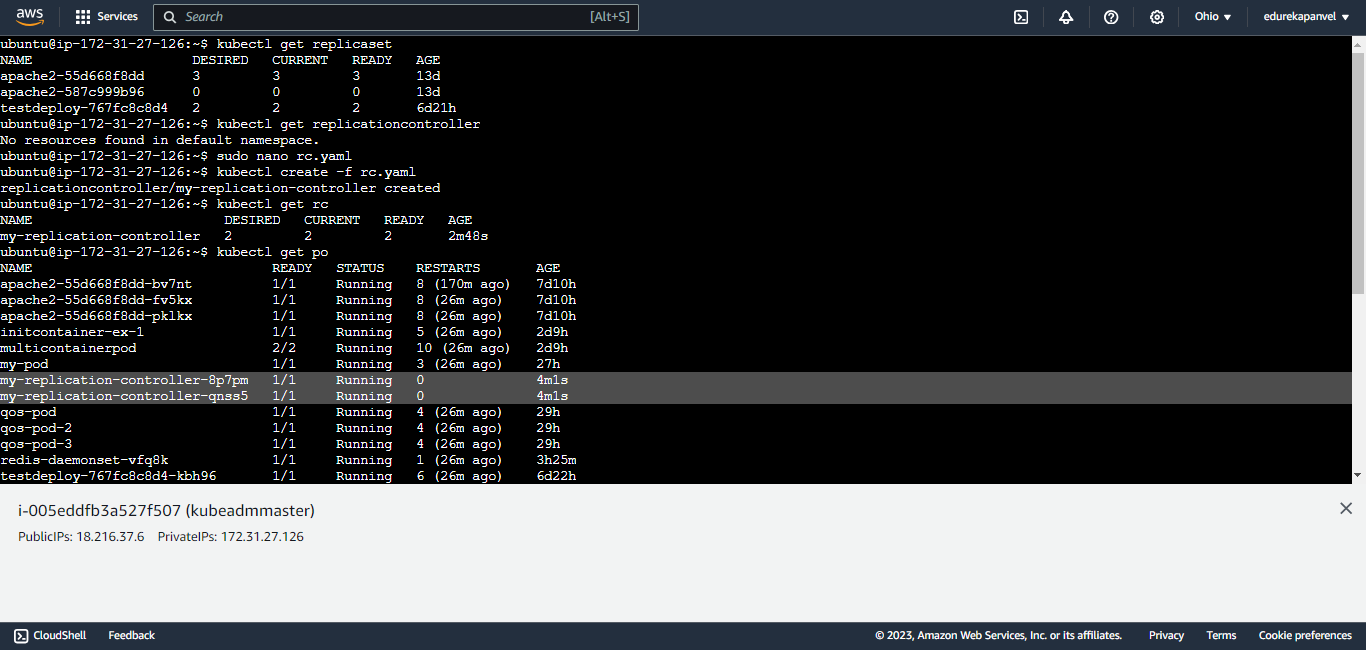
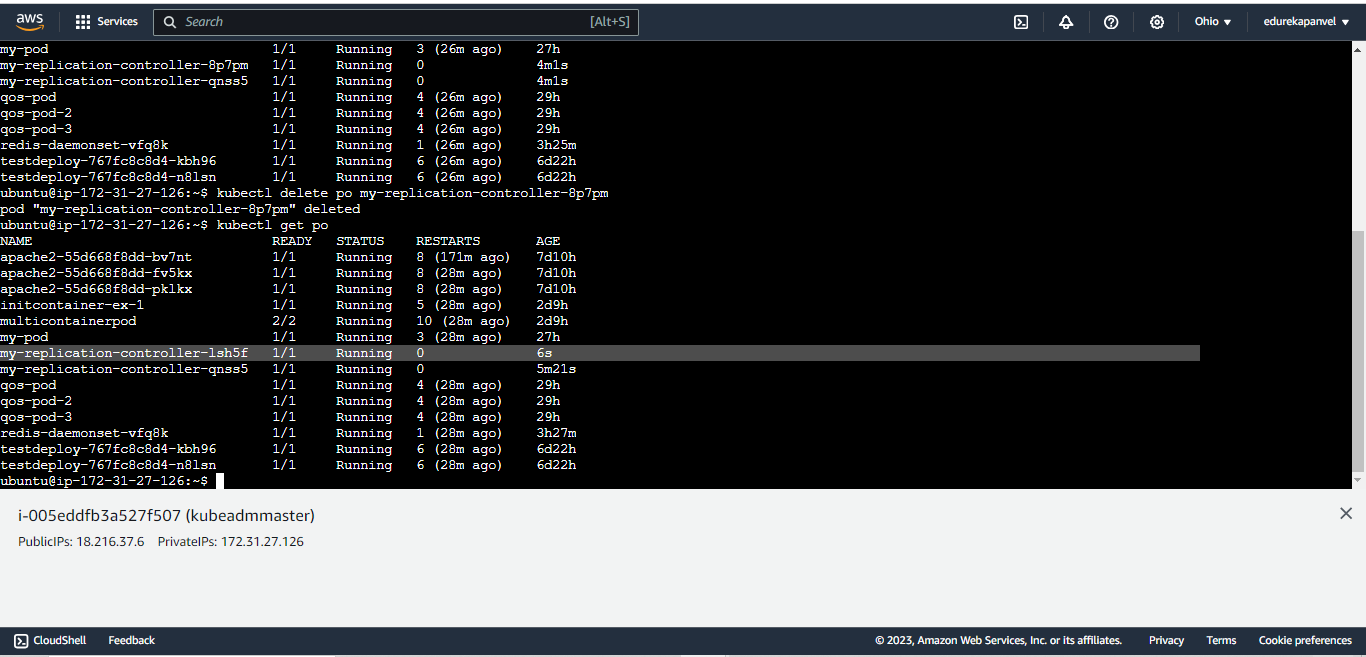
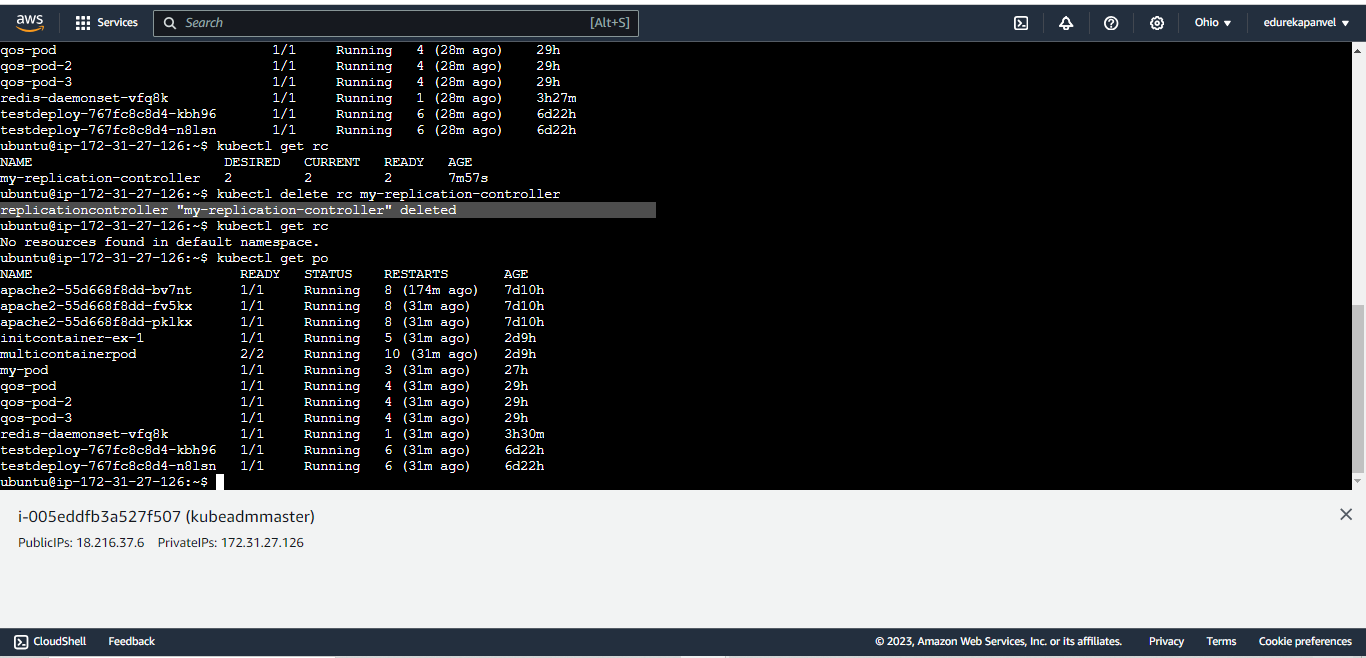
Lu1

Replication controller

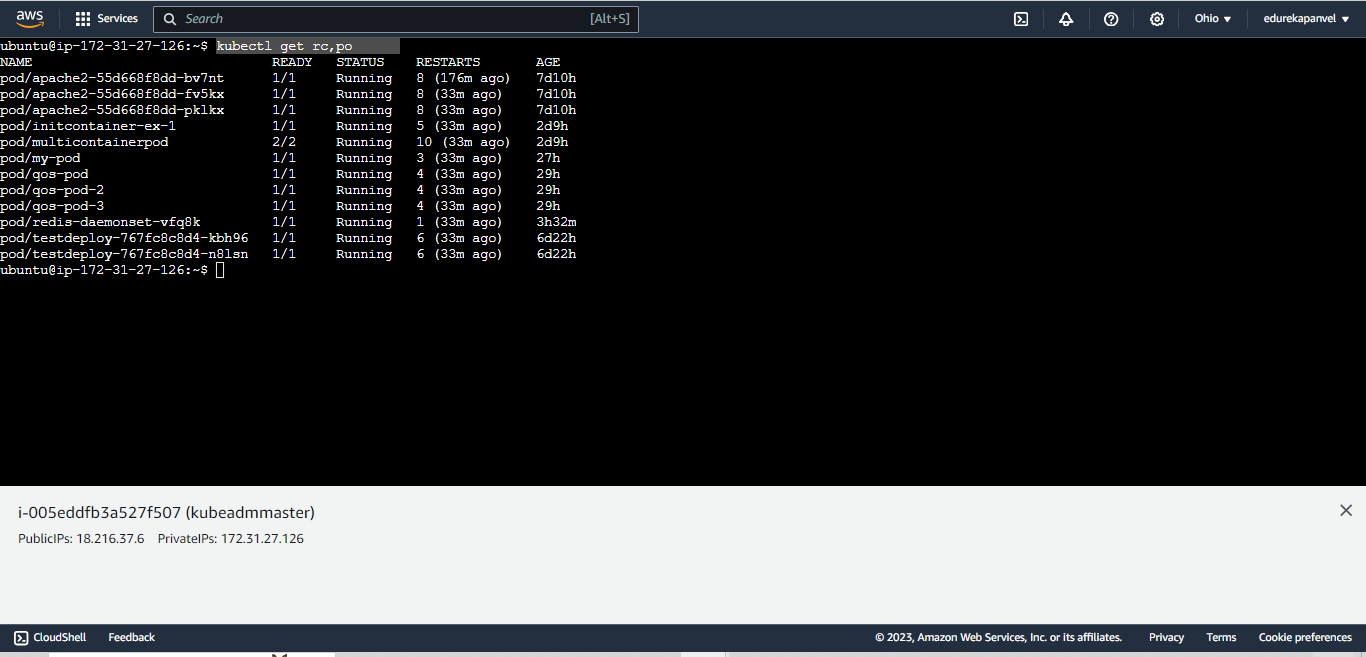




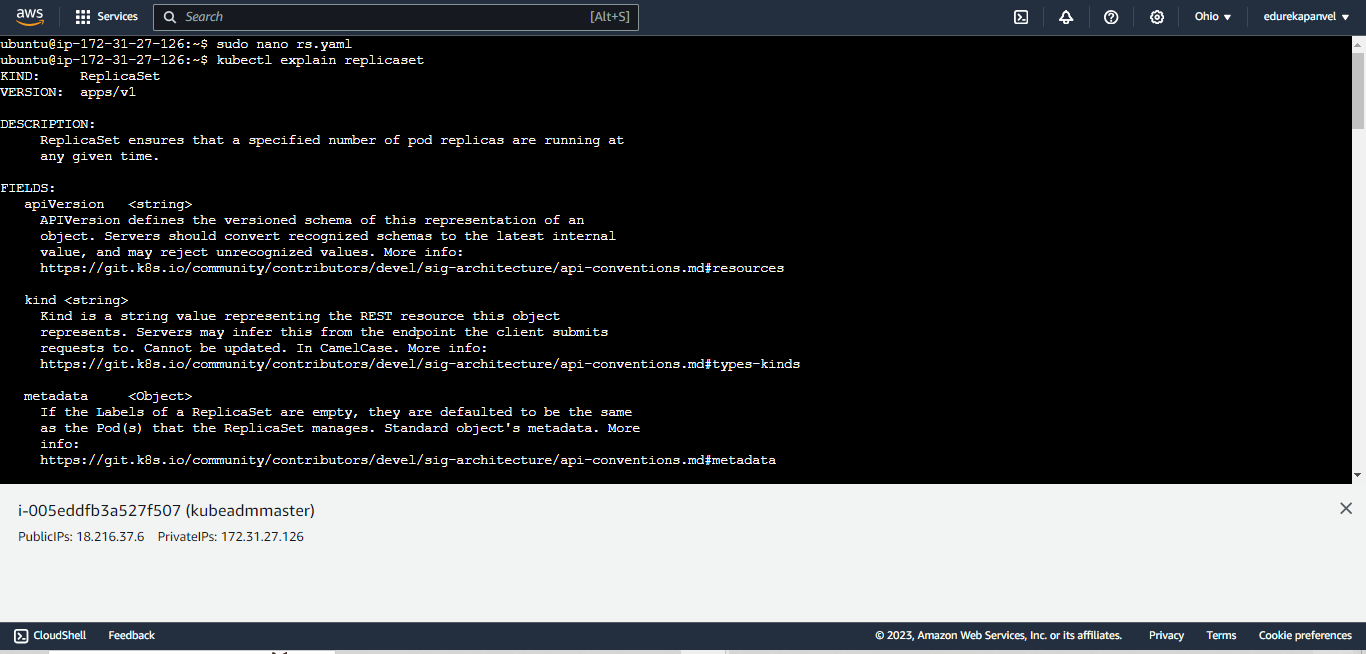
Pod is also deleted automatically

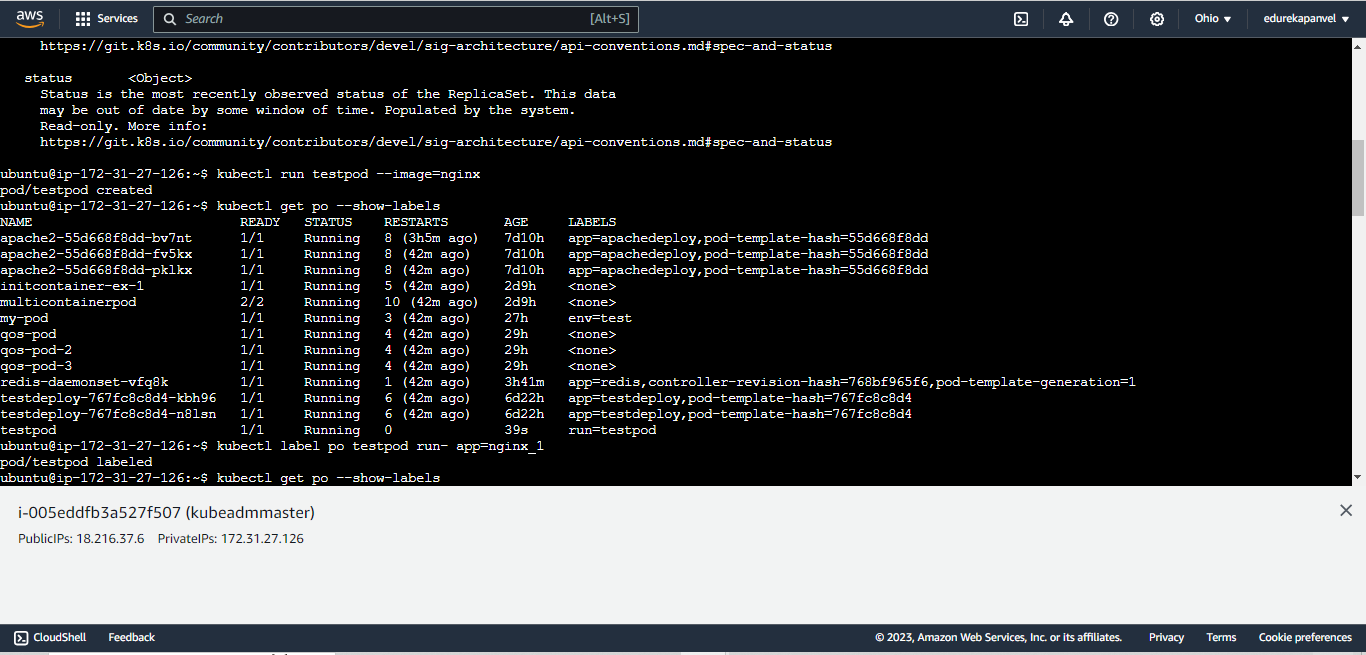


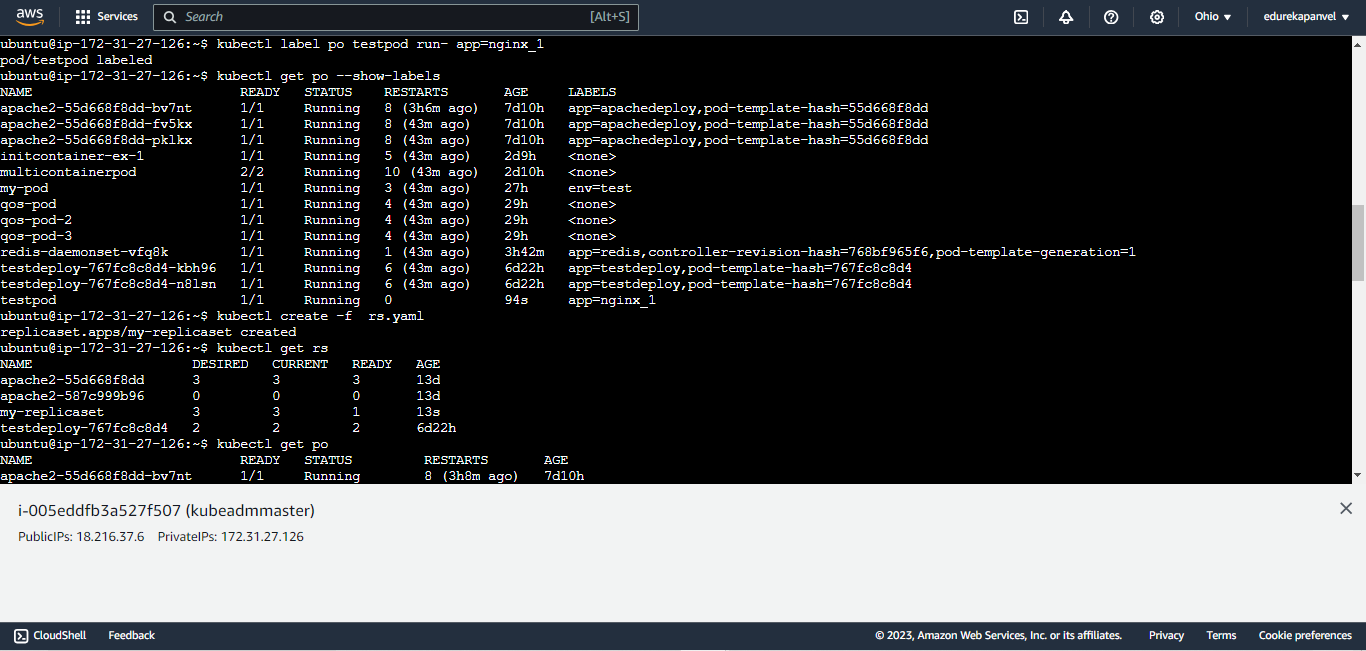
Wecan also run

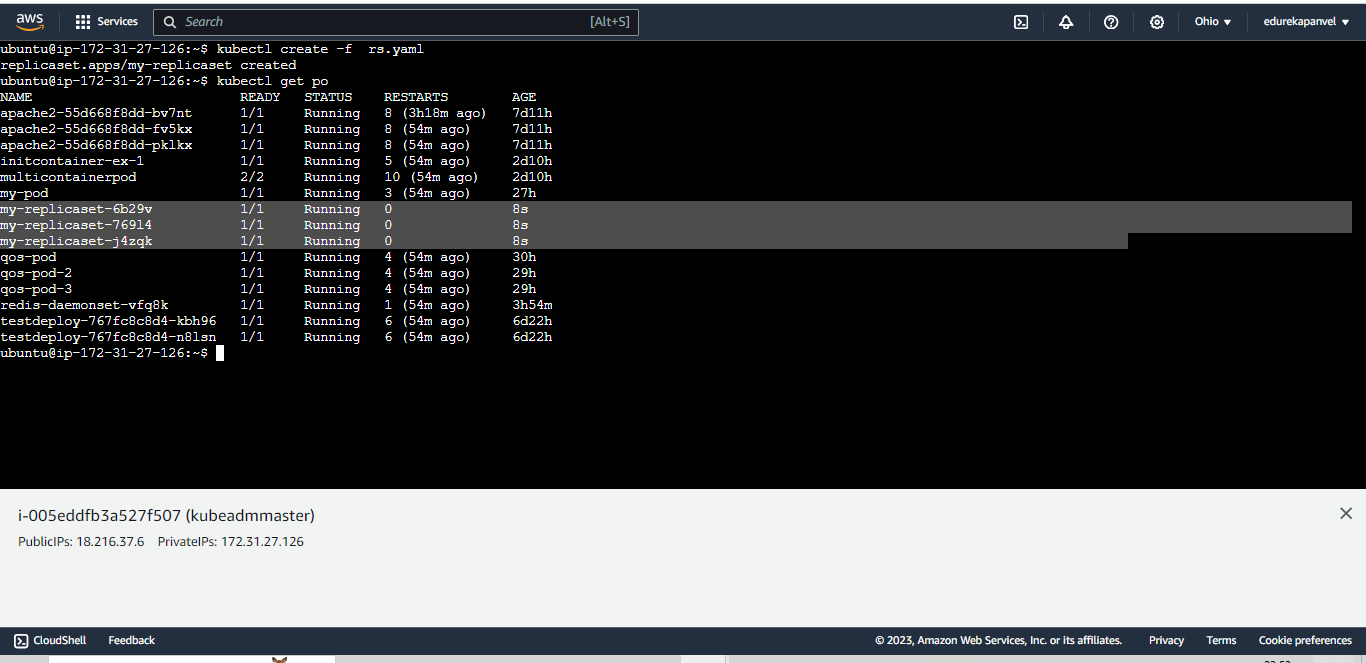


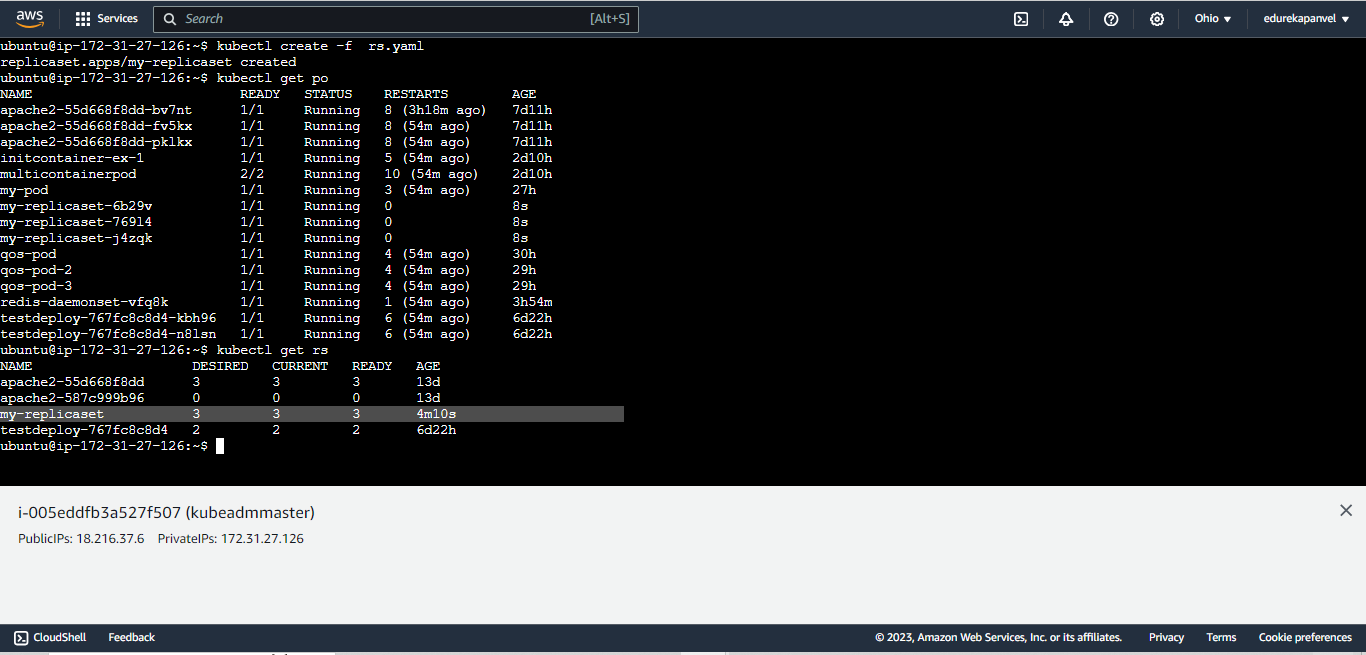
# command for replcation set



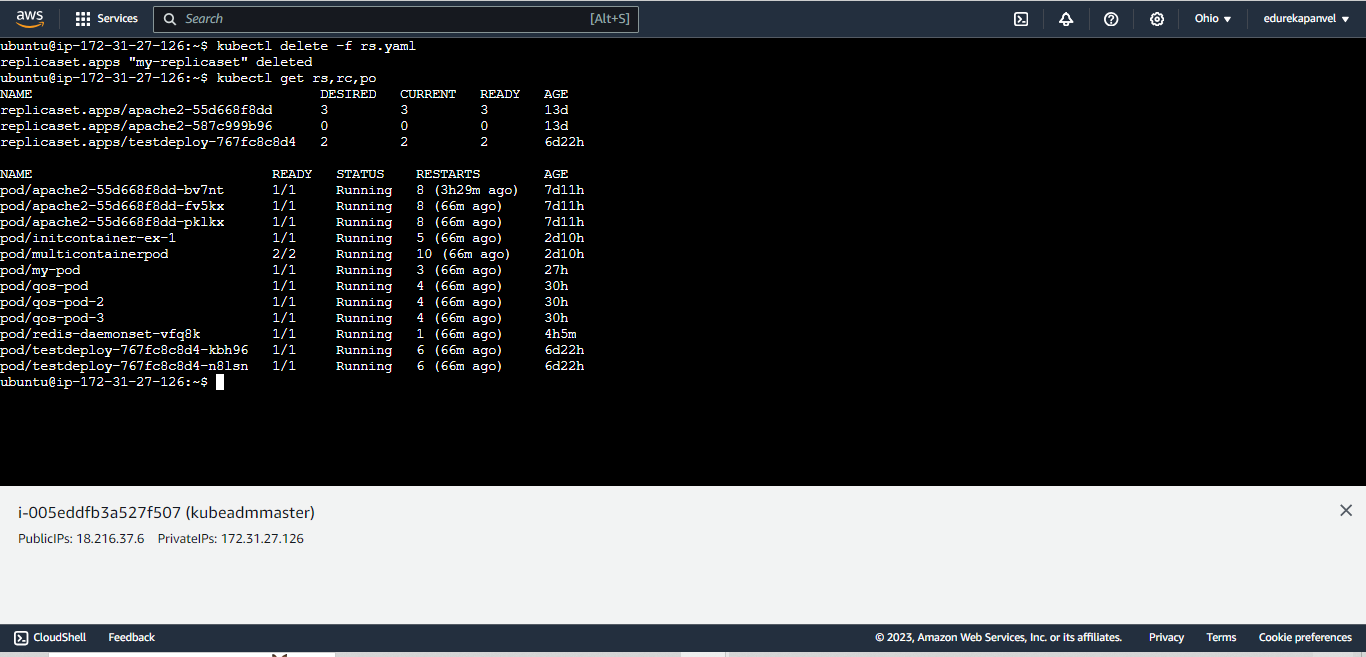




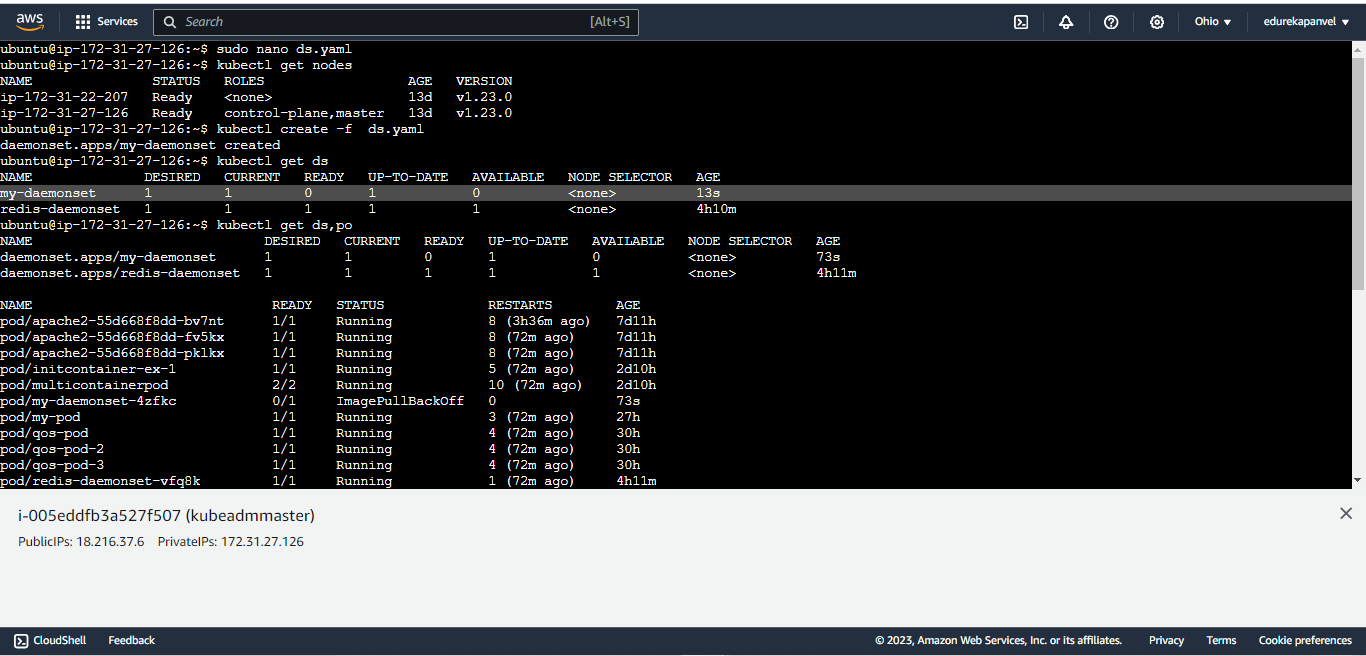


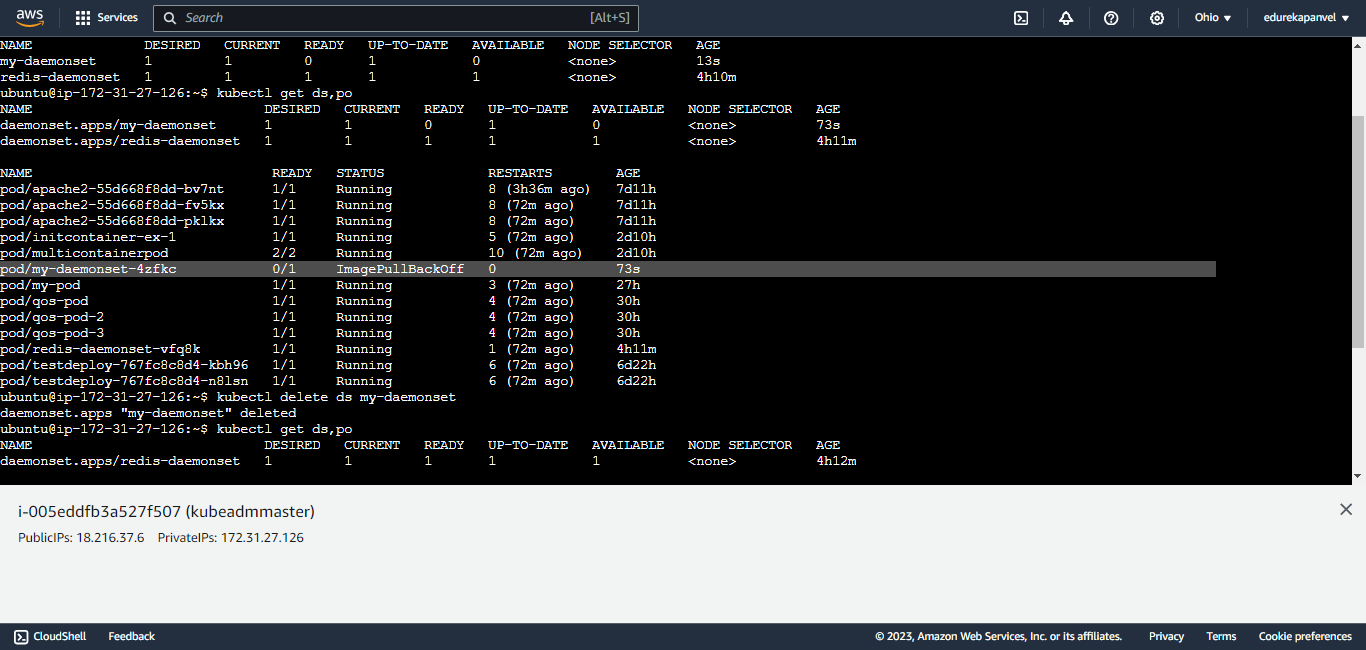


## delete all of this

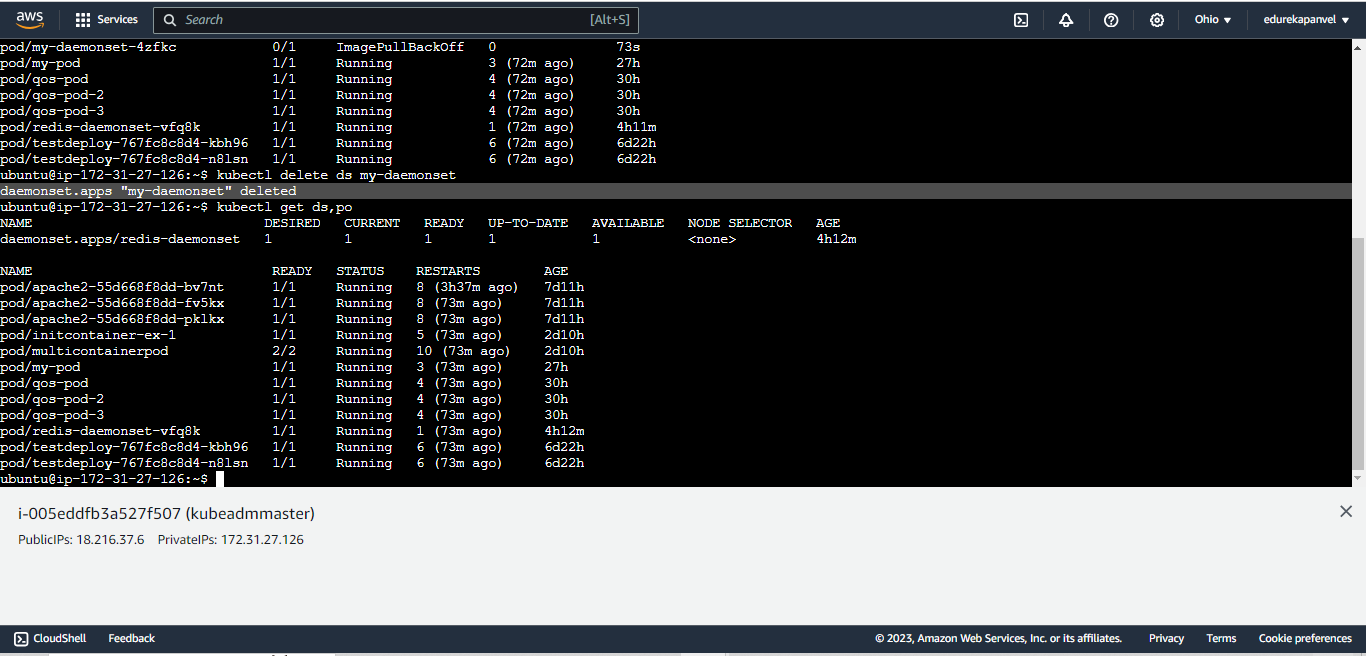


### create daemonset

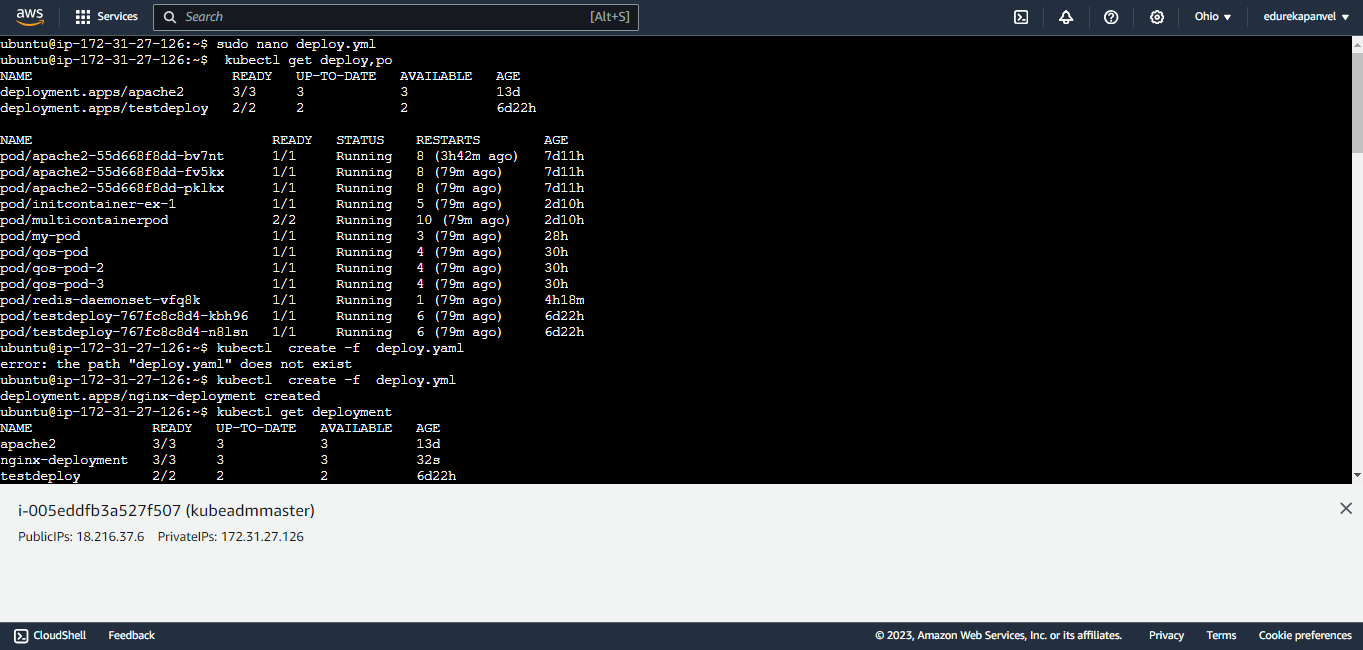


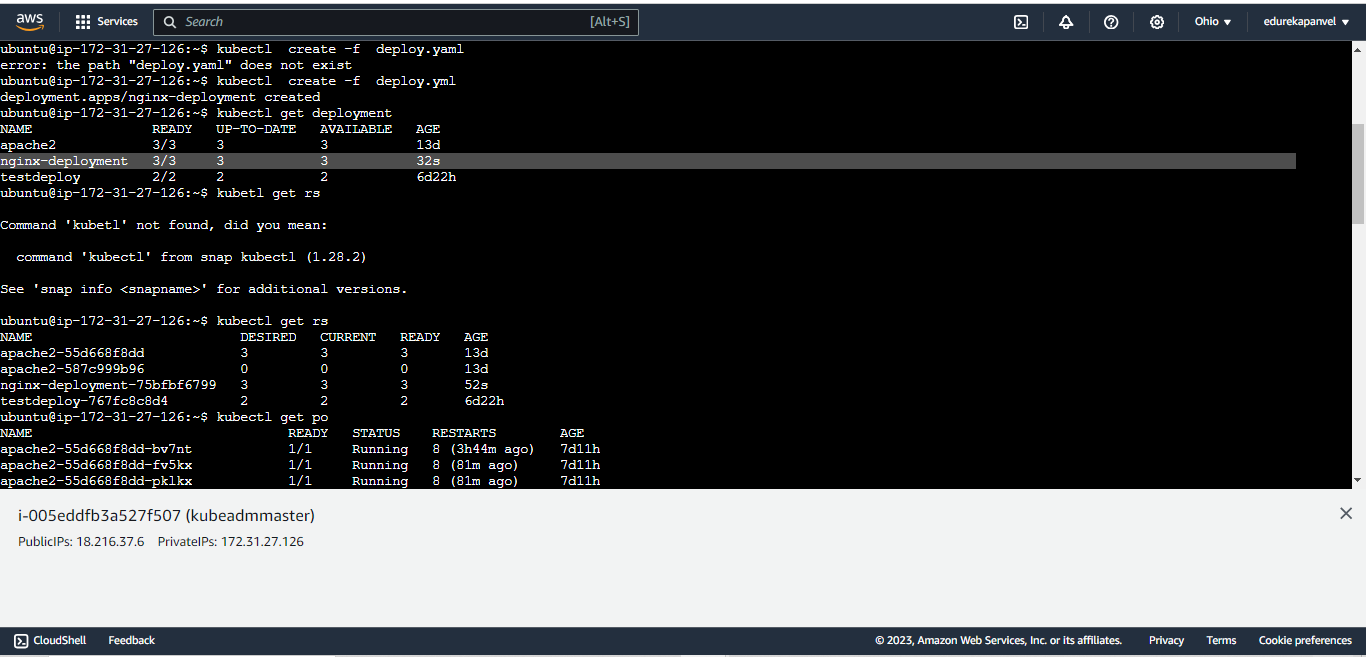


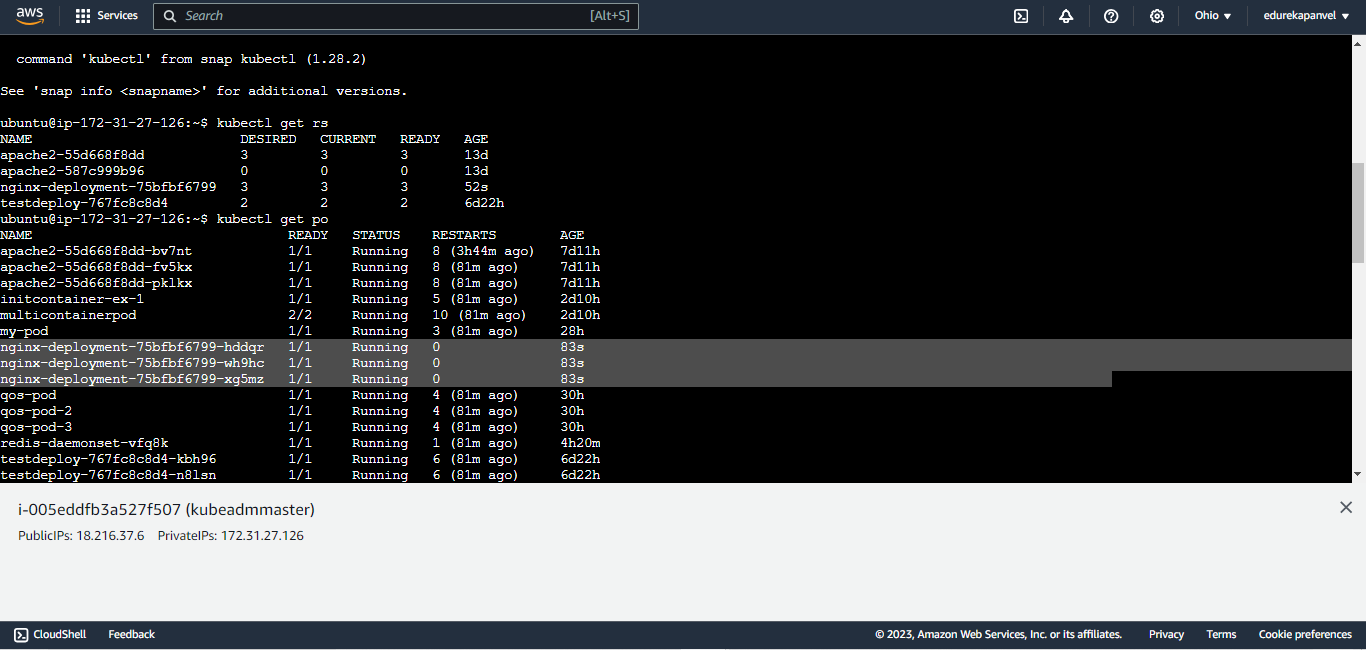
Deleted

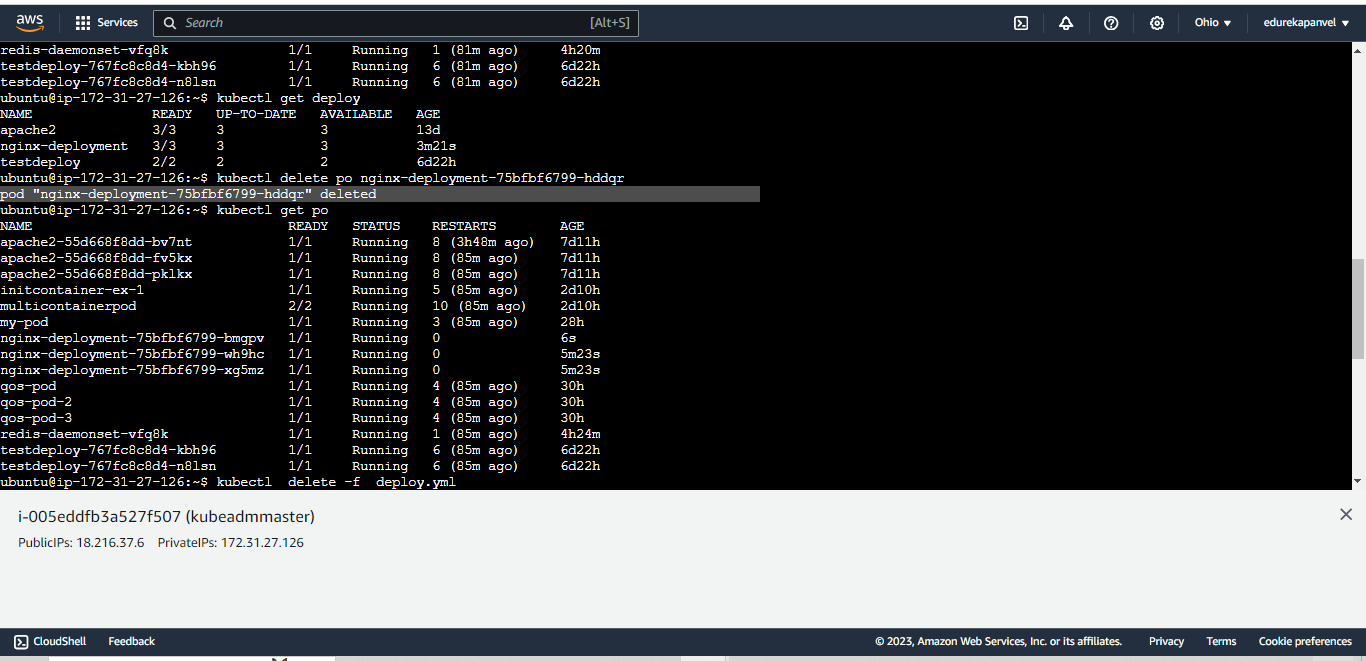


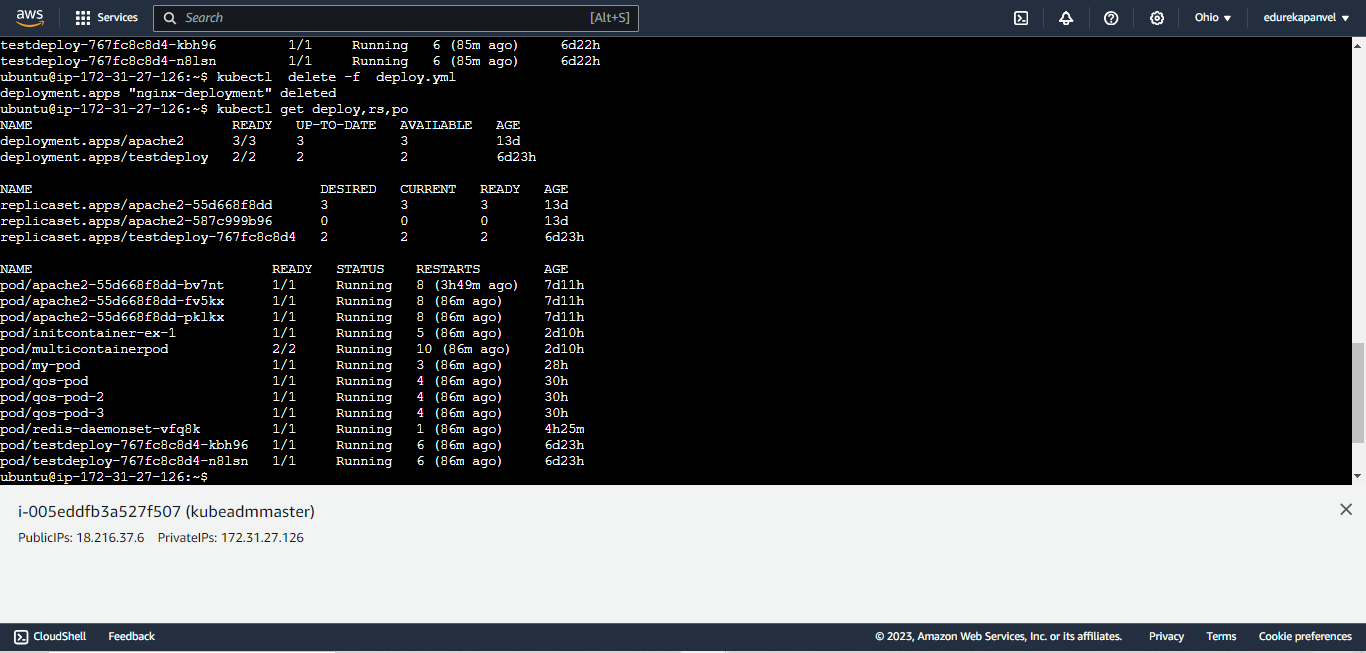
## deployment



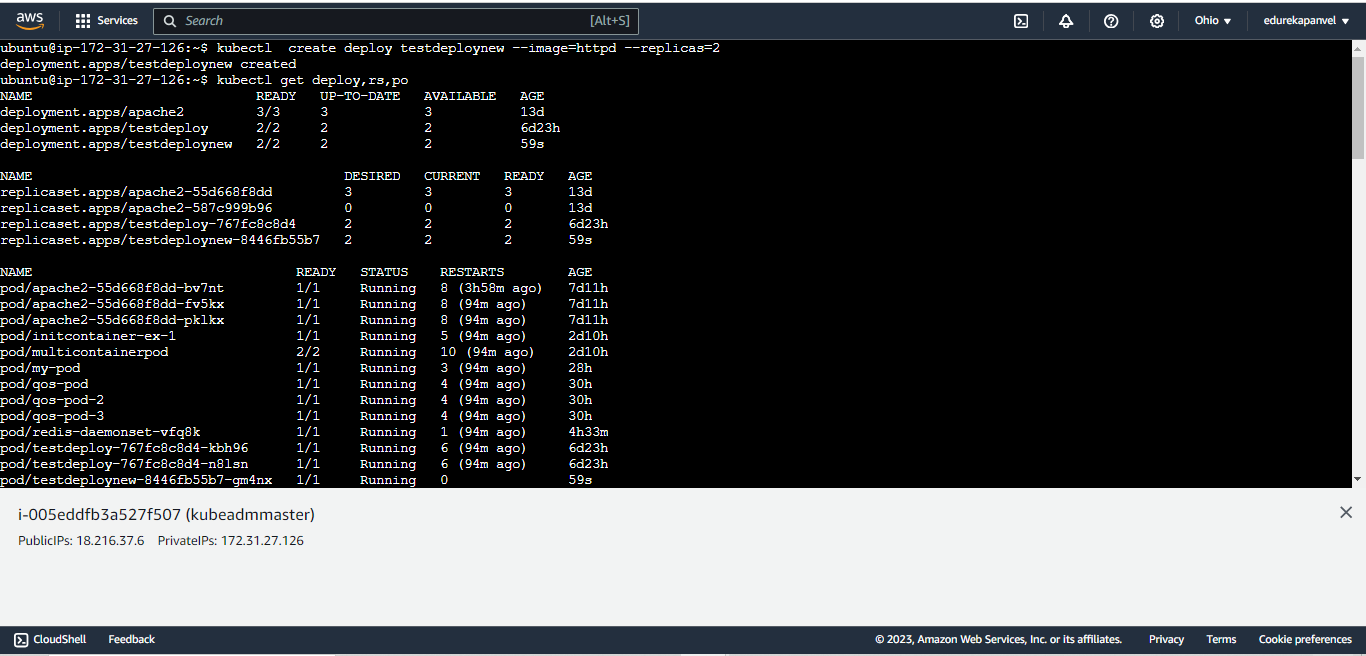


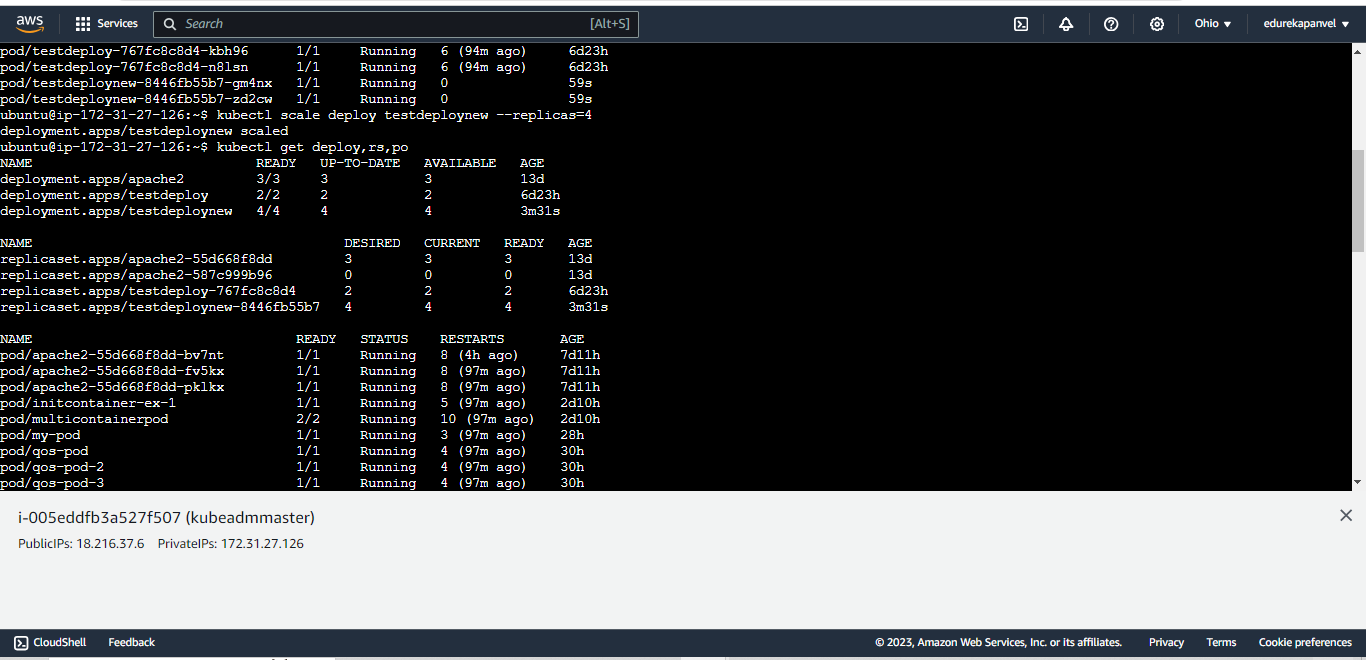


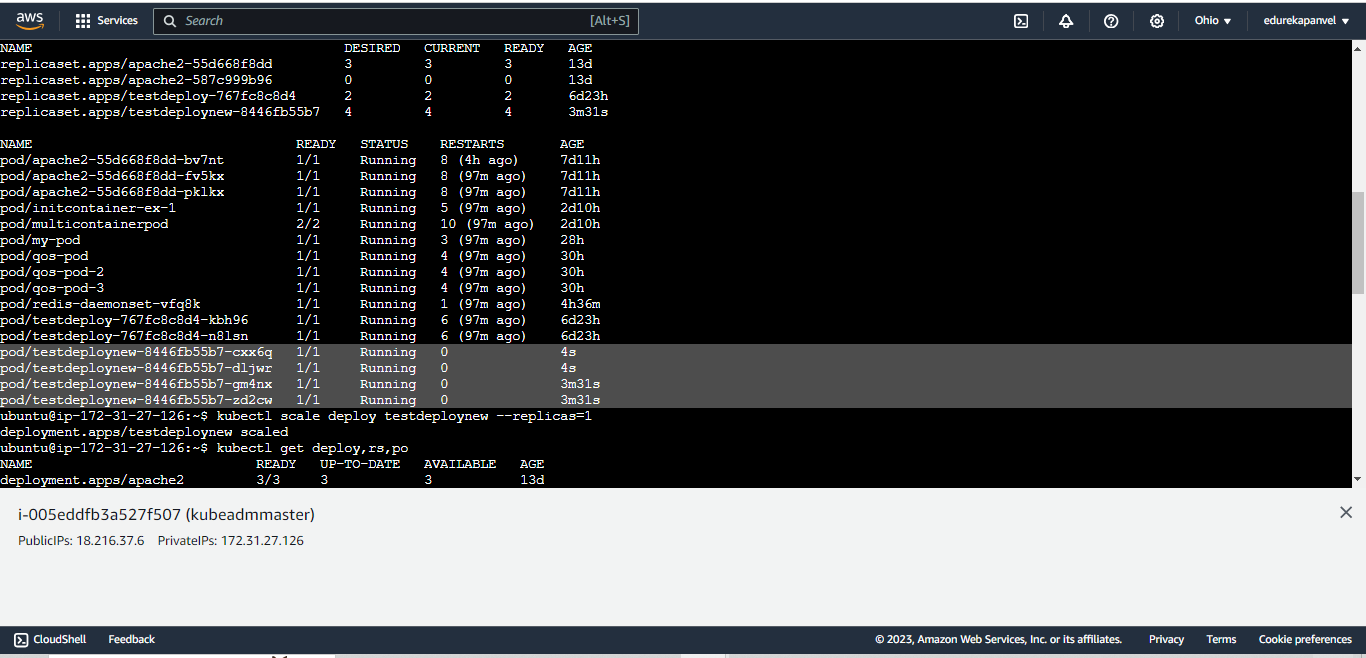


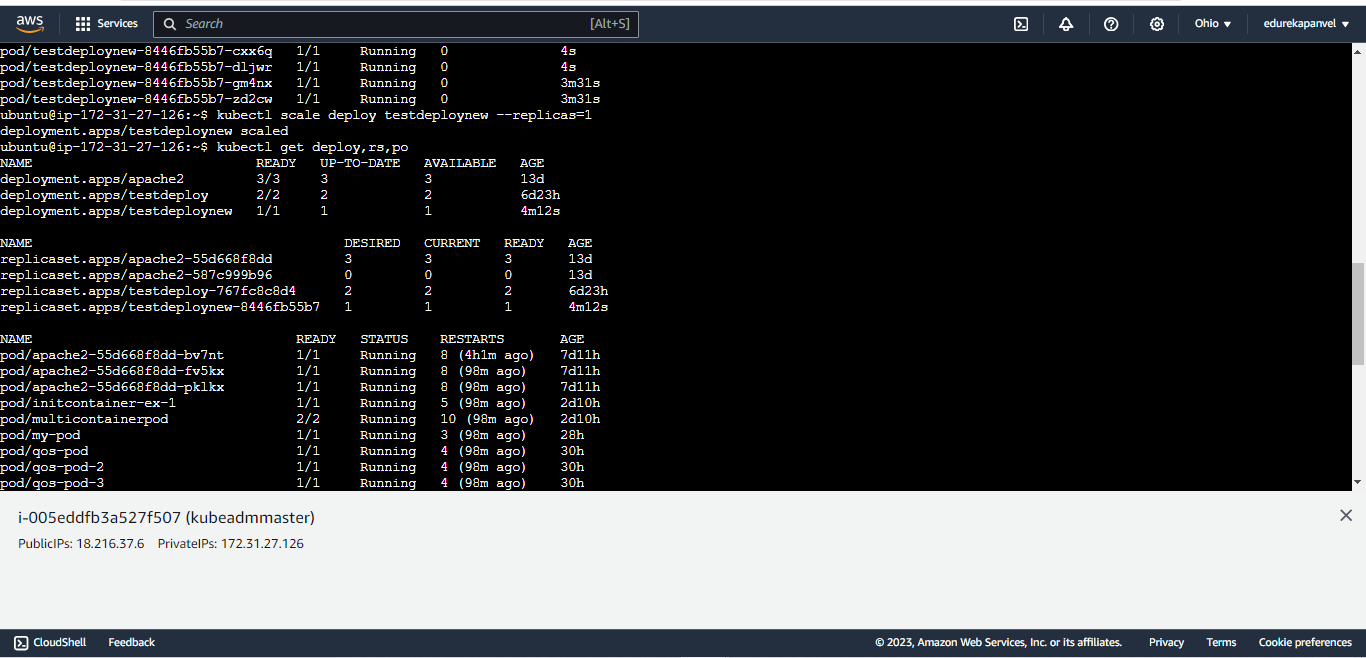


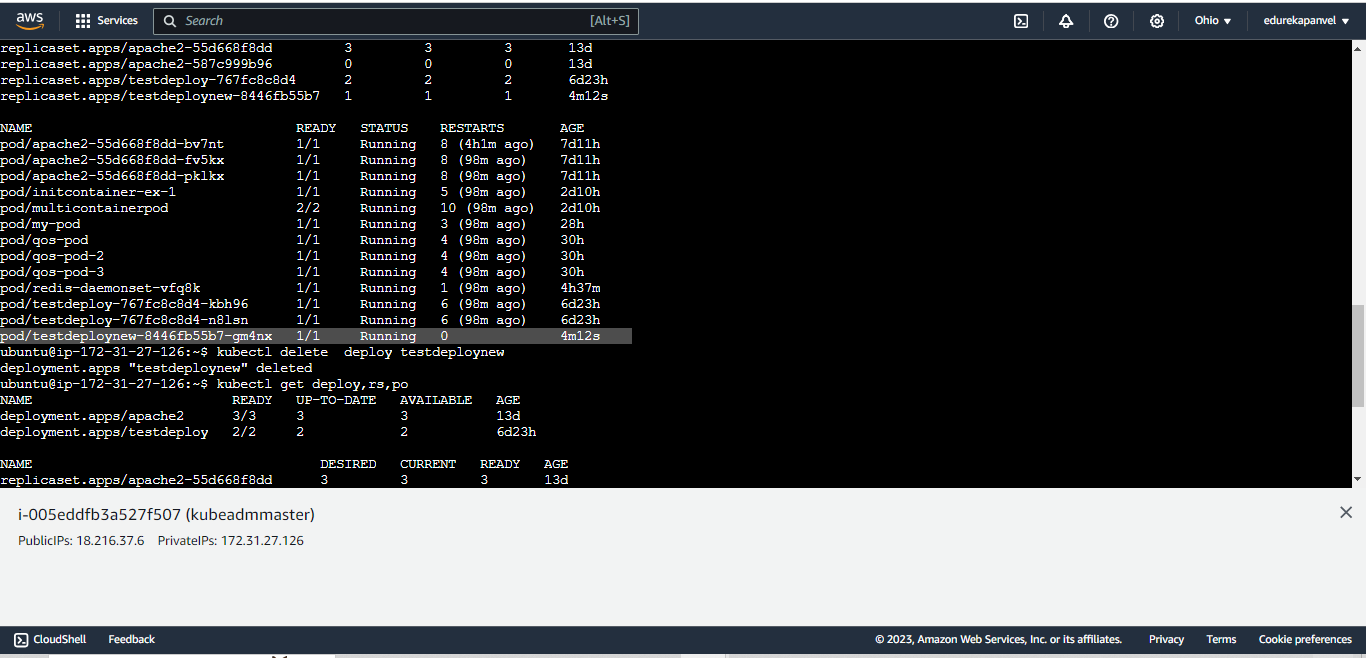
\*\*## now create resstdeploynew as pod scale up and scale down

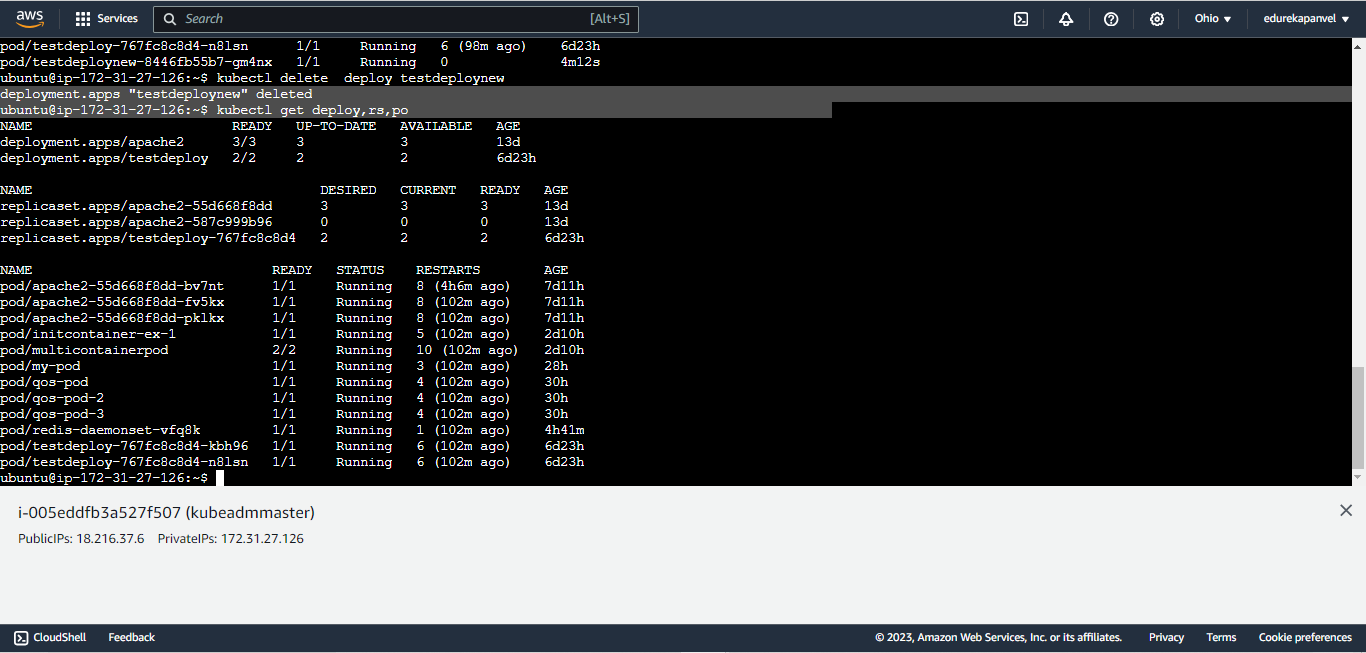




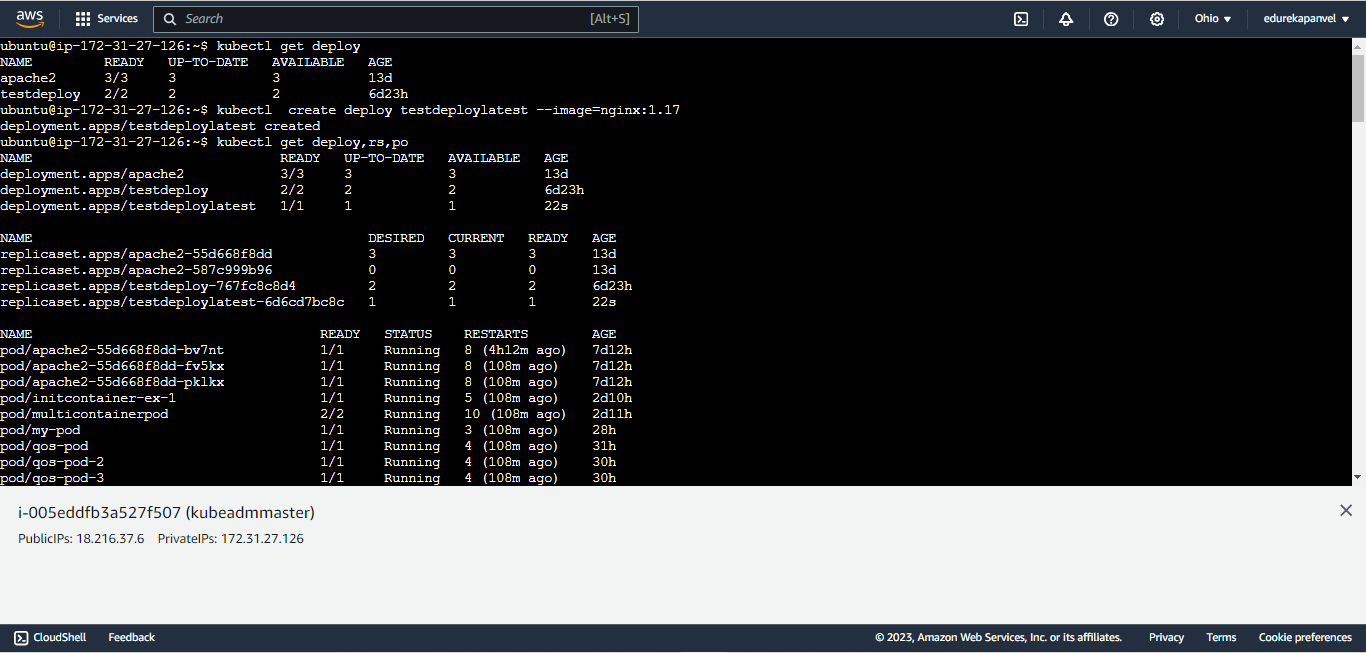


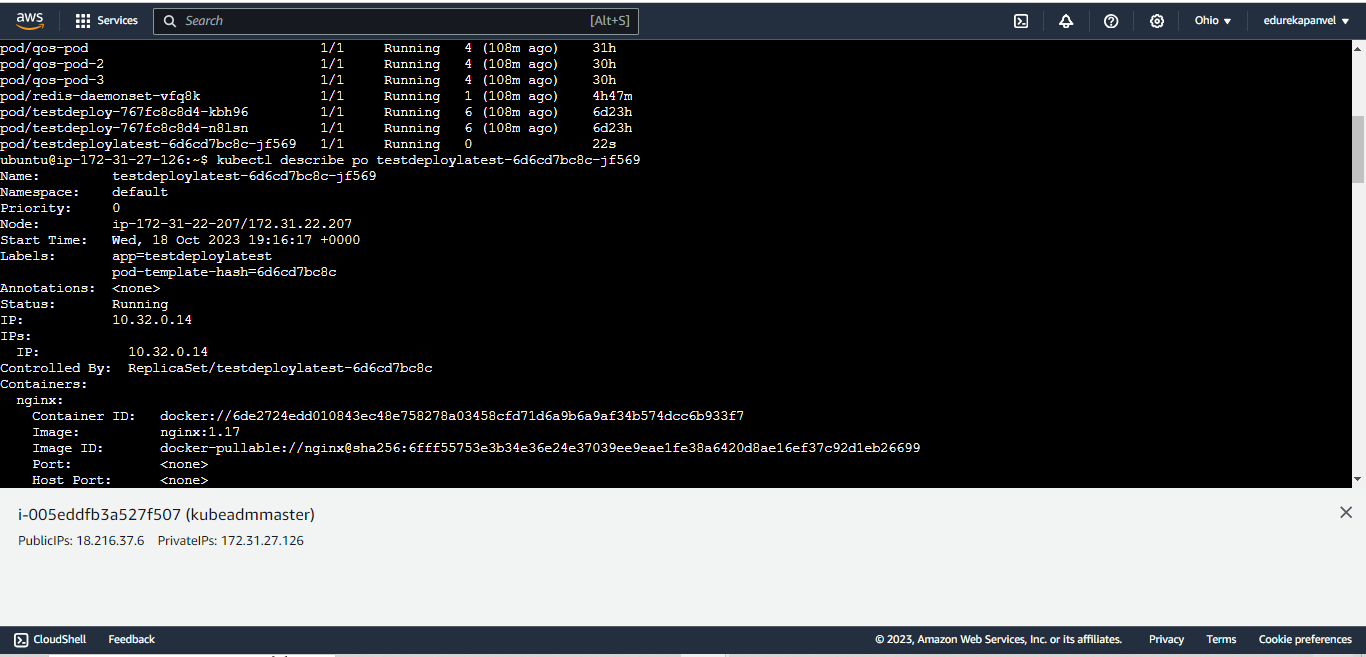


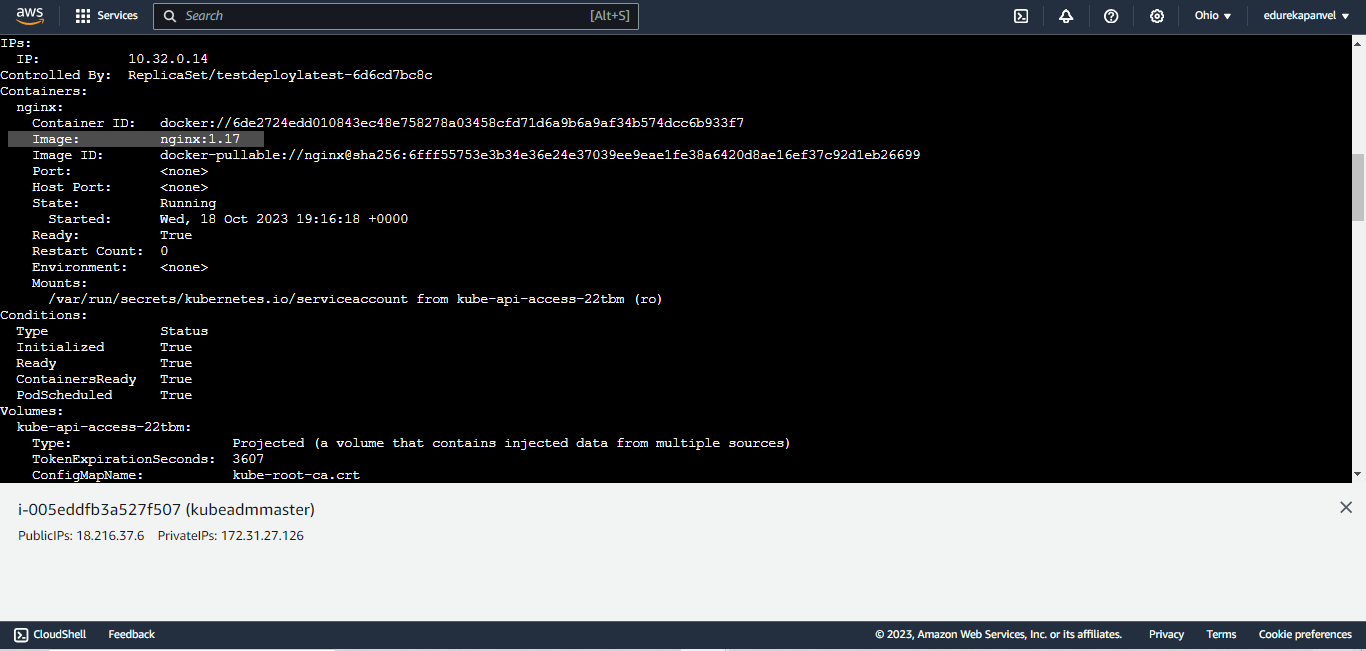




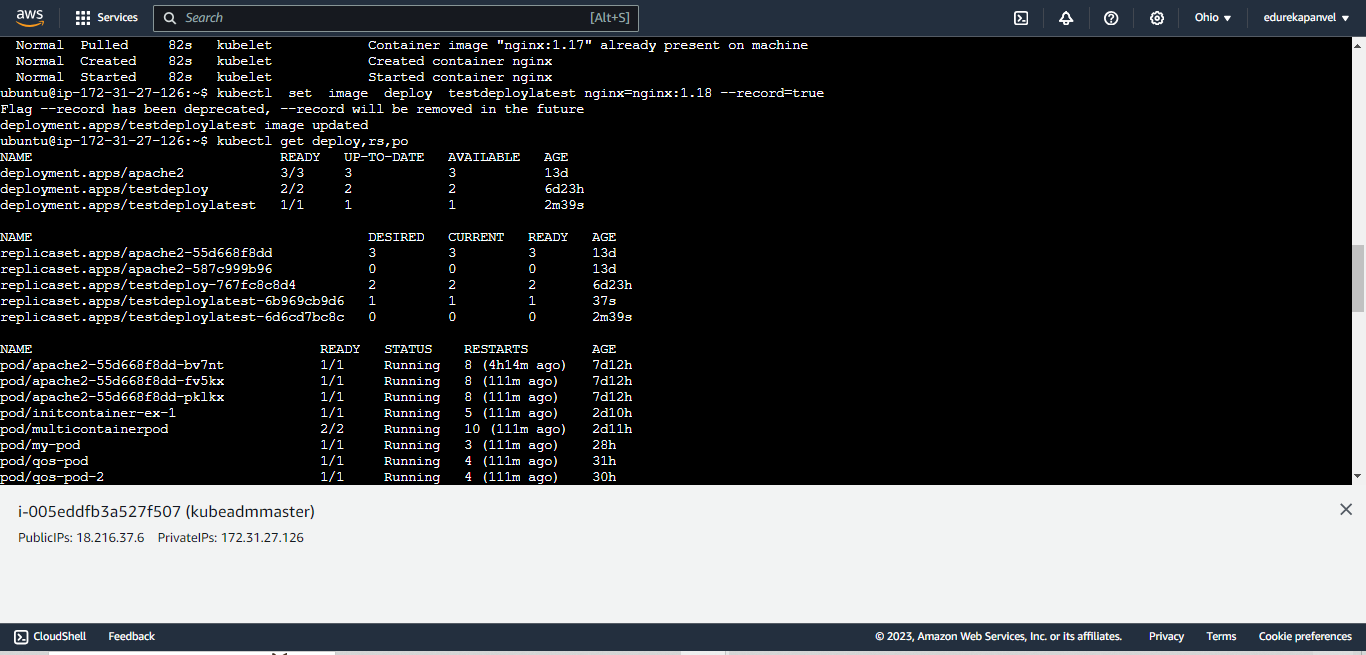
### rolling updates and rollback

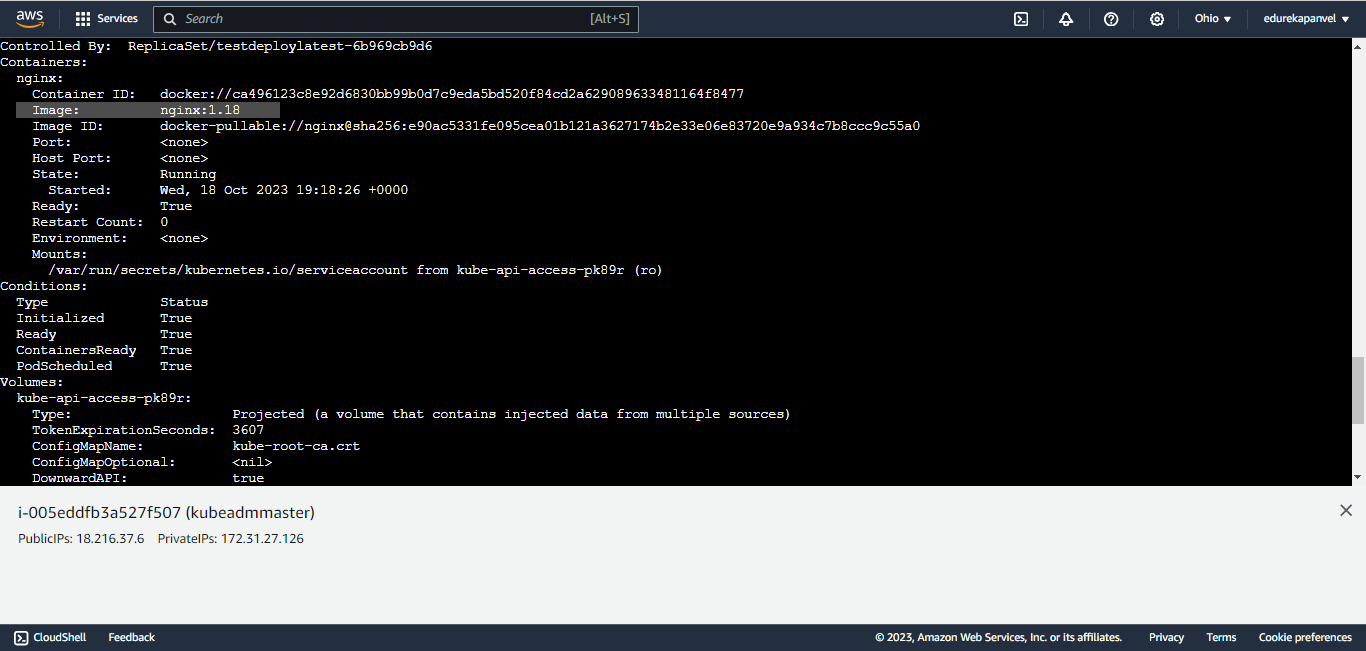


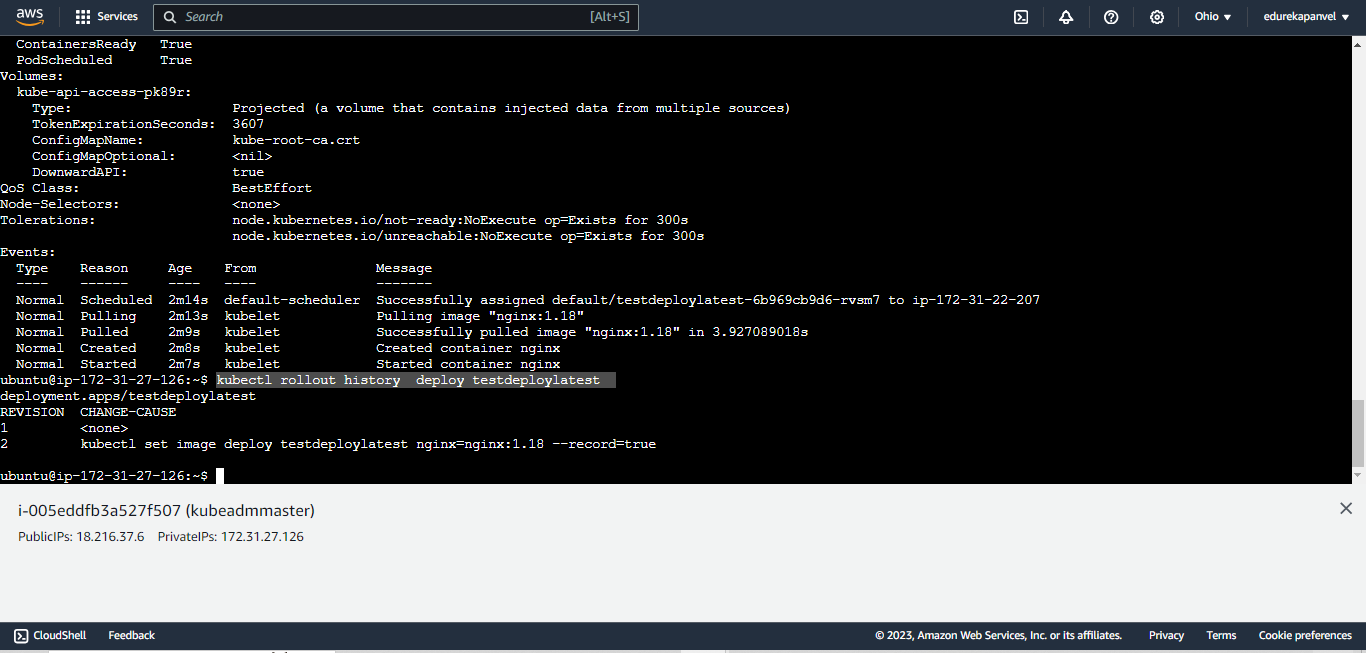




Update image to 1.18







Lu2

Kubectl gert nodes

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

spec:

selector:

matchLabels:

app: nginx\_deploy

replicas: 1 # Number of replica pods

template:

metadata:

labels:

app: nginx\_deploy

spec:

containers:

- name: nginx-container

image: nginx:1.17

ports:

- containerPort: 80 # Port your application listens on

resources:

requests:

memory: "30Mi"

cpu: "1m"

limits:

memory: "40Mi"

cpu: "1m"

Manual scalling

--- **watch kubectl get po**

**--- kubectl create –f deploy.yaml**

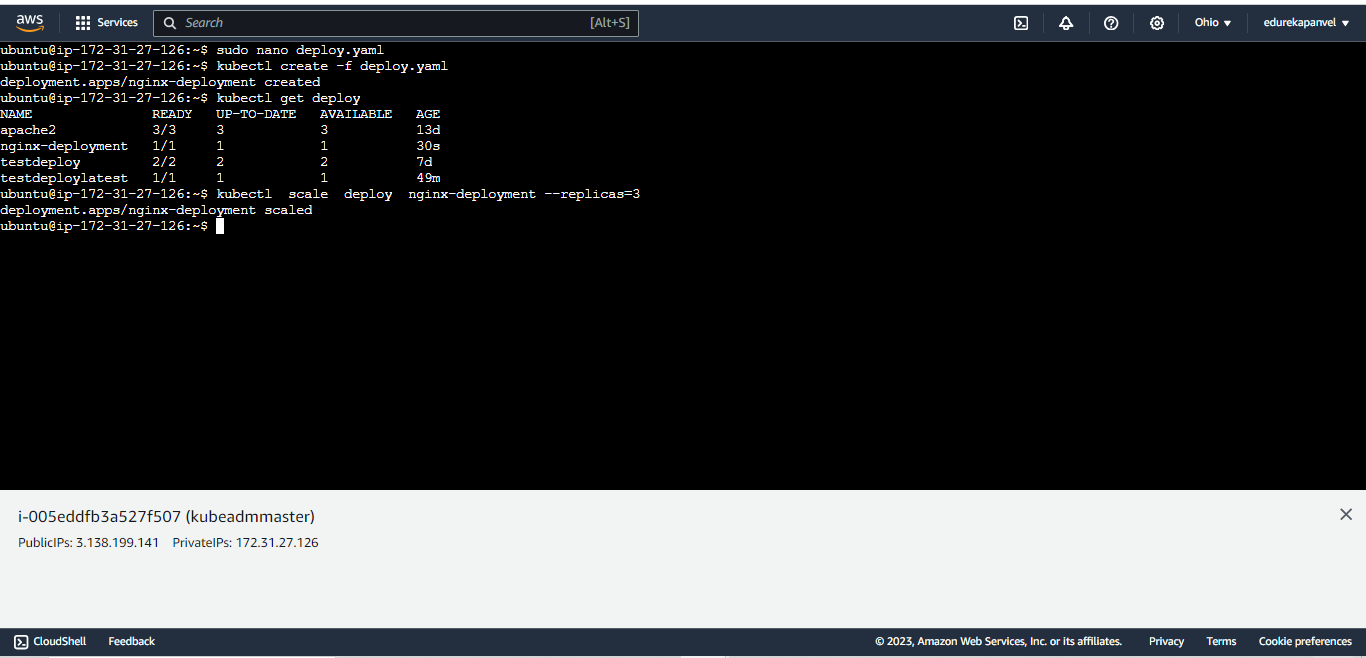
**--kubectl get deploy**

--- **watch kubectl get po**

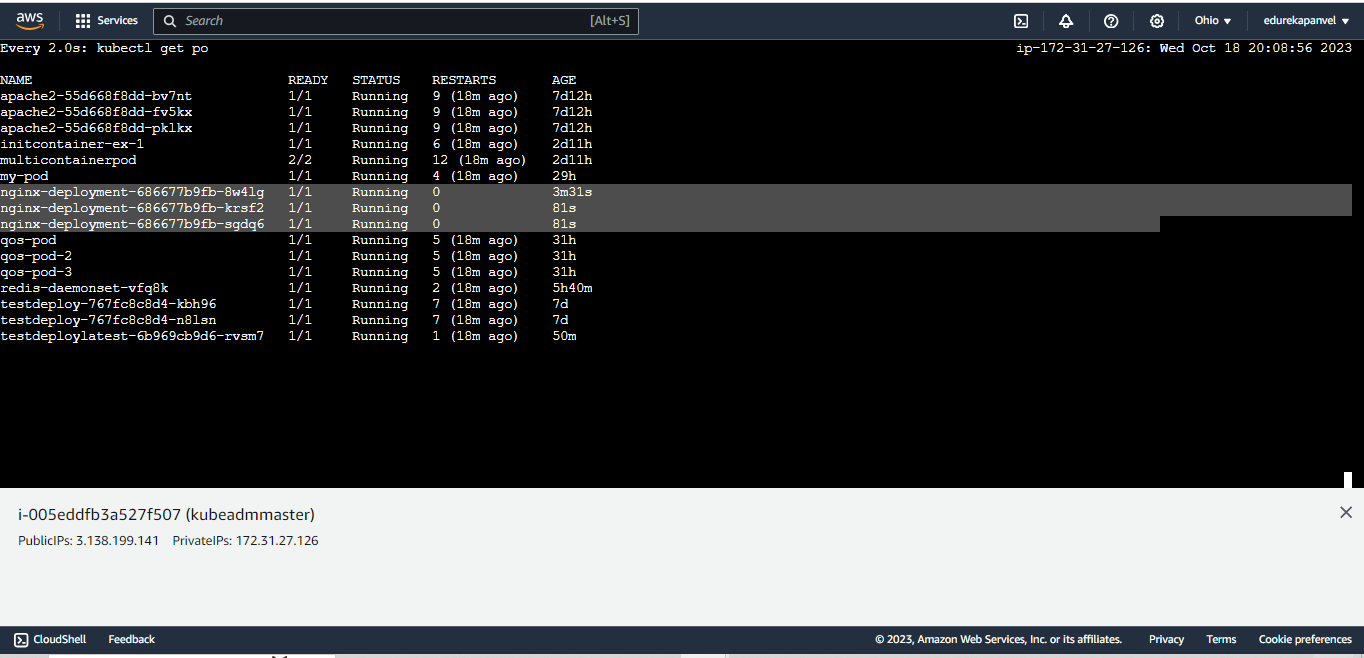
**(container taking time to creating )**

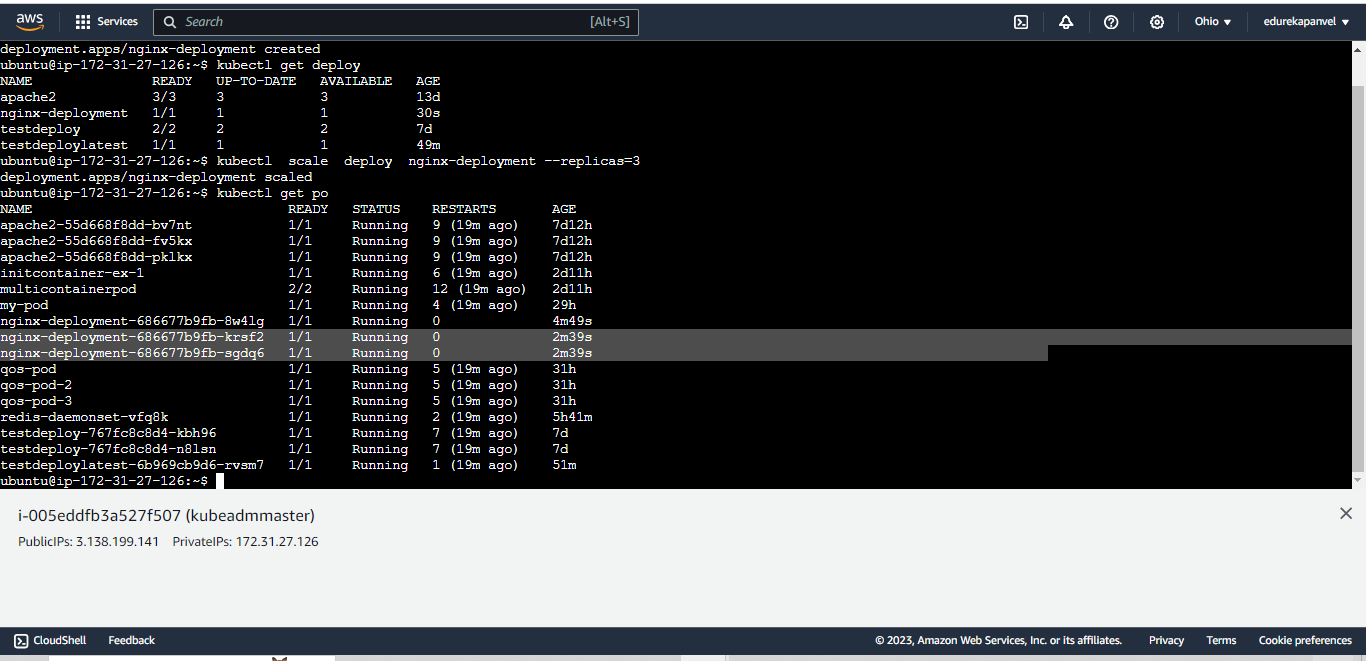
**--- kubectl scale deploy (deployment name) --replicas=3**

**kubectl scale deploy** nginx-deployment **--replicas=3**

****

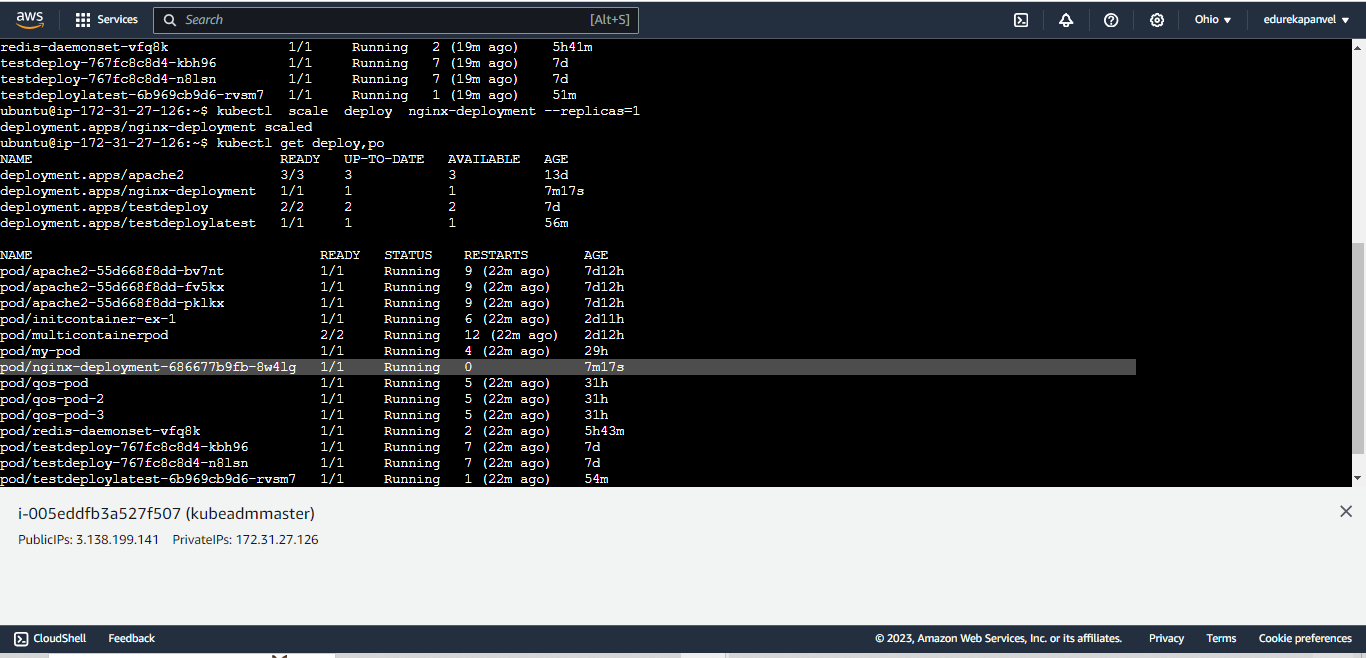
**When on watching mode**

****

****

* **kubectl get po**
* **we can see here two pods are created here**

**now downscale the application :**

****

**--- kubectl scale deploy (deployment name) --replicas=1**

**--kubectl get deploy**

**We can see here application pod are again go to 1 ( agfain left with 1 pods)**

**kubectl get deploy,po**

**Now create HPA**

**Sudo nano hpa.yaml**

**Kubectl explainhpa hpa ( check version )**

**apiVersion: autoscaling/v1**

**kind: HorizontalPodAutoscaler**

**metadata:**

**name: nginx-deployment-hpa**

**spec:**

**minReplicas: 1 # Minimum number of pods**

**maxReplicas: 3 # Maximum number of pods**

**scaleTargetRef:**

**apiVersion: apps/v1**

**kind: Deployment**

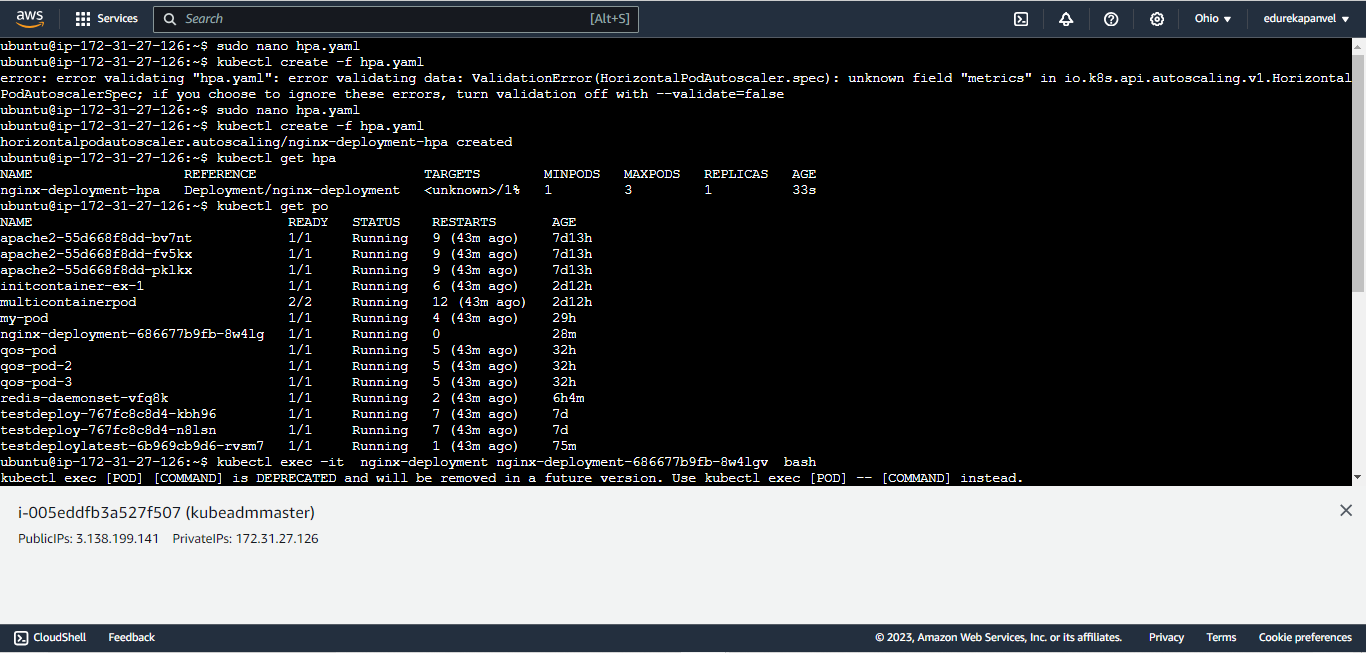
**name: nginx-deployment**

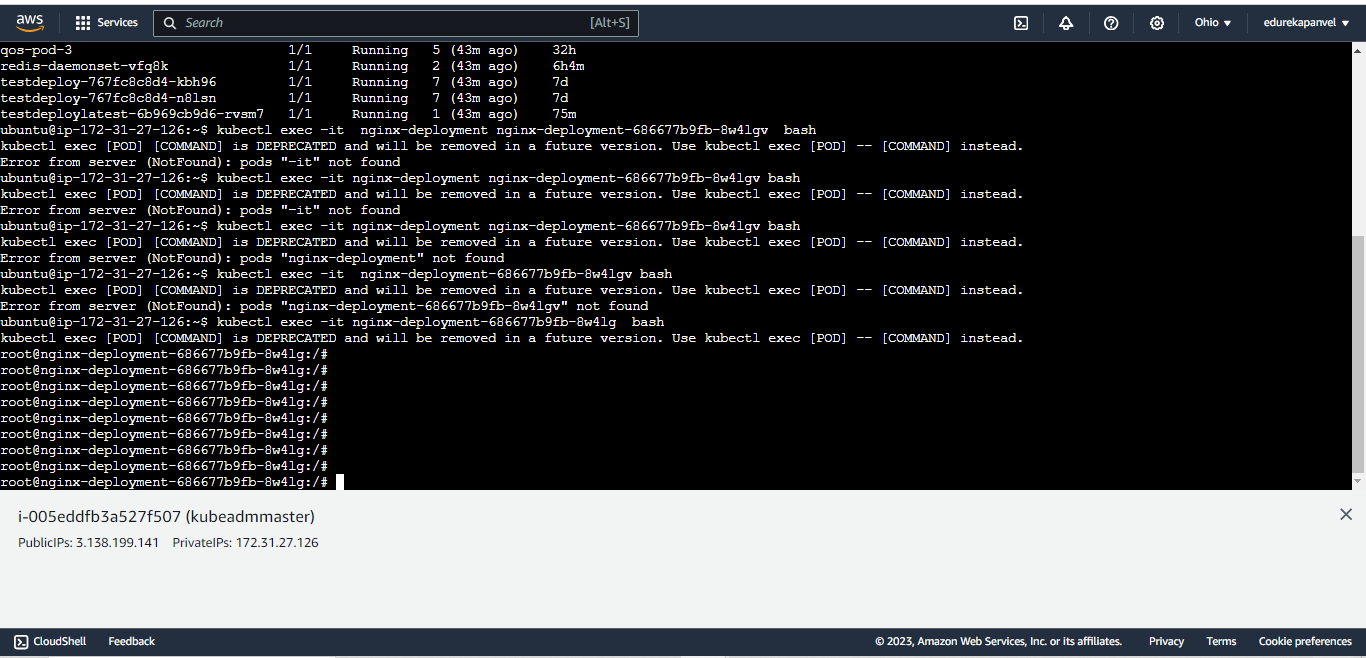
**targetCPUUtilizationPercentage: 1**

**kubectl create –f hpa.yaml**

**kubectl get hpa**

**kubectl get po**

****

****

**#!/bin/bash**

**while :**

**do**

**echo $HOME**

**echo $SHELL**

**mkdir /testing**

**touch /testing/file{1,2,3,4,5,6,7,8,9}**

**ls -al /testing**

**rm -rf /testing**

**done**

**kubectl exec –it nginx-deployment (cxontainer id )bash**

**now copy this file sript.sh and paste into container**

**now check cat infinite.sh ( just check file is creted or not )**

**now give the permission to the file**

**chmod +X infinite.sh**

**ls –al ( just for check is it executable or not )  
./infinite.sh ( run the script)**

**we can see it continuously going to run**

**now check is automatically is creted**

**this is how autoscalling is happen**