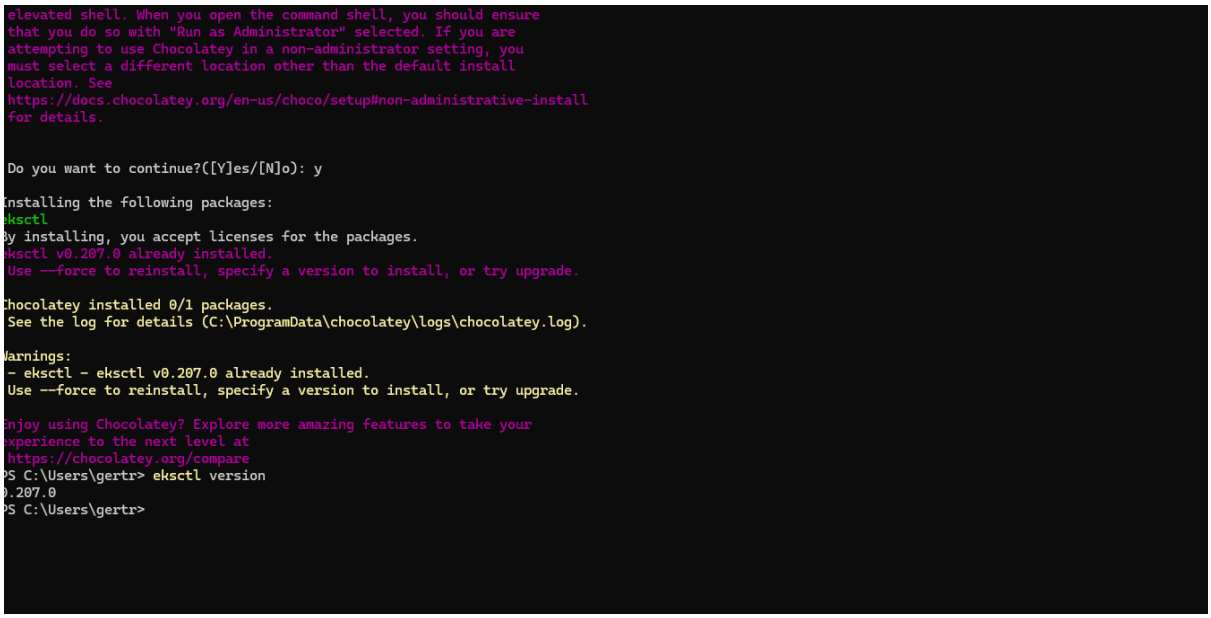
Containerized Nginx Deployment on Amazon Elastic Container Service for Kubernetes (EKS)

This lab demonstrates the steps to deploy and manage a containerized Nginx application on an Amazon Elastic Container Service for Kubernetes cluster.

The procedures are highlighted below:

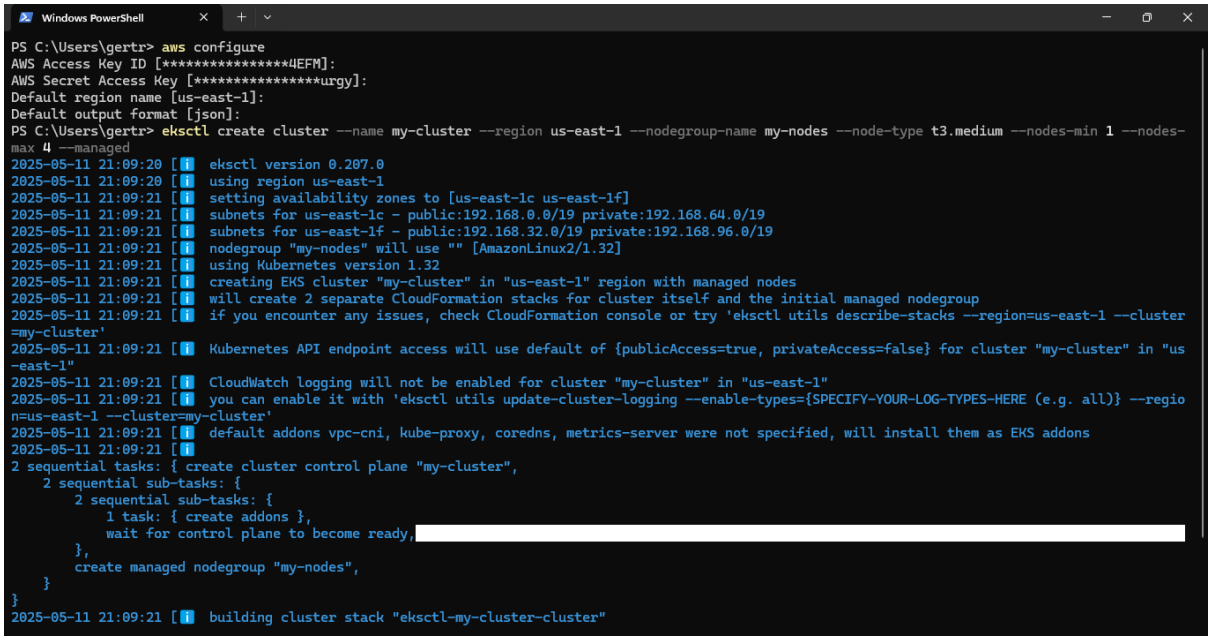
EKSCTL was installed because it was needed to make API calls to AWS for the provisioning and management of the EKS cluster.

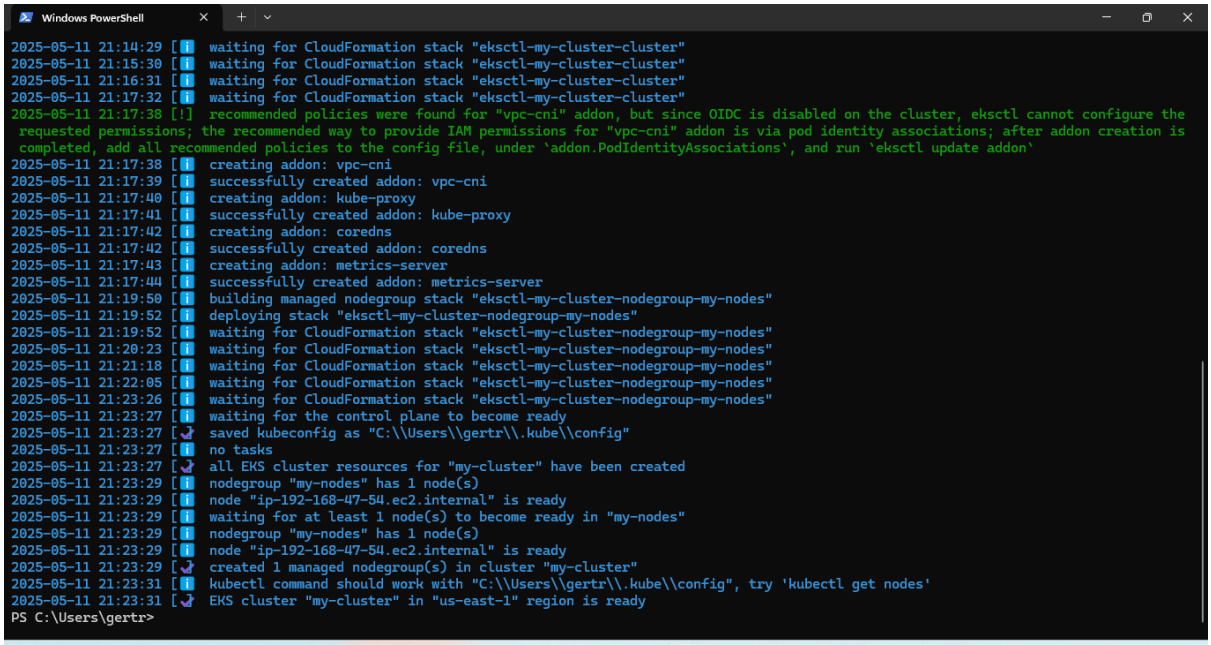


Kubectl was also installed to interact with the Kubernetes control plane in other to manage applications, nodes and pods in the cluster upon provisioning.

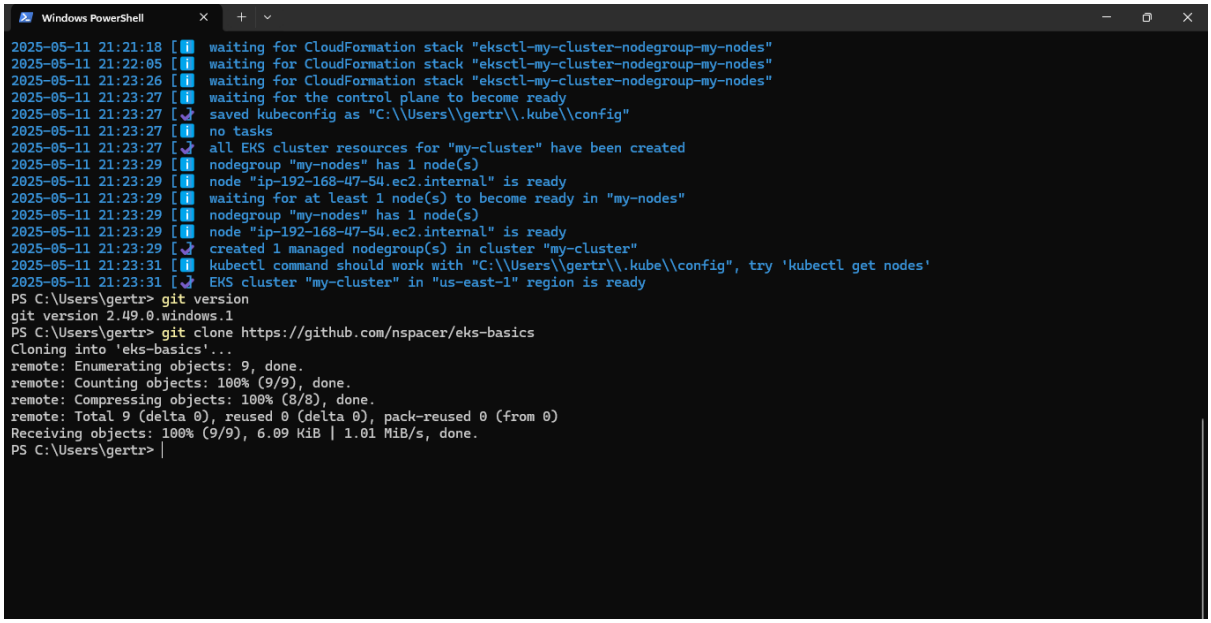
```
PS C:\Users\gertr> kubectl version --client
Client Version: v1.33.0
Kustomize Version: v5.6.0
PS C:\Users\gertr>
```

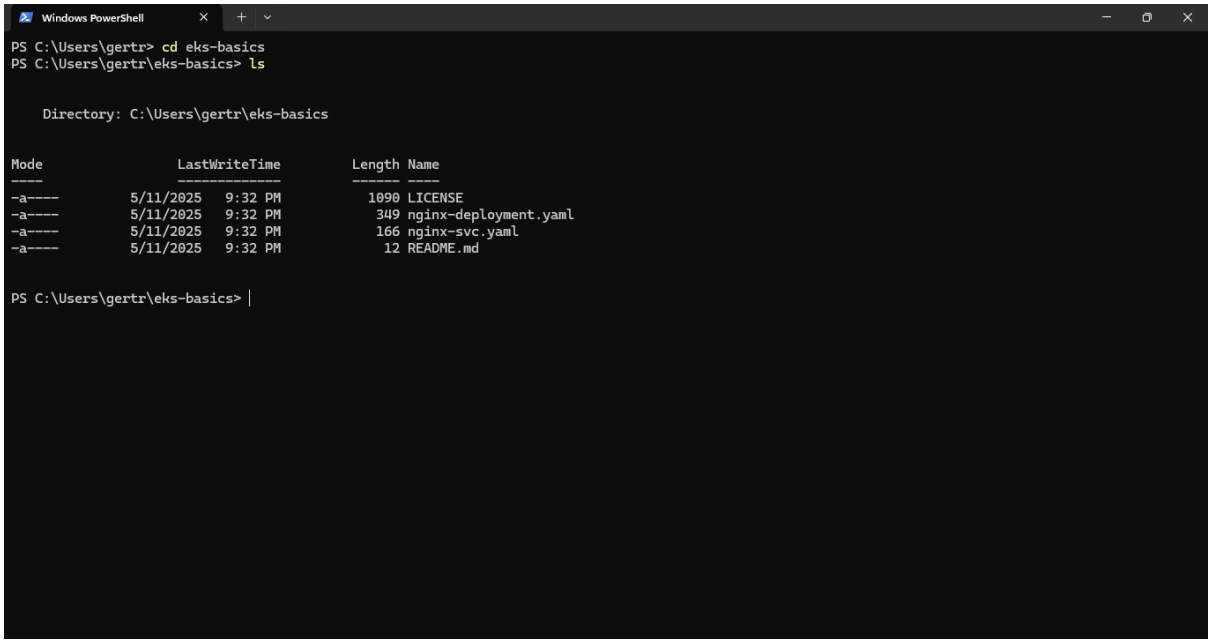
The EKS cluster was provisioned with the configurations below using the eksctl command line tool

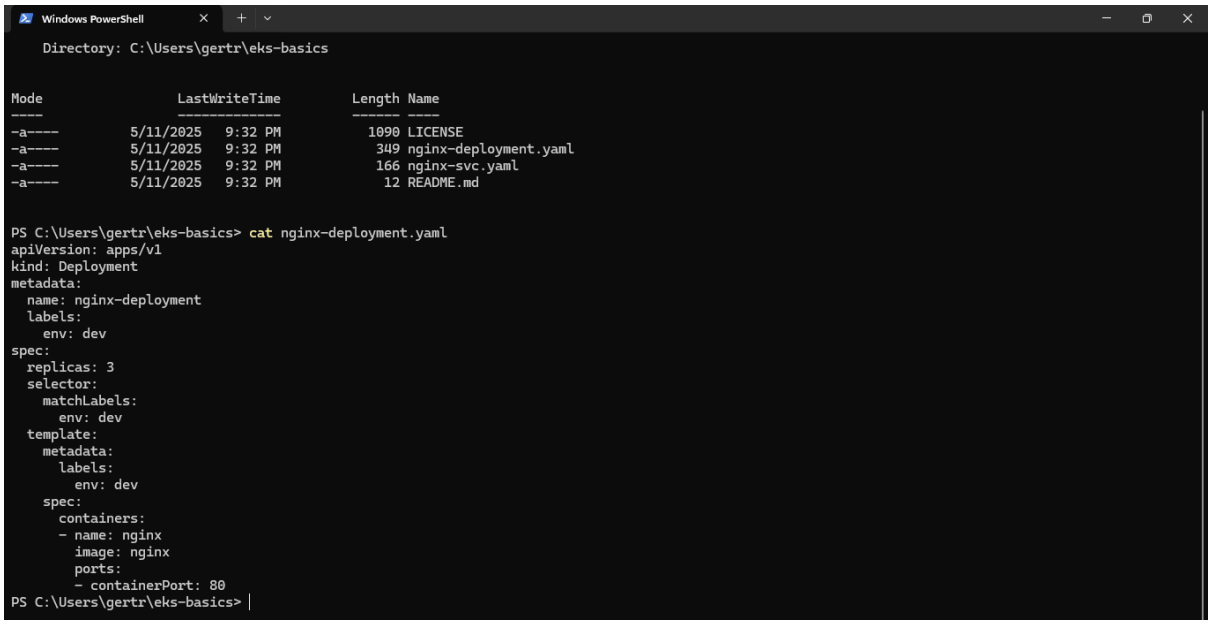


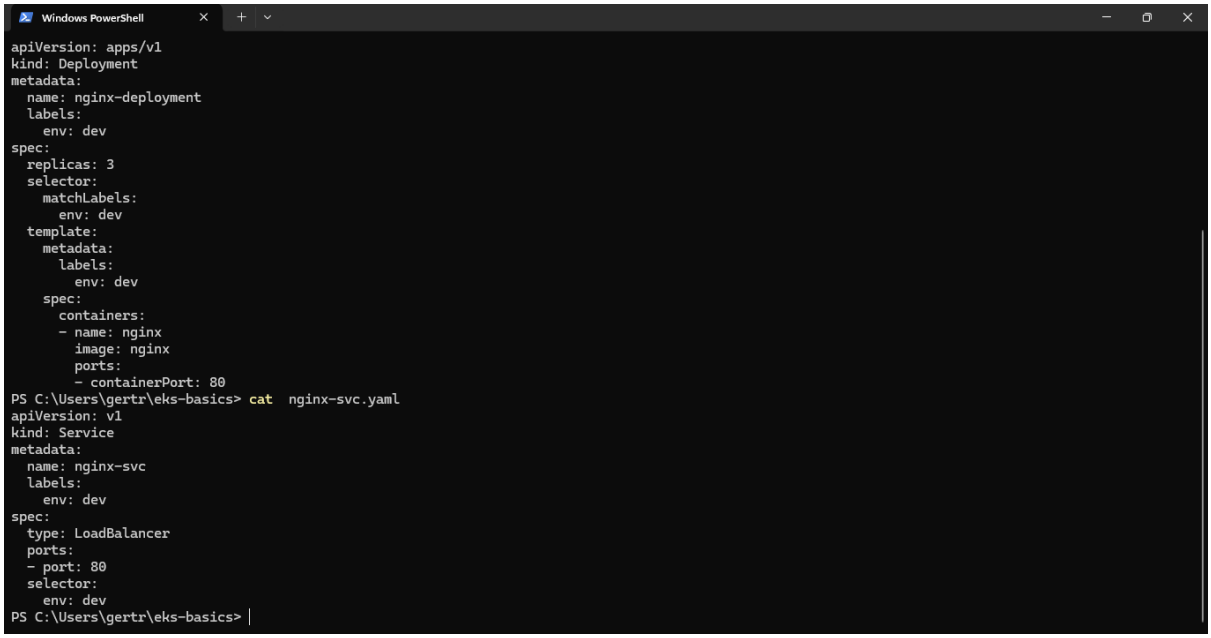


A deployment file and a service file was cloned from a git hub repository as shown below

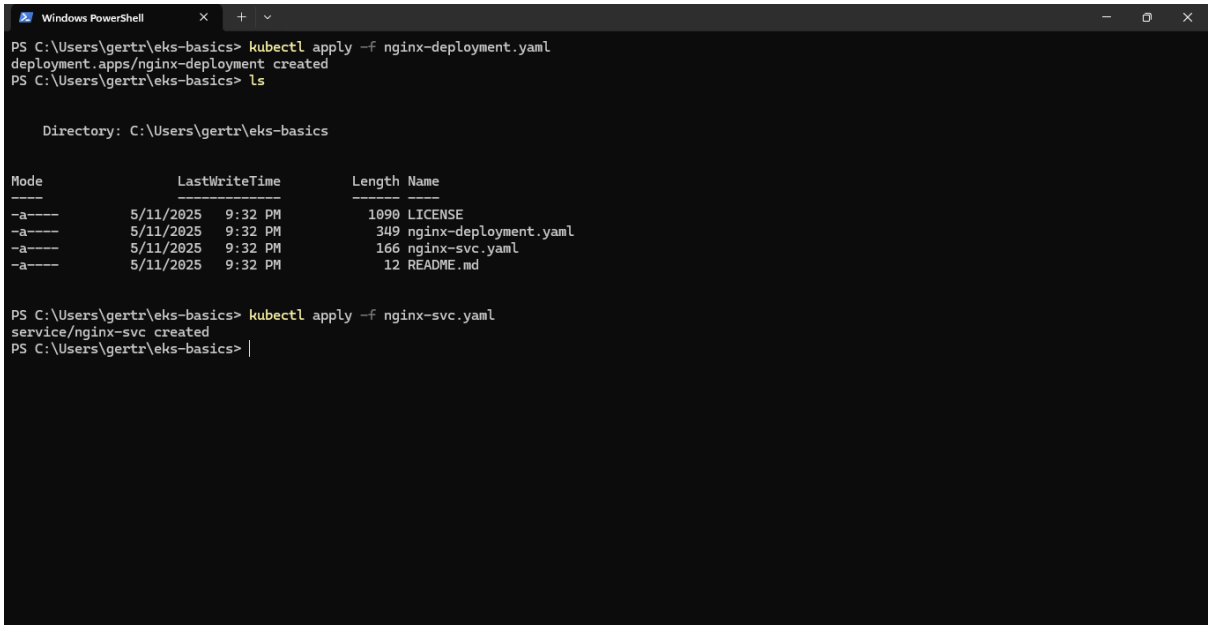



Windows PowerShell
PS C:\Users\gertr> cd eks-basics
PS C:\Users\gertr\eks-basics> ls
Directory: C:\Users\gertr\eks-basics
Mode LastWriteTime Length Name
-a---- 5/11/2025 9:32 PM 1090 LICENSE
-a---- 5/11/2025 9:32 PM 349 nginx-deployment.yaml
-a---- 5/11/2025 9:32 PM 166 nginx-svc.yaml
-a---- 5/11/2025 9:32 PM 12 README.md
PS C:\Users\gertr\eks-basics>


Windows PowerShell
Directory: C:\Users\gertr\eks-basics
Mode LastWriteTime Length Name
-a---- 5/11/2025 9:32 PM 1090 LICENSE
-a---- 5/11/2025 9:32 PM 349 nginx-deployment.yaml
-a---- 5/11/2025 9:32 PM 166 nginx-svc.yaml
-a---- 5/11/2025 9:32 PM 12 README.md
PS C:\Users\gertr\eks-basics> cat nginx-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
name: nginx-deployment
labels:
env: dev
spec:
replicas: 3
selector:
matchLabels:
env: dev
template:
metadata:
labels:
env: dev
spec:
containers:
- name: nginx
image: nginx
ports:
- containerPort: 80
PS C:\Users\gertr\eks-basics>



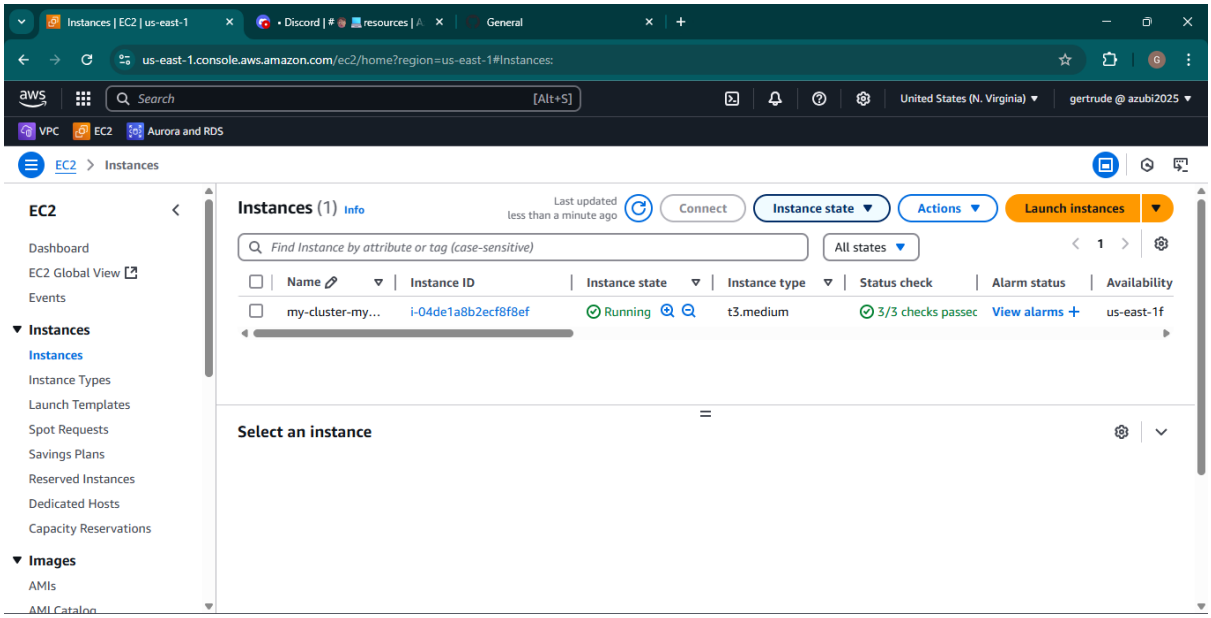
After cloning the repository containing the Deployment and Service files, both files were applied to the Kubernetes cluster to ensure that the ngnix application is deployed and running in the cluster and also ensure that the application is exposed to the public internet or internal network through the Service.

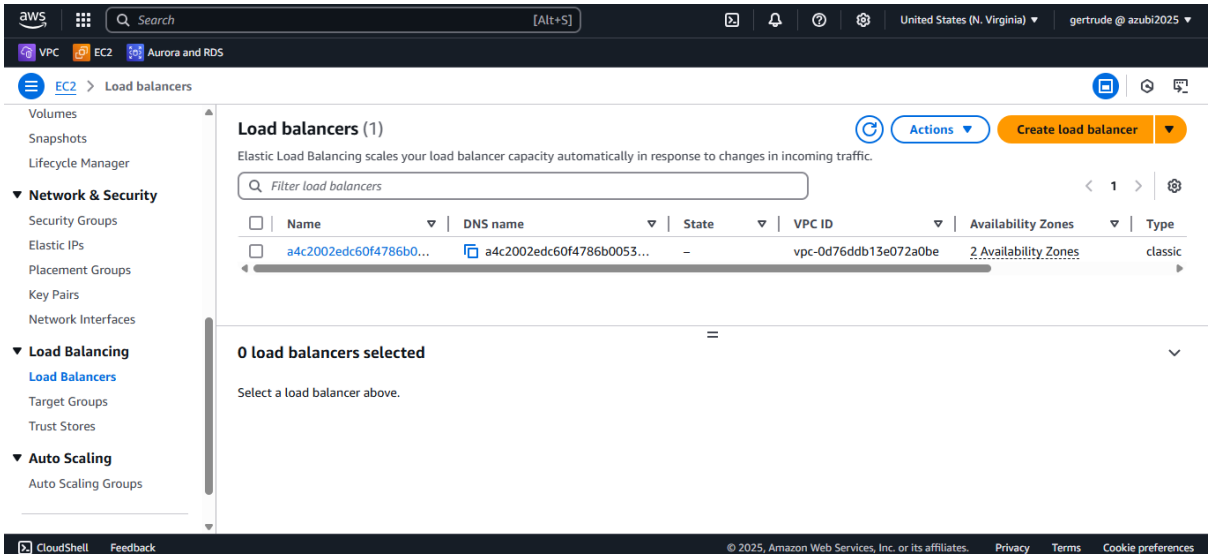


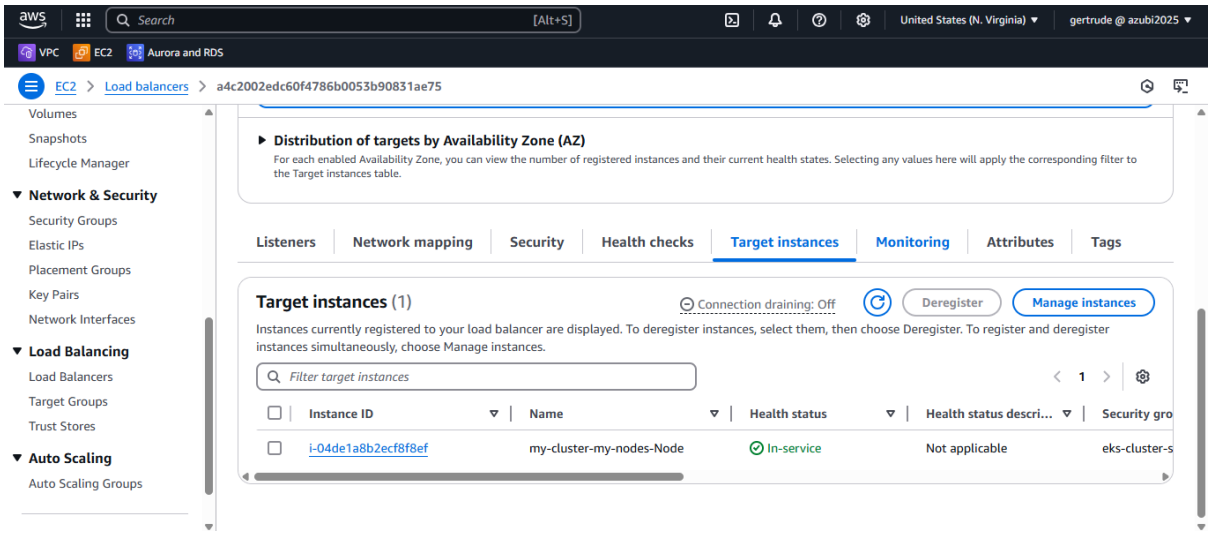
The nodes were ready and the pods were up and running as well. The load balancer was attached to the node as well.

```
Windows PowerShell
PS C:\Users\gertr\eks-basics> kubectl get service
NAME         TYPE           CLUSTER-IP       EXTERNAL-IP                                                                  PORT(S)        AGE
kubernetes   ClusterIP      10.100.0.1       <none>                                                                       443/TCP        24m
nginx-svc    LoadBalancer   10.100.254.155   a4c2002edc60f4786b0053b90831ae75-1719180930.us-east-1.elb.amazonaws.com   80:31810/TCP   94s
PS C:\Users\gertr\eks-basics> kubectl get deployment
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           3m12s
PS C:\Users\gertr\eks-basics> kubectl get nodes
NAME                           STATUS   ROLES    AGE   VERSION
ip-192-168-47-54.ec2.internal   Ready    <none>   19m   v1.32.3-eks-473151a
PS C:\Users\gertr\eks-basics> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6b9f544669-45zs8   1/1     Running   0          4m17s
nginx-deployment-6b9f544669-bbzkt   1/1     Running   0          4m17s
nginx-deployment-6b9f544669-wv26b   1/1     Running   0          4m17s
PS C:\Users\gertr\eks-basics>
```

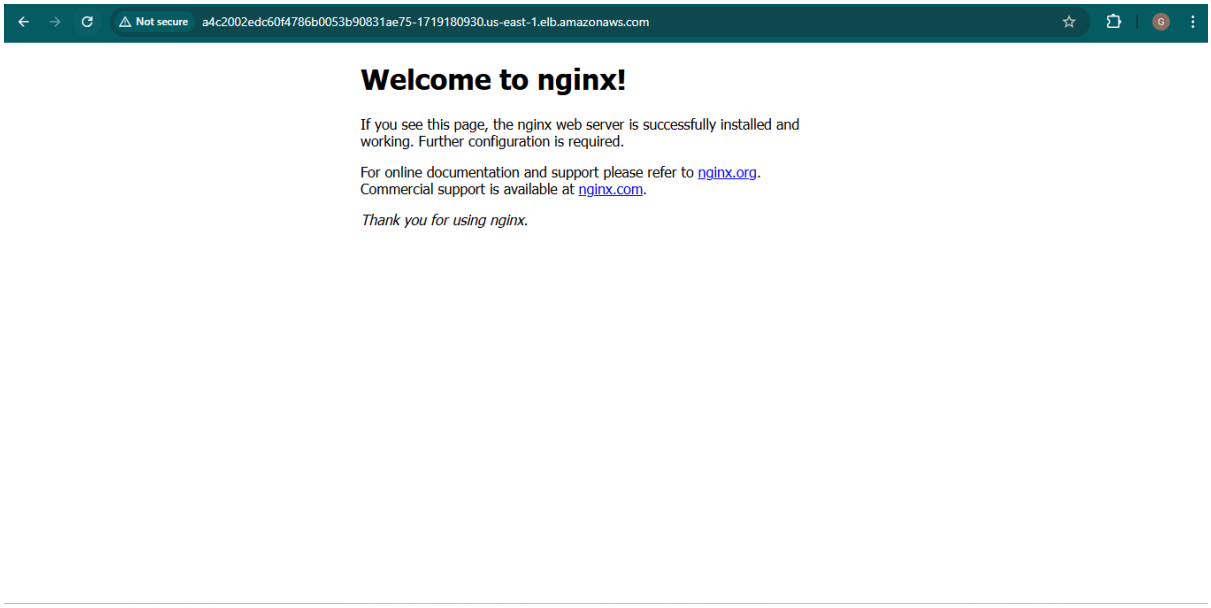
This was verified from the AWS console.





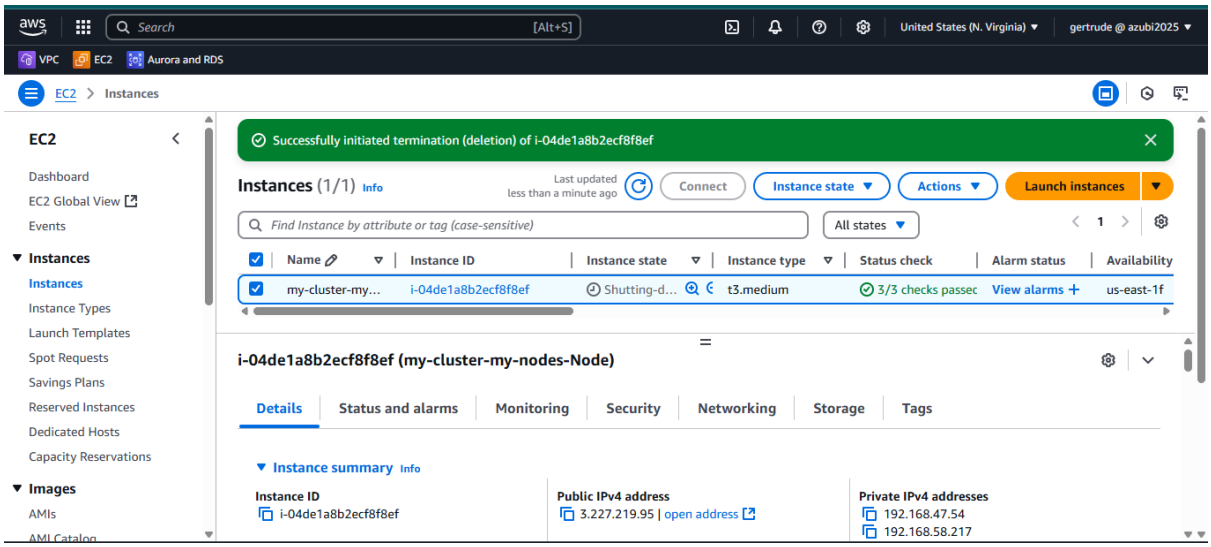


After pasting the DNS of the load balancer on the web browser, this was displayed.

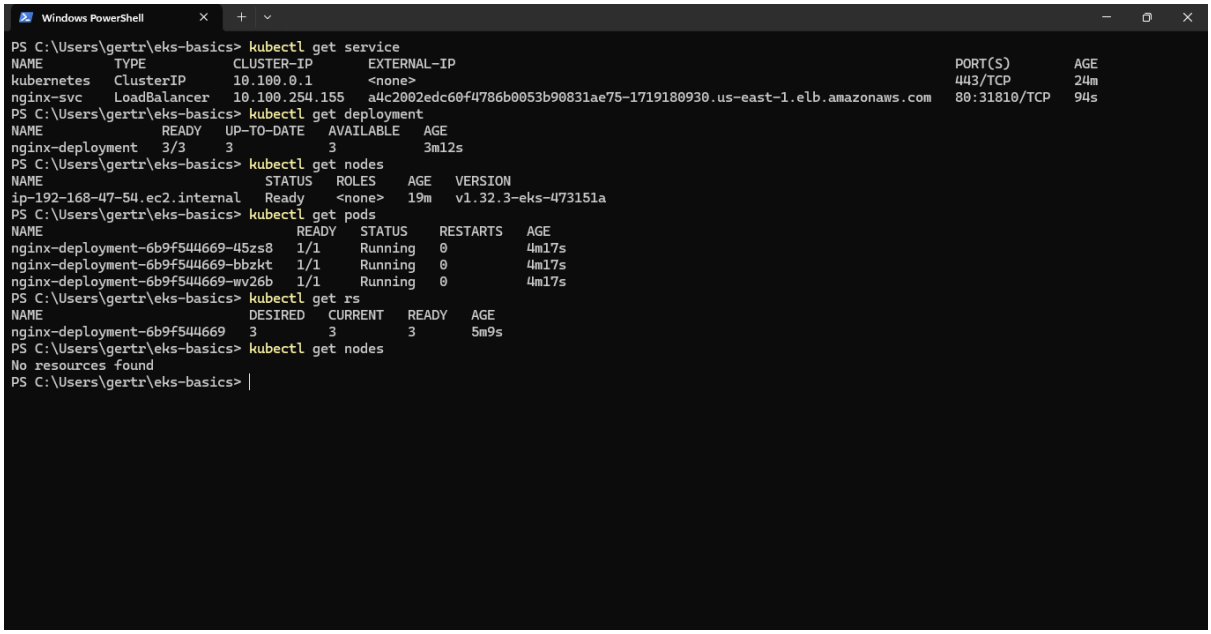


I tested cluster availability by stopping nodes and monitoring the status.

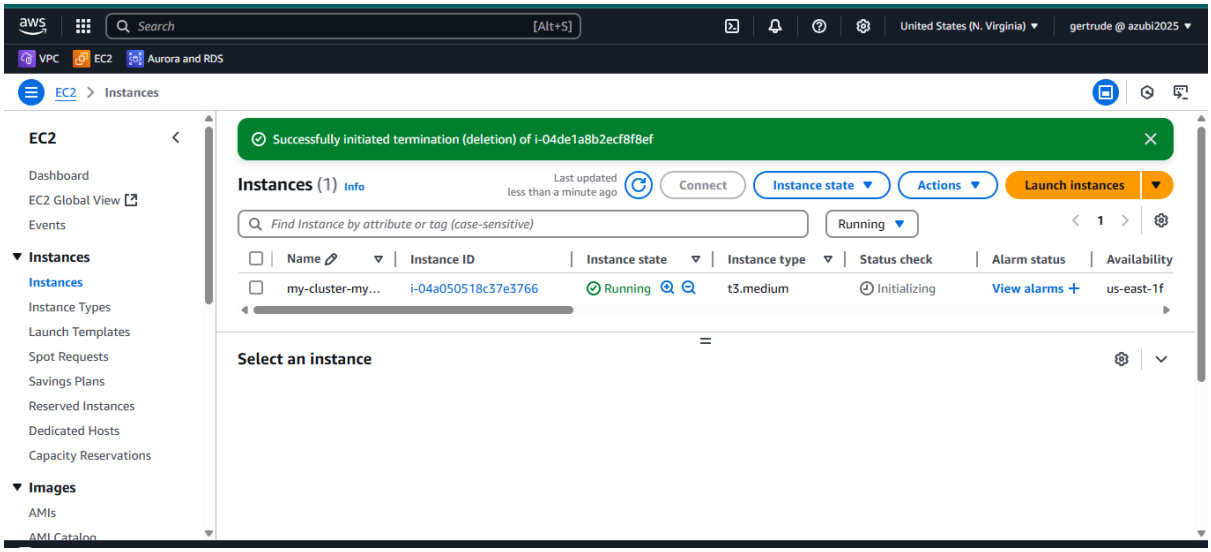
The screenshot below shows that the node was shutting down .



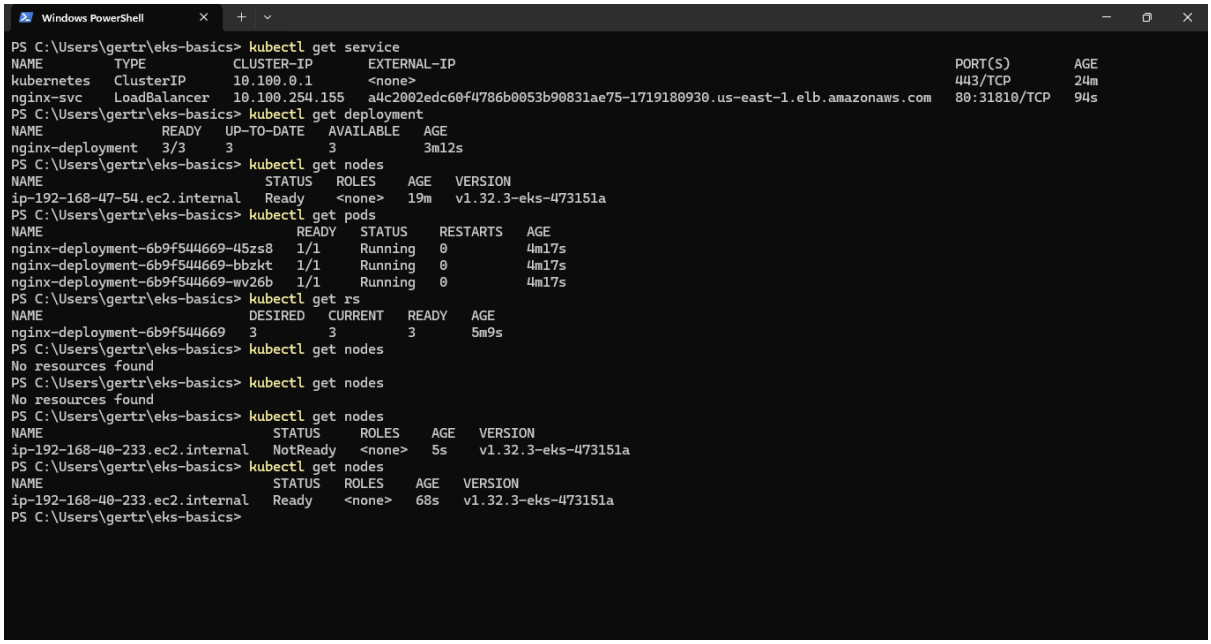
After sometime it was confirmed that there was no resource available meaning the node was successfully terminated.



After sometime, there was an initiation of the node on the console



And the node was ready after a period as shown below.



As expected, when nodes were terminated, Kubernetes detected the loss and automatically provisioned new ones, ensuring high availability and reducing administrative burden.