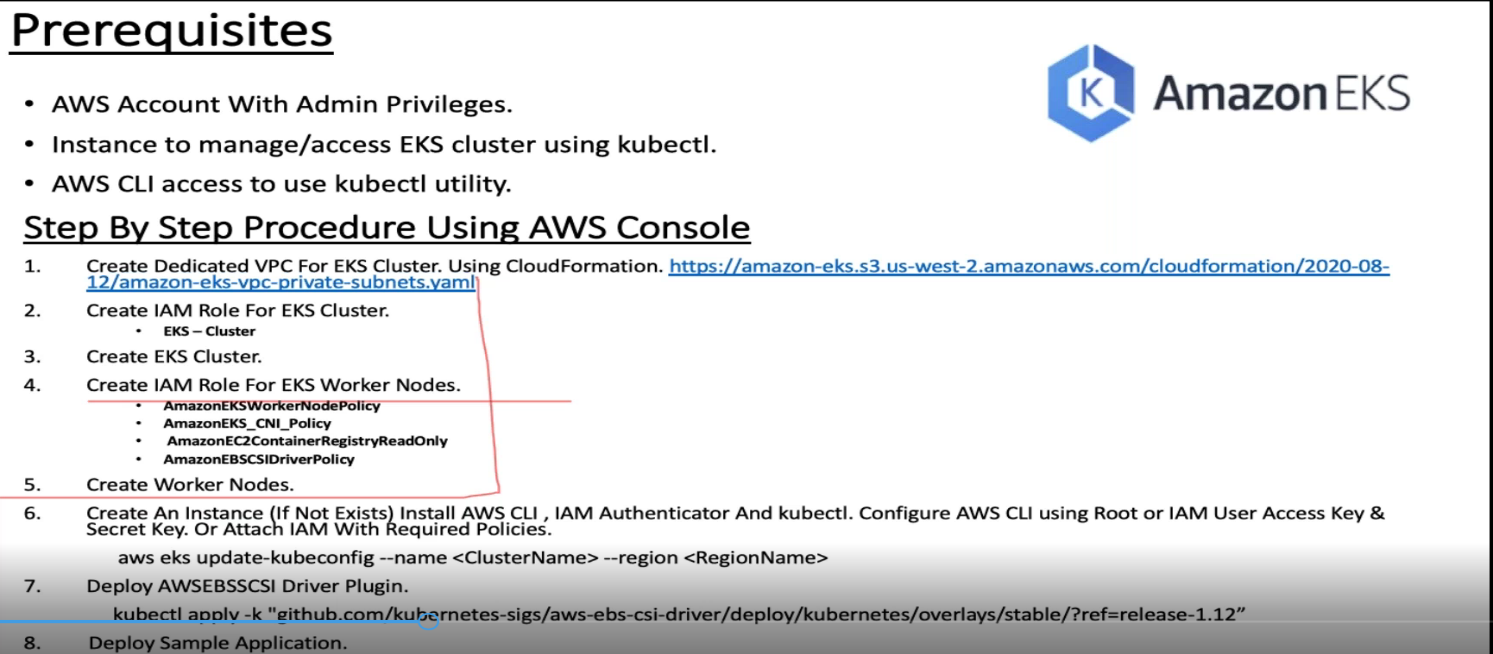
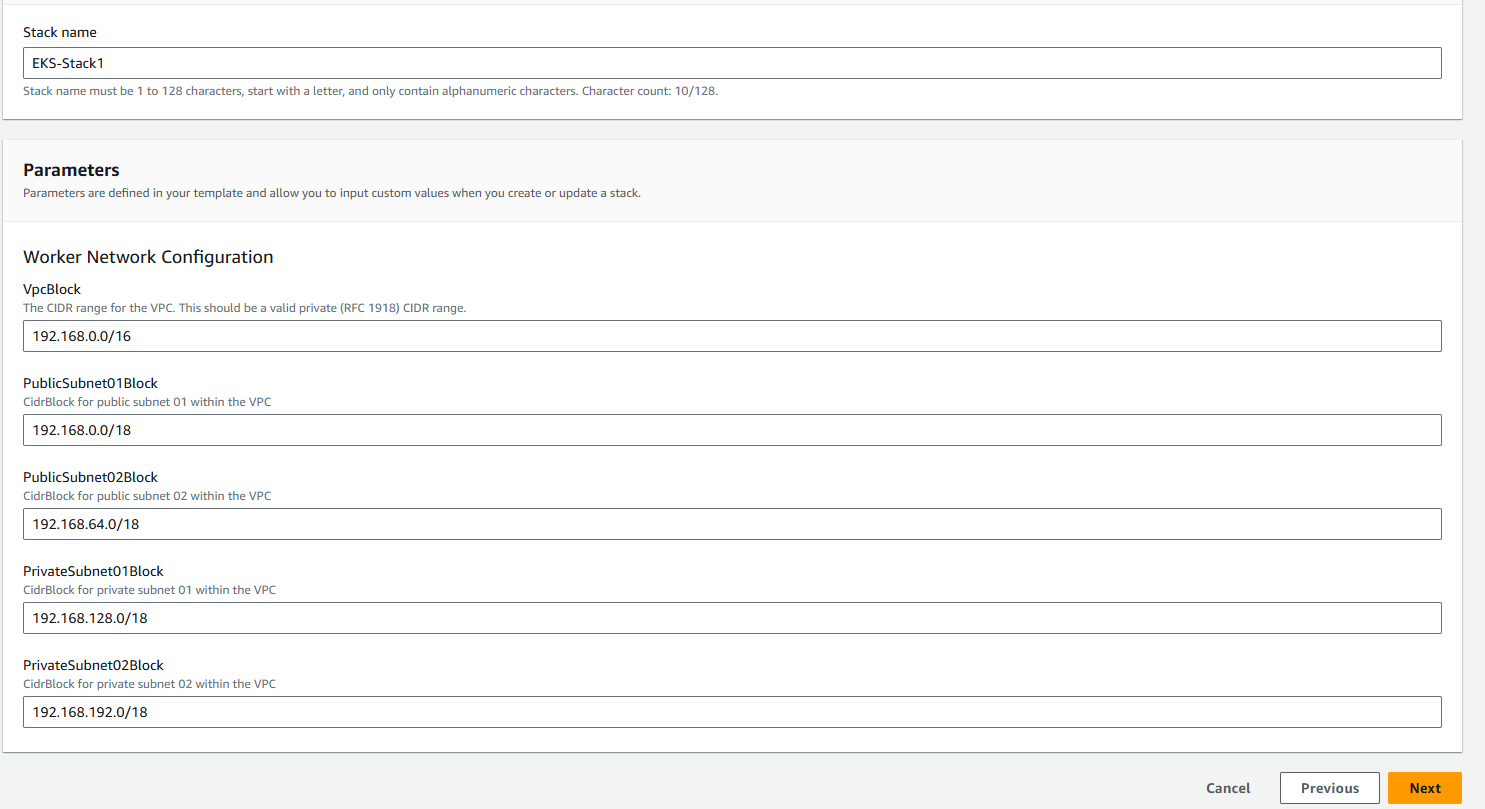
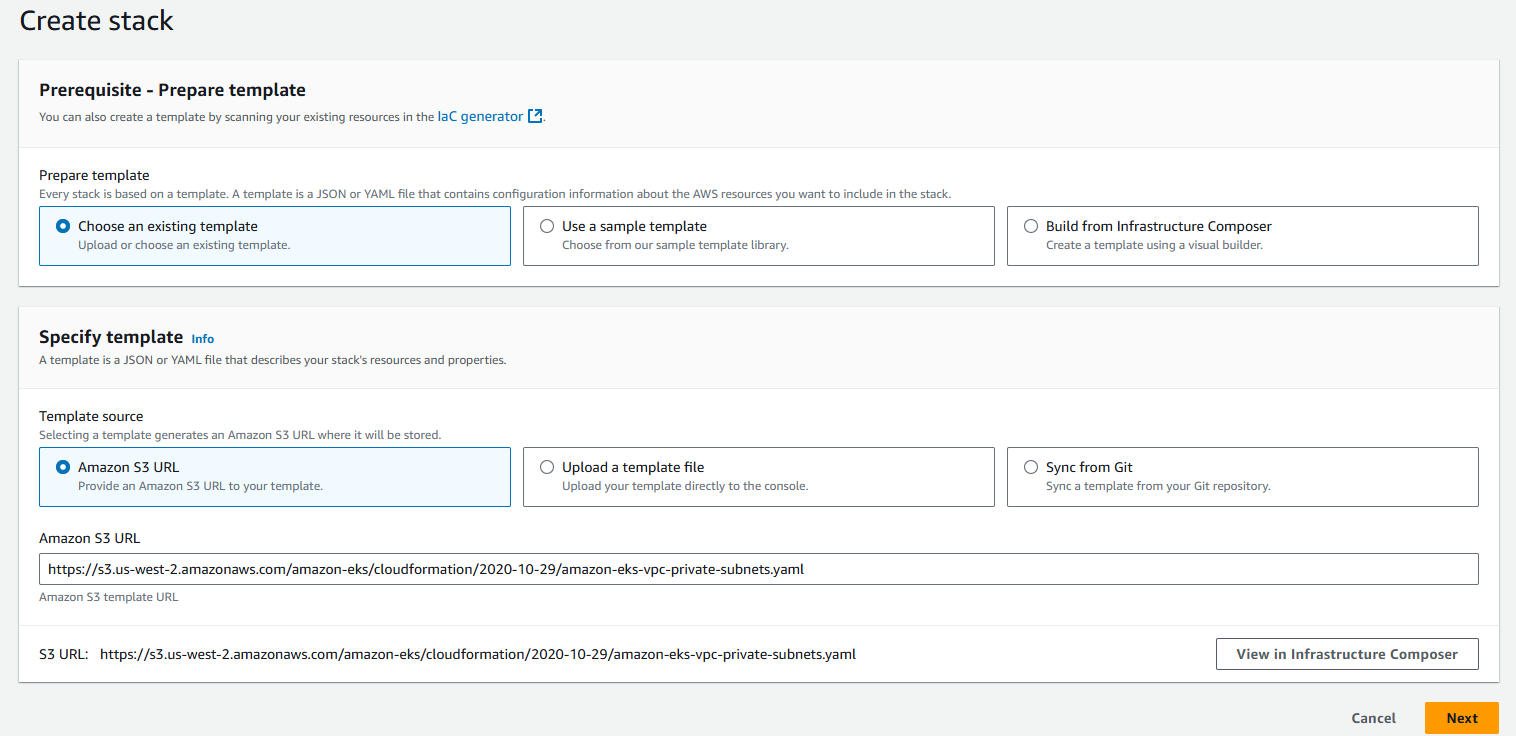
**STEPS TO CREATE EKS CLUSTER IN AWS**

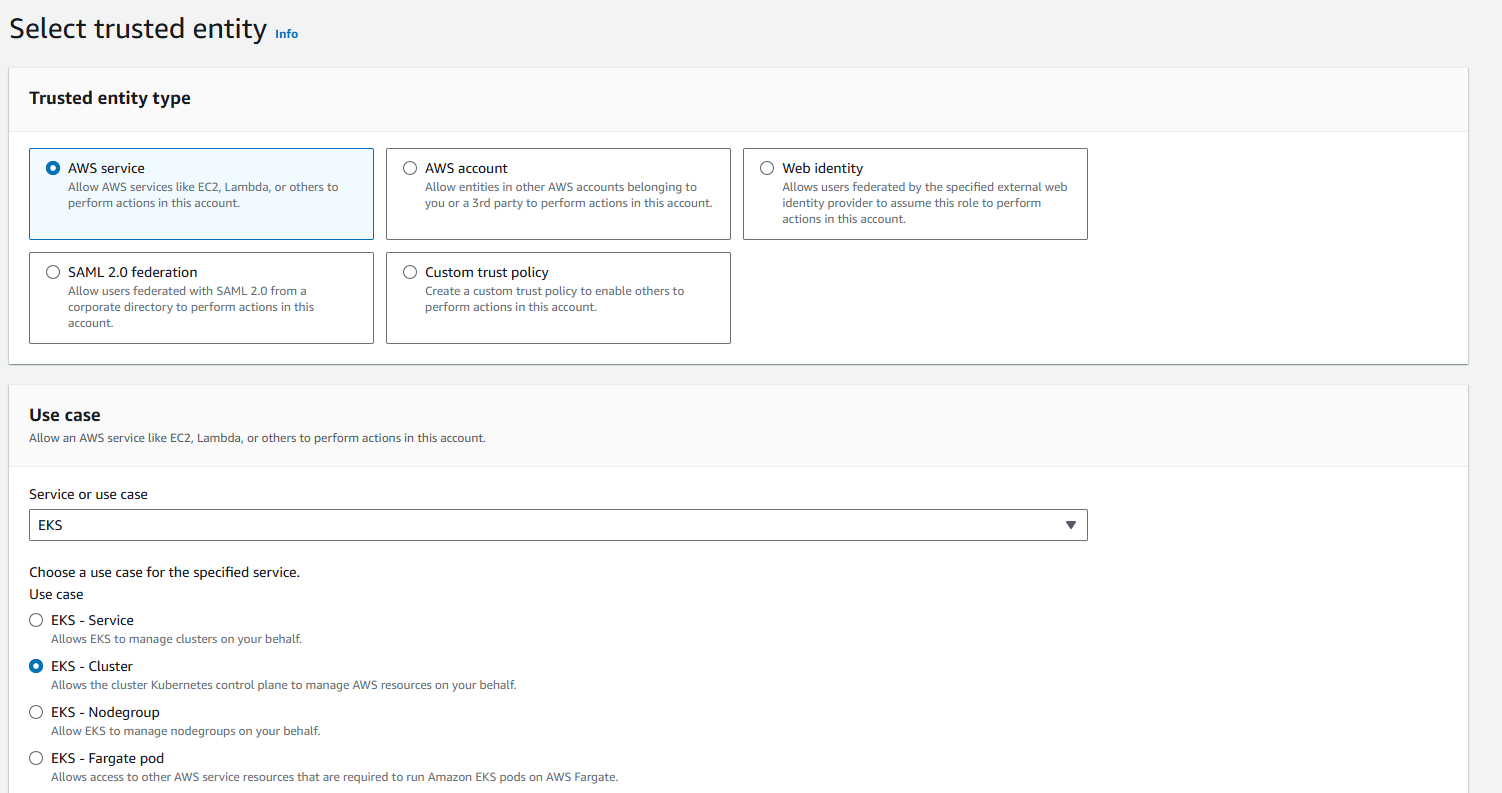


**Step 1: Create a Dedicated VPC for EKS cluster**

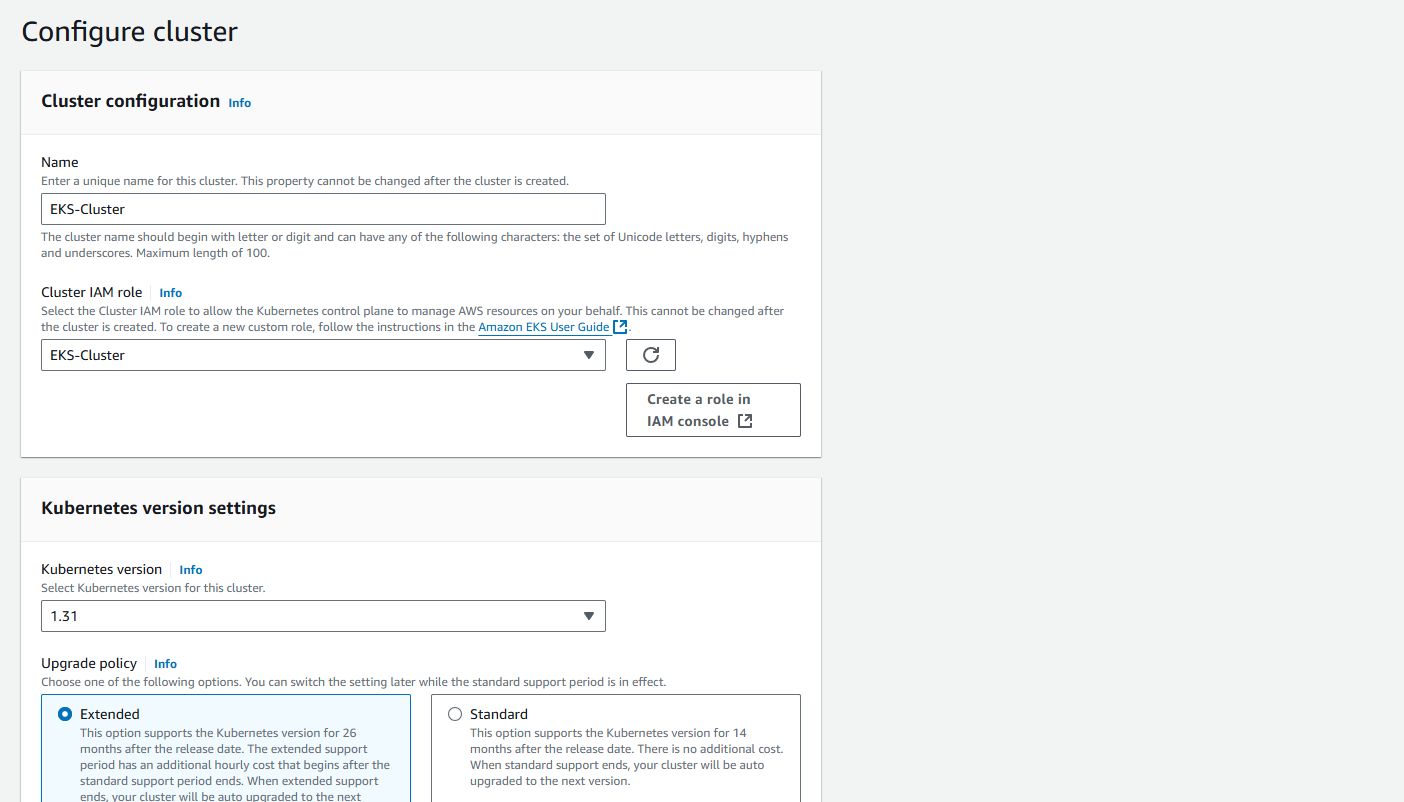
<https://s3.us-west-2.amazonaws.com/amazon-eks/cloudformation/2020-10-29/amazon-eks-vpc-private-subnets.yaml>

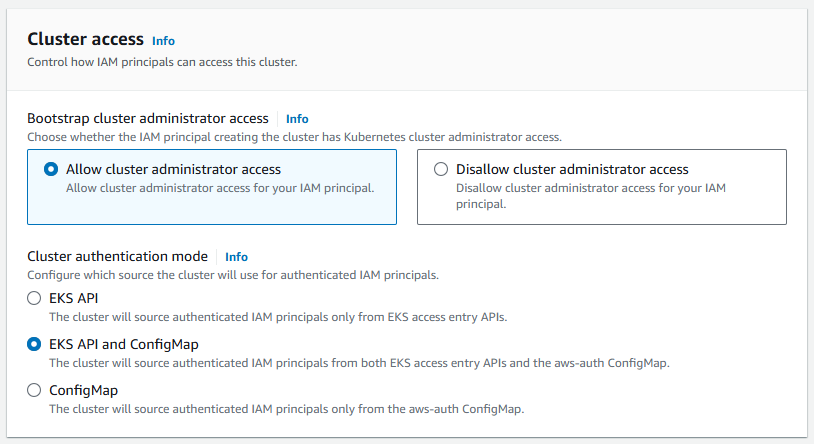
Go to Cloud formation in AWS 🡪 Create Stack 

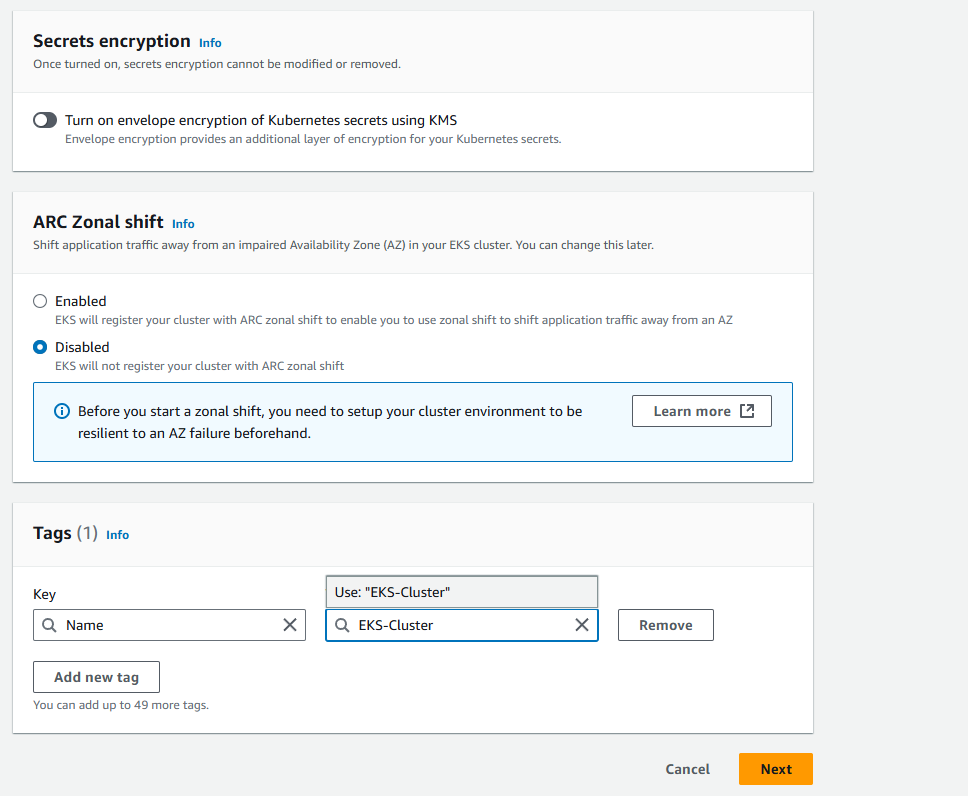
**Step 2: Create an IAM role EKS cluster.**

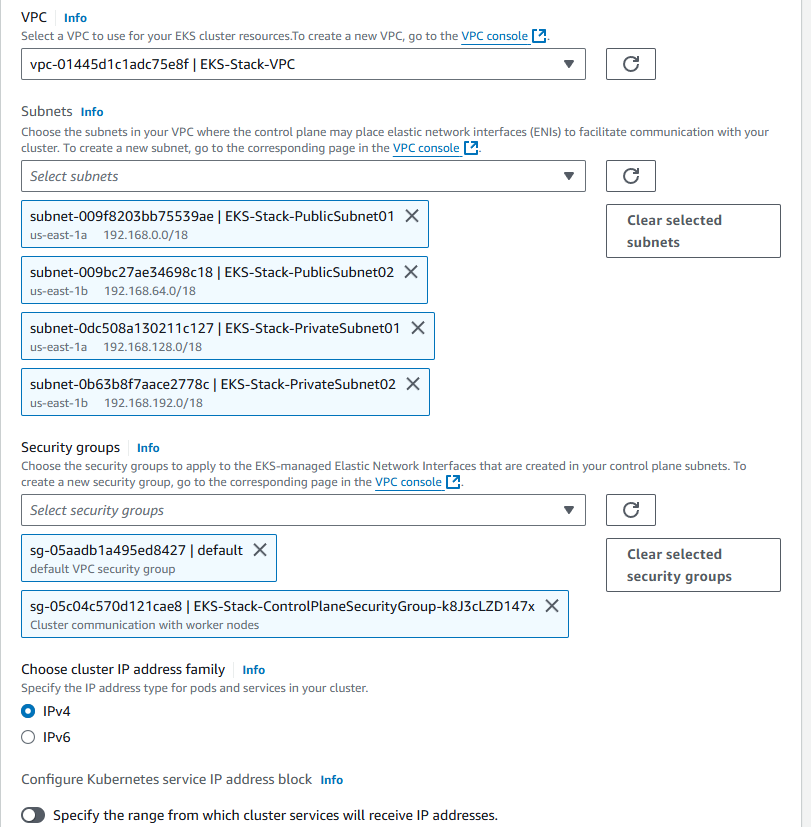


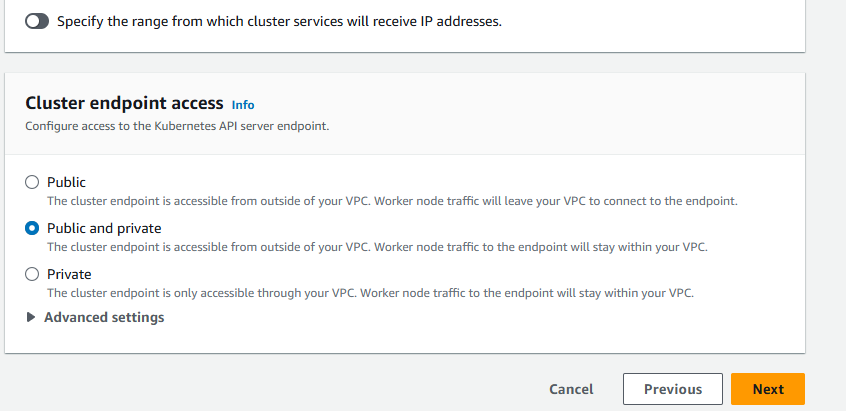
Step3: Create EKS Cluster

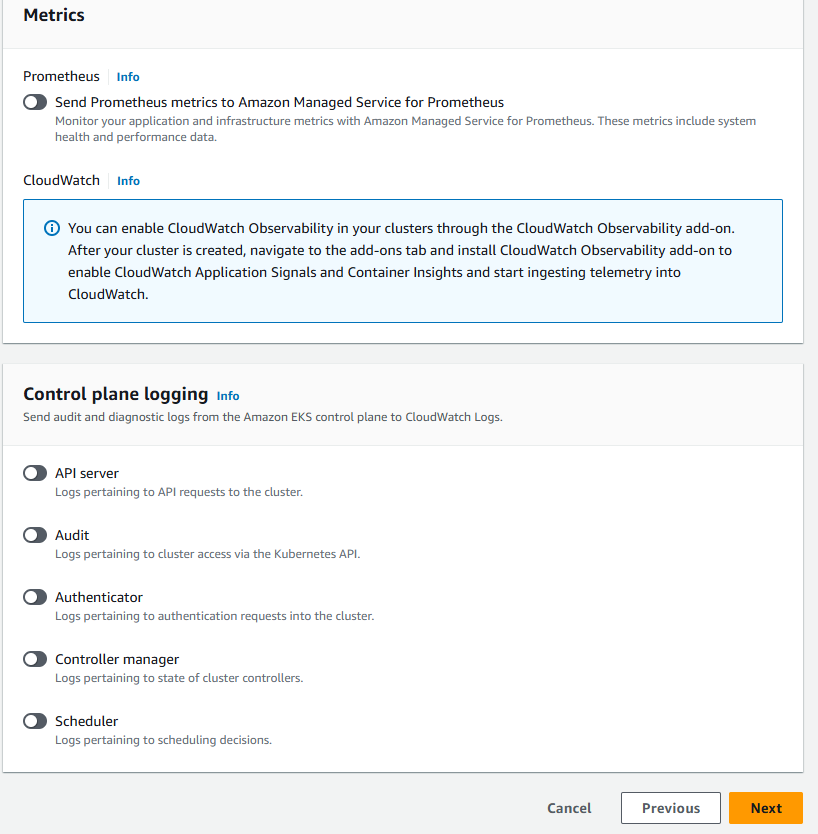


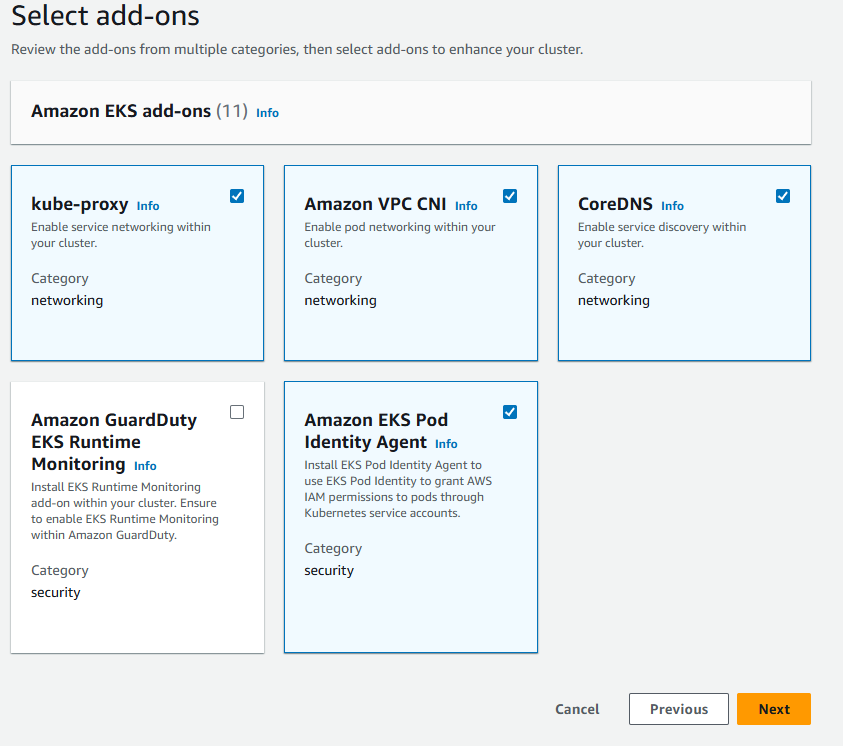


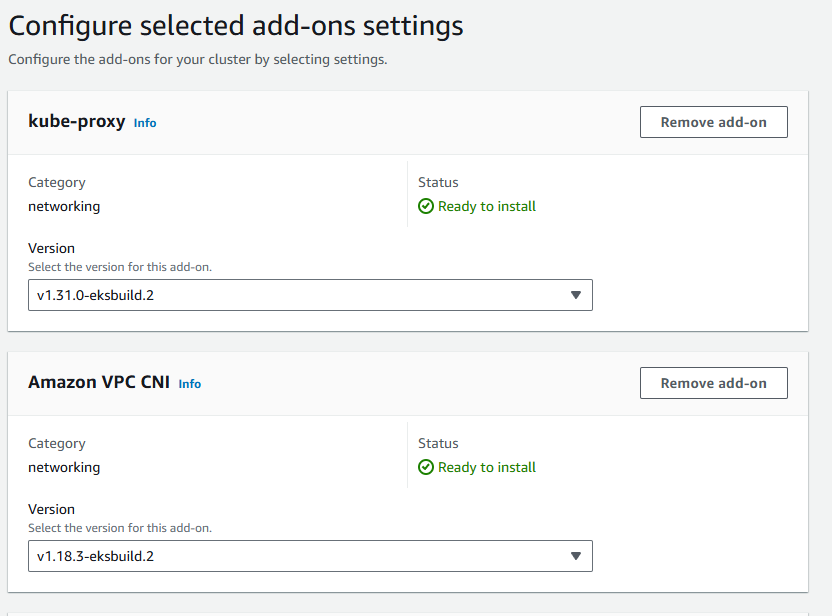


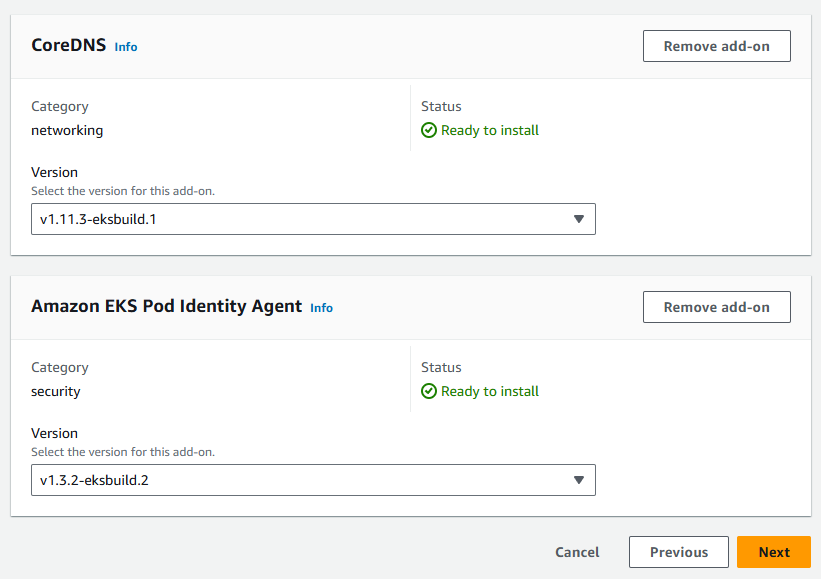






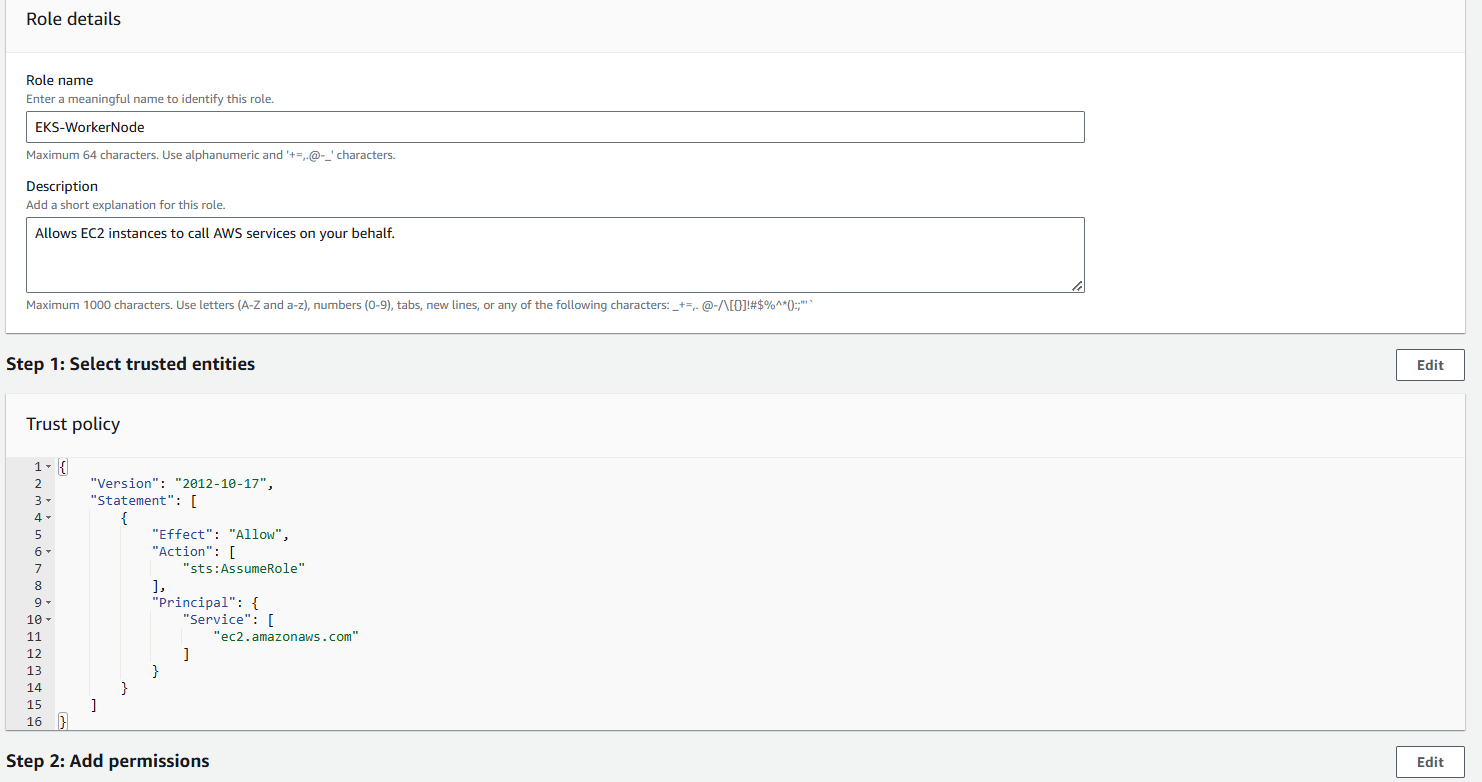


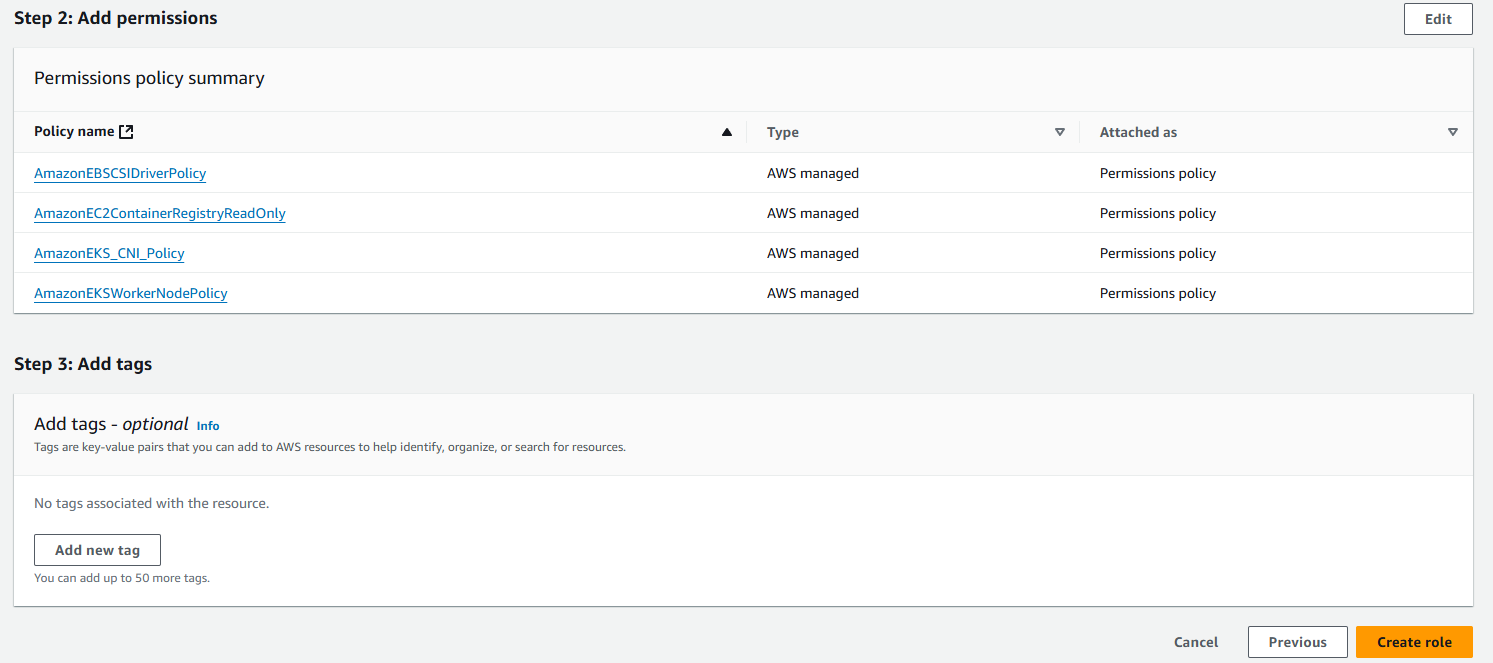




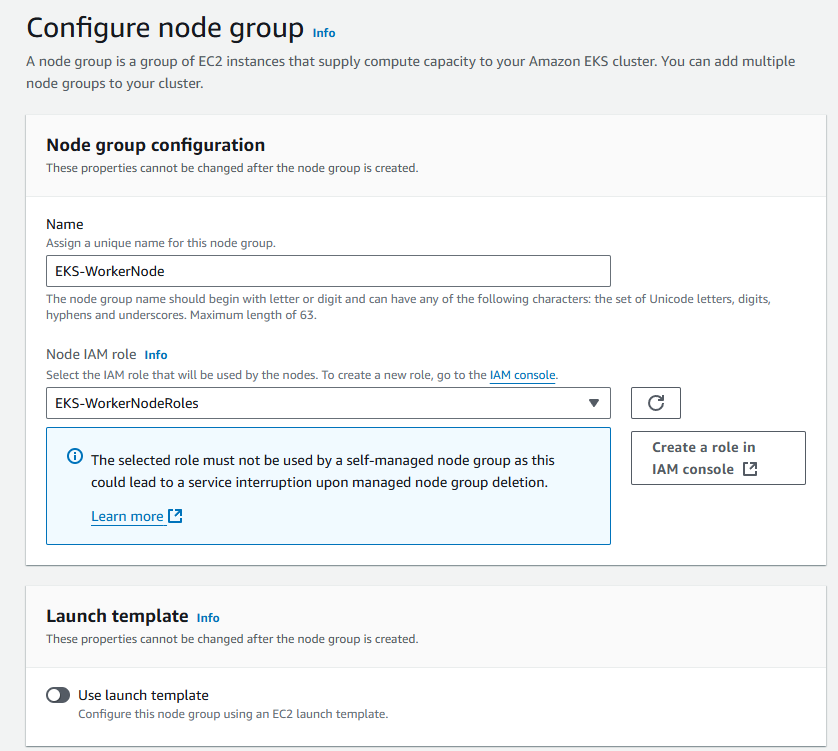
**Then Create**

**Step3: Create IAM role for Worker nodes**

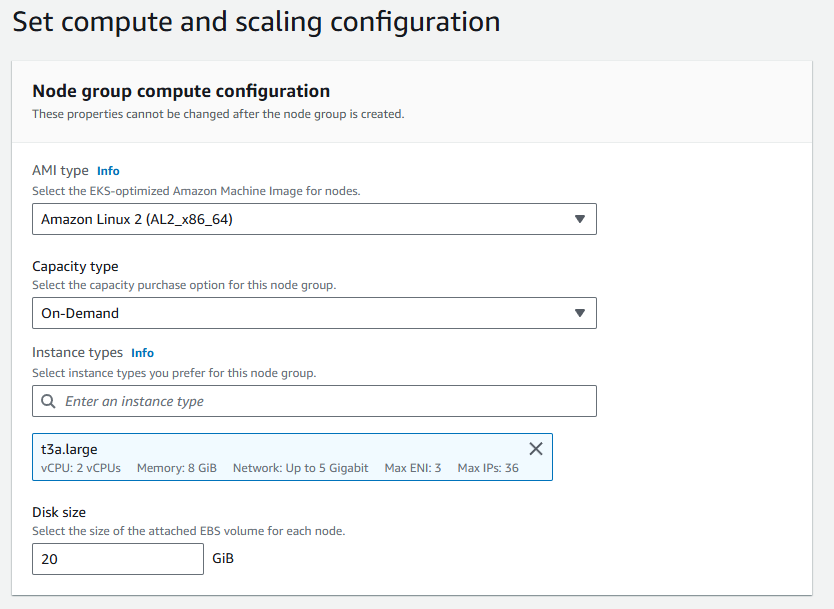
****

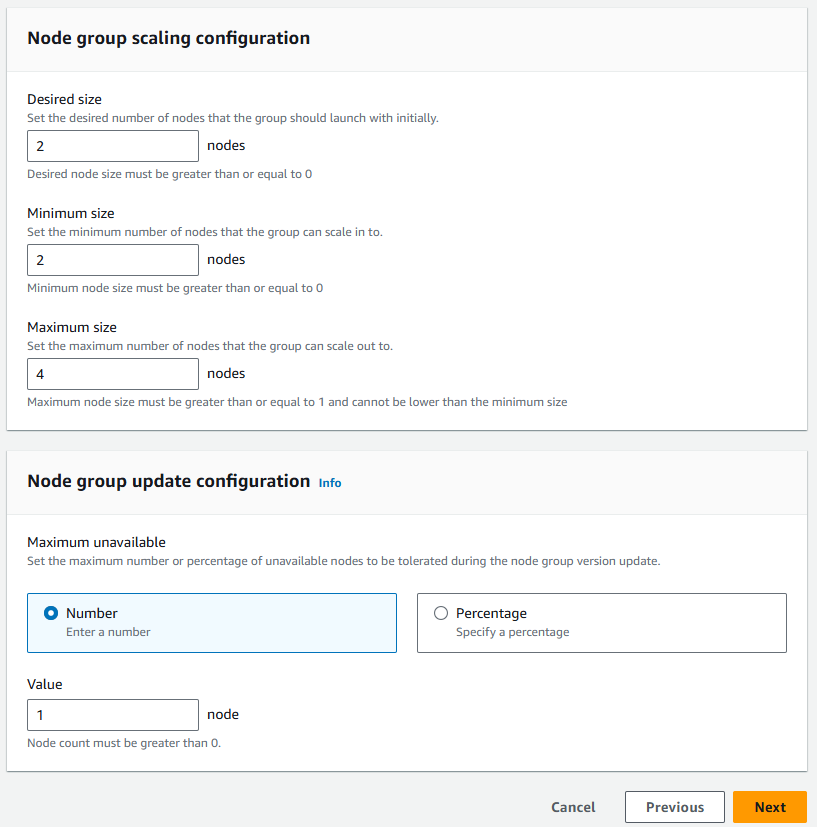
****

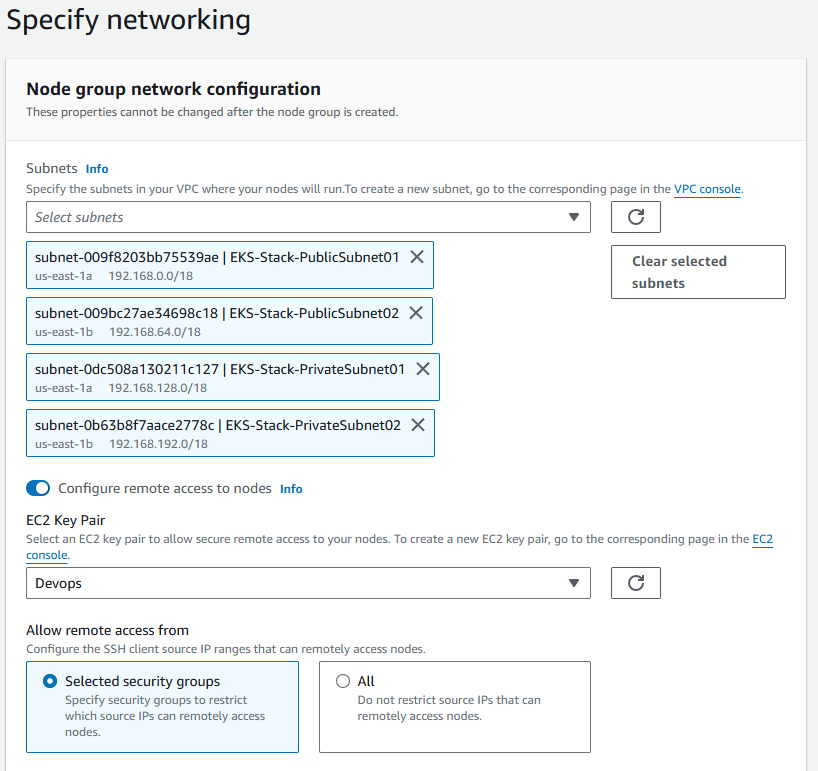
**Step4: Go to EKS cluster 🡪 go to compute 🡪 Add Node group**

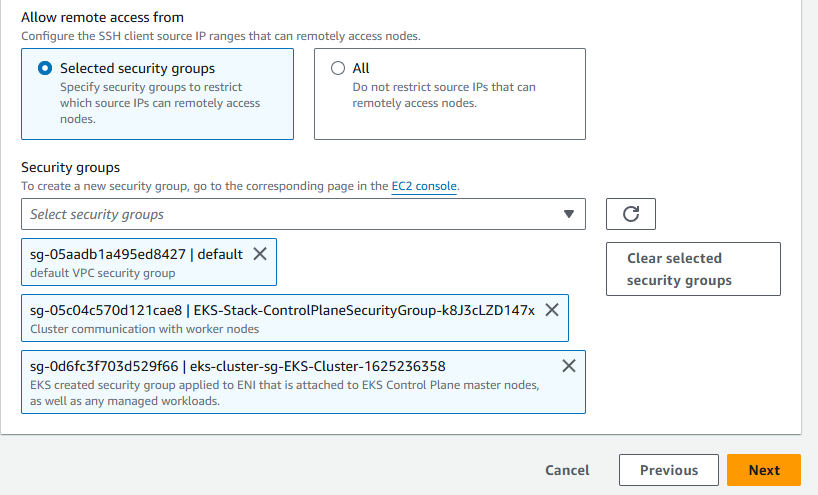
****

**Select the IAM which we create in 3rd step**

****

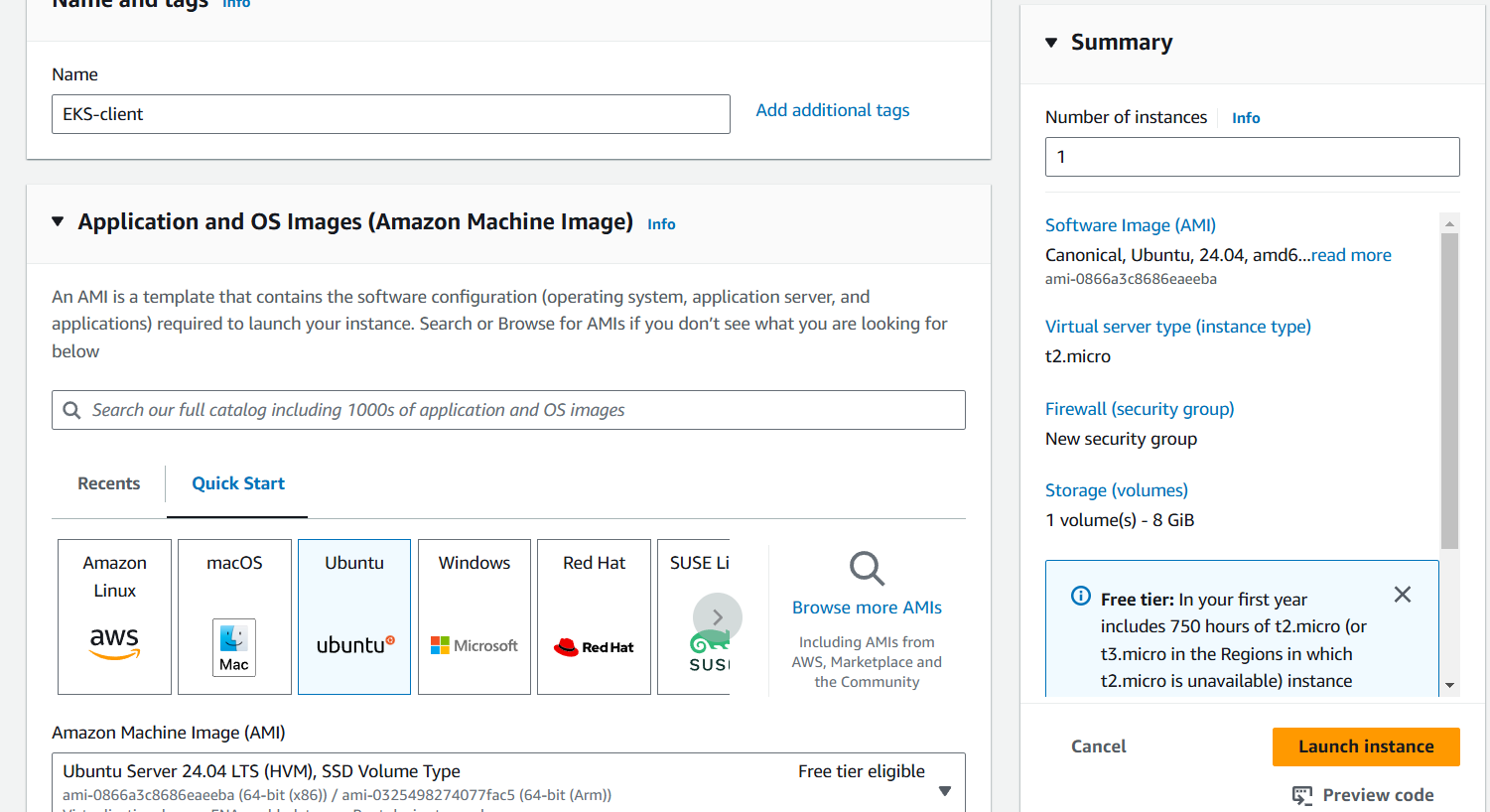
****

****

****

**Then click on Create**

**Step 6: Create an EC2 instance with t2.micro for Kubectl**

****

**Step 7: Login to instance using ssh 🡪 Paste the below commands one by one**

* curl -LO [https://dl.k8s.io/release/**$(**curl -L -s https://dl.k8s.io/release/stable.txt**)**/bin/linux/amd64/kubectl](https://dl.k8s.io/release/$(curl%20-L%20-s%20https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl)
* sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
* If you do not have root access on the target system, you can still install kubectl to the ~/.local/bin directory:

chmod +x kubectl

mkdir -p ~/.local/bin

mv ./kubectl ~/.local/bin/kubectl

* kubectl version –client

**Step 8: Install AWS CLI**

* apt install unzip
* curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"
* unzip awscliv2.zip
* sudo ./aws/install

**Step 9: aws configure**

AWS Access Key ID [None]: give access key

AWS Secret Access Key [None]: give secret key

Default region name [None]: us-east-1

Default output format [None]: table

**Step 10: aws eks update-kubeconfig --name EKS-Cluster --region us-east-1**

**Step 11: kubectl get nodes**

**Step 12: Deploy AWSEBSSCSI Driver**

**kubectl apply -k "github.com/kubernetes-sigs/aws-ebs-csi-driver/deploy/kubernetes/overlays/stable/?ref=release-1.36"**

**Cluster Auto Scalar**

**Step1: Go to IAM 🡪 Policies 🡪JSON🡪 paste the below script 🡪 then Create the policy 🡪 go to EKS Worker Node IAM role 🡪 Attach policies**

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"autoscaling:DescribeAutoScalingGroups",

"autoscaling:DescribeAutoScalingInstances",

"autoscaling:DescribeLaunchConfigurations",

"autoscaling:DescribeScalingActivities",

"ec2:DescribeImages",

"ec2:DescribeInstanceTypes",

"ec2:DescribeLaunchTemplateVersions",

"ec2:GetInstanceTypesFromInstanceRequirements",

"eks:DescribeNodegroup"

],

"Resource": ["\*"]

},

{

"Effect": "Allow",

"Action": [

"autoscaling:SetDesiredCapacity",

"autoscaling:TerminateInstanceInAutoScalingGroup"

],

"Resource": ["\*"]

}

]

}

**Step 2: curl -O** [**https://raw.githubusercontent.com/kubernetes/autoscaler/master/cluster-autoscaler/cloudprovider/aws/examples/cluster-autoscaler-autodiscover.yaml**](https://raw.githubusercontent.com/kubernetes/autoscaler/master/cluster-autoscaler/cloudprovider/aws/examples/cluster-autoscaler-autodiscover.yaml)

**Step 3: vim cluster-autoscaler-autodiscover.yaml**

edit the cluster name to our cluster name

**Step 4: kubectl apply -f cluster-autoscaler-autodiscover.yaml**