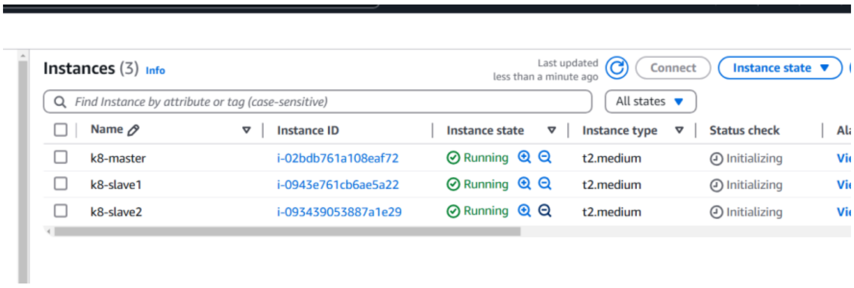
Module-9: Kubernetes Assignment - 1

You have been asked to:

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

Solution:

1) Create a Kubernetes cluster:
   a) Pre-requisites- Ubuntu OS & t2.medium instance type or higher
   b) Make sure all traffic is allowed on all instance



Instances (3) Info

Last updated less than a minute ago

□	Name	Instance ID	Instance state	Instance type	Status check	Al
□	k8-master	i-02bdb761a108eaf72	Running	t2.medium	Initializing	Vi
□	k8-slave1	i-0943e761cb6ae5a22	Running	t2.medium	Initializing	Vi
□	k8-slave2	i-093439053887a1e29	Running	t2.medium	Initializing	Vi

2) Run the following commands on both the master and worker nodes to prepare them for kubeadm:
   a) Update the system's package list and install necessary things:

```
sudo apt-get update
sudo apt install apt-transport-https curl -y
```

   b) Install containerd:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg

echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update
sudo apt-get install containerd.io -y
```

   c) Create containerd configuration:

```
sudo mkdir -p /etc/containerd
sudo containerd config default | sudo tee /etc/containerd/config.toml

sudo sed -i -e 's/SystemdCgroup = false/SystemdCgroup = true/g'
/etc/containerd/config.toml

sudo systemctl restart containerd
```

d) Install Kubernetes:

```
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-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
sudo systemctl enable --now kubelet
```

e) Disable swap

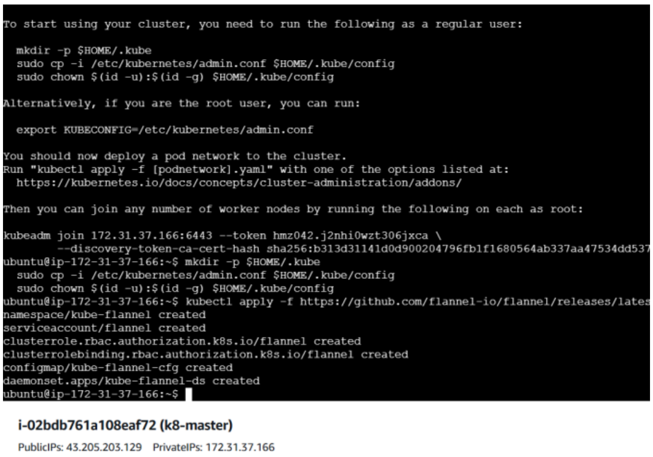
```
sudo swapoff -a
sudo modprobe br_netfilter
sudo sysctl -w net.ipv4.ip_forward=1
```

3) Execute ONLY on "Master Node":

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16

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:

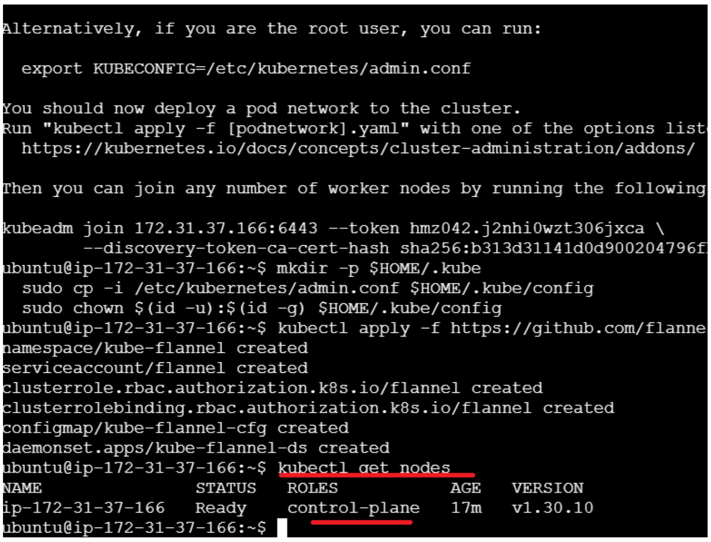
kubectl apply -f https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml



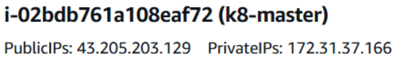
```
To start using your cluster, you need to run the following as a regular user:

  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 172.31.37.166:6443 --token hmz042.j2nhi0wzt306jxca \
        --discovery-token-ca-cert-hash sha256:b313d31141d0d900204796fb1f1680564ab337aa47534dd537
ubuntu@ip-172-31-37-166:~$ mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-37-166:~$ kubectl apply -f https://github.com/flannel-io/flannel/releases/lates
namespace/kube-flannel created
serviceaccount/flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@ip-172-31-37-166:~$
```

i-02bdb761a108eaf72 (k8-master)

PublicIPs: 43.205.203.129   PrivateIPs: 172.31.37.166



```
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 list
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following

kubeadm join 172.31.37.166:6443 --token hmz042.j2nhi0wzt306jxca \
        --discovery-token-ca-cert-hash sha256:b313d31141d0d900204796f
ubuntu@ip-172-31-37-166:~$ mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-37-166:~$ kubectl apply -f https://github.com/flanne
namespace/kube-flannel created
serviceaccount/flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@ip-172-31-37-166:~$ kubectl get nodes
NAME               STATUS   ROLES           AGE   VERSION
ip-172-31-37-166   Ready    control-plane   17m   v1.30.10
ubuntu@ip-172-31-37-166:~$
```



i-02bdb761a108eaf72 (k8-master)

PublicIPs: 43.205.203.129   PrivateIPs: 172.31.37.166

4) Execute on ALL of your Worker Node's:

a) Perform pre-flight checks: sudo kubeadm reset pre-flight checks

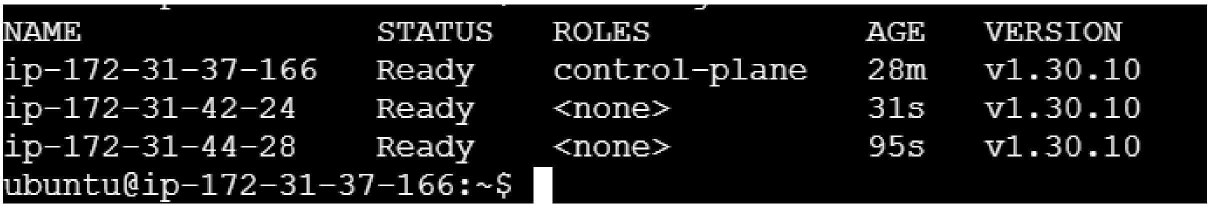
b) Paste the join command you got from the master node and append `--v=5` at the end but first use sudo su command to become root (avoid using sudo your-token).

sudo su

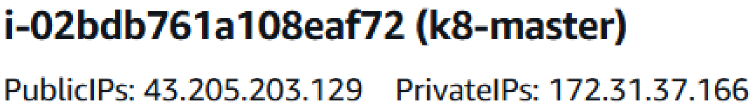
<your-token --v=5>

c) Verify Cluster Connection: **On Master Node:** kubectl get nodes

5) My 3 node cluster is ready



```
NAME               STATUS   ROLES           AGE   VERSION
ip-172-31-37-166   Ready    control-plane   28m   v1.30.10
ip-172-31-42-24    Ready    <none>          31s   v1.30.10
ip-172-31-44-28    Ready    <none>          95s   v1.30.10
ubuntu@ip-172-31-37-166:~$
```

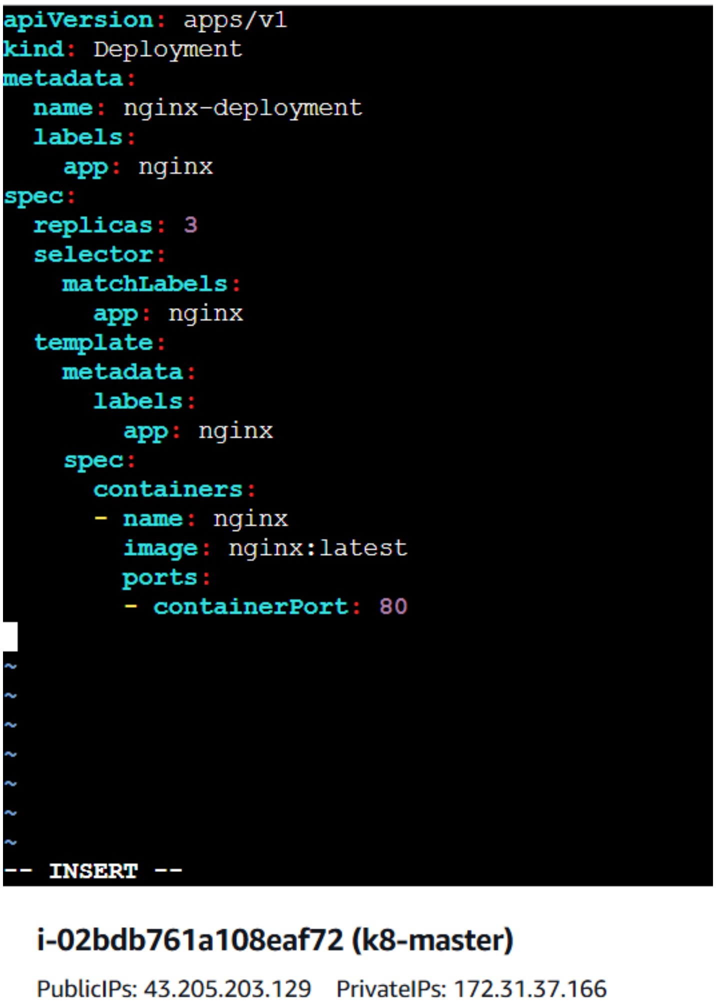


i-02bdb761a108eaf72 (k8-master)

PublicIPs: 43.205.203.129   PrivateIPs: 172.31.37.166

6) Creating a nginx deployment of 3 replicas

a) Create a file named nginx-deployment.yml with the following content:

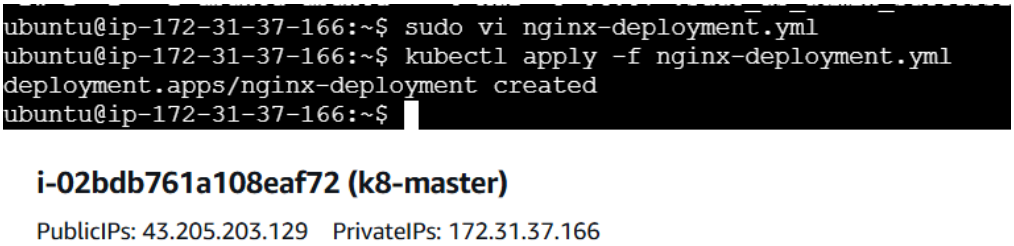


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80

~
~
~
~
~
~
~
-- INSERT --
```

i-02bdb761a108eaf72 (k8-master)

PublicIPs: 43.205.203.129 PrivateIPs: 172.31.37.166

b) Use the kubectl command to apply the deployment file to your Kubernetes cluster:
kubectl apply -f nginx-deployment.yml

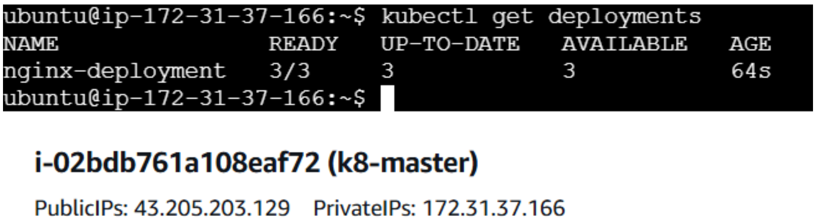


```
ubuntu@ip-172-31-37-166:~$ sudo vi nginx-deployment.yml
ubuntu@ip-172-31-37-166:~$ kubectl apply -f nginx-deployment.yml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-37-166:~$
```

i-02bdb761a108eaf72 (k8-master)

PublicIPs: 43.205.203.129 PrivateIPs: 172.31.37.166

c) Verify that the deployment and replicas have been created:
kubectl get deployments

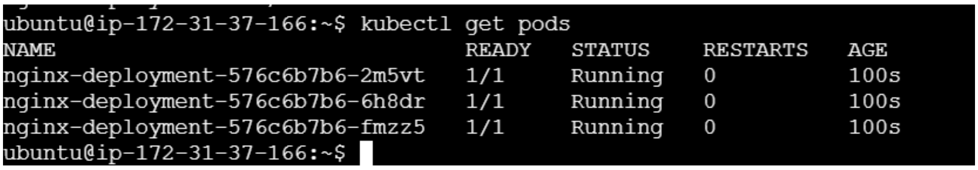


```
ubuntu@ip-172-31-37-166:~$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           64s
ubuntu@ip-172-31-37-166:~$
```

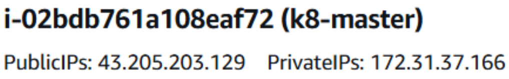
i-02bdb761a108eaf72 (k8-master)

PublicIPs: 43.205.203.129 PrivateIPs: 172.31.37.166

kubectl get pods



```
ubuntu@ip-172-31-37-166:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-576c6b7b6-2m5vt   1/1     Running   0          100s
nginx-deployment-576c6b7b6-6h8dr   1/1     Running   0          100s
nginx-deployment-576c6b7b6-fmzz5   1/1     Running   0          100s
ubuntu@ip-172-31-37-166:~$
```



i-02bdb761a108eaf72 (k8-master)

PublicIPs: 43.205.203.129   PrivateIPs: 172.31.37.166