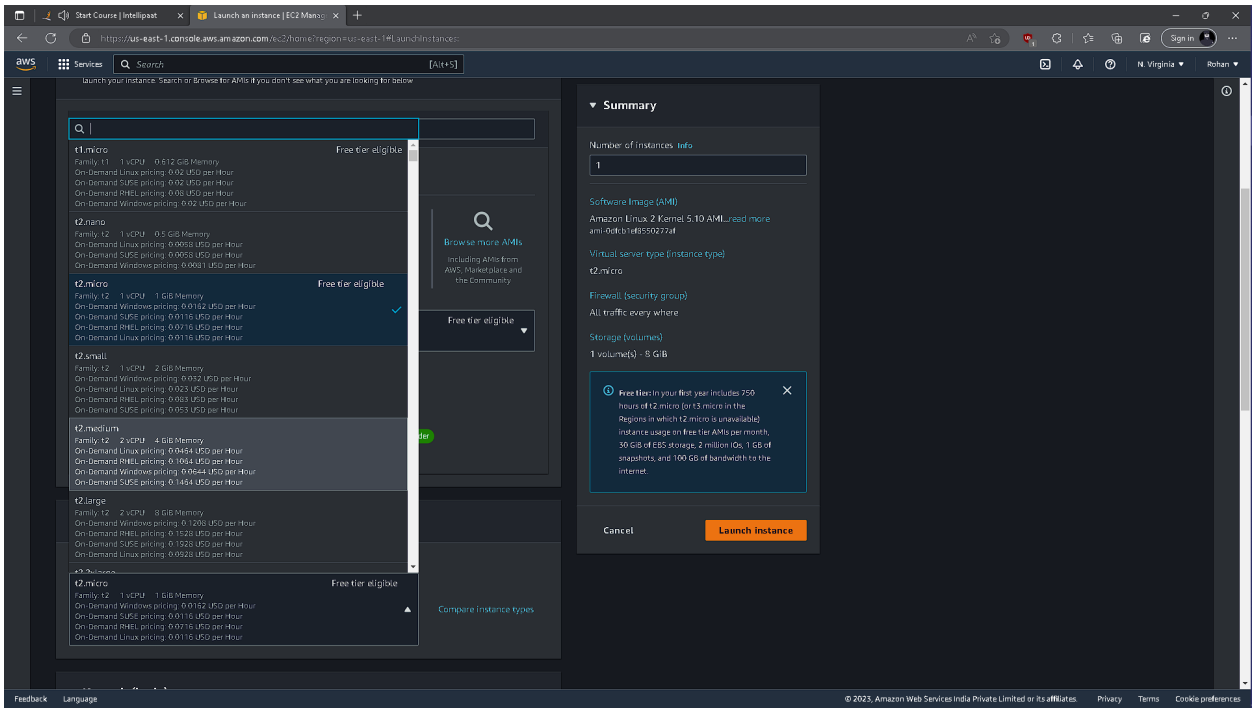
Kubernetes - 1

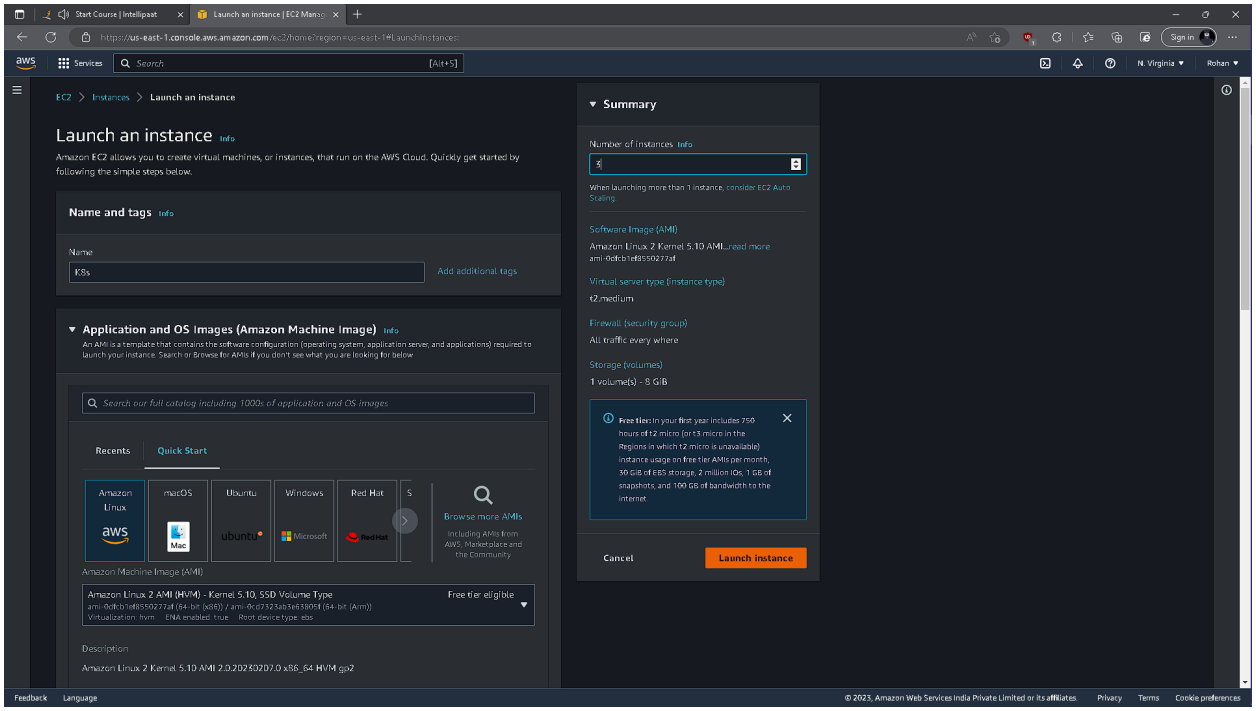
You have been asked to:

- **Deploy a Kubernetes Cluster for 3 nodes**
- **Create a nginx deployment of 3 replicas**

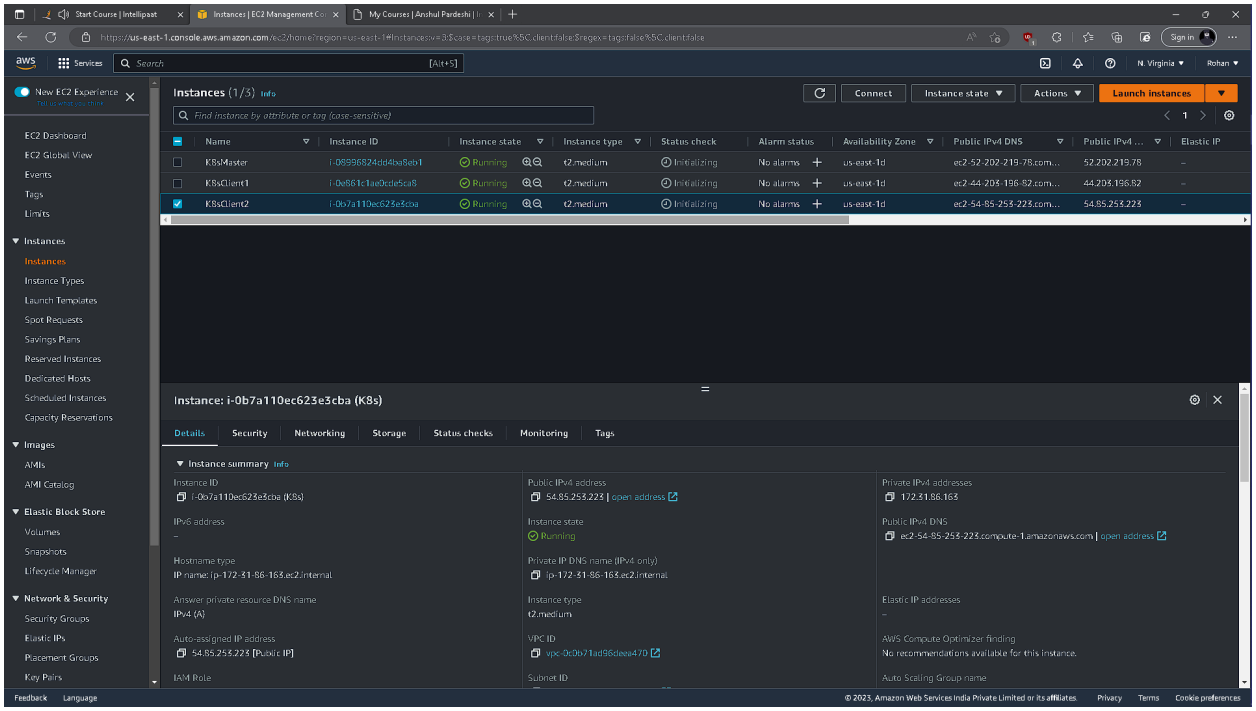
Let’s create an instance and install k8s first. Choose a minimum t2.med AWS instance for it.



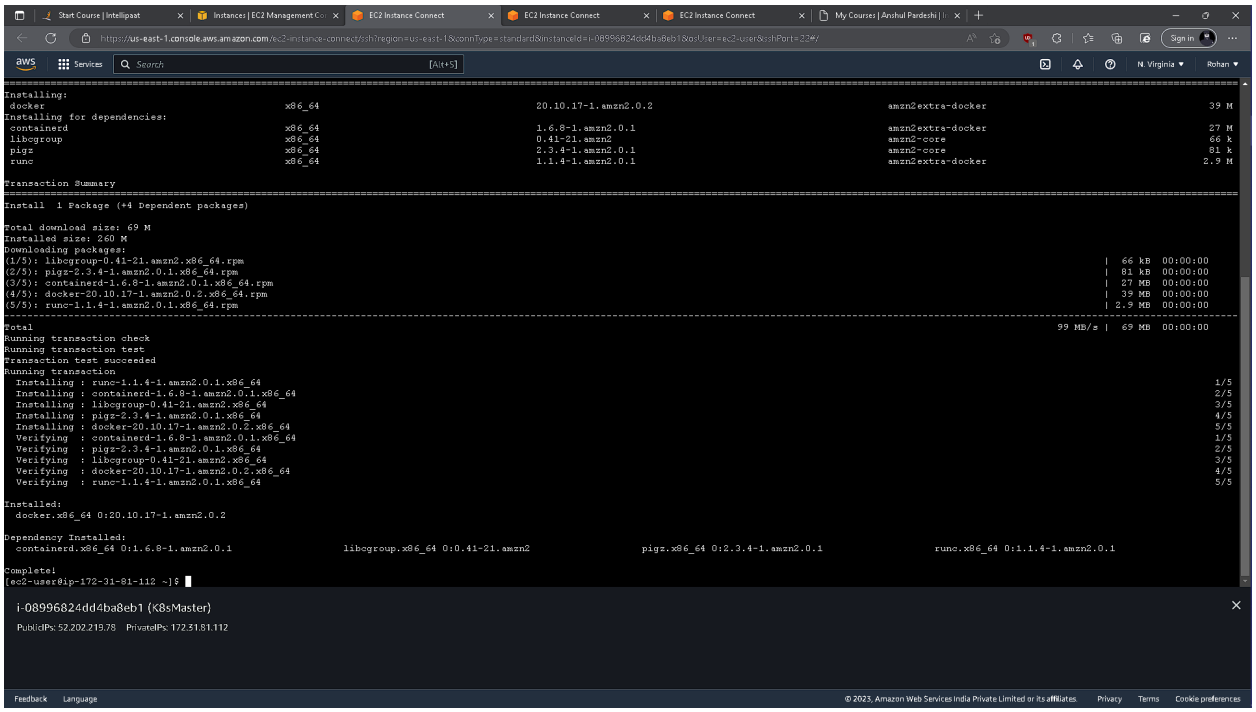
3 instances since 3 node clusters are needed.



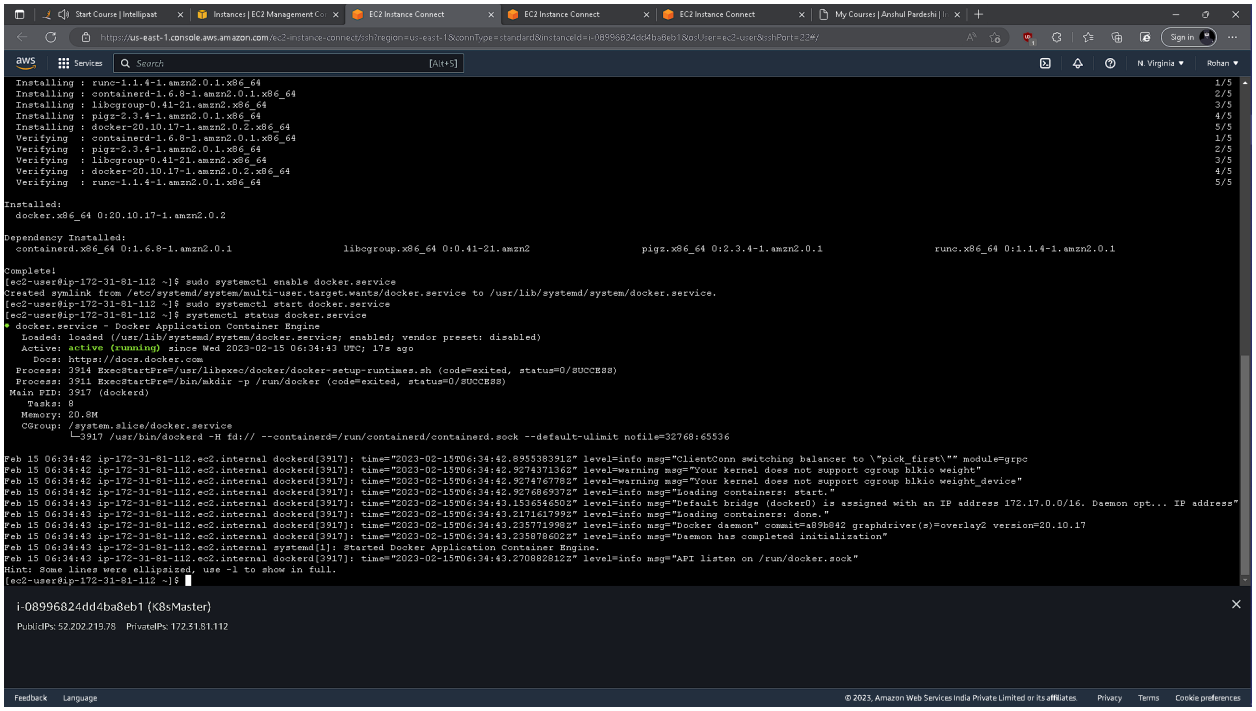
Instances created.



Use these command to install docker:
sudo yum update
sudo yum install docker
sudo systemctl enable docker.service
sudo systemctl start docker.service
sudo systemctl start docker.service
systemctl status docker.service



Docker is running



=======installation of kubelet kubeadm kubectl====
Run on BOTH MASTER and Worker

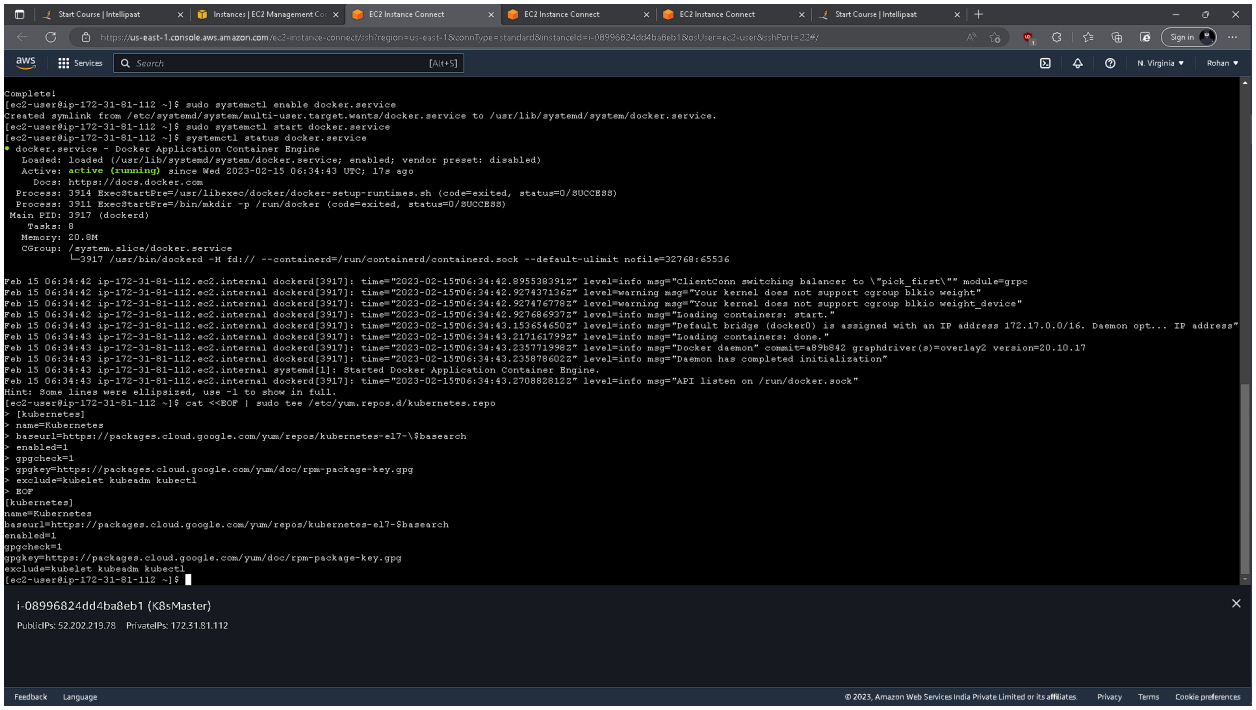
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-\$basearch
enabled=1
gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
exclude=kubelet kubeadm kubectl
EOF

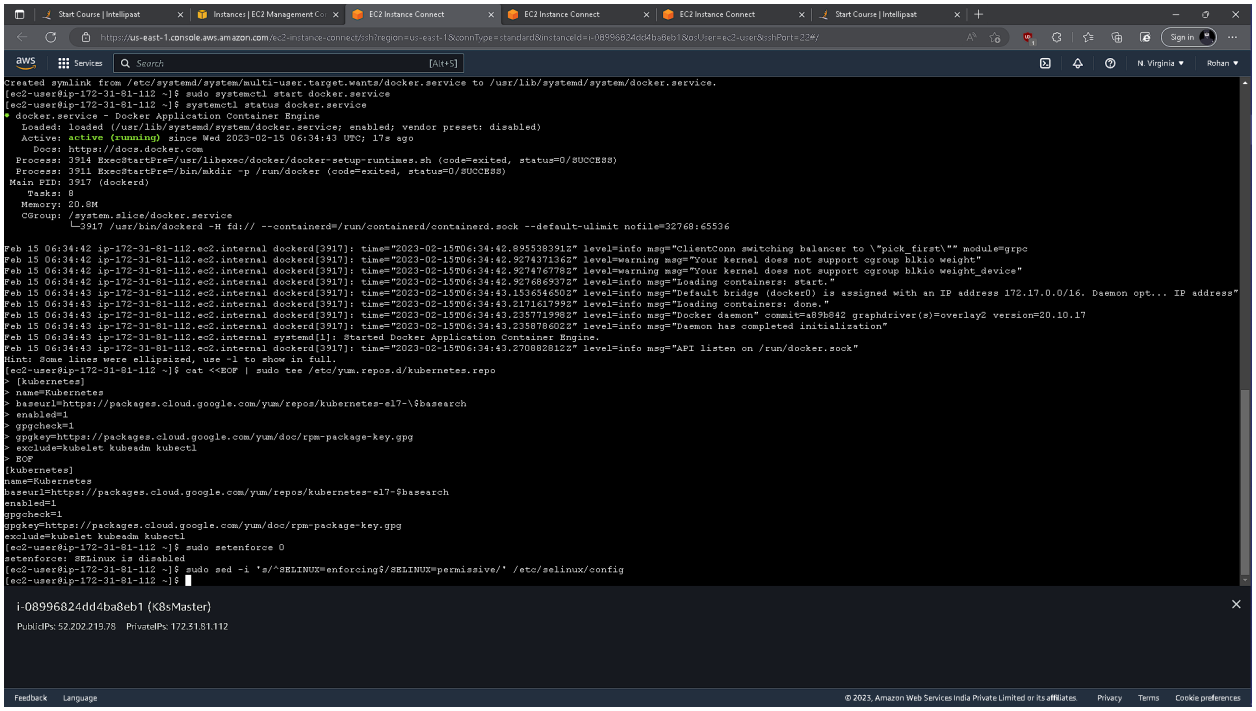
 # Set SELinux in permissive mode (effectively disabling it)

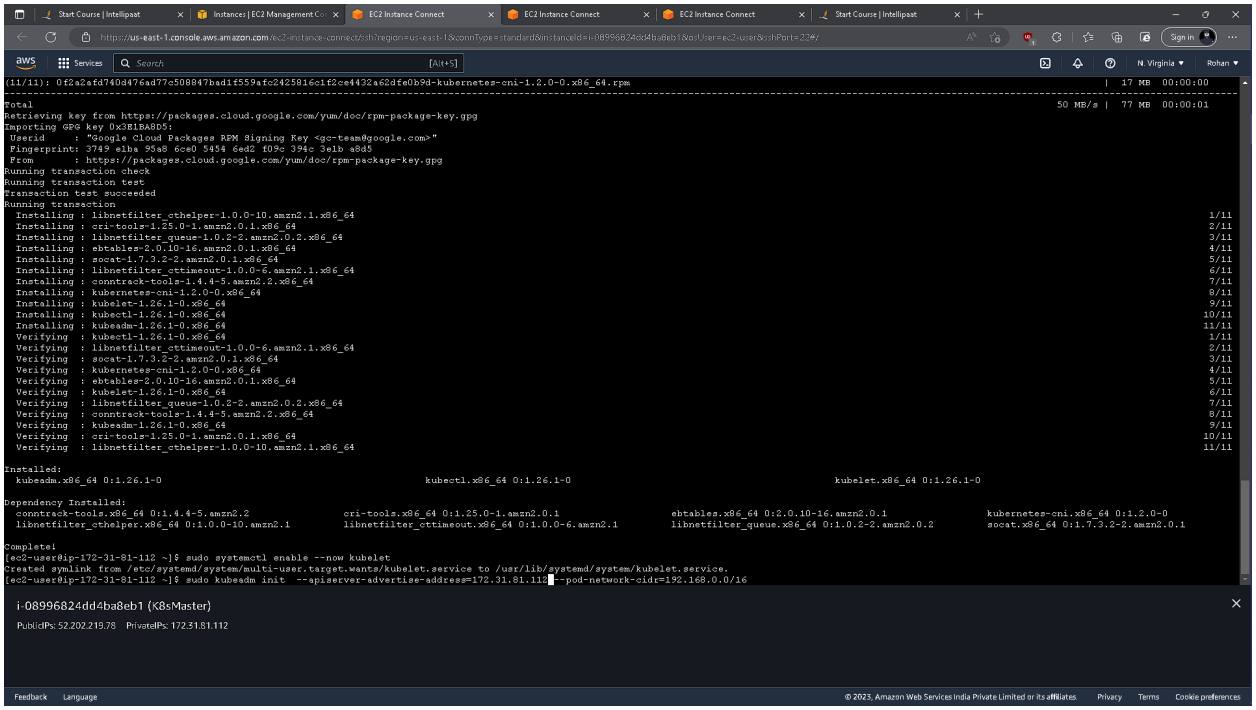
sudo setenforce 0
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config

sudo yum install -y kubelet kubeadm kubectl --disableexcludes=kubernetes

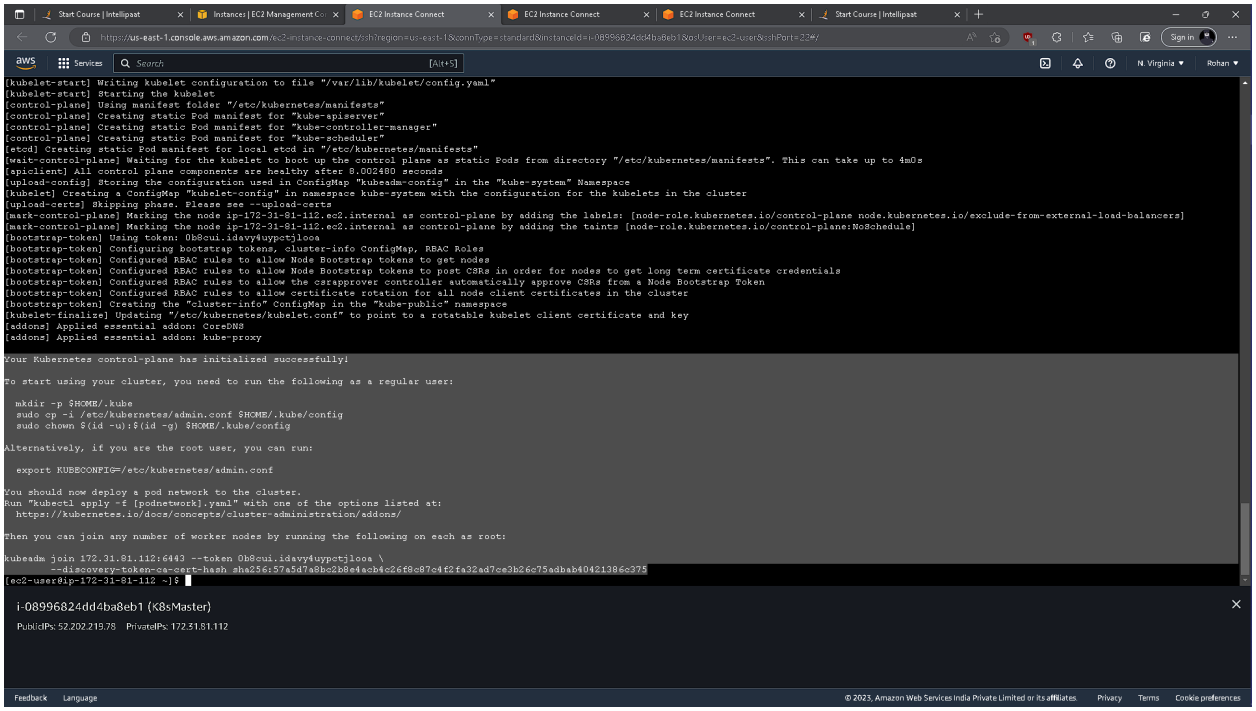
sudo systemctl enable --now kubelet


Start Course | Intellipaat
Instances | EC2 Management Co
EC2 Instance Connect
EC2 Instance Connect
EC2 Instance Connect
Start Course | Intellipaat
https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-08996824dd4ba8eb1&osUser=ec2-user&sshPort=22#/
Sign in
aws
Services
Search
[Alt+S]
N. Virginia
Rohan
Complete!
[ec2-user@ip-172-31-81-112 ~]$ sudo systemctl enable docker.service
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-81-112 ~]$ sudo systemctl start docker.service
[ec2-user@ip-172-31-81-112 ~]$ systemctl status docker.service
● docker.service - Docker Application Container Engine
Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
Active: active (running) since Wed 2023-02-15 06:34:43 UTC; 17s ago
Docs: https://docs.docker.com
Process: 3914 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
Process: 3911 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
Main PID: 3917 (dockerd)
Tasks: 8
Memory: 20.8M
CGroup: /system.slice/docker.service
└─3917 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.895538391Z" level=info msg="ClientConn switching balancer to \"pick_first\"" module=grpc
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927437136Z" level=warning msg="Your kernel does not support cgroup blkio weight"
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927476778Z" level=warning msg="Your kernel does not support cgroup blkio weight_device"
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927686937Z" level=info msg="Loading containers: start."
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.153654650Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daemon opt... IP address"
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.217161799Z" level=info msg="Loading containers: done."
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.235771998Z" level=info msg="Docker daemon" commit=a89b842 graphdriver(s)=overlay2 version=20.10.17
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.235878602Z" level=info msg="Daemon has completed initialization"
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal systemd[1]: Started Docker Application Container Engine.
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.270882812Z" level=info msg="API listen on /run/docker.sock"
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-81-112 ~]$ cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
> [kubernetes]
> name=Kubernetes
> baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-\$basearch
> enabled=1
> gpgcheck=1
> gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
> exclude=kubelet kubeadm kubectl
> EOF
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-$basearch
enabled=1
gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
exclude=kubelet kubeadm kubectl
[ec2-user@ip-172-31-81-112 ~]$
i-08996824dd4ba8eb1 (K8sMaster)
PublicIPs: 52.202.213.78 PrivateIPs: 172.31.81.112
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences


Start Course | Intellipaat
Instances | EC2 Management Co
EC2 Instance Connect
EC2 Instance Connect
EC2 Instance Connect
Start Course | Intellipaat
https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-08996824dd4ba8eb1&osUser=ec2-user&sshPort=22#/
Sign in
aws
Services
Search
[Alt+S]
N. Virginia
Rohan
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-81-112 ~]$ sudo systemctl start docker.service
[ec2-user@ip-172-31-81-112 ~]$ systemctl status docker.service
● docker.service - Docker Application Container Engine
Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
Active: active (running) since Wed 2023-02-15 06:34:43 UTC; 17s ago
Docs: https://docs.docker.com
Process: 3914 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
Process: 3911 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
Main PID: 3917 (dockerd)
Tasks: 8
Memory: 20.8M
CGroup: /system.slice/docker.service
└─3917 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.895538391Z" level=info msg="ClientConn switching balancer to \"pick_first\"" module=grpc
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927437136Z" level=warning msg="Your kernel does not support cgroup blkio weight"
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927476778Z" level=warning msg="Your kernel does not support cgroup blkio weight_device"
Feb 15 06:34:42 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:42.927686937Z" level=info msg="Loading containers: start."
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.153654650Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daemon opt... IP address"
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.217161799Z" level=info msg="Loading containers: done."
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.235771998Z" level=info msg="Docker daemon" commit=a89b842 graphdriver(s)=overlay2 version=20.10.17
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.235878602Z" level=info msg="Daemon has completed initialization"
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal systemd[1]: Started Docker Application Container Engine.
Feb 15 06:34:43 ip-172-31-81-112.ec2.internal dockerd[3917]: time="2023-02-15T06:34:43.270882812Z" level=info msg="API listen on /run/docker.sock"
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-81-112 ~]$ cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
> [kubernetes]
> name=Kubernetes
> baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-\$basearch
> enabled=1
> gpgcheck=1
> gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
> exclude=kubelet kubeadm kubectl
> EOF
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-$basearch
enabled=1
gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
exclude=kubelet kubeadm kubectl
[ec2-user@ip-172-31-81-112 ~]$ sudo setenforce 0
setenforce: SELinux is disabled
[ec2-user@ip-172-31-81-112 ~]$ sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
[ec2-user@ip-172-31-81-112 ~]$
i-08996824dd4ba8eb1 (K8sMaster)
PublicIPs: 52.202.213.78 PrivateIPs: 172.31.81.112
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

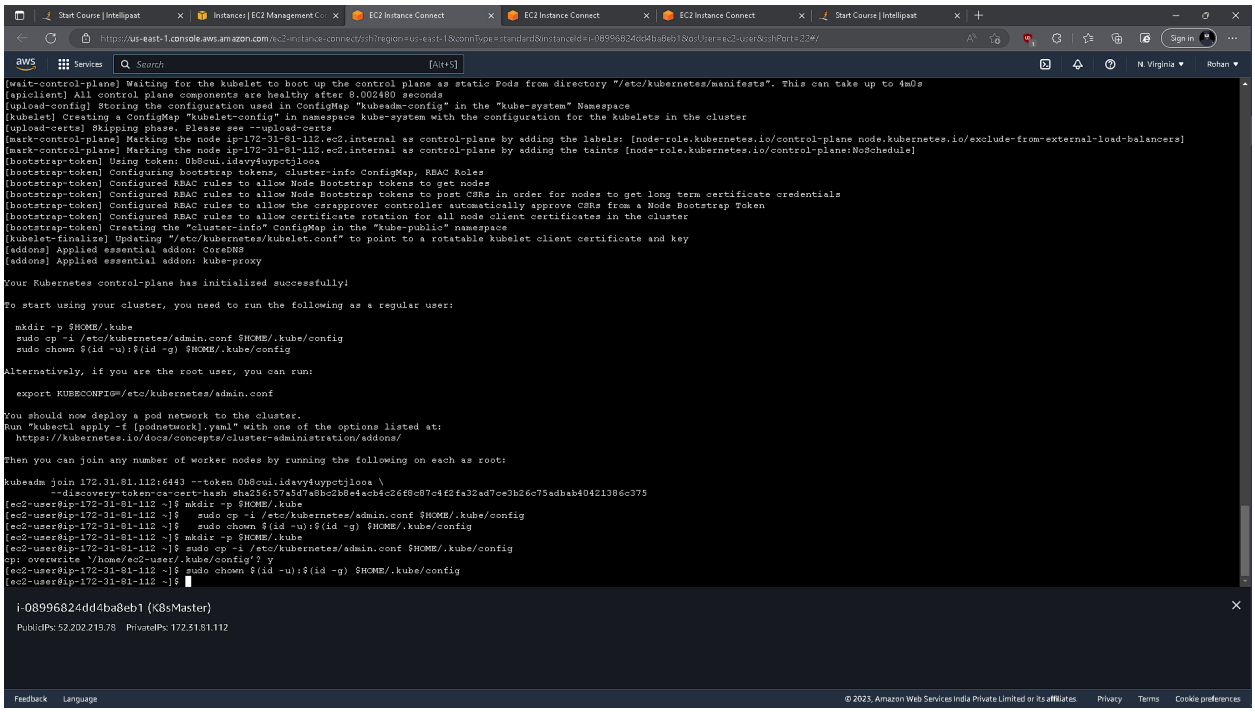


Note this output.

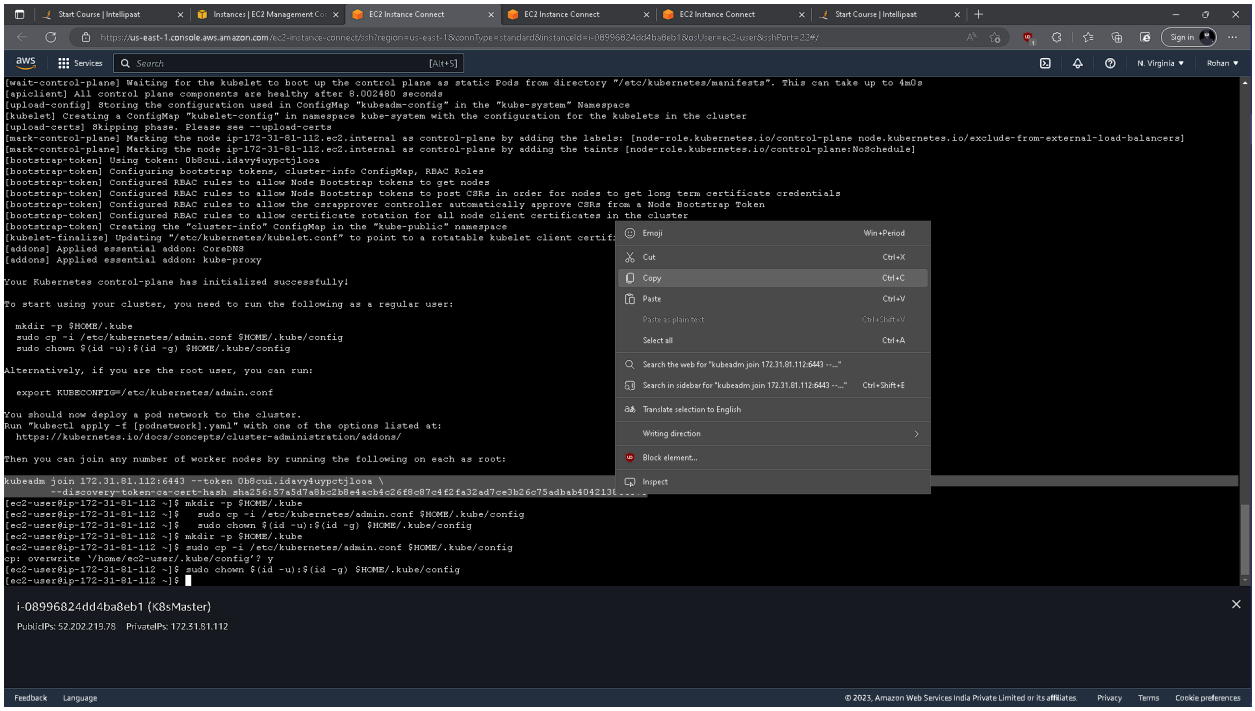


To start using your cluster, you need to run the following as a regular user:

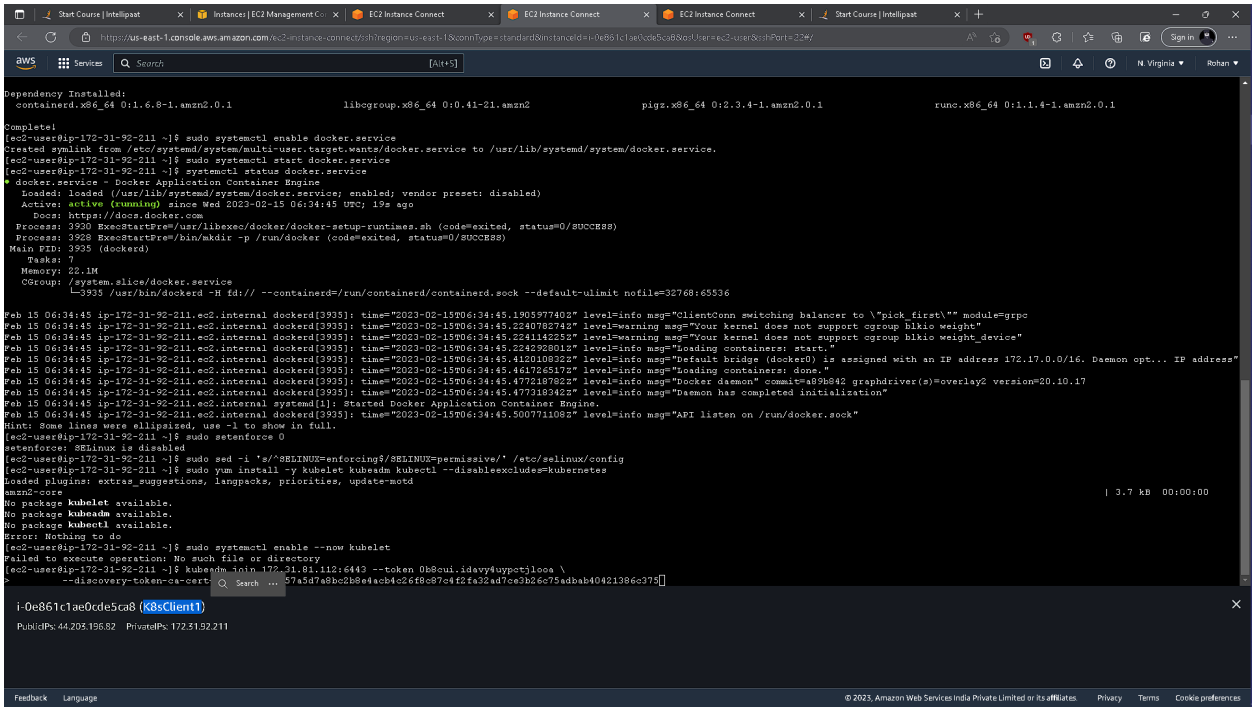
```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```



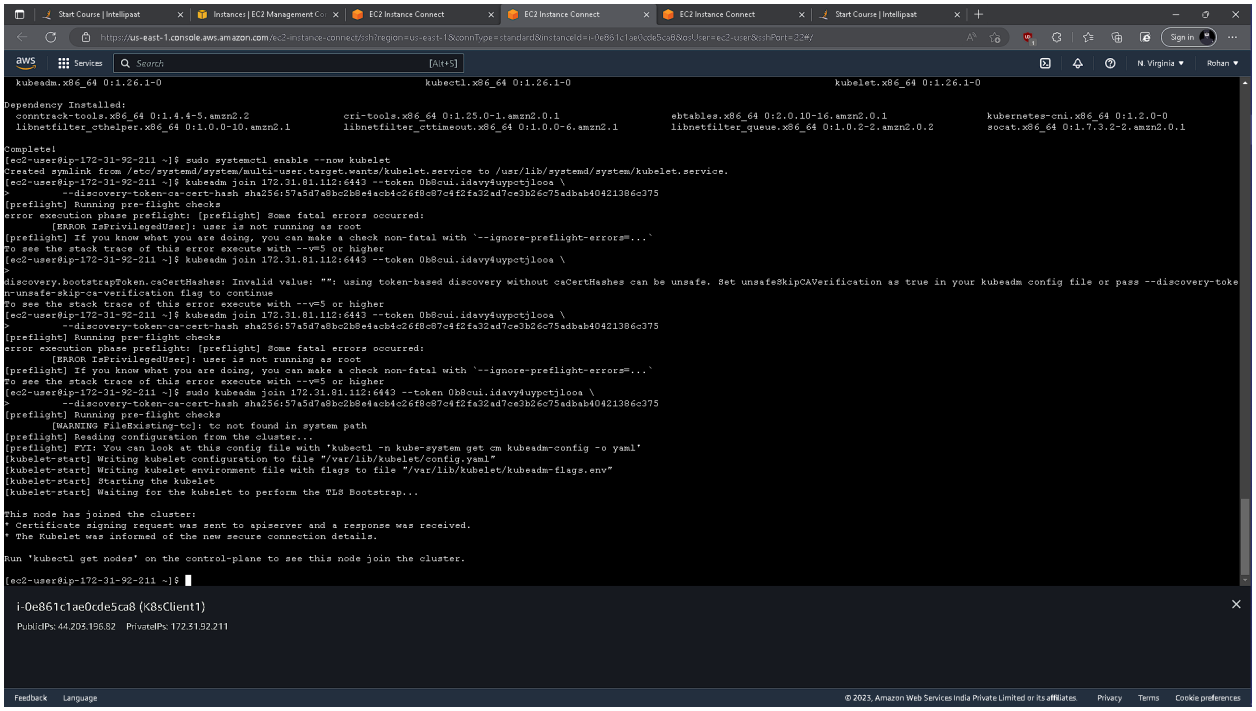
You need to copy paste this in client.



Pasted in client 1

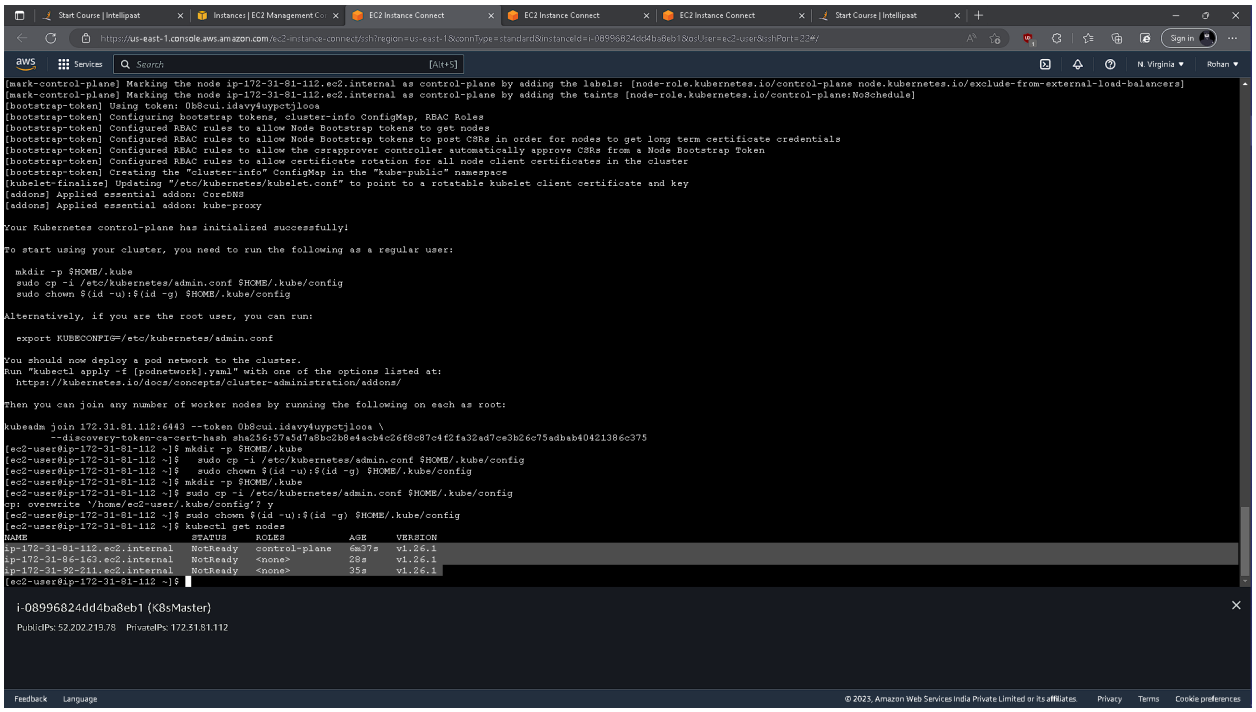


Worker node has been joined.Similarly do it for another client/worker server as well.



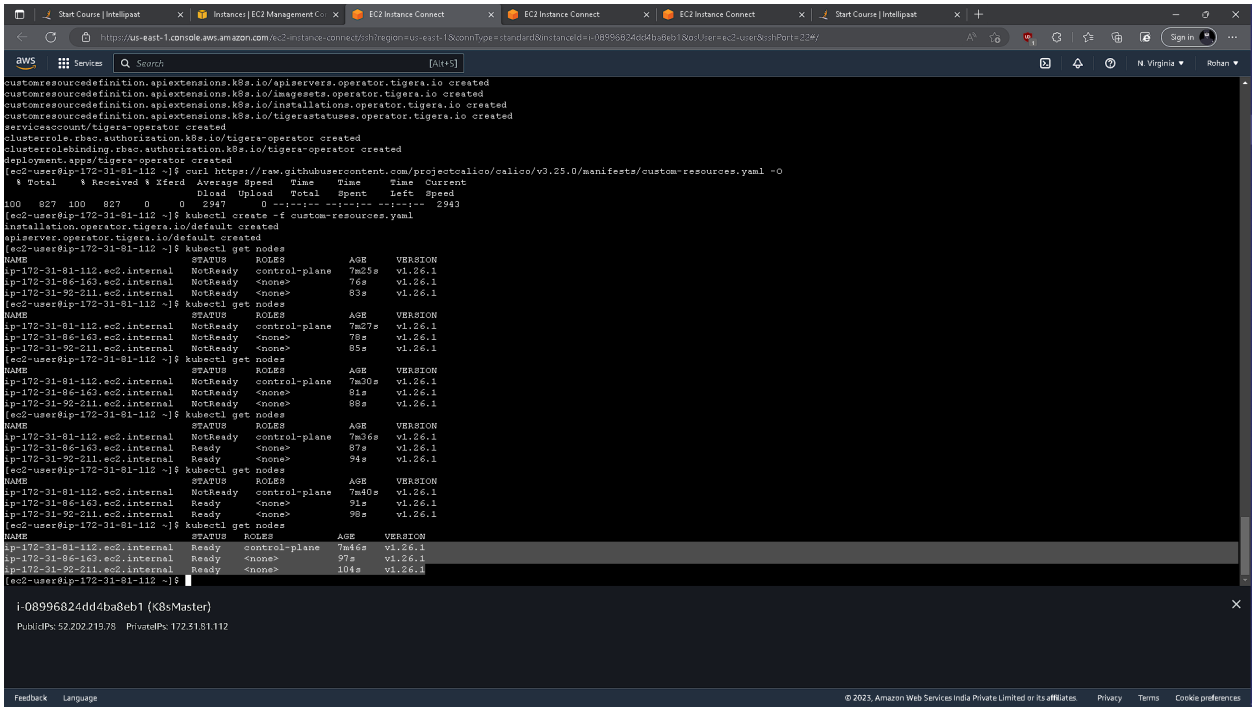
kubectl get node

You can see 3 node cluster is created. They are not ready yet. For it install calico on master.



Install Calico on MASTER ONLY:
kubectl create -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/tigera-operator.yaml
curl https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/custom-resources.yaml -O
kubectl create -f custom-resources.yaml

Nodes are ready.

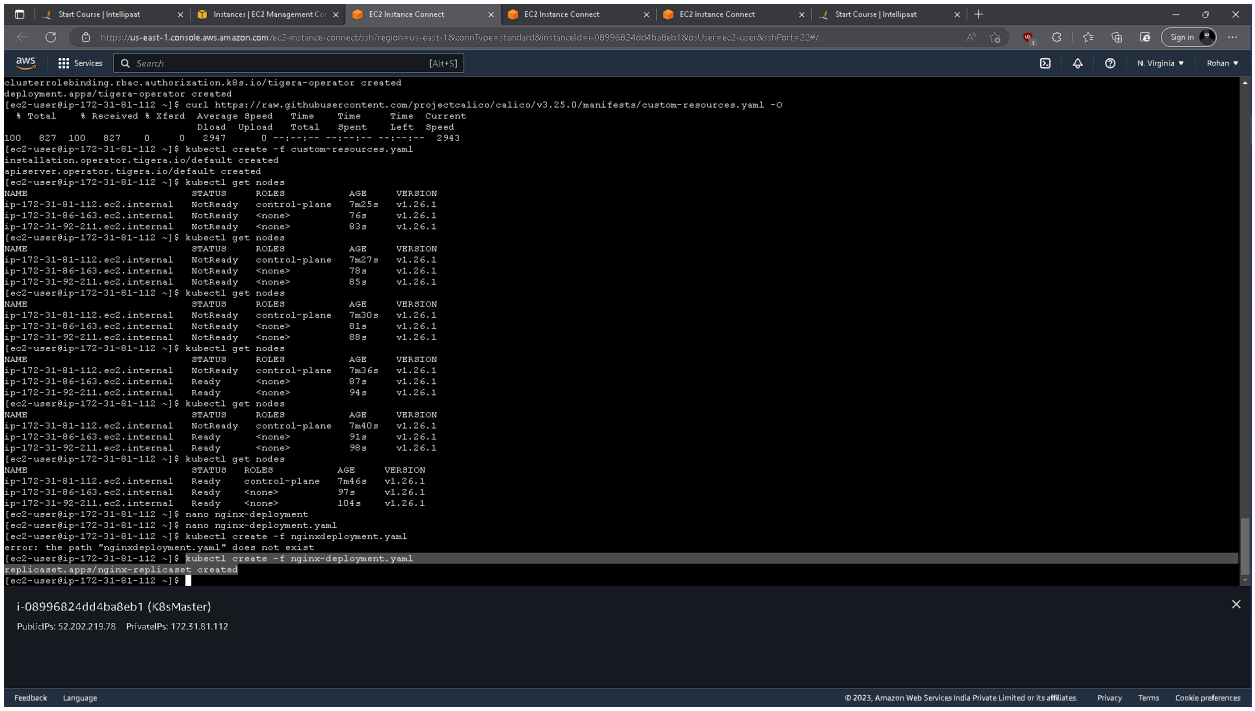


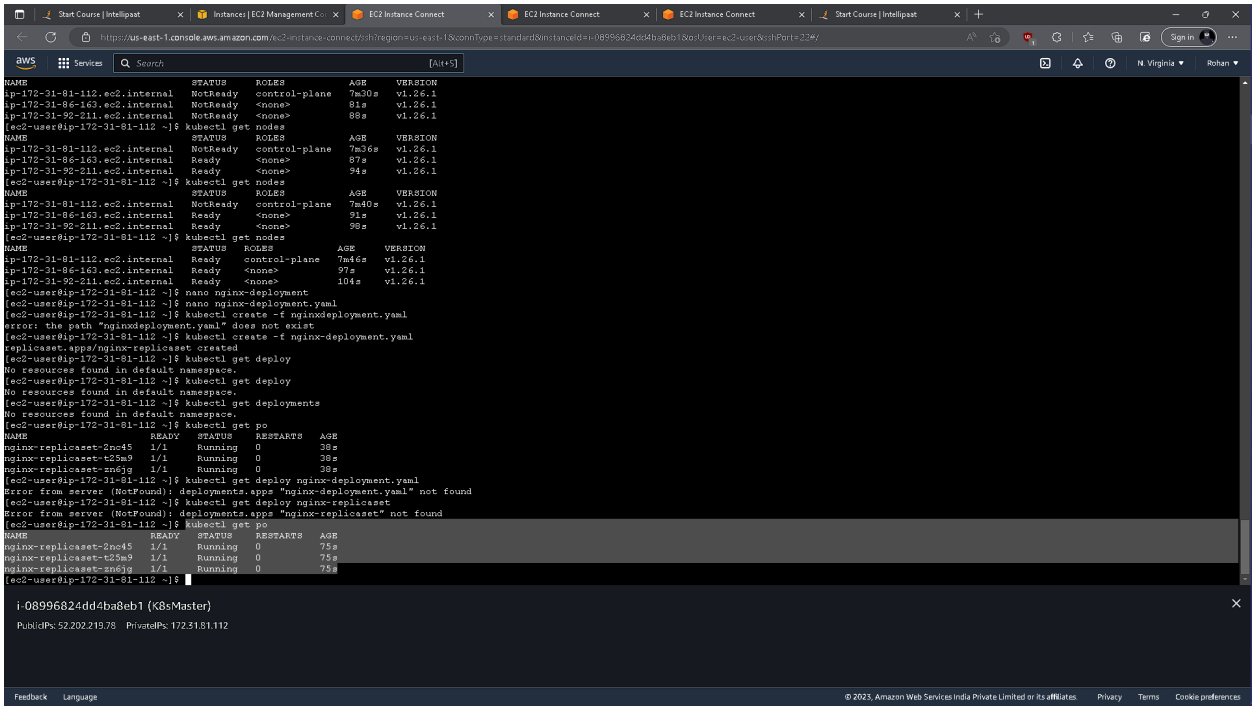
To create deployment:

nano nginx-deployment.yaml

```
#paste this:
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 1
  selector:
    matchLabels:
      app: nginx
      environment: production
  template:
    metadata:
      labels:
        app: nginx
        environment: production
    spec:
      containers:
      - name: nginx-container
        image: nginx
kubectl create -f nginx-deployment.yaml
kubectl get deploy   #will get you deployment pods
kubectl get pods
kubectl get rs
```


GNU nano 2.9.8
nginx-deployment
Modified
apiVersion: apps/v1
kind: ReplicaSet
metadata:
name: nginx-replicaset
labels:
app: nginx
spec:
replicas : 3
selector:
matchLabels:
environment: production
template:
metadata:
labels:
environment: production
spec:
containers:
- name: nginx-container
image: nginx
i-08996824dd4ba8eb1 (K8sMaster)
PublicIPs: 52.202.213.78 PrivateIPs: 172.31.81.112


clusterrolebinding.rbac.authorization.k8s.io/tigera-operator created
deployment.apps/tigera-operator created
[ec2-user@ip-172-31-81-112 ~]$ curl https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/custom-resources.yaml -O
[ec2-user@ip-172-31-81-112 ~]$ kubectl create -f custom-resources.yaml
installation.operator.tigera.io/default created
apiserver.operator.tigera.io/default created
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
[ec2-user@ip-172-31-81-112 ~]$ kubectl get nodes
[ec2-user@ip-172-31-81-112 ~]$ nano nginx-deployment
[ec2-user@ip-172-31-81-112 ~]$ nano nginx-deployment.yaml
[ec2-user@ip-172-31-81-112 ~]$ kubectl create -f nginxdeployment.yaml
error: the path "nginxdeployment.yaml" does not exist
[ec2-user@ip-172-31-81-112 ~]$ kubectl create -f nginx-deployment.yaml
replicaset.apps/nginx-replicaset created
[ec2-user@ip-172-31-81-112 ~]$
i-08996824dd4ba8eb1 (K8sMaster)
PublicIPs: 52.202.213.78 PrivateIPs: 172.31.81.112



kubectl get rs.

You can see three rs created.