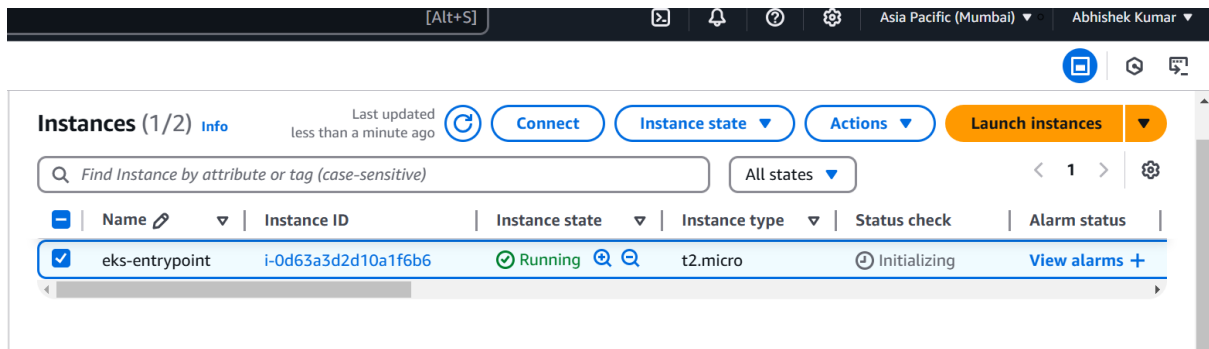


Two-tier Application Deployment on EKS (Elastic Kubernetes Service)

Create an EC2 Instance



Connect to EC2 via SSH

4. Connect to your instance using its Public DNS:
`ec2-3-110-218-38.ap-south-1.compute.amazonaws.com`

Example:
`ssh -i "tws-live-jenkins.pem" ubuntu@ec2-3-110-218-38.ap-south-1.compute.amazonaws.com`

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

[Cancel](#)

```
HP@DESKTOP-J085TTO MINGW64 ~/Downloads
$ ssh -i "tws-live-jenkins.pem" ubuntu@ec2-3-110-218-38.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-3-110-218-38.ap-south-1.compute.amazonaws.com (3.110.218.38)' can't be established.
ED25519 key fingerprint is SHA256:asny19819dx9pPipbBtHP3+pz7Zd5XDQvqAH+udhgc0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-110-218-38.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sun Jan 26 10:49:17 UTC 2025

System load:  0.02          Processes:      105
Usage of /:   24.9% of 6.71GB Users logged in:  0
Memory usage: 20%          IPv4 address for enX0: 172.31.12.203
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-12-203:~$
```

Update the Instance

```
ubuntu@ip-172-31-12-203:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [5260 kB]
```

Install AWS CLI

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
sudo apt install unzip
unzip awscliv2.zip
sudo ./aws/install -i /usr/local/aws-cli -b /usr/local/bin --update
```

```
ubuntu@ip-172-31-12-203:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
sudo apt install unzip
unzip awscliv2.zip
sudo ./aws/install -i /usr/local/aws-cli -b /usr/local/bin --update
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 64.5M  100 64.5M    0     0  100M      0  --:--:-- --:--:-- --:--:--  100M
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
0 upgraded, 1 newly installed, 0 to remove and 27 not upgraded.
Need to get 174 kB of archives.
After this operation, 384 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/main amd64 unzip amd64 6.0-28ubuntu4.1 [174 kB]
Fetched 174 kB in 0s (9940 kB/s)
Selecting previously unselected package unzip.
(Reading database ... 70610 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-28ubuntu4.1_amd64.deb ...
Unpacking unzip (6.0-28ubuntu4.1) ...
Setting up unzip (6.0-28ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

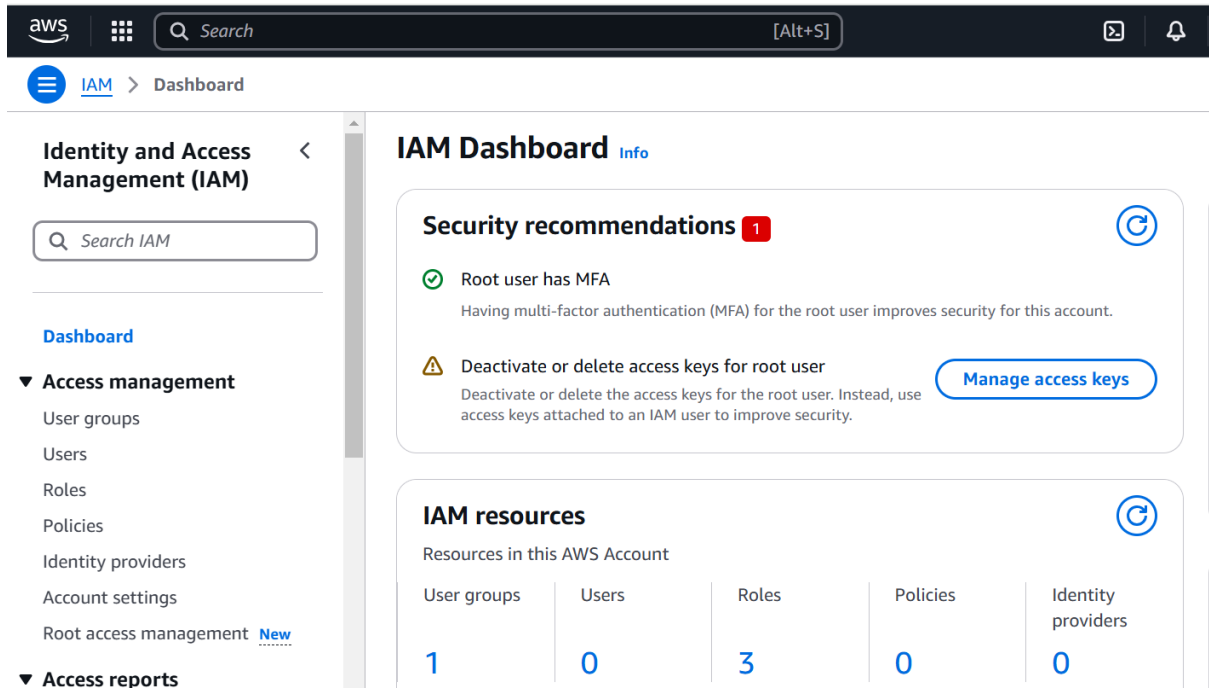
It will Install AWS Command line Interface.

Check the installed version

```
aws --version
```

```
ubuntu@ip-172-31-12-203:~$ aws --version
aws-cli/2.23.6 python/3.12.6 Linux/6.8.0-1021-aws exe/x86_64.ubuntu.24
ubuntu@ip-172-31-12-203:~$
```

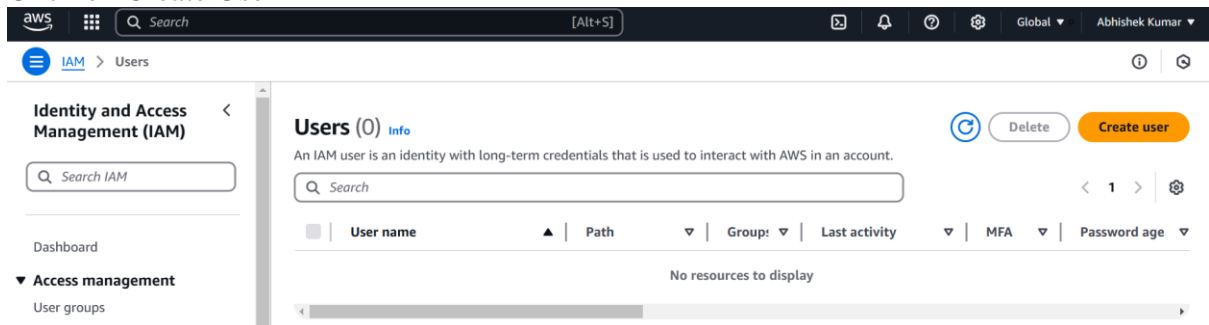
Create IAM Role



The screenshot shows the AWS IAM Dashboard. The left sidebar contains the 'Identity and Access Management (IAM)' menu with options like 'Dashboard', 'Access management', and 'Access reports'. The main content area is titled 'IAM Dashboard' and includes a 'Security recommendations' section with one recommendation: 'Root user has MFA'. Below this is the 'IAM resources' section, which displays a table of resources in the AWS account.

User groups	Users	Roles	Policies	Identity providers
1	0	3	0	0

Click on Create User



The screenshot shows the AWS IAM 'Users' page. The left sidebar is the same as the previous screenshot. The main content area is titled 'Users (0)' and includes a 'Create user' button. Below the button is a table with columns for 'User name', 'Path', 'Group', 'Last activity', 'MFA', and 'Password age'. The table is currently empty, displaying 'No resources to display'.

Specify user details

User details

User name

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☐ Provide user access to the AWS Management Console - *optional*

If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Info If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

[Cancel](#)[Next](#)

Click on next

Attach the required access policy for the user.

☐ Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.

☒ Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1319)

Create policy

Choose one or more policies to attach to your new user.

Filter by Type

Search

All types

< 1 2 3 4 5 6 7 ... 66 >

	Policy name	Type	Attached entities
<input type="checkbox"/>	AccessAnalyzerServiceRole...	AWS managed	0
<input checked="" type="checkbox"/>	AdministratorAccess	AWS managed - job function	1
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	0

Click on Next

IAM > Users > Create user

Step 1: Specify user details
Step 2: Set permissions
Step 3: Review and create

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name: eks-admin
Console password type: None
Require password reset: No

Permissions summary

Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

Click on “Create User”

User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

View user

Users (1)

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Search

< 1 >

User name	Path	Group	Last activity	MFA	Password age
eks-admin	/	0	-	-	-

Delete Create user

Click on the Created User

eks-admin

Info

Delete

Summary

ARN

arn:aws:iam::533267105378:user/eks-admin

Console access

Disabled

Access key 1

Create access key

Created

January 26, 2025, 16:38 (UTC+05:30)

Last console sign-in

-

Permissions

Groups

Tags

Security credentials

Last Accessed

Console sign-in

Enable console access

Console sign-in link

https://533267105378.signin.aws.amazon.com/console

Console password

Not enabled

Click on “Security Credentials”

Access keys (0)

Create access key

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](#)

No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#)

Create access key

SSH public keys for AWS CodeCommit (0)

Actions

Upload SSH public key

Use SSH public keys to authenticate access to AWS CodeCommit repositories. You can have a maximum of five SSH public keys (active or inactive) at a time. [Learn more](#)

SSH Key ID

Uploaded

Status

No SSH public keys

Upload SSH public key

Click on “Create Access Key”

Access key best practices & alternatives

Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

Command Line Interface (CLI)

You plan to use this access key to enable the AWS CLI to access your AWS account.

Local code

You plan to use this access key to enable application code in a local development environment to access your AWS account.

Application running on an AWS compute service

You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

Alternatives recommended

- Use [AWS CloudShell](#), a browser-based CLI, to run commands. [Learn more](#)
- Use the [AWS CLI V2](#) and enable authentication through a user in IAM Identity Center. [Learn more](#)

Confirmation

☒
I understand the above recommendation and want to proceed to create an access key.

Cancel
Next

click on Next

Click on “Create Access Key”

Step 1

Access key best practices & alternatives

Step 2 - optional

Set description tag

Step 3

Retrieve access keys

Set description tag - optional
[Info](#)

The description for this access key will be attached to this user as a tag and shown alongside the access key.

Description tag value
Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @

Cancel
Previous
Create access key

IAM > Users > eks-admin > Create access key

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1

Access key best practices & alternatives

Step 2 - optional

Set description tag

Step 3

Retrieve access keys

Retrieve access keys
[Info](#)

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key
Secret access key

AKIAXYKJSIZRGATJX5U2

Show

Access Key has been Successfully Created.

Copy the Access Key and Secret Key

Go to terminal

Paste the AWS Access Key and Secret Key

```
ubuntu@ip-172-31-12-203: ~
ubuntu@ip-172-31-12-203:~$ aws configure
AWS Access Key ID [None]: AKIAXYKJSIZRGATJX5U2
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]:
ubuntu@ip-172-31-12-203:~$
```

Install kubectl

```
curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectl
chmod +x ./kubectl
sudo mv ./kubectl /usr/local/bin
kubectl version --short --client
```

```
ubuntu@ip-172-31-12-203:~$ curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectl
chmod +x ./kubectl
sudo mv ./kubectl /usr/local/bin
kubectl version --short --client
% Total % Received % Xferd Average Speed Time Time Time Current
Dload upload Total Spent Left Speed
100 57.4M 100 57.4M 0 0 4177k 0 0:00:14 0:00:14 --:--:-- 5054k
Client Version: v1.19.6-eks-49a6c0
ubuntu@ip-172-31-12-203:~$ kubectl version
Client Version: version.Info{Major:"1", Minor:"19+", GitVersion:"v1.19.6-eks-49a6c0", GitCommit:"49a6c0bf091506e7bafcdb1b142351b69363355a", GitTreeState:"clean", BuildDate:"2020-12-23T22:13:28Z", GoVersion:"go1.15.5", Compiler:"gc", Platform:"linux/amd64"}
The connection to the server localhost:8080 was refused - did you specify the right host or port?
ubuntu@ip-172-31-12-203:~$
```

Check version :

- kubectl version

Install eksctl

```
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
```

```
ubuntu@ip-172-31-12-203:~$ 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
0.202.0
```

To Create Cluster

```
create cluster --name tws-cluster --region ap-south-1 --node-type t3.small --nodes-min 2 --nodes-max 3
```

```
ubuntu@ip-172-31-12-203:~$ eksctl create cluster --name tws-cluster --region ap-south-1 --node-type t3.small --nodes-min 2 --nodes-max 3
2025-01-26 11:35:19 [i] eksctl version 0.202.0
2025-01-26 11:35:19 [i] using region ap-south-1
2025-01-26 11:35:19 [i] setting availability zones to [ap-south-1a ap-south-1b ap-south-1c]
2025-01-26 11:35:19 [i] subnets for ap-south-1a - public:192.168.0.0/19 private:192.168.96.0/19
2025-01-26 11:35:19 [i] subnets for ap-south-1b - public:192.168.32.0/19 private:192.168.128.0/19
2025-01-26 11:35:19 [i] subnets for ap-south-1c - public:192.168.64.0/19 private:192.168.160.0/19
2025-01-26 11:35:19 [i] nodegroup "ng-2b125820" will use "" [AmazonLinux2/1.30]
2025-01-26 11:35:19 [i] using Kubernetes version 1.30
2025-01-26 11:35:19 [i] creating EKS cluster "tws-cluster" in "ap-south-1" region with managed nodes
2025-01-26 11:35:19 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-01-26 11:35:19 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=ap-south-1 --cluster=tws-cluster'
2025-01-26 11:35:19 [i] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "tws-cluster" in "ap-south-1"
2025-01-26 11:35:19 [i] Cloudwatch logging will not be enabled for cluster "tws-cluster" in "ap-south-1"
2025-01-26 11:35:19 [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=tws-cluster'
2025-01-26 11:35:19 [i] default addons metrics-server, vpc-cni, kube-proxy, coredns were not specified, will install them as EKS addons
2025-01-26 11:35:19 [i]
2 sequential tasks: { create cluster control plane "tws-cluster",
  2 sequential sub-tasks: {
    2 sequential sub-tasks: {
      1 task: { create addons },
      wait for control plane to become ready,
    },
    create managed nodegroup "ng-2b125820",
  },
}
2025-01-26 11:35:19 [i] building cluster stack "eksctl-tws-cluster-cluster"
2025-01-26 11:35:19 [i] deploying stack "eksctl-tws-cluster-cluster"
```

It will create 2 two nodes but if loads increases, it will scale up to 3 nodes.

Run Manifests

kubectl create namespace two-tier-ns

kubectl apply -f .

Kubectl delete -f .

```
ubuntu@ip-172-31-12-203:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
ip-192-168-13-129.ap-south-1.compute.internal Ready    <none>   28m   v1.30.8-eks-aeac579
ip-192-168-91-122.ap-south-1.compute.internal Ready    <none>   28m   v1.30.8-eks-aeac579
ubuntu@ip-172-31-12-203:~$
```



```

ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ cat mysql-secrets.yml
apiVersion: v1
kind: Secret
metadata:
  name: mysql-secret
type: Opaque
data:
  MYSQL_ROOT_PASSWORD: YWRtaW4=
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ kubectl apply -f mysql-secrets.yml
secret/mysql-secret created
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ cat mysql-configmap.yml
apiVersion: v1
kind: ConfigMap
metadata:
  name: mysql-initdb-config
data:
  init.sql: |
    CREATE DATABASE IF NOT EXISTS mydb;
    USE mydb;
    CREATE TABLE messages (id INT AUTO_INCREMENT PRIMARY KEY, message TEXT);
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ kubectl apply -f mysql-configmap.yml
configmap/mysql-initdb-config created
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ cat mysql-deployment.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql
  labels:
    app: mysql
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mysql

```

```

  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:latest
          env:
            - name: MYSQL_ROOT_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: mysql-secret
                  key: MYSQL_ROOT_PASSWORD
            - name: MYSQL_DATABASE
              value: "mydb"
            - name: MYSQL_USER
              value: "admin"
            - name: MYSQL_PASSWORD
              value: "admin"
          ports:
            - containerPort: 3306
          volumeMounts:
            - name: mysql-initdb
              mountPath: docker-entrypoint-initdb.d
          volumes:
            - name: mysql-initdb
              configMap:
                name: mysql-initdb-config # Config name
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ |

```

```

ubuntu@ip-172-31-12-203: ~/two-tier-flask-app/eks-manifests
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ cat mysql-secrets.yml
apiVersion: v1
kind: Secret
metadata:
  name: mysql-secret
type: Opaque
data:
  MYSQL_ROOT_PASSWORD: YWRtaW4=
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ |

```

```

ubuntu@ip-172-31-12-203: ~/two-tier-flask-app/eks-manifests$ cat mysql-svc.yml
apiVersion: v1
kind: Service
metadata:
  name: mysql
spec:
  selector:
    app: mysql
  ports:
    - port: 3306
      targetPort: 3306
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ kubectl apply -f mysql-svc.yml
service/mysql created
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ s|

```

```

ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ cat two-tier-app-deployment.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: two-tier-app
  labels:
    app: two-tier-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: two-tier-app
  template:
    metadata:
      labels:
        app: two-tier-app
    spec:
      containers:
        - name: two-tier-app
          image: trainwithshubham/flaskapp:latest
          env:
            - name: MYSQL_HOST
              value: mysql # this is your mysql's service clusture IP, Make sure to change it with yours
            - name: MYSQL_PASSWORD
              value: "admin"
            - name: MYSQL_USER
              value: "root"
            - name: MYSQL_DB
              value: "mydb"
          ports:
            - containerPort: 5000
          imagePullPolicy: Always
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ |

```

```

ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ kubectl apply -f two-tier-app-deployment.yml
deployment.apps/two-tier-app created
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ cat two-tier-app-svc.yml
apiVersion: v1
kind: Service
metadata:
  name: two-tier-app-service
spec:
  selector:
    app: two-tier-app
  type: LoadBalancer
  ports:
    - protocol: TCP
      port: 80
      targetPort: 5000
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ kubectl apply -f two-tier-app-svc.yml
service/two-tier-app-service created
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$ |

```

KubectI get all

```

ubuntu@ip-172-31-12-203: ~/two-tier-flask-app/eks-manifests$ kubectl get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/mysql-c869ddbbc-hfc7s          1/1     Running   0           91s
pod/two-tier-app-784fcf9f5d-2kl7w  1/1     Running   0           22m

NAME                                TYPE                CLUSTER-IP    EXTERNAL-IP
service/kubernetes                  ClusterIP           10.100.0.1     <none>
service/mysql                        ClusterIP           10.100.43.78   <none>
service/two-tier-app-service        LoadBalancer       10.100.93.56   a79234174dfe247c7870ca9d8ef243b6-543344609.ap-south-1.elb
.amazonaws.com 80:30409/TCP    21m

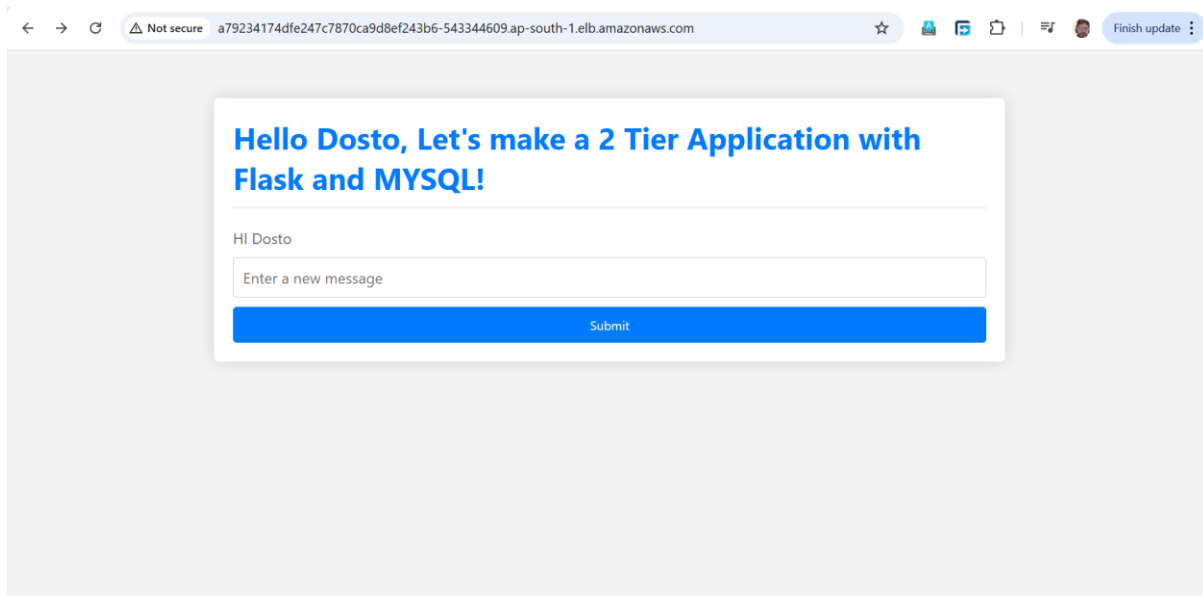
NAME                                READY    UP-TO-DATE    AVAILABLE   AGE
deployment.apps/mysql              1/1      1              1           91s
deployment.apps/two-tier-app       1/1      1              1           22m

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/mysql-c869ddbbc    1          1          1        91s
replicaset.apps/two-tier-app-784fcf9f5d 1          1          1        22m
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$

```

To check whether the secrets reached to configmap

Describe the Pod: `kubectl describe pod mysql-c869ddbbc-hfc7s`



To delete a Cluster

`eksctl delete cluster --name tws-cluster --region ap-south-1`

```

ubuntu@ip-172-31-12-203: ~/two-tier-flask-app/eks-manifests$ eksctl delete cluster --name tws-cluster --region ap-south-1
2025-01-26 13:46:20 [i] deleting EKS cluster "tws-cluster"
2025-01-26 13:46:20 [i] will drain 0 unmanaged nodegroup(s) in cluster "tws-cluster"
2025-01-26 13:46:20 [i] starting parallel draining, max in-flight of 1
2025-01-26 13:46:20 [i] deleted 0 Fargate profile(s)
2025-01-26 13:46:20 [✓] kubeconfig has been updated
2025-01-26 13:46:20 [i] cleaning up AWS load balancers created by Kubernetes objects of kind Service or Ingress
2025-01-26 13:46:57 [i] 2 sequential tasks: { delete nodegroup "ng-2b125820", delete cluster control plane "tws-cluster" [async]
2025-01-26 13:46:57 [i] }
2025-01-26 13:46:57 [i] will delete stack "eksctl-tws-cluster-nodegroup-ng-2b125820"
2025-01-26 13:46:57 [i] waiting for stack "eksctl-tws-cluster-nodegroup-ng-2b125820" to get deleted
2025-01-26 13:46:57 [i] waiting for cloudFormation stack "eksctl-tws-cluster-nodegroup-ng-2b125820"
2025-01-26 13:47:27 [i] waiting for cloudFormation stack "eksctl-tws-cluster-nodegroup-ng-2b125820"
2025-01-26 13:48:05 [i] waiting for cloudFormation stack "eksctl-tws-cluster-nodegroup-ng-2b125820"
2025-01-26 13:48:45 [i] waiting for cloudFormation stack "eksctl-tws-cluster-nodegroup-ng-2b125820"
2025-01-26 13:49:38 [i] waiting for cloudFormation stack "eksctl-tws-cluster-nodegroup-ng-2b125820"
2025-01-26 13:50:32 [i] waiting for cloudFormation stack "eksctl-tws-cluster-nodegroup-ng-2b125820"
2025-01-26 13:51:53 [i] waiting for cloudFormation stack "eksctl-tws-cluster-nodegroup-ng-2b125820"
2025-01-26 13:53:41 [i] waiting for cloudFormation stack "eksctl-tws-cluster-nodegroup-ng-2b125820"
2025-01-26 13:55:39 [i] waiting for cloudFormation stack "eksctl-tws-cluster-nodegroup-ng-2b125820"
2025-01-26 13:55:39 [i] will delete stack "eksctl-tws-cluster-cluster"
2025-01-26 13:55:39 [✓] all cluster resources were deleted
ubuntu@ip-172-31-12-203:~/two-tier-flask-app/eks-manifests$

```

