**Horizontal Pod Autoscaler in Kubernetes**

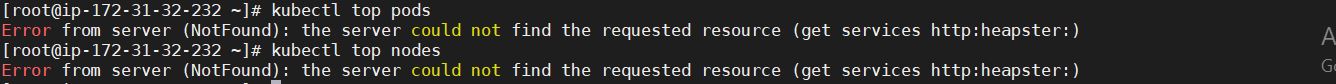
Pre-Requisites:

* Install EKS Cluster
* Install git

Check whether metrics installed or not in our Instance by using below commands

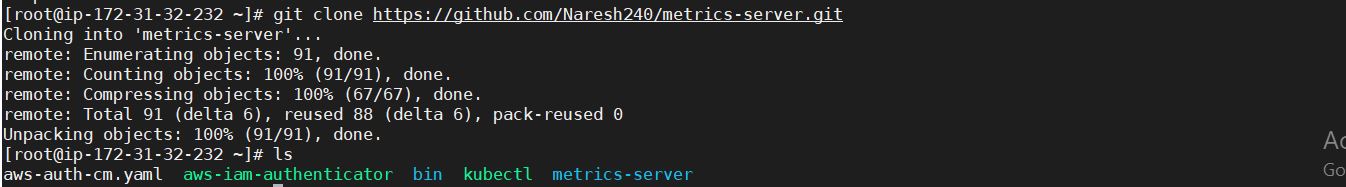
kubectl top pods

kubectl top nodes



Clone Metric-Server repository from github

git clone <https://github.com/Naresh240/metrics-server.git>



Open metrics-server-deployment.yaml file and add below data with in the Spec of Container

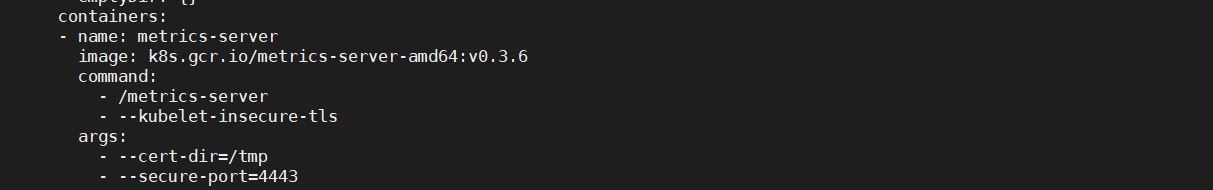
cd /root/metrics-server/deploy/Kubernetes

vi metrics-server-deployment.yaml

command:

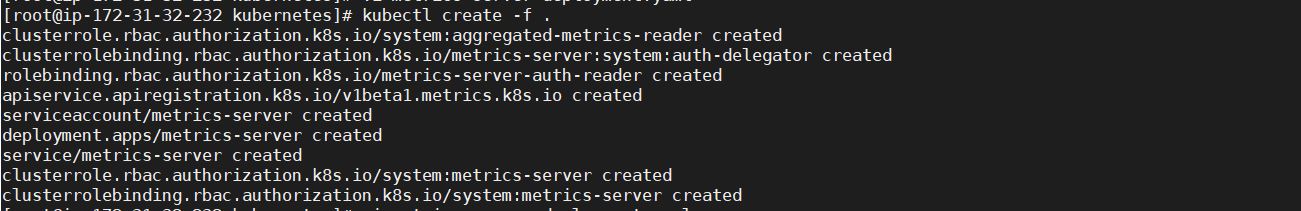
- /metrics-server

- --kubelet-insecure-tls



Now deploy metrics-server using below command:

kubectl create -f .



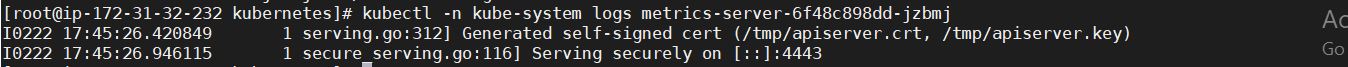
Check status of the metrics-server:

kubectl -n kube-system get pods



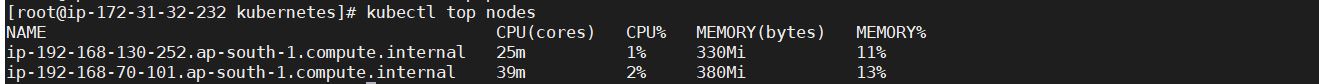
Check logs for metrics-server:

kubectl -n kube-system logs metrics-server-6f48c898dd-jzbmj



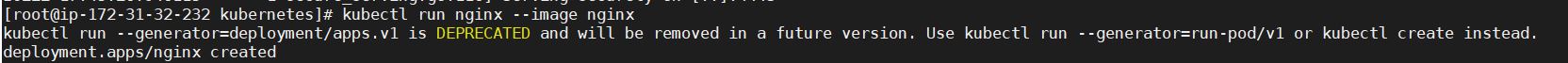
Now check whether metrics installed or not:

kubectl top nodes



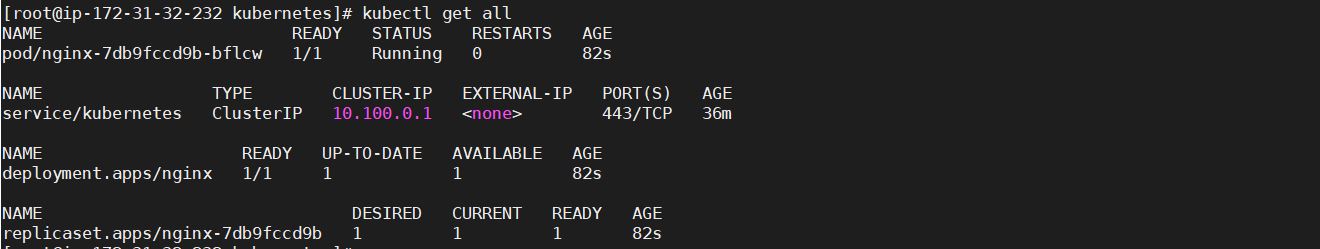
Now deploy nginx:

kubectl run nginx --image nginx



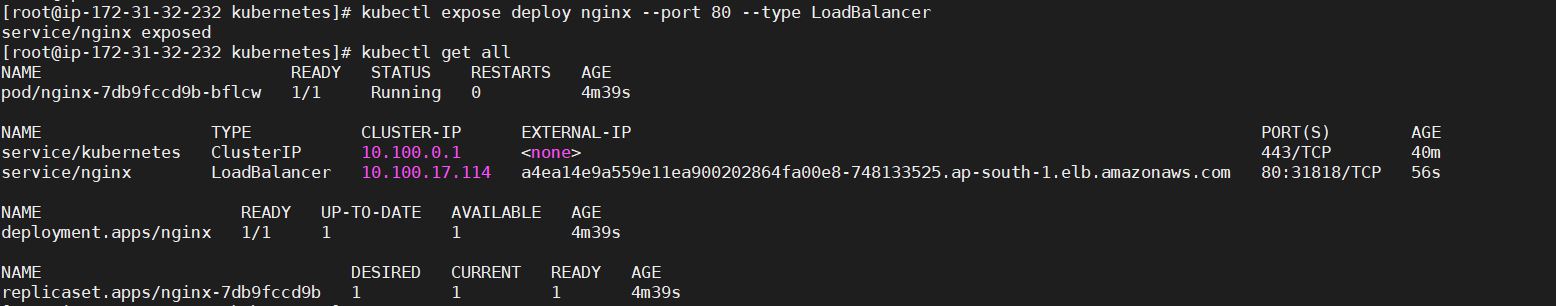
To check what are all running:

Kubectl get all

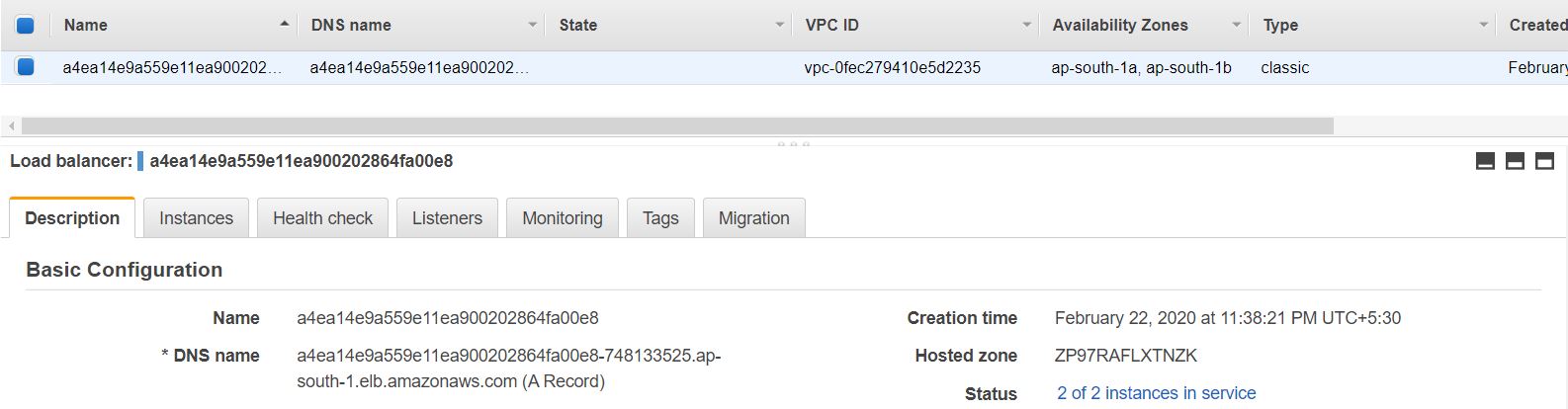


Now expose nginx:

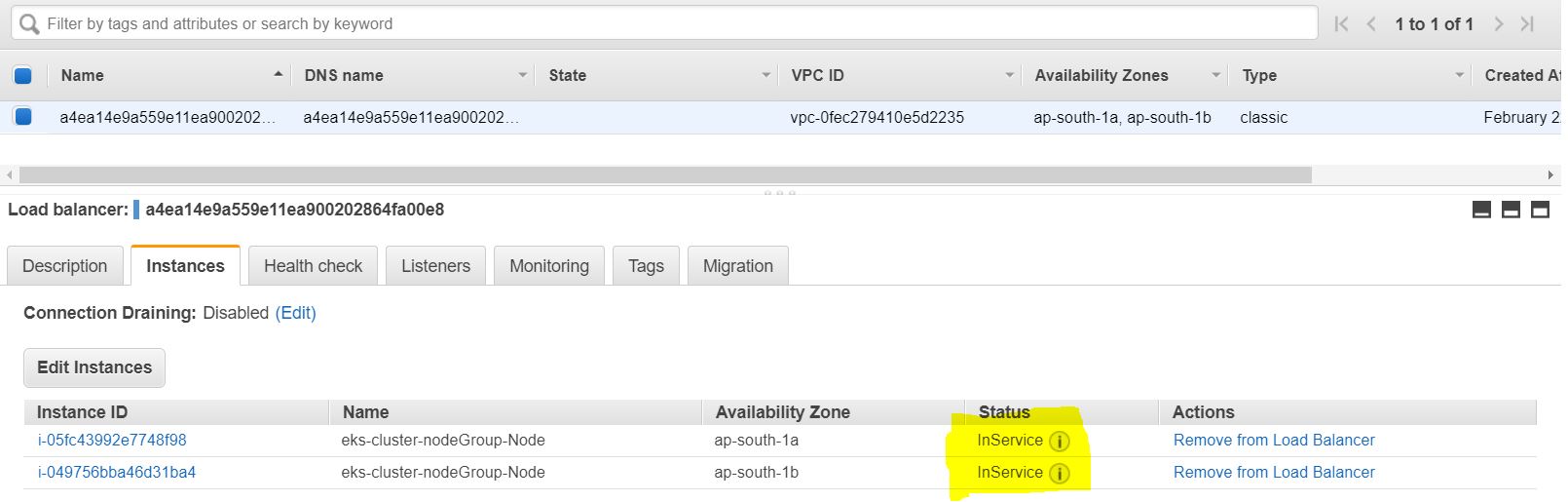
kubectl expose deploy nginx --port 80 --type LoadBalancer



Now check in AWS Console, check whether LoadBalancer created or not:

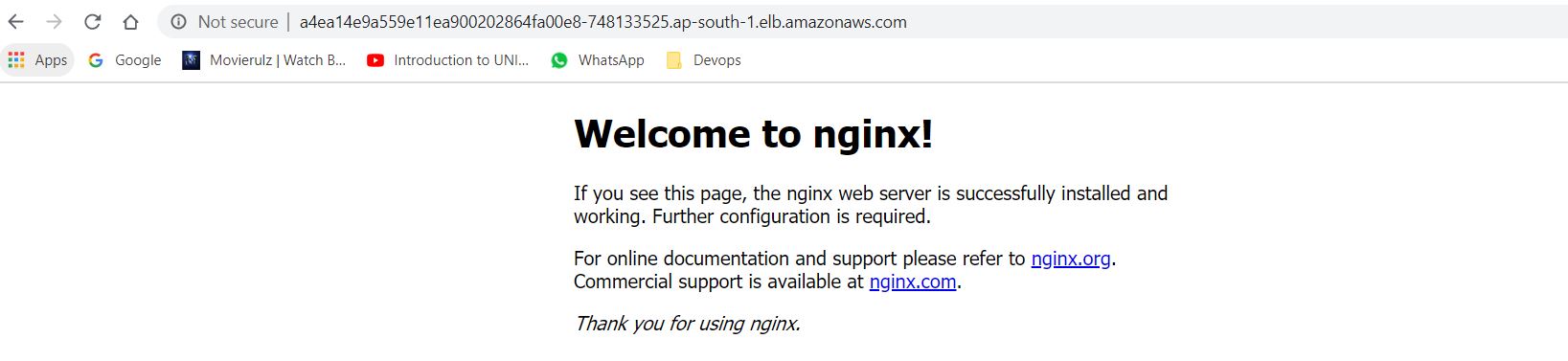


Check whether nodes comes into service or not:



Copy DNS name of LoadBalancer and check in UI:

<http://a4ea14e9a559e11ea900202864fa00e8-748133525.ap-south-1.elb.amazonaws.com/>



**Horizontal Auto Scaling for nginx:**

Create 10-hpa.yaml file with in Kubernetes directory

mkdir /root/kubernetes

vi 10-hpa.yaml

apiVersion: autoscaling/v1

kind: HorizontalPodAutoscaler

metadata:

name: nginx

spec:

maxReplicas: 5

minReplicas: 1

scaleTargetRef:

apiVersion: extensions/v1beta1

kind: Deployment

name: nginx

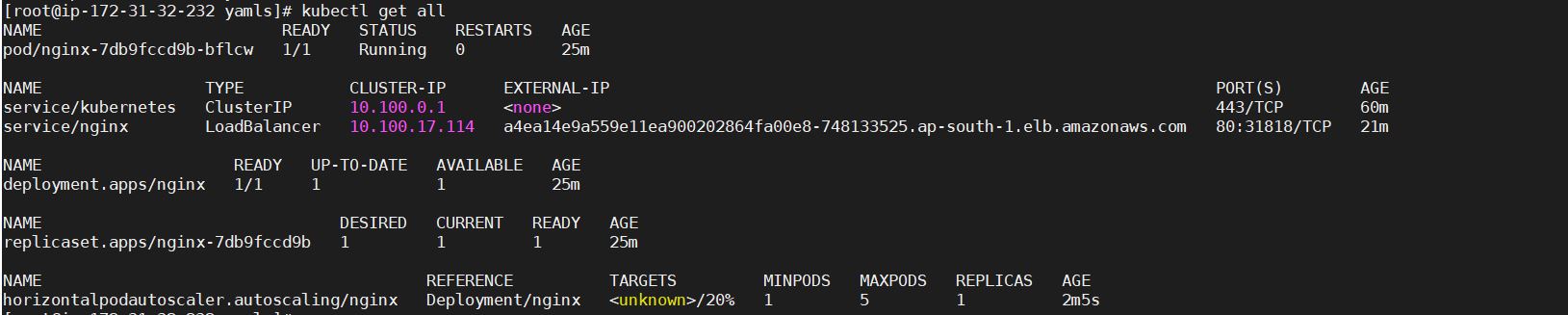
targetCPUUtilizationPercentage: 20

Now deploy HorizontalPodAutoscaler for nginx:

kubectl create -f 10-hpa.yaml

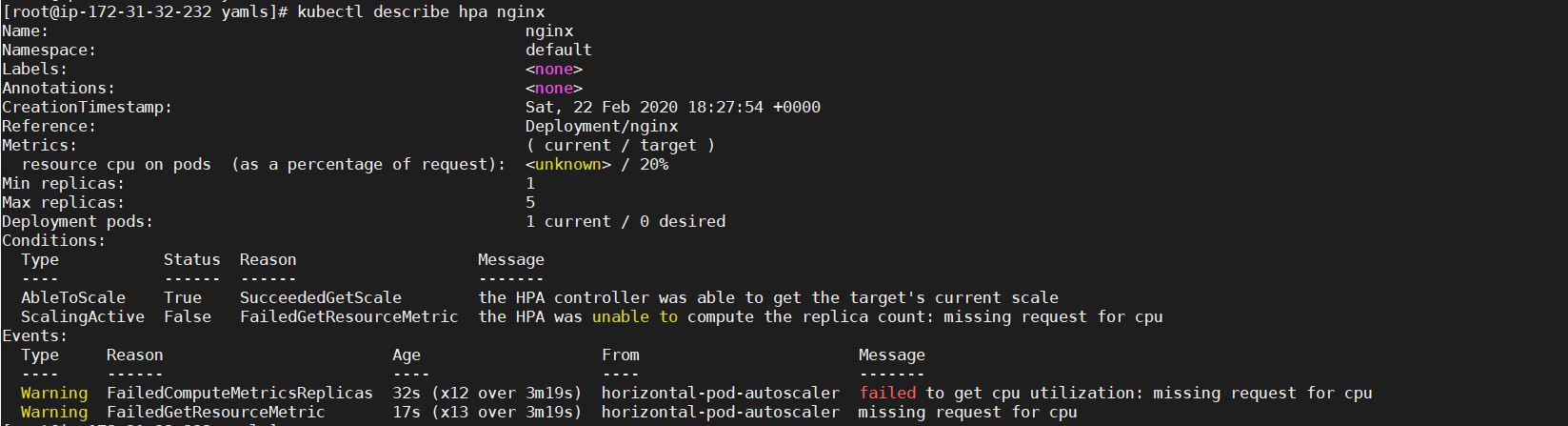
D:\Naresh\snipping\Capture.JPG

kubectl get all



Check description of horizontalpodautoscaler for nginx:

kubectl describe hpa nginx



Now edit the nginx deployment for showing howmuch cpu utilizing:

kubectl edit deploy nginx

Add below details below the container of spec:

resources:

limits:

cpu: 100m

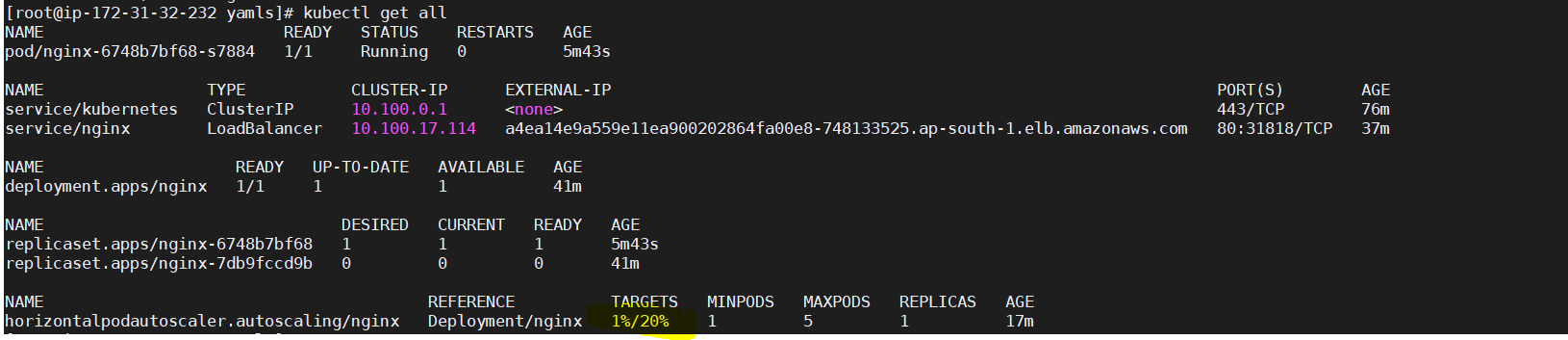
requests:

cpu: 100m



Now you can check whether it shows CPU utilization or not:

kubectl get all

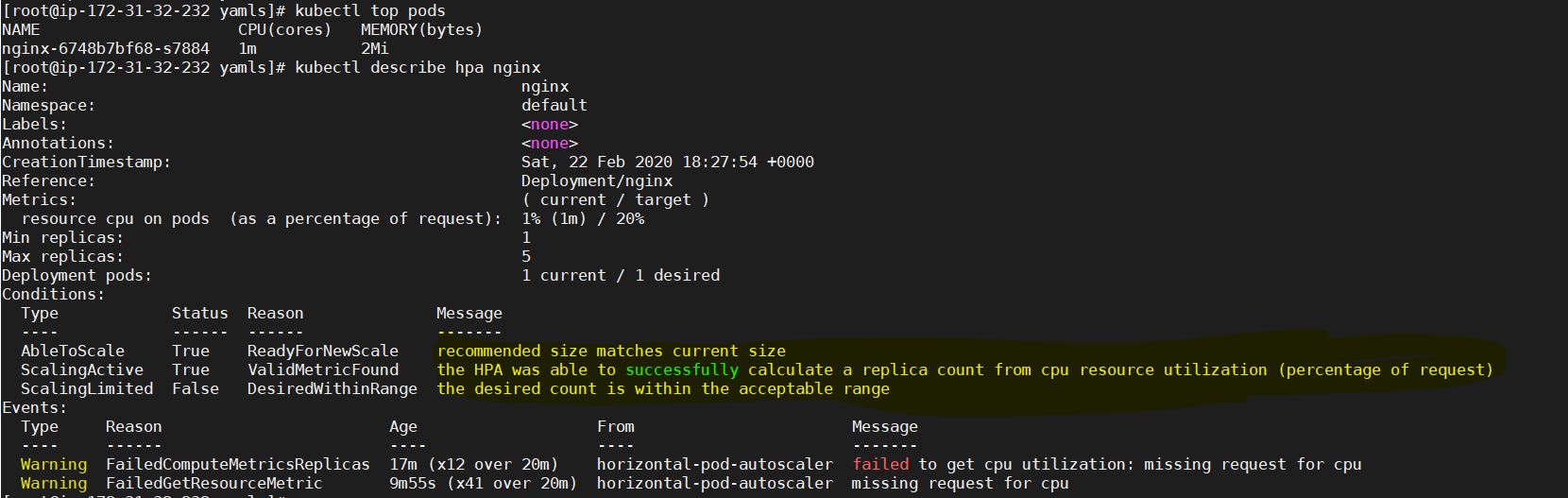


Here you can find only one Replica running…

Now check description of hpa for ngix:

kubectl top pods

kubectl describe hpa nginx



Here you can see HPA was able to successfully calculate a replica count from CPU utilization

Connect to pod and increase CPU utilization and check whether the pods are increased or not:

kubectl get pods

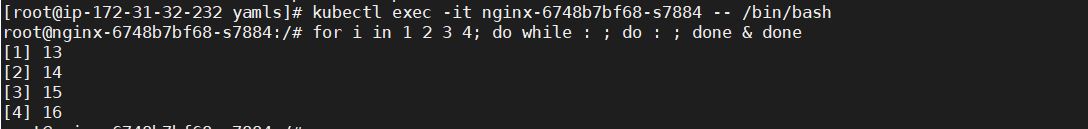


Connect to POD:

kubectl exec -it nginx-6748b7bf68-s7884 -- /bin/bash

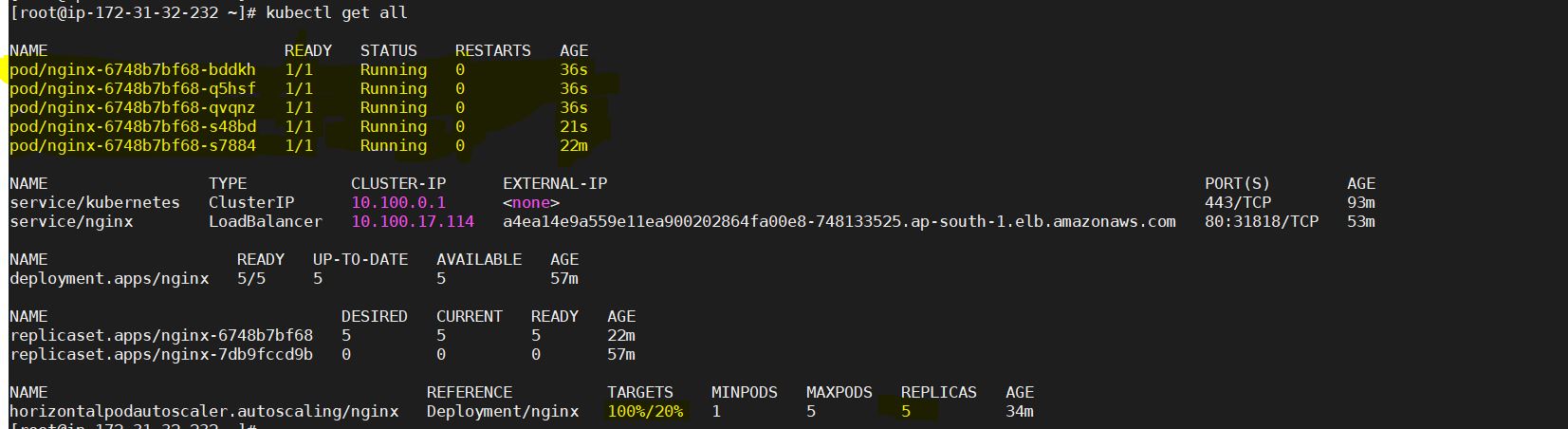
To increase CPU utilization give below command:

for i in 1 2 3 4; do while : ; do : ; done & done



Check whether the pods are increased or not:

kubectl get all



Because of HPA pods are created when CPU utilization increased more than 20%

Now going to Reduce CPU utilization to check whether the pods are reducing accordingly or not:

for i in 1 2 3 4; do kill %$i; done

Now check with below command whether the pods are decreased or not: (Wait 2min to 3min)

kubectl get all

