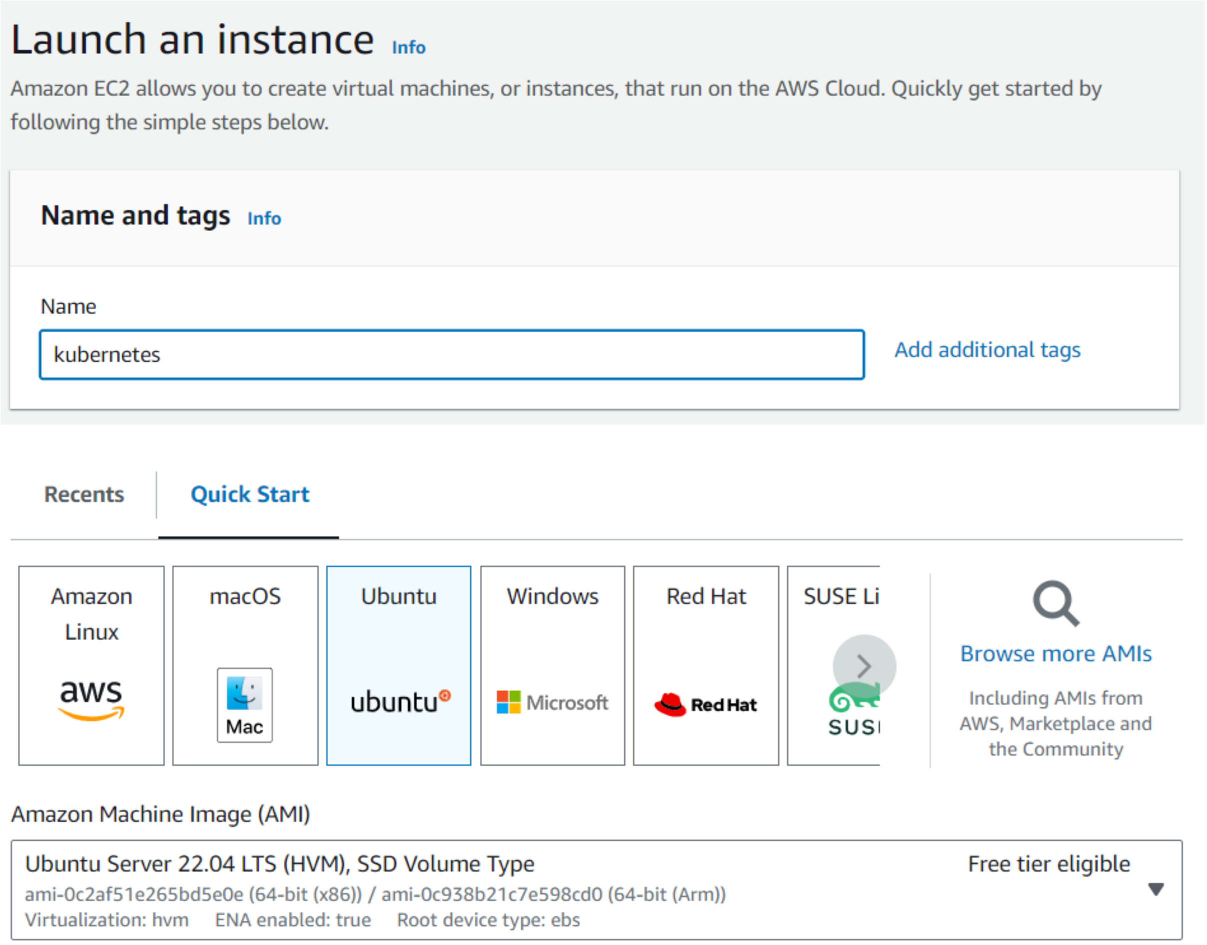
Task 20: Pods and Services in Kubernetes Deployment

1. Launch an instance:



Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

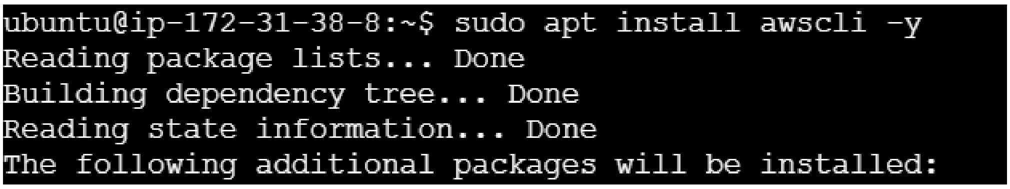
Name and tags

Name: kubernetes

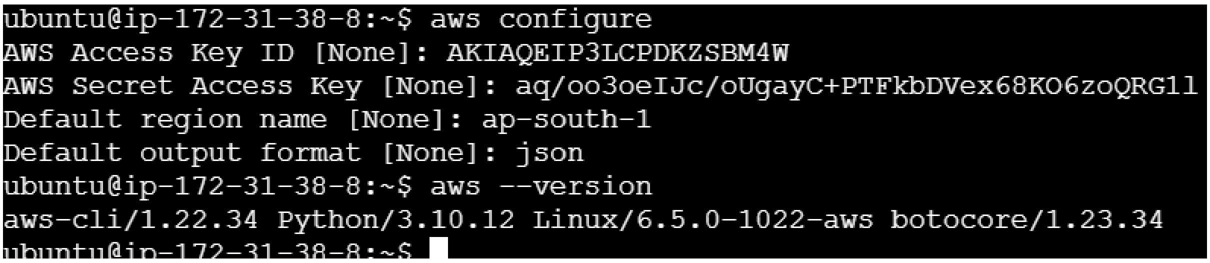
Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0c2af51e265bd5e0e (64-bit (x86)) / ami-0c938b21c7e598cd0 (64-bit (Arm))
Virtualization: hvm   ENA enabled: true   Root device type: ebs

2. Install awscli:

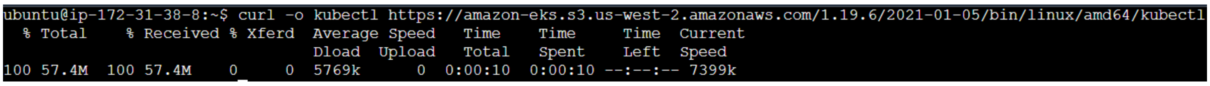


```
ubuntu@ip-172-31-38-8:~$ sudo apt install awscli -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

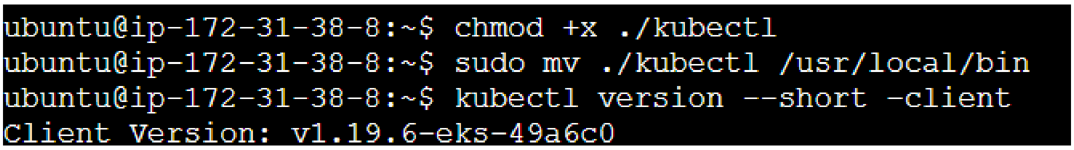


```
ubuntu@ip-172-31-38-8:~$ aws configure
AWS Access Key ID [None]: AKIAQEIP3LCPDKZSBM4W
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]: json
ubuntu@ip-172-31-38-8:~$ aws --version
aws-cli/1.22.34 Python/3.10.12 Linux/6.5.0-1022-aws botocore/1.23.34
ubuntu@ip-172-31-38-8:~$
```

3. Install kubectl:

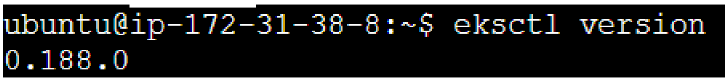


```
ubuntu@ip-172-31-38-8:~$ curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectl
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 57.4M  100 57.4M    0     0  5769k      0  0:00:10  0:00:10 --:--:-- 7399k
```



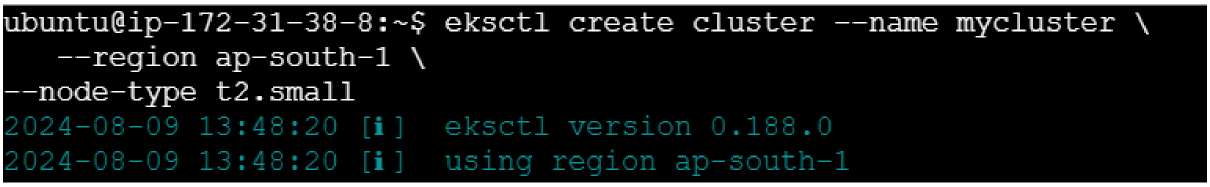
```
ubuntu@ip-172-31-38-8:~$ chmod +x ./kubectl
ubuntu@ip-172-31-38-8:~$ sudo mv ./kubectl /usr/local/bin
ubuntu@ip-172-31-38-8:~$ kubectl version --short -client
Client Version: v1.19.6-eks-49a6c0
```

4. Install eksctl:

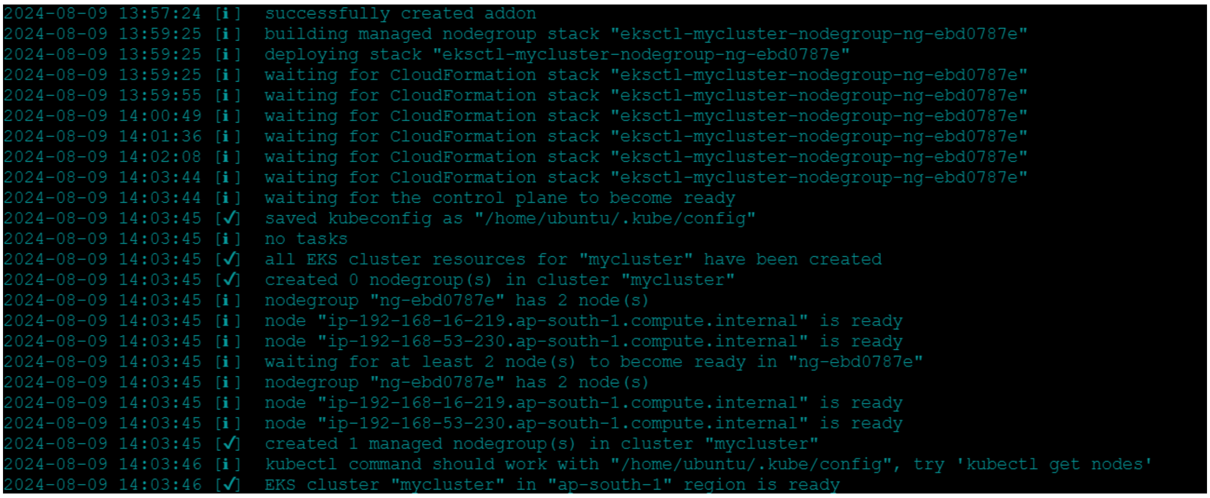


```
ubuntu@ip-172-31-38-8:~$ eksctl version
0.188.0
```

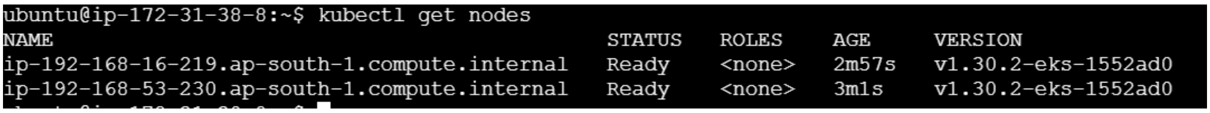
5. Create eks cluster:



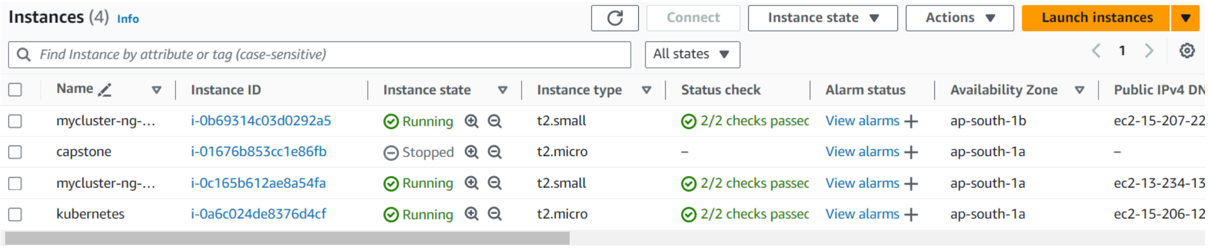
```
ubuntu@ip-172-31-38-8:~$ eksctl create cluster --name mycluster \
   --region ap-south-1 \
--node-type t2.small
2024-08-09 13:48:20 [ℹ]  eksctl version 0.188.0
2024-08-09 13:48:20 [ℹ]  using region ap-south-1
```



```
2024-08-09 13:57:24 [ℹ]  successfully created addon
2024-08-09 13:59:25 [ℹ]  building managed nodegroup stack "eksctl-mycluster-nodegroup-ng-ebd0787e"
2024-08-09 13:59:25 [ℹ]  deploying stack "eksctl-mycluster-nodegroup-ng-ebd0787e"
2024-08-09 13:59:25 [ℹ]  waiting for CloudFormation stack "eksctl-mycluster-nodegroup-ng-ebd0787e"
2024-08-09 13:59:55 [ℹ]  waiting for CloudFormation stack "eksctl-mycluster-nodegroup-ng-ebd0787e"
2024-08-09 14:00:49 [ℹ]  waiting for CloudFormation stack "eksctl-mycluster-nodegroup-ng-ebd0787e"
2024-08-09 14:01:36 [ℹ]  waiting for CloudFormation stack "eksctl-mycluster-nodegroup-ng-ebd0787e"
2024-08-09 14:02:08 [ℹ]  waiting for CloudFormation stack "eksctl-mycluster-nodegroup-ng-ebd0787e"
2024-08-09 14:03:44 [ℹ]  waiting for CloudFormation stack "eksctl-mycluster-nodegroup-ng-ebd0787e"
2024-08-09 14:03:44 [ℹ]  waiting for the control plane to become ready
2024-08-09 14:03:45 [✓]  saved kubeconfig as "/home/ubuntu/.kube/config"
2024-08-09 14:03:45 [ℹ]  no tasks
2024-08-09 14:03:45 [✓]  all EKS cluster resources for "mycluster" have been created
2024-08-09 14:03:45 [✓]  created 0 nodegroup(s) in cluster "mycluster"
2024-08-09 14:03:45 [ℹ]  nodegroup "ng-ebd0787e" has 2 node(s)
2024-08-09 14:03:45 [ℹ]  node "ip-192-168-16-219.ap-south-1.compute.internal" is ready
2024-08-09 14:03:45 [ℹ]  node "ip-192-168-53-230.ap-south-1.compute.internal" is ready
2024-08-09 14:03:45 [ℹ]  waiting for at least 2 node(s) to become ready in "ng-ebd0787e"
2024-08-09 14:03:45 [ℹ]  nodegroup "ng-ebd0787e" has 2 node(s)
2024-08-09 14:03:45 [ℹ]  node "ip-192-168-16-219.ap-south-1.compute.internal" is ready
2024-08-09 14:03:45 [ℹ]  node "ip-192-168-53-230.ap-south-1.compute.internal" is ready
2024-08-09 14:03:45 [✓]  created 1 managed nodegroup(s) in cluster "mycluster"
2024-08-09 14:03:46 [ℹ]  kubectl command should work with "/home/ubuntu/.kube/config", try 'kubectl get nodes'
2024-08-09 14:03:46 [✓]  EKS cluster "mycluster" in "ap-south-1" region is ready
```

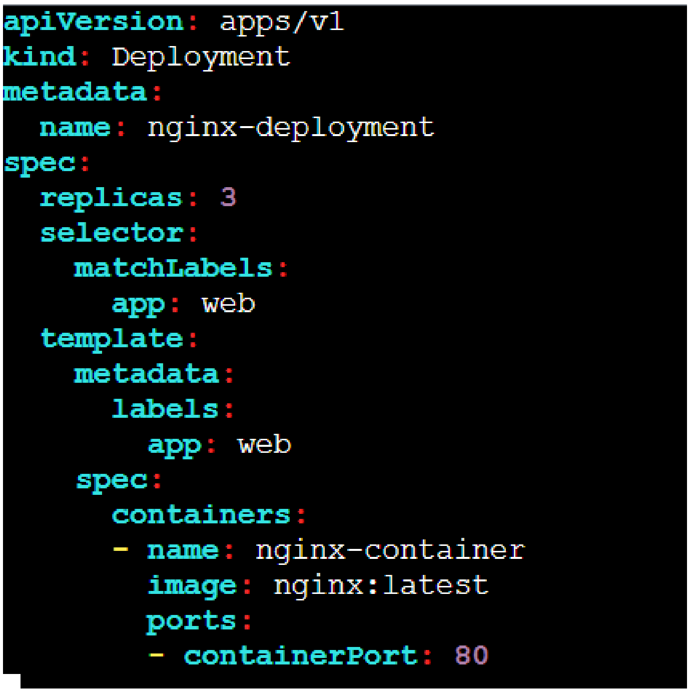


```
ubuntu@ip-172-31-38-8:~$ kubectl get nodes
NAME                                            STATUS   ROLES    AGE     VERSION
ip-192-168-16-219.ap-south-1.compute.internal   Ready    <none>   2m57s   v1.30.2-eks-1552ad0
ip-192-168-53-230.ap-south-1.compute.internal   Ready    <none>   3m1s    v1.30.2-eks-1552ad0
```



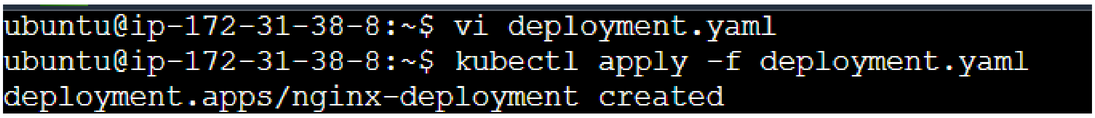
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
□	mycluster-ng-...	i-0b69314c03d0292a5	✓ Running	t2.small	✓ 2/2 checks passec	View alarms +	ap-south-1b	ec2-15-207-22
□	capstone	i-01676b853cc1e86fb	⊖ Stopped	t2.micro	–	View alarms +	ap-south-1a	–
□	mycluster-ng-...	i-0c165b612ae8a54fa	✓ Running	t2.small	✓ 2/2 checks passec	View alarms +	ap-south-1a	ec2-13-234-13
□	kubernetes	i-0a6c024de8376d4cf	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	ap-south-1a	ec2-15-206-12

6. **Deployment file:**



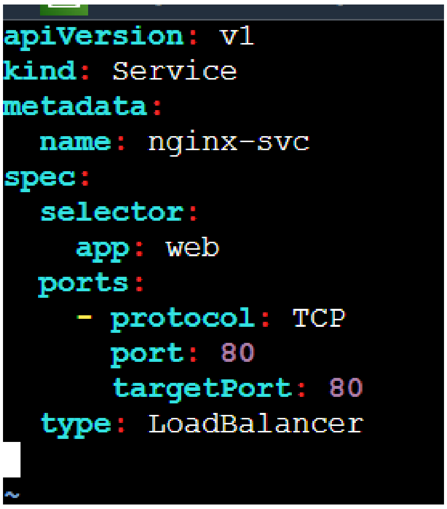
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
      - name: nginx-container
        image: nginx:latest
        ports:
        - containerPort: 80
```

Execute deployment.yaml file:



```
ubuntu@ip-172-31-38-8:~$ vi deployment.yaml
ubuntu@ip-172-31-38-8:~$ kubectl apply -f deployment.yaml
deployment.apps/nginx-deployment created
```

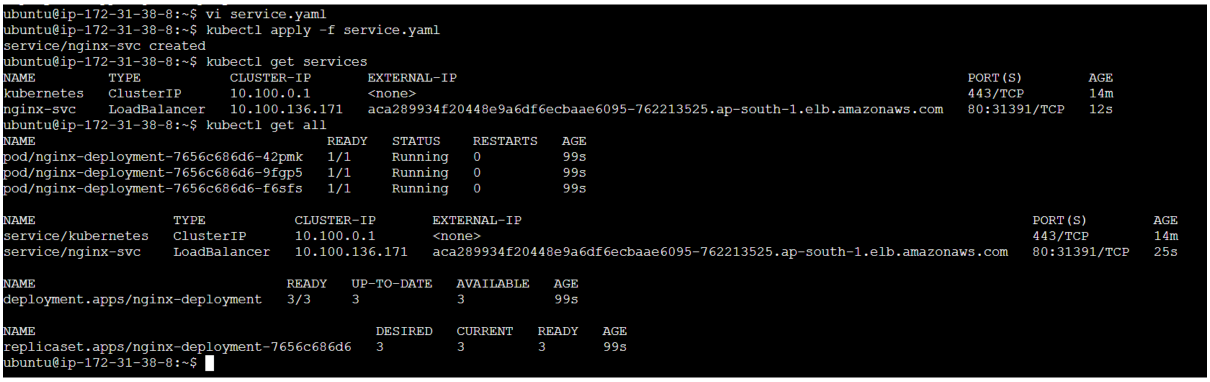
7. **service file:**



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

~
```

Execute service.yaml file:

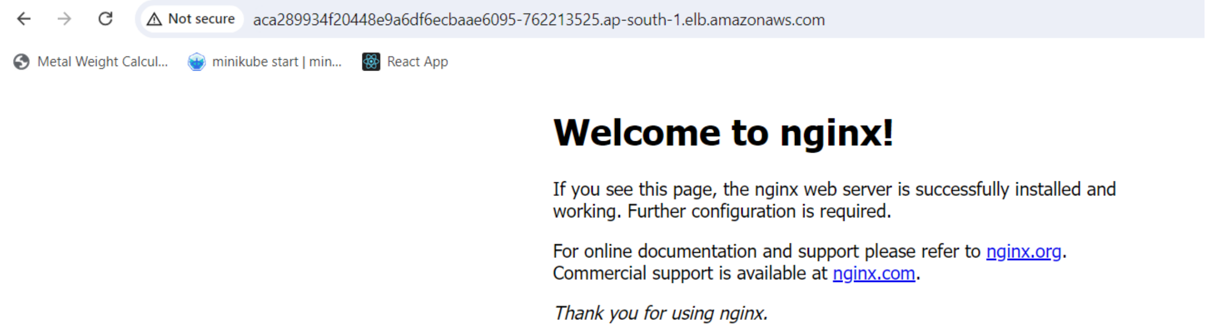


```
ubuntu@ip-172-31-38-8:~$ vi service.yaml
ubuntu@ip-172-31-38-8:~$ kubectl apply -f service.yaml
service/nginx-svc created
ubuntu@ip-172-31-38-8:~$ kubectl get services
NAME         TYPE           CLUSTER-IP       EXTERNAL-IP                                                                PORT(S)        AGE
kubernetes   ClusterIP      10.100.0.1       <none>                                                                     443/TCP        14m
nginx-svc    LoadBalancer   10.100.136.171   aca289934f20448e9a6df6ecbaae6095-762213525.ap-south-1.elb.amazonaws.com   80:31391/TCP   12s
ubuntu@ip-172-31-38-8:~$ kubectl get all
NAME                                    READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-7656c686d6-42pmk   1/1     Running   0          99s
pod/nginx-deployment-7656c686d6-9fgp5   1/1     Running   0          99s
pod/nginx-deployment-7656c686d6-f6sfs   1/1     Running   0          99s

NAME                 TYPE           CLUSTER-IP       EXTERNAL-IP                                                                PORT(S)        AGE
service/kubernetes   ClusterIP      10.100.0.1       <none>                                                                     443/TCP        14m
service/nginx-svc    LoadBalancer   10.100.136.171   aca289934f20448e9a6df6ecbaae6095-762213525.ap-south-1.elb.amazonaws.com   80:31391/TCP   25s

NAME                               READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment   3/3     3            3           99s

NAME                                          DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deployment-7656c686d6   3         3         3       99s
ubuntu@ip-172-31-38-8:~$ 
```



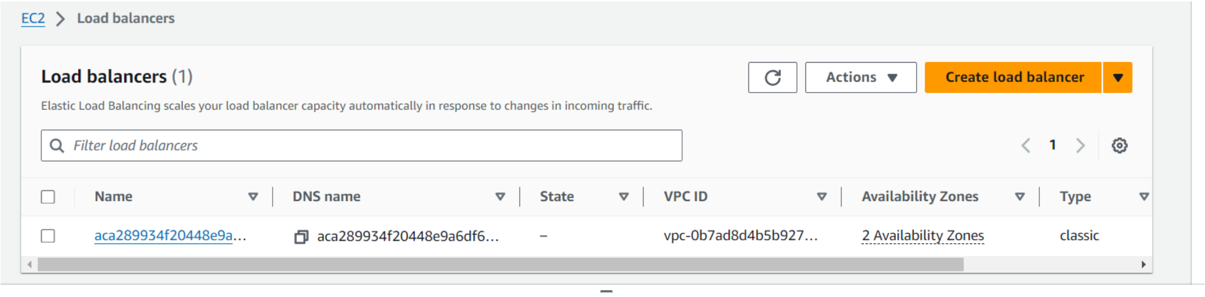
Not secure
aca289934f20448e9a6df6ecbaae6095-762213525.ap-south-1.elb.amazonaws.com
Metal Weight Calcul...
minikube start | min...
React App

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

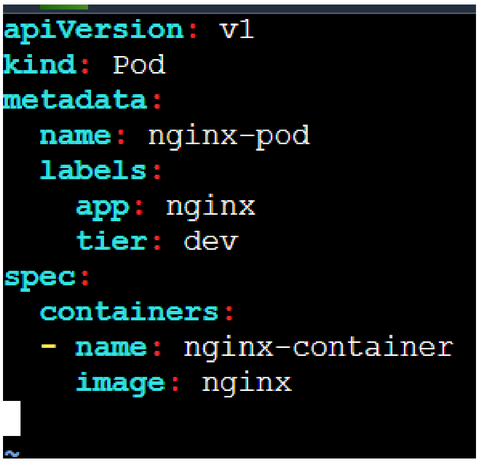


Load balancers (1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

□	Name	DNS name	State	VPC ID	Availability Zones	Type
□	aca289934f20448e9a...	aca289934f20448e9a6df6...	–	vpc-0b7ad8d4b5b927...	2 Availability Zones	classic

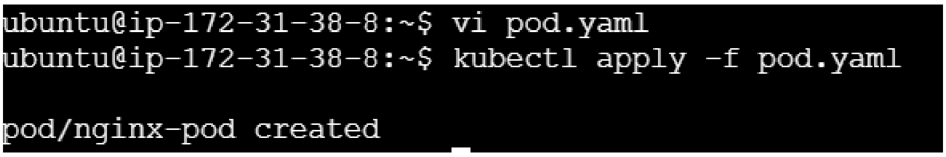
8. pod file:



```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
    tier: dev
spec:
  containers:
  - name: nginx-container
    image: nginx

~
```

Execute pod.yaml file:



```
ubuntu@ip-172-31-38-8:~$ vi pod.yaml
ubuntu@ip-172-31-38-8:~$ kubectl apply -f pod.yaml

pod/nginx-pod created
```