**Managed K8s on the Cloud**

Deploy a Containerized nginx:alpine webserver with 5 Replicas on a 3-node GKE Cluster. Give it the label app=nginx-alpine. Make sure all five replicas are scheduled on the same node. List the running pods output with host node information in a file named hostnode.txt. Once done, update the container image to nginx:latest.

**Process:**

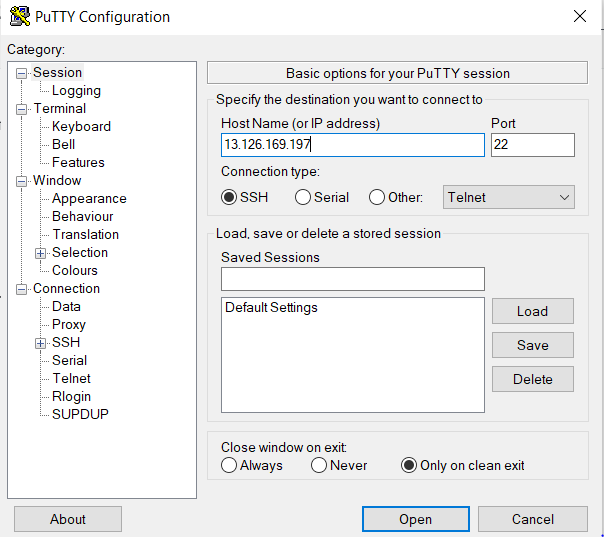
**Set Up AWS Resources**

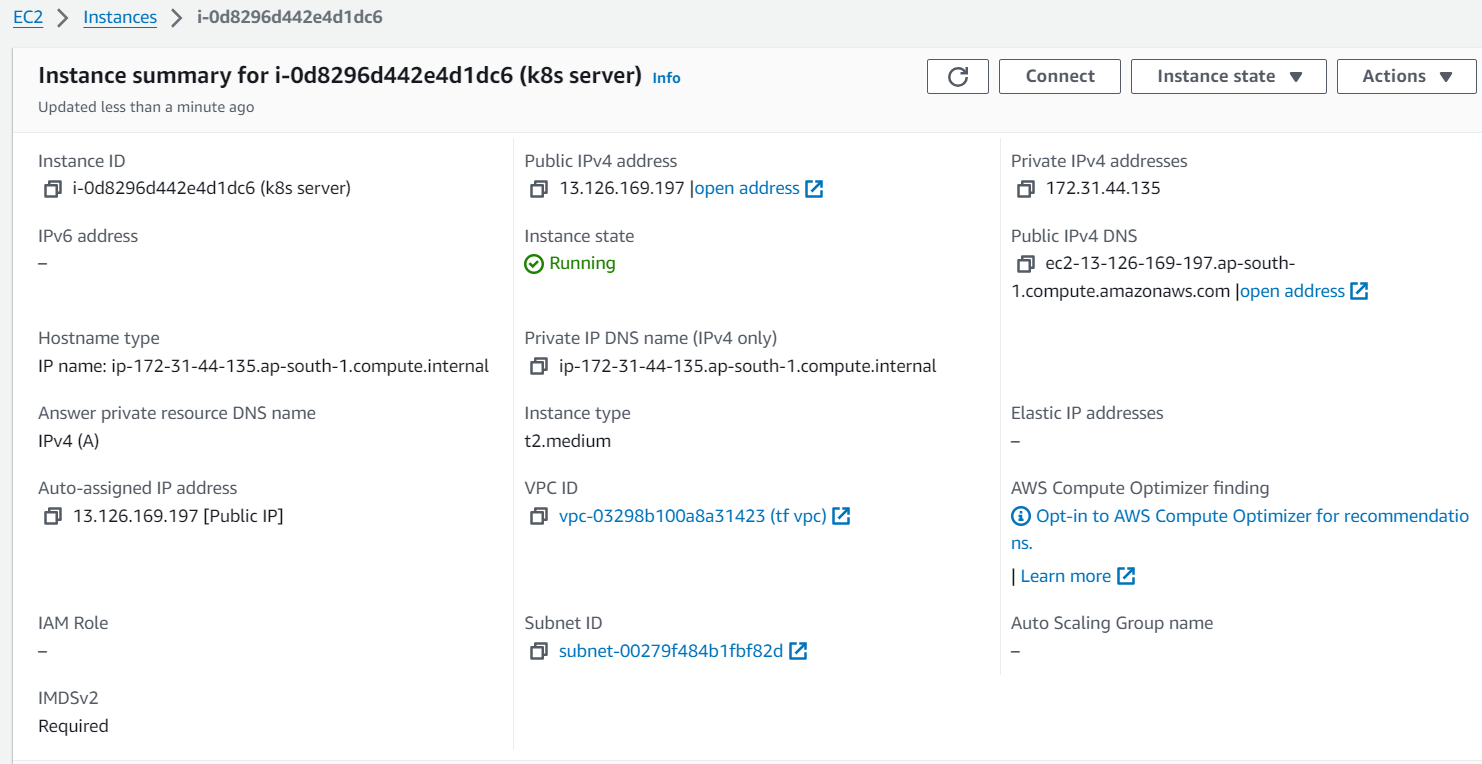
Logged in to AWS Management Console.  
• Launched EC2 Instance: (virtual machine - Ubuntu).

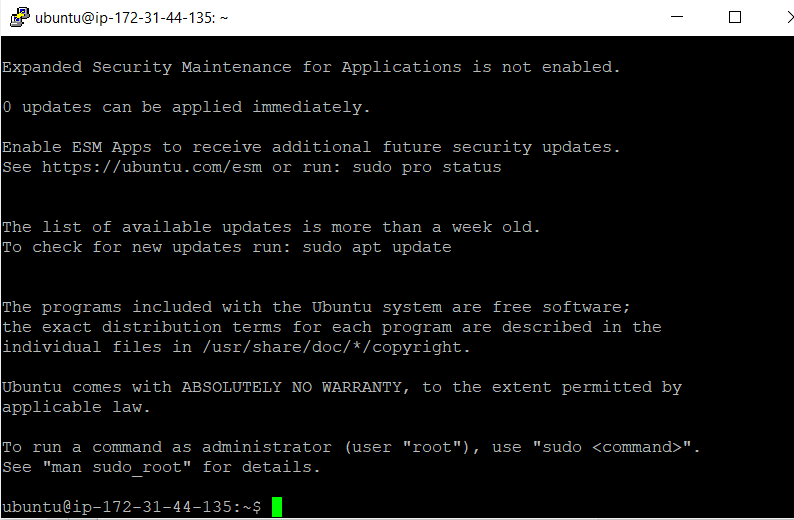
And security groups to allowed HTTP (port 80) and SSH (port 22) traffic with instance type - T2.medium and 15Gb disk size and created PPK file.

**Connect to Your EC2 Instance**

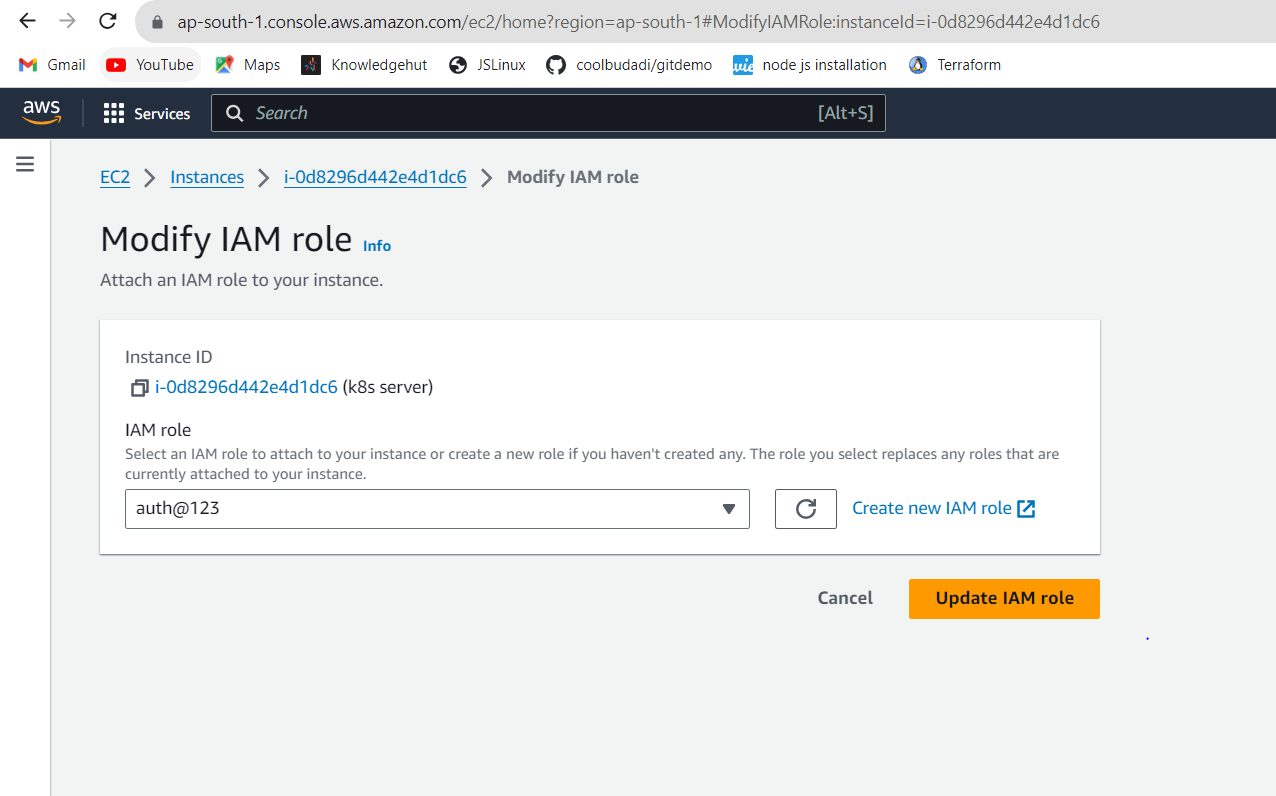
•    Used SSH to connect to EC2 instance using the PPK file via Putty as below



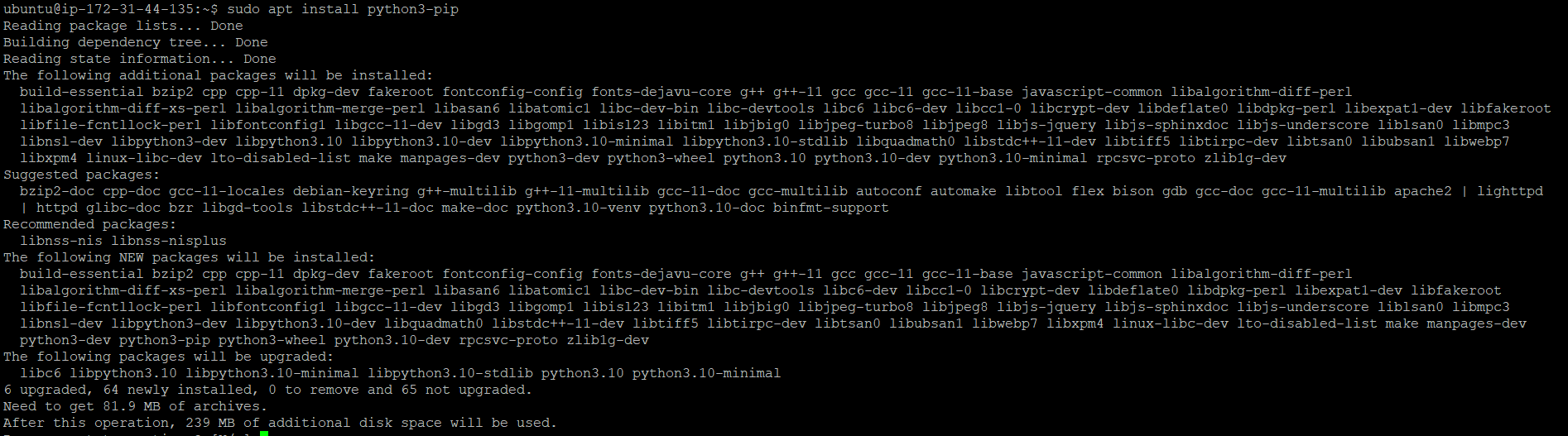




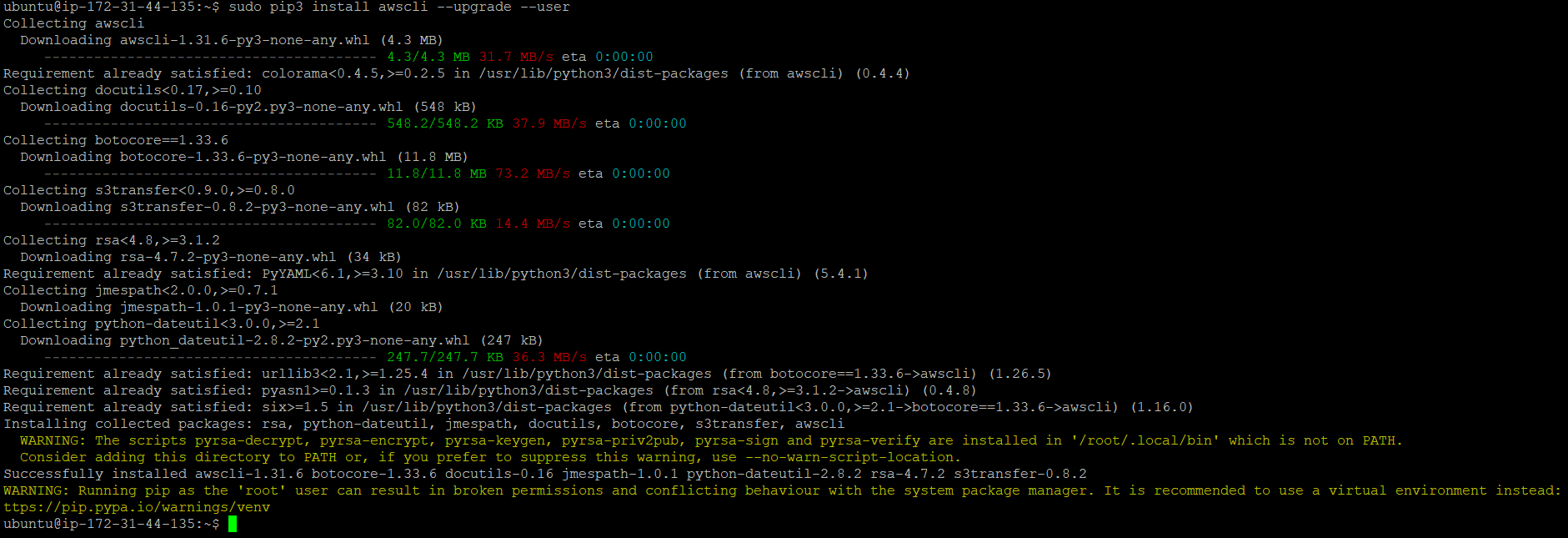
Attached IAM role **AmazonEKSClusterPolicy** and **AmazonEKSServicePolicy** policies attached.



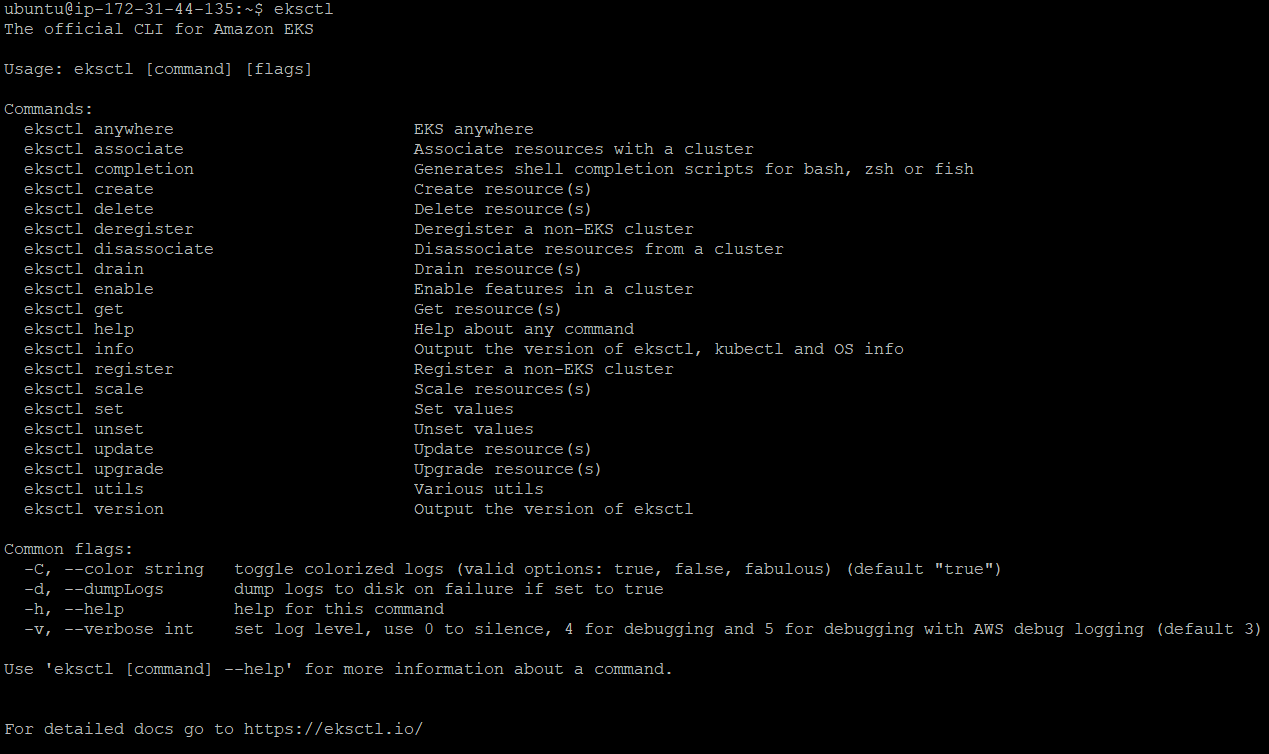
**Installed Python on to the machine :**



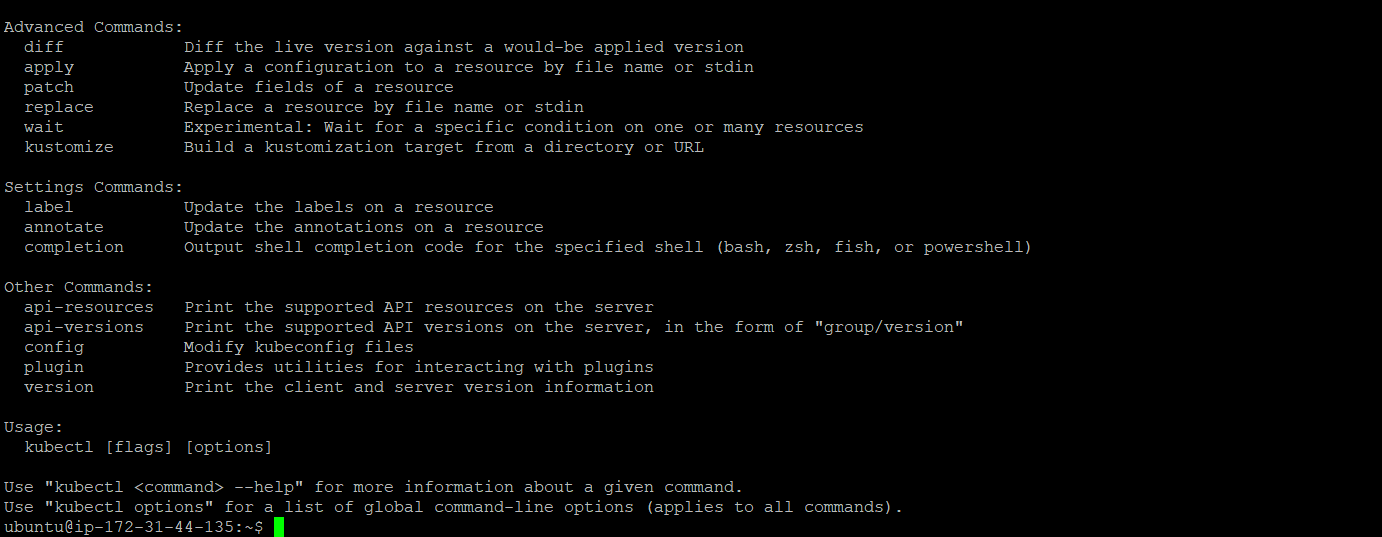
**Installed AWS cli on to the machine**



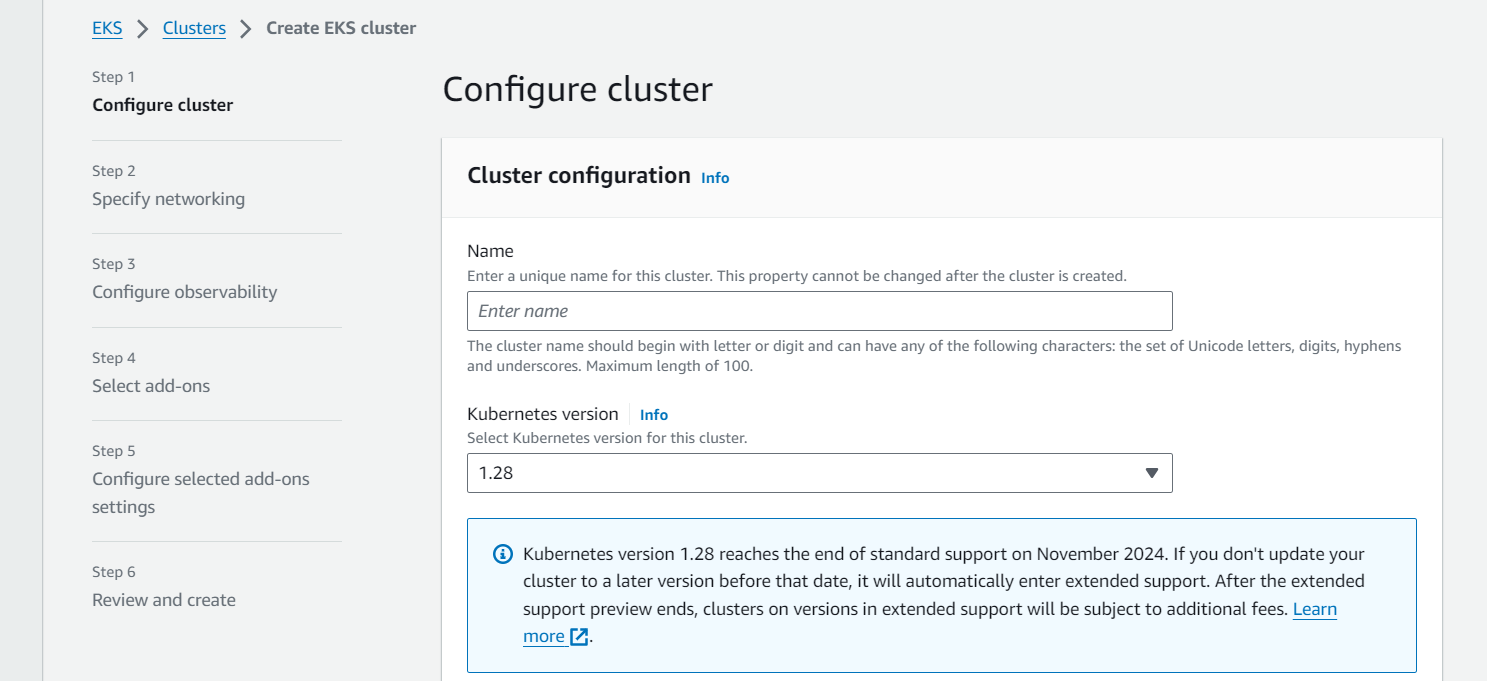
**Installed eksctl on to the machine :**



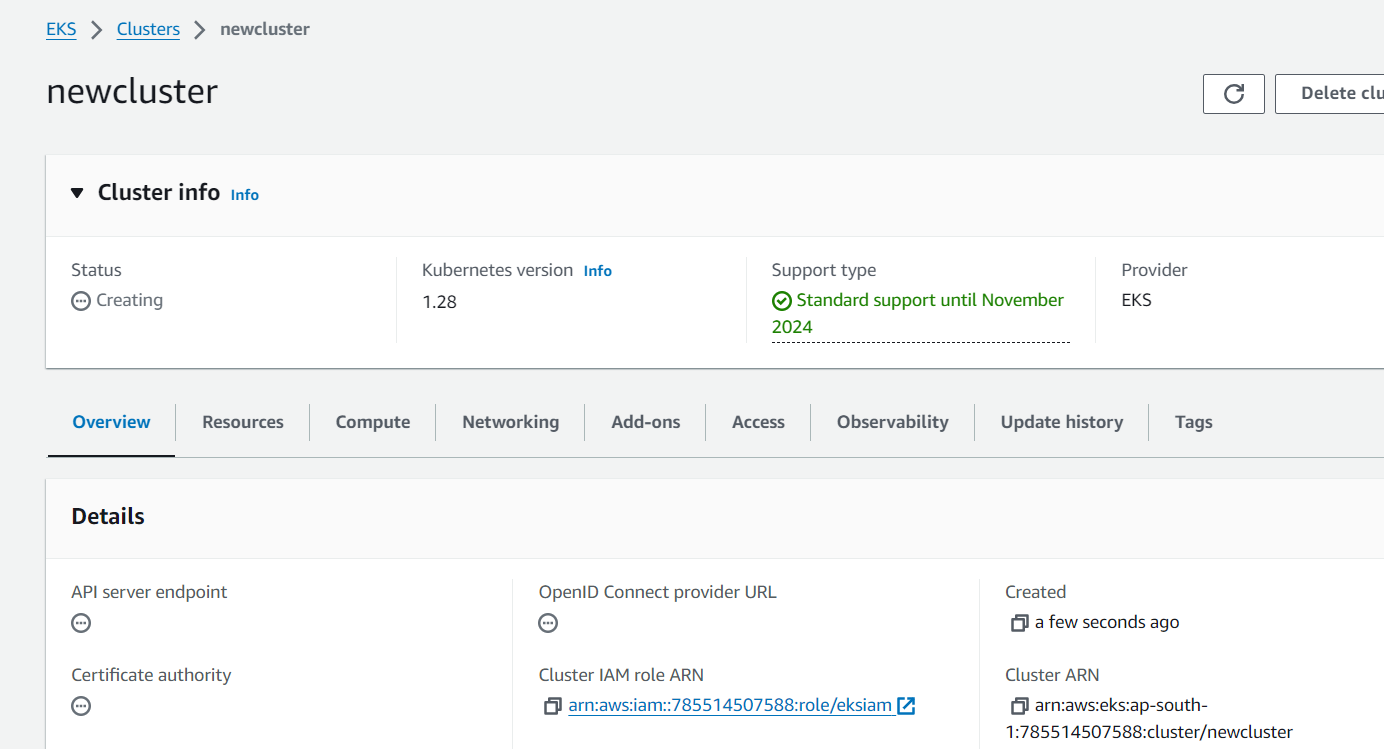
**Installed Kubectl on to the machine.**



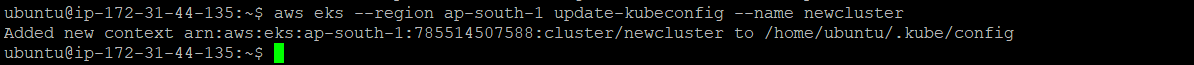
**Created a eks cluster on aws console as below:**



**Cluster name : newcluster**

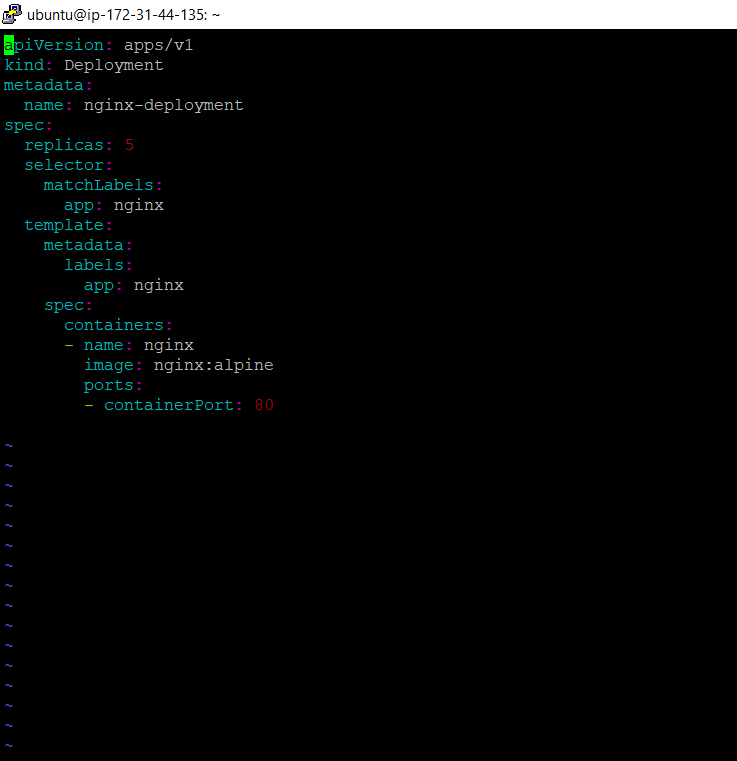


**Configured kubectl to Use Your Cluster .**



‘

**Created a file named nginx-deployment.yaml for deploying web server.**

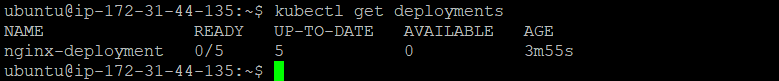


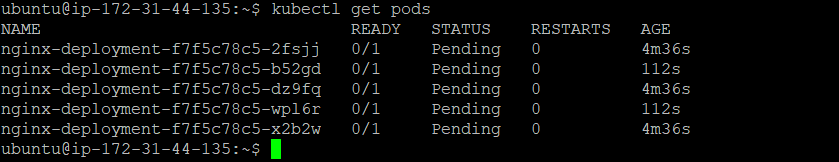
Exposed the NGINX Deployment:

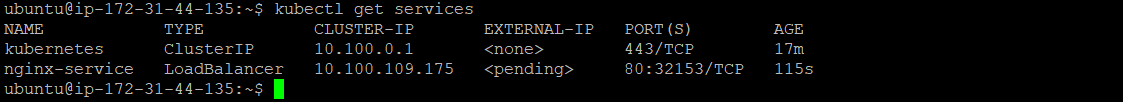


Monitored Deployment and Service by below command execution.

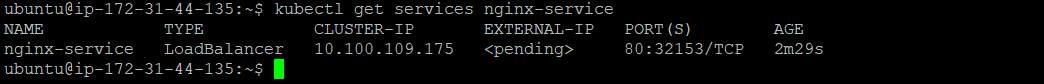
As we can see the 5 replicas created as below:

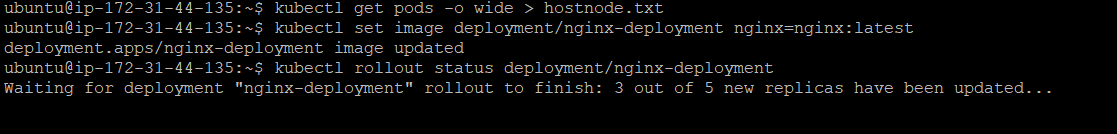






For Accessing NGINX Web Server below is the command.





Listed the running pods output with host node information in a file named hostnode.txt. Once done, update the container image to nginx:latest.

