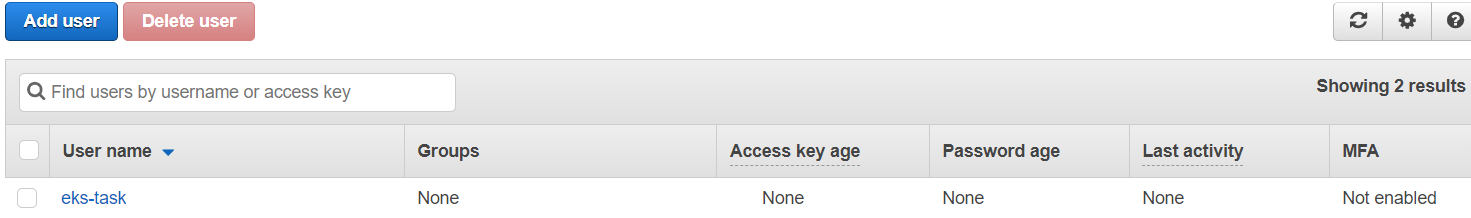
Let,s start the project

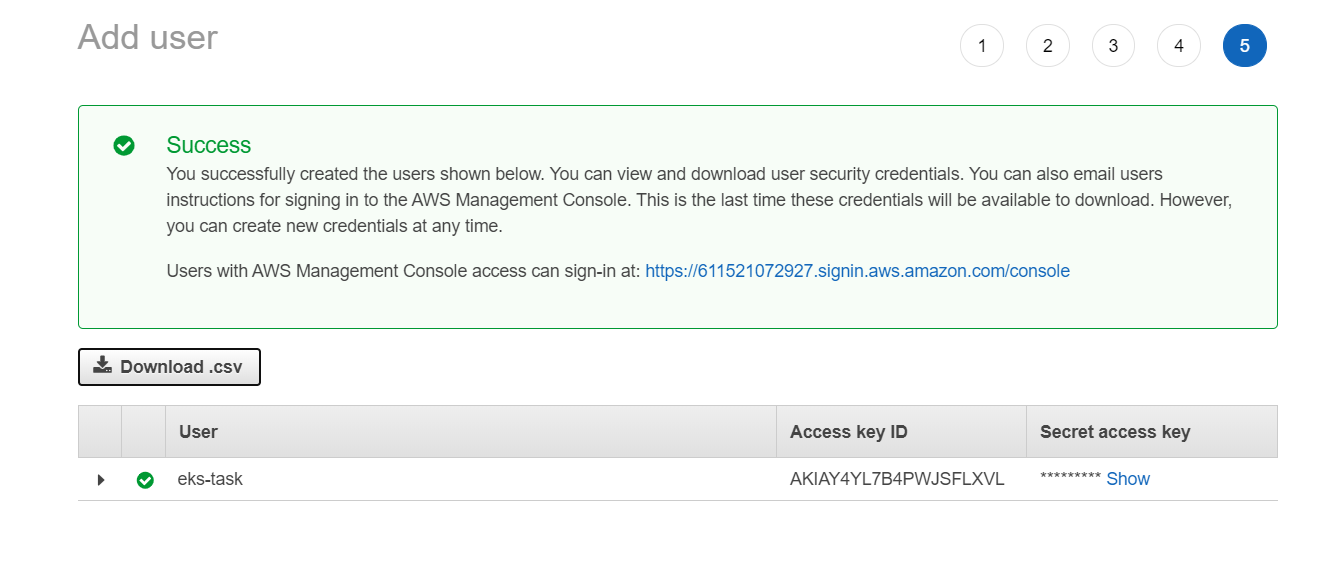
Step 1->

First I have created the user with administration access , this user have all the power same as admin . But , when multiple teams are working together then creating the user with administration access is not a good idea.



When we create the user they provide us with the .csv file that contains the Access key ID & Secret access key , this can be downloaded only once.

So don,t forget to download it !!



Login to AWS using the created user

C:\Users\LENOVO PC>aws configure

AWS Access Key ID [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DVPJ]: AKIAY4YL7B4PWJSFLXVL

AWS Secret Access Key [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*dbUB]: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Default region name [ap-south-1]:

Default output format [None]

Step 2 ->

Now let’s create the cluster

This cluster is launched in Mumbai Region , in this cluster I have created two node groups

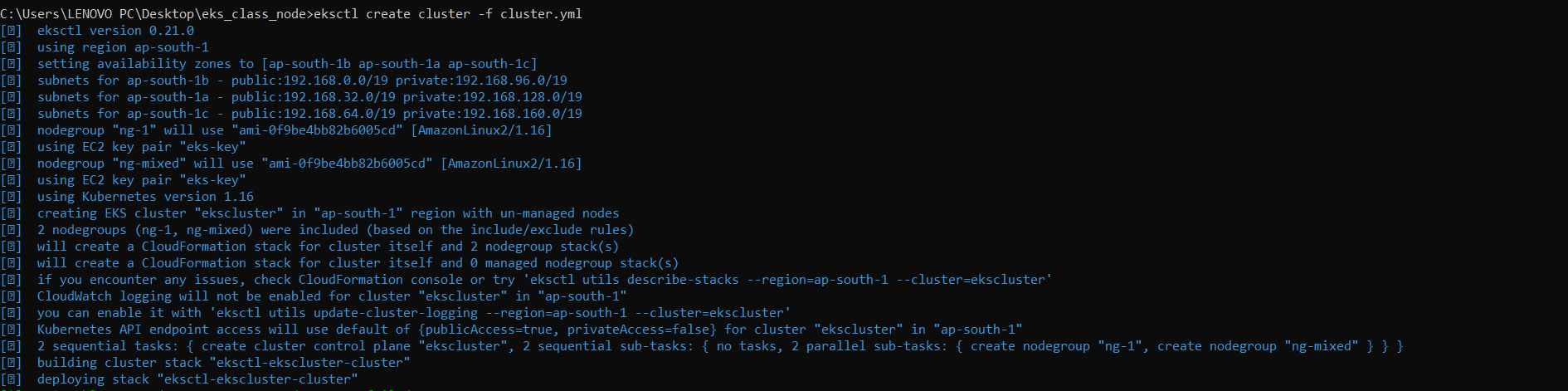
NODE GROUP 1 – This node group has desired capacity of 3 instances with t2.micro instance type

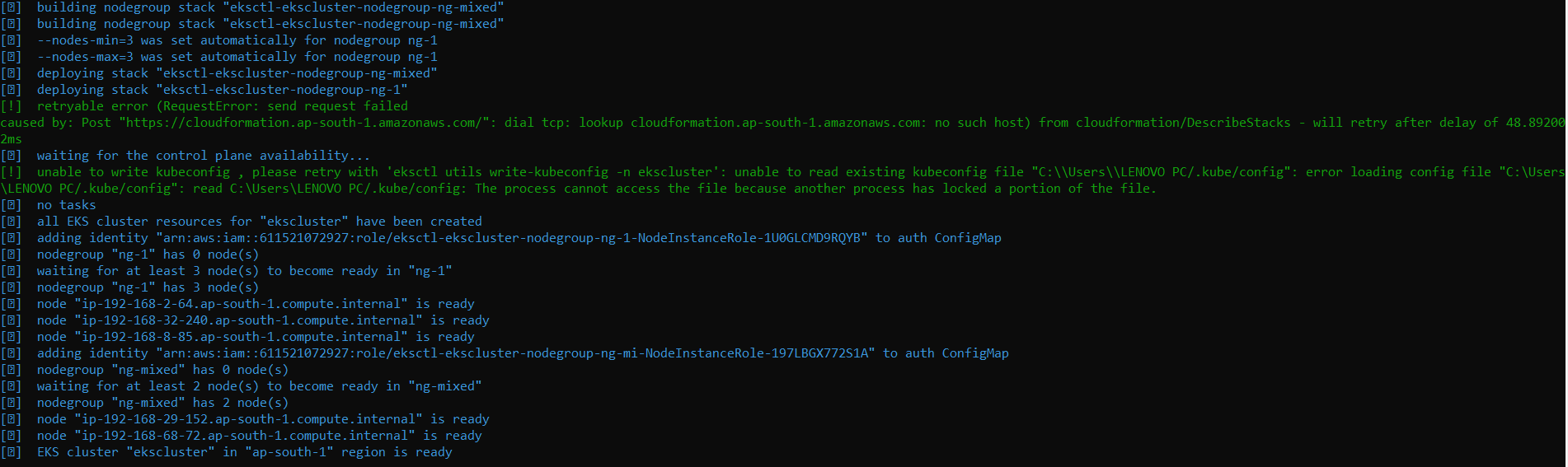
NODE GROUP MIXED – In this node I have launched the spot instances which will we created automatically when need arises , the spot instances are cheaper than Ondemand Instance

For creating the project we project we have to cluster we have to run the command:

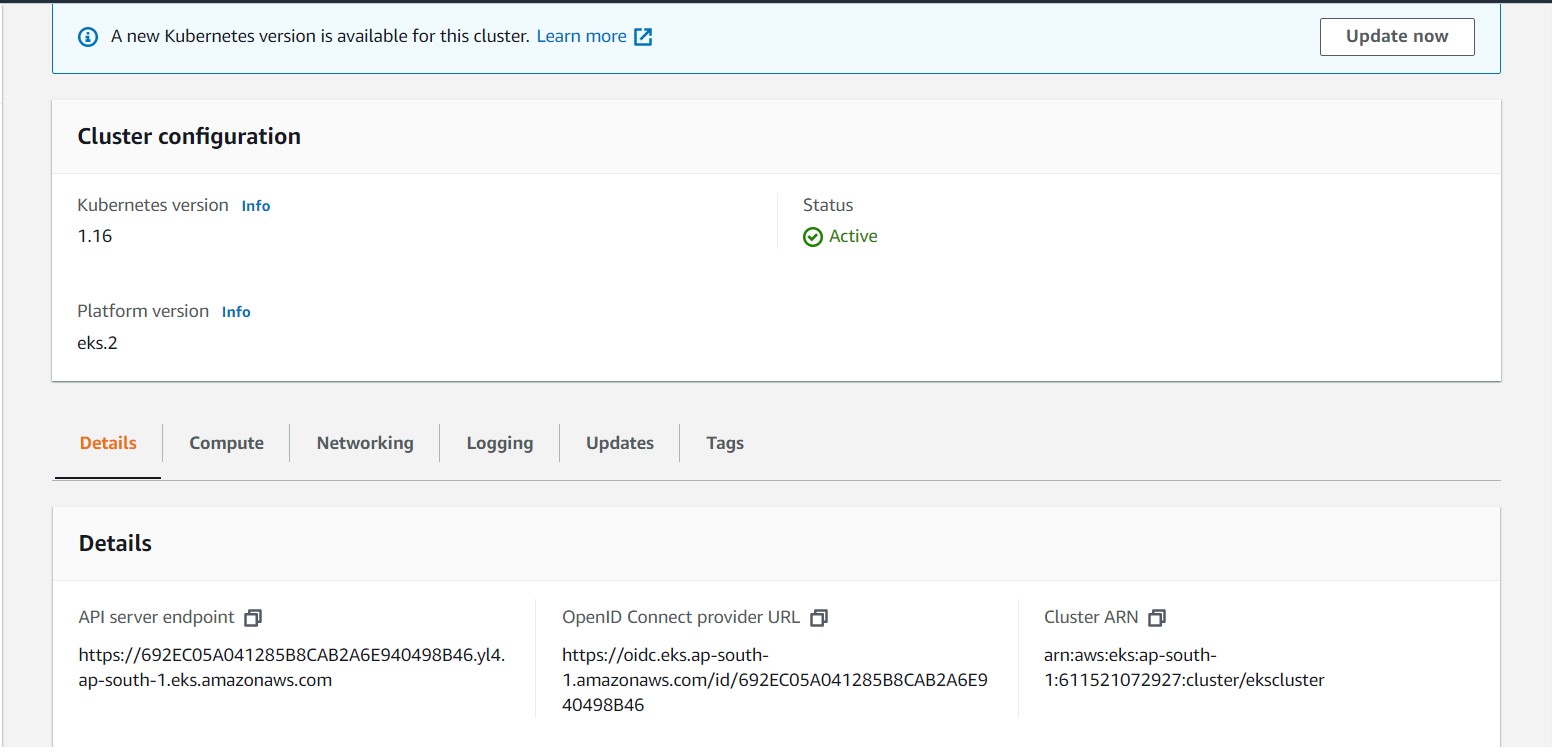
eksctl create cluster –f cluster.yml

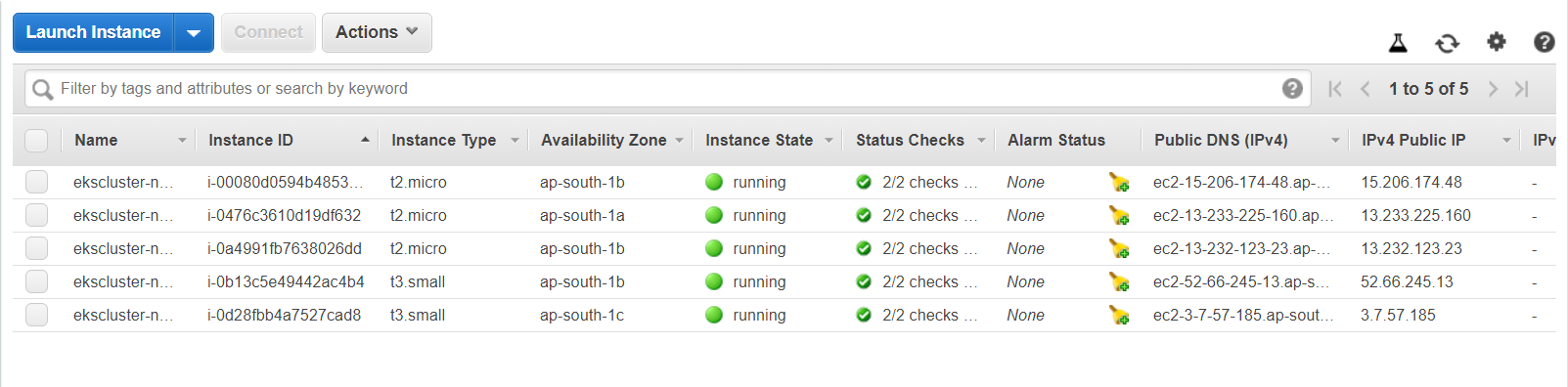
For code refer to the cluster.yml file





See here our cluster created with all the required instances ,it will take around 15-20min to create and activate our cluster





We also have to update the config file of kubernetes in .kube folder , this file contains where our Kubernetes is running

Rather than doing it manually we can do it automatically by running this command

aws eks update-kubeconfig -–name ekscluster

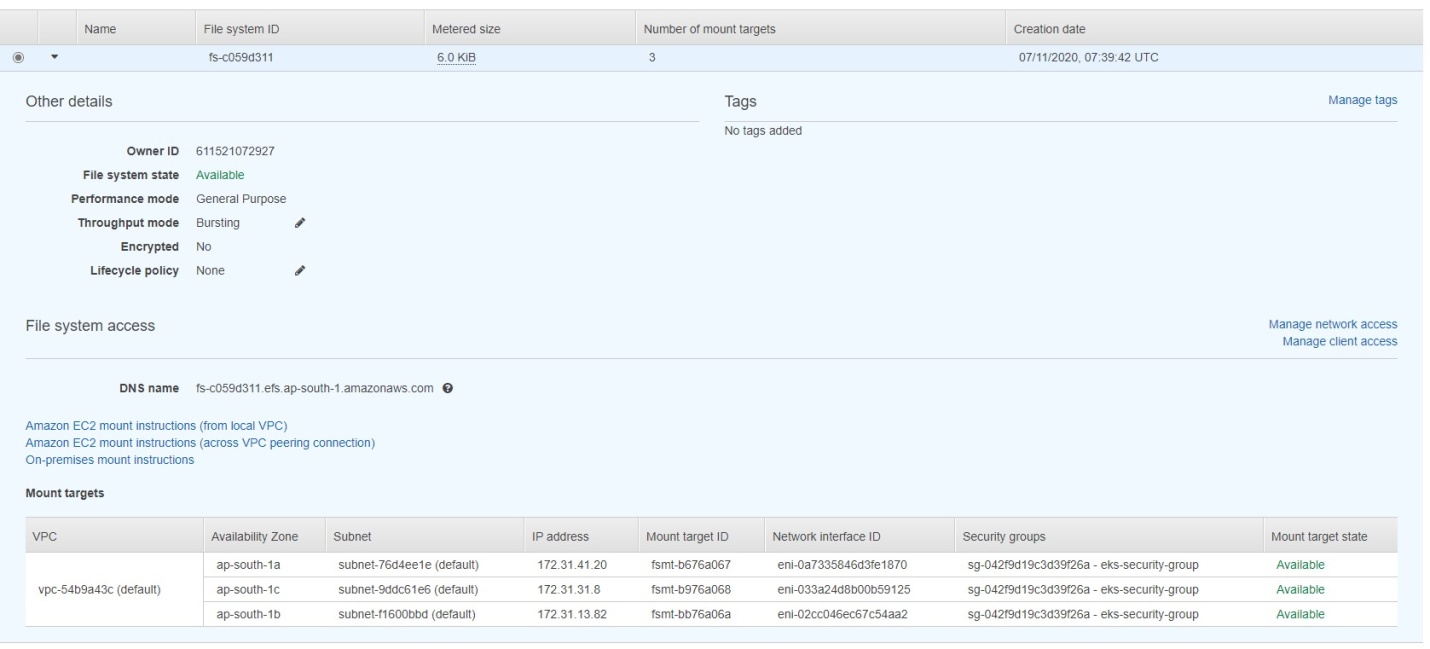


Step 3 ->

Now let’s create EFS storage so that further it can be used as PVC . We haven’t used the EBS storage due to some reasons:

1. EBS can’t be attached to multiple Instances
2. EBS is can’t be attached to instance running in other subnet

So I have created the EFS in cluster VPC , this VPC is created for our cluster and with a Security Group that has all incoming and outgoing traffic , it is also created when we created the cluster



Step 4->

Now here we create EFS provisioner , this will allows us to mount the EFS storage as PersistentVolumes(PV).I have also created a separate namespace as eks-task to perform this task

For creating the namespace and EFS provisioner run the following command:

kubectl create ns eks-task

kubectl create –f create-efs-provisioner.yaml –n eks-task

For code refer to the create-efs-provisioner.yaml file

NOTE: By default the instance doesn’t support EFS , so go inside the instance and run the yum install amazon-efs-utils , it will download the required softwares

Step 5->

In this step we modifiy some permission using the ROLE BASED ACCESS CONTROL by few lines of code.

For creating the ROLE run the command:

kubectl create –f create-rbac.yaml -n eks-task

For code refer to the create-rbac.yaml file

Step 6->

Here we create the storage class that will create pvc and provide pv dynamically.I have also created two pvc each of 10GiB. These pvc will be attached to mysql and ghost

For creating the storage and pvc run the following command:

kubectl create –f create-storage.yaml –n eks-task

For code refer to the create-storage.yaml file

Step 7->

Since now we will be launching our MYSQL and Ghost ,so for their login we require username & password but specifying the actual will make our login unsafe ,so for this I have created a secret.yml file that contains login and password in encoded form . For encoding I have used base64 encoder

For running the secret.yml run the command:

kubectl create –f mysecret.yml –n eks-task

For code refer to the mysecret.yml file

Step 8->

Now we will launch our MYSQL Database that will connect to our Ghost architecture. For this I have created a deploy-mysql.yaml file

For launching the MYSQL run the following command:

Kubectl create –f deploy-mysql.yaml –n eks-task

For code refer to the deploy-mysql.yaml file

Step 9->

Now at last I have launched the Ghost architecture , this will connect to our MYSQL database. For this I have created a deploy-ghost.yaml file

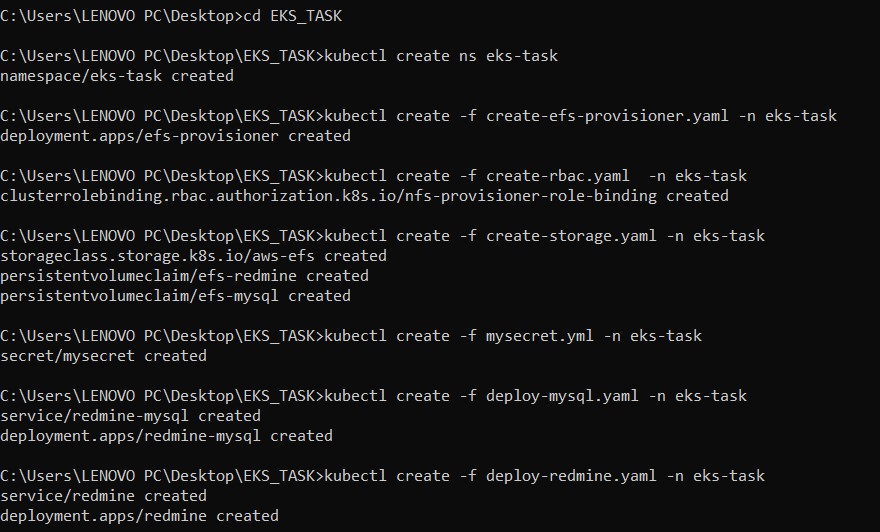
For launching the ghost architecture run the following command:

Kubectl create –f deploy-ghost.yaml –n eks-task

For code refer to the deploy-ghost.yaml file

NOTE: Launch the MYSQL before launching the Ghost architecture , otherwise the architecture would fail

Screenshot for all the above commands :



Thank you!!! ;-)