LAB ASSIGNMENTS 7: AWS EKS

What is Amazon EKS?

Amazon EKS is a managed service that is used to run Kubernetes on AWS.

Using EKS users don't have to maintain a Kubernetes control plan on their own.

It is used to automate the deployment, scaling, and maintenance of the containerized application.

It works with most operating systems.

Benefits of Amazon EKS

Normally time-consuming tasks, such as constructing the Kubernetes master cluster and setting service discovery, Kubernetes primitives, and networking, are handled by AWS EKS.

Existing tools will almost certainly work through EKS with minimal if any, modifications.

The Kubernetes control plane, comprising the backend persistence layer and API servers,

is provisioned and scaled across multiple AWS availability zones using Amazon EKS,

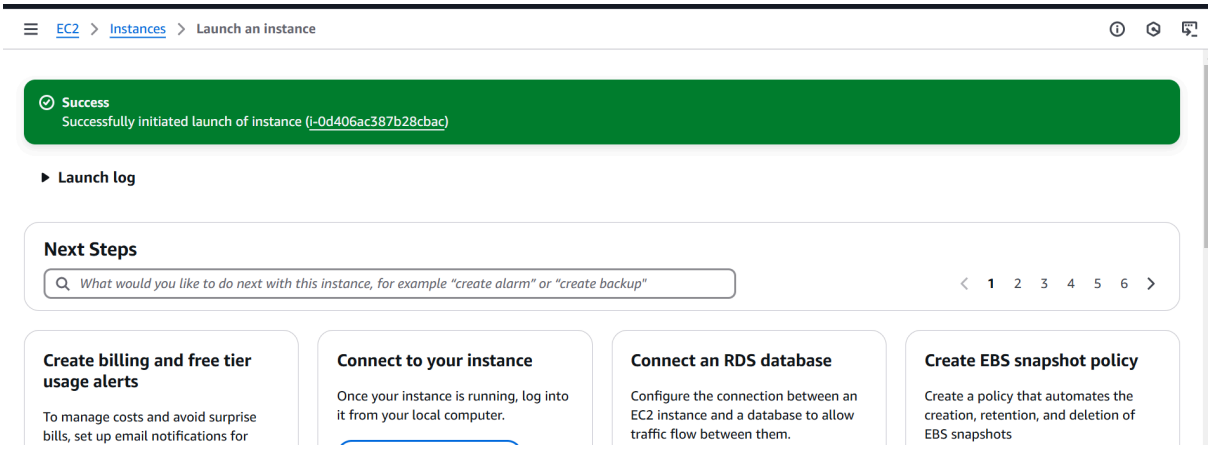
resulting in high availability and the elimination of a single point of failure.

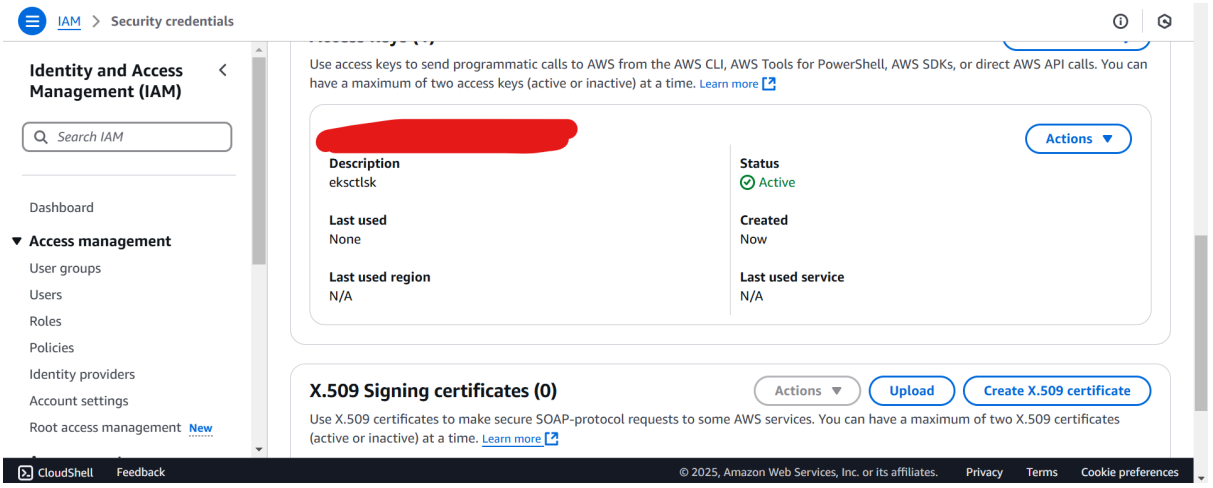
Unhealthy control plane nodes are identified and replaced, and control plane patching is given.As a result, an AWS-managed Kubernetes cluster that can endure the loss of an availability zone has been created.

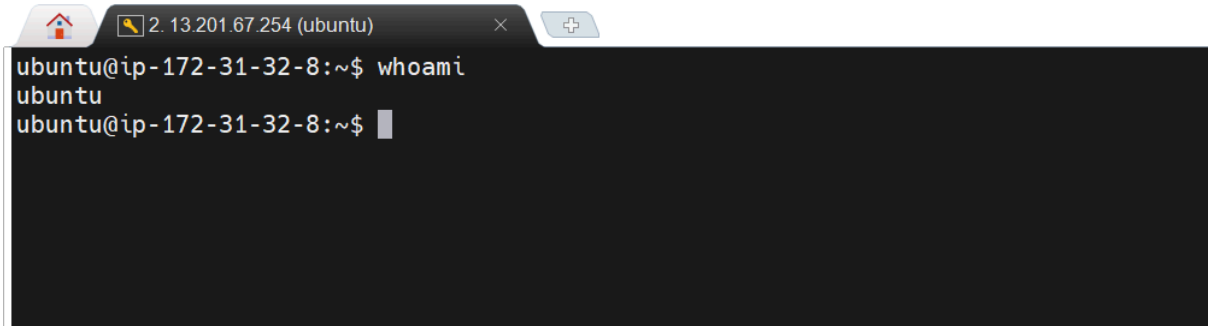
Amazon EKS cluster setup:

1. using UI console
2. using Terraform
3. Using eksctl

In this lab you need to configure EKS using eksctl


EC2 > Instances > Launch an instance
Success
Successfully initiated launch of instance (i-0d406ac387b28cbac)
Launch log
Next Steps
What would you like to do next with this instance, for example "create alarm" or "create backup"
1 2 3 4 5 6
Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for
Connect to your instance
Once your instance is running, log into it from your local computer.
Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots


IAM > Security credentials
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more
Actions
Description
eksctlsk
Status
Active
Last used
None
Created
Now
Last used region
N/A
Last used service
N/A
X.509 Signing certificates (0)
Actions
Upload
Create X.509 certificate
Use X.509 certificates to make secure SOAP-protocol requests to some AWS services. You can have a maximum of two X.509 certificates (active or inactive) at a time. Learn more
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


2. 13.201.67.254 (ubuntu)
ubuntu@ip-172-31-32-8:~$ whoami
ubuntu
ubuntu@ip-172-31-32-8:~$

```
ubuntu@ip-172-31-32-8:~$ whoami
ubuntu
ubuntu@ip-172-31-32-8:~$ wget https://github.com/eksctl-io/eksctl/releases/download/v0.204.0/eksctl_Linux_amd64.tar.gz
--2025-02-27 14:34:37--  https://github.com/eksctl-io/eksctl/releases/download/v0.204.0/eksctl_Linux_amd64.tar.gz
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/134539560/3ae52b00-ae5a-474f-b65a-85b8ae7592f0?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250227%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250227T143437Z&X-Amz-Expires=300&X-Amz-Signature=c11f4cb2e2798bb4e1a749ba2213f7c234ca1acc417ac928f52c4b151d8e7feb&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Deksctl_Linux_amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2025-02-27 14:34:37--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/134539560/3ae52b00-ae5a-474f-b65a-85b8ae7592f0?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20250227%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20250227T143437Z&X-Amz-Expires=300&X-Amz-Signature=c11f4cb2e2798bb4e1a749ba2213f7c234ca1acc417ac928f52c4b151d8e7feb&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Deksctl_Linux_amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 38989216 (37M) [application/octet-stream]
Saving to: 'eksctl_Linux_amd64.tar.gz'

eksctl_Linux_amd64.tar.gz        100%[=====================================================>]  37.18M  21.3MB/s    in 1.7s

2025-02-27 14:34:40 (21.3 MB/s) - 'eksctl_Linux_amd64.tar.gz' saved [38989216/38989216]

ubuntu@ip-172-31-32-8:~$ tar -xvzf eksctl_Linux_amd64.tar.gz
eksctl
ubuntu@ip-172-31-32-8:~$ sudo mv eksctl /usr/local/bin
ubuntu@ip-172-31-32-8:~$ eksctl version
0.204.0
ubuntu@ip-172-31-32-8:~$
```

```
ubuntu@ip-172-31-32-8:~$ mkdir demo
ubuntu@ip-172-31-32-8:~$ cd demo
ubuntu@ip-172-31-32-8:~/demo$ eksctl version
0.204.0
ubuntu@ip-172-31-32-8:~/demo$
```

```
  inflating: aws/dist/docutils/writers/html5_polyglot/tuftig.css
  inflating: aws/dist/docutils/writers/html5_polyglot/minimal.css
  inflating: aws/dist/docutils/writers/html5_polyglot/responsive.css
  inflating: aws/dist/docutils/writers/html5_polyglot/math.css
  inflating: aws/dist/docutils/writers/pep_html/template.txt
  inflating: aws/dist/docutils/writers/pep_html/pep.css
  inflating: aws/dist/docutils/writers/odf_odt/styles.odt
  inflating: aws/dist/docutils/writers/html4css1/template.txt
  inflating: aws/dist/docutils/writers/html4css1/html4css1.css
ubuntu@ip-172-31-32-8:~/demo$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
ubuntu@ip-172-31-32-8:~/demo$ aws --version
aws-cli/2.24.13 Python/3.12.9 Linux/6.8.0-1021-aws exe/x86_64.ubuntu.24
ubuntu@ip-172-31-32-8:~/demo$
```

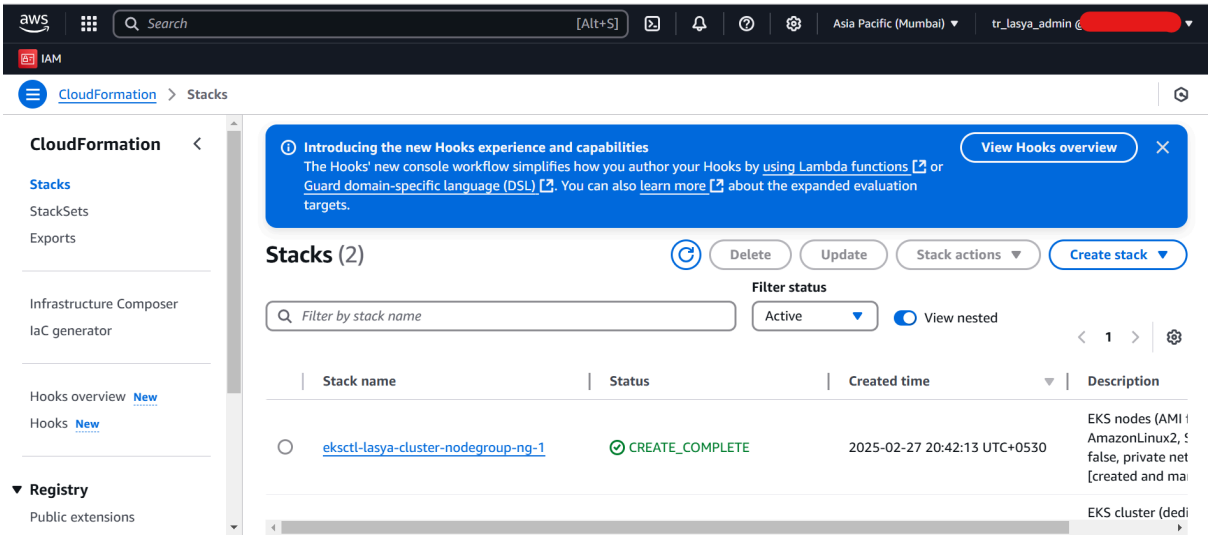
```
ubuntu@ip-172-31-32-8:~/demo$ aws --version
aws-cli/2.24.13 Python/3.12.9 Linux/6.8.0-1021-aws exe/x86_64.ubuntu.24
ubuntu@ip-172-31-32-8:~/demo$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]:
Default output format [None]:
ubuntu@ip-172-31-32-8:~/demo$
```

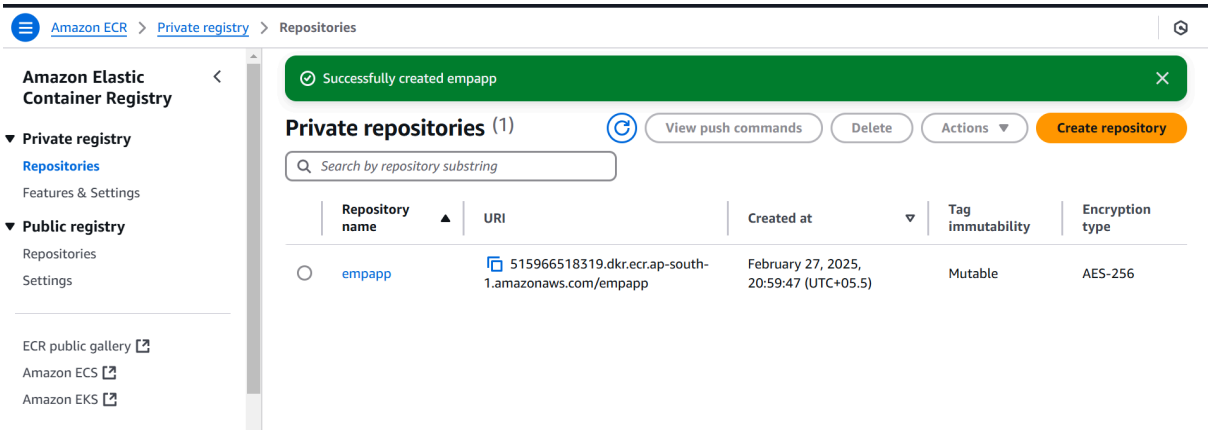
```
ubuntu@ip-172-31-32-8:~/demo$ eksctl version
0.204.0
ubuntu@ip-172-31-32-8:~/demo$ aws --version
aws-cli/2.24.13 Python/3.12.9 Linux/6.8.0-1021-aws exe/x86_64.ubuntu.24
ubuntu@ip-172-31-32-8:~/demo$ kubectl version
Client Version: v1.32.0-eks-5ca49cb
Kustomize Version: v5.5.0
The connection to the server localhost:8080 was refused - did you specify the right host or port?
ubuntu@ip-172-31-32-8:~/demo$
```

```
ubuntu@ip-172-31-32-8:~/demo$ vim eks.yaml
ubuntu@ip-172-31-32-8:~/demo$ cat eks.yaml
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig

metadata:
  name: lasya-cluster
  region: ap-south-1

nodeGroups:
  - name: ng-1
    instanceType: t2.micro
    desiredCapacity: 2

ubuntu@ip-172-31-32-8:~/demo$
```

```
2025-02-27 15:12:13 [ℹ]  waiting for CloudFormation stack "eksctl-lasya-cluster-nodegroup-ng-1"
2025-02-27 15:12:43 [ℹ]  waiting for CloudFormation stack "eksctl-lasya-cluster-nodegroup-ng-1"
2025-02-27 15:13:26 [ℹ]  waiting for CloudFormation stack "eksctl-lasya-cluster-nodegroup-ng-1"
2025-02-27 15:15:05 [ℹ]  waiting for CloudFormation stack "eksctl-lasya-cluster-nodegroup-ng-1"
2025-02-27 15:15:05 [ℹ]  waiting for the control plane to become ready
2025-02-27 15:15:05 [✔]  saved kubeconfig as "/home/ubuntu/.kube/config"
2025-02-27 15:15:05 [ℹ]  no tasks
2025-02-27 15:15:05 [✔]  all EKS cluster resources for "lasya-cluster" have been created
2025-02-27 15:15:05 [ℹ]  nodegroup "ng-1" has 2 node(s)
2025-02-27 15:15:05 [ℹ]  node "ip-192-168-2-80.ap-south-1.compute.internal" is ready
2025-02-27 15:15:05 [ℹ]  node "ip-192-168-55-176.ap-south-1.compute.internal" is ready
2025-02-27 15:15:05 [ℹ]  waiting for at least 2 node(s) to become ready in "ng-1"
2025-02-27 15:15:05 [ℹ]  nodegroup "ng-1" has 2 node(s)
2025-02-27 15:15:05 [ℹ]  node "ip-192-168-2-80.ap-south-1.compute.internal" is ready
2025-02-27 15:15:05 [ℹ]  node "ip-192-168-55-176.ap-south-1.compute.internal" is ready
2025-02-27 15:15:05 [✔]  created 1 nodegroup(s) in cluster "lasya-cluster"
2025-02-27 15:15:06 [ℹ]  kubectl command should work with "/home/ubuntu/.kube/config", try 'kubectl get nodes'
2025-02-27 15:15:06 [✔]  EKS cluster "lasya-cluster" in "ap-south-1" region is ready
ubuntu@ip-172-31-32-8:~/demo$
```

```
ubuntu@ip-172-31-32-8:~/demo$ eksctl get cluster --region ap-south-1
NAME            REGION          EKSCTL CREATED
lasya-cluster   ap-south-1      True
ubuntu@ip-172-31-32-8:~/demo$ eksctl get nodegroup --cluster lasya-cluster --region ap-south-1
CLUSTER         NODEGROUP       STATUS          CREATED                 MIN SIZE        MAX SIZE        DESIRED CAPACITY        INS
TANCE TYPE      IMAGE ID                ASG NAME                                                        TYPE
lasya-cluster   ng-1            CREATE_COMPLETE 2025-02-27T15:12:13Z    2               2               2                       t2.
micro   ami-0c5c573feb08cfa4d   eksctl-lasya-cluster-nodegroup-ng-1-NodeGroup-4HeSarOSMofj      unmanaged
ubuntu@ip-172-31-32-8:~/demo$ kubectl get nodes
NAME                                            STATUS   ROLES    AGE     VERSION
ip-192-168-2-80.ap-south-1.compute.internal     Ready    <none>   4m14s   v1.30.9-eks-5d632ec
ip-192-168-55-176.ap-south-1.compute.internal   Ready    <none>   4m34s   v1.30.9-eks-5d632ec
ubuntu@ip-172-31-32-8:~/demo$
```



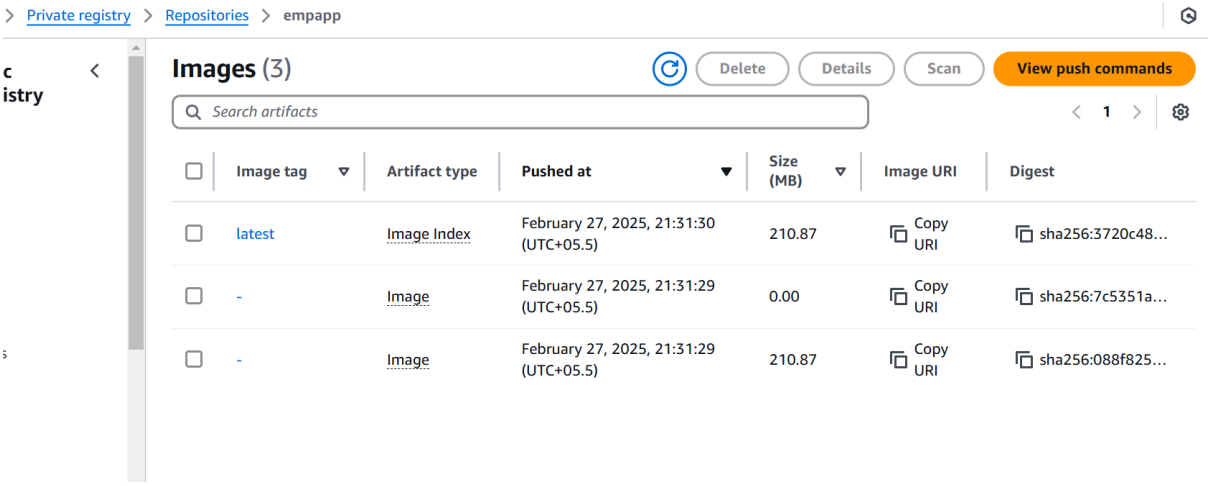



```
KSCSGLOBAL+lasya.talasila@IN-D71DG74 MINGW64 ~/OneDrive - Körber Supply Chain So
ftware/Desktop/project2/employeeappaws
$ aws ecr get-login-password --region ap-south-1 | docker login --username AWS -
-password-stdin 515966518319.dkr.ecr.ap-south-1.amazonaws.com
Login Succeeded
```

```
ftware/Desktop/project2/employeeappaws
$ docker images
REPOSITORY                                                      TAG       IMAGE ID
       CREATED          SIZE
empapp                                                          latest    3720c48d
54e5   14 seconds ago   562MB
bookapp                                                         1.1       ee3d309c
890f   11 days ago      822MB
bookapp                                                         latest    ee3d309c
890f   11 days ago      822MB
pes2ug21cs254lasyatalasilabookapp                               1.1       ee3d309c
```

```
KSCSGLOBAL+lasya.talasila@IN-D71DG74 MINGW64 ~/OneDrive - Körber Supply Chain So
ftware/Desktop/project2/employeeappaws
$ docker tag empapp:latest 515966518319.dkr.ecr.ap-south-1.amazonaws.com/empapp:latest

KSCSGLOBAL+lasya.talasila@IN-D71DG74 MINGW64 ~/OneDrive - Körber Supply Chain Software/Desktop/project2/employeeappaws
$ docker images
REPOSITORY                                                TAG       IMAGE ID       CREATED          SIZE
515966518319.dkr.ecr.ap-south-1.amazonaws.com/empapp      latest    3720c48d54e5   2 minutes ago    562MB
empapp                                                    latest    3720c48d54e5   2 minutes ago    562MB
bookapp                                                   1.1       ee3d309c890f   11 days ago      822MB
bookapp                                                   latest    ee3d309c890f   11 days ago      822MB
pes2ug21cs254lasyatalasilabookapp                         1.1       ee3d309c890f   11 days ago      822MB
pes2ug21cs254lasyatalasila/bookapp                        1.1       ee3d309c890f   11 days ago      822MB
pes2ug21cs254lasyatalasila/bookapp                        1.2       ee3d309c890f   11 days ago      822MB
pes2ug21cs254lasyatalasila/bookapp                        1.3       ee3d309c890f   11 days ago      822MB
docker_mysql-bookstore-container                          1.1       69beca87614e   11 days ago      872MB
docker_mysql-bookstore-container                          latest    69beca87614e   11 days ago      872MB
pes2ug21cs254lasyatalasila/docker_mysql-bookstore-container   1.1       69beca87614e   11 days ago      872MB
<none>                                                    <none>    4d13cc3ba103   11 days ago      822MB
docker_network-currency-conversion                        latest    696445b372f8   2 weeks ago      894MB
```

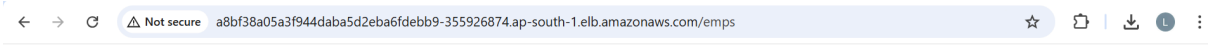
```
KSCSGLOBAL+lasya.talasila@IN-D71DG74 MINGW64 ~/OneDrive - Körber Supply Chain Software/Desktop/project2/employeeappaws
$ docker push 515966518319.dkr.ecr.ap-south-1.amazonaws.com/empapp:latest
The push refers to repository [515966518319.dkr.ecr.ap-south-1.amazonaws.com/empapp]
9e54a387bec9: Waiting
35e0d1fc6dfd: Waiting
5843afab3874: Waiting
53c9466125e4: Waiting
d8d715783b80: Waiting
9e54a387bec9: Waiting
35e0d1fc6dfd: Waiting
5843afab3874: Waiting
```




```
KSCSGLOBAL+lasya.talasila@IN-D71DG74 MINGW64 ~/OneDrive - Körber Supply Chain Software/Desktop/project2/employeeappaws
$ aws eks --region ap-south-1 update-kubeconfig --name lasya-cluster
Updated context arn:aws:eks:ap-south-1:515966518319:cluster/lasya-cluster in C:\Users\lasya.talasila\.kube\config

KSCSGLOBAL+lasya.talasila@IN-D71DG74 MINGW64 ~/OneDrive - Körber Supply Chain Software/Desktop/project2/employeeappaws
$ ls
Dockerfile  k8s/  mvnw*  mvnw.cmd  pom.xml  src/  target/

KSCSGLOBAL+lasya.talasila@IN-D71DG74 MINGW64 ~/OneDrive - Körber Supply Chain Software/Desktop/project2/employeeappaws
$ cd k8s

KSCSGLOBAL+lasya.talasila@IN-D71DG74 MINGW64 ~/OneDrive - Körber Supply Chain Software/Desktop/project2/employeeappaws/k8s
$ ls
k8s.yaml

KSCSGLOBAL+lasya.talasila@IN-D71DG74 MINGW64 ~/OneDrive - Körber Supply Chain Software/Desktop/project2/employeeappaws/k8s
$ cal k8s.yaml
bash: cal: command not found

KSCSGLOBAL+lasya.talasila@IN-D71DG74 MINGW64 ~/OneDrive - Körber Supply Chain Software/Desktop/project2/employeeappaws/k8s
$ cat k8s.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: empapp
spec:
  replicas: 3
  selector:
    matchLabels:
      app: empapp
  template:
    metadata:
      labels:
        app: empapp
    spec:
      containers:
        - name: empapp
          image: 515966518319.dkr.ecr.ap-south-1.amazonaws.com/empapp:latest
          ports:
            - containerPort: 8080

---
apiVersion: v1
kind: Service
metadata:
  name: newapp-service
```

```
    targetPort: 8080
  type: LoadBalancer
KSCSGLOBAL+lasya.talasila@IN-D71DG74 MINGW64 ~/OneDrive - Körber Supply Chain Software/Desktop/project2/employeeappaws/k8s
$ kubectl apply -f k8s.yaml
deployment.apps/empapp created
service/newapp-service created

KSCSGLOBAL+lasya.talasila@IN-D71DG74 MINGW64 ~/OneDrive - Körber Supply Chain Software/Desktop/project2/employeeappaws/k8s
$ kubectl get all
```

```
  type: LoadBalancer

ubuntu@ip-172-31-37-175:~/demo$ kubectl apply -f k8s.yaml
deployment.apps/newapp created
service/newapp-service created
ubuntu@ip-172-31-37-175:~/demo$ kubectl get svc
NAME             TYPE           CLUSTER-IP      EXTERNAL-IP                                                              PORT(S)
      AGE
kubernetes       ClusterIP      10.100.0.1      <none>                                                                   443/TCP
      14m
newapp-service   LoadBalancer   10.100.194.88   a8bf38a05a3f944daba5d2eba6fdebb9-355926874.ap-south-1.elb.amazonaws.com   80:32750/
TCP   16s
ubuntu@ip-172-31-37-175:~/demo$ kubectl get deployment
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
newapp   3/3     3            3           27s
ubuntu@ip-172-31-37-175:~/demo$ kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
newapp-6d99d6d4b8-4b65p   1/1     Running   0          37s
newapp-6d99d6d4b8-h66f9   1/1     Running   0          37s
newapp-6d99d6d4b8-sskfr   1/1     Running   0          37s
ubuntu@ip-172-31-37-175:~/demo$ ^C
ubuntu@ip-172-31-37-175:~/demo$ ^C
ubuntu@ip-172-31-37-175:~/demo$
```

emp v4: 4b65p

```
ubuntu@ip-172-31-37-175:~/demo$ eksctl delete cluster lasya-cluster --region ap-south-1
2025-02-27 19:02:54 [ℹ]  deleting EKS cluster "lasya-cluster"
2025-02-27 19:02:54 [ℹ]  will drain 0 unmanaged nodegroup(s) in cluster "lasya-cluster"
2025-02-27 19:02:54 [ℹ]  starting parallel draining, max in-flight of 1
2025-02-27 19:02:55 [ℹ]  deleted 0 Fargate profile(s)
2025-02-27 19:02:55 [✔]  kubeconfig has been updated
2025-02-27 19:02:55 [ℹ]  cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
2025-02-27 19:03:20 [ℹ]
2 sequential tasks: { delete nodegroup "ng-015cd747", delete cluster control plane "lasya-cluster" [async]
}
2025-02-27 19:03:20 [ℹ]  will delete stack "eksctl-lasya-cluster-nodegroup-ng-015cd747"
2025-02-27 19:03:20 [ℹ]  waiting for stack "eksctl-lasya-cluster-nodegroup-ng-015cd747" to get deleted
2025-02-27 19:03:20 [ℹ]  waiting for CloudFormation stack "eksctl-lasya-cluster-nodegroup-ng-015cd747"
2025-02-27 19:03:50 [ℹ]  waiting for CloudFormation stack "eksctl-lasya-cluster-nodegroup-ng-015cd747"
2025-02-27 19:04:44 [ℹ]  waiting for CloudFormation stack "eksctl-lasya-cluster-nodegroup-ng-015cd747"
2025-02-27 19:06:08 [ℹ]  waiting for CloudFormation stack "eksctl-lasya-cluster-nodegroup-ng-015cd747"
2025-02-27 19:07:24 [ℹ]  waiting for CloudFormation stack "eksctl-lasya-cluster-nodegroup-ng-015cd747"
2025-02-27 19:08:05 [ℹ]  waiting for CloudFormation stack "eksctl-lasya-cluster-nodegroup-ng-015cd747"
2025-02-27 19:09:15 [ℹ]  waiting for CloudFormation stack "eksctl-lasya-cluster-nodegroup-ng-015cd747"
2025-02-27 19:10:24 [ℹ]  waiting for CloudFormation stack "eksctl-lasya-cluster-nodegroup-ng-015cd747"
2025-02-27 19:10:25 [ℹ]  will delete stack "eksctl-lasya-cluster-cluster"
2025-02-27 19:10:25 [✔]  all cluster resources were deleted
ubuntu@ip-172-31-37-175:~/demo$
```