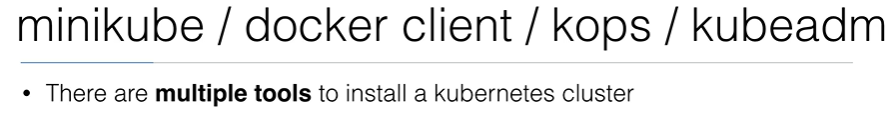
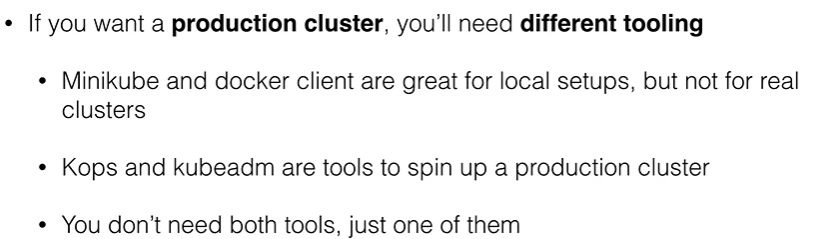
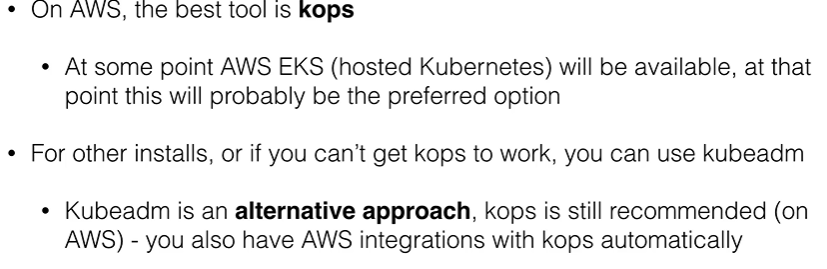
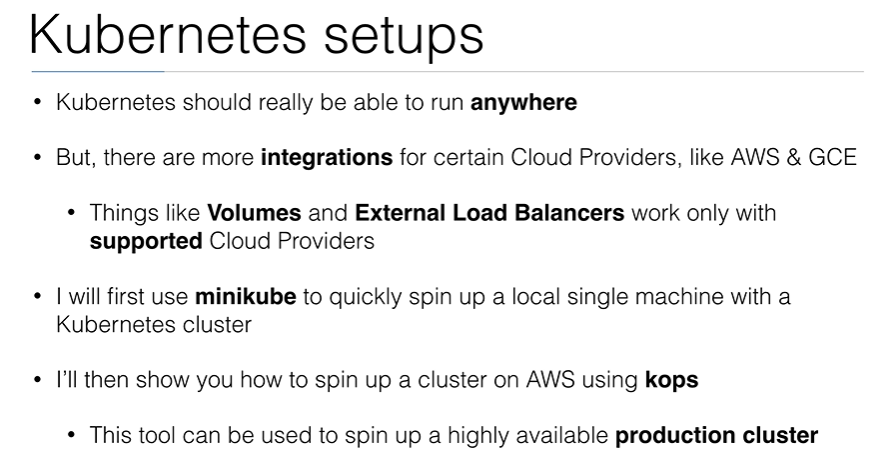
Kubernetes Setup

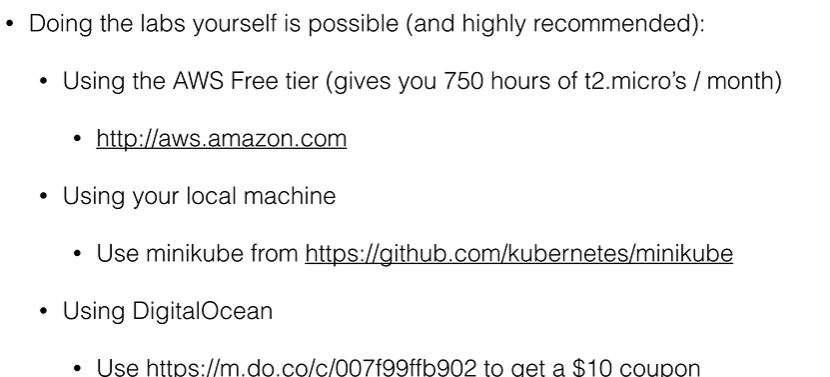
There are multiple tools to install k8s



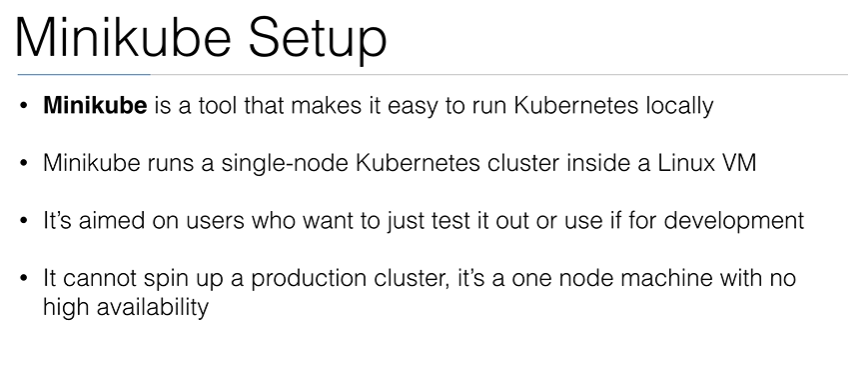


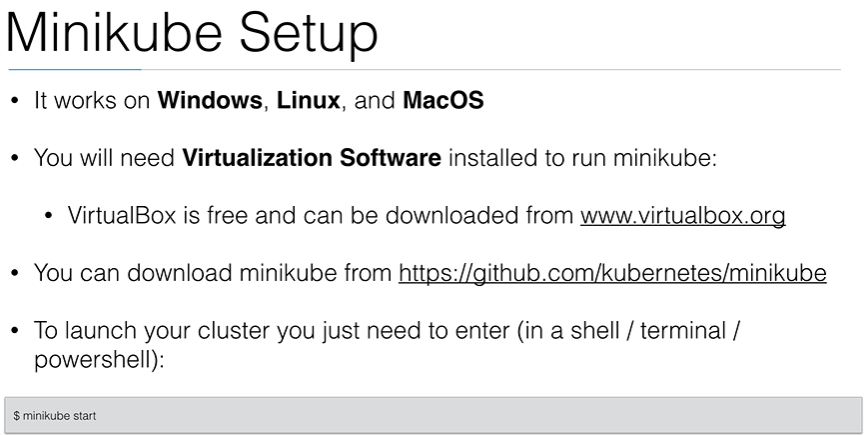






1. MiniKube





**Kubernetes on AWS**

To install k8s on AWS, We will use tool KOPS (Kubernetes Operation)

This tool allows to do production level deployment, upgradation and management.

There is also Legacy tool called kube-up.sh

Kops only work on Mac and Linux.

If you want to use it on Windows you have to use Vagrant.

K8s using EKS

<https://kubernetes.io/docs/setup/>

#### Pre-requisites:

- an EC2 Instance

- Install AWSCLI latest verison

step 1 : Launch EC2 instance

step2 : go to root user

sudo su –

step3: check aws version

aws –version

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

unzip awscliv2.zip

sudo ./aws/install

exit

sudo su -

Now check aws –version



1. Setup kubectl

https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html

a. Download kubectl version 1.21

b. Grant execution permissions to kubectl executable

c. Move kubectl onto /usr/local/bin

d. Test that your kubectl installation was successful

curl -O kubectl https://s3.us-west-2.amazonaws.com/amazon-eks/1.28.2/2023-10-17/bin/linux/arm64/kubectl

chmod +x ./kubectl

mv ./kubectl /usr/local/bin

1. Setup eksctl

https://eksctl.io/installation/

a. Download and extract the latest release

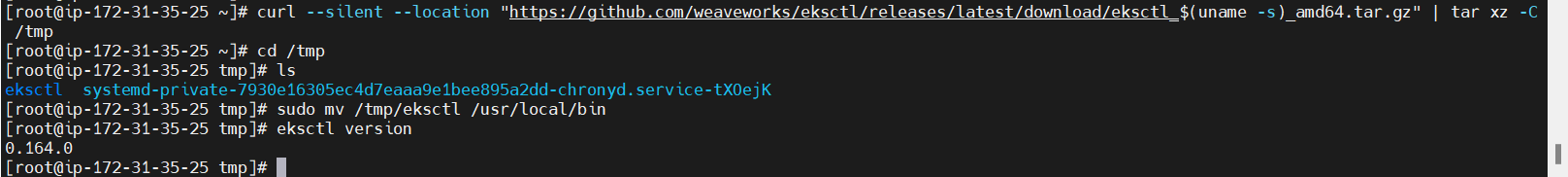
b. Move the extracted binary to /usr/local/bin

c. Test that your eksclt installation was successful

curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" | tar xz -C /tmp

sudo mv /tmp/eksctl /usr/local/bin

eksctl version



1. Create an IAM Role and attache it to EC2 instance

`Note: create IAM user with programmatic access if your bootstrap system is outside of AWS`

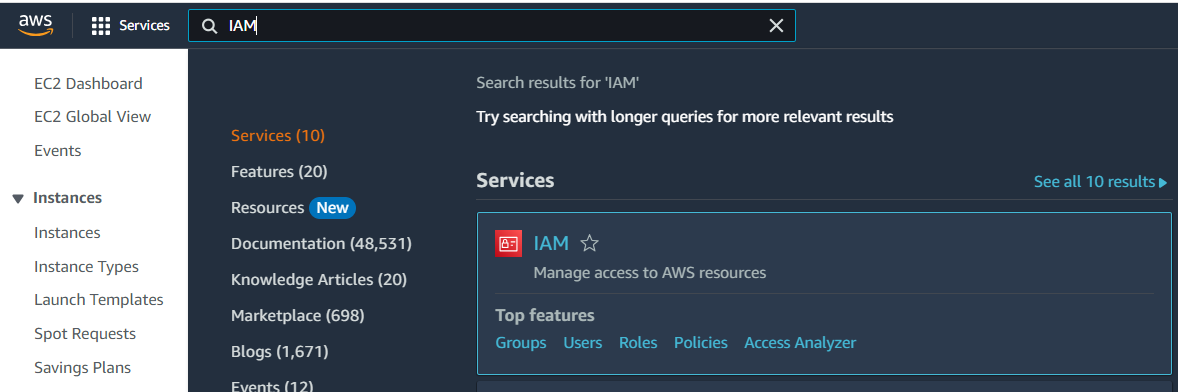
IAM user should have access to

IAM

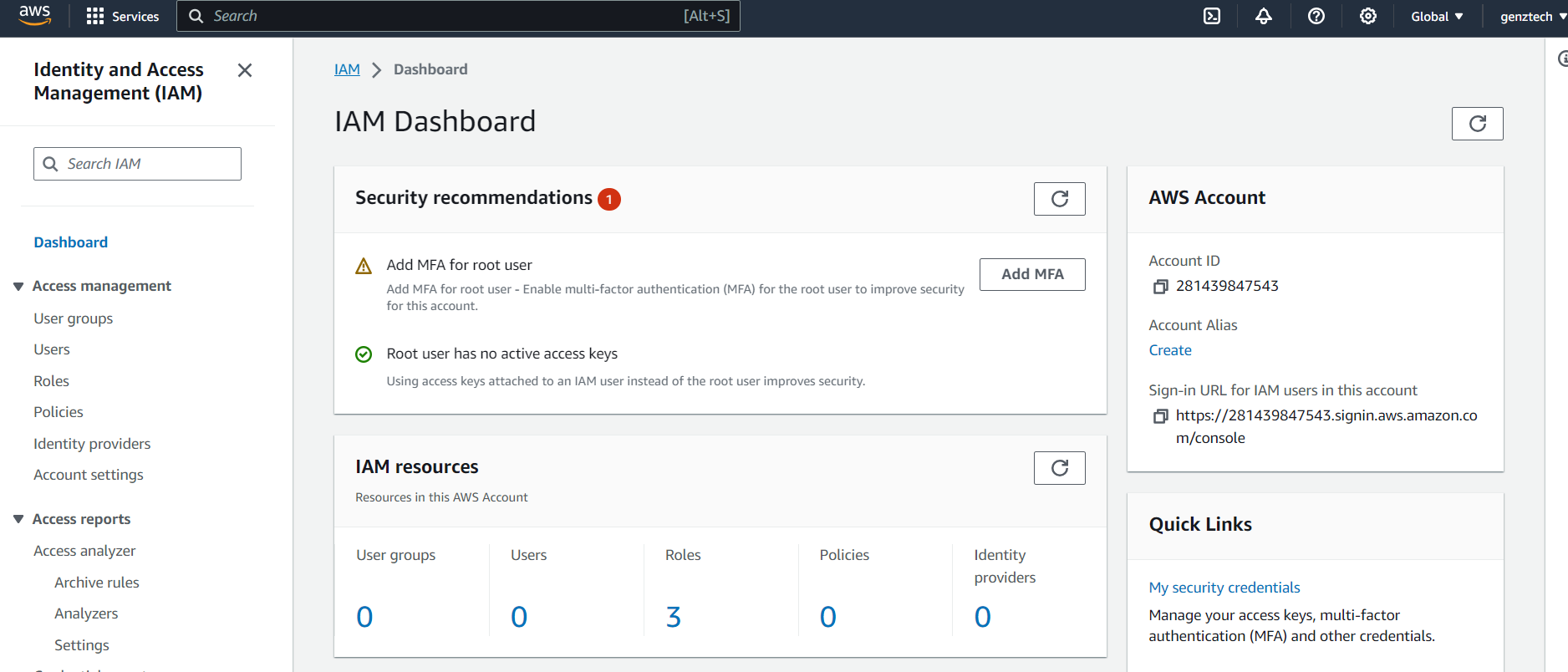
EC2

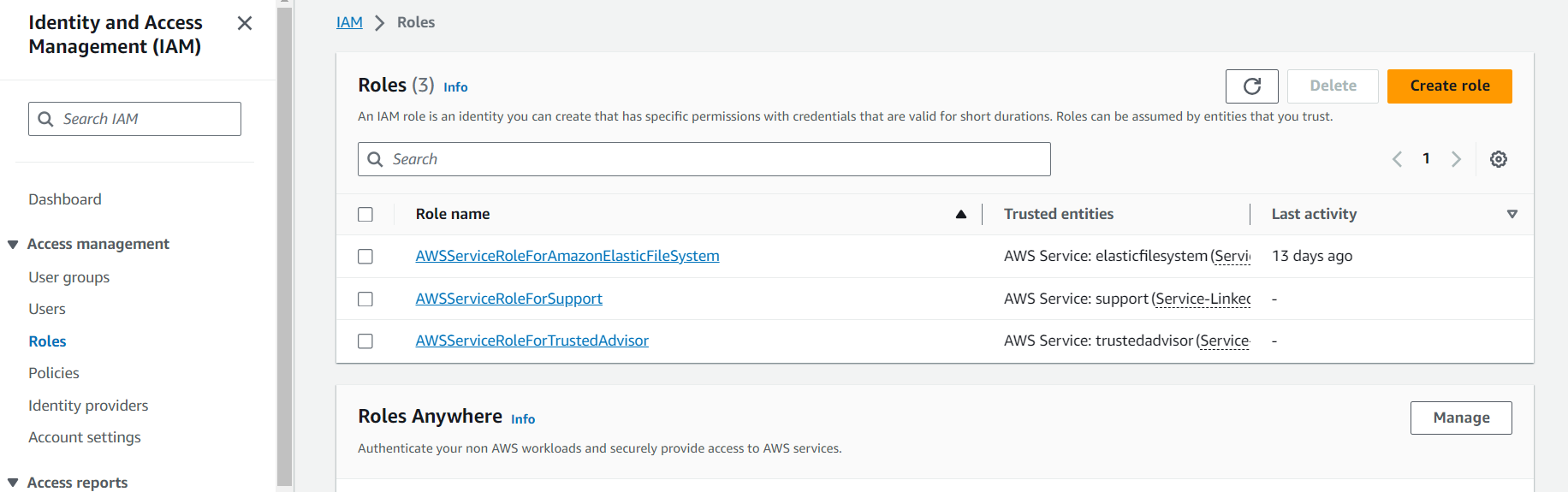
CloudFormation

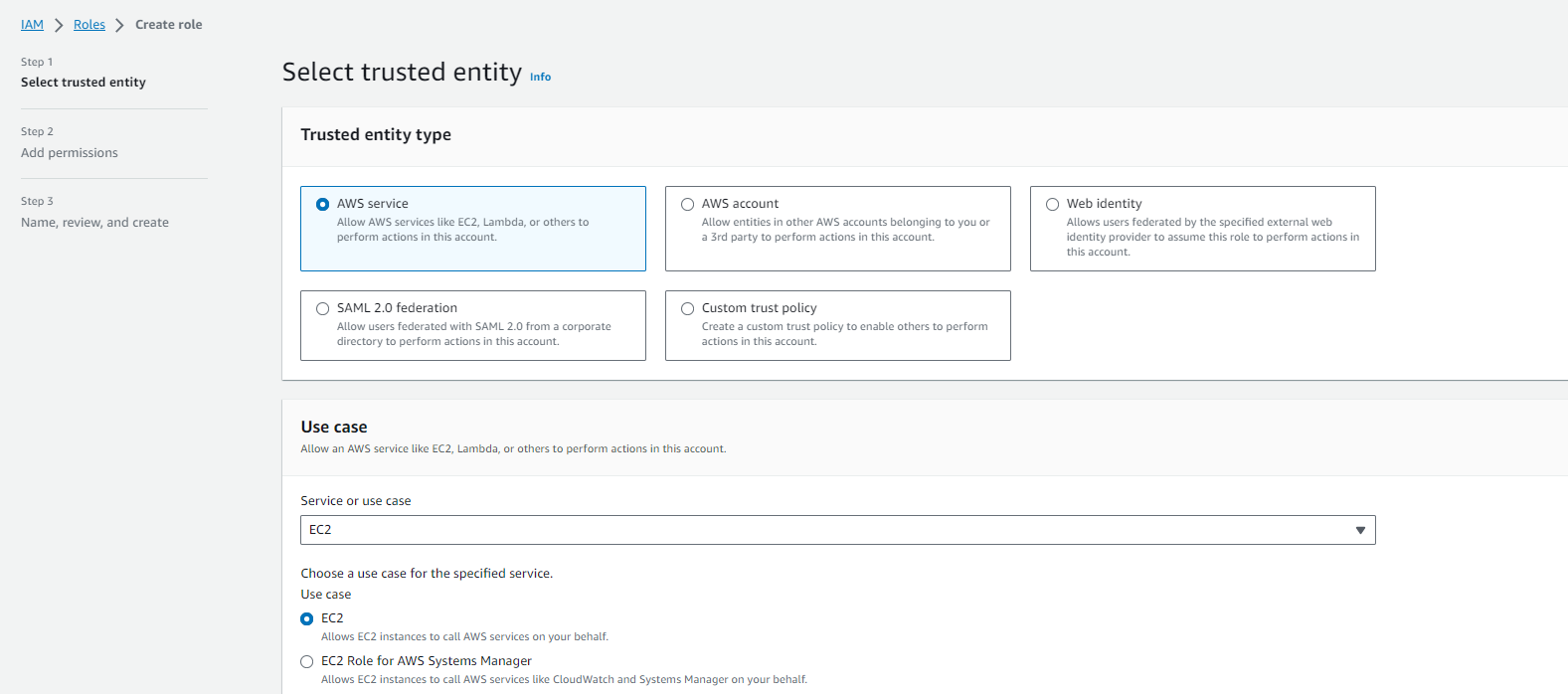
Note: Check eksctl documentaiton for [Minimum IAM policies](https://eksctl.io/usage/minimum-iam-policies/)



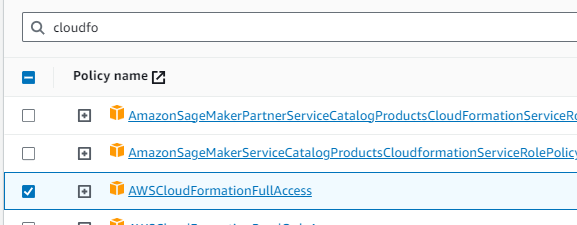
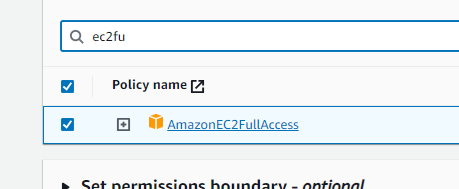
Click on Roles 🡪 Create roles 🡪

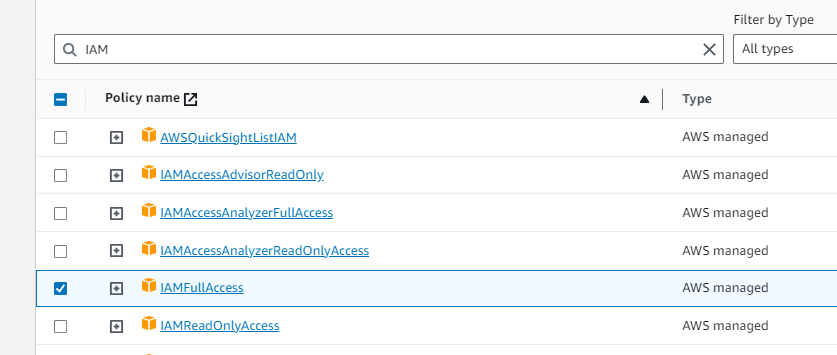


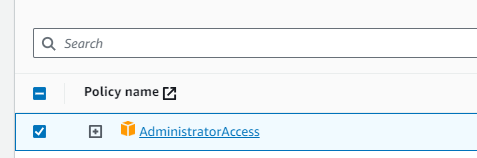




ec2ful

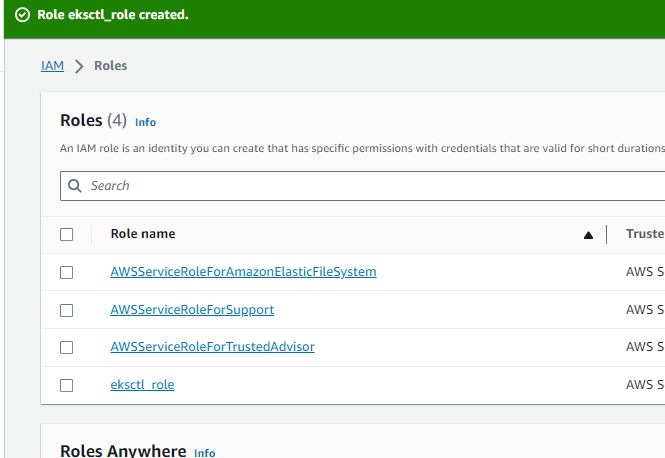
 





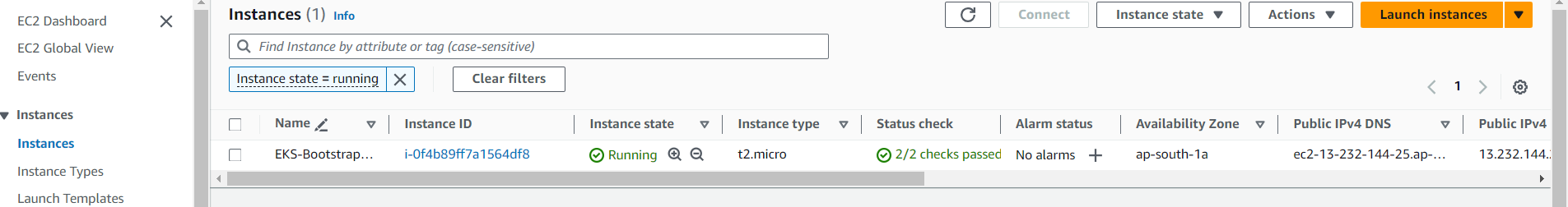
Click next

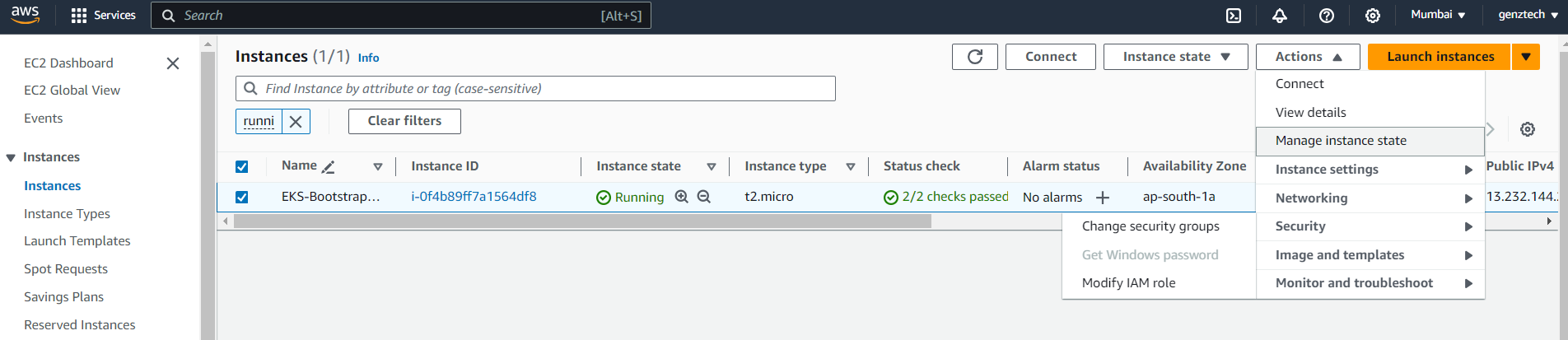
Click on create role

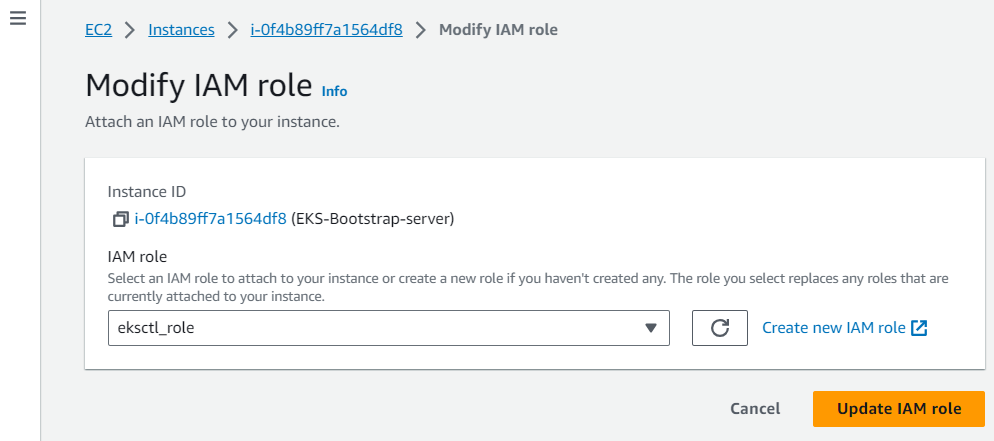


Now add this role to your eks\_bootstrap\_server

GO to EC2 instance , select instance, go to action, security, modify IAm ROLE







4. Create your cluster and nodes

```sh

eksctl create cluster --name cluster-name \

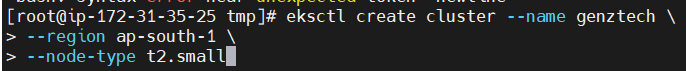
--region region-name \

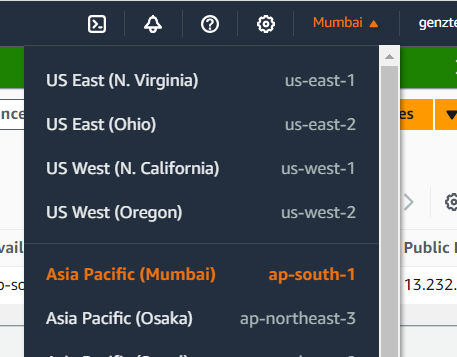
--node-type instance-type \

--nodes-min 2 \

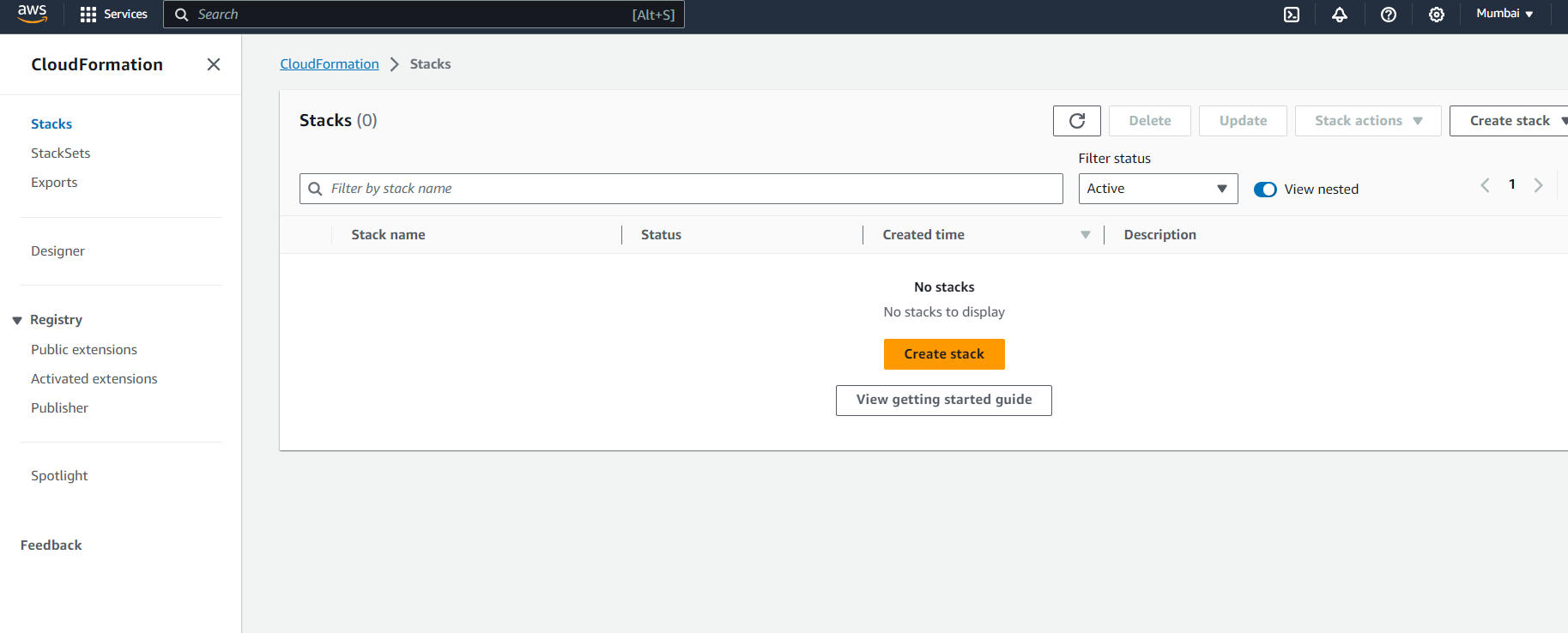
--nodes-max 2 \

--zones <AZ-1>,<AZ-2>

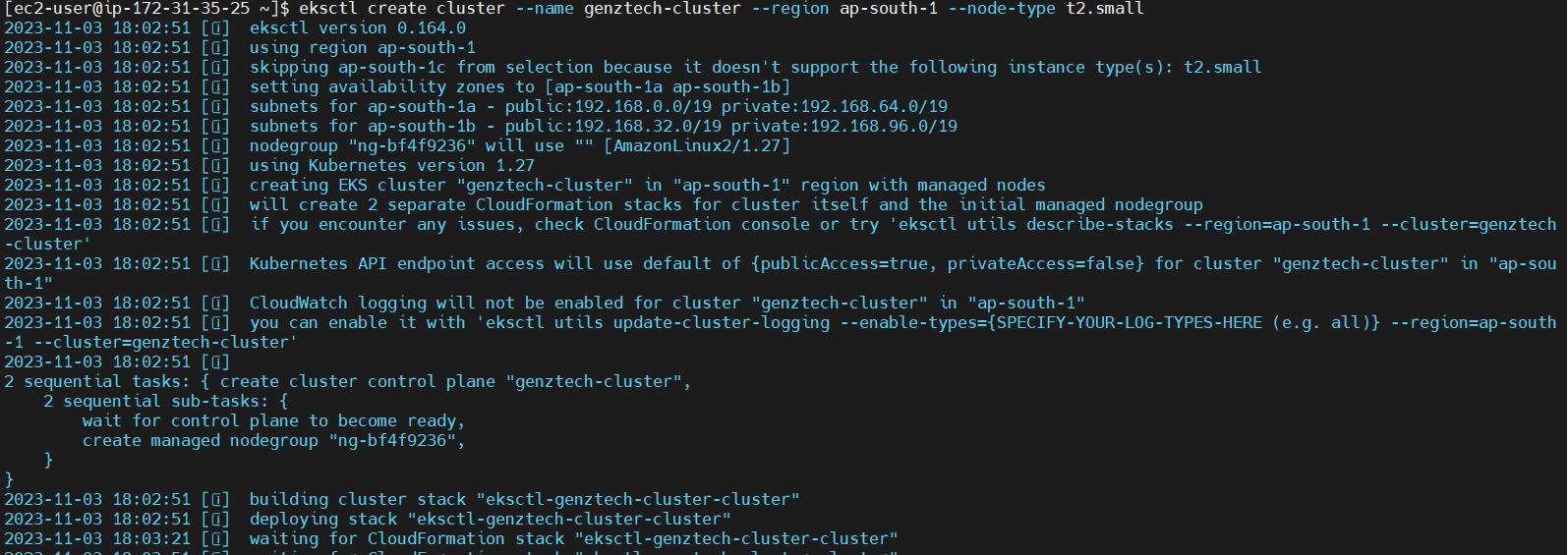


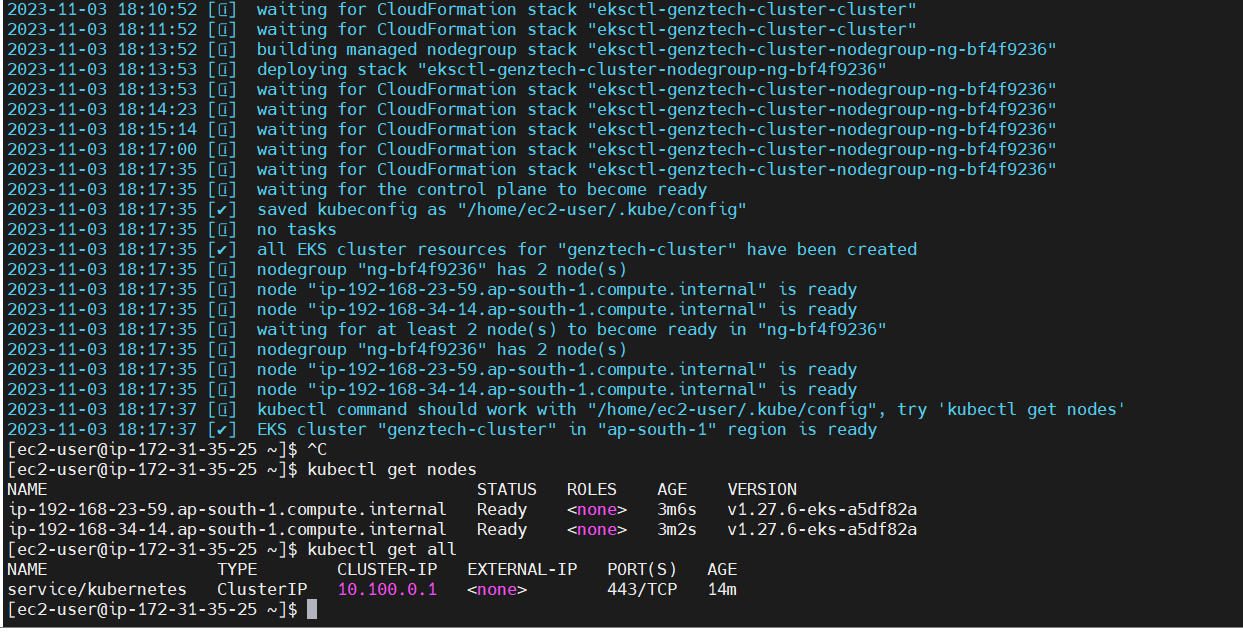


Before running create cluster command please check cloud formation



After running command for create cluster we have to wait for 20/30 mins, it will create fclous formation template

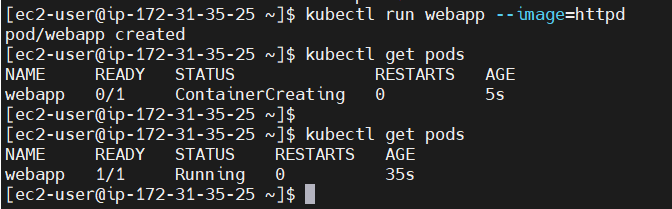




Kubectl get nodes

Kubectl gel all 🡪 all resources

kubectl run webapp --image=httpd 🡪 create demo pod



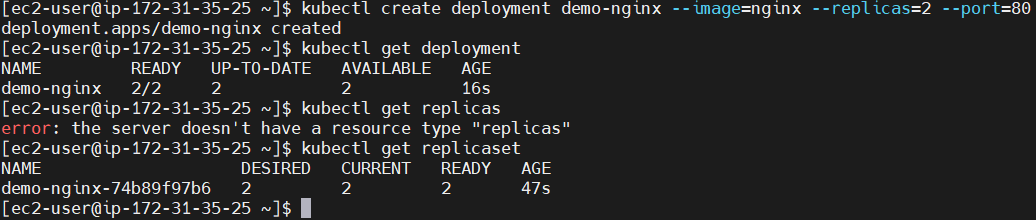
kubectl exec -it webapp /bin/bash

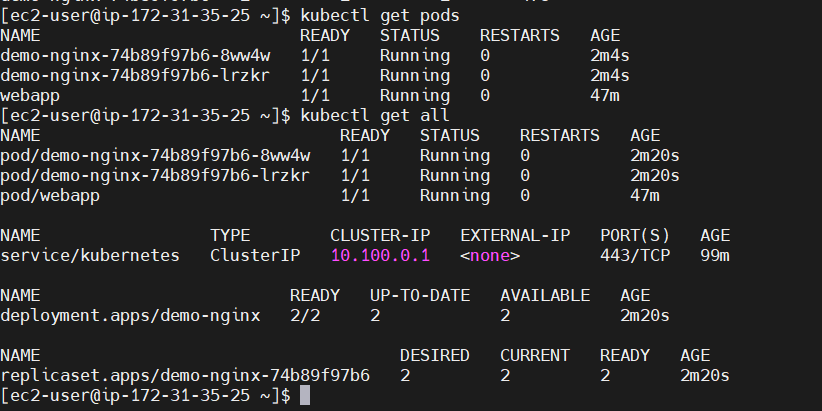
Deploying Nginx container

kubectl create deployment demo-nginx –image=nginx –replicas=2 –port=80

kubectl get deployment

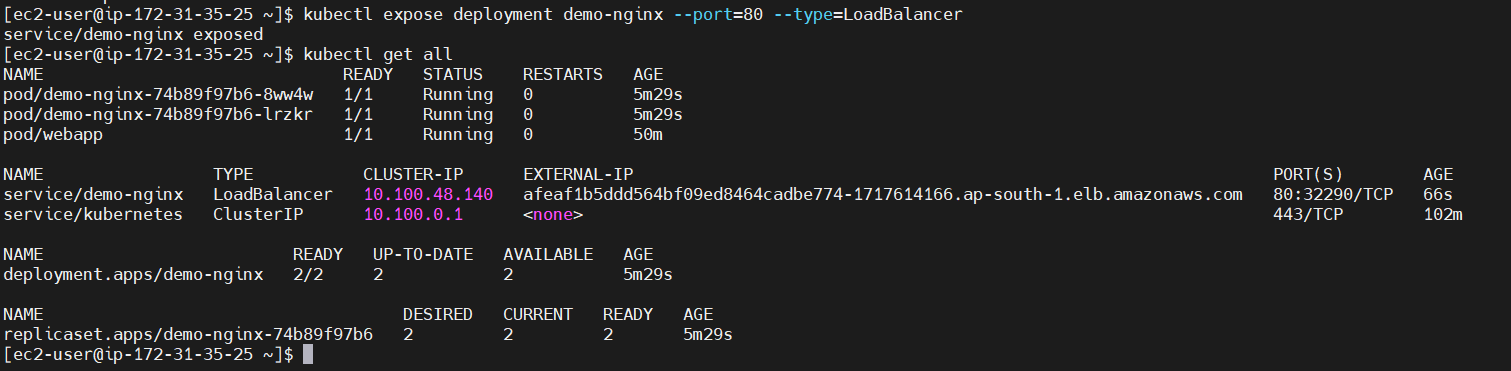
kubectl get replicaset



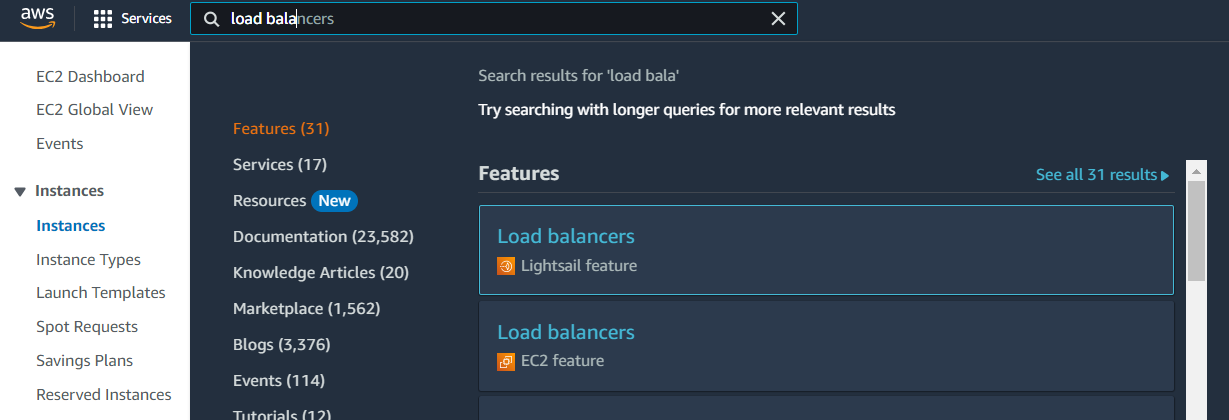


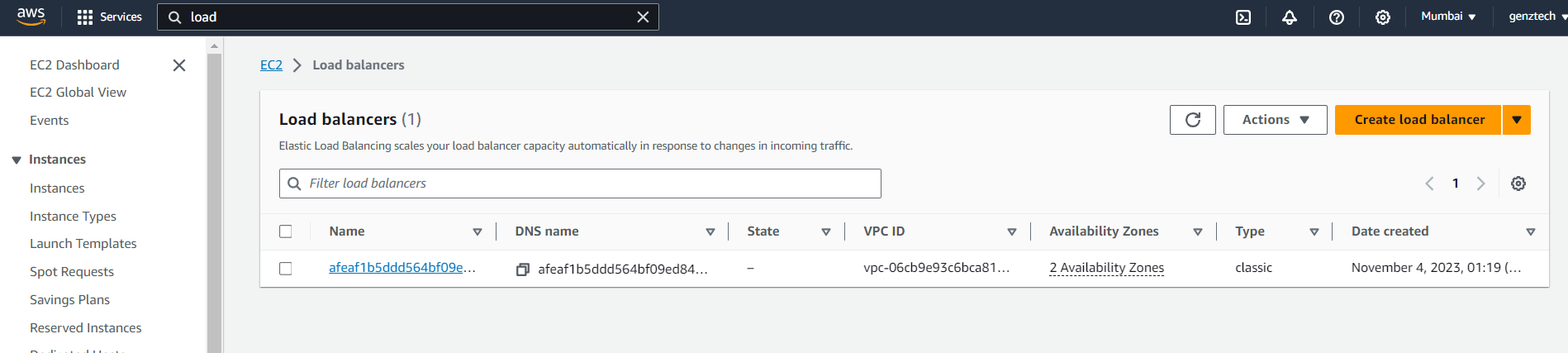
Now expose this to external world

kubectl expose deployment demo-nginx --port=80 --type=LoadBalancer



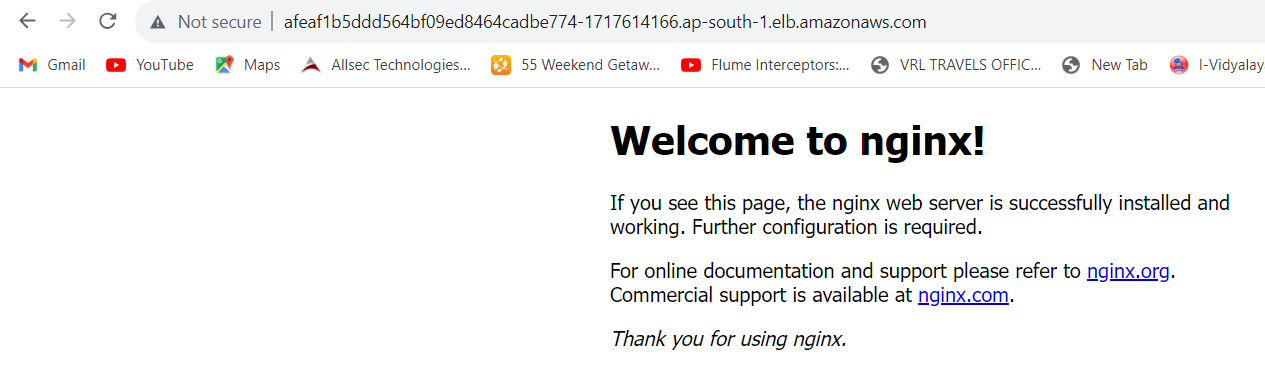
It creates Load balancer also, which you can check in console





In browser use external ip to access nginx

afeaf1b5ddd564bf09ed8464cadbe774-1717614166.ap-south-1.elb.amazonaws.com



Now we will create manifest file to do above things but first we will delete deployment

Whenever we delete deployment , it will delete pods and replicaset

Delete

kubectl delete deployment demo-nginx

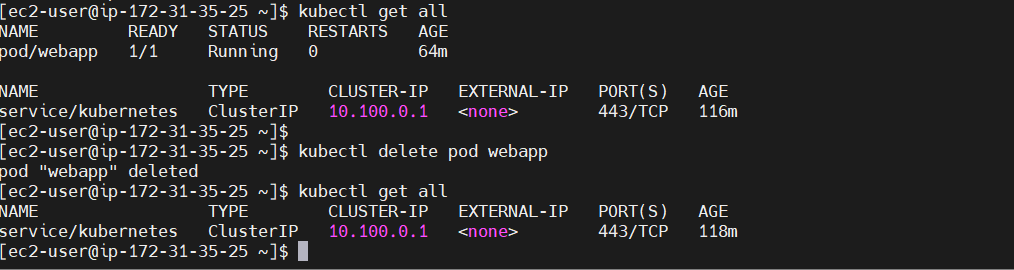


We also need to delete LoadBalancer i.e. service

kubectl delete service/demo-nginx



Kubectl delete pod webapp



Create POD and service using manifest file

<https://kubernetes.io/docs/concepts/workloads/pods/>

<https://kubernetes.io/docs/tasks/configure-pod-container/static-pod/>

[ec2-user@ip-172-31-35-25 ~]$ cat pod.yml

apiVersion: v1

kind: Pod

metadata:

name: demo-pod

labels:

app: demo-app

spec:

containers:

- name: demo-nginx

image: nginx

ports:

- name: demo-nginx

containerPort: 80

**Now we have to create service**

[ec2-user@ip-172-31-35-25 ~]$ cat service.yml

apiVersion: v1

kind: Service

metadata:

name: demo-service

spec:

ports:

- name: nginx-port

port: 80

targetPort: 80

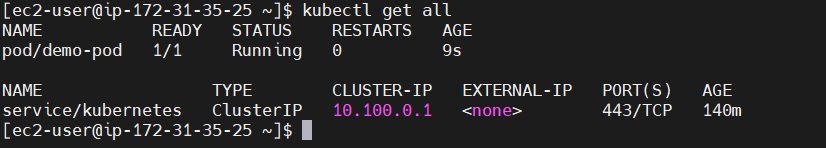
selector:

app: demo-app

type: LoadBalancer

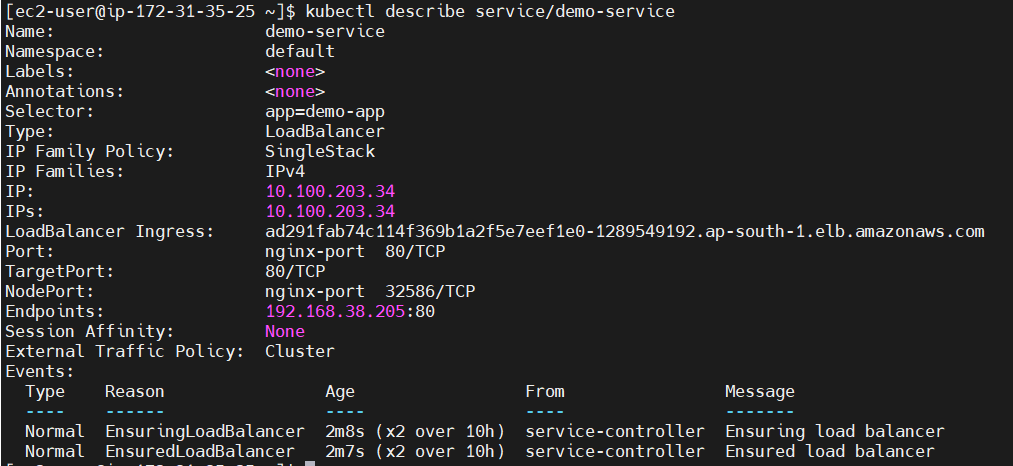
Create POD

kubectl apply -f pod.yml



Create service

kubectl apply -f service.yml



Access it in browser using external ip

Kubectl get pod -o wide

