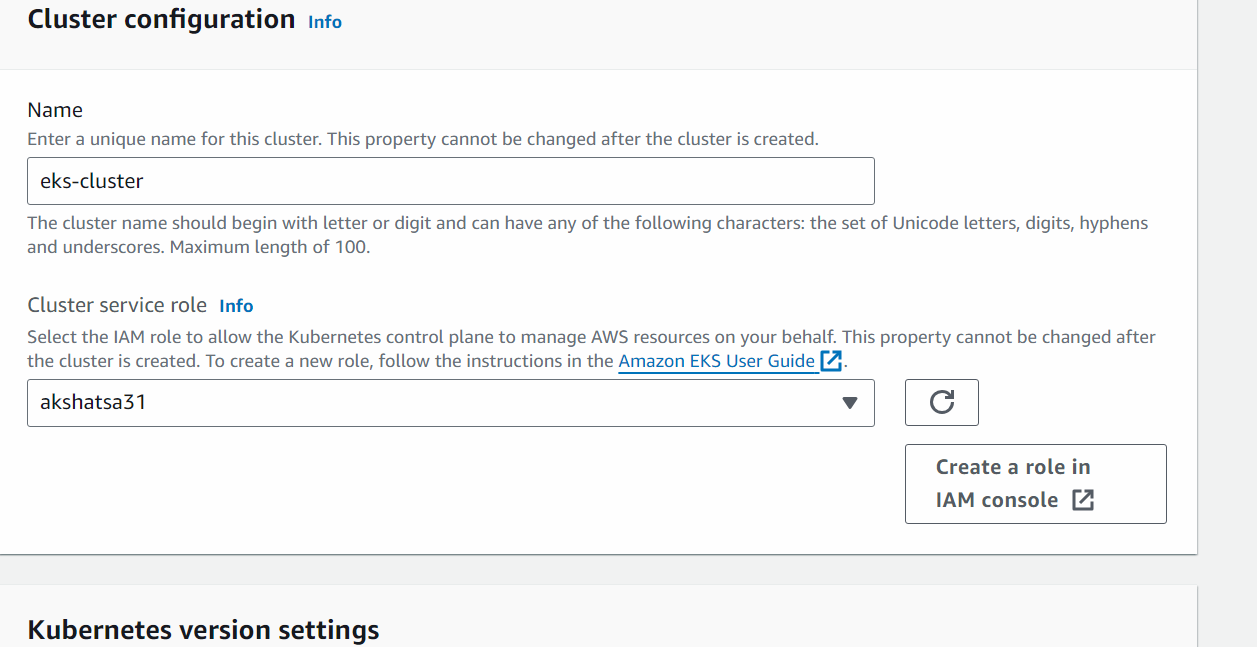
EKS

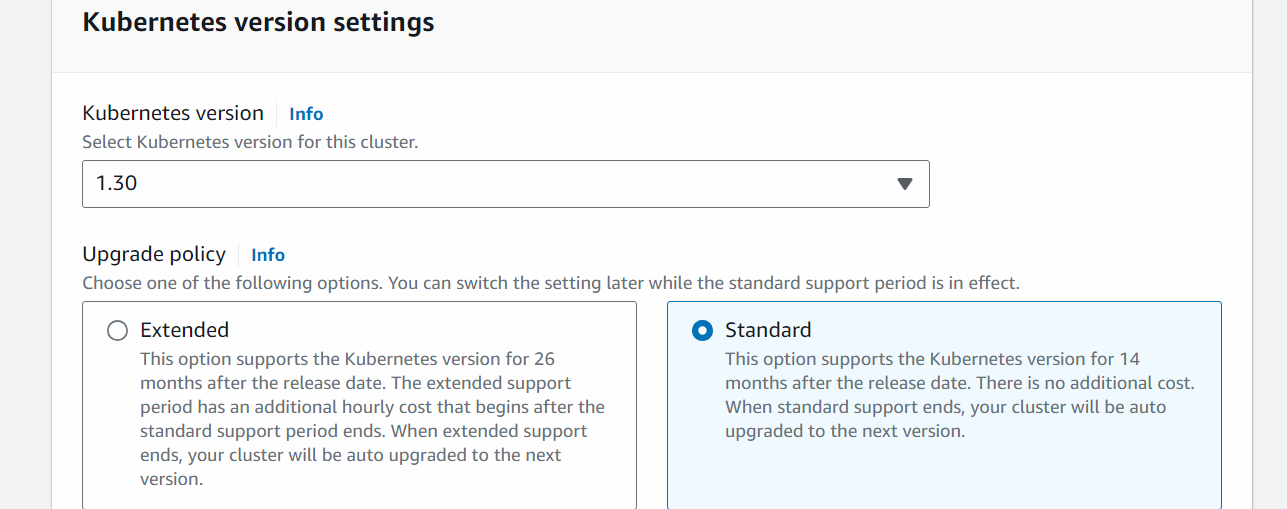
Elastic kubernetes service

Go to eks

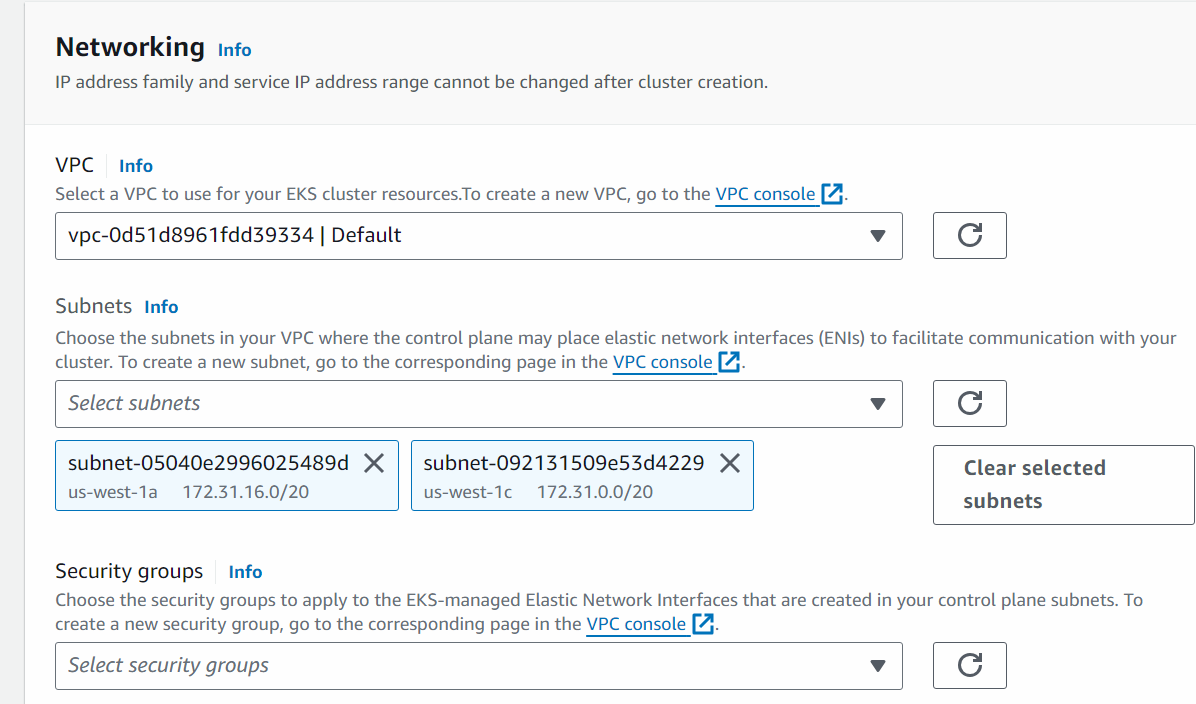
Create a cluster



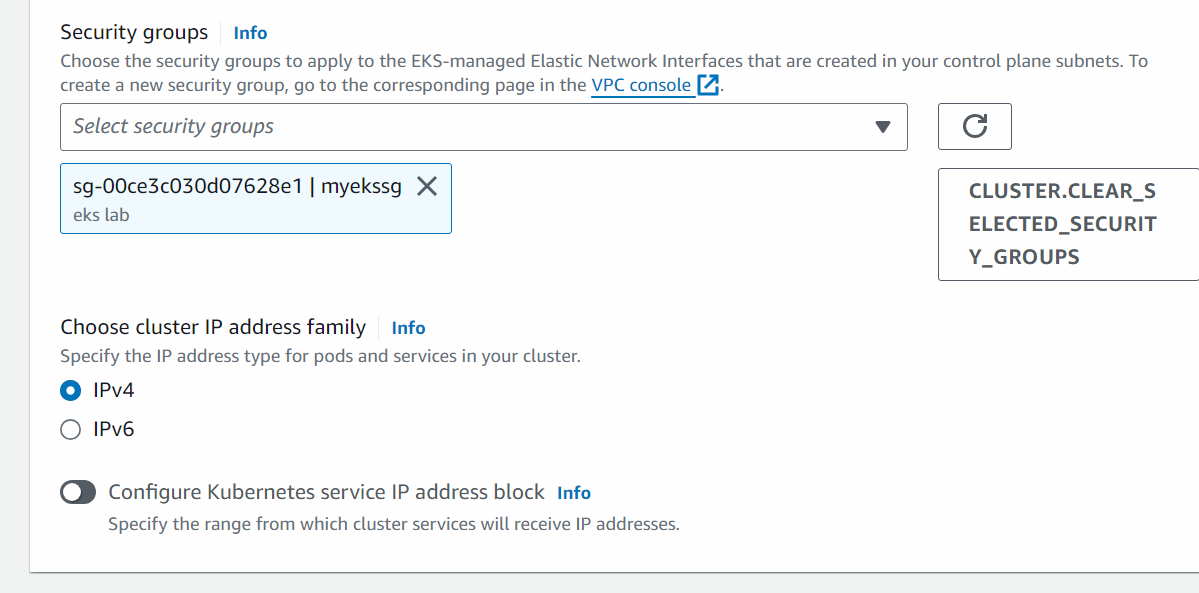
(Create a role in iam console -> next -> next)



Next



Go to ec2 -> go to security group -> create a security group will all traffic enabled



Next

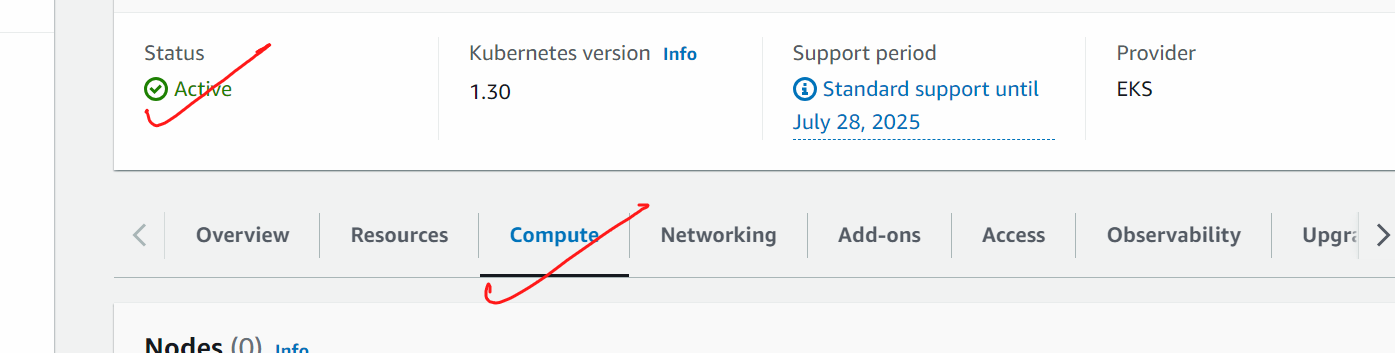
Next

Next

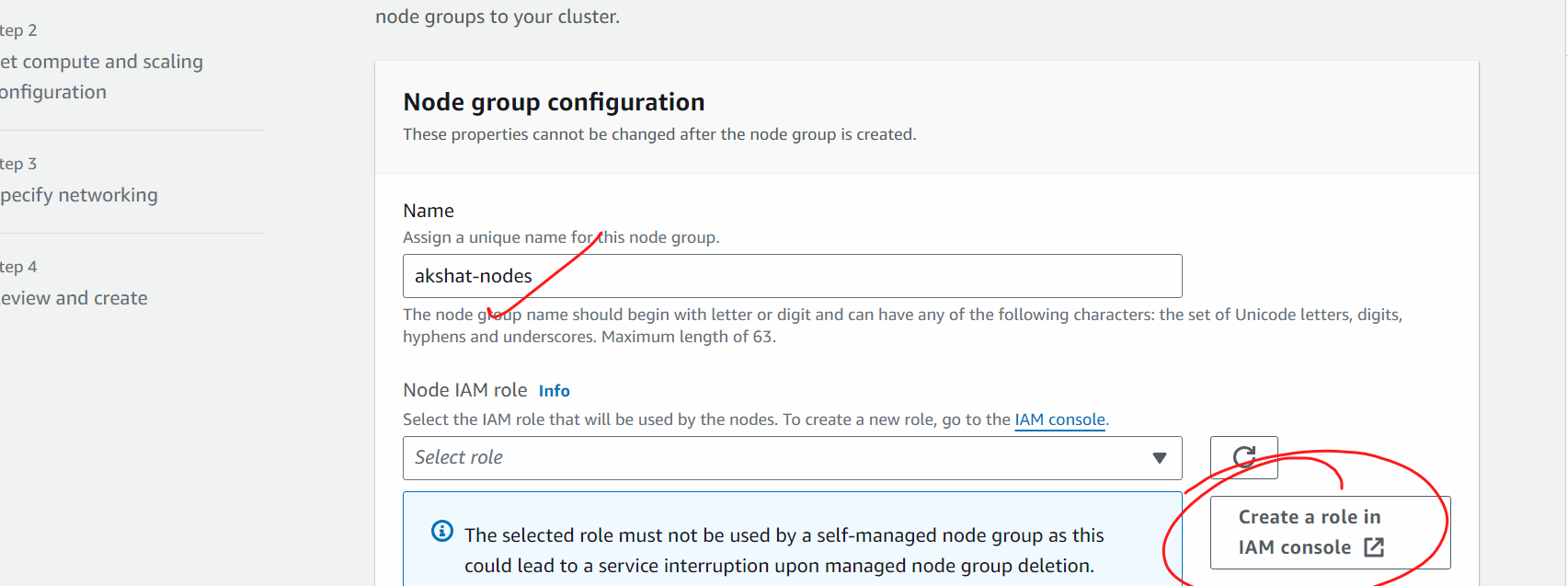
Next

Create

After the EKS cluster is created then we will create the node group

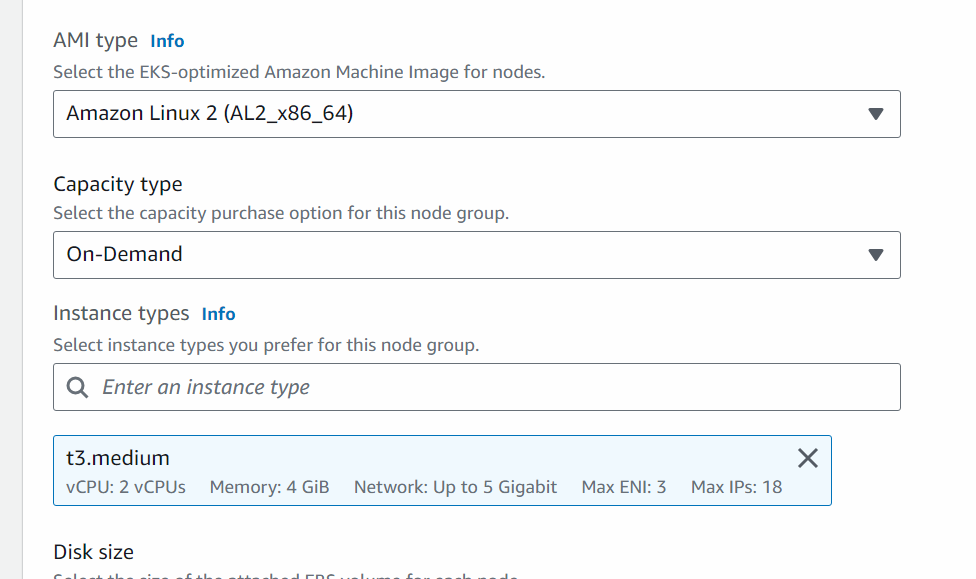


Add node group



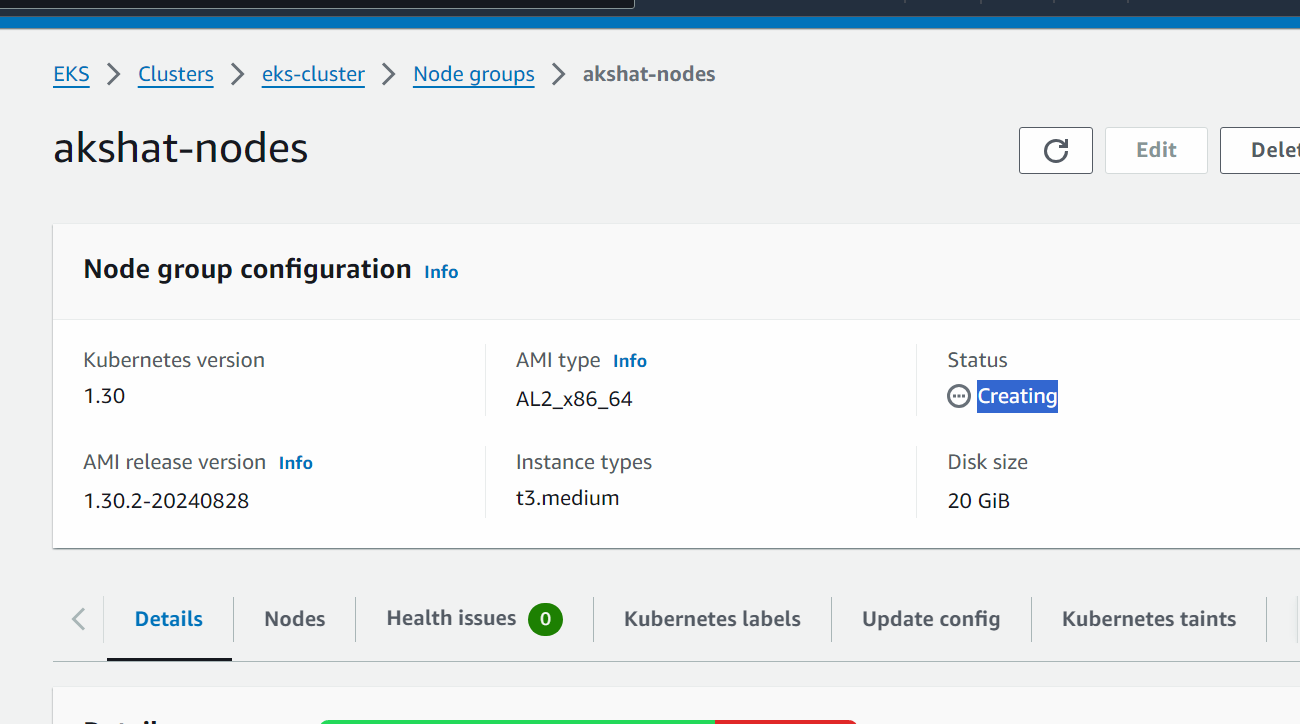
(in iam roles -< next -< next (permission will automatically set by aws) -> give some name -> create)

Next



Next

Next



After the nodes are created we will connect the cluster from the ec2 machine so that we can talk to the cluster

Go to eks cluster

Go to compute

Add node group

And create a node group

Wait till the time node group is live

My eks service is now working…we need to access the eks from ec2 machine

Launch a ubuntu ec2 machine with all traffic enabled

Connect to the ec2 machine

#sudo su

# apt update

#we will now install aws cli

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

apt install unzip -y

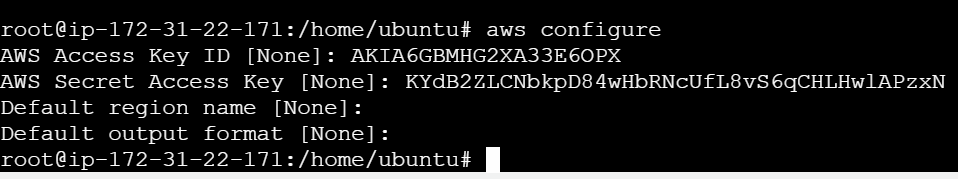
unzip awscliv2.zip

sudo ./aws/install

We will generate access key and secret access keys

In the machine ,  
  
# aws configure

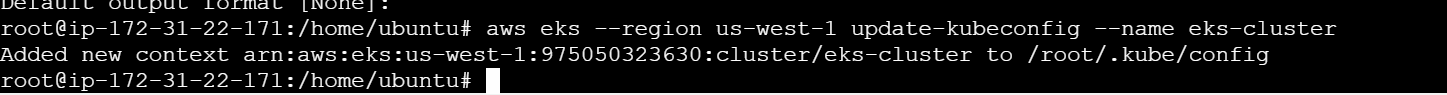
Put access key and secret access keys there



After this is done,,,,,

We will connect this machine with eks cluster

aws eks --region example\_region update-kubeconfig --name cluster\_name



sudo apt update

curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

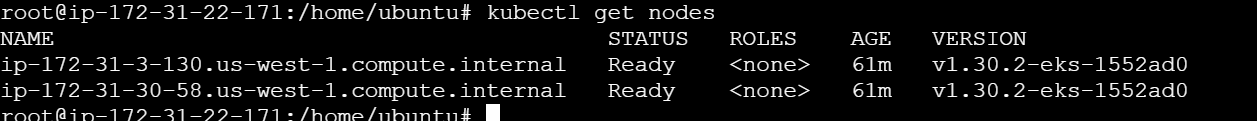
chmod +x ./kubectl

sudo mv ./kubectl /usr/local/bin/kubectl

kubectl version --client

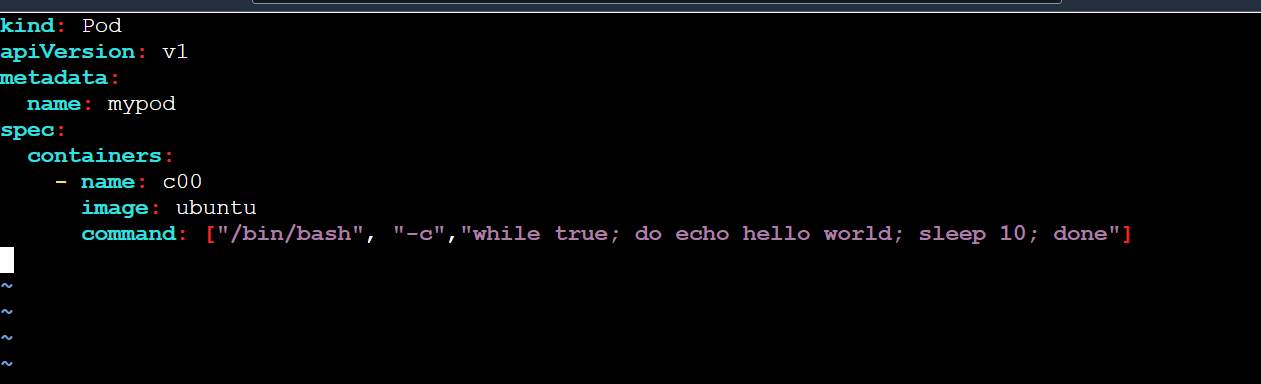
sudo curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" | sudo tar xz -C /usr/local/bin

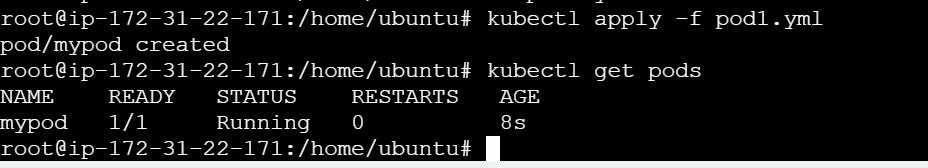
eksctl version





# vi pod1.yml





vi deploy.yml

Deployment in kubernetes is a high level resource that provides declarative updates to applications. its a type of deployment controller that manages replica set and ensures that a specified number of pod replicas are running at any given time.

Deployment controllers handle the rollout and rollback of updates to your application ,

making it a key component in k8s.

vi mydeploy.yml

kind: Deployment

apiVersion: apps/v1

metadata:

name: mydep

spec:

replicas: 2

selector:

matchLabels:

name: deployment

template:

metadata:

name: testpod

labels:

name: deployment

spec:

containers:

- name: c00

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo hello world; sleep 5; done"]