**EKS (elastic Kubernetes service)**

**In eks aws allows us to only create and configure working nodes**

**It manages master node in the backend**

**To create EKS (be in IAM user account)**

**(pre-requisites)**

* **Create and launch instances**
* **Install aws cli and configure it**
* **Install kubectl**

sudo apt update

sudo apt install curl unzip -y

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

unzip awscliv2.zip

sudo ./aws/install

aws --version

sudo apt-get update

sudo apt-get install -y apt-transport-https ca-certificates curl gnupg

sudo mkdir -p -m 755 /etc/apt/keyrings

curl -fsSL https:*//pkgs.k8s.io/core:/stable:/v1.32/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg*

sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg *# allow unprivileged APT programs to read this keyring*

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.32/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo chmod 644 /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update

sudo apt-get install -y kubectl

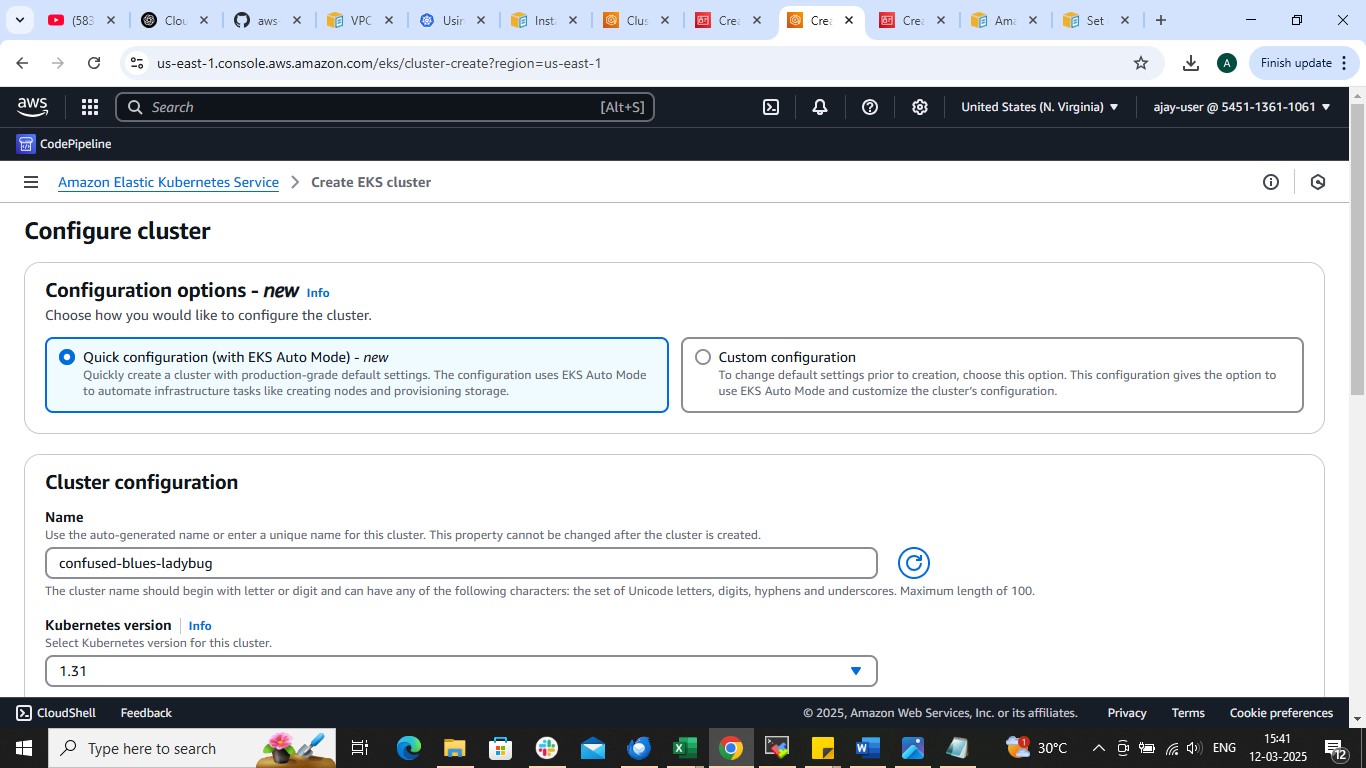
kubectl version --client

**Now open Aws console**

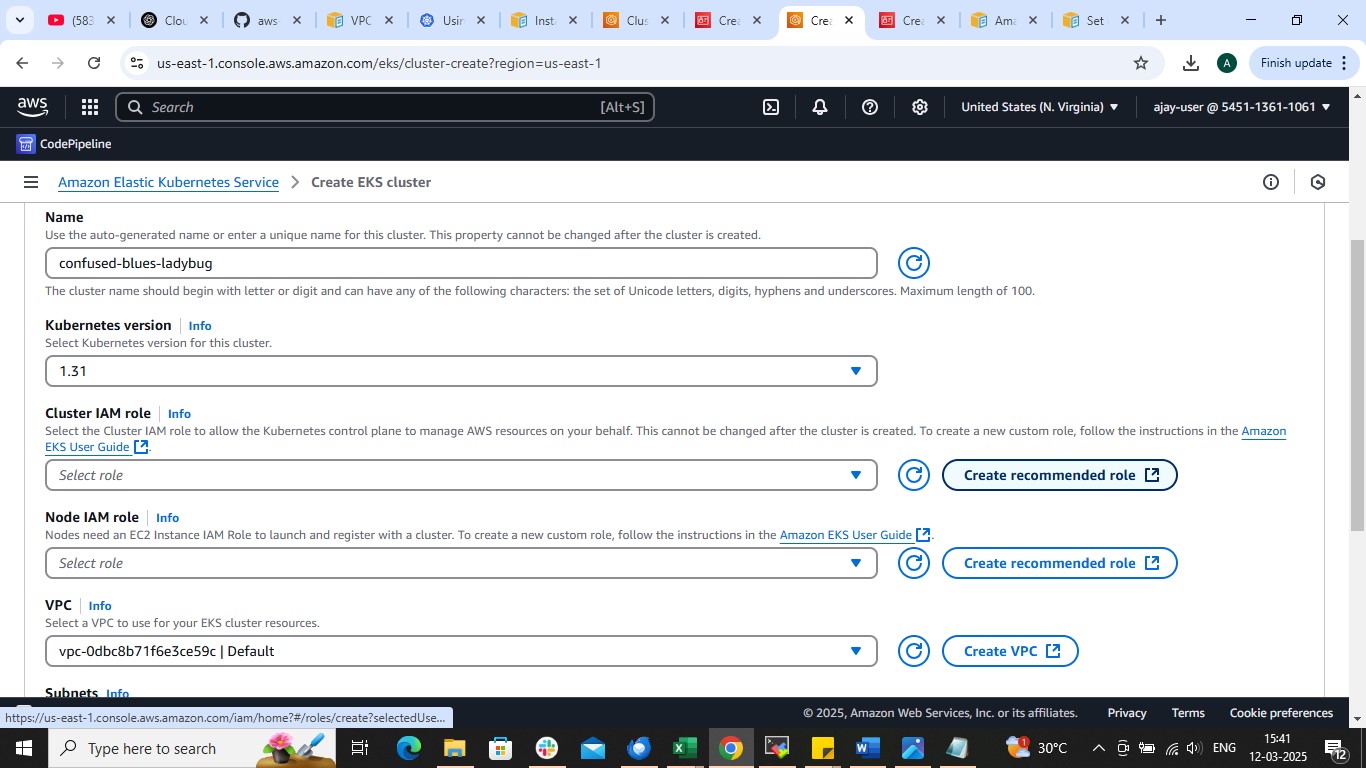
**Search EKS🡺eks**

**🡺clusters**

**Choose quick configuration or custom configuration (in CC we need configure everything including type instance ,vpc and subnet .After this create node-group for to create working node)**

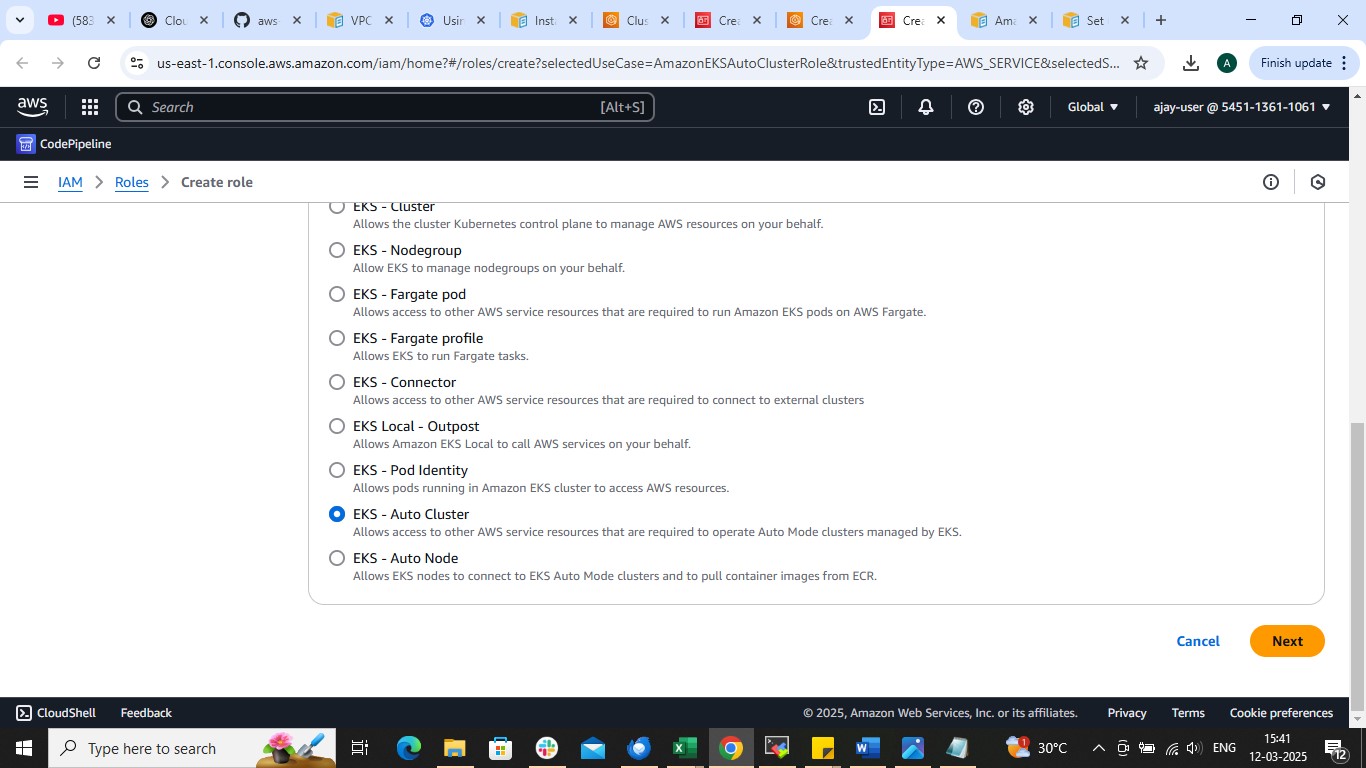
****

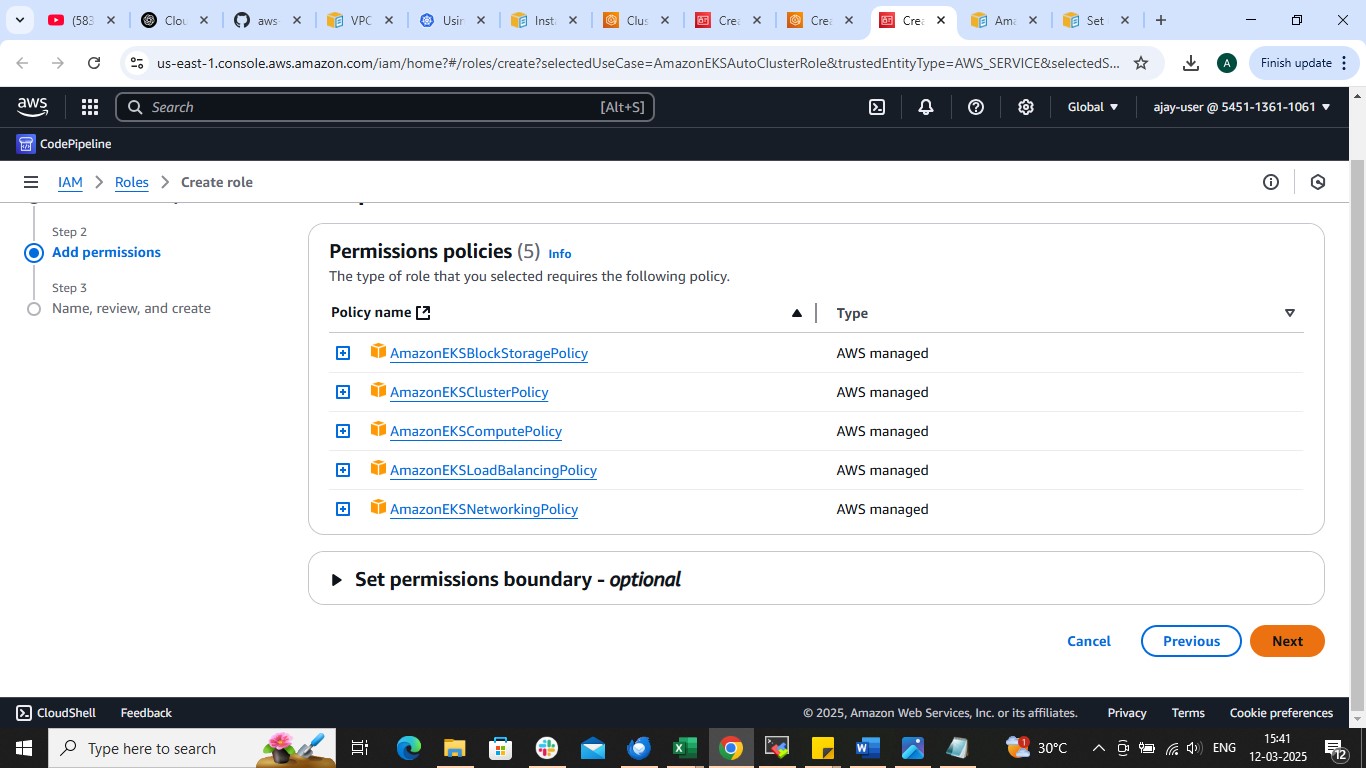
**Create IAM role for Cluster and Node**

****

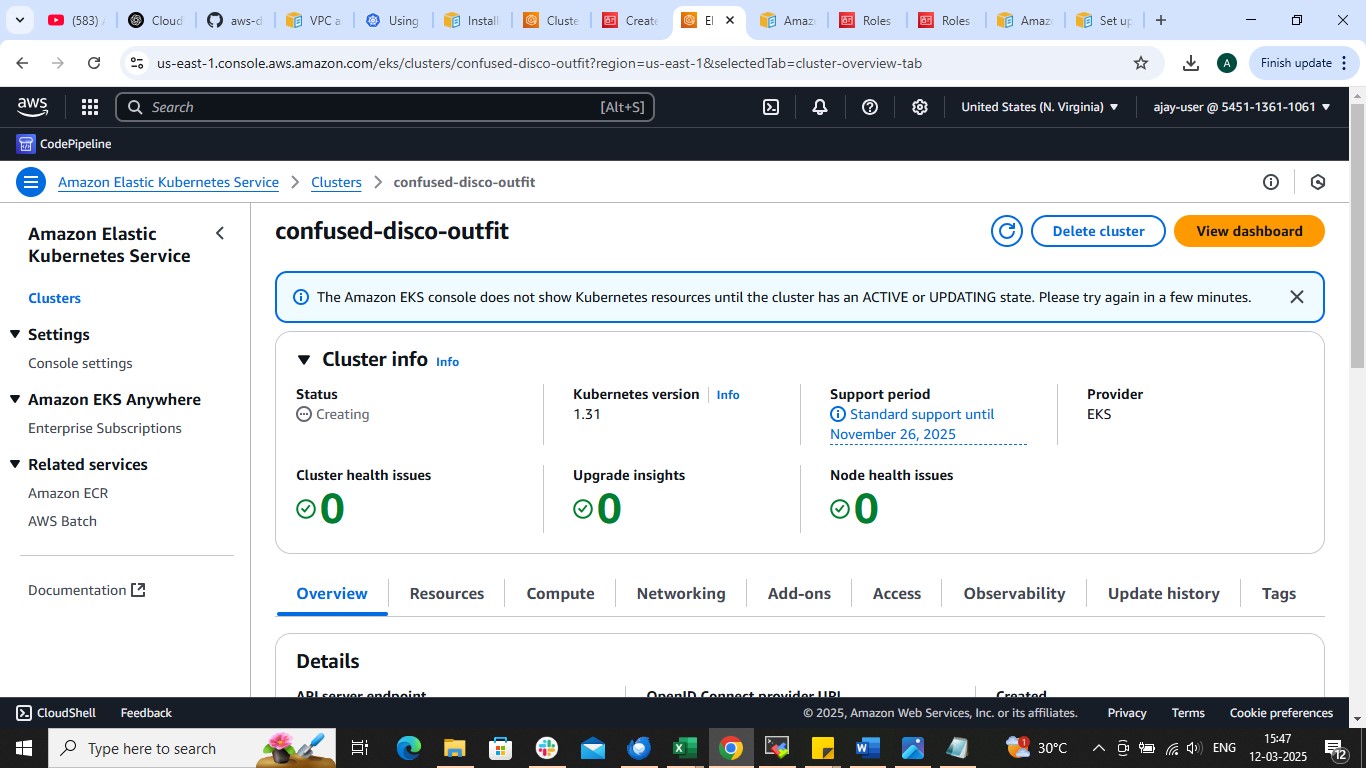
**Role for cluster**

****

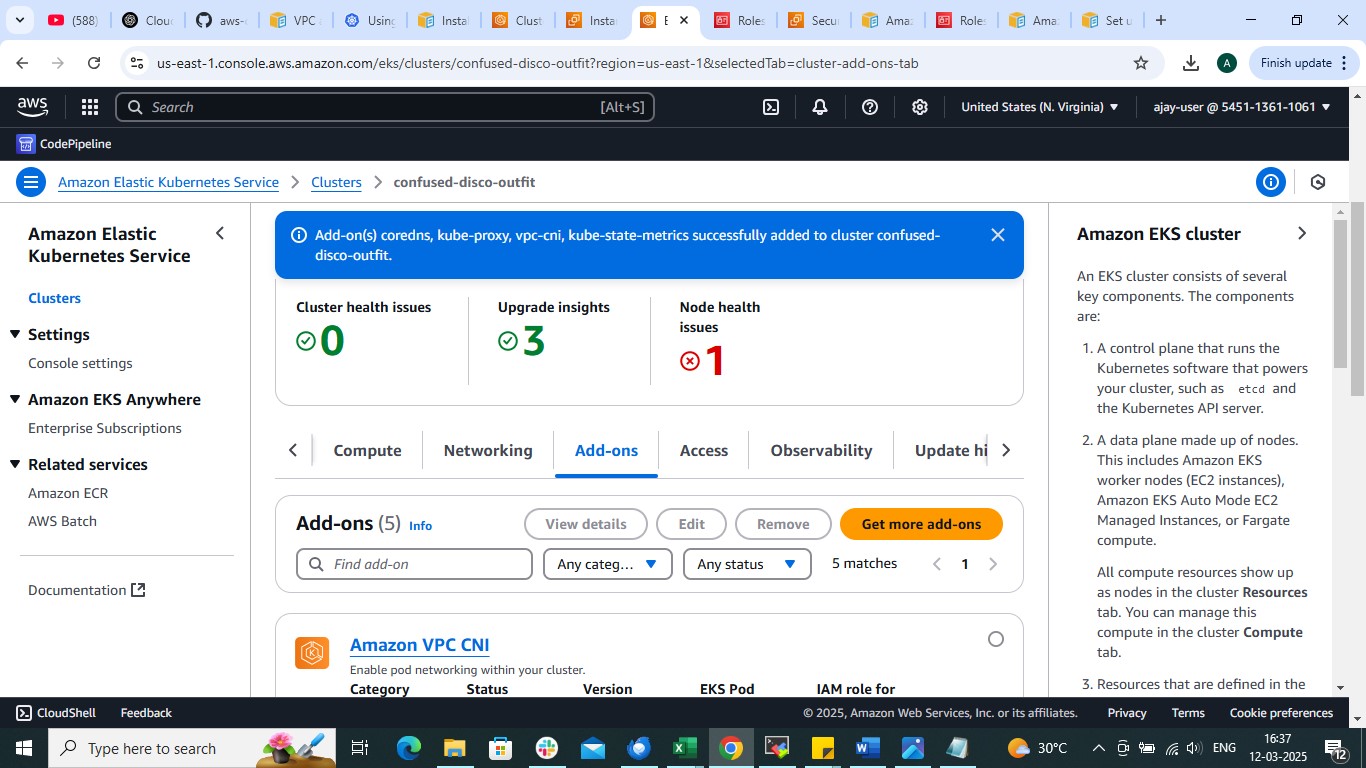
****

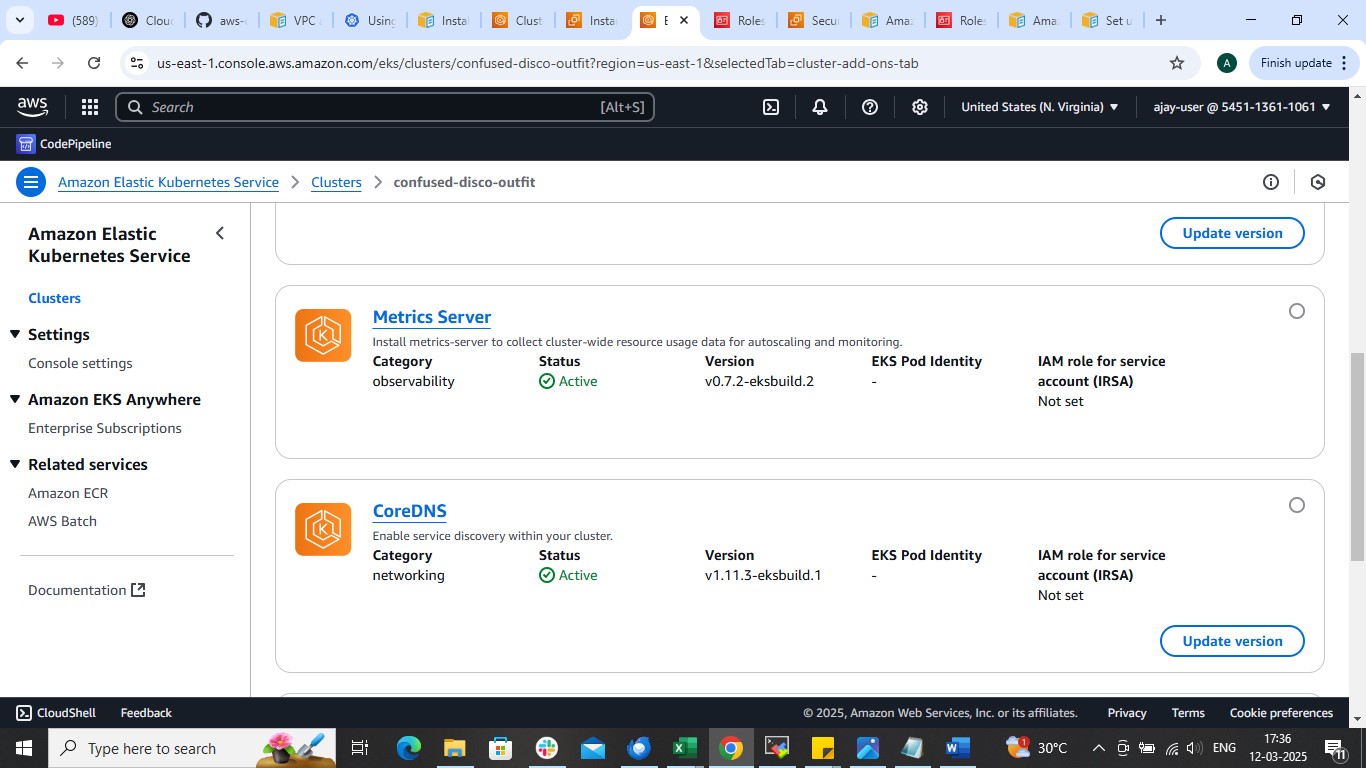
****

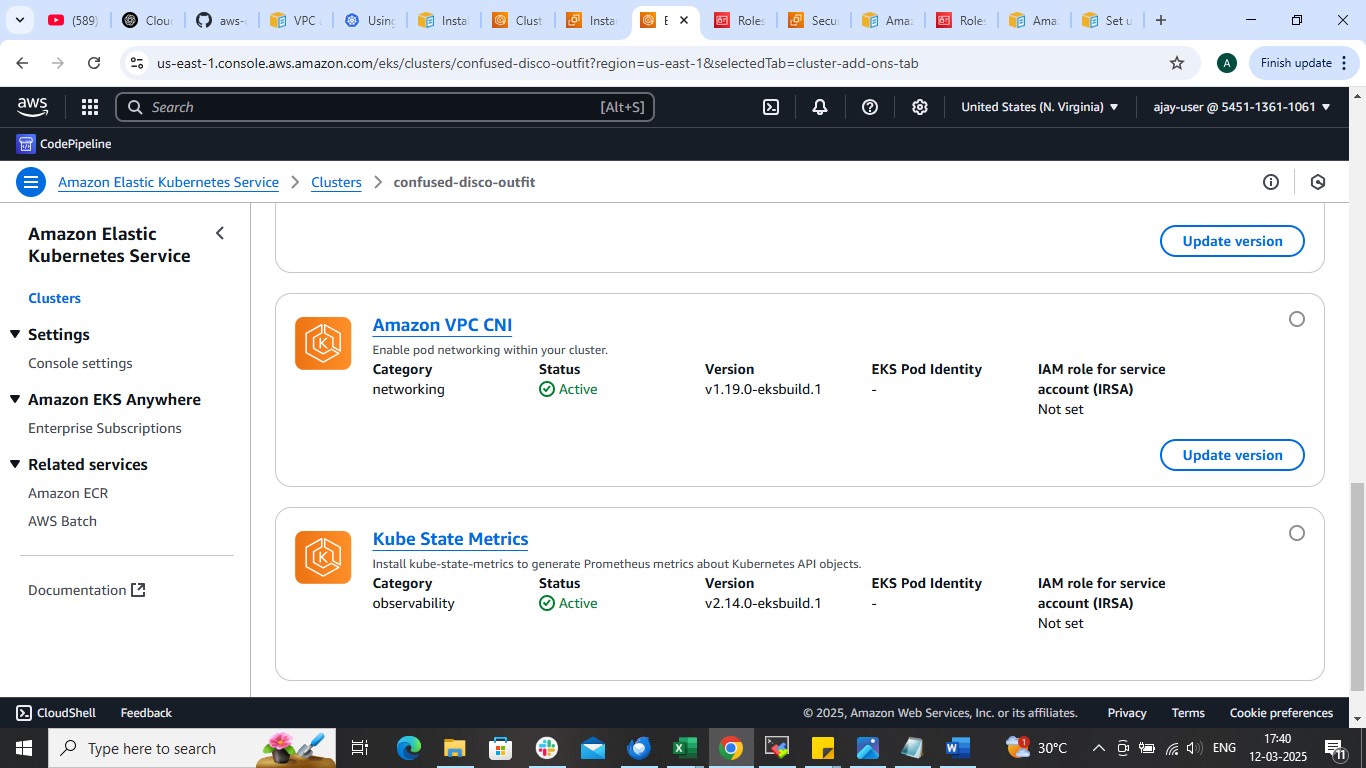
**Rest is same 🡺 create**

****

**Now 🡺cluster name 🡺add-ons (to add plugins)**

****

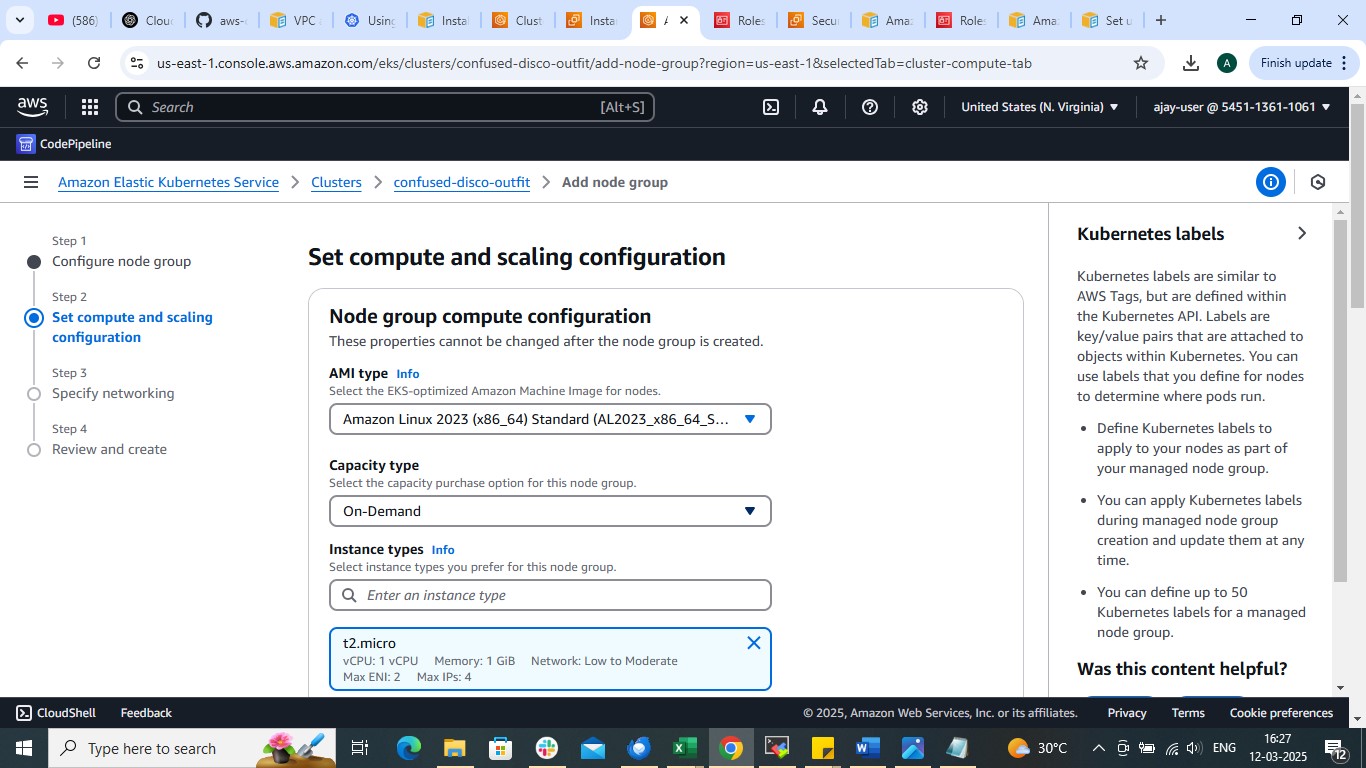
****

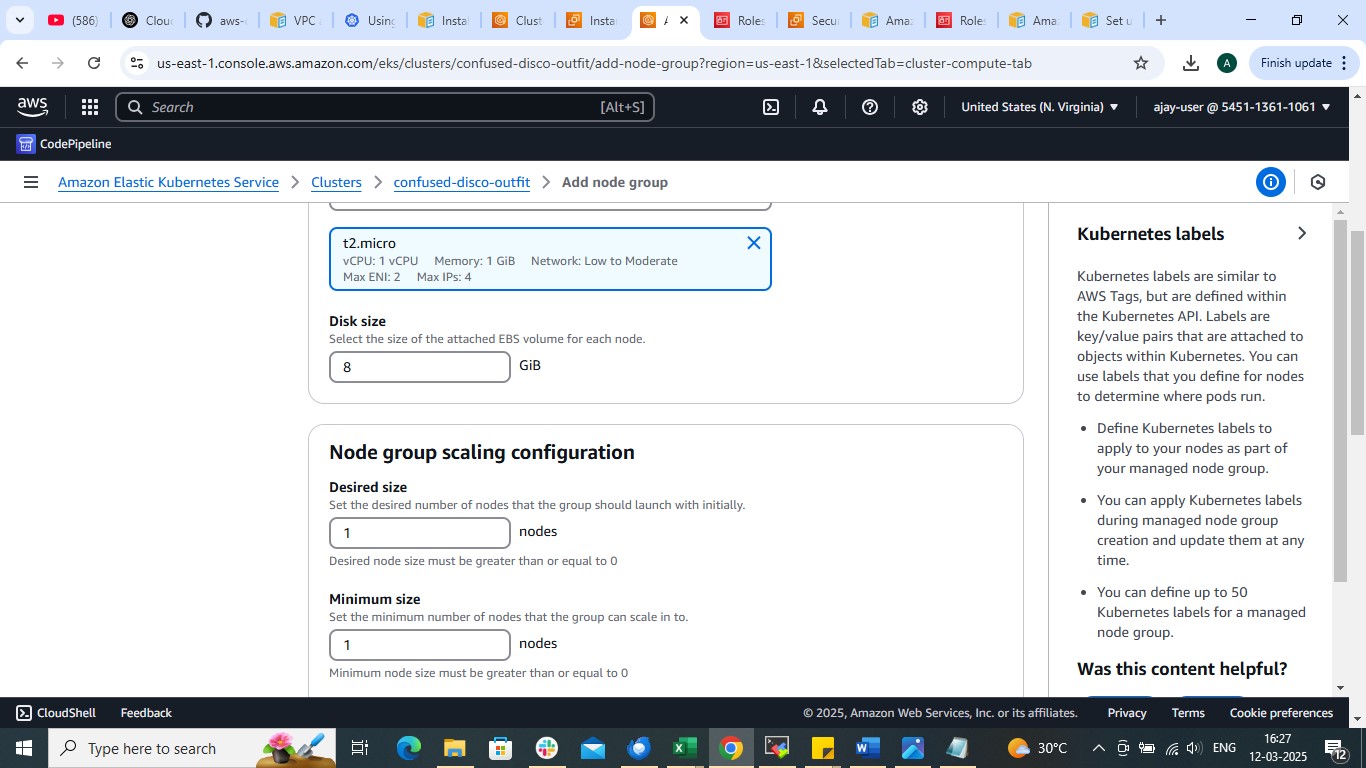
****

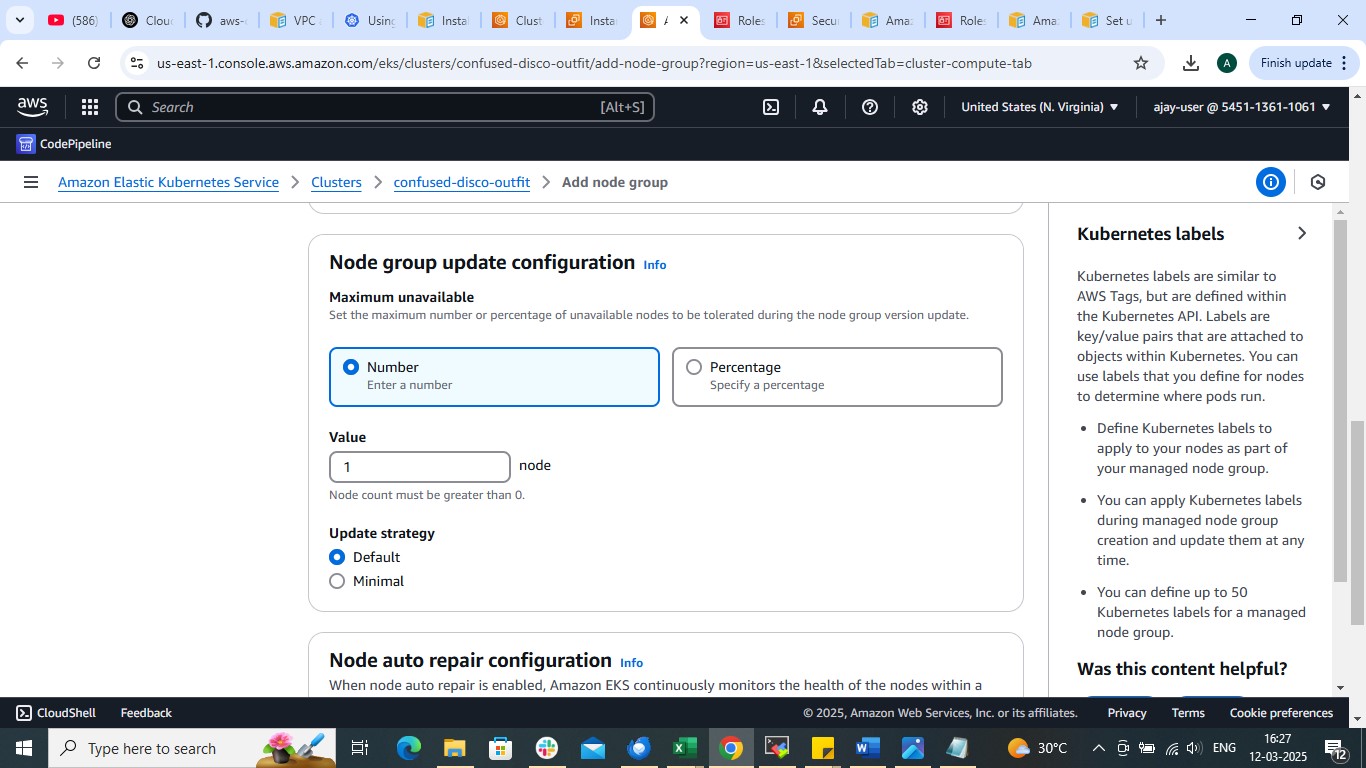
**BY this method we will get node(instance) configured by aws automatically**

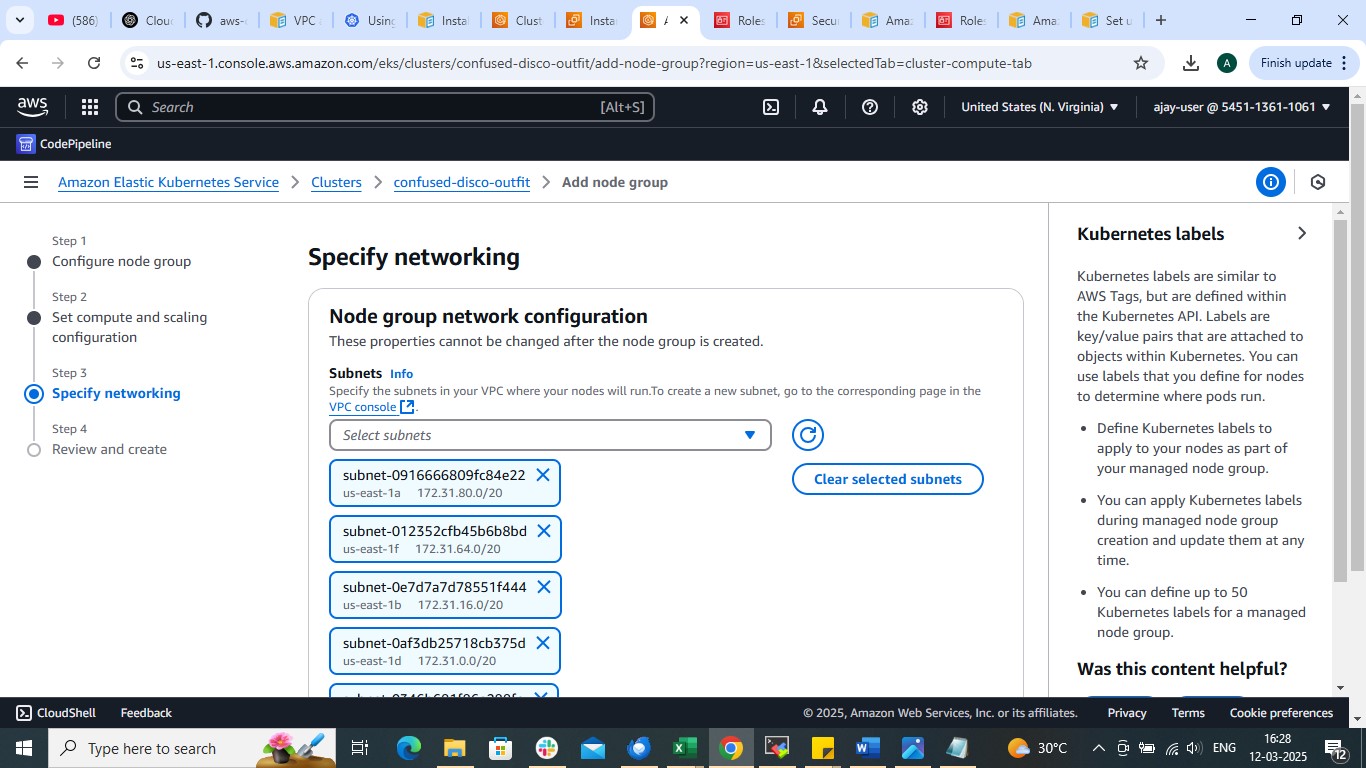
**Now to add more instance (node) create node group**

**🡺cluster🡺compute🡺add node group**

****

****

****

****

**🡺create**

**Wait for few minutes . Node group along with node will get created**

**Now Open instance in which you configured aws cli and kubectl**

**run command**

aws eks update-kubeconfig --name cluster-name --region region name

**Now run**

kubectl get nodes

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

labels:

app: nginx

spec:

replicas: 2

selector:

matchLabels:

app: nginx

template:

metadata:

name: test-pod

labels:

app: nginx

spec:

containers:

- name: nginx-container

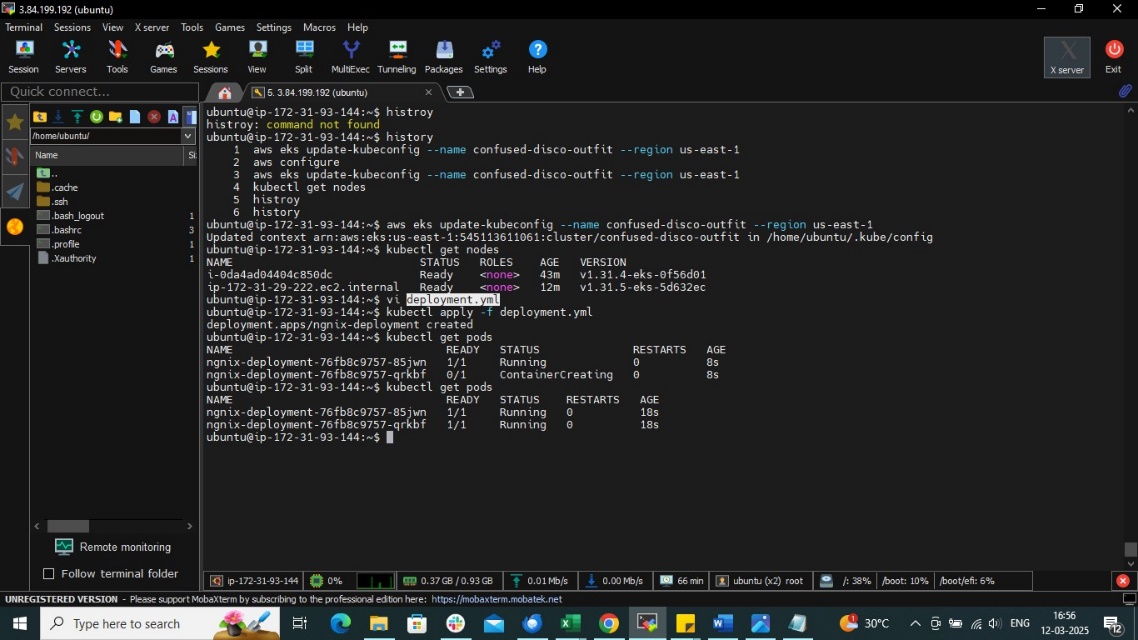
image: nginx

ports:

- containerPort: 80

**Run**

Kubectl apply -f deployment.yml

****

**Now create service.yml file**

apiVersion: v1

kind: Service

metadata:

name: nginx-service

spec:

selector:

app: nginx

ports:

- protocol: TCP

port: 80

targetPort: 80

type: NodePort

**run**

kubectl apply -f service.yml

kubectl get svc

**copy servive’s nodeport**

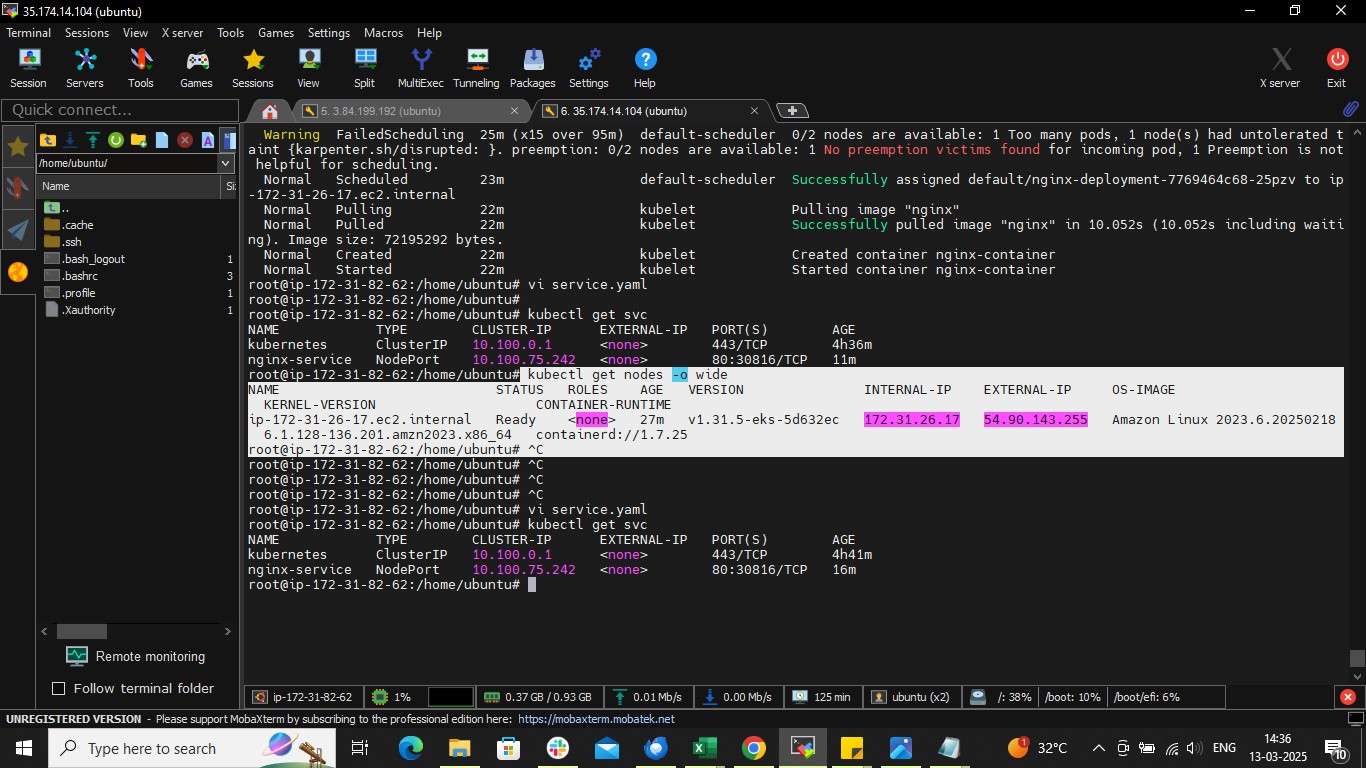
****

**To access our application from browser**

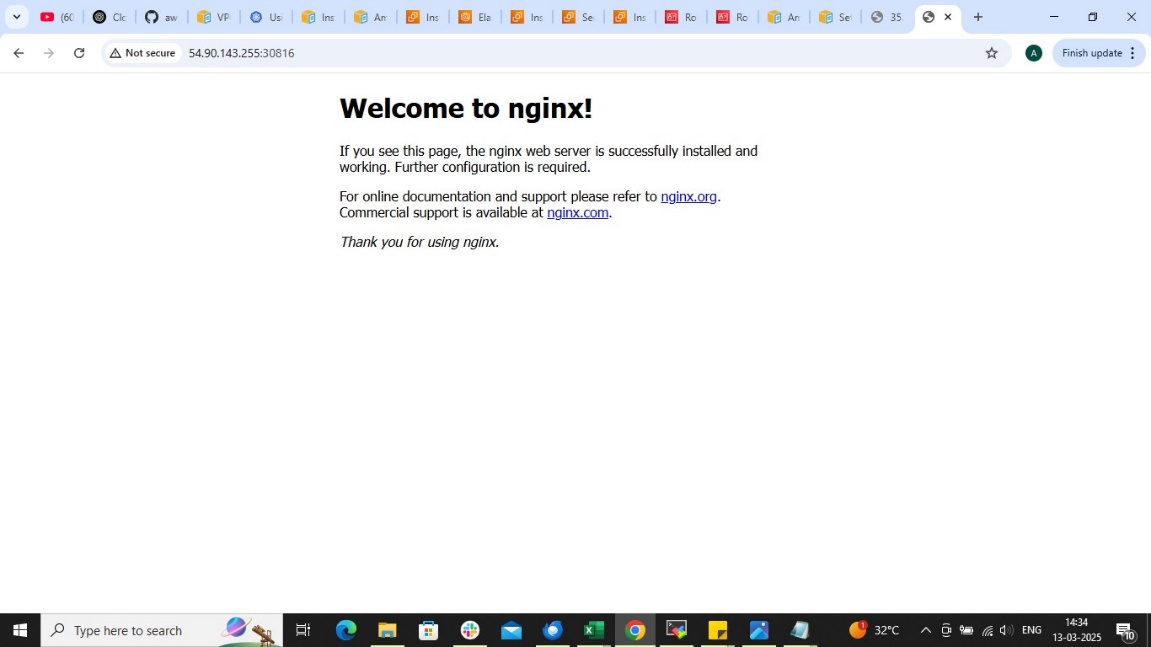
**Run**

Kubectl get nodes -o wide

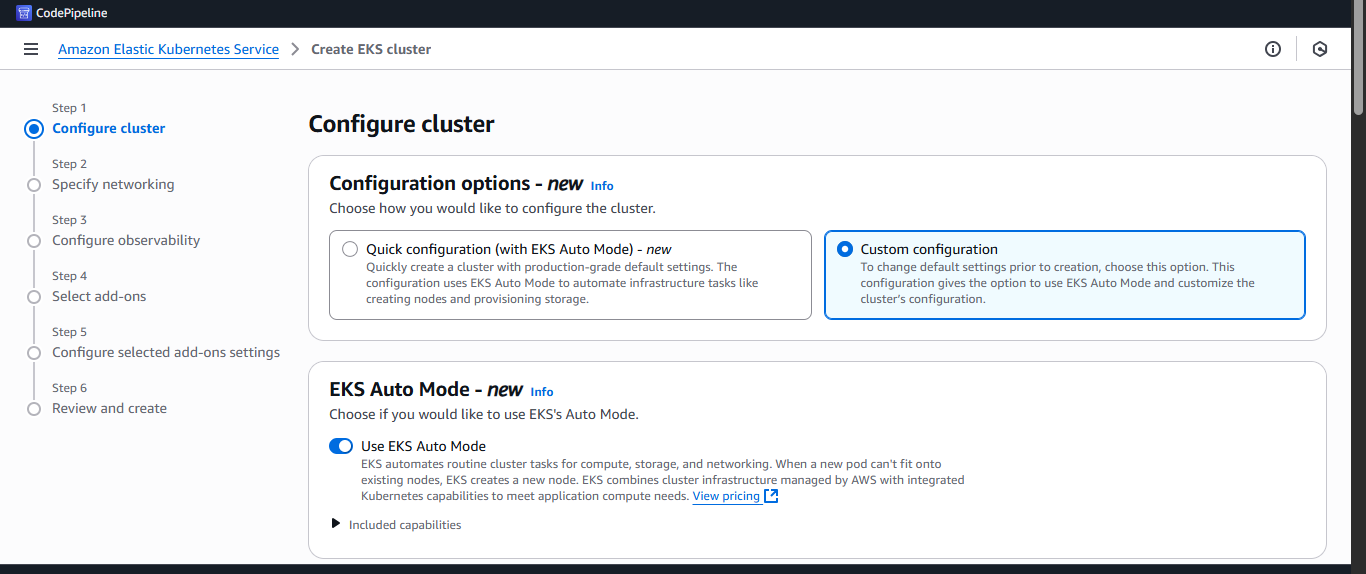
**Copy the external Ip of working node**

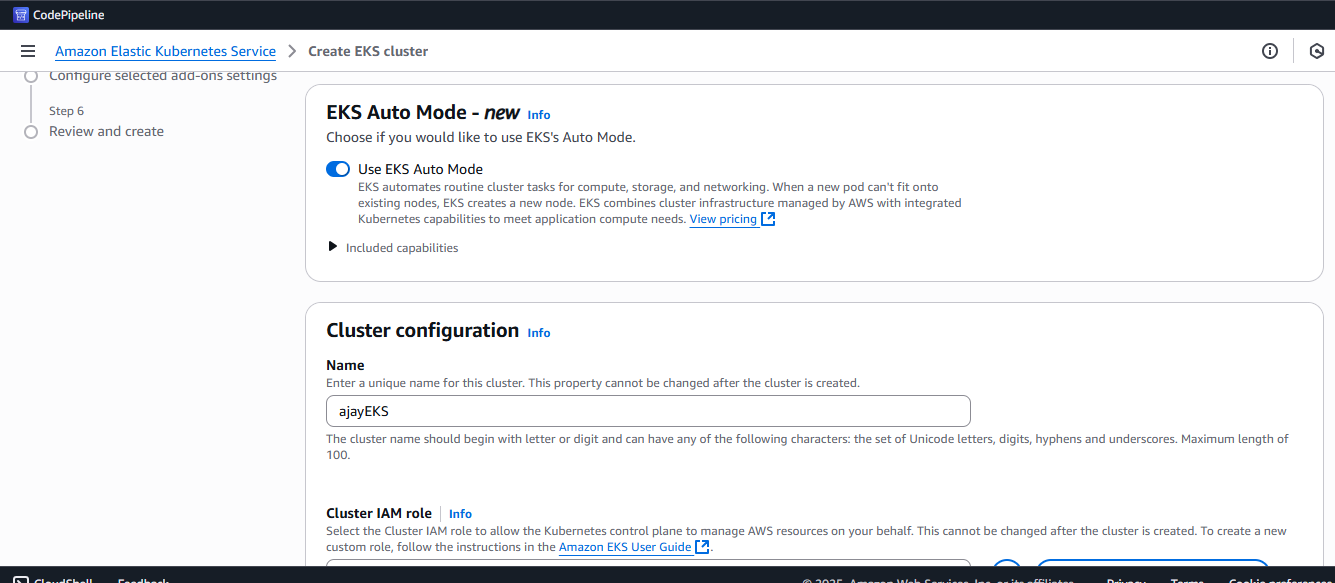
****

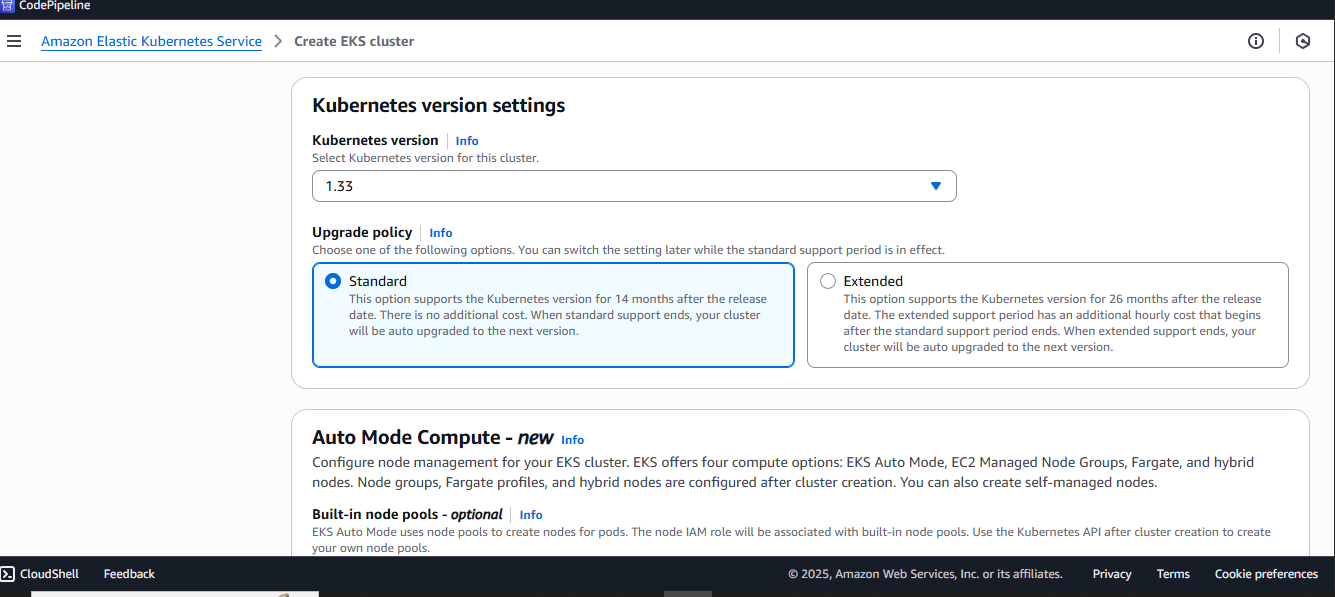
**Open the browser and paste in browser along with nodeport**

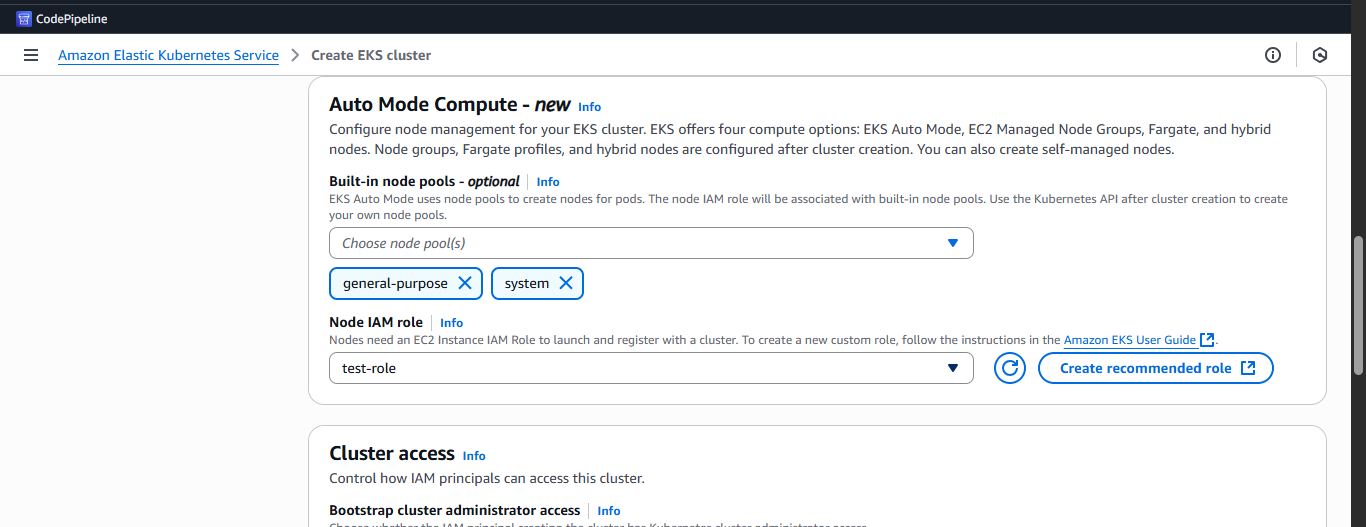
****

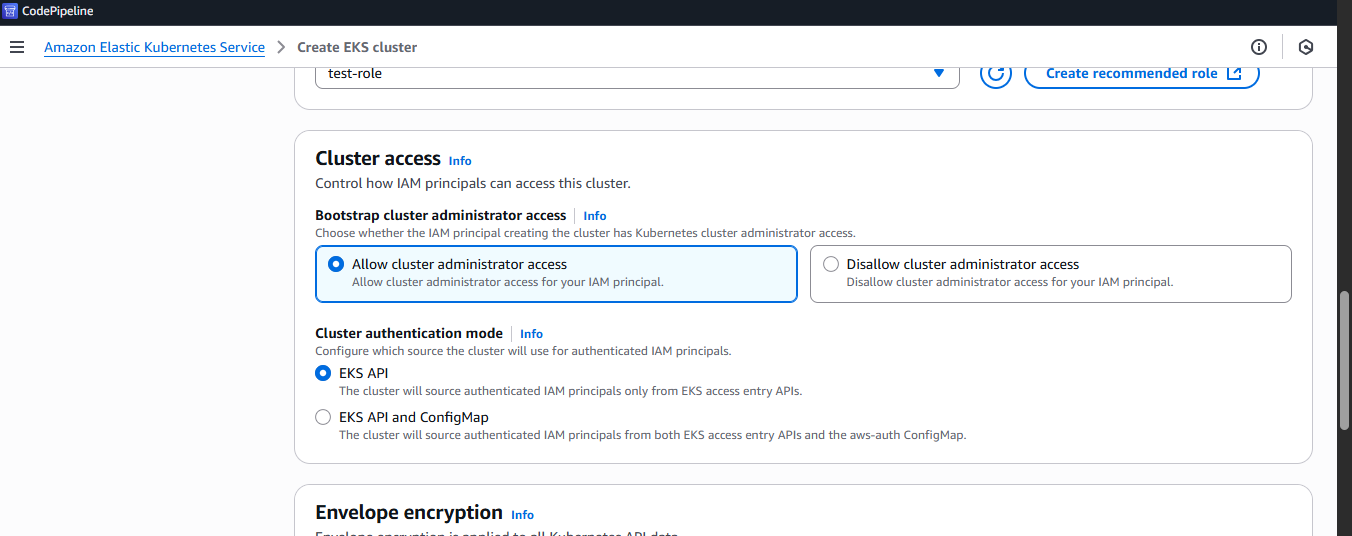
**If any error occurs while accessing application in browser then check the security group of working node**

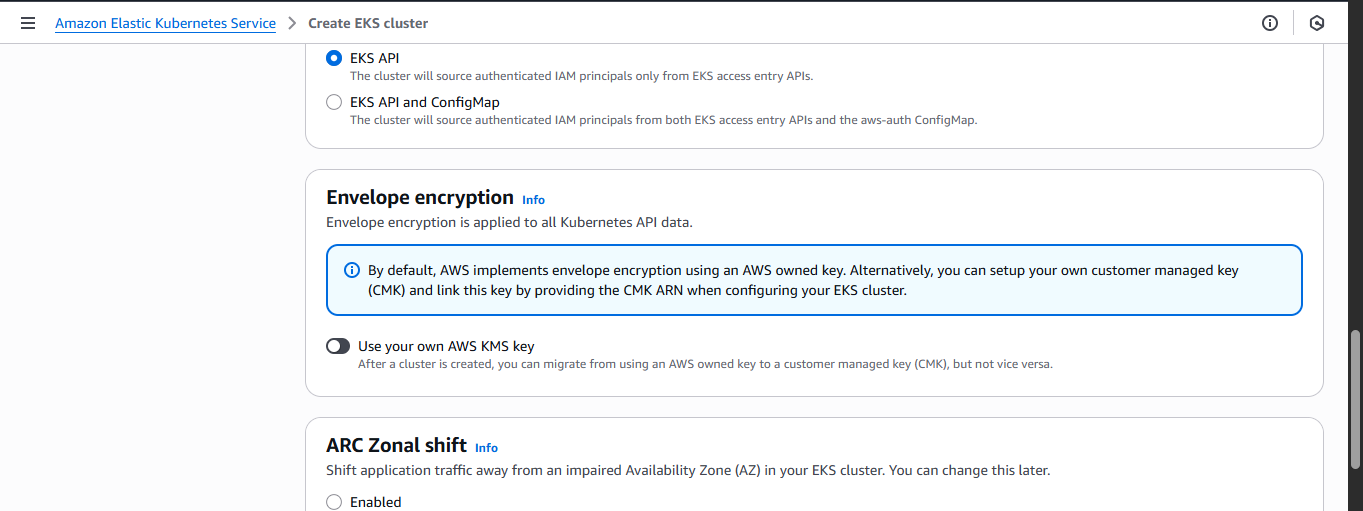
****

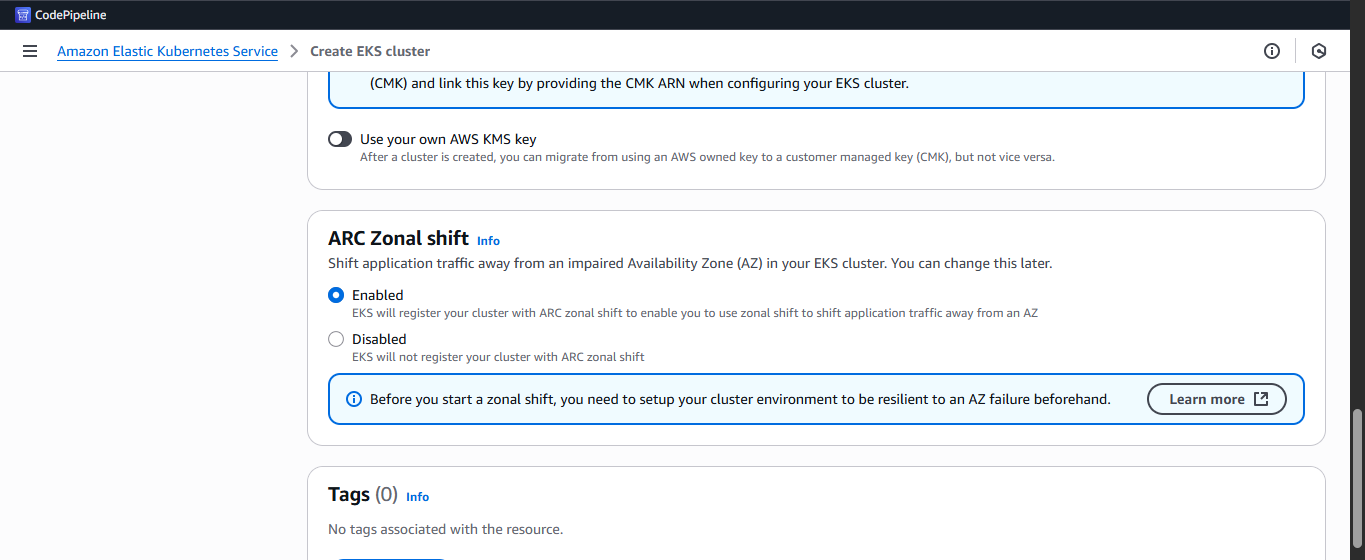
****

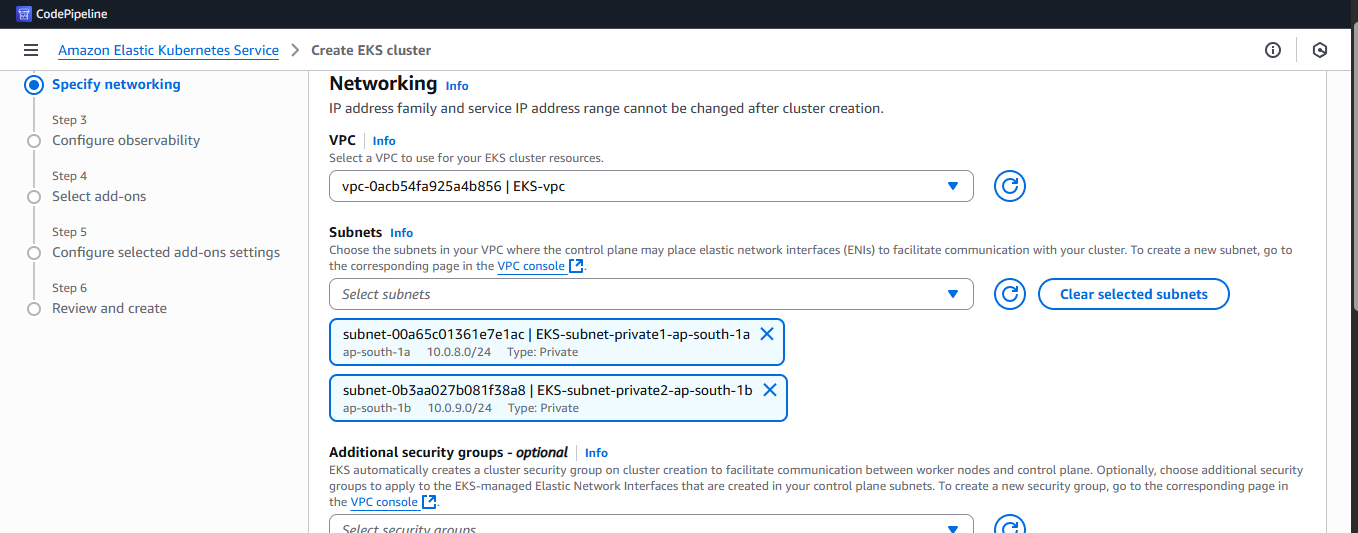
****

****

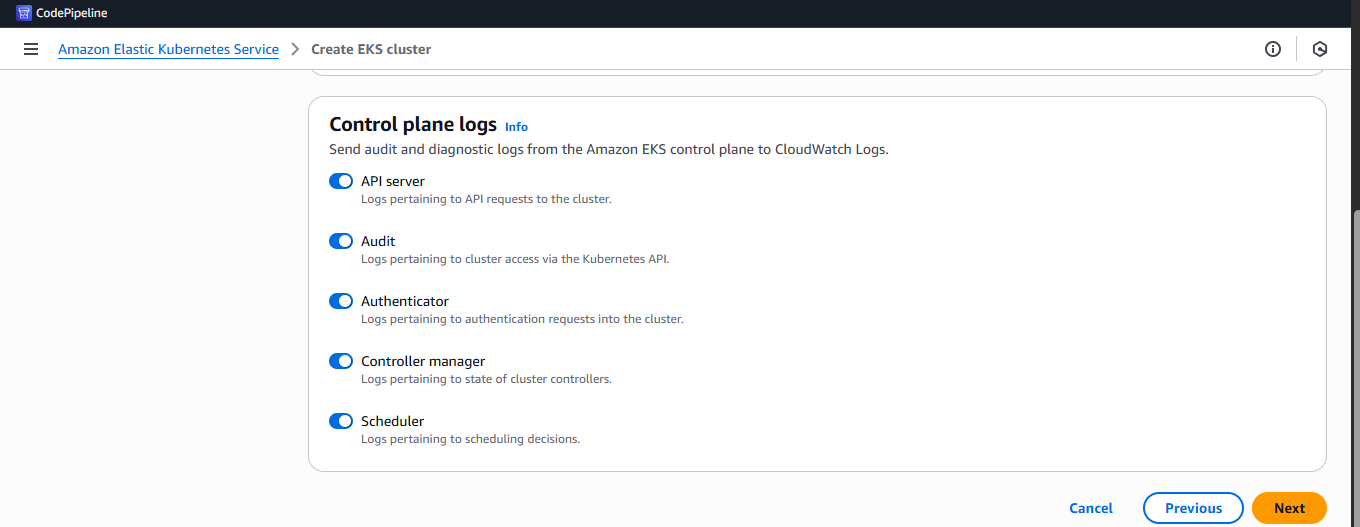
****

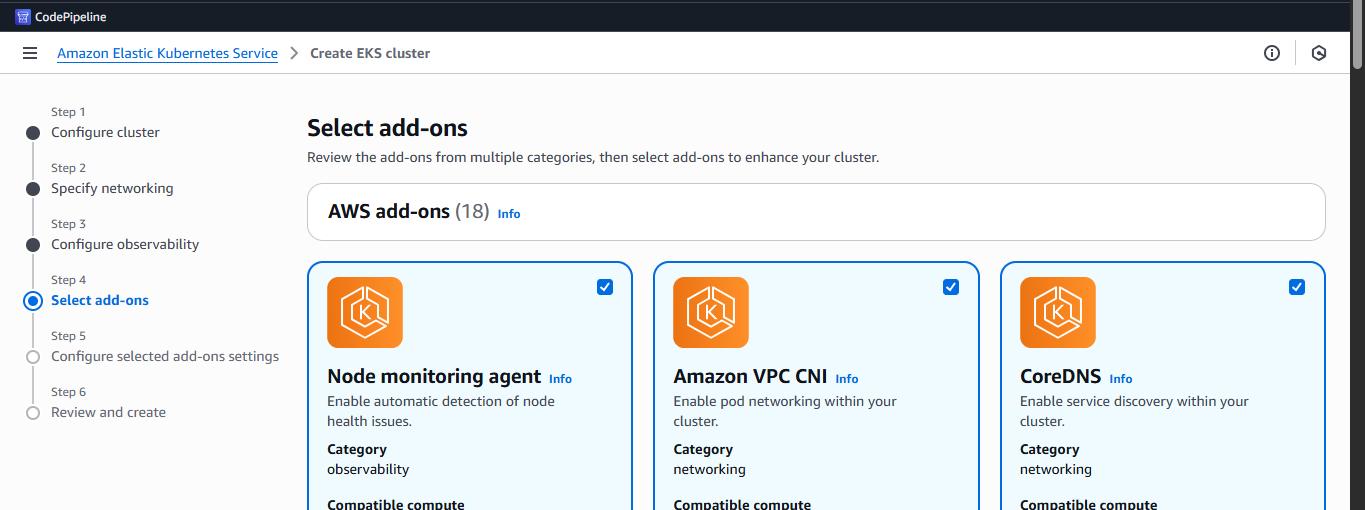
****

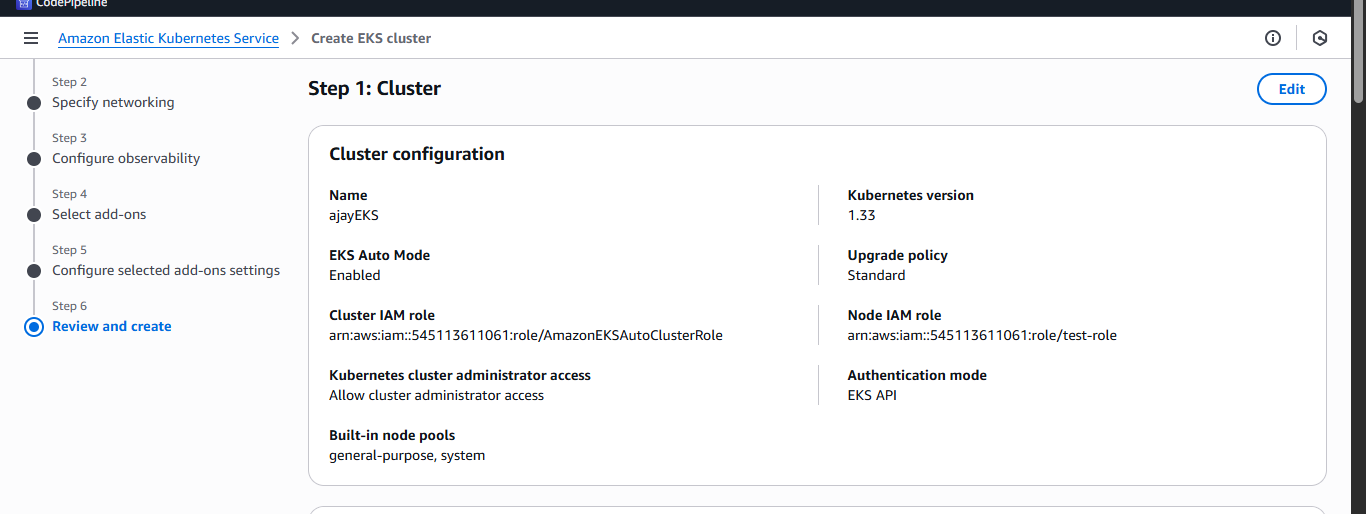
****

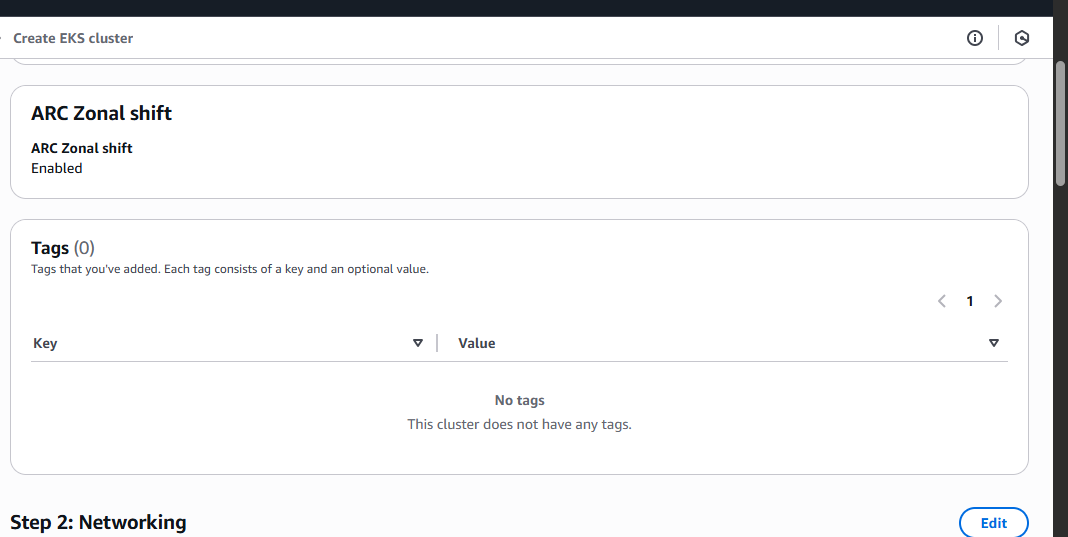
****

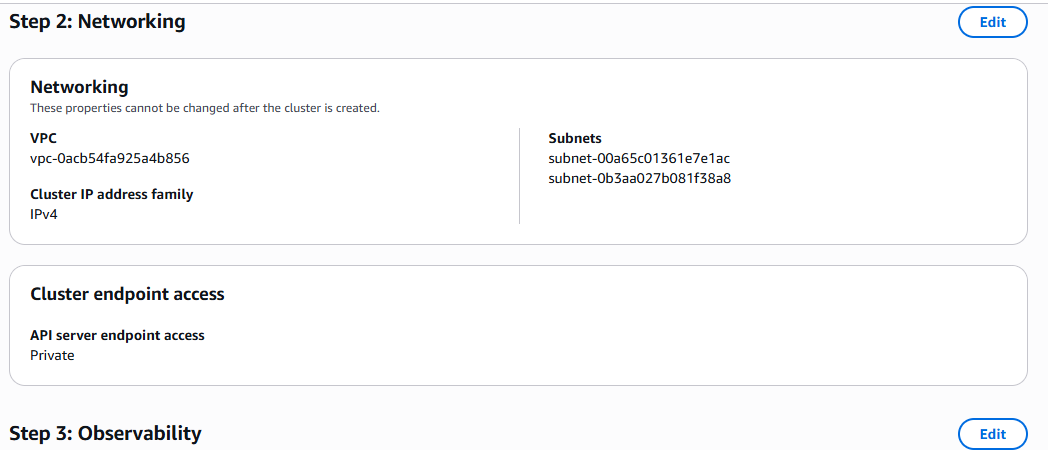
****

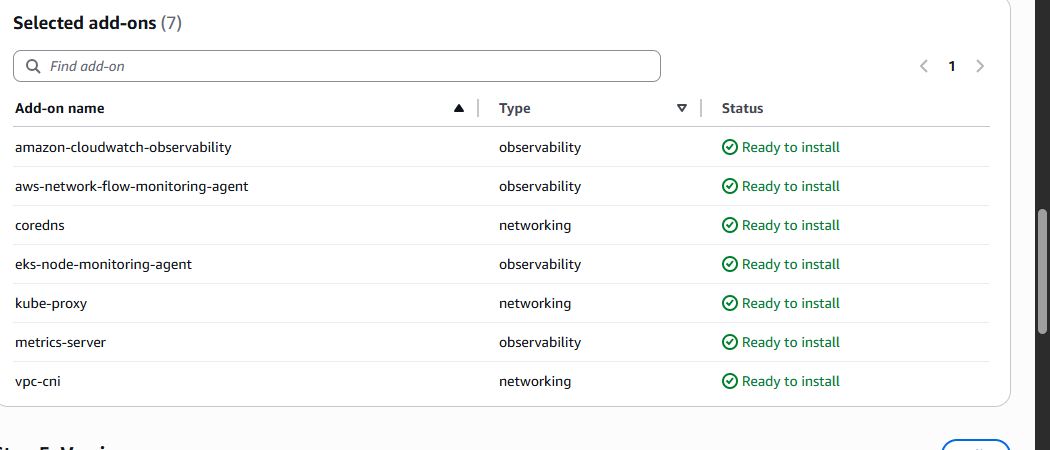
****

****

****

****

****

****

**🡺create node**

**kubectl get nodes**

**E0715 10:57:39.732400 17720 memcache.go:265] "Unhandled Error" err="couldn't get current server API group list: Get \"https://01D0372DFF883FABD08940C5028E8238.gr7.ap-south-1.eks.amazonaws.com/api?timeout=32s\": dial tcp 10.0.9.175:443: i/o timeout"**

**If getting this kind of error then allow port 443 on all node**