**Step 1: Set Up a 3-Node Kubernetes Cluster**

1. **Set up Kubernetes Cluster**: You can create a Kubernetes cluster with 3 nodes on a AWS Cloud
2. **Install software of all servers**: kubeadm,and kubectl on mater node

**Using kubeadm :**

1. Initialize the master node:

sudo kubeadm init --pod-network-cidr=192.168.0.0/16

1. Configure kubectl for the root user:

mkdir -p $HOME/.kube

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

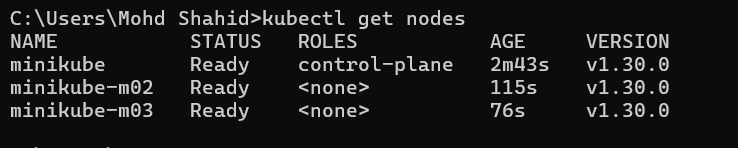
sudo chown $(id -u):$(id -g) $HOME/.kube/config

1. Install a Pod network add-on (like Calico):

kubectl apply -f https://docs.projectcalico.org/v3.14/manifests/calico.yaml

1. Join the other nodes: On each worker node, run the join command (from the output of kubeadm init) to join them to the cluster. If you lost this command, you can regenerate it using:

kubeadm token create --print-join-command



**Step 2: Deploy NGINX with 3 Replicas**

Once the Kubernetes cluster is set up, you can create an NGINX deployment with 3 replicas.

1. **Create a Deployment YAML File** (e.g., nginx-deployment.yaml):

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

spec:

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:latest

ports:

- containerPort: 80

1. **Apply the Deployment**: Use kubectl to apply the deployment:

kubectl apply -f nginx-deployment.yaml

1. **Verify the Deployment**: Check that the deployment and the replicas are running:

kubectl get deployments

kubectl get pods

