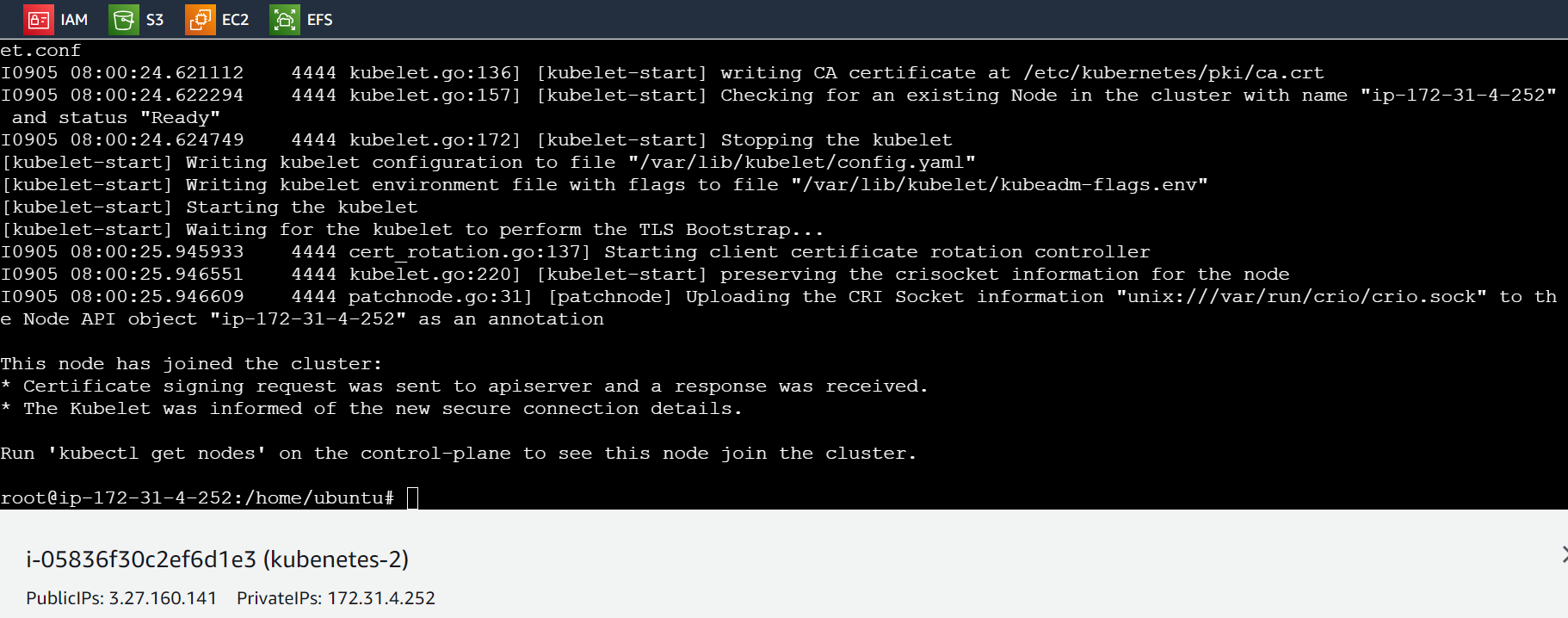
sudo kubeadm reset pre-flight checks

2. Paste the join command you got from the master node and append `--v=5` at the end but first use sudo su command to become root (avoid using sudo your-token).

sudo su

<your-token --v=5>

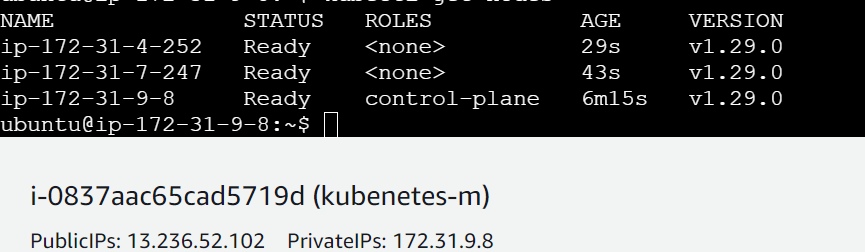




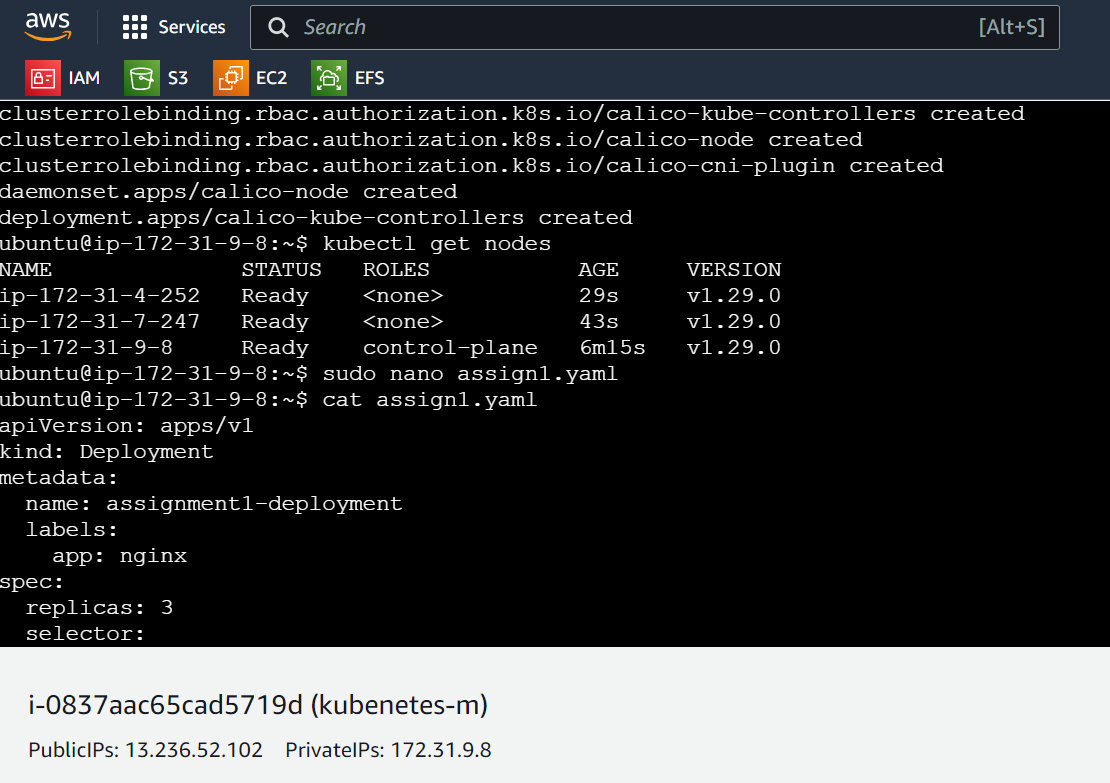
## Verify Cluster Connection

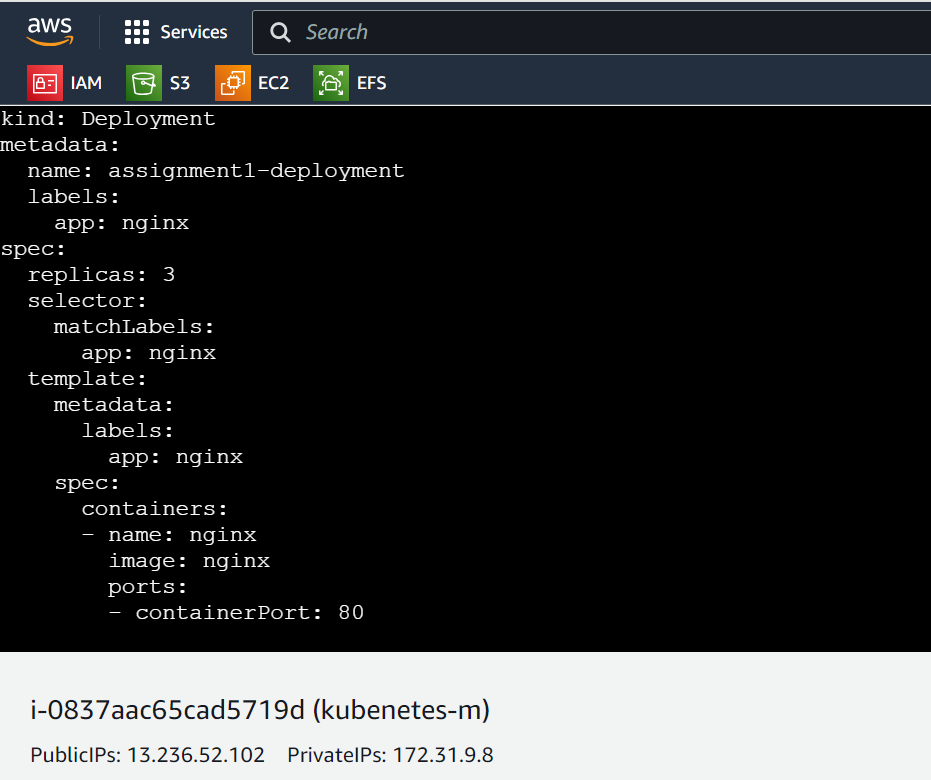
\*\*On Master Node:\*\*

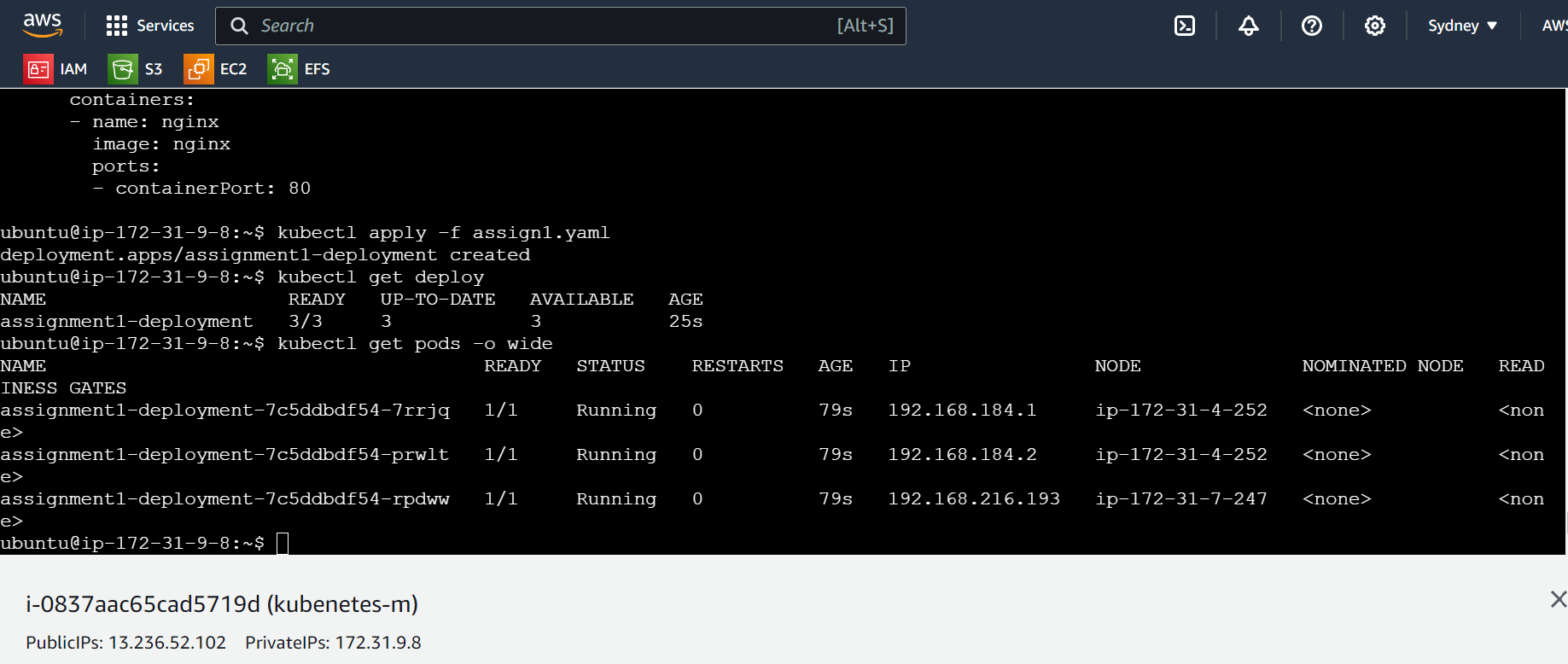
kubectl get nodes



● Create a nginx deployment of 3 replicas







1 sudo apt update

2 sudo nano k8.sh

3 bash k8.sh

4 sudo kubeadm config image pull

5 sudo kubeadm config images pull

6 sudo kubeadm init

7 kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.26.0/manifests/calico.yaml

8 kubectl get nodes

9 sudo nano assign1.yaml

10 cat assign1.yaml

11 kubectl apply -f assign1.yaml

12 kubectl get deploy

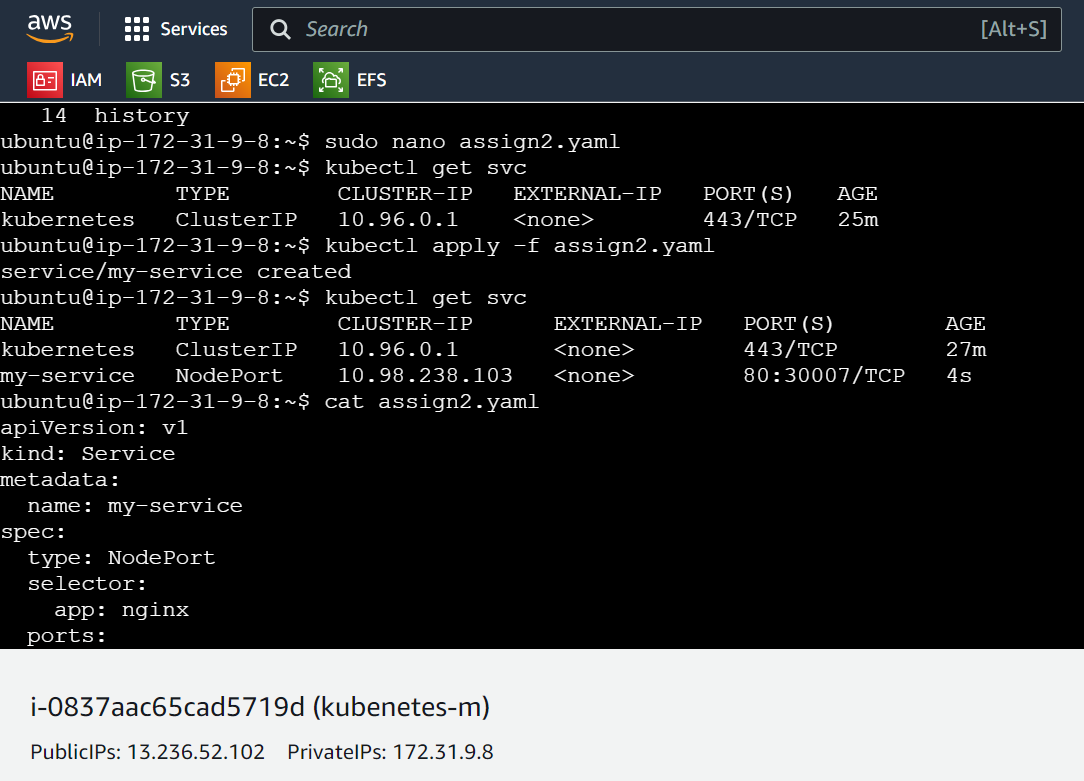
13 kubectl get pods -o wide

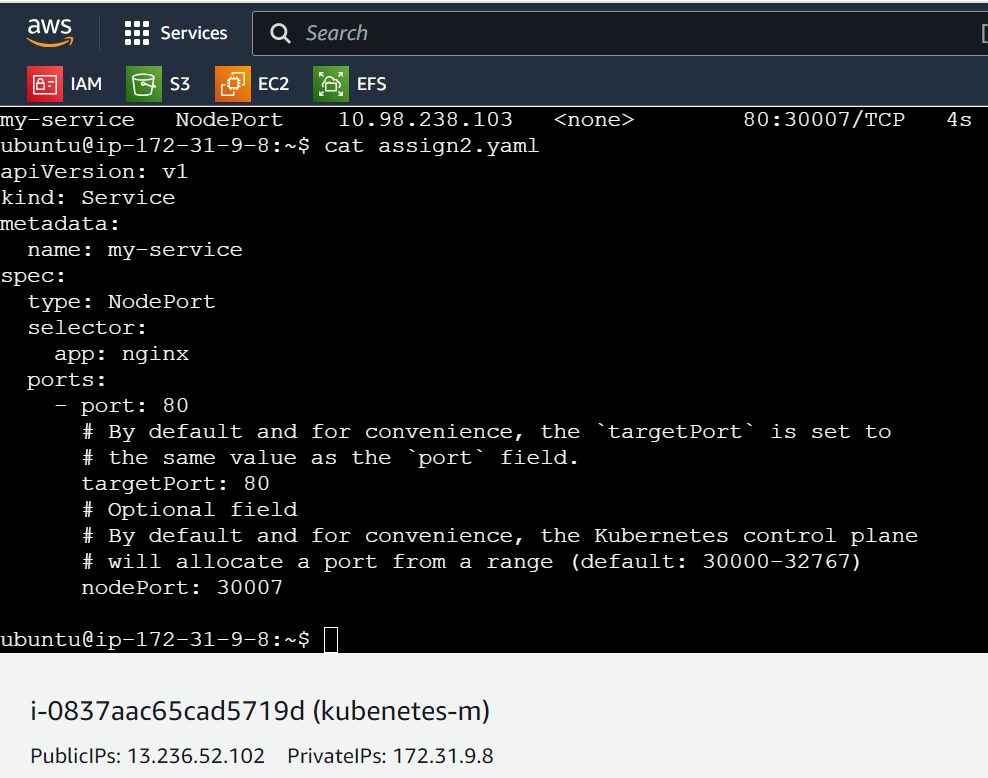
14 history

**Module-9: Kubernetes Assignment – 2**

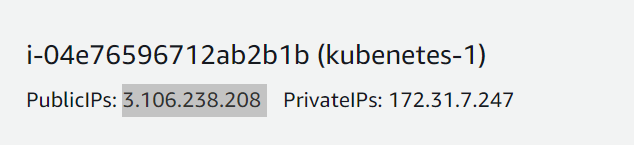
● Use the previous deployment

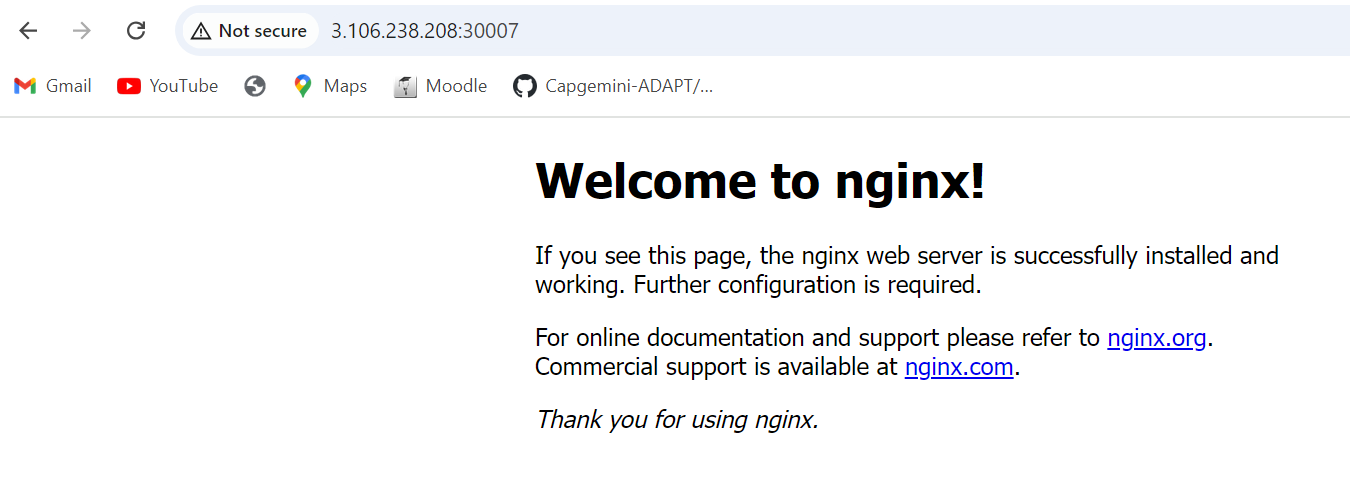
● Create a service of type NodePort for nginx deployment





● Check the nodeport service on a browser to verify





15 sudo nano assign2.yaml

16 kubectl get svc

17 kubectl apply -f assign2.yaml

18 kubectl get svc

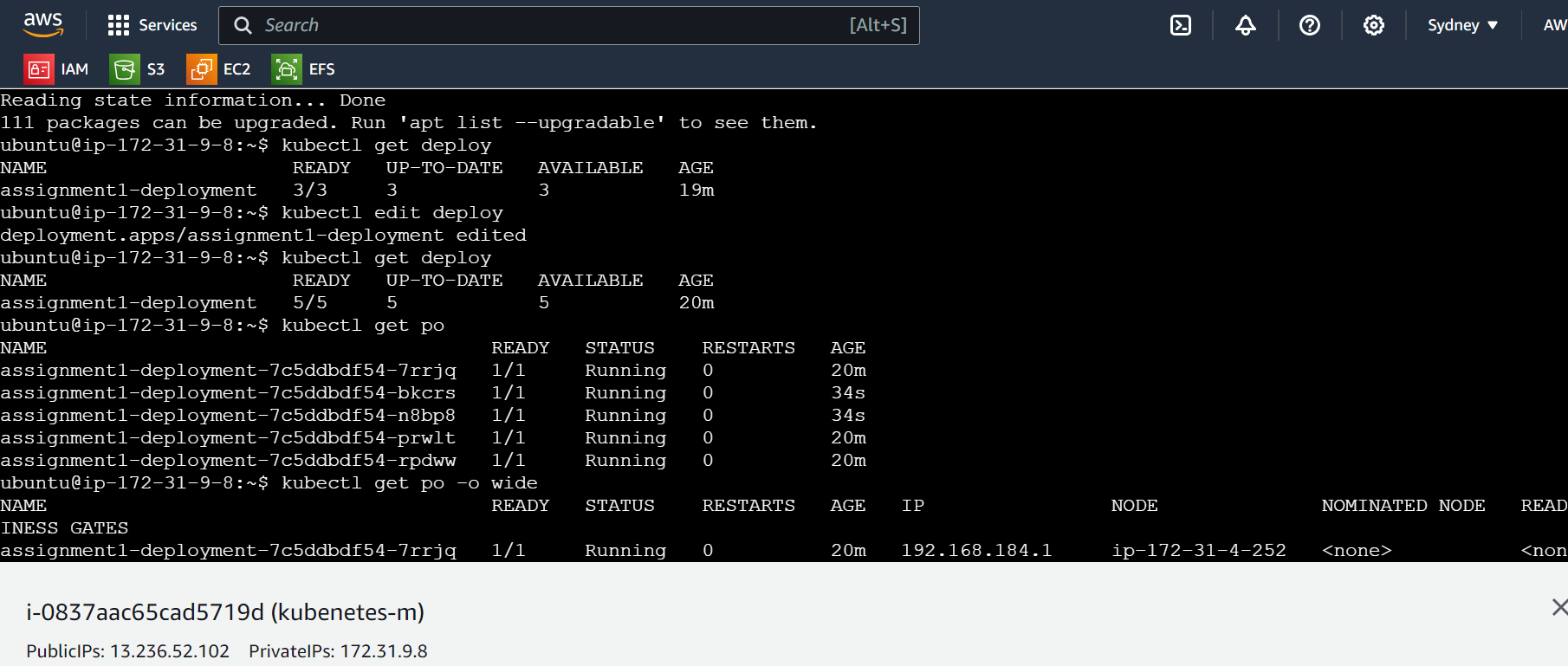
19 cat assign2.yaml

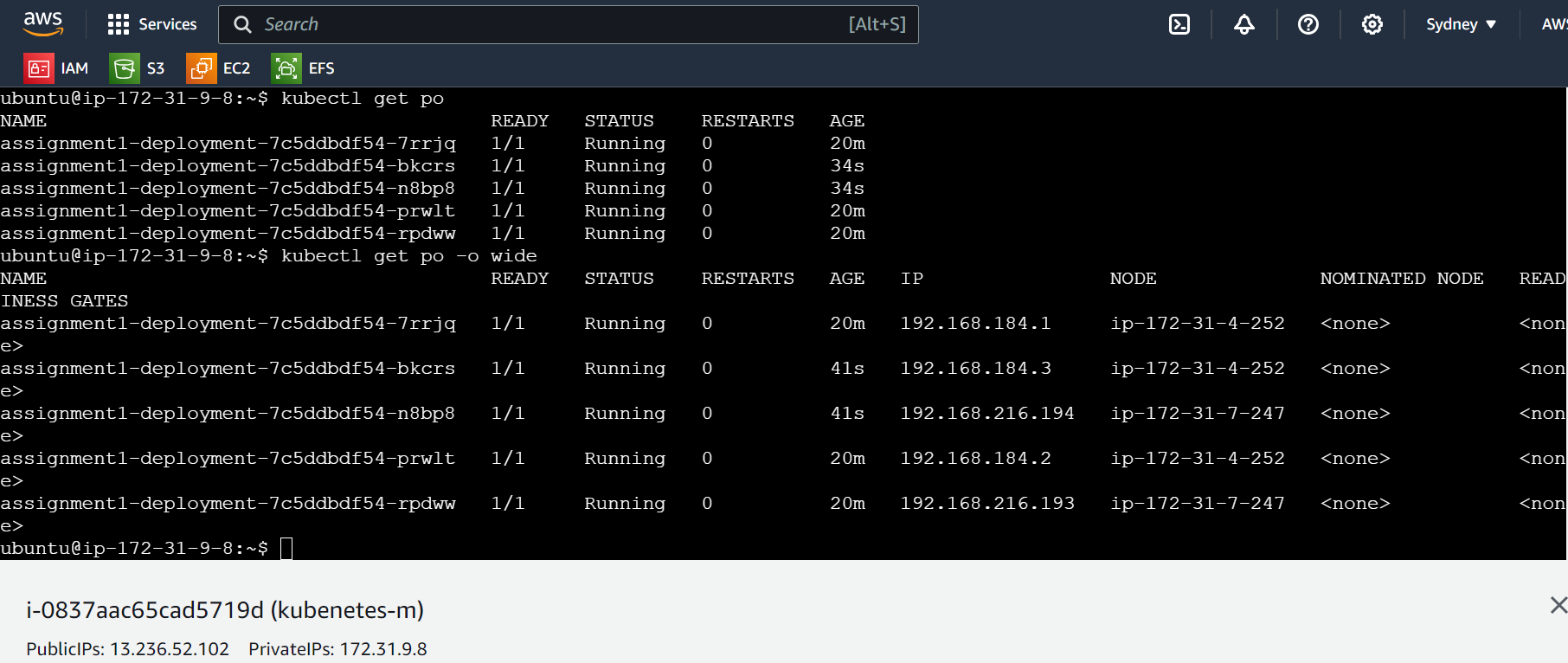
20 history

**Module-9: Kubernetes Assignment – 3**

● Use the previous deployment

● Change the replicas to 5 for the deployment





21 kubectl get deploy

22 kubectl edit deploy

23 sudo apt update

24 kubectl get deploy

25 kubectl edit deploy

26 kubectl get deploy

27 kubectl get po

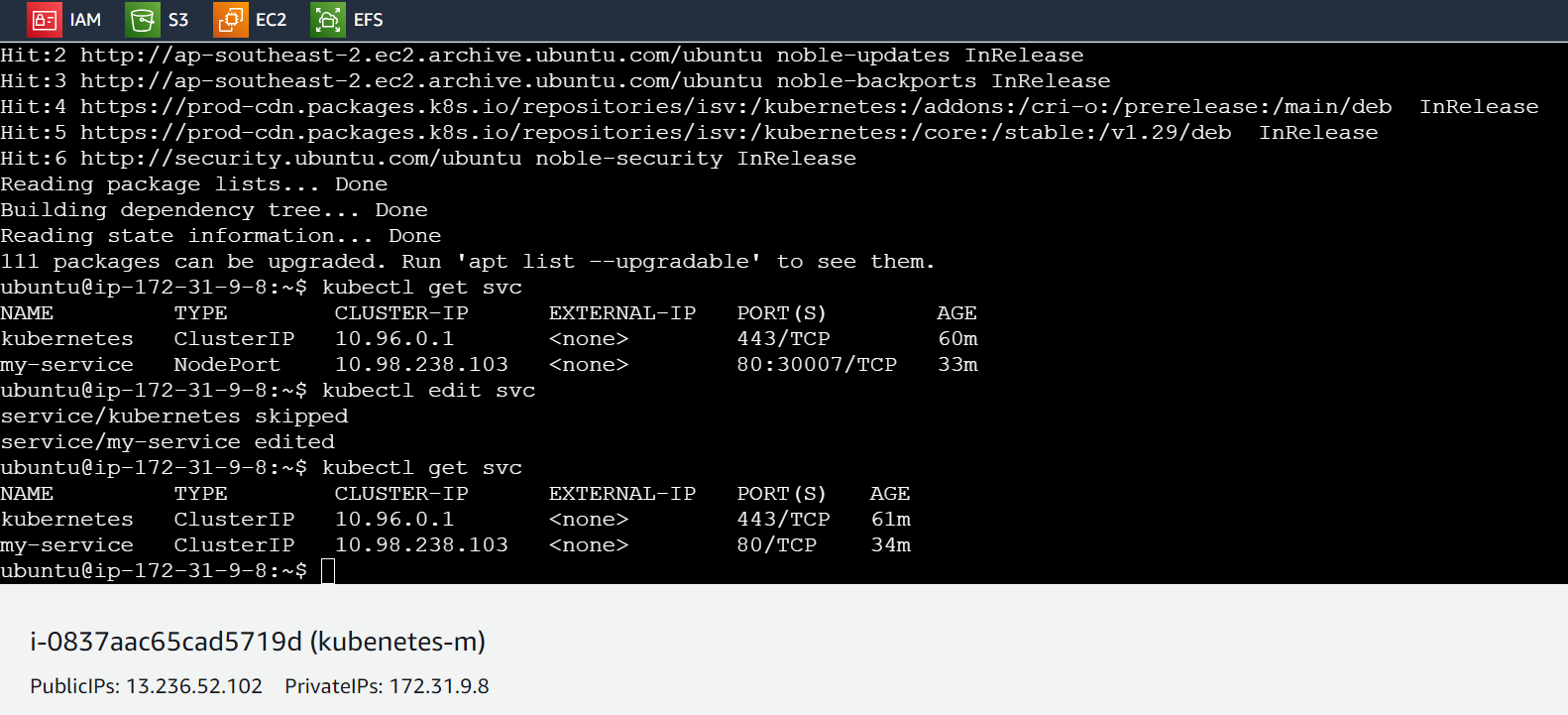
28 kubectl get po -o wide

29 history

**Module-9: Kubernetes Assignment – 4**

● Use the previous deployment

● Change the svc type to clusterIP



30 sudo apt update

31 kubectl get svc

32 kubectl edit svc

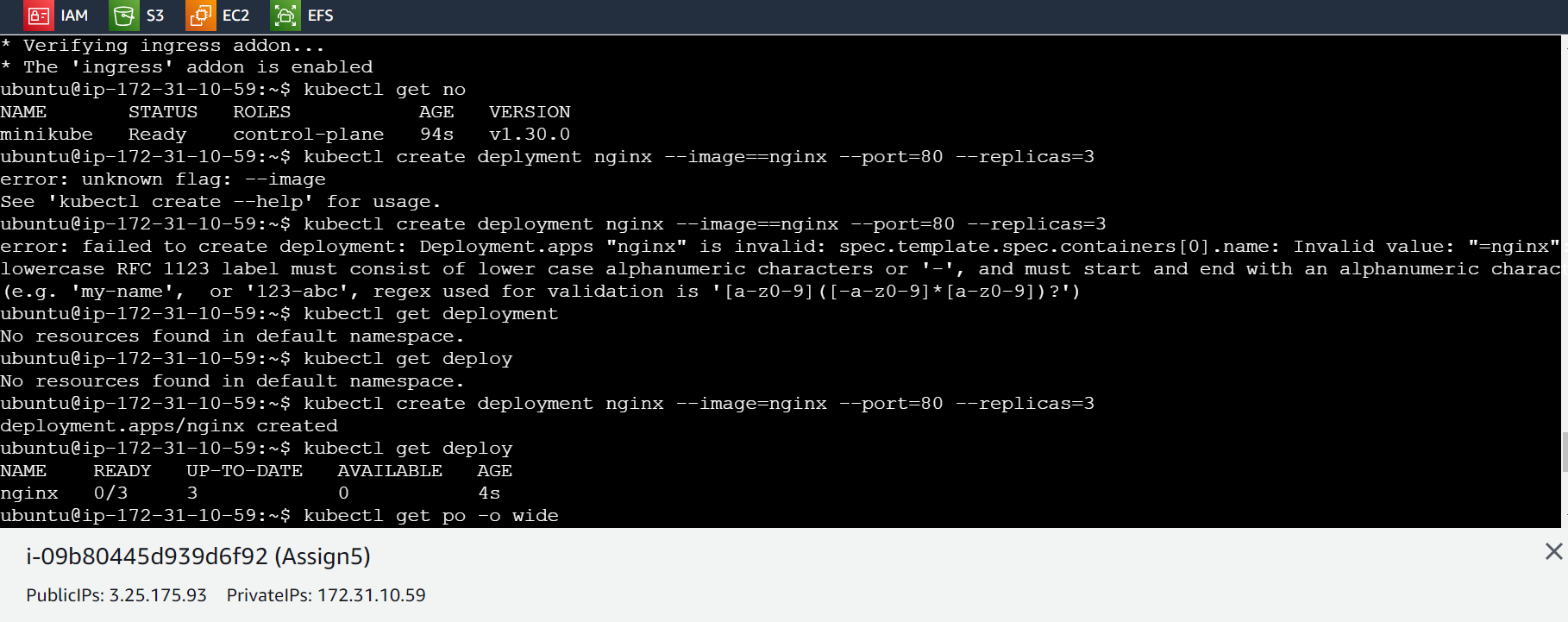
33 kubectl get svc

34 history

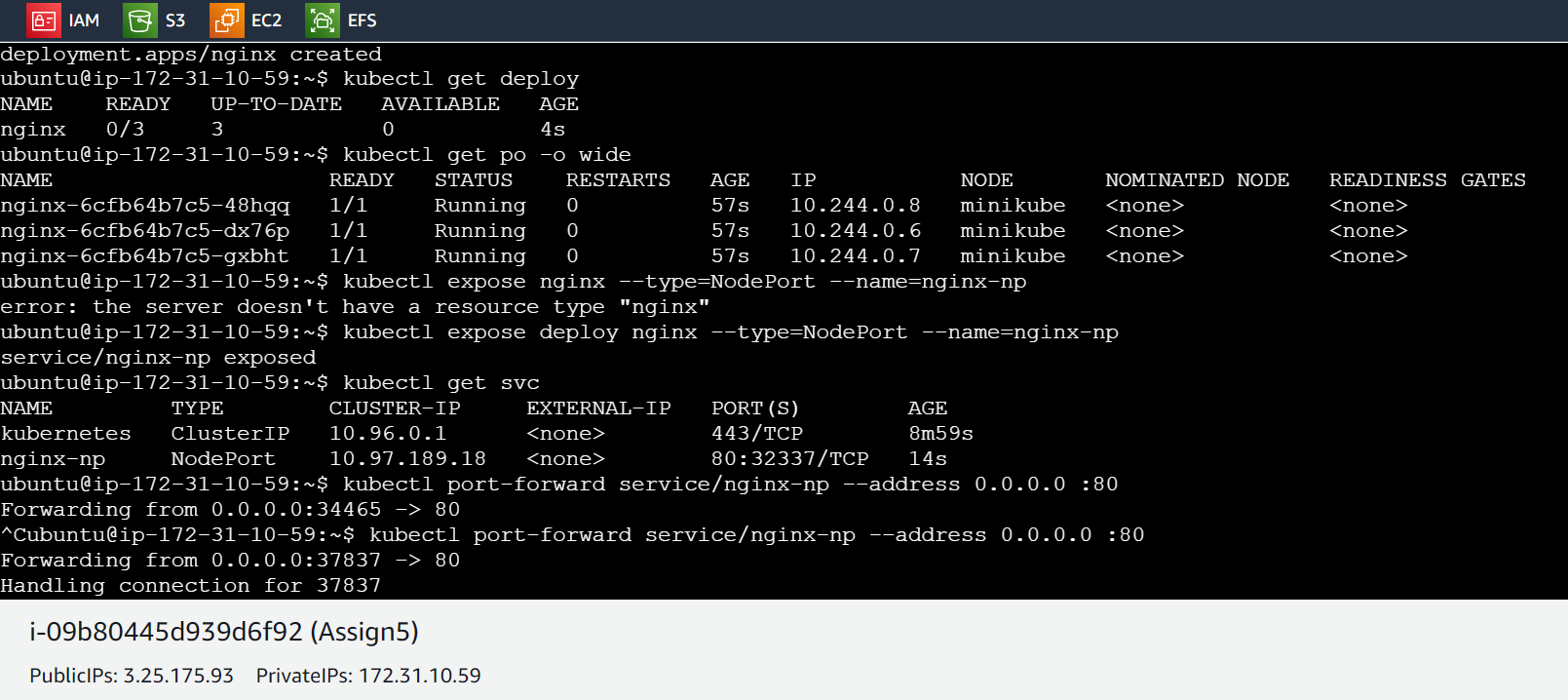
**Module-9: Kubernetes Assignment - 5**

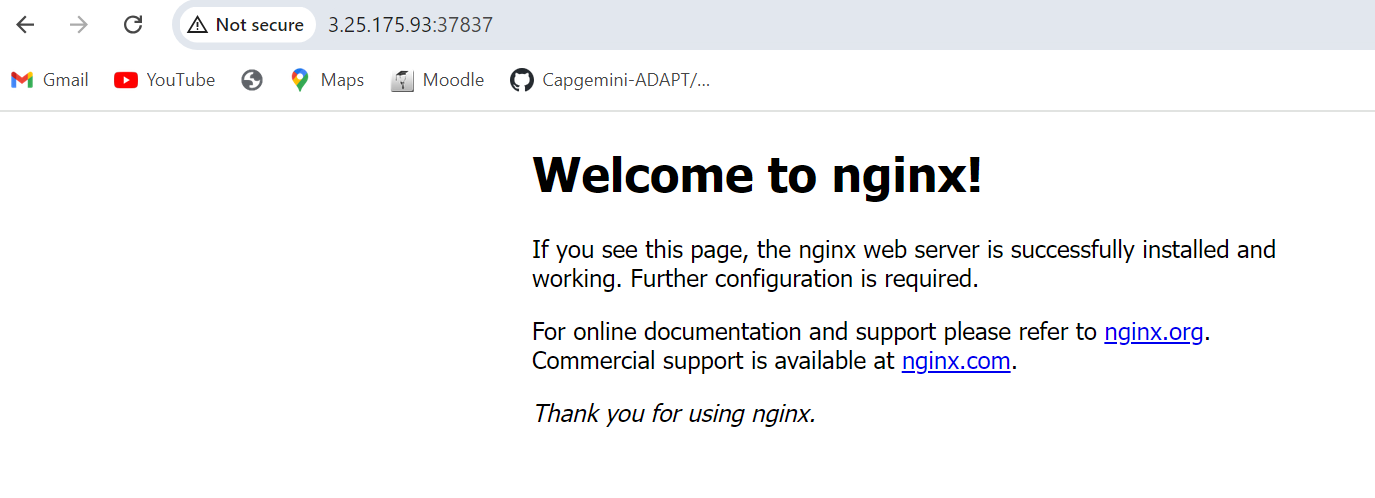
● Use the previous deployment

● Deploy an nginx deployment of 3 replicas

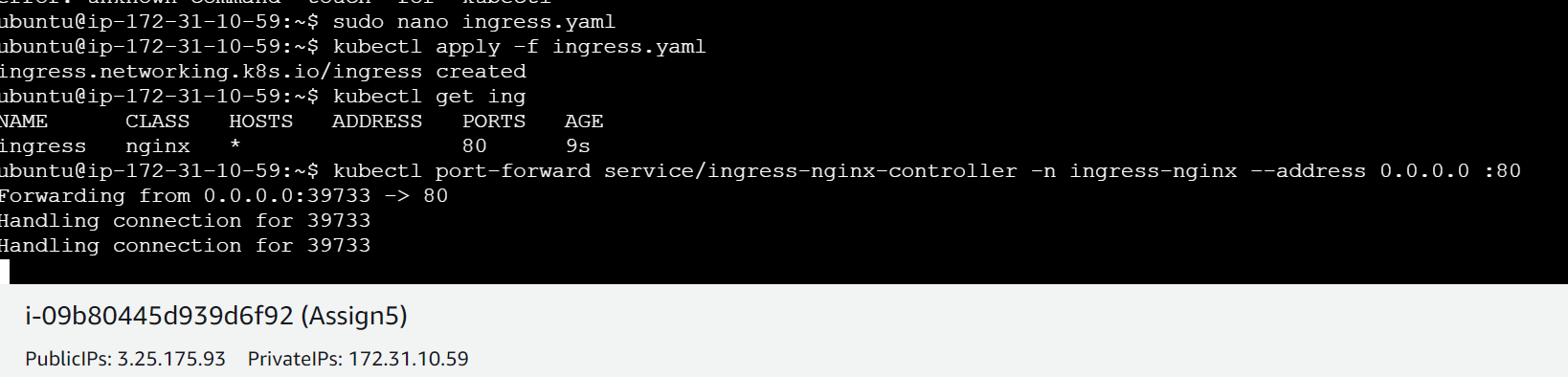


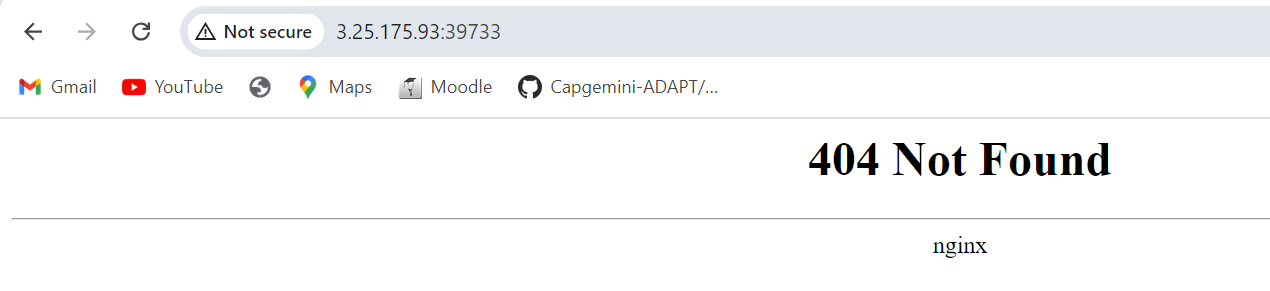
● Create an nginx service of type clusterip





● Create an ingress service /apache to apache service /nginx to nginx service





(Ingress is frozen i.e., we are unable to see service nginx)

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: ingress

annotations:

nginx.ingress.kubernetes.io/rewrite-target: /

spec:

ingressClassName: nginx

rules:

- http:

paths:

- path: /nginx

pathType: Prefix

backend:

service:

name: nginx

port:

number: 80

History-

17 sudo nano ingress.yaml

18 kubectl apply -f ingress.yaml

19 kubectl get ing

20 kubectl port-forward service/ingress-nginx-controller -n ingress-nginx --address 0.0.0.0 :80

21 history