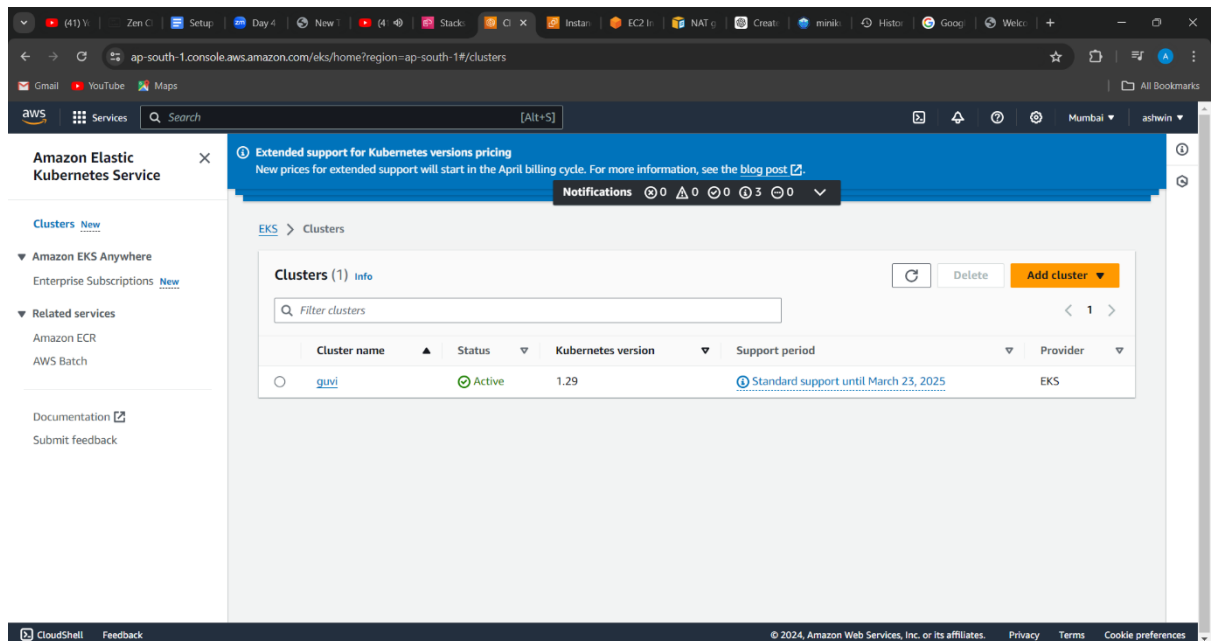
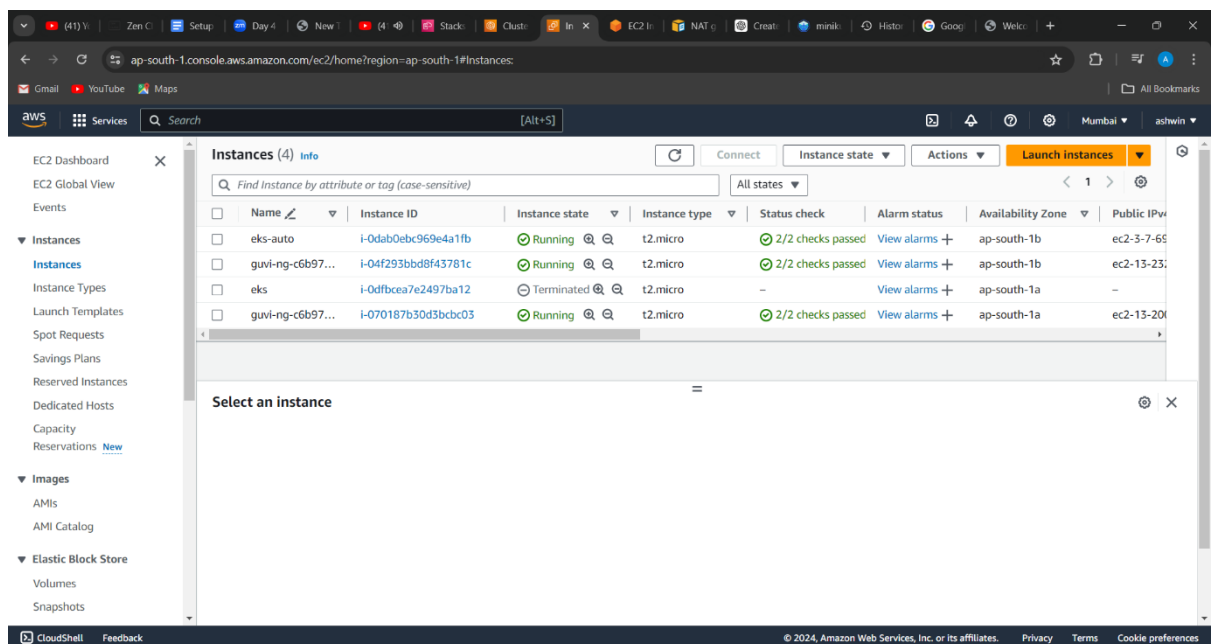


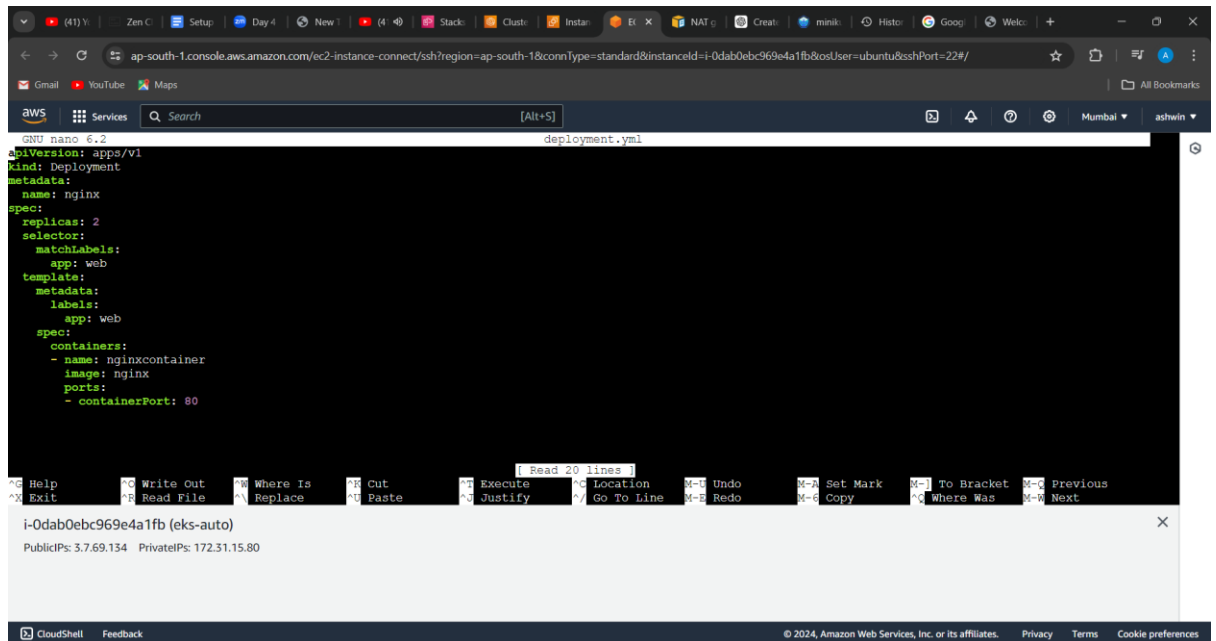
1. created eks cluster



2. 2nodes created



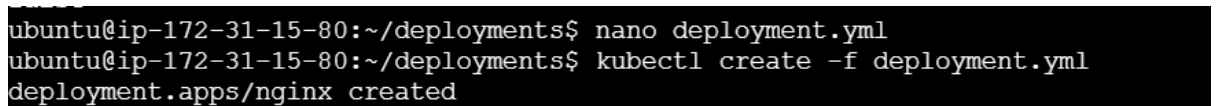
3. deployment file



The screenshot shows the AWS CloudShell interface with a terminal window displaying the content of a file named `deployment.yaml`. The file is a Kubernetes Deployment manifest for an application named `nginx`. The manifest includes metadata, a selector, and a template with a single container named `nginxcontainer` using the `nginx` image and port 80.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
        - name: nginxcontainer
          image: nginx
          ports:
            - containerPort: 80
```

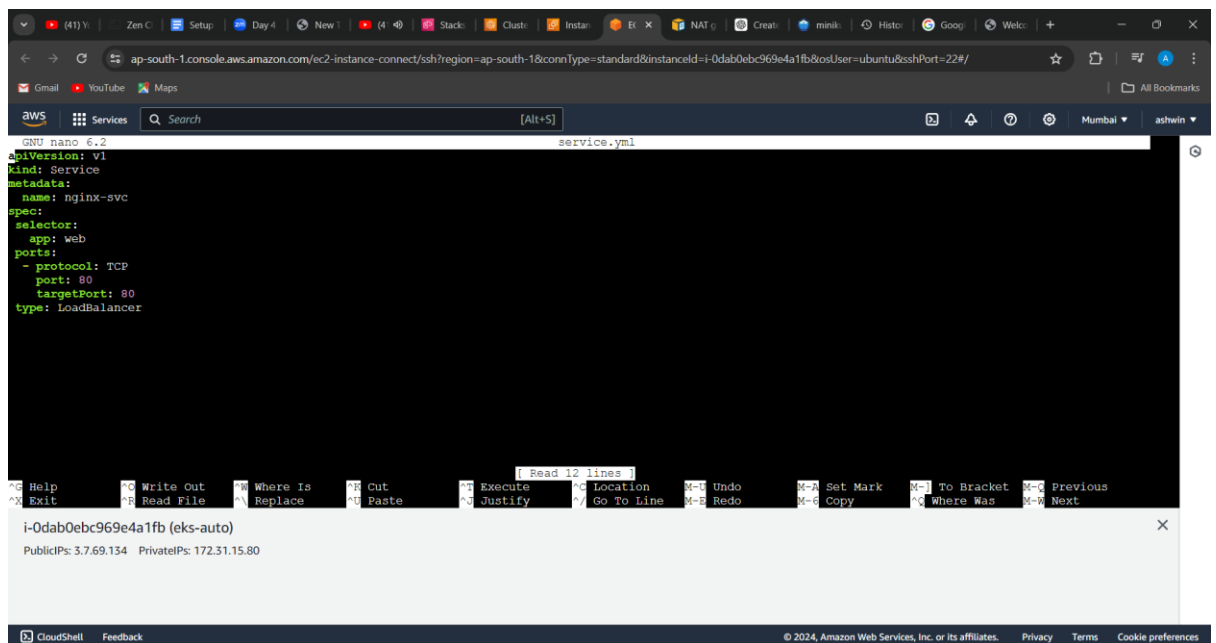
4.created deployment



The screenshot shows a terminal window with the following commands and output:

```
ubuntu@ip-172-31-15-80:~/deployments$ nano deployment.yaml
ubuntu@ip-172-31-15-80:~/deployments$ kubectl create -f deployment.yaml
deployment.apps/nginx created
```

5.created service file



The screenshot shows the AWS CloudShell interface with a terminal window displaying the content of a file named `service.yaml`. The file is a Kubernetes Service manifest for an application named `nginx-svc`. The manifest includes metadata, a selector, and a port definition for TCP on port 80, with a target port of 80 and a type of `LoadBalancer`.

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-svc
spec:
  selector:
    app: web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer
```

6. created a service

```
ubuntu@ip-172-31-15-80:~/deployments$ nano service.yml
ubuntu@ip-172-31-15-80:~/deployments$ kubectl create -f service.yml
service/nginx-svc created
```

7.output of nginx page.

