**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

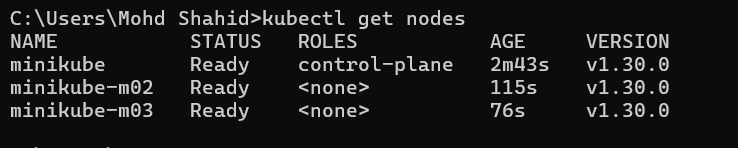
sudo chown $(id -u):$(id -g) $HOME/.kube/config

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



**1. Deploy an NGINX Deployment with 3 Replicas**

Create a file named nginx-deployment.yaml 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

Apply this configuration to create the deployment:

kubectl apply -f nginx-deployment.yaml

**2. Create an NGINX Service of Type ClusterIP**

Create a file named nginx-service.yaml with the following content:

apiVersion: v1

kind: Service

metadata:

name: nginx-service

spec:

type: ClusterIP

selector:

app: nginx

ports:

- port: 80

targetPort: 80

Apply this configuration to create the service:

kubectl apply -f nginx-service.yaml

**3. Create an Ingress Service**

deploy the NGINX Ingress Controller:

1. **Install NGINX Ingress Controller:**

Create a file named nginx-ingress-controller.yaml with the following content:

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-ingress-controller

namespace: kube-system

spec:

replicas: 1

selector:

matchLabels:

app: nginx-ingress-controller

template:

metadata:

labels:

app: nginx-ingress-controller

spec:

containers:

- name: nginx-ingress-controller

image: quay.io/kubernetes-ingress-controller/nginx-ingress-controller:latest

ports:

- containerPort: 80

- containerPort: 443

args:

- /nginx-ingress-controller

- --default-backend-service=kube-system/default-http-backend

Apply the configuration:

kubectl apply -f nginx-ingress-controller.yaml

1. **Create an Ingress Resource:**

Create a file named nginx-ingress.yaml with the following content:

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: nginx-ingress

spec:

rules:

- host: nginx.local

http:

paths:

- path: /

pathType: Prefix

backend:

service:

name: nginx-service

port:

number: 80

Apply the configuration:

kubectl apply -f nginx-ingress.yaml

**4. Verify Everything**

1. **Check the Deployment and Service:**

kubectl get deployments

kubectl get services

1. **Check Ingress:**

Make sure the Ingress resource is correctly set up:

kubectl get ingress

