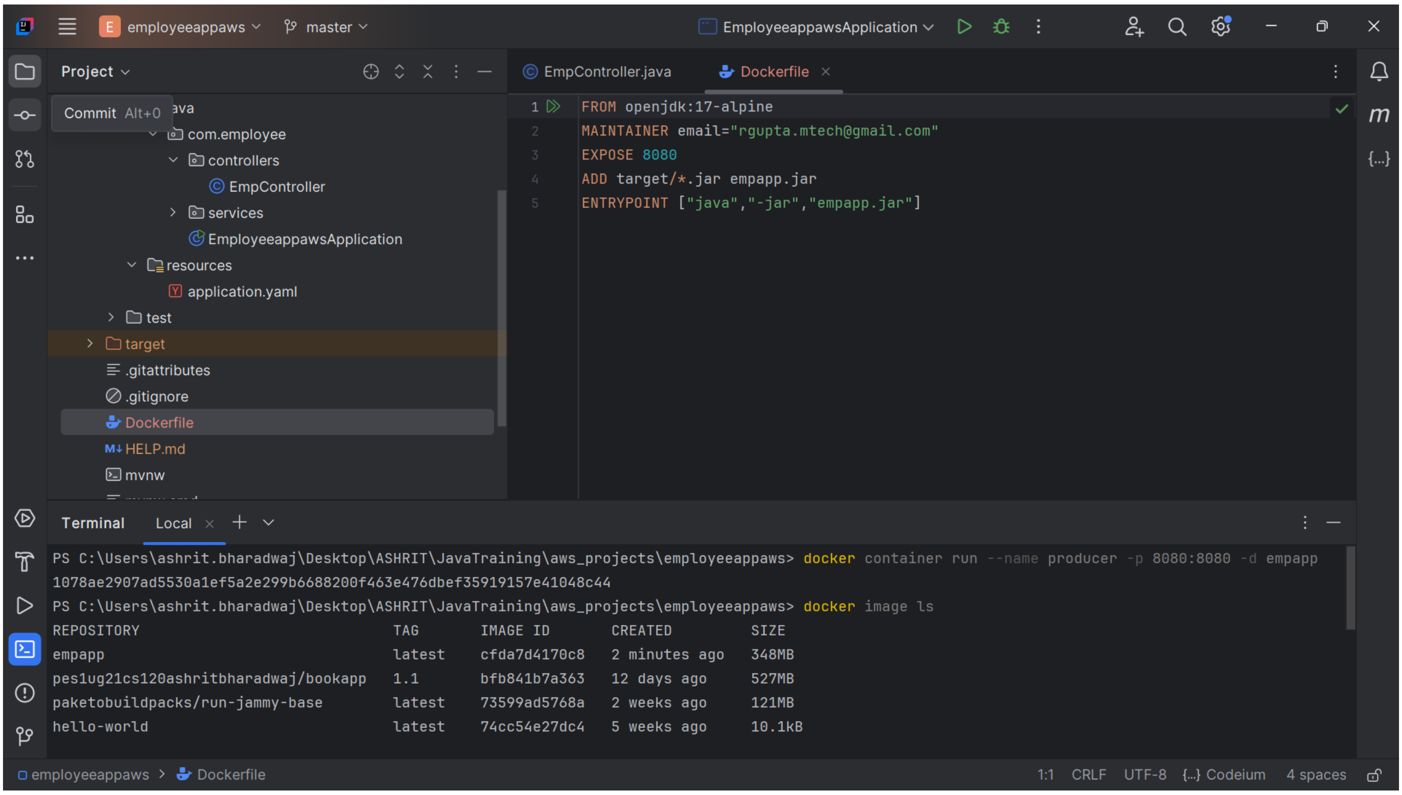
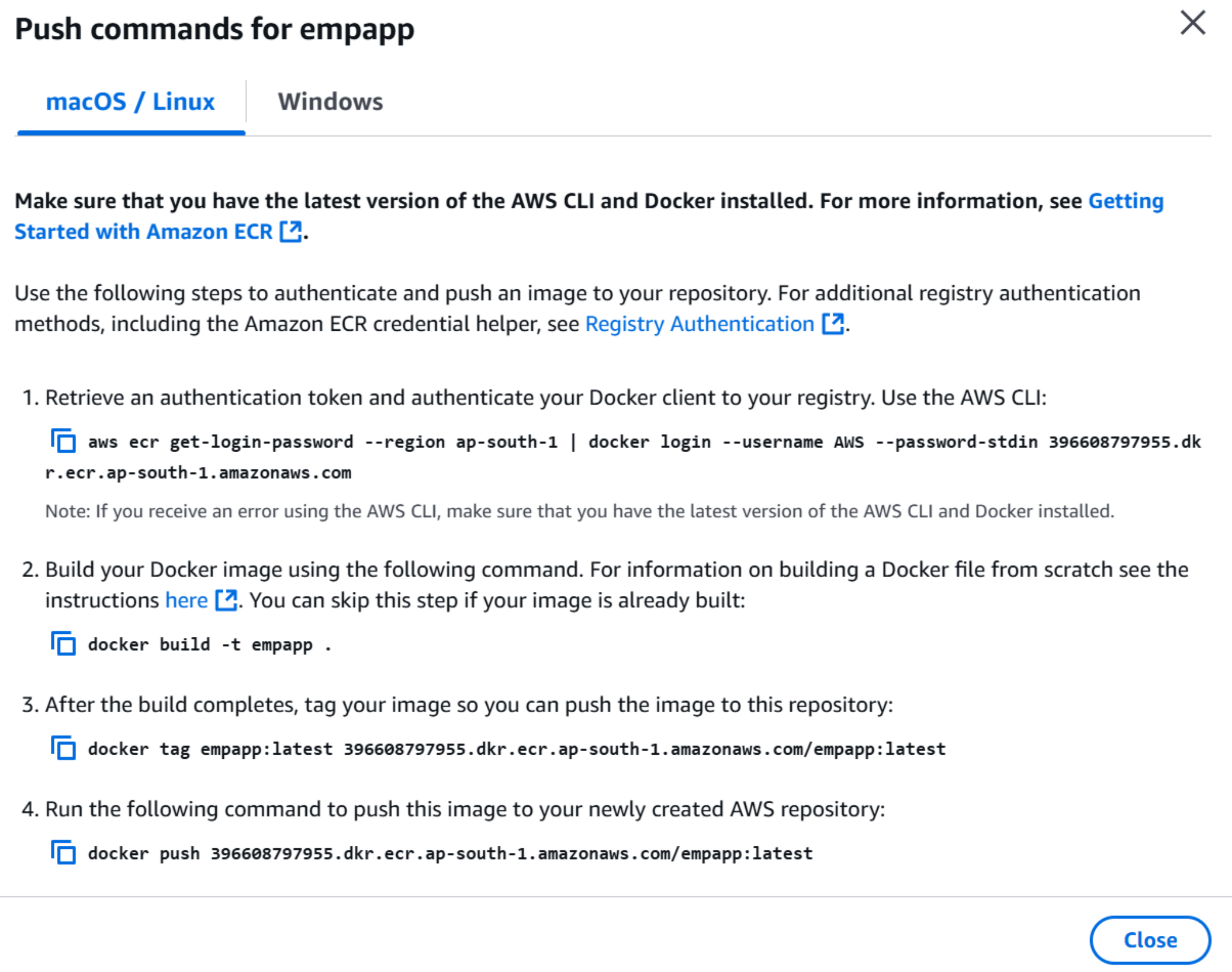
Lab Assignment 7



```
FROM openjdk:17-alpine
MAINTAINER email="rgupta.mtech@gmail.com"
EXPOSE 8080
ADD target/*.jar empapp.jar
ENTRYPOINT ["java","-jar","empapp.jar"]
```

```
PS C:\Users\ashrit.bharadwaj\Desktop\ASHRIT\JavaTraining\aws_projects\employeeappaws> docker container run --name producer -p 8080:8080 -d empapp
1078ae2907ad5530a1ef5a2e299b6688200f463e476dbef35919157e41048c44
PS C:\Users\ashrit.bharadwaj\Desktop\ASHRIT\JavaTraining\aws_projects\employeeappaws> docker image ls
REPOSITORY                             TAG      IMAGE ID       CREATED          SIZE
empapp                                 latest   cfda7d4170c8   2 minutes ago    348MB
pes1ug21cs120ashritbharadwaj/bookapp   1.1      bfb841b7a363   12 days ago      527MB
paketobuildpacks/run-jammy-base        latest   73599ad5768a   2 weeks ago      121MB
hello-world                            latest   74cc54e27dc4   5 weeks ago      10.1kB
```



Push commands for empapp

macOS / Linux | Windows

Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see Getting Started with Amazon ECR.

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see Registry Authentication.

1. Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS CLI:

```
aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 396608797955.dkr.ecr.ap-south-1.amazonaws.com
```

Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.

2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions here. You can skip this step if your image is already built:

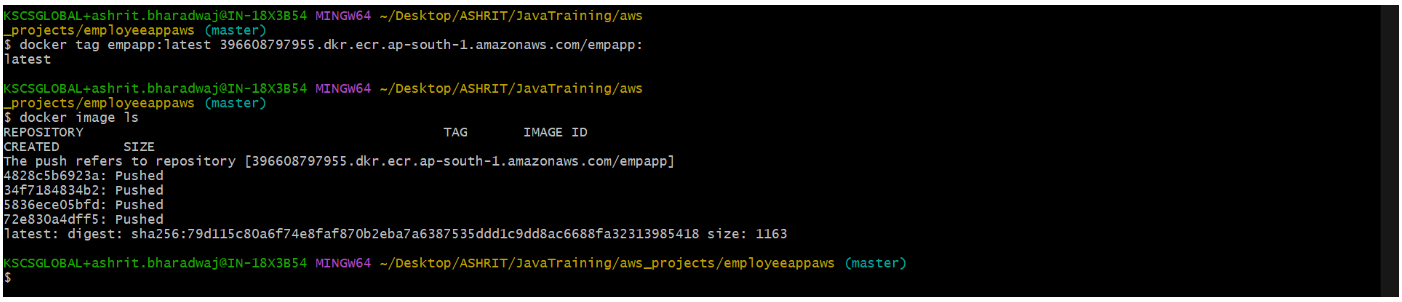
```
docker build -t empapp .
```

3. After the build completes, tag your image so you can push the image to this repository:

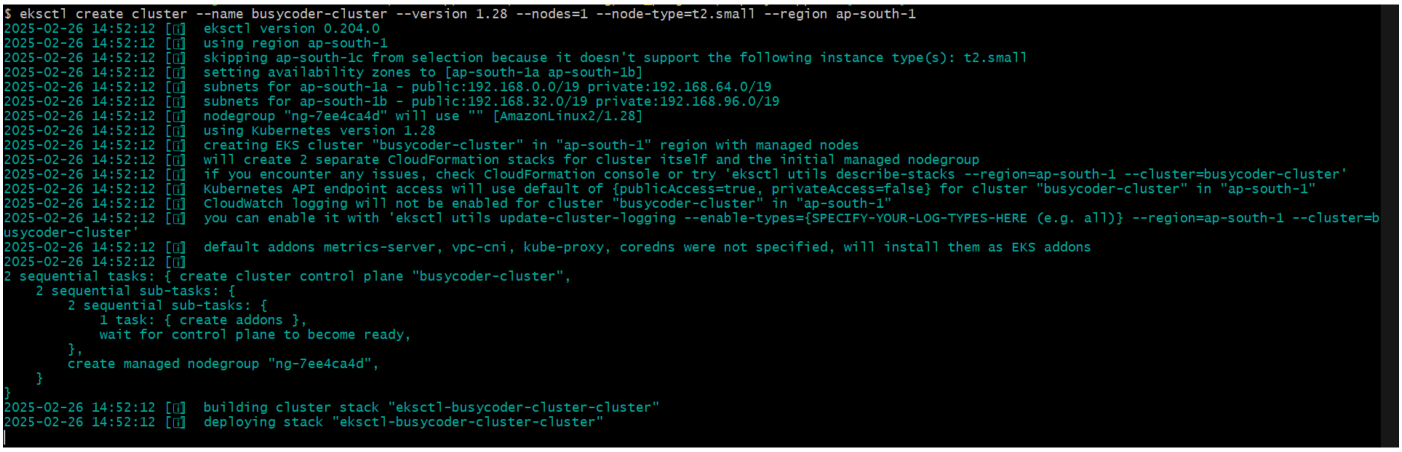
```
docker tag empapp:latest 396608797955.dkr.ecr.ap-south-1.amazonaws.com/empapp:latest
```

4. Run the following command to push this image to your newly created AWS repository:

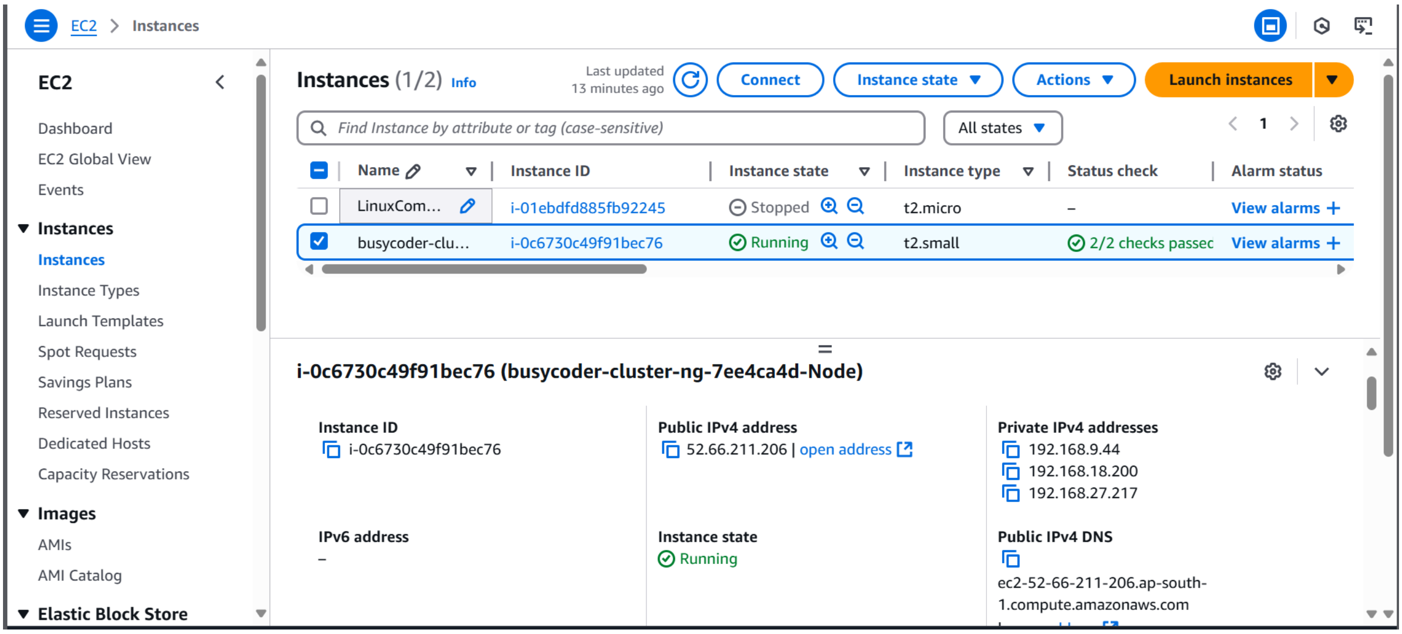
```
docker push 396608797955.dkr.ecr.ap-south-1.amazonaws.com/empapp:latest
```



```
KSCSGLOBAL+ashrit.bharadwaj@IN-18X3B54 MINGW64 ~/Desktop/ASHRIT/JavaTraining/aws
_projects/employeeappaws (master)
$ docker tag empapp:latest 396608797955.dkr.ecr.ap-south-1.amazonaws.com/empapp:
latest

KSCSGLOBAL+ashrit.bharadwaj@IN-18X3B54 MINGW64 ~/Desktop/ASHRIT/JavaTraining/aws
_projects/employeeappaws (master)
$ docker image ls
REPOSITORY                                                    TAG       IMAGE ID
CREATED        SIZE
The push refers to repository [396608797955.dkr.ecr.ap-south-1.amazonaws.com/empapp]
4828c5b6923a: Pushed
34f7184834b2: Pushed
5836ece05bfd: Pushed
72e830a4dff5: Pushed
latest: digest: sha256:79d115c80a6f74e8faf870b2eba7a6387535ddd1c9dd8ac6688fa32313985418 size: 1163

KSCSGLOBAL+ashrit.bharadwaj@IN-18X3B54 MINGW64 ~/Desktop/ASHRIT/JavaTraining/aws_projects/employeeappaws (master)
$
```



```
$ eksctl create cluster --name busycoder-cluster --version 1.28 --nodes=1 --node-type=t2.small --region ap-south-1
2025-02-26 14:52:12 [ℹ]  eksctl version 0.204.0
2025-02-26 14:52:12 [ℹ]  using region ap-south-1
2025-02-26 14:52:12 [ℹ]  skipping ap-south-1c from selection because it doesn't support the following instance type(s): t2.small
2025-02-26 14:52:12 [ℹ]  setting availability zones to [ap-south-1a ap-south-1b]
2025-02-26 14:52:12 [ℹ]  subnets for ap-south-1a - public:192.168.0.0/19 private:192.168.64.0/19
2025-02-26 14:52:12 [ℹ]  subnets for ap-south-1b - public:192.168.32.0/19 private:192.168.96.0/19
2025-02-26 14:52:12 [ℹ]  nodegroup "ng-7ee4ca4d" will use "" [AmazonLinux2/1.28]
2025-02-26 14:52:12 [ℹ]  using Kubernetes version 1.28
2025-02-26 14:52:12 [ℹ]  creating EKS cluster "busycoder-cluster" in "ap-south-1" region with managed nodes
2025-02-26 14:52:12 [ℹ]  will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-02-26 14:52:12 [ℹ]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=busycoder-cluster'
2025-02-26 14:52:12 [ℹ]  Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "busycoder-cluster" in "ap-south-1"
2025-02-26 14:52:12 [ℹ]  CloudWatch logging will not be enabled for cluster "busycoder-cluster" in "ap-south-1"
2025-02-26 14:52:12 [ℹ]  you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=b
usycoder-cluster'
2025-02-26 14:52:12 [ℹ]  default addons metrics-server, vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
2025-02-26 14:52:12 [ℹ]
2 sequential tasks: { create cluster control plane "busycoder-cluster",
    2 sequential sub-tasks: {
        2 sequential sub-tasks: {
            1 task: { create addons },
            wait for control plane to become ready,
        },
        create managed nodegroup "ng-7ee4ca4d",
    }
}
2025-02-26 14:52:12 [ℹ]  building cluster stack "eksctl-busycoder-cluster-cluster"
2025-02-26 14:52:12 [ℹ]  deploying stack "eksctl-busycoder-cluster-cluster"
```



Instances (1/2) Info

Last updated 13 minutes ago

Find Instance by attribute or tag (case-sensitive)

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☐	LinuxCom...	i-01ebdfd885fb92245	Stopped	t2.micro	–	View alarms +
☑	busycoder-clu...	i-0c6730c49f91bec76	Running	t2.small	2/2 checks passec	View alarms +

i-0c6730c49f91bec76 (busycoder-cluster-ng-7ee4ca4d-Node)

Instance ID
i-0c6730c49f91bec76

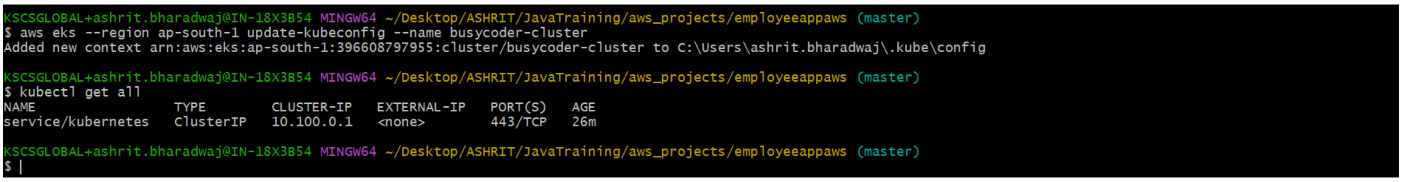
Public IPv4 address
52.66.211.206 | open address

Private IPv4 addresses
192.168.9.44
192.168.18.200
192.168.27.217

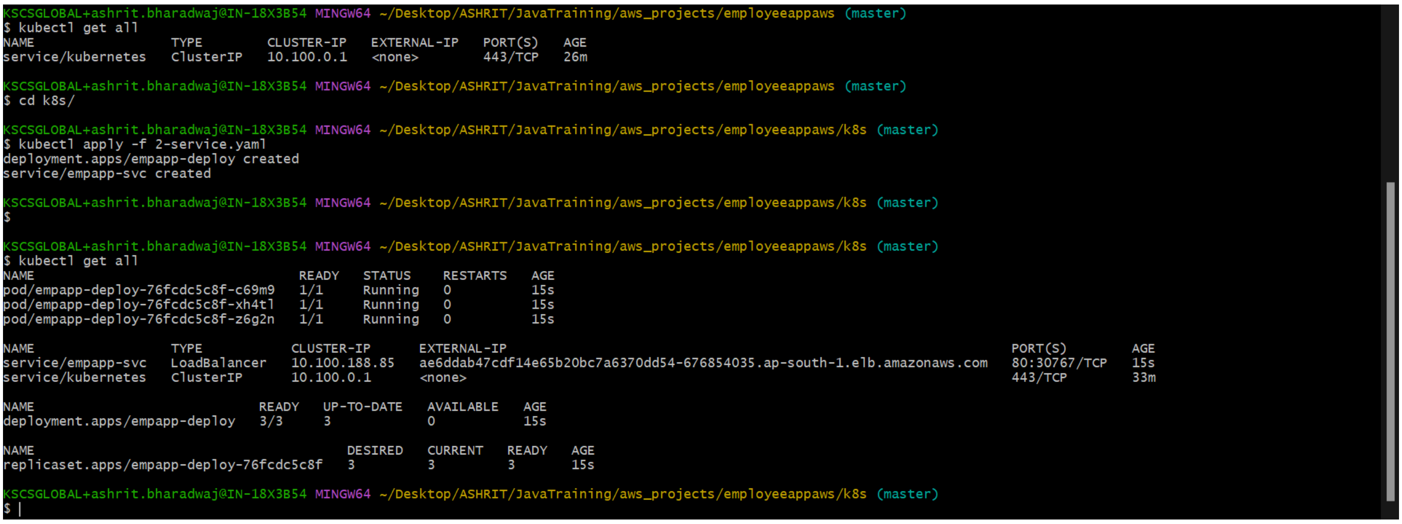
IPv6 address
–

Instance state
Running

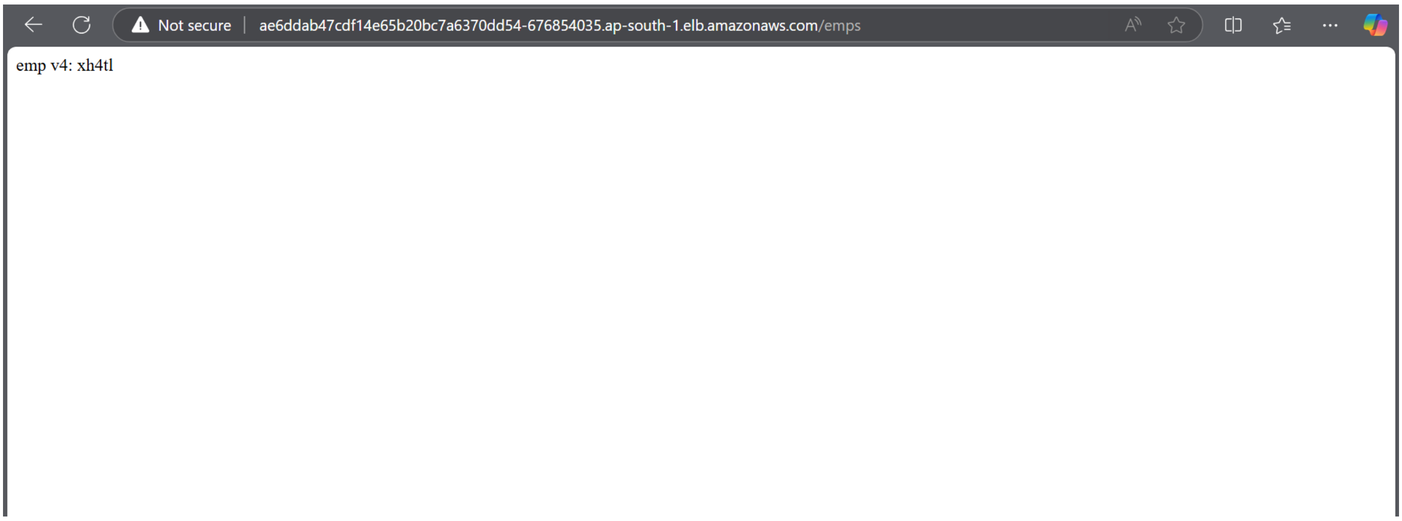
Public IPv4 DNS
ec2-52-66-211-206.ap-south-1.compute.amazonaws.com



```
KSCSGLOBAL+ashrit.bharadwaj@IN-18X3B54 MINGW64 ~/Desktop/ASHRIT/JavaTraining/aws_projects/employeeappaws (master)
$ aws eks --region ap-south-1 update-kubeconfig --name busycoder-cluster
Added new context arn:aws:eks:ap-south-1:396608797955:cluster/busycoder-cluster to C:\Users\ashrit.bharadwaj\.kube\config

KSCSGLOBAL+ashrit.bharadwaj@IN-18X3B54 MINGW64 ~/Desktop/ASHRIT/JavaTraining/aws_projects/employeeappaws (master)
$ kubectl get all
NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.100.0.1   <none>        443/TCP   26m

KSCSGLOBAL+ashrit.bharadwaj@IN-18X3B54 MINGW64 ~/Desktop/ASHRIT/JavaTraining/aws_projects/employeeappaws (master)
$ |
```

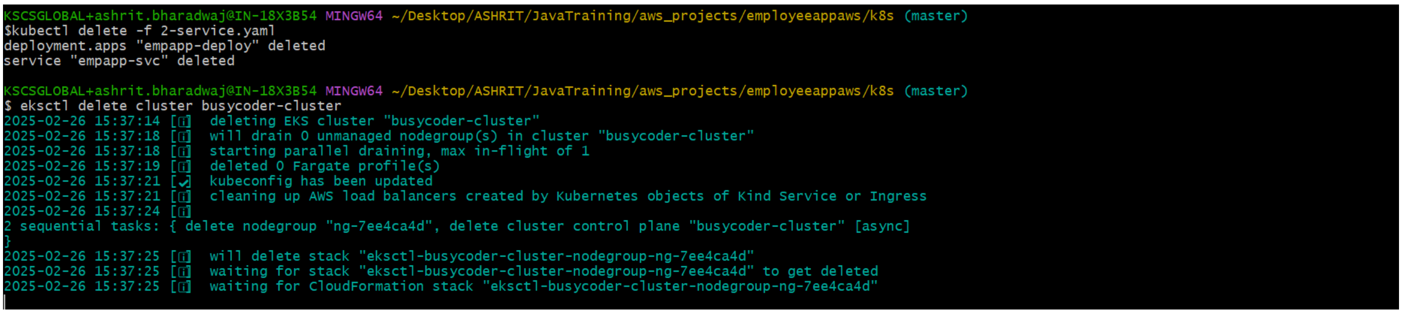


```
KSCSGLOBAL+ashrit.bharadwaj@IN-18X3B54 MINGW64 ~/Desktop/ASHRIT/JavaTraining/aws_projects/employeeappaws (master)
$ kubectl get all
NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.100.0.1   <none>        443/TCP   26m

KSCSGLOBAL+ashrit.bharadwaj@IN-18X3B54 MINGW64 ~/Desktop/ASHRIT/JavaTraining/aws_projects/employeeappaws (master)
$ cd k8s/

KSCSGLOBAL+ashrit.bharadwaj@IN-18X3B54 MINGW64 ~/Desktop/ASHRIT/JavaTraining/aws_projects/employeeappaws/k8s (master)
$ kubectl apply -f 2-service.yaml
deployment.apps/empapp-deploy created
service/empapp-svc created

KSCSGLOBAL+ashrit.bharadwaj@IN-18X3B54 MINGW64 ~/Desktop/ASHRIT/JavaTraining/aws_projects/employeeappaws/k8s (master)
$

KSCSGLOBAL+ashrit.bharadwaj@IN-18X3B54 MINGW64 ~/Desktop/ASHRIT/JavaTraining/aws_projects/employeeappaws/k8s (master)
$ kubectl get all
NAME                                 READY   STATUS    RESTARTS   AGE
pod/empapp-deploy-76fcdc5c8f-c69m9   1/1     Running   0          15s
pod/empapp-deploy-76fcdc5c8f-xh4tl   1/1     Running   0          15s
pod/empapp-deploy-76fcdc5c8f-z6g2n   1/1     Running   0          15s

NAME                 TYPE           CLUSTER-IP      EXTERNAL-IP                                                                PORT(S)        AGE
service/empapp-svc   LoadBalancer   10.100.188.85   ae6ddab47cdf14e65b20bc7a6370dd54-676854035.ap-south-1.elb.amazonaws.com   80:30767/TCP   15s
service/kubernetes   ClusterIP      10.100.0.1      <none>                                                                     443/TCP        33m

NAME                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/empapp-deploy   3/3     3            0           15s

NAME                                       DESIRED   CURRENT   READY   AGE
replicaset.apps/empapp-deploy-76fcdc5c8f   3         3         3       15s

KSCSGLOBAL+ashrit.bharadwaj@IN-18X3B54 MINGW64 ~/Desktop/ASHRIT/JavaTraining/aws_projects/employeeappaws/k8s (master)
$ |
```



ae6ddab47cdf14e65b20bc7a6370dd54-676854035.ap-south-1.elb.amazonaws.com/emps

emp v4: xh4tl



```
KSCSGLOBAL+ashrit.bharadwaj@IN-18X3B54 MINGW64 ~/Desktop/ASHRIT/JavaTraining/aws_projects/employeeappaws/k8s (master)
$kubectl delete -f 2-service.yaml
deployment.apps "empapp-deploy" deleted
service "empapp-svc" deleted

KSCSGLOBAL+ashrit.bharadwaj@IN-18X3B54 MINGW64 ~/Desktop/ASHRIT/JavaTraining/aws_projects/employeeappaws/k8s (master)
$ eksctl delete cluster busycoder-cluster
2025-02-26 15:37:14 [ℹ]  deleting EKS cluster "busycoder-cluster"
2025-02-26 15:37:18 [ℹ]  will drain 0 unmanaged nodegroup(s) in cluster "busycoder-cluster"
2025-02-26 15:37:18 [ℹ]  starting parallel draining, max in-flight of 1
2025-02-26 15:37:19 [ℹ]  deleted 0 Fargate profile(s)
2025-02-26 15:37:21 [✔]  kubeconfig has been updated
2025-02-26 15:37:21 [ℹ]  cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
2025-02-26 15:37:24 [ℹ]
2 sequential tasks: { delete nodegroup "ng-7ee4ca4d", delete cluster control plane "busycoder-cluster" [async]
}
2025-02-26 15:37:25 [ℹ]  will delete stack "eksctl-busycoder-cluster-nodegroup-ng-7ee4ca4d"
2025-02-26 15:37:25 [ℹ]  waiting for stack "eksctl-busycoder-cluster-nodegroup-ng-7ee4ca4d" to get deleted
2025-02-26 15:37:25 [ℹ]  waiting for CloudFormation stack "eksctl-busycoder-cluster-nodegroup-ng-7ee4ca4d"
|
```