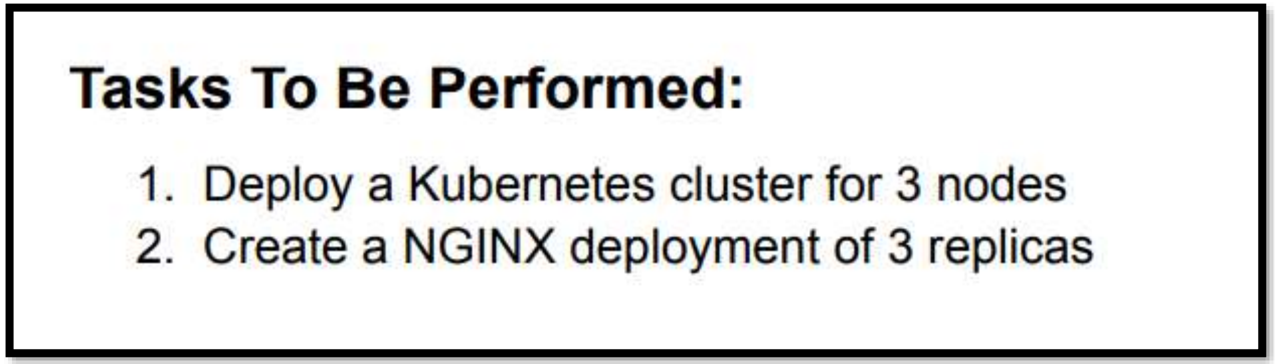
MODULE 7: KUBERNETES ASSIGNMENTS

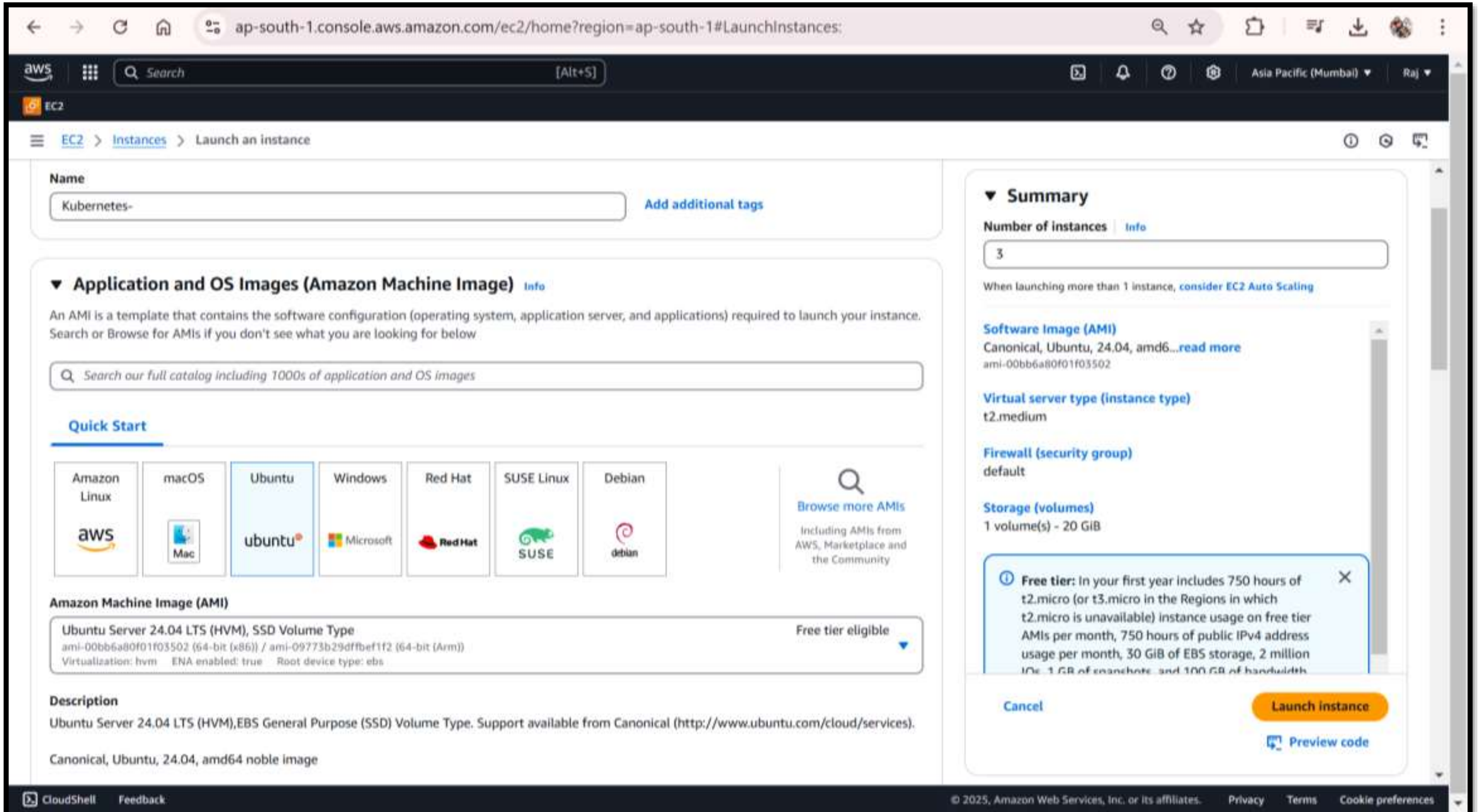
MODULE 7: KUBERNETES ASSIGNMENTS -1



Tasks To Be Performed:

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

Launching 3 EC2 instance for master, 2 slave machine


ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

EC2 > Instances > Launch an instance

Name

Kubernetes-

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-00bb6a80f01f03502 (64-bit (x86)) / ami-09773b29dffbef1f2 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Canonical, Ubuntu, 24.04, amd64 noble image

Summary

Number of instances Info

3

When launching more than 1 instance, consider EC2 Auto Scaling

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-00bb6a80f01f03502

Virtual server type (instance type)
t2.medium

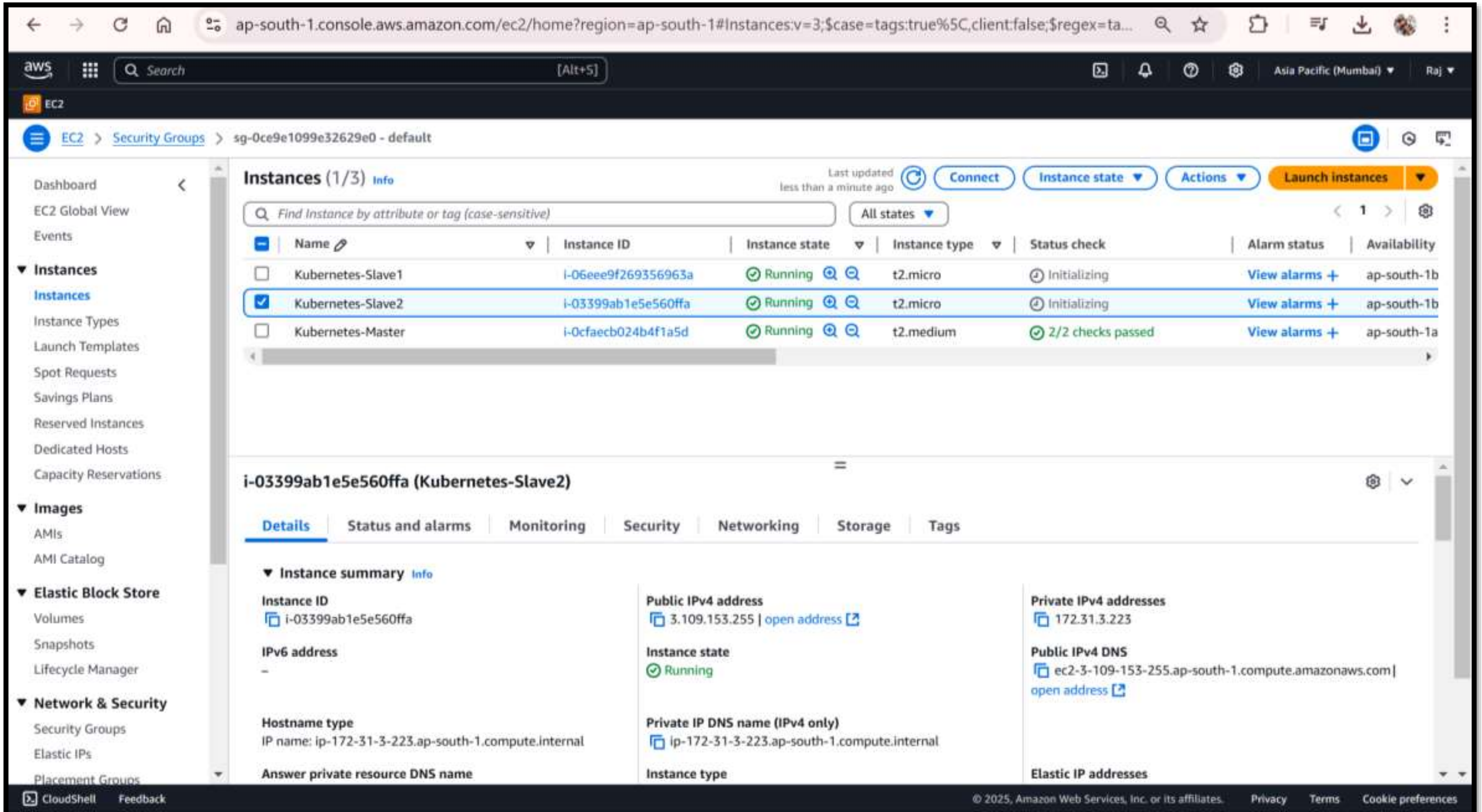
Firewall (security group)
default

Storage (volumes)
1 volume(s) - 20 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million

Cancel | Launch instance | Preview code

Launched 3 EC2 instance for master, 2 slave machine all running state



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:v=3;$case=tags:true%5C,client:false;$regex=ta...

EC2 > Security Groups > sg-0ce9e1099e32629e0 - default

Instances (1/3) Info

Last updated less than a minute ago

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
☐	Kubernetes-Slave1	i-06eee9f269356963a	Running	t2.micro	Initializing	View alarms +	ap-south-1b
☑	Kubernetes-Slave2	i-03399ab1e5e560ffa	Running	t2.micro	Initializing	View alarms +	ap-south-1b
☐	Kubernetes-Master	i-0cfaecb024b4f1a5d	Running	t2.medium	2/2 checks passed	View alarms +	ap-south-1a

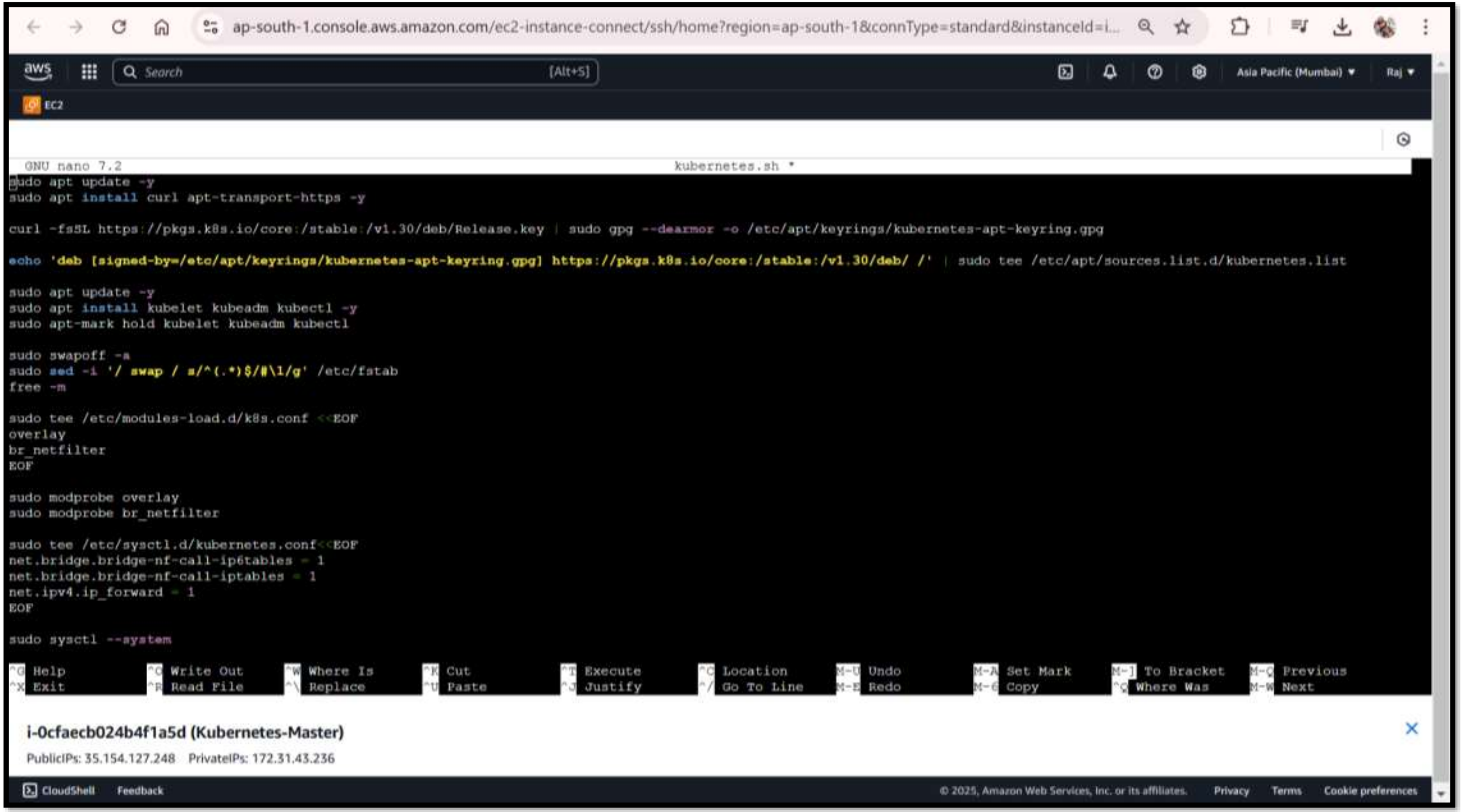
i-03399ab1e5e560ffa (Kubernetes-Slave2)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary Info

Instance ID: i-03399ab1e5e560ffa	Public IPv4 address: 3.109.153.255 | open address	Private IPv4 addresses: 172.31.3.223
IPv6 address: –	Instance state: Running	Public IPv4 DNS: ec2-3-109-153-255.ap-south-1.compute.amazonaws.com | open address
Hostname type: IP name: ip-172-31-3-223.ap-south-1.compute.internal	Private IP DNS name (IPv4 only): ip-172-31-3-223.ap-south-1.compute.internal	
Answer private resource DNS name	Instance type	Elastic IP addresses

Kubernetes installation script



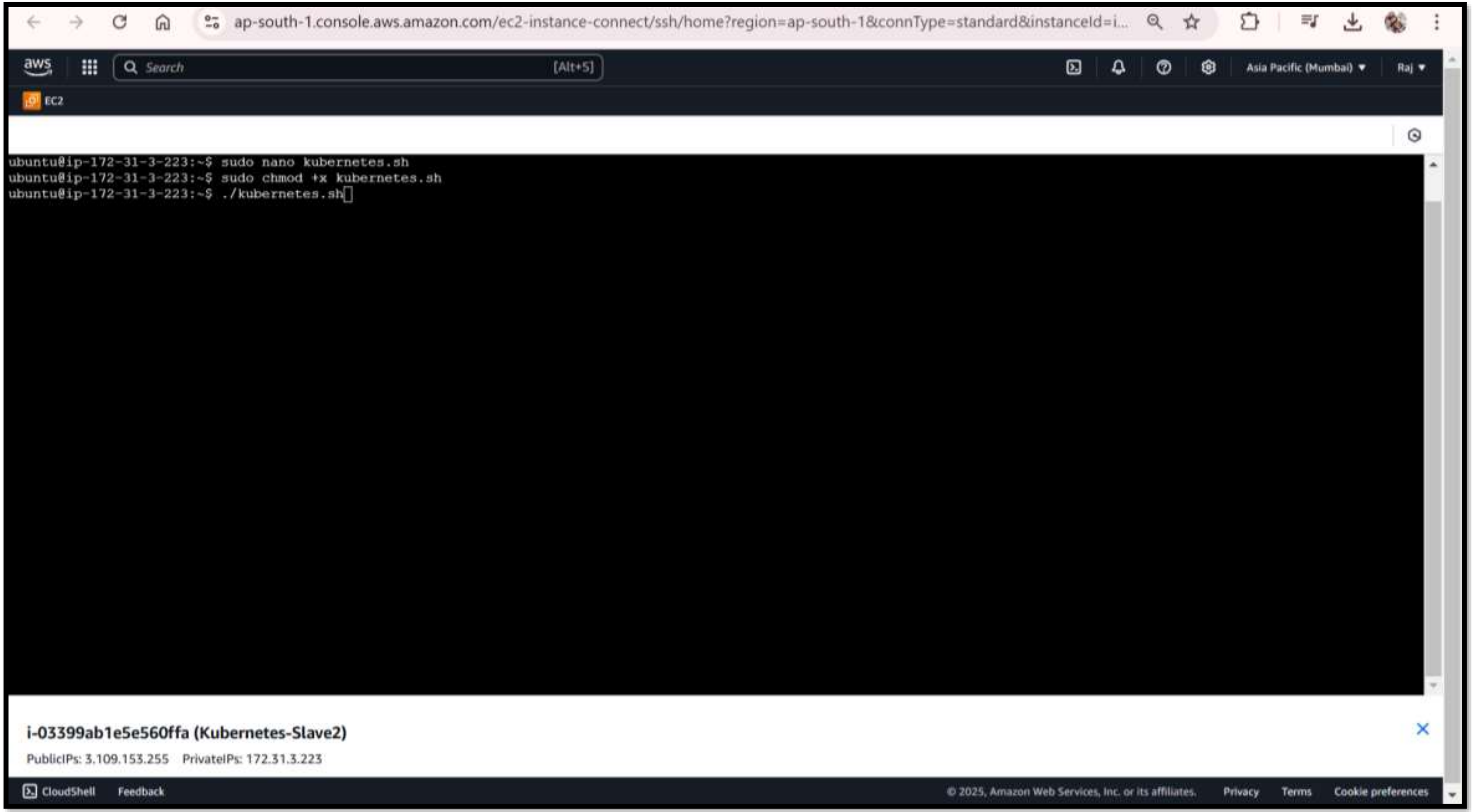
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i...

```
  GNU nano 7.2                                        kubernetes.sh *
sudo apt update -y
sudo apt install curl apt-transport-https -y

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.30/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:/v1.30/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt update -y
sudo apt install kubelet kubeadm kubectl -y
sudo apt-mark hold kubelet kubeadm kubectl

sudo swapoff -a
sudo sed -i '/ swap / s/^(.*)$/#\1/g' /etc/fstab
free -m

sudo tee /etc/modules-load.d/k8s.conf <<EOF
overlay
br_netfilter
EOF

sudo modprobe overlay
sudo modprobe br_netfilter

sudo tee /etc/sysctl.d/kubernetes.conf<<EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF

sudo sysctl --system

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket M-Q Previous
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^Q Where Was   M-W Next
```

i-0cfaecb024b4f1a5d (Kubernetes-Master)

PublicIPs: 35.154.127.248 PrivateIPs: 172.31.43.236

Running shell script for Kubernetes installation



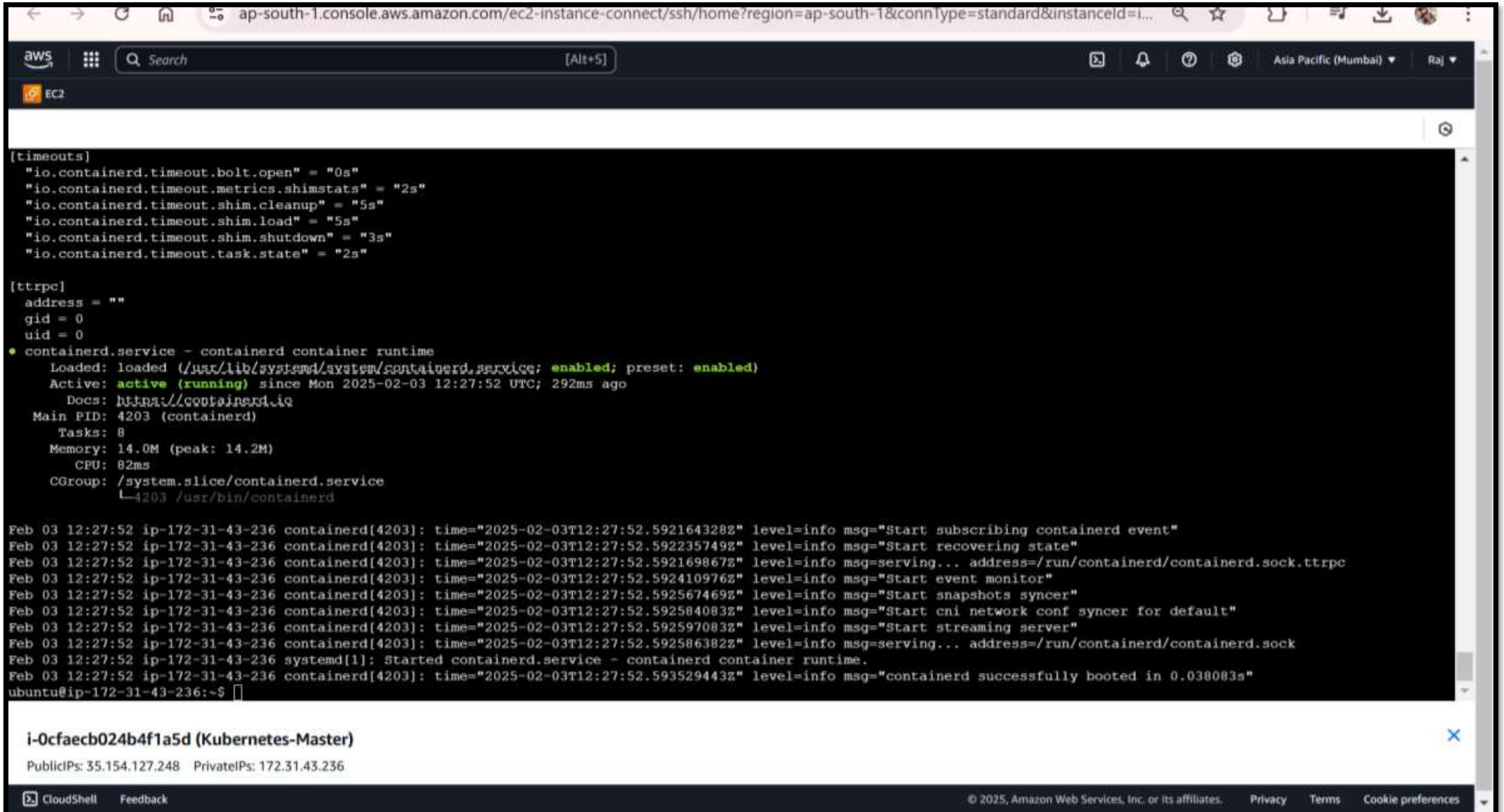
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i...

```
ubuntu@ip-172-31-3-223:~$ sudo nano kubernetes.sh
ubuntu@ip-172-31-3-223:~$ sudo chmod +x kubernetes.sh
ubuntu@ip-172-31-3-223:~$ ./kubernetes.sh
```

i-03399ab1e5e560ffa (Kubernetes-Slave2)

PublicIPs: 3.109.153.255 PrivateIPs: 172.31.3.223

Kubernetes installation completed



```
[timeouts]
  "io.containerd.timeout.bolt.open" = "0s"
  "io.containerd.timeout.metrics.shimstats" = "2s"
  "io.containerd.timeout.shim.cleanup" = "5s"
  "io.containerd.timeout.shim.load" = "5s"
  "io.containerd.timeout.shim.shutdown" = "3s"
  "io.containerd.timeout.task.state" = "2s"

[ttrpc]
  address = ""
  gid = 0
  uid = 0
● containerd.service - containerd container runtime
     Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-02-03 12:27:52 UTC; 292ms ago
       Docs: https://containerd.io
   Main PID: 4203 (containerd)
      Tasks: 8
     Memory: 14.0M (peak: 14.2M)
        CPU: 82ms
     CGroup: /system.slice/containerd.service
             └─4203 /usr/bin/containerd

Feb 03 12:27:52 ip-172-31-43-236 containerd[4203]: time="2025-02-03T12:27:52.592164328Z" level=info msg="Start subscribing containerd event"
Feb 03 12:27:52 ip-172-31-43-236 containerd[4203]: time="2025-02-03T12:27:52.592235749Z" level=info msg="Start recovering state"
Feb 03 12:27:52 ip-172-31-43-236 containerd[4203]: time="2025-02-03T12:27:52.592169867Z" level=info msg=serving... address=/run/containerd/containerd.sock.ttrpc
Feb 03 12:27:52 ip-172-31-43-236 containerd[4203]: time="2025-02-03T12:27:52.592410976Z" level=info msg="Start event monitor"
Feb 03 12:27:52 ip-172-31-43-236 containerd[4203]: time="2025-02-03T12:27:52.592567469Z" level=info msg="Start snapshots syncer"
Feb 03 12:27:52 ip-172-31-43-236 containerd[4203]: time="2025-02-03T12:27:52.592584083Z" level=info msg="Start cni network conf syncer for default"
Feb 03 12:27:52 ip-172-31-43-236 containerd[4203]: time="2025-02-03T12:27:52.592597083Z" level=info msg="Start streaming server"
Feb 03 12:27:52 ip-172-31-43-236 containerd[4203]: time="2025-02-03T12:27:52.592586382Z" level=info msg=serving... address=/run/containerd/containerd.sock
Feb 03 12:27:52 ip-172-31-43-236 systemd[1]: Started containerd.service - containerd container runtime.
Feb 03 12:27:52 ip-172-31-43-236 containerd[4203]: time="2025-02-03T12:27:52.593529443Z" level=info msg="containerd successfully booted in 0.038083s"
ubuntu@ip-172-31-43-236:~$ 
```

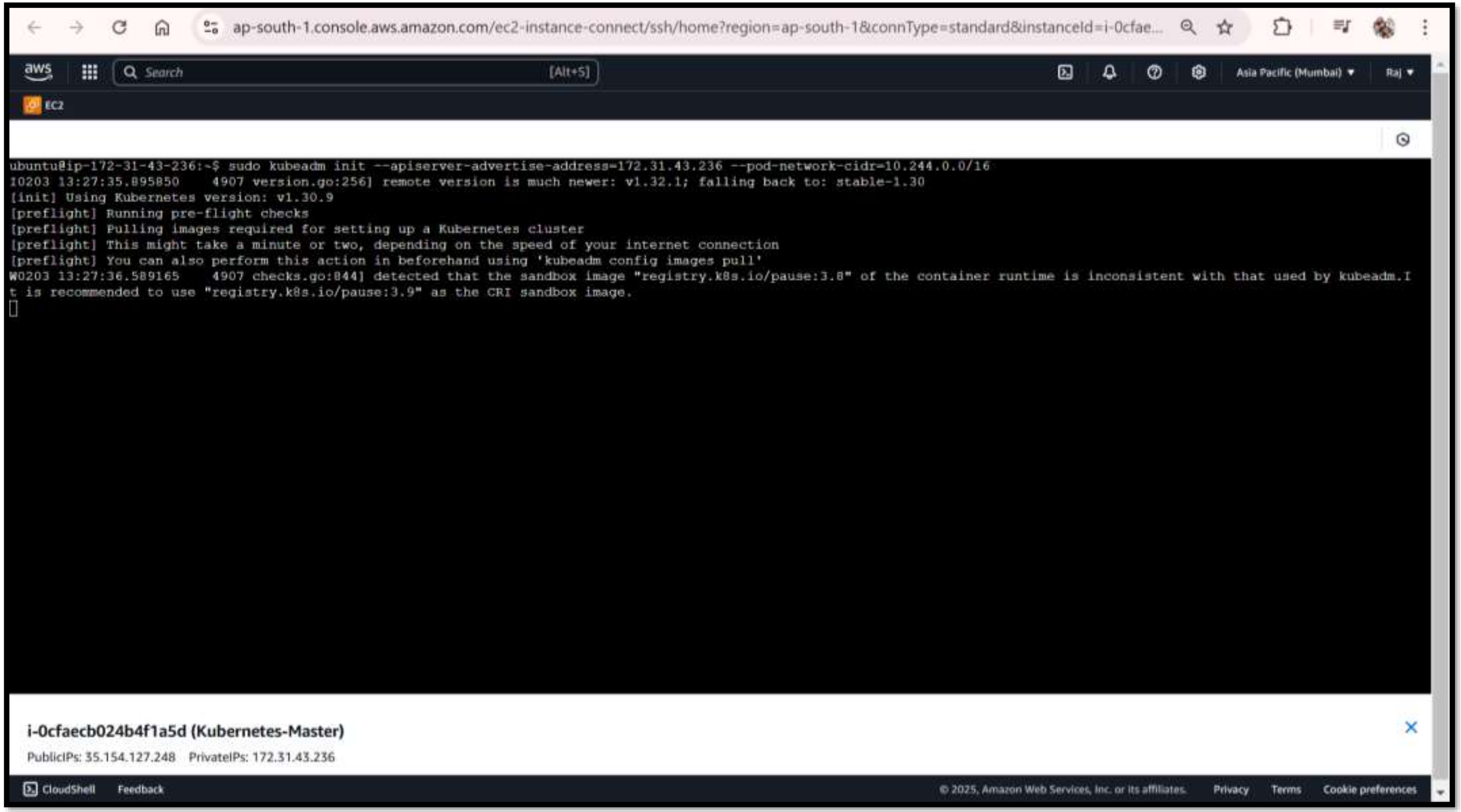
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i...
aws
Search
[Alt+S]
Asia Pacific (Mumbai) ▼
Raj ▼
EC2

i-0cfaecb024b4f1a5d (Kubernetes-Master)

PublicIPs: 35.154.127.248 PrivateIPs: 172.31.43.236

Installing a networking tool Kubernetes

```
kubeadm init --apiserver-advertise-address=172.31.43.236 --pod-network-cidr=10.244.0.0/16
```



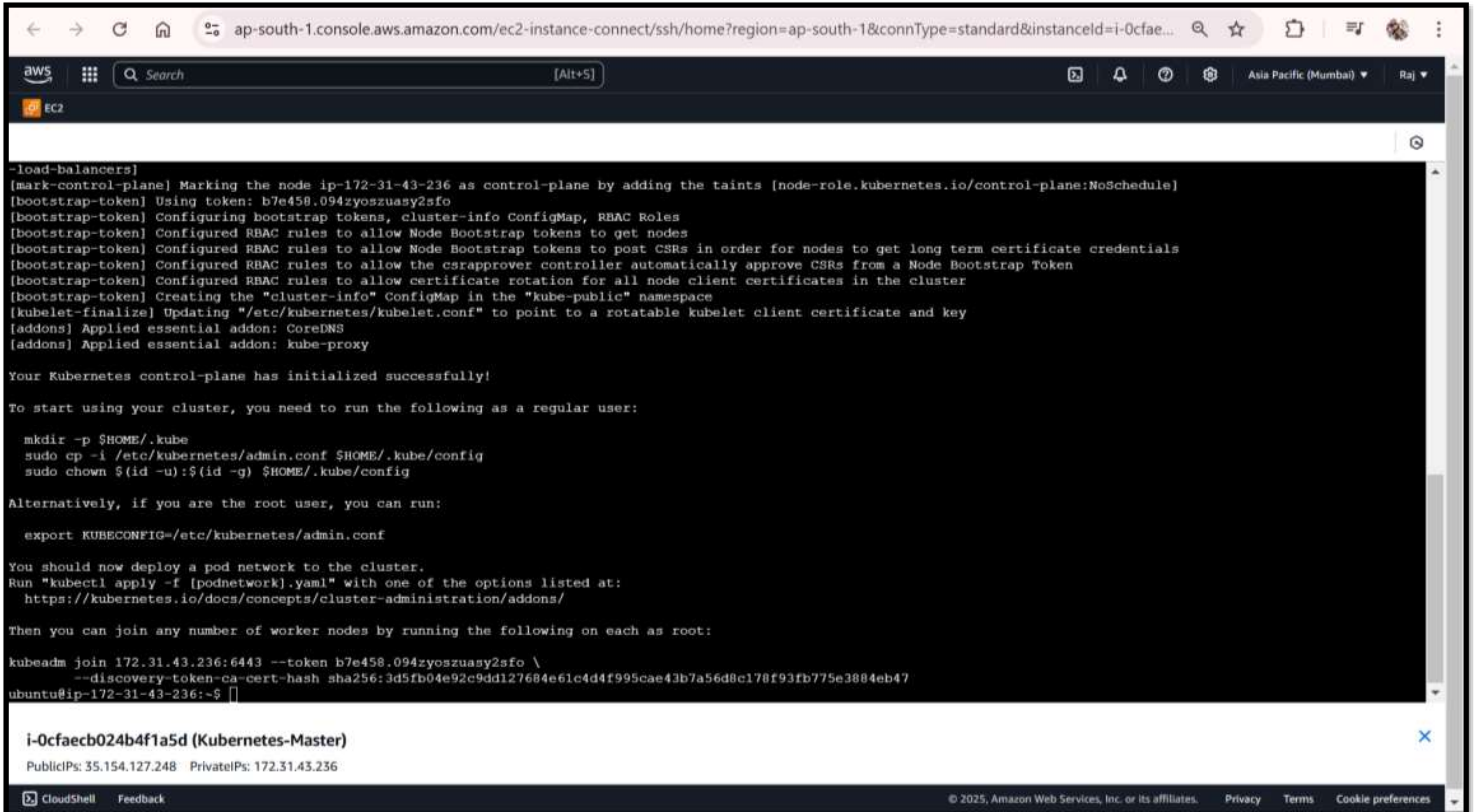
```
ubuntu@ip-172-31-43-236:~$ sudo kubeadm init --apiserver-advertise-address=172.31.43.236 --pod-network-cidr=10.244.0.0/16
I0203 13:27:35.895850    4907 version.go:256] remote version is much newer: v1.32.1; falling back to: stable-1.30
[init] Using Kubernetes version: v1.30.9
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0203 13:27:36.589165    4907 checks.go:844] detected that the sandbox image "registry.k8s.io/pause:3.8" of the container runtime is inconsistent with that used by kubeadm.I
t is recommended to use "registry.k8s.io/pause:3.9" as the CRI sandbox image.
```

i-0cfaecb024b4f1a5d (Kubernetes-Master)

PublicIPs: 35.154.127.248 PrivateIPs: 172.31.43.236

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-0cfae...
Search
[Alt+S]
Asia Pacific (Mumbai)
Raj
EC2
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

We are going to connect master node to slave node

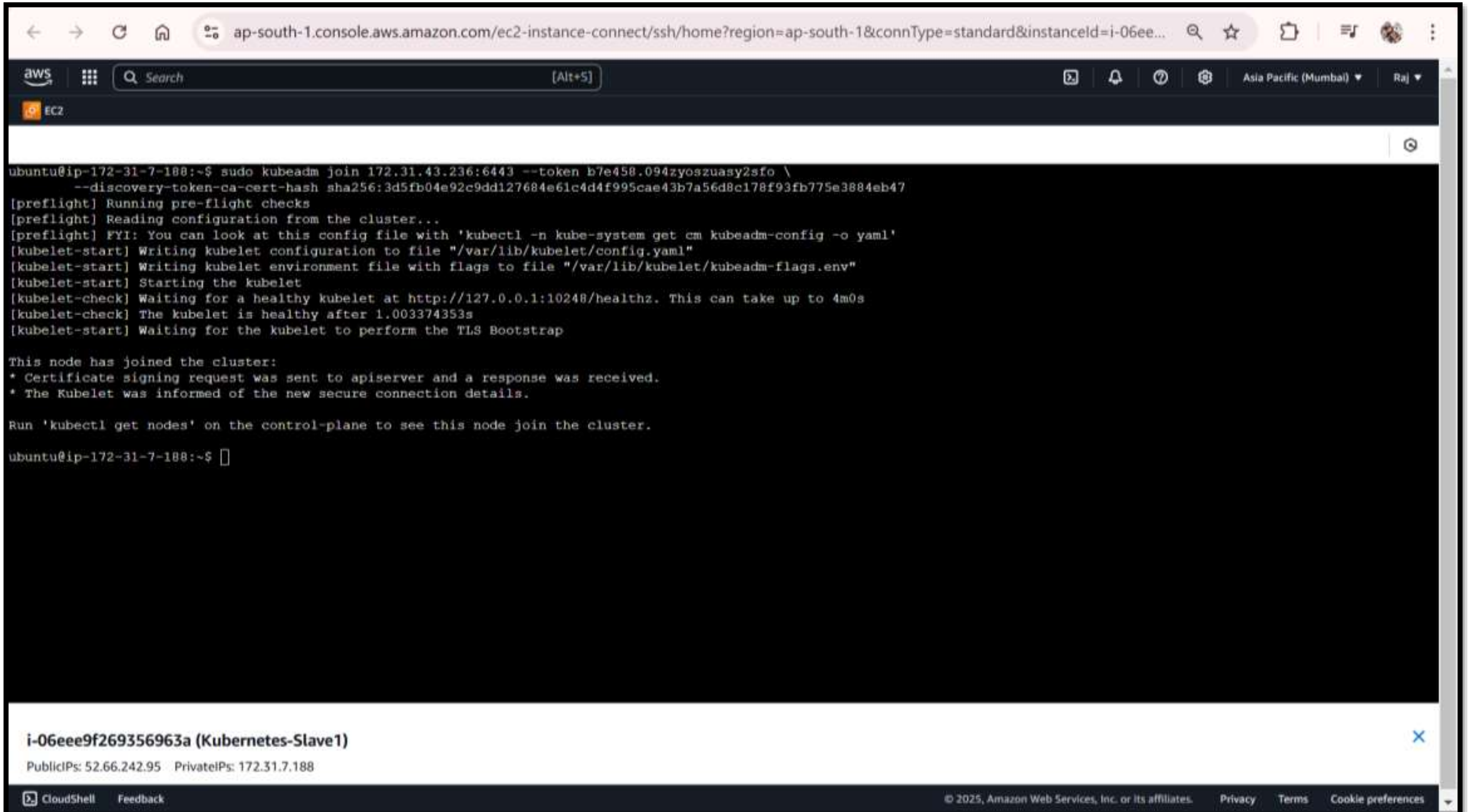

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-0cfae...

```
-load-balancers]
[mark-control-plane] Marking the node ip-172-31-43-236 as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: b7e458.094zyoszuasy2sfo
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[kubelet-finalize] Updating "/etc/kubernetes/kubelet.conf" to point to a rotatable kubelet client certificate and key
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular 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/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.43.236:6443 --token b7e458.094zyoszuasy2sfo \
        --discovery-token-ca-cert-hash sha256:3d5fb04e92c9dd127684e61c4d4f995cae43b7a56d8c178f93fb775e3884eb47
ubuntu@ip-172-31-43-236:~$ 
```

i-0cfaecb024b4f1a5d (Kubernetes-Master)

PublicIPs: 35.154.127.248   PrivateIPs: 172.31.43.236

Connecting salve1 node to master node



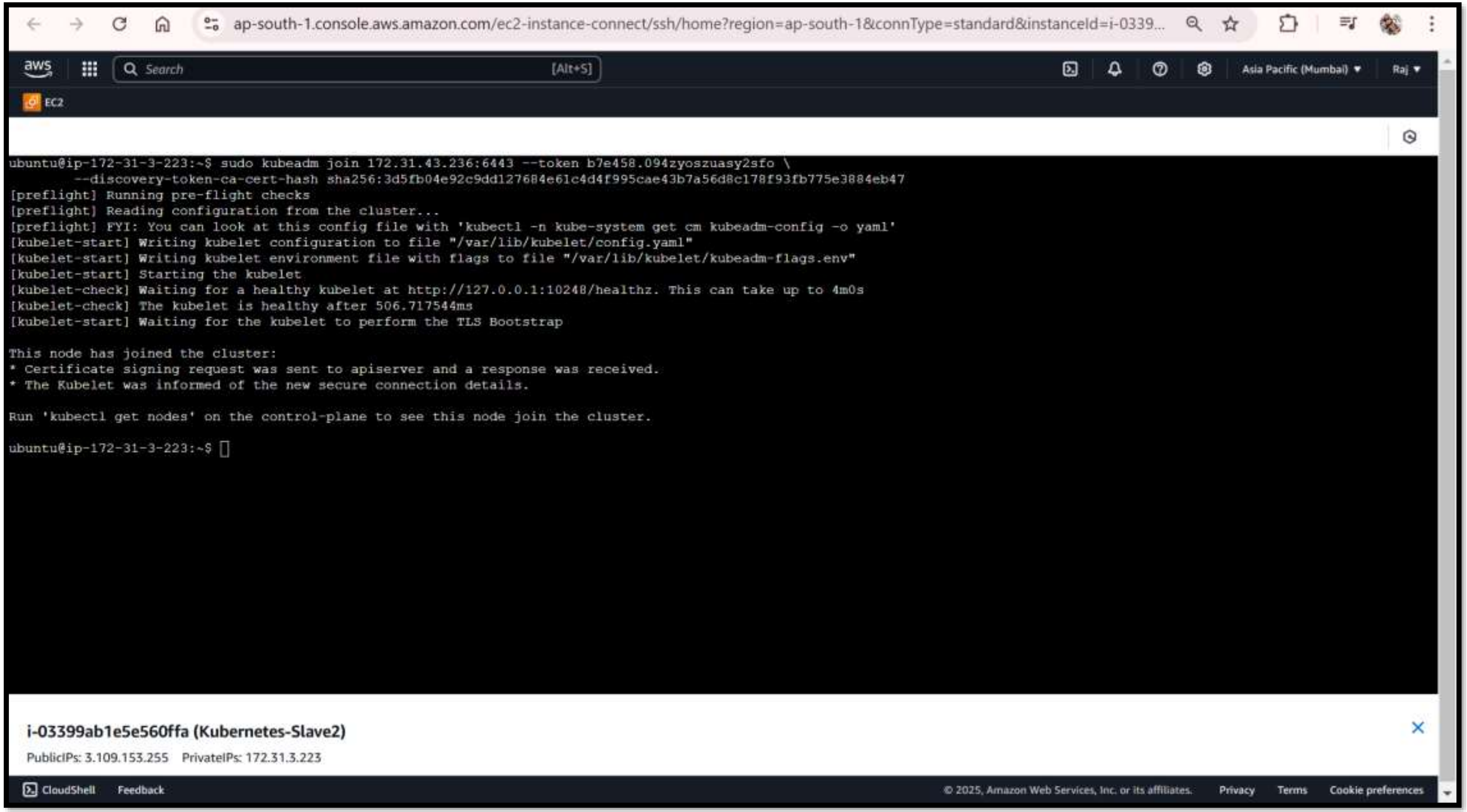
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-06ee...

```
ubuntu@ip-172-31-7-188:~$ sudo kubeadm join 172.31.43.236:6443 --token b7e458.094zyoszuasy2sfo \
        --discovery-token-ca-cert-hash sha256:3d5fb04e92c9dd127684e61c4d4f995cae43b7a56d8c178f93fb775e3884eb47
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 1.003374353s
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@ip-172-31-7-188:~$ 
```

i-06eee9f269356963a (Kubernetes-Slave1)

PublicIPs: 52.66.242.95 PrivateIPs: 172.31.7.188

Connecting salve2 node to master node



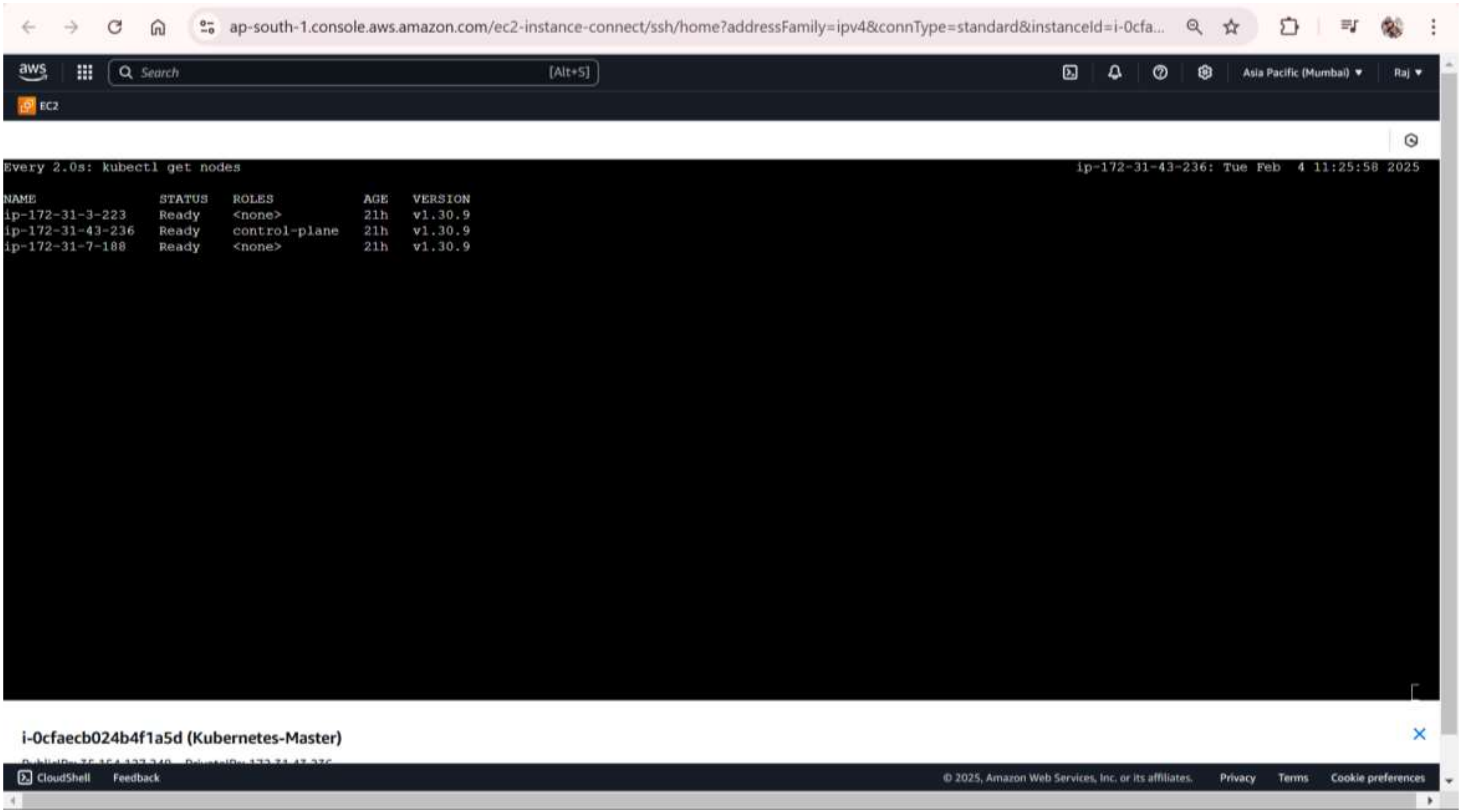
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-0339...

```
ubuntu@ip-172-31-3-223:~$ sudo kubeadm join 172.31.43.236:6443 --token b7e458.094zyoszuasy2sfo \
        --discovery-token-ca-cert-hash sha256:3d5fb04e92c9dd127684e61c4d4f995cae43b7a56d8c178f93fb775e3884eb47
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-check] Waiting for a healthy kubelet at http://127.0.0.1:10248/healthz. This can take up to 4m0s
[kubelet-check] The kubelet is healthy after 506.717544ms
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

ubuntu@ip-172-31-3-223:~$ 
```

i-03399ab1e5e560ffa (Kubernetes-Slave2)

PublicIPs: 3.109.153.255 PrivateIPs: 172.31.3.223

Get list of all connect nodes


ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0cfa...
Search
[Alt+S]
Asia Pacific (Mumbai)
Raj
EC2
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

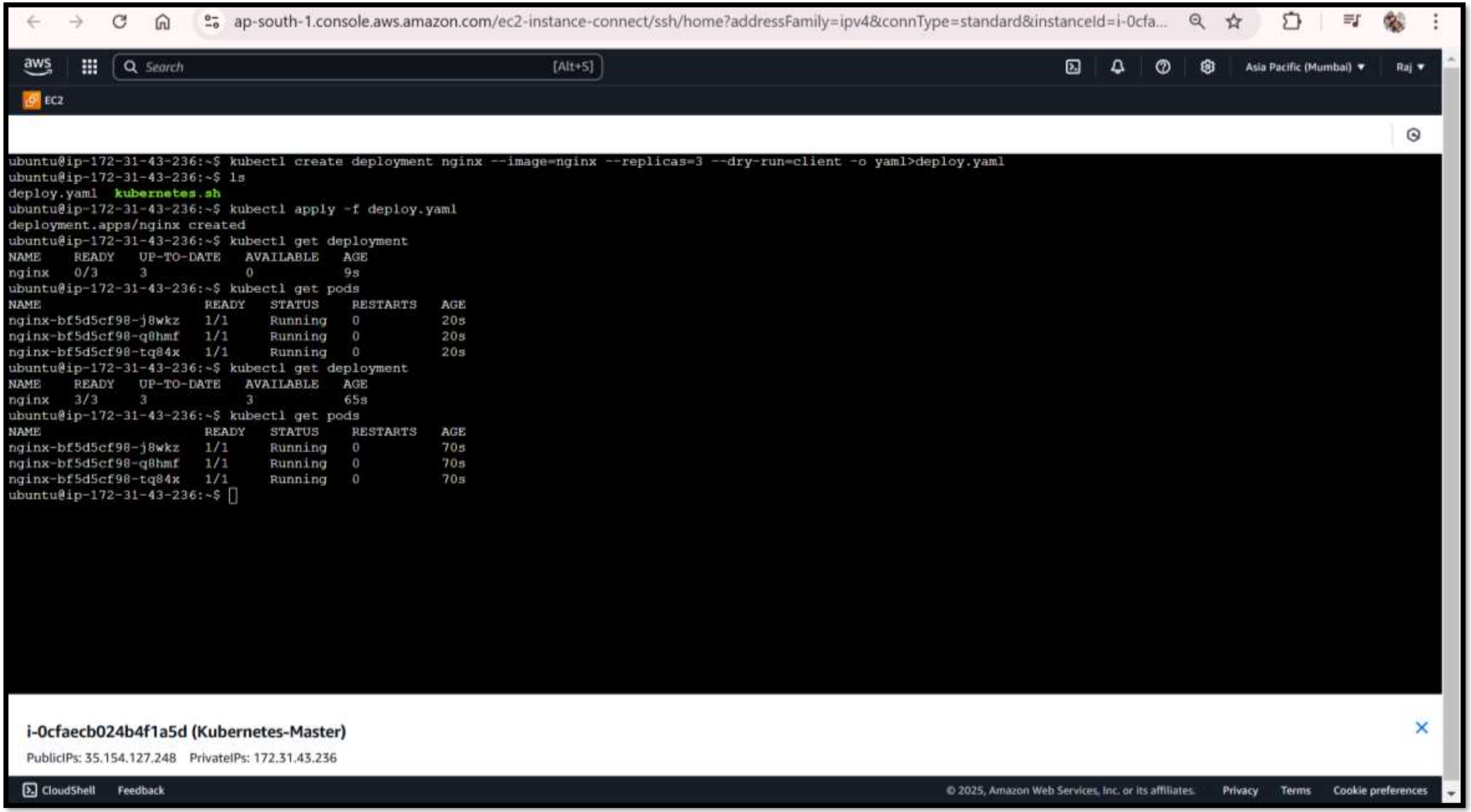
```
Every 2.0s: kubectl get nodes                                                    ip-172-31-43-236: Tue Feb  4 11:25:58 2025

NAME               STATUS   ROLES           AGE   VERSION
ip-172-31-3-223    Ready    <none>          21h   v1.30.9
ip-172-31-43-236   Ready    control-plane   21h   v1.30.9
ip-172-31-7-188    Ready    <none>          21h   v1.30.9
```

i-0cfaecb024b4f1a5d (Kubernetes-Master)

Nginx replica deployment commands

```
1. kubectl create deployment nginx --image=nginx --replicas=3 --dry-run=client -o
   yaml>deploy.yaml

2. kubectl apply -f deploy.yaml

3. kubectl get deployment

4. kubectl get pods
```

Nginx replica deployment command execution & Validation



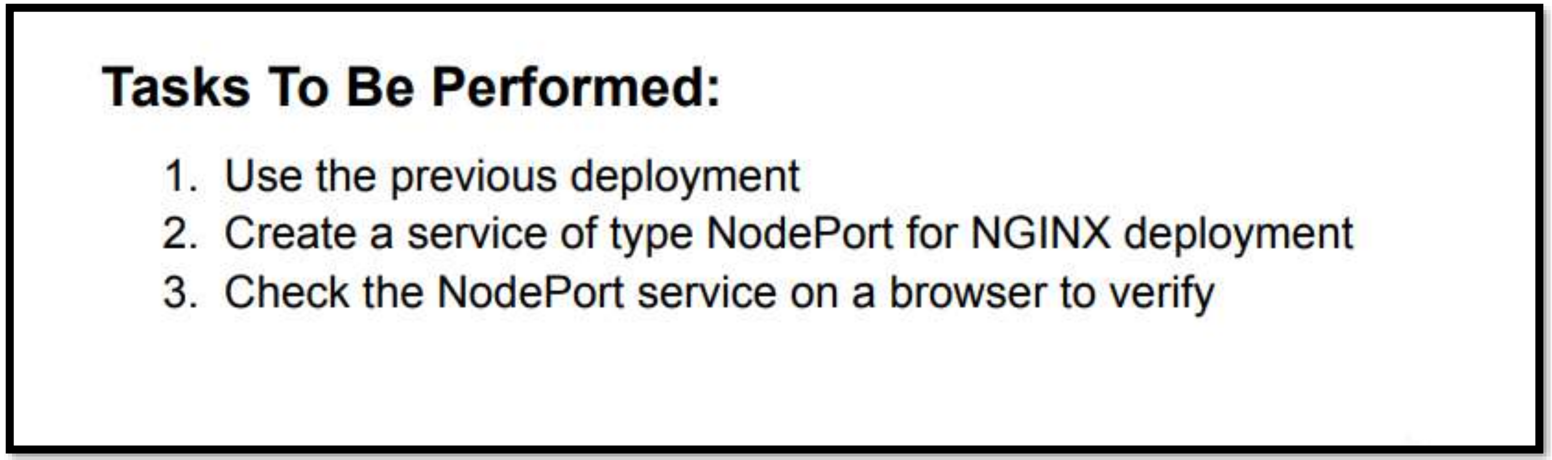
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0cfa...
Search
[Alt+S]
Asia Pacific (Mumbai)
Raj
EC2

```
ubuntu@ip-172-31-43-236:~$ kubectl create deployment nginx --image=nginx --replicas=3 --dry-run=client -o yaml>deploy.yaml
ubuntu@ip-172-31-43-236:~$ ls
deploy.yaml  kubernetes.sh
ubuntu@ip-172-31-43-236:~$ kubectl apply -f deploy.yaml
deployment.apps/nginx created
ubuntu@ip-172-31-43-236:~$ kubectl get deployment
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nginx   0/3     3            0           9s
ubuntu@ip-172-31-43-236:~$ kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
nginx-bf5d5cf98-j8wkz    1/1     Running   0          20s
nginx-bf5d5cf98-q8hmf    1/1     Running   0          20s
nginx-bf5d5cf98-tq84x    1/1     Running   0          20s
ubuntu@ip-172-31-43-236:~$ kubectl get deployment
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
nginx   3/3     3            3           65s
ubuntu@ip-172-31-43-236:~$ kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
nginx-bf5d5cf98-j8wkz    1/1     Running   0          70s
nginx-bf5d5cf98-q8hmf    1/1     Running   0          70s
nginx-bf5d5cf98-tq84x    1/1     Running   0          70s
ubuntu@ip-172-31-43-236:~$
```

i-0cfaecb024b4f1a5d (Kubernetes-Master)

PublicIPs: 35.154.127.248   PrivateIPs: 172.31.43.236

MODULE 7: KUBERNETES ASSIGNMENTS -2



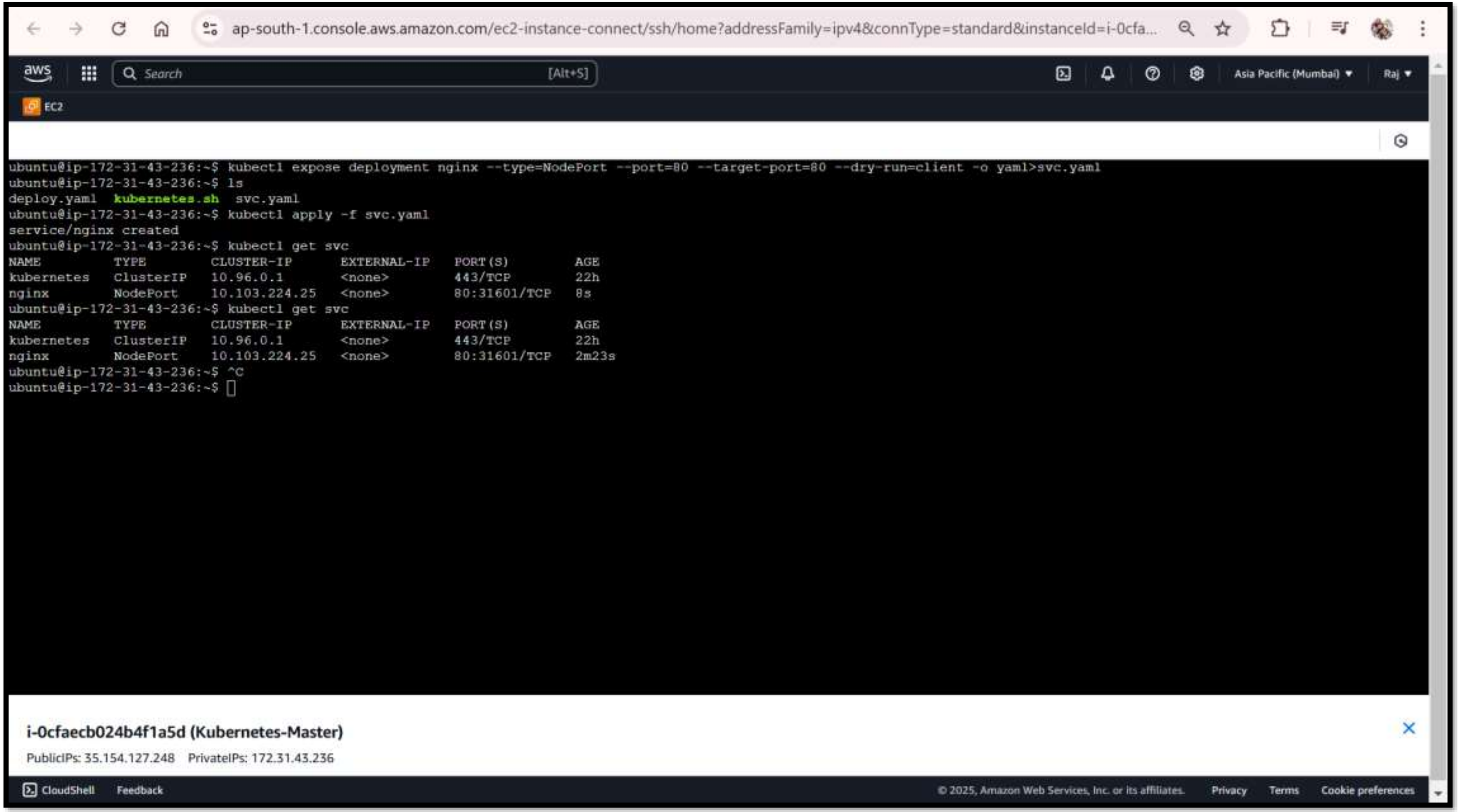
Tasks To Be Performed:

1. Use the previous deployment
2. Create a service of type NodePort for NGINX deployment
3. Check the NodePort service on a browser to verify

Node Port deployment commands

```
1. kubectl expose deployment nginx --type=NodePort --port=80 --target-port=80 -
   -dry-run=client -o yaml>svc.yaml

2. kubectl apply -f svc.yaml

3. kubectl get svc
```

Node Port deployment commands execution & validation



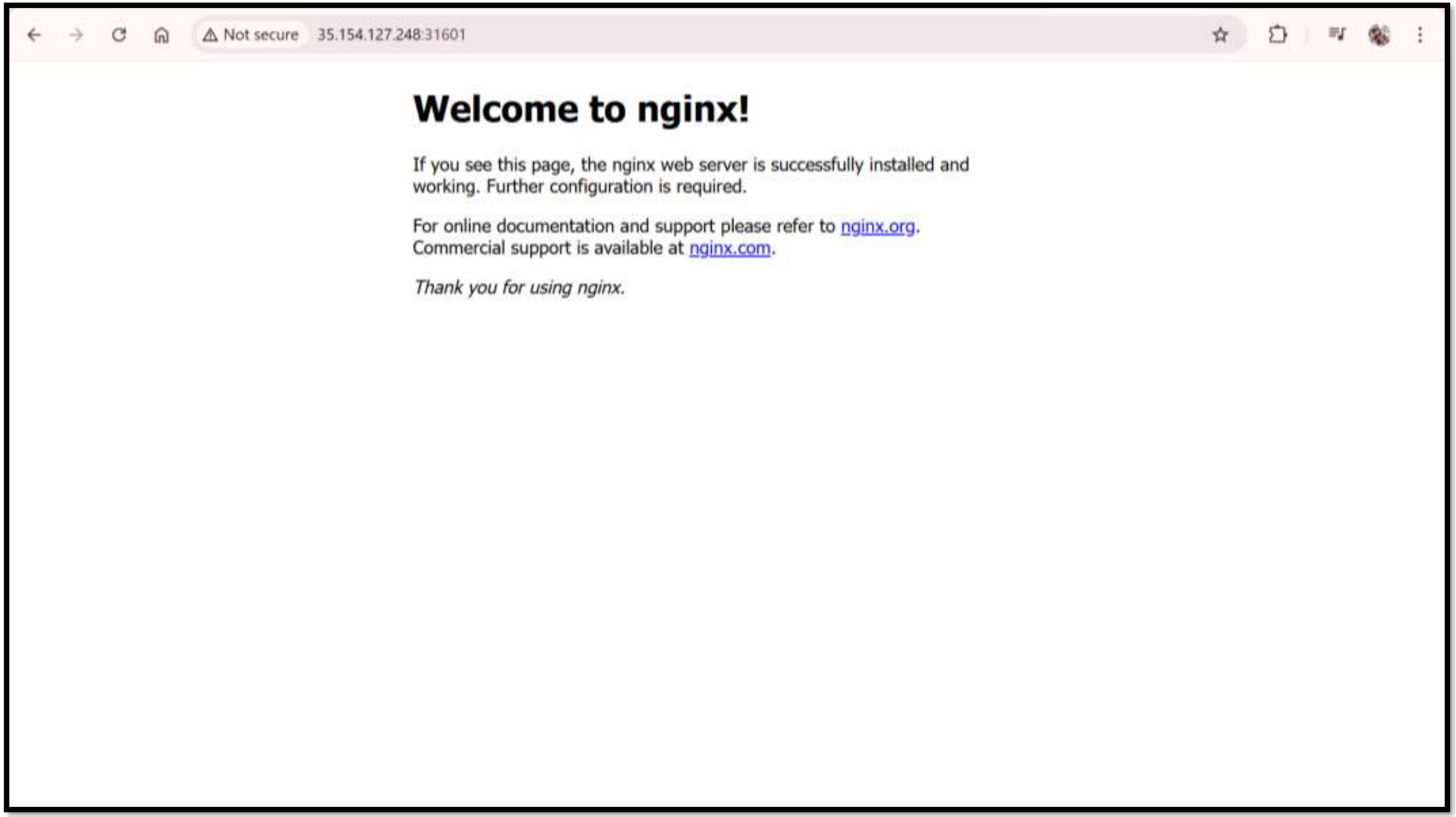
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0cfa...

```
ubuntu@ip-172-31-43-236:~$ kubectl expose deployment nginx --type=NodePort --port=80 --target-port=80 --dry-run=client -o yaml>svc.yaml
ubuntu@ip-172-31-43-236:~$ ls
deploy.yaml  kubernetes.sh  svc.yaml
ubuntu@ip-172-31-43-236:~$ kubectl apply -f svc.yaml
service/nginx created
ubuntu@ip-172-31-43-236:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP        22h
nginx        NodePort    10.103.224.25   <none>        80:31601/TCP   8s
ubuntu@ip-172-31-43-236:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP        22h
nginx        NodePort    10.103.224.25   <none>        80:31601/TCP   2m23s
ubuntu@ip-172-31-43-236:~$ ^C
ubuntu@ip-172-31-43-236:~$ 
```

i-0cfaecb024b4f1a5d (Kubernetes-Master)

PublicIPs: 35.154.127.248 PrivateIPs: 172.31.43.236

We can see the Nginx server running on port 31601



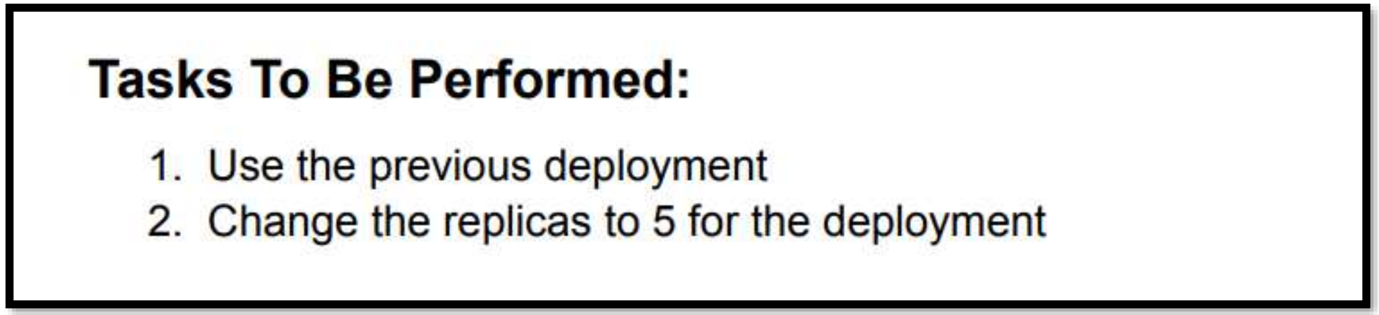
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

MODULE 7: KUBERNETES ASSIGNMENTS -3

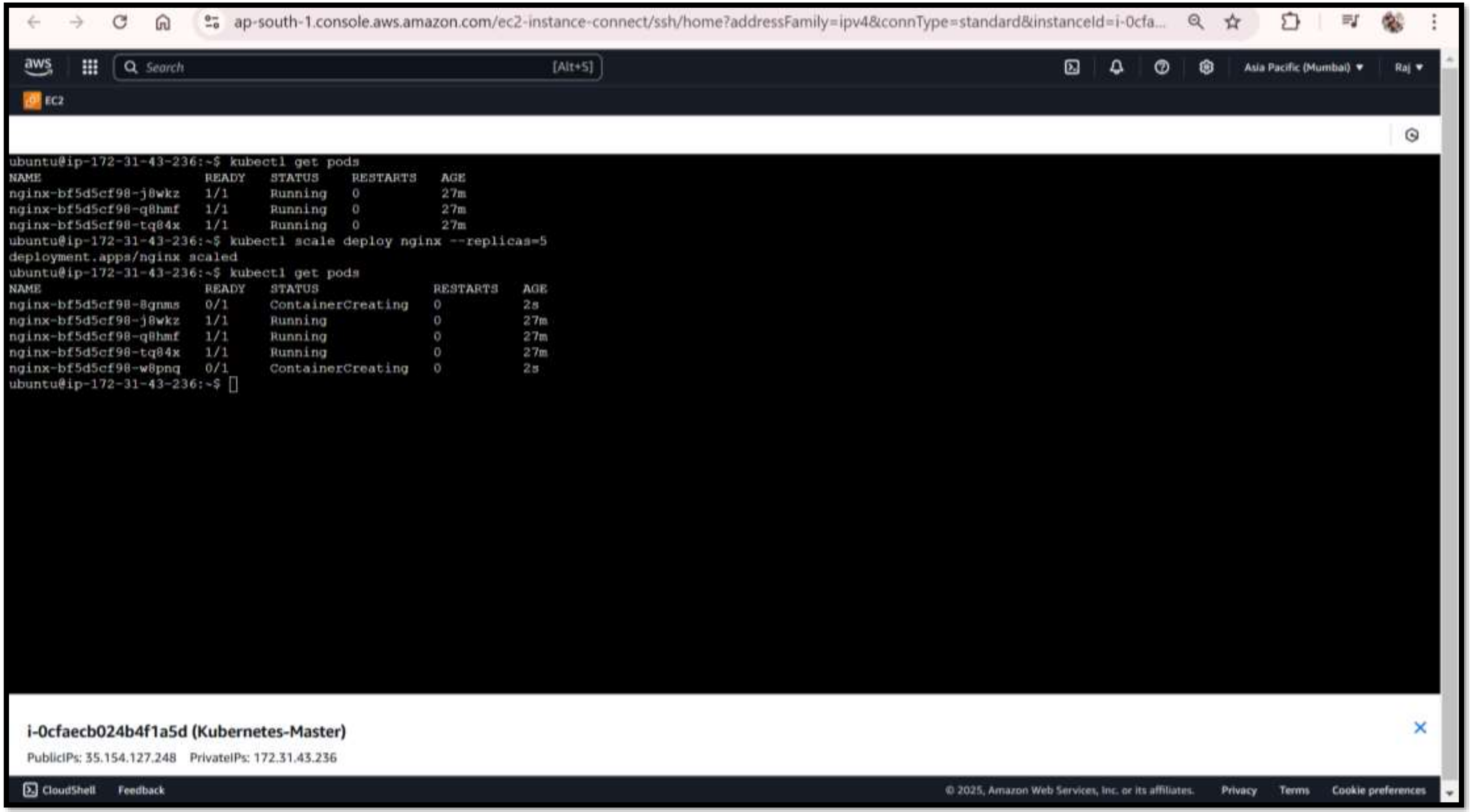


Tasks To Be Performed:

1. Use the previous deployment
2. Change the replicas to 5 for the deployment

Scaling nginx replicas 3 to replicas 5

```
kubectl scale deploy nginx --replicas=5
```



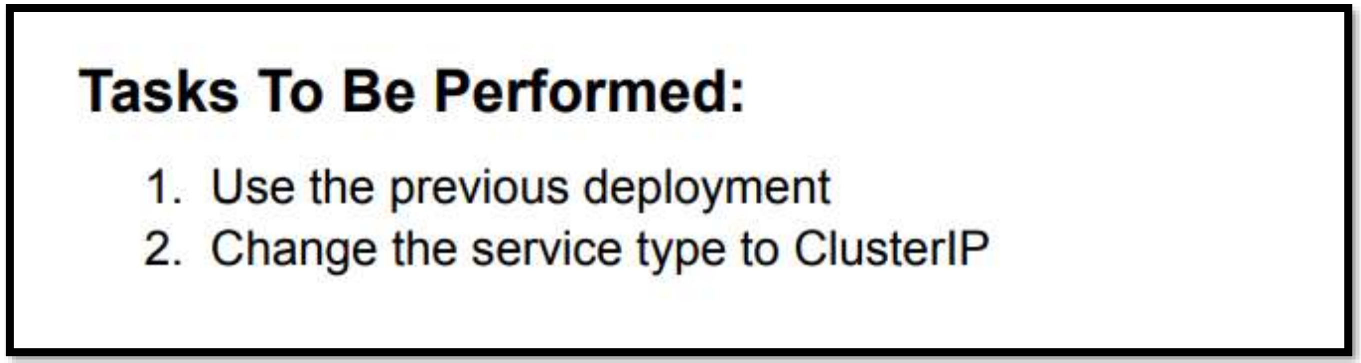
```
ubuntu@ip-172-31-43-236:~$ kubectl get pods
NAME                    READY   STATUS    RESTARTS   AGE
nginx-bf5d5cf98-j8wkz   1/1     Running   0          27m
nginx-bf5d5cf98-q8hmf   1/1     Running   0          27m
nginx-bf5d5cf98-tq84x   1/1     Running   0          27m
ubuntu@ip-172-31-43-236:~$ kubectl scale deploy nginx --replicas=5
deployment.apps/nginx scaled
ubuntu@ip-172-31-43-236:~$ kubectl get pods
NAME                    READY   STATUS              RESTARTS   AGE
nginx-bf5d5cf98-8gnms   0/1     ContainerCreating   0          2s
nginx-bf5d5cf98-j8wkz   1/1     Running             0          27m
nginx-bf5d5cf98-q8hmf   1/1     Running             0          27m
nginx-bf5d5cf98-tq84x   1/1     Running             0          27m
nginx-bf5d5cf98-w8pnq   0/1     ContainerCreating   0          2s
ubuntu@ip-172-31-43-236:~$
```

i-0cfaecb024b4f1a5d (Kubernetes-Master)

PublicIPs: 35.154.127.248 PrivateIPs: 172.31.43.236

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0cfa...
Search
[Alt+S]
Asia Pacific (Mumbai)
Raj
EC2
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

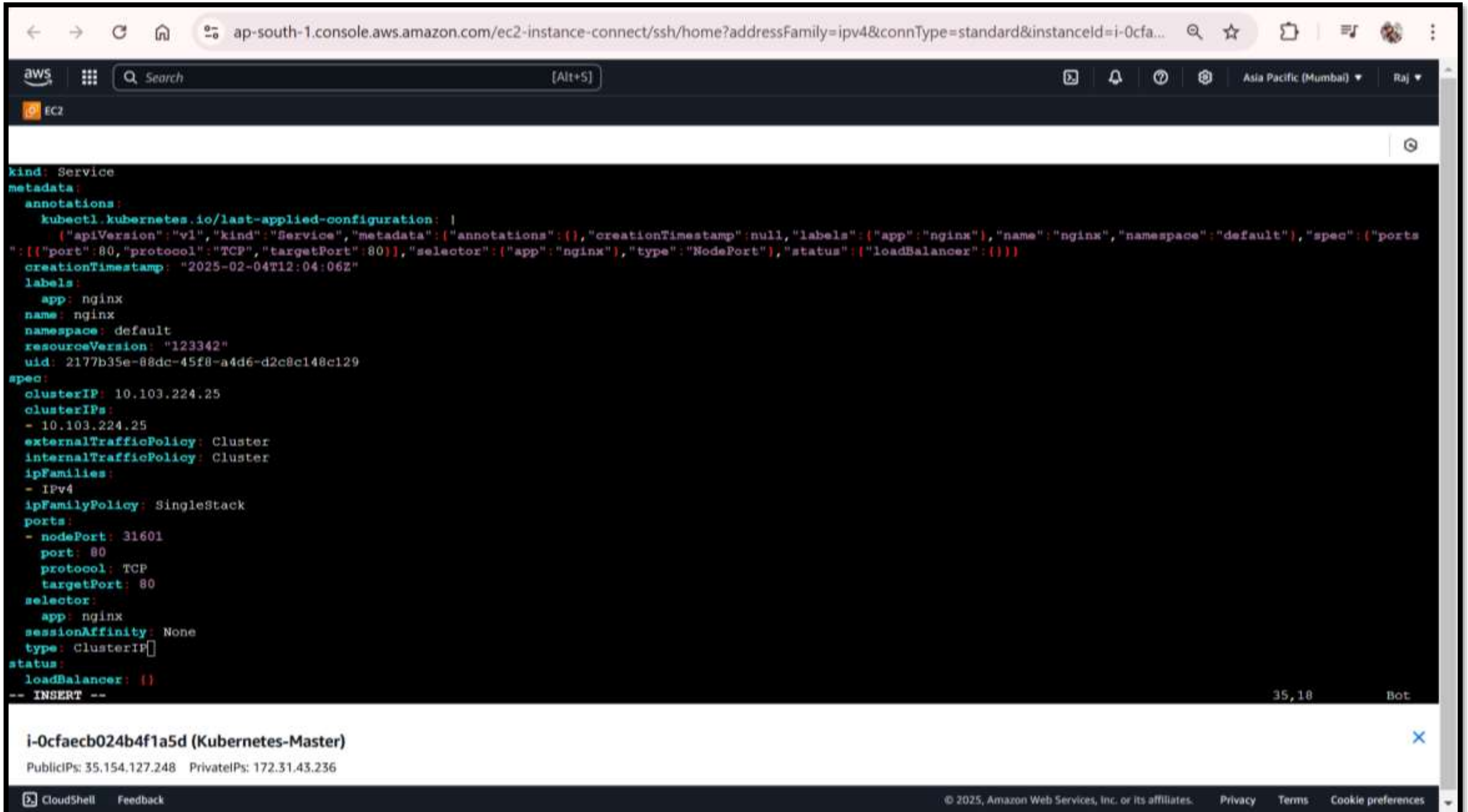
MODULE 7: KUBERNETES ASSIGNMENTS -4



Tasks To Be Performed:

1. Use the previous deployment
2. Change the service type to ClusterIP

Changing service type NodePort to Cluster IP



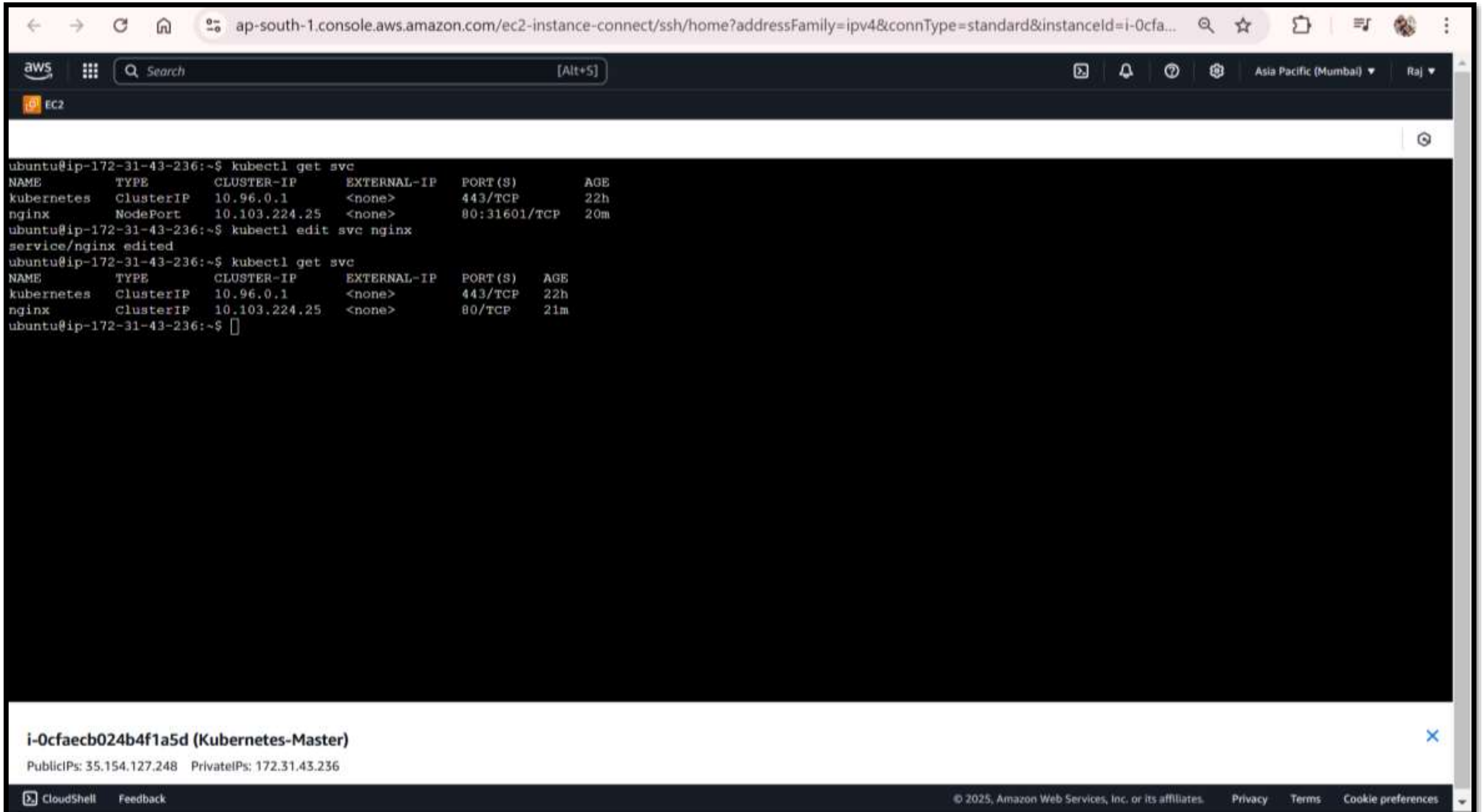
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0cfa...

```
kind: Service
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"creationTimestamp":null,"labels":{"app":"nginx"},"name":"nginx","namespace":"default"},"spec":{"ports
":[{"port":80,"protocol":"TCP","targetPort":80}],"selector":{"app":"nginx"},"type":"NodePort"},"status":{"loadBalancer":{}}}
  creationTimestamp: "2025-02-04T12:04:06Z"
  labels:
    app: nginx
  name: nginx
  namespace: default
  resourceVersion: "123342"
  uid: 2177b35e-88dc-45f8-a4d6-d2c8c148c129
spec:
  clusterIP: 10.103.224.25
  clusterIPs:
  - 10.103.224.25
  externalTrafficPolicy: Cluster
  internalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - nodePort: 31601
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
-- INSERT --                                                                                    35,18         Bot
```

i-0cfaecb024b4f1a5d (Kubernetes-Master)

PublicIPs: 35.154.127.248 PrivateIPs: 172.31.43.236

Validate service type NodePort to Cluster IP is changed



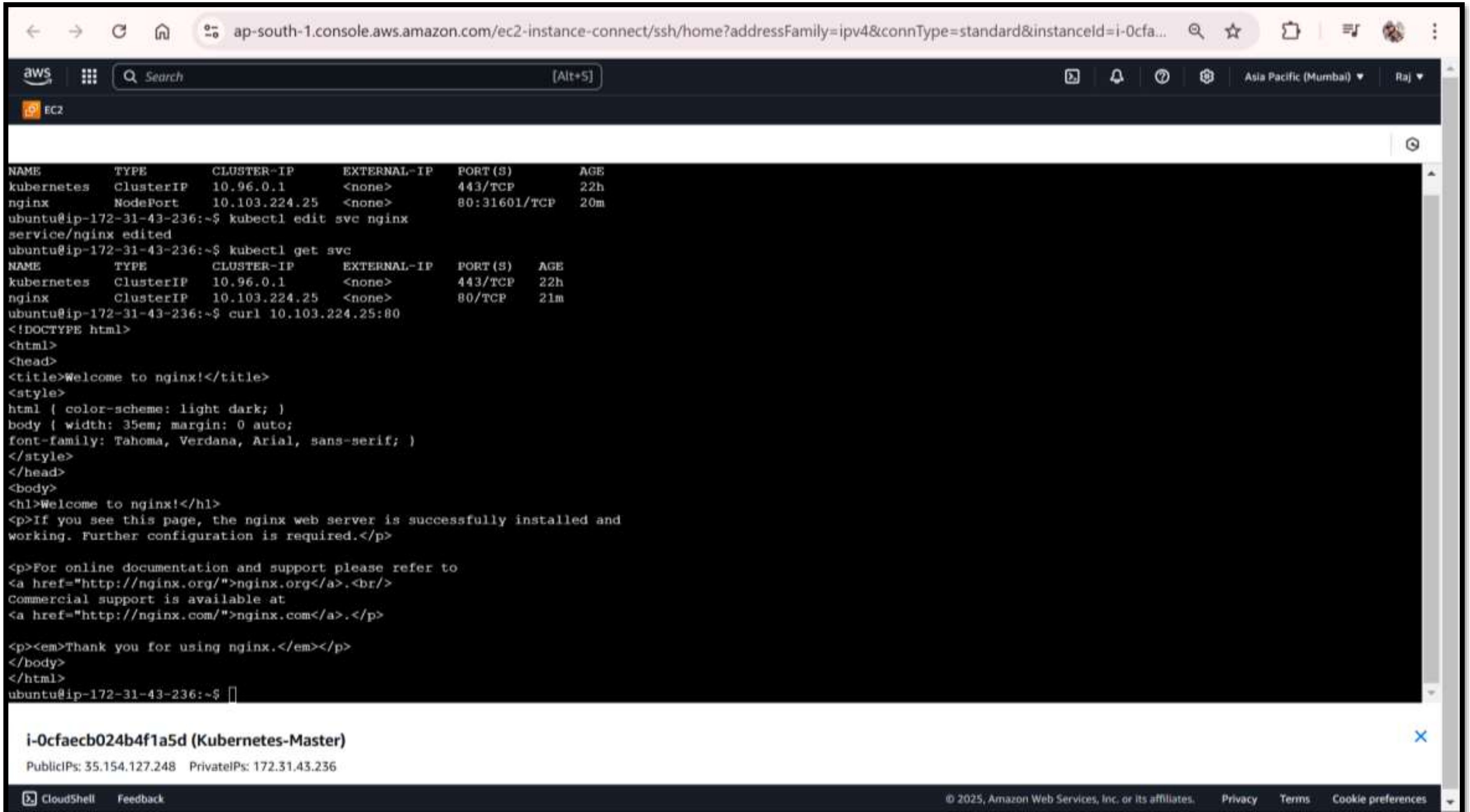
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0cfa...

```
ubuntu@ip-172-31-43-236:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP        22h
nginx        NodePort    10.103.224.25   <none>        80:31601/TCP   20m
ubuntu@ip-172-31-43-236:~$ kubectl edit svc nginx
service/nginx edited
ubuntu@ip-172-31-43-236:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP   22h
nginx        ClusterIP   10.103.224.25   <none>        80/TCP    21m
ubuntu@ip-172-31-43-236:~$ 
```

i-0cfaecb024b4f1a5d (Kubernetes-Master)

PublicIPs: 35.154.127.248   PrivateIPs: 172.31.43.236

Validate Cluster IP server we call internaly



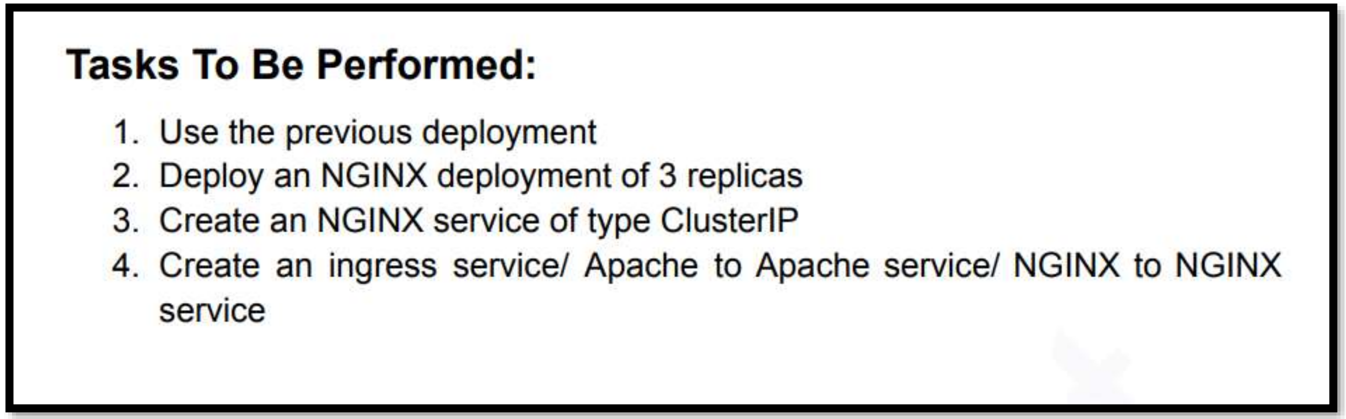
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0cfa...

```
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP        22h
nginx        NodePort    10.103.224.25   <none>        80:31601/TCP   20m
ubuntu@ip-172-31-43-236:~$ kubectl edit svc nginx
service/nginx edited
ubuntu@ip-172-31-43-236:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP   22h
nginx        ClusterIP   10.103.224.25   <none>        80/TCP    21m
ubuntu@ip-172-31-43-236:~$ curl 10.103.224.25:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
ubuntu@ip-172-31-43-236:~$
```

i-0cfaecb024b4f1a5d (Kubernetes-Master)

PublicIPs: 35.154.127.248   PrivateIPs: 172.31.43.236

MODULE 7: KUBERNETES ASSIGNMENTS -5



Tasks To Be Performed:

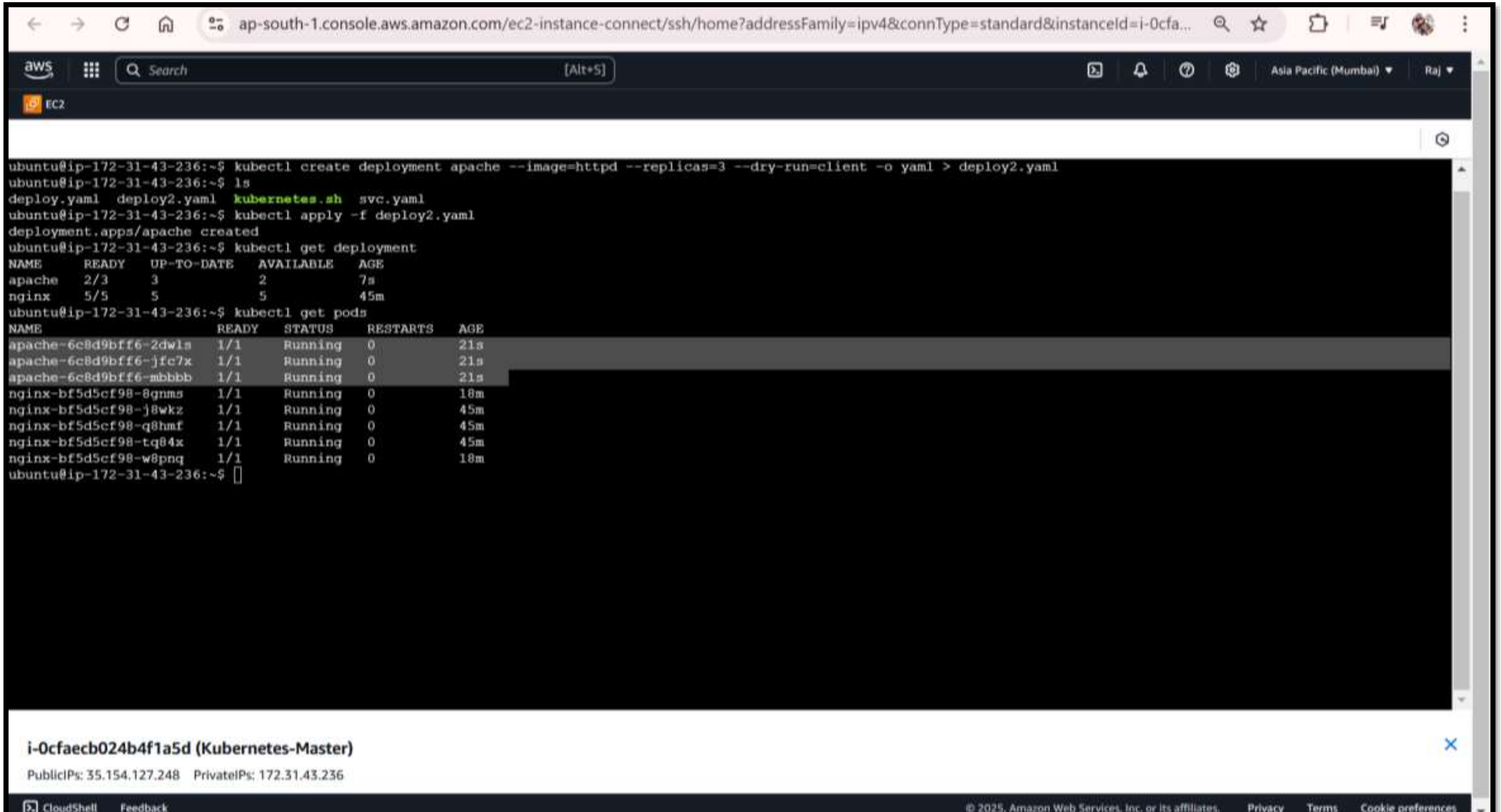
1. Use the previous deployment
2. Deploy an NGINX deployment of 3 replicas
3. Create an NGINX service of type ClusterIP
4. Create an ingress service/ Apache to Apache service/ NGINX to NGINX service

Note : as the ingress service is in freeze as team guide for 5 assignment I have implemented

Deploying a Apache replicas

```
1. kubectl create deployment apache --image=httpd --replicas=3 --dry-
   run=client -o yaml > deploy2.yaml

2. kubectl apply -f deploy2.yaml

3. kubectl get deployment

4. kubectl get pods
```

Validating Deployed a Apache replicas



ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0cfa...
Search
[Alt+S]
Asia Pacific (Mumbai)
Raj
EC2

```
ubuntu@ip-172-31-43-236:~$ kubectl create deployment apache --image=httpd --replicas=3 --dry-run=client -o yaml > deploy2.yaml
ubuntu@ip-172-31-43-236:~$ ls
deploy.yaml  deploy2.yaml  kubernetes.sh  svc.yaml
ubuntu@ip-172-31-43-236:~$ kubectl apply -f deploy2.yaml
deployment.apps/apache created
ubuntu@ip-172-31-43-236:~$ kubectl get deployment
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
apache   2/3     3            2           7s
nginx    5/5     5            5           45m
ubuntu@ip-172-31-43-236:~$ kubectl get pods
NAME                       READY   STATUS    RESTARTS   AGE
apache-6c8d9bff6-2dwls     1/1     Running   0          21s
apache-6c8d9bff6-jfc7x     1/1     Running   0          21s
apache-6c8d9bff6-mbbbb     1/1     Running   0          21s
nginx-bf5d5cf98-8gnms      1/1     Running   0          18m
nginx-bf5d5cf98-j8wkz      1/1     Running   0          45m
nginx-bf5d5cf98-q8hmf      1/1     Running   0          45m
nginx-bf5d5cf98-tq84x      1/1     Running   0          45m
nginx-bf5d5cf98-w8pnq      1/1     Running   0          18m
ubuntu@ip-172-31-43-236:~$ 
```

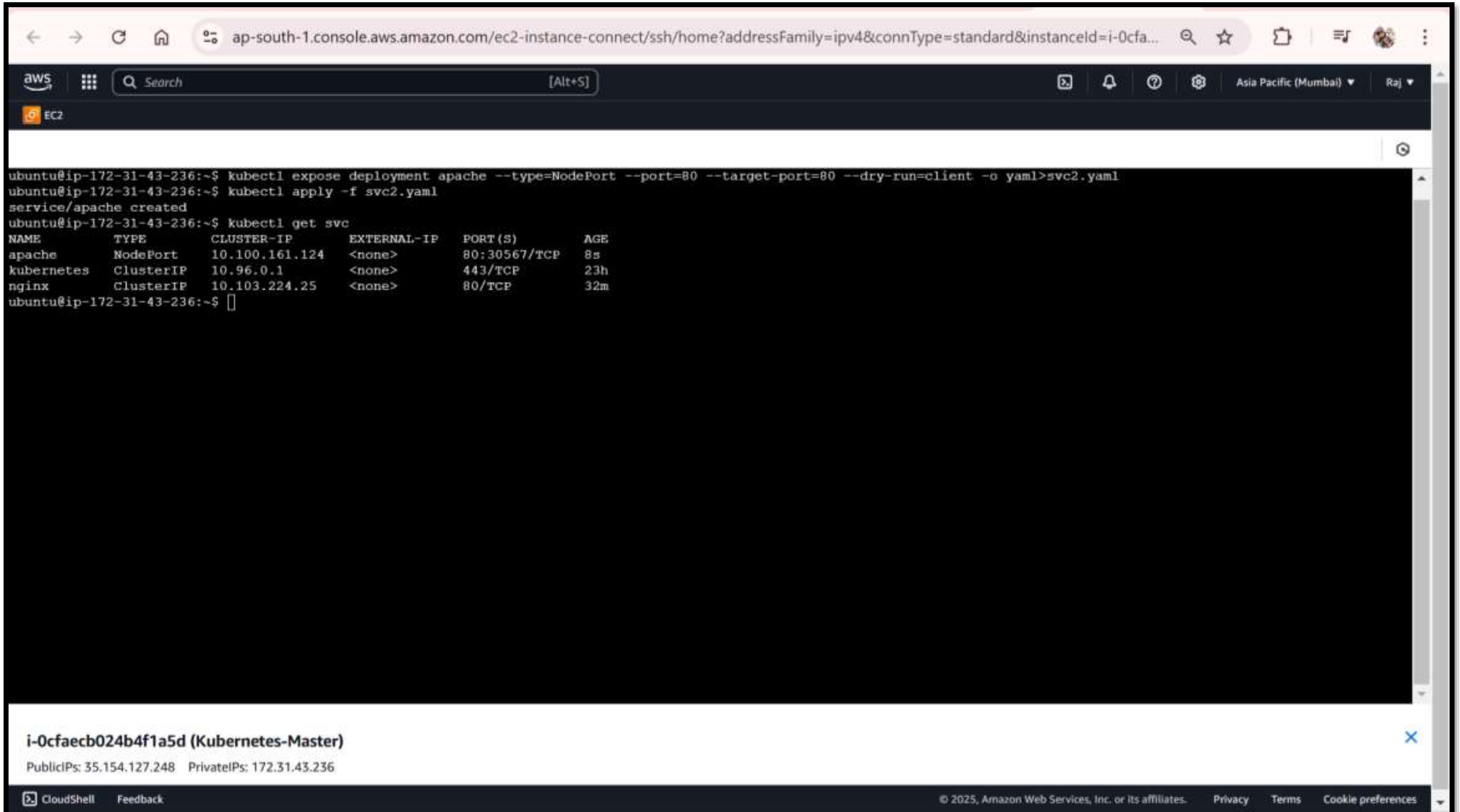
i-0cfaecb024b4f1a5d (Kubernetes-Master)

PublicIPs: 35.154.127.248   PrivateIPs: 172.31.43.236

Deploying Apache NodePort Service

```
1. kubectl expose deployment apache --type=NodePort --port=80 --
   target-port=80 --dry-run=client -o yaml>svc2.yaml

2. kubectl apply -f svc2.yaml

3. kubectl get svc
```

Validating Apache NodePort Service



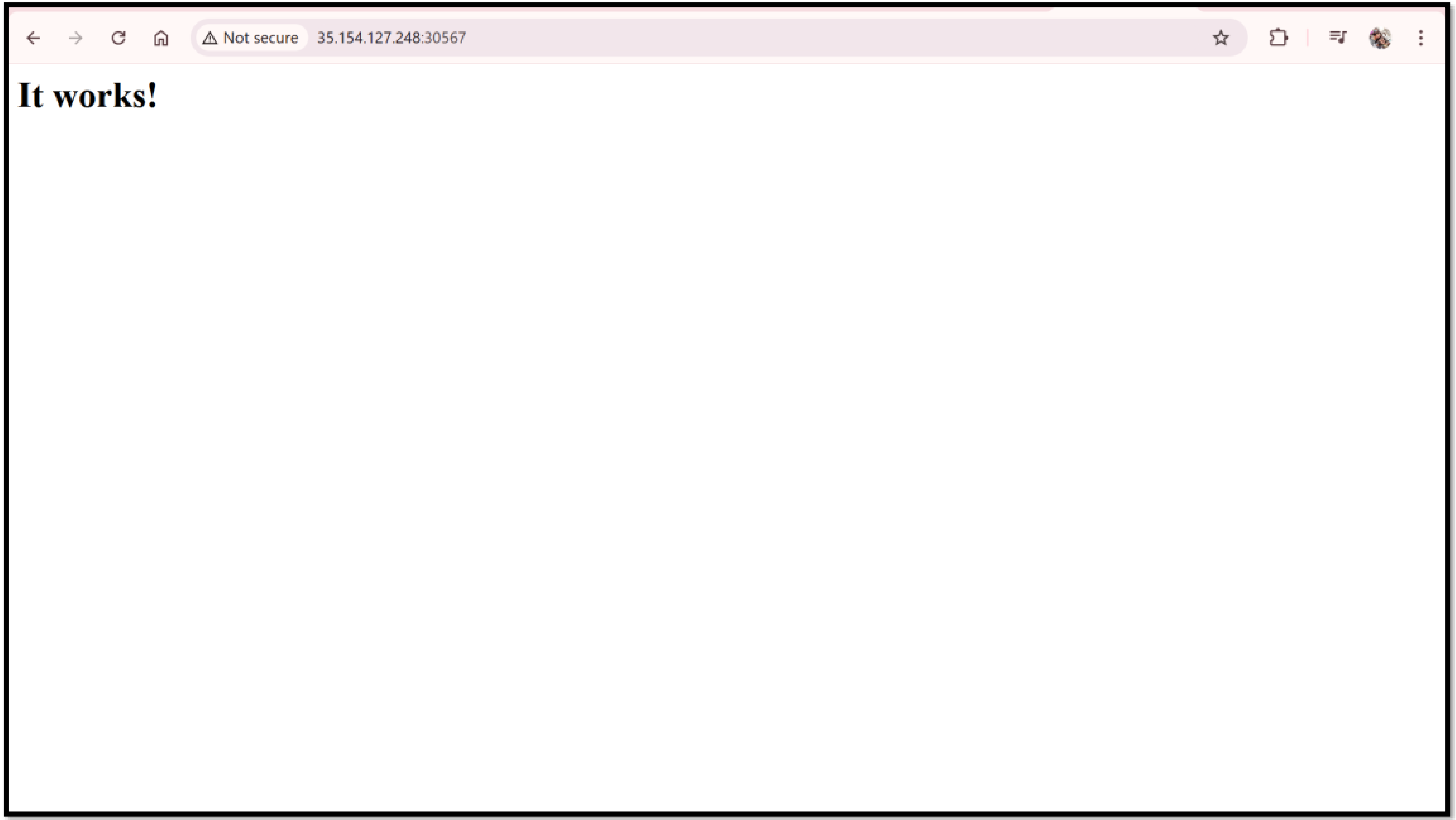
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0cfa...
aws
Search
[Alt+S]
Asia Pacific (Mumbai) ▼
Raj ▼
EC2

```
ubuntu@ip-172-31-43-236:~$ kubectl expose deployment apache --type=NodePort --port=80 --target-port=80 --dry-run=client -o yaml>svc2.yaml
ubuntu@ip-172-31-43-236:~$ kubectl apply -f svc2.yaml
service/apache created
ubuntu@ip-172-31-43-236:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
apache       NodePort    10.100.161.124   <none>        80:30567/TCP   8s
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP        23h
nginx        ClusterIP   10.103.224.25    <none>        80/TCP         32m
ubuntu@ip-172-31-43-236:~$ 
```

i-0cfaecb024b4f1a5d (Kubernetes-Master)

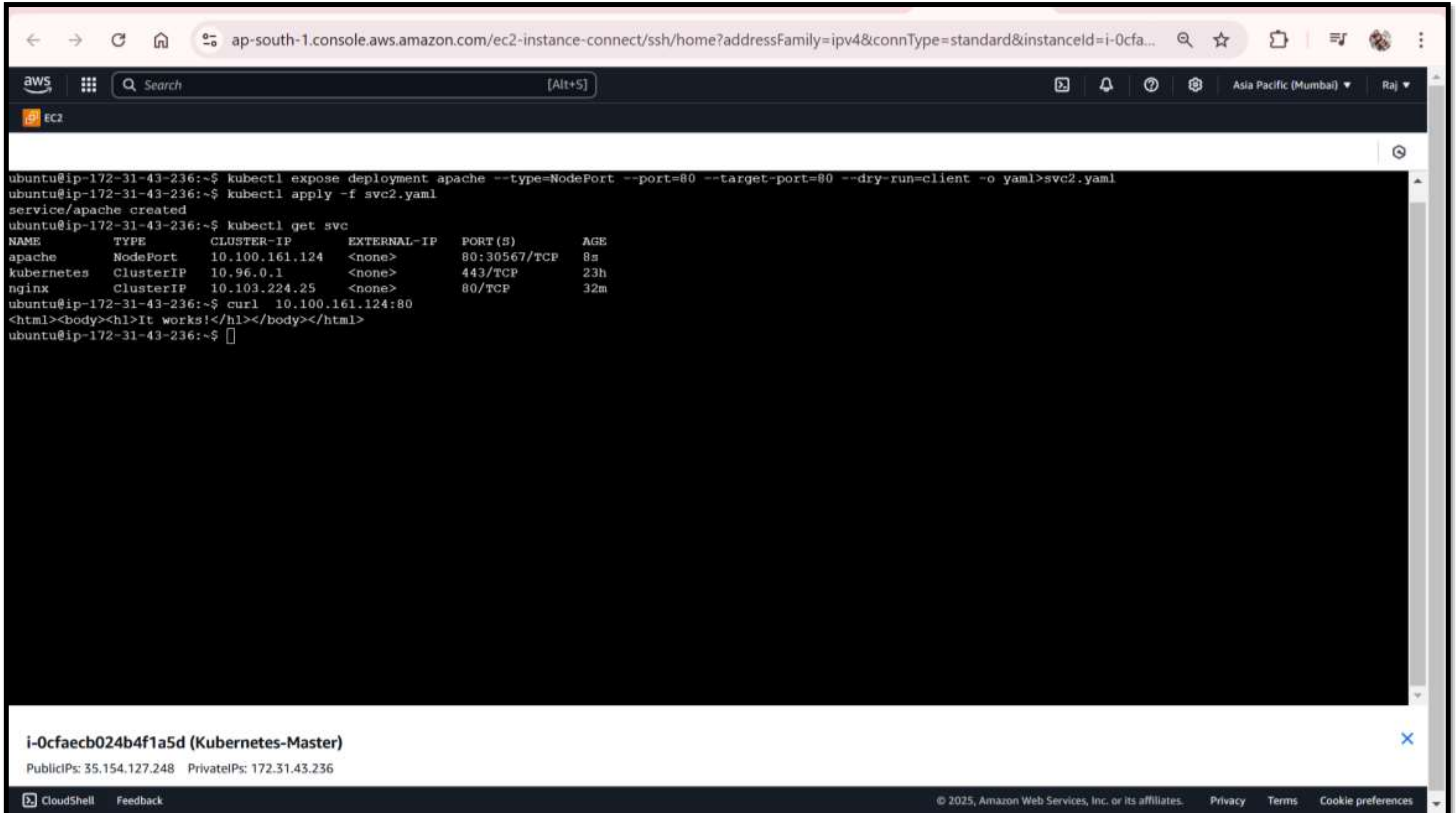
PublicIPs: 35.154.127.248   PrivateIPs: 172.31.43.236

We can access Apache NodePort Service in browser



It works!

Validating page content using curl command


ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0cfa...
Search
[Alt+S]
Asia Pacific (Mumbai)
Raj
EC2

```
ubuntu@ip-172-31-43-236:~$ kubectl expose deployment apache --type=NodePort --port=80 --target-port=80 --dry-run=client -o yaml>svc2.yaml
ubuntu@ip-172-31-43-236:~$ kubectl apply -f svc2.yaml
service/apache created
ubuntu@ip-172-31-43-236:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
apache       NodePort    10.100.161.124   <none>        80:30567/TCP   8s
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP        23h
nginx        ClusterIP   10.103.224.25    <none>        80/TCP         32m
ubuntu@ip-172-31-43-236:~$ curl  10.100.161.124:80
<html><body><h1>It works!</h1></body></html>
ubuntu@ip-172-31-43-236:~$
```

i-0cfaecb024b4f1a5d (Kubernetes-Master)

PublicIPs: 35.154.127.248   PrivateIPs: 172.31.43.236