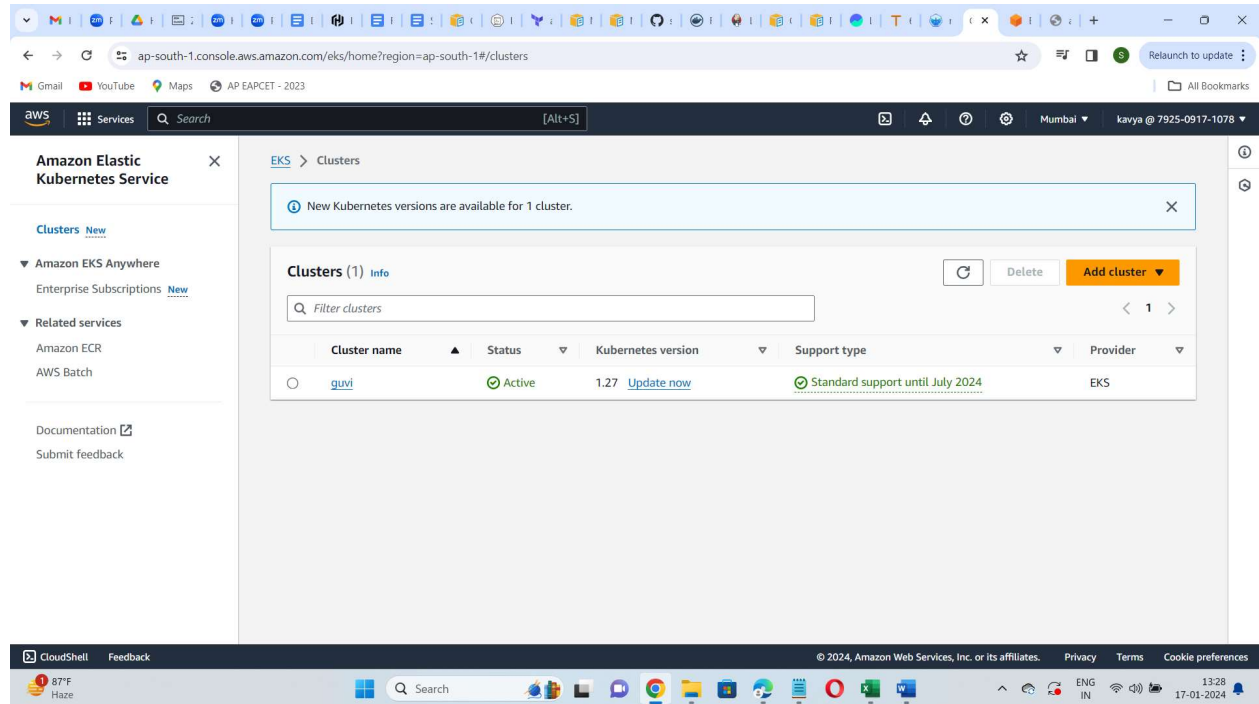
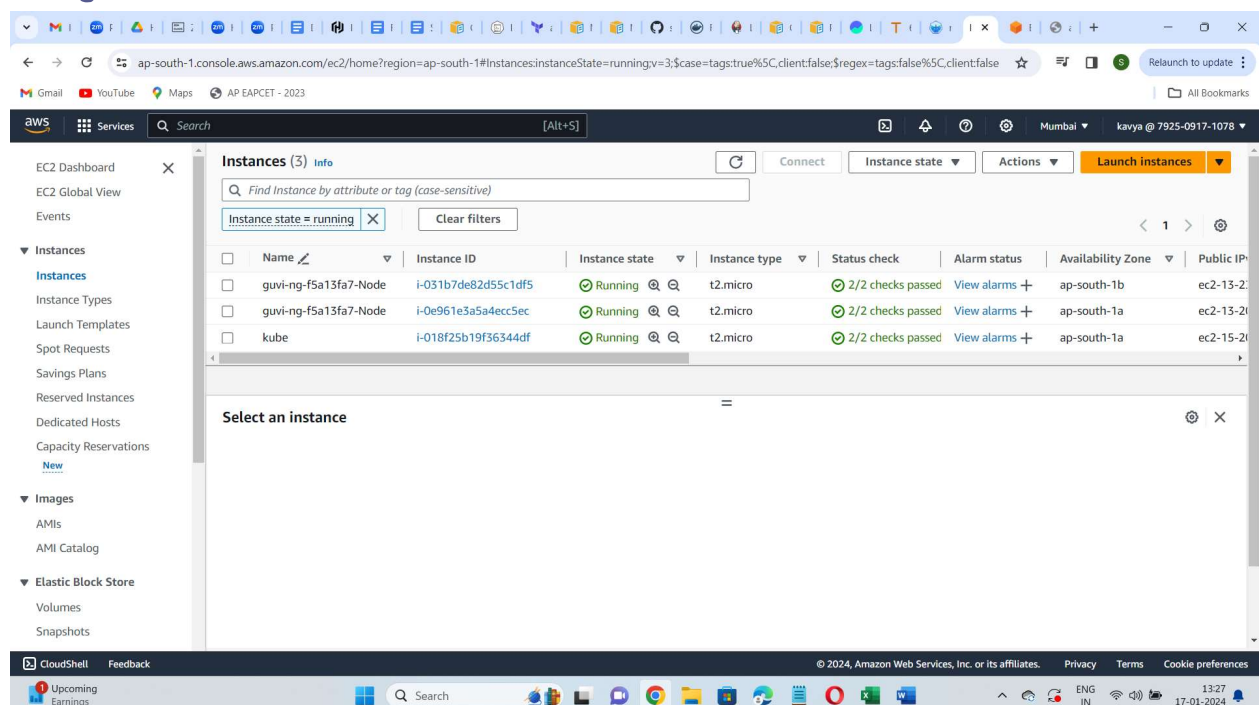


Create the K8s EKS, further you have to do the deployment of Nginx application

The below image shows the cluster created



Along with cluster two ec2 machines are created



Along with cluster vpc also created

The screenshot displays the AWS Management Console interface for the 'Your VPCs (2)' page. The left sidebar shows the navigation menu with 'Virtual private cloud' expanded. The main content area shows a table of VPCs:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP o
-	vpc-0243f75401e554f7b	Available	172.31.0.0/16	-	dopt-0t
eksctl-guvi-cluster/VPC	vpc-0a1b252b3d4ec98bd	Available	192.168.0.0/16	-	dopt-0t

The bottom of the screen shows a Windows taskbar with the date 17-01-2024.

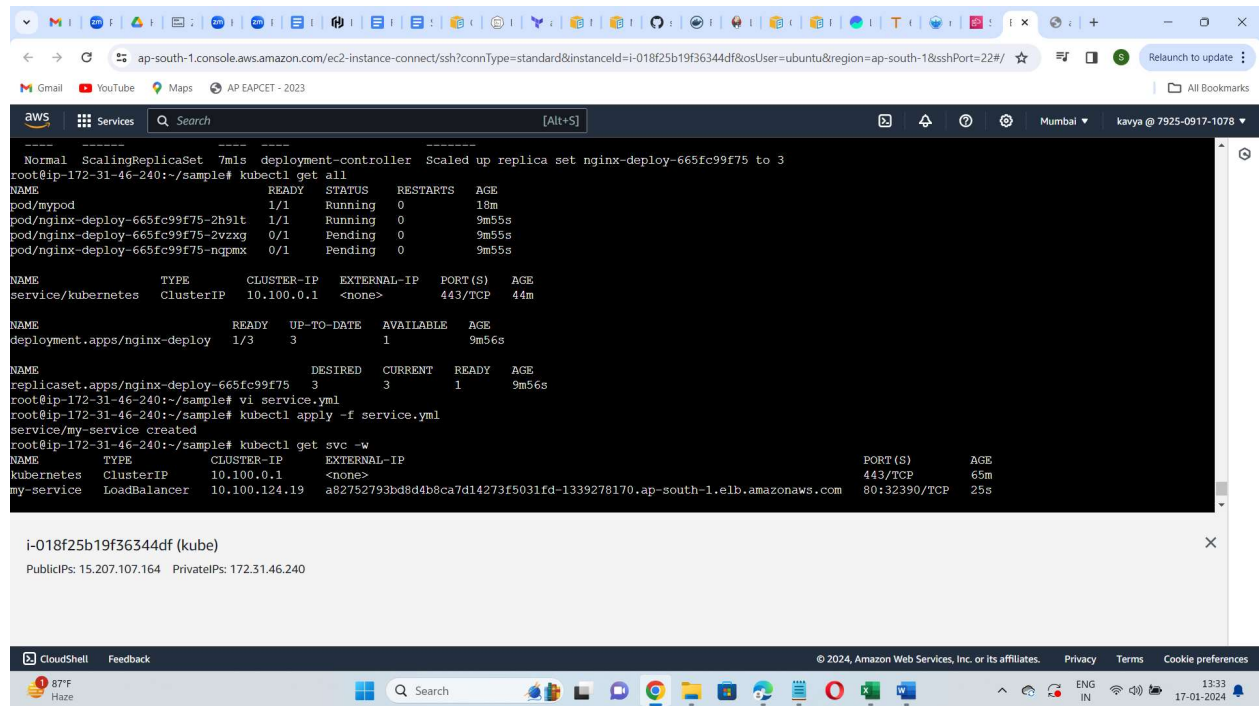
Cloud formation created

The screenshot displays the AWS Management Console interface for the 'CloudFormation' page. The left sidebar shows the navigation menu with 'CloudFormation' expanded. The main content area shows a table of stacks:

Stack name	Status	Created time	Description
eksctl-guvi-nodegroup-ng-f5a13fa7	CREATE_COMPLETE	2024-01-17 12:19:16 UTC+0530	EKS Managed Nodes (SSH access: false) [created by eksctl]
eksctl-guvi-cluster	CREATE_COMPLETE	2024-01-17 12:09:15 UTC+0530	EKS cluster (dedicated VPC: true, dedicated IAM: true) [created and managed by eksctl]

The bottom of the screen shows a Windows taskbar with the date 17-01-2024.

The deployed of nginx application and also created service to open through external-IP



The screenshot shows the AWS CloudShell interface with a terminal window. The terminal output displays the deployment of an nginx application and the creation of a service to expose it externally.

```
Normal ScalingReplicaSet 7m1s deployment-controller Scaled up replica set nginx-deploy-665fc99f75 to 3
root@ip-172-31-46-240:~/sample# kubectl get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/mypod                           1/1      Running   0           18m
pod/nginx-deploy-665fc99f75-2h91t    1/1      Running   0           9m55s
pod/nginx-deploy-665fc99f75-2vzqx    0/1      Pending   0           9m55s
pod/nginx-deploy-665fc99f75-ngqmx    0/1      Pending   0           9m55s

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/kubernetes                  ClusterIP     10.100.0.1    <none>          443/TCP    44m

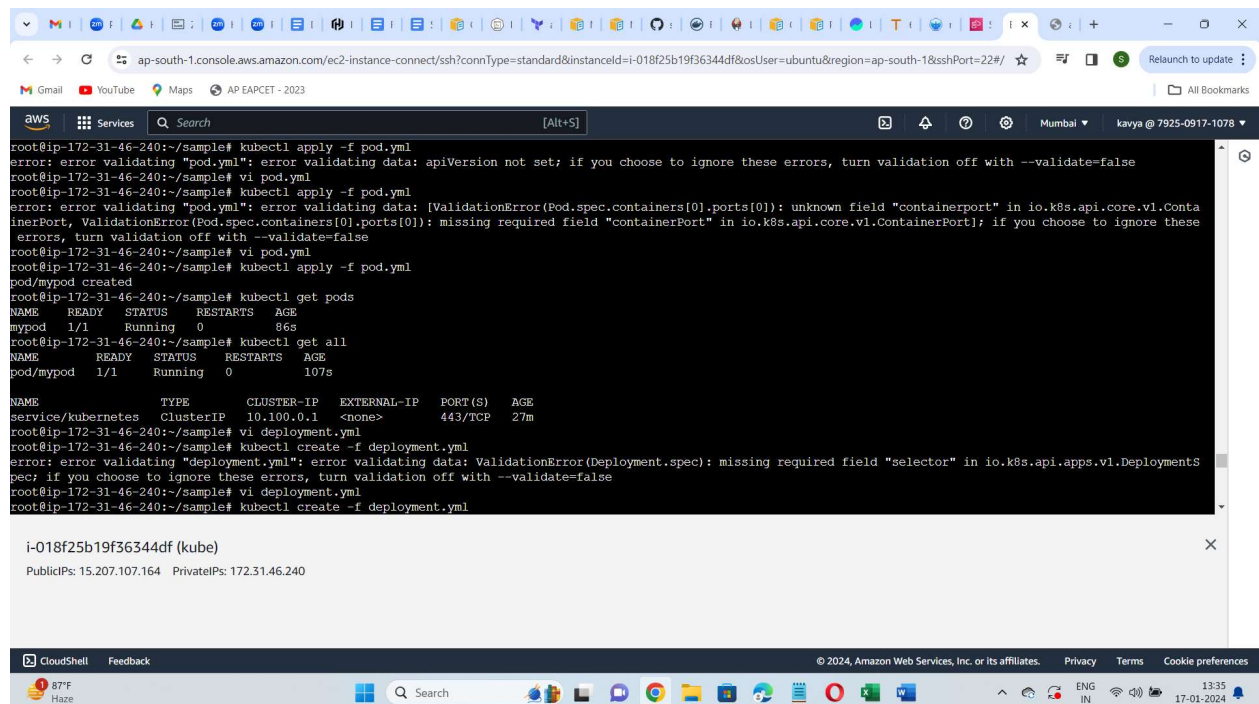
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/nginx-deploy        1/3      3              1            9m56s

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/nginx-deploy-665fc99f75 3          3          1        9m56s
root@ip-172-31-46-240:~/sample# vi service.yml
root@ip-172-31-46-240:~/sample# kubectl apply -f service.yml
service/my-service created
root@ip-172-31-46-240:~/sample# kubectl get svc -w
NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes                          ClusterIP     10.100.0.1    <none>          443/TCP    65m
my-service                          LoadBalancer  10.100.124.19 a82752793bd8d4b8ca7d14273f5031fd-1339278170.ap-south-1.elb.amazonaws.com 80:32390/TCP 25s
```

Below the terminal output, a metadata box for the kubelet shows the instance ID and IP addresses:

```
i-018f25b19f36344df (kube)
PublicIPs: 15.207.107.164 PrivateIPs: 172.31.46.240
```

The pod that i have created



The screenshot shows the AWS CloudShell interface with a terminal window. The terminal output displays the creation of a pod and a deployment, along with the application of a deployment file.

```
root@ip-172-31-46-240:~/sample# kubectl apply -f pod.yml
error: error validating "pod.yml": error validating data: apiVersion not set; if you choose to ignore these errors, turn validation off with --validate=false
root@ip-172-31-46-240:~/sample# vi pod.yml
root@ip-172-31-46-240:~/sample# kubectl apply -f pod.yml
error: error validating "pod.yml": error validating data: [ValidationError(Pod.spec.containers[0].ports[0]): unknown field "containerPort" in io.k8s.api.core.v1.ContainerPort, ValidationError(Pod.spec.containers[0].ports[0]): missing required field "containerPort" in io.k8s.api.core.v1.ContainerPort]; if you choose to ignore these errors, turn validation off with --validate=false
root@ip-172-31-46-240:~/sample# vi pod.yml
root@ip-172-31-46-240:~/sample# kubectl apply -f pod.yml
pod/mypod created
root@ip-172-31-46-240:~/sample# kubectl get pods
NAME    READY    STATUS    RESTARTS   AGE
mypod   1/1      Running   0           86s
root@ip-172-31-46-240:~/sample# kubectl get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/mypod                           1/1      Running   0           107s

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/kubernetes                  ClusterIP     10.100.0.1    <none>          443/TCP    27m
root@ip-172-31-46-240:~/sample# vi deployment.yml
root@ip-172-31-46-240:~/sample# kubectl create -f deployment.yml
error: error validating "deployment.yml": error validating data: ValidationError(Deployment.spec): missing required field "selector" in io.k8s.api.apps.v1.DeploymentSpec; if you choose to ignore these errors, turn validation off with --validate=false
root@ip-172-31-46-240:~/sample# vi deployment.yml
root@ip-172-31-46-240:~/sample# kubectl create -f deployment.yml
```

Below the terminal output, a metadata box for the kubelet shows the instance ID and IP addresses:

```
i-018f25b19f36344df (kube)
PublicIPs: 15.207.107.164 PrivateIPs: 172.31.46.240
```