

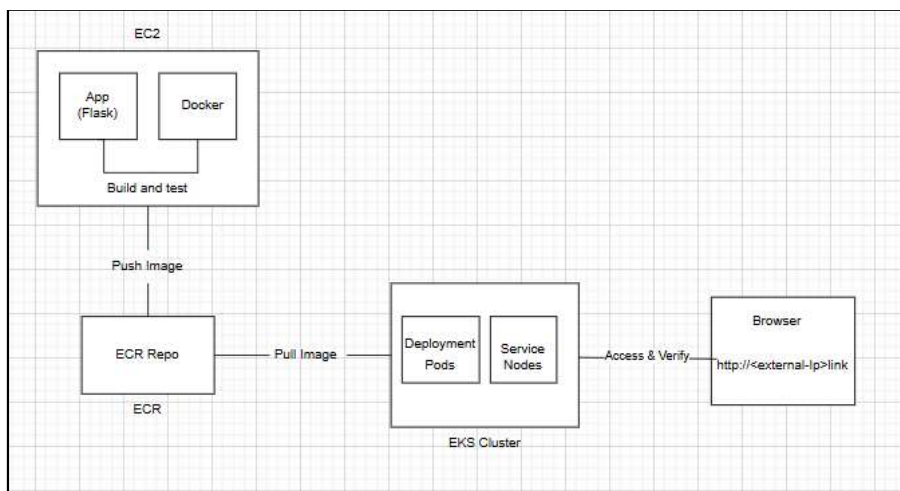
## **Task : 18**

### **Kubernetes , ECR and Docker concept**

#### **Objective**

Deployed a Python Flask application on AWS EKS using Kubernetes Deployments and Services, enabling external access via LoadBalancer while managing containerized workloads in a scalable cloud environment.

#### **Flow Diagram :**



## Step 1 : Create Ec2 Instance.

- Name : DevOps-Server
- Security group :
  - Allow ports:

Port	Reason
22	SSH
5000	Flask
80	HTTP
30000-32767	Kubernetes NodePort

**Instance summary for i-0031792373a2da024 (DevOps-Server)**

Updated 2 minutes ago

**Instance ID**  
i-0031792373a2da024

**Public IPv4 address**  
3.133.104.129 | [open address](#)

**Instance state**  
Running

**Private IPv4 addresses**  
172.31.31.62

**Public DNS**  
ec2-3-133-104-129.us-east-2.compute.amazonaws.com | [open address](#)

**IPv6 address**  
-

**Hostname type**  
IP name: ip-172-31-31-62.us-east-2.compute.internal

**Private IP DNS name (IPv4 only)**  
ip-172-31-31-62.us-east-2.compute.internal

**Answer private resource DNS name**  
IPv4 (A)

**Instance type**  
t3.micro

**Auto-assigned IP address**  
3.133.104.129 [Public IP]

**VPC ID**  
vpc-0c50459220437c177

**Subnet ID**  
subnet-01786fd5529f79144

**Elastic IP addresses**  
-

**AWS Compute Optimizer finding**  
Opt-in to AWS Compute Optimizer for recommendations. | [Learn more](#)

**IAM Role**  
-

**Auto Scaling Group name**  
-

**IMDSv2**  
Required

**Managed**  
false

**Edit inbound rules**

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	Actions
sg-0d7224a5fe48753c2	HTTPS	TCP	443	Custom	0.0.0.0/0	Delete
sg-0be97a8992cc01a7f	SSH	TCP	22	Custom	0.0.0.0/0	Delete
sg-0c533657575fb0e9c	Custom TCP	TCP	5000	Custom	0.0.0.0/0	Delete
sg-0eed1c4aa38e8b140	HTTP	TCP	80	Custom	0.0.0.0/0	Delete

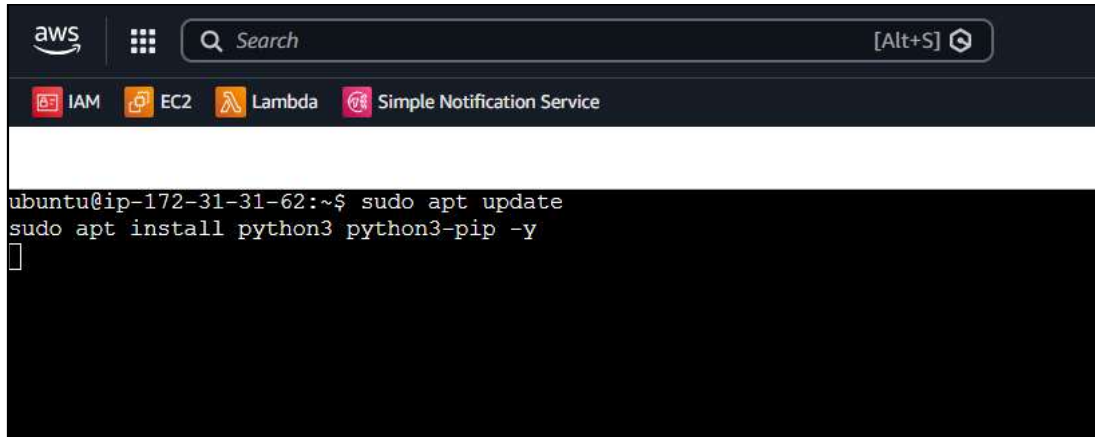
[Add rule](#)

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Preview changes](#) [Save rules](#)

## Step 2 : Create and deploy application on Server

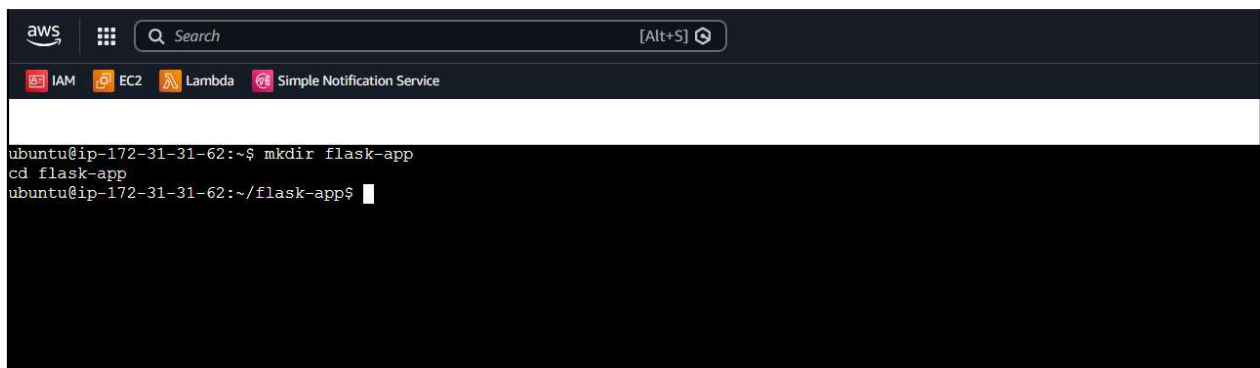
- Install Python & tools
  - `sudo apt update`
  - `sudo apt install python3 python3-pip -y`



A screenshot of an AWS terminal window. The top bar shows the AWS logo, a search bar, and icons for IAM, EC2, Lambda, and Simple Notification Service. The terminal output shows the following commands and their execution:

```
ubuntu@ip-172-31-31-62:~$ sudo apt update
sudo apt install python3 python3-pip -y
```

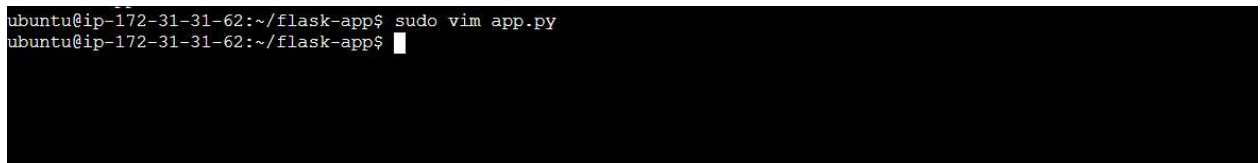
- Create project folder:
  - `mkdir flask-app`
  - `cd flask-app`



A screenshot of an AWS terminal window. The top bar shows the AWS logo, a search bar, and icons for IAM, EC2, Lambda, and Simple Notification Service. The terminal output shows the following commands and their execution:

```
ubuntu@ip-172-31-31-62:~$ mkdir flask-app
cd flask-app
ubuntu@ip-172-31-31-62:~/flask-app$
```

- Create app file
  - `Vim app.py`



A screenshot of an AWS terminal window. The top bar shows the AWS logo, a search bar, and icons for IAM, EC2, Lambda, and Simple Notification Service. The terminal output shows the following commands and their execution:

```
ubuntu@ip-172-31-31-62:~/flask-app$ sudo vim app.py
ubuntu@ip-172-31-31-62:~/flask-app$
```

```
aws | Search [Alt+S]
IAM EC2 Lambda Simple Notification Service

from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello():
    return "Hello from Docker running in Kubernetes"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

- Create Virtual environment and active it
  - Python -m venv venv
  - Source venv/bin/activate

```
ubuntu@ip-172-31-31-62:~/flask-app$
ubuntu@ip-172-31-31-62:~/flask-app$ source venv/bin/activate
(venv) ubuntu@ip-172-31-31-62:~/flask-app$ ls
app.py  venv
(venv) ubuntu@ip-172-31-31-62:~/flask-app$
```

- Install Flask:
  - pip3 install flask

```
(venv) ubuntu@ip-172-31-31-62:~/flask-app$ pip install flask
Collecting flask
  Downloading flask-3.1.2-py3-none-any.whl.metadata (3.2 kB)
Collecting blinker>=1.9.0 (from flask)
  Downloading blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting click>=8.1.3 (from flask)
  Downloading click-8.3.1-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.2.0 (from flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting jinja2>=3.1.2 (from flask)
  Downloading jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting markupsafe>=2.1.1 (from flask)
  Downloading markupsafe-3.0.3-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl.metadata (2.7 kB)
```

- Run application
  - Python app.py

```
^C(venv) ubuntu@ip-172-31-31-62:~/flask-app$ sudo vim app.py
(venv) ubuntu@ip-172-31-31-62:~/flask-app$ python app.py
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.31.31.62:5000
Press CTRL+C to quit
```

- Check application
  - http://<public\_ip>:5000



⚠ Not secure 3.133.104.129:5000

Hello from Docker running in Kubernetes

### Step 3 : Docker Image Creation

- Install Docker
  - `sudo apt install docker.io -y`
  - `sudo systemctl start docker`
  - `sudo usermod -aG docker ubuntu newgrp docker`

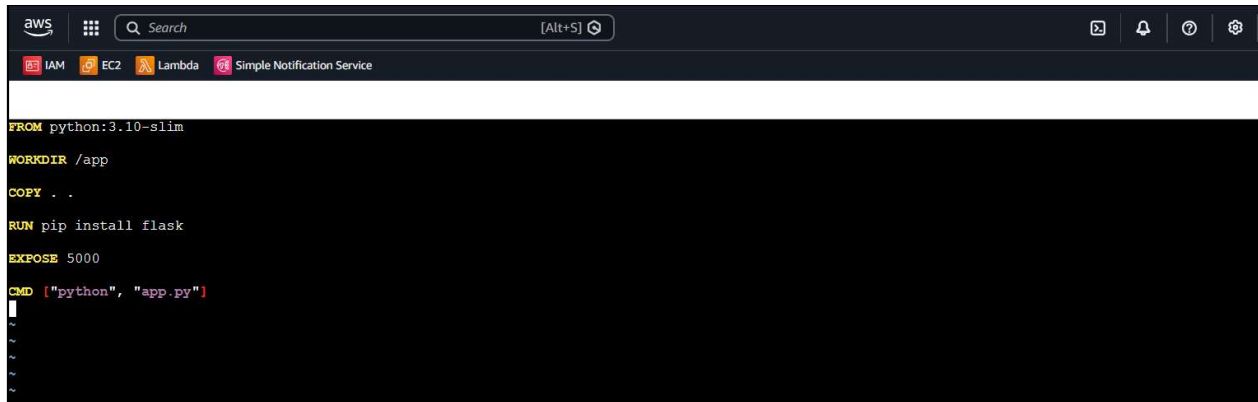
```
^C(venv) ubuntu@ip-172-31-31-62:~/flask-app$ sudo apt install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 4 not upgraded.
Need to get 76.1 MB of archives.
After this operation, 288 MB of additional disk space will be used.
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.3.3-0ubuntu1~24.04.3 [8815 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.28-0ubuntu1~24.04.2 [38.4 MB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dns-root-data all 2024071801-ubuntu0.24.04.1 [5918 B]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dnsmasq-base amd64 2.90-2ubuntu0.1 [376 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 28.2.2-0ubuntu1~24.04.1 [28.3 MB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 ubuntu-fan all 0.12.16+24.04.1 [34.2 kB]
Fetched 76.1 MB in 1s (89.3 MB/s)
```

```
(venv) ubuntu@ip-172-31-31-62:~/flask-app$ sudo systemctl start docker
(venv) ubuntu@ip-172-31-31-62:~/flask-app$ █
```

```
(venv) ubuntu@ip-172-31-31-62:~/flask-app$ sudo usermod -aG docker ubuntu
newgrp docker
ubuntu@ip-172-31-31-62:~/flask-app$ █
```

- Create Dockerfile
  - Inside flask-app folder: nano Dockerfile

```
ubuntu@ip-172-31-31-62:~/flask-app$ sudo vim Dockerfile
ubuntu@ip-172-31-31-62:~/flask-app$
```



```
FROM python:3.10-slim
WORKDIR /app
COPY . .
RUN pip install flask
EXPOSE 5000
CMD ["python", "app.py"]
```

```
ubuntu@ip-172-31-31-62:~/flask-app$ ls
Dockerfile  app.py  venv
ubuntu@ip-172-31-31-62:~/flask-app$
```

- Build Docker Image
  - docker build -t flask-k8s-app .

```
ubuntu@ip-172-31-31-62:~/flask-app$ docker build -t flask-k8s-app .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 19.02MB
Step 1/6 : FROM python:3.10-slim
3.10-slim: Pulling from library/python
0c8d55a45c0d: Pulling fs layer
b9561eba91ba: Pulling fs layer
0ccaa604ed2: Pulling fs layer
d00020a00819: Pulling fs layer
d00020a00819: Waiting
b9561eba91ba: Verifying Checksum
b9561eba91ba: Download complete
d00020a00819: Verifying Checksum
```

- Check image:
  - docker images

```
ubuntu@ip-172-31-31-62:~/flask-app$ docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
flask-k8s-app       latest         e2c6f5ef37d6   37 seconds ago 152MB
python              3.10-slim     6973bbbb27d3   31 hours ago   122MB
ubuntu@ip-172-31-31-62:~/flask-app$
```

- Run container locally
  - `docker run -d -p 5000:5000 flask-k8s-app`

```
ubuntu@ip-172-31-31-62:~/flask-app$ docker run -d -p 5000:5000 flask-k8s-app
aee3a431e998fd176b02a48e9db8c4c901f5759d0575167136e4105254d101a0
ubuntu@ip-172-31-31-62:~/flask-app$
```

- Check application on browser.
  - `http://<public_ip>:5000`



- stop container
  - `docker ps`
  - `docker stop CONTAINER_ID`

```
ubuntu@ip-172-31-31-62:~/flask-app$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                                                                 NAMES
aee3a431e998   flask-k8s-app  "python app.py"         About a minute ago    Up About a minute    0.0.0.0:5000->5000/tcp, [::]:5000->5000/tcp    festive_galois
ubuntu@ip-172-31-31-62:~/flask-app$
```

```
ubuntu@ip-172-31-31-62:~/flask-app$ docker stop aee3a431e998
aee3a431e998
```

```
ubuntu@ip-172-31-31-62:~/flask-app$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
ubuntu@ip-172-31-31-62:~/flask-app$
```



## Step 4 : Create IAM User

- Left menu → click **Users**
- Click **Create user**
  - User name: DevOps-User
- Attach Permissions

DevOps-User

Delete

Summary

ARN  
arn:aws:iam::891146182066:user/DevOps-User

Created  
February 04, 2026, 16:21 (UTC+05:30)

Console access  
Enabled without MFA

Last console sign-in  
Never

Access key 1  
AKIA467DTTWZH2SQYT3N - Active  
Used today, 20 hours old.

Access key 2  
Create access key

Permissions

Groups

Tags

Security credentials

Last Accessed

Permissions policies (8)

Remove

Add permissions

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type  
All types

<input type="checkbox"/>	Policy name	Type	Attached via
<input type="checkbox"/>	AdministratorAccess	AWS managed - job function	Directly
<input type="checkbox"/>	AmazonEC2ContainerRegistryFullAccess	AWS managed	Directly
<input type="checkbox"/>	AmazonEC2ContainerRegistryReadOnly	AWS managed	Directly
<input type="checkbox"/>	AmazonEKS_CNI_Policy	AWS managed	Directly

Access keys (1)

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

AKIA467DTTWZH2SQYT3N

Actions

Description  
-

Status  
Active

Last used  
33 minutes ago

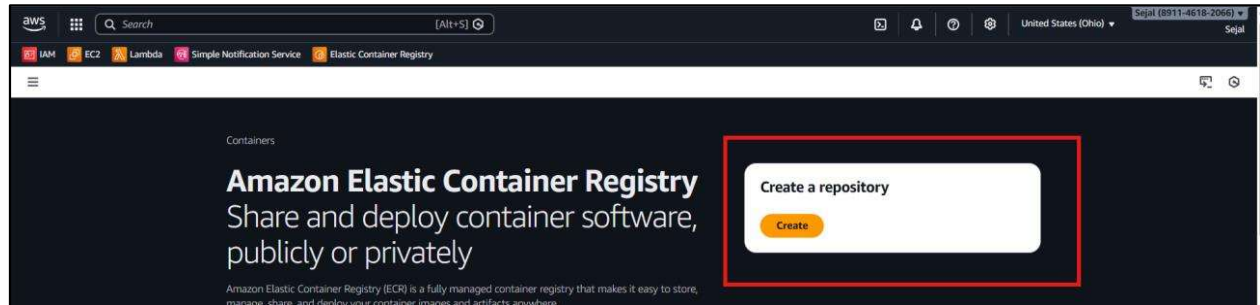
Created  
20 hours ago

Last used region  
us-east-2

Last used service  
eks

## Step 5 : Amazon ECR Repository

- Go AWS Console → ECR



- Repository
- Name: flask-k8s-repo

Amazon ECR > Private registry > Repositories > Create private repository

### Create private repository

**General settings**

**Repository name**  
Enter a concise name. Repositories support namespaces, which you can use to group similar repositories.  
891146182066.dkr.ecr.us-east-2.amazonaws.com/ **flask-k8s-repo**  
14 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, and special characters \_./.

**Image tag settings** [Info](#)

**Image tag mutability**  
Choose the tag mutability setting.

☒ **Mutable**  
Image tags can be overwritten.

☐ **Immutable**  
Image tags can't be overwritten.

**Mutable tag exclusions**  
Tags that match these filters will be immutable (can't be overwritten). Using wildcards (\*) will match zero or more image tag characters.  
Tags must only contain letters, numbers, and special characters [\_\*:]. Each filter is limited to 128 characters, 2 wildcards (\*), and you can add up to 5 filters in the exclusion list.

[Add filter](#)

aws | Search | [Alt+S] | United States (Ohio) | Sejal (8911-4618-2066) | Sejal

Amazon ECR > Private registry > Repositories

**Amazon Elastic Container Service**

- Private registry
  - Repositories
  - Features & Settings
- Public registry
  - Repositories
  - Settings

**Private repositories (1)**

Search by repository substring

Repository name	URI	Created at	Tag immutability	Encryption type
flask-k8s-repo	891146182066.dkr.ecr.us-east-2.amazonaws.com/flask-k8s-repo	February 04, 2026, 16:12:57 (UTC+05.5)	Mutable	AES-256

View push commands | Delete | Actions | [Create repository](#)

- Install AWS CLI on EC2
  - `sudo apt install awscli -y`

```
ubuntu@ip-172-31-31-62:~$ aws --version
aws-cli/2.33.14 Python/3.13.11 Linux/6.14.0-1018-aws exe/x86_64.ubuntu.24
```

- Configure:
  - aws configure
- Enter:
  - Access key
  - Secret key
  - Region (example: ap-south-1)
  - output: json

```
ubuntu@ip-172-31-31-62:~$ aws configure
AWS Access Key ID [None]: AKIA467DTTWZHZSQYT3N
AWS Secret Access Key [None]: R9lGTIGyuMzMY9YJ5Xdw1xrflKd7eEJcNcTpFA
Default region name [None]:
Default output format [None]:
ubuntu@ip-172-31-31-62:~$
```

- Login Docker to ECR
  - aws ecr get-login-password --region <region\_name> | docker login --username AWS --password-stdin <ecr\_url>

```
ubuntu@ip-172-31-31-62:~$ aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 891146182066.dkr.ecr.us-east-2.amazonaws.com/flask-k8s-repo
WARNING! Your credentials are stored unencrypted in '/home/ubuntu/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
ubuntu@ip-172-31-31-62:~$
```

- Tag Image
  - docker tag flask-k8s-app:latest <ecr\_url>:latest

```
ubuntu@ip-172-31-31-62:~$ docker tag flask-k8s-app:latest 891146182066.dkr.ecr.us-east-2.amazonaws.com/flask-k8s-repo:latest
ubuntu@ip-172-31-31-62:~$
```

- Push to ECR
  - docker push <ecr\_url>:latest

```
ubuntu@ip-172-31-31-62:~$ docker push 891146182066.dkr.ecr.us-east-2.amazonaws.com/flask-k8s-repo:latest
The push refers to repository [891146182066.dkr.ecr.us-east-2.amazonaws.com/flask-k8s-repo]
5fd6f784a1a5: Pushing [=====>] 12.64MB
67cb3830c39b: Pushing [=====>] 19.02MB
c3a2e1ac4bc0: Pushed
73fac8bb9d80: Pushed
d394714285eb: Pushing [=====>] 17.43MB/39.44MB
47593b3f7136: Pushed
a8ff6f8cbdfd: Pushing [=====>] 20.69MB/78.62MB
```

- Image is successfully upload in RCR

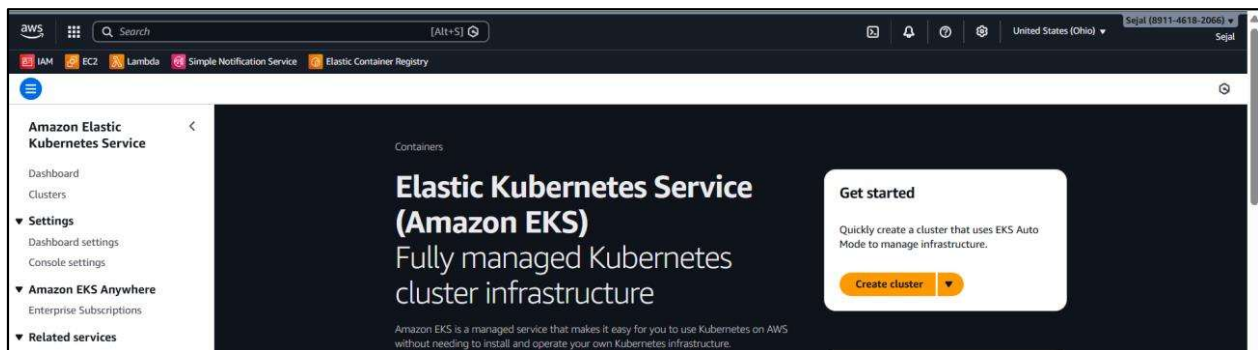
The screenshot shows the Amazon ECR console interface. The left sidebar displays the navigation menu with 'Private registry' and 'Public registry' sections. The main content area is titled 'flask-k8s-repo' and shows the 'Images' tab. A table lists the images in the repository, with one image highlighted by a red box.

<input type="checkbox"/>	Image tags	Type	Created at	Image size (MB)	Image digest	Last pulled at
<input type="checkbox"/>	latest	Image	February 04, 2026, 16:29:58 (UTC+05.5)	56.43	sha256:ab249d9a...	February 05, 2026, 14:47:06 (UTC+05.5)

## Step 6 : Create EKS Cluster

### Create a New Cluster.

- Name : flask-k8s-cluster
- Kubernetes Version : 1.34
- Upgrade Policy : Standard Policy
- Attach role : Recommended
- Allow cluster administrator access
- Cluster authentication mode : EKS API
- ARC Zonal shift : Disable
- Select VPC and Subnets
- Attach security group
- cluster endpoint access : Public
- enable CloudWatch, Network monitoring,
- Control plane logs : Check all
- select add ons : Metrics Server, Cube proxy, CoreDNS



### Configure cluster

**Configuration options - new** Info  
Choose how you would like to configure the cluster.

☐ Quick configuration (with EKS Auto Mode) - new  
Quickly create a cluster with production-grade default settings. The configuration uses EKS Auto Mode to automate infrastructure tasks like creating nodes and provisioning storage.

☒ Custom configuration  
To change default settings prior to creation, choose this option. This configuration gives the option to use EKS Auto Mode and customize the cluster's configuration.

## EKS Auto Mode - *new* [Info](#)

Choose if you would like to use EKS's Auto Mode.

### ☒ Use EKS Auto Mode

EKS automates routine cluster tasks for compute, storage, and networking. When a new pod can't fit onto existing nodes, EKS creates a new node. EKS combines cluster infrastructure managed by AWS with integrated Kubernetes capabilities to meet application compute needs. [View pricing](#)

► Included capabilities

## Cluster configuration [Info](#)

### Name

Enter a unique name for this cluster. This property cannot be changed after the cluster is created.

flask-k8s-cluster

The cluster name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 100.

### Cluster IAM role [Info](#)

Select the Cluster IAM role to allow the Kubernetes control plane to manage AWS resources on your behalf. This cannot be changed after the cluster is created. To create a new custom role, follow the instructions in the [Amazon EKS User Guide](#).

AmazonEKSAutoClusterRole-1



Create recommended role

## Kubernetes version settings

### Kubernetes version [Info](#)

Select Kubernetes version for this cluster.

1.34

### Upgrade policy [Info](#)

Choose one of the following options. You can switch the setting later while the standard support period is in effect.

#### ☒ Standard support

This option supports the Kubernetes version for 14 months after the release date. There is no additional cost. When standard support ends, your cluster will be auto upgraded to the next version.

#### ☐ Extended support

This option supports the Kubernetes version for 26 months after the release date. The extended support period has an additional hourly cost that begins after the standard support period ends. When extended support ends, your cluster will be auto upgraded to the next version.

## Control plane scaling tier - *new* [Info](#)

Select a scaling tier for your control plane. While Standard mode scales dynamically, higher tiers pre-provision your environment with fixed, high-performance resources. This eliminates scaling latency and provides the sustained throughput required for demanding workloads. [View tier pricing](#)

### ☐ Use a scaling tier

For predictable and high performance from your cluster's control plane, choose a scaling tier.

## Auto Mode Compute - *new* [Info](#)

Configure node management for your EKS cluster. EKS offers four compute options: EKS Auto Mode, EC2 Managed Node Groups, Fargate, and hybrid nodes. Node groups, Fargate profiles, and hybrid nodes are configured after cluster creation. You can also create self-managed nodes.

### Built-in node pools - *optional* [Info](#)

EKS Auto Mode uses node pools to create nodes for pods. The node IAM role will be associated with built-in node pools. Use the Kubernetes API after cluster creation to create your own node pools.

Choose node pool(s)

general-purpose

system

### Node IAM role [Info](#)

Nodes need an EC2 Instance IAM Role to launch and register with a cluster. To create a new custom role, follow the instructions in the [Amazon EKS User Guide](#).

AmazonEKSAutoNodeRole



Create recommended role

## Cluster access [Info](#)

Control how IAM principals can access this cluster.

### Bootstrap cluster administrator access [Info](#)

Choose whether the IAM principal creating the cluster has Kubernetes cluster administrator access.

#### ☒ Allow cluster administrator access

Allow cluster administrator access for your IAM principal.

#### ☐ Disallow cluster administrator access

Disallow cluster administrator access for your IAM principal.

### Cluster authentication mode [Info](#)

Configure which source the cluster will use for authenticated IAM principals.

#### ☒ EKS API

The cluster will source authenticated IAM principals only from EKS access entry APIs.

#### ☐ EKS API and ConfigMap

The cluster will source authenticated IAM principals from both EKS access entry APIs and the aws-auth ConfigMap.

### ARC Zonal shift [Info](#)

Shift application traffic away from an impaired Availability Zone (AZ) in your EKS cluster. You can change this later.

- ☐ Enabled  
EKS will register your cluster with ARC zonal shift to enable you to use zonal shift to shift application traffic away from an AZ
- ☒ Disabled  
EKS will not register your cluster with ARC zonal shift.

 Before you start a zonal shift, you need to setup your cluster environment to be resilient to an AZ failure beforehand.

[Learn more](#) 

### Deletion protection

Deletion protection must be turned off to be able to delete a cluster. It can be turned on and off after the cluster is created.

- ☒ Turn on deletion protection  
Deletion protection provides additional security against accidental cluster deletion.

## Specify networking

### Networking [Info](#)

IP address family and service IP address range cannot be changed after cluster creation.

#### VPC [Info](#)

Select a VPC to use for your EKS cluster resources.

vpc-0c50459220437c177 | Default




#### Subnets [Info](#)


Choose the subnets in your VPC where the control plane may place elastic network interfaces (ENIs) to facilitate communication with your cluster. To create a new subnet, go to the corresponding page in the [VPC console](#).

Select subnets



[Clear selected subnets](#)

subnet-01786fd5529f79144   
us-east-2b 172.31.16.0/20

subnet-07aa8d3d3b90b12b2   
us-east-2a 172.31.0.0/20

subnet-08df3d810ef26634b   
us-east-2c 172.31.32.0/20

### Additional security groups [Info](#)

EKS automatically creates a cluster security group on cluster creation to facilitate communication between worker nodes and control plane. Optionally, choose additional security groups to apply to the EKS-managed Elastic Network Interfaces that are created in your control plane subnets. To create a new security group, go to the corresponding page in the [VPC console](#).

Select security groups



[Clear selected security groups](#)

sg-0e42268f2434e1d1f  | default  
default VPC security group

### Choose cluster IP address family [Info](#)

Specify the IP address type for pods and services in your cluster.

- ☒ IPv4
- ☐ IPv6

### Configure Kubernetes service IP address block [Info](#)

- ☒ Specify the range from which cluster services will receive IP addresses.

### Configure remote networks to enable hybrid nodes [Info](#)

EKS Hybrid Nodes enables you to use on-premises and edge infrastructure as nodes in EKS clusters.

- ☒ Specify the CIDR blocks for your on-premises environments that you will use for hybrid nodes.

### Cluster endpoint access [Info](#)

Configure access to the Kubernetes API server endpoint.

- ☒ Public  
The cluster endpoint is accessible from outside of your VPC. Worker node traffic will leave your VPC to connect to the endpoint.
- ☐ Public and private  
The cluster endpoint is accessible from outside of your VPC. Worker node traffic to the endpoint will stay within your VPC.
- ☐ Private  
The cluster endpoint is only accessible through your VPC. Worker node traffic to the endpoint will stay within your VPC.

► [Advanced settings](#)

[Cancel](#)

[Previous](#)

[Next](#)



## Configure observability

### ► About observability

#### Metrics

##### Prometheus [Info](#)

☐ Send Prometheus metrics to Amazon Managed Service for Prometheus  
Monitor your application and infrastructure metrics with Amazon Managed Service for Prometheus. These metrics include system health and performance data.

ⓘ Agentless Prometheus metrics collection requires the cluster API server to be available privately. To make the following toggle available, select either the Public and private option or the Private option for Cluster endpoint access in Specify networking.

##### CloudWatch [Info](#)

☒ Send application and infrastructure telemetry to Amazon CloudWatch  
Installs the Amazon CloudWatch Observability add-on to send application metrics from CloudWatch APM and infrastructure telemetry from CloudWatch Container insights.

#### ► Services and telemetry included

## Container network observability [Info](#)

### Network monitoring [Info](#)

☒ Enable network monitoring  
Network monitoring enables you to easily troubleshoot by allowing you to visualize traffic in a service map, explore detailed flow tables for granular traffic analysis, monitor network performance metrics in real-time. [Learn more](#)

#### Included capabilities

**Service map** **Flow table** **Performance metric endpoint**

#### Name for your monitor

Service map, flow table, and performance metrics are enabled by Network Flow Monitor. After your monitor is created, you cannot change this name. [Learn more](#)

eks-flask-k8s-cluster-flow-monitor

Valid characters are a-z, A-Z, 0-9, underscores (\_), dashes (-), and periods (.). Character count: 34/255.

### ► Additional information

## Control plane logs [Info](#)

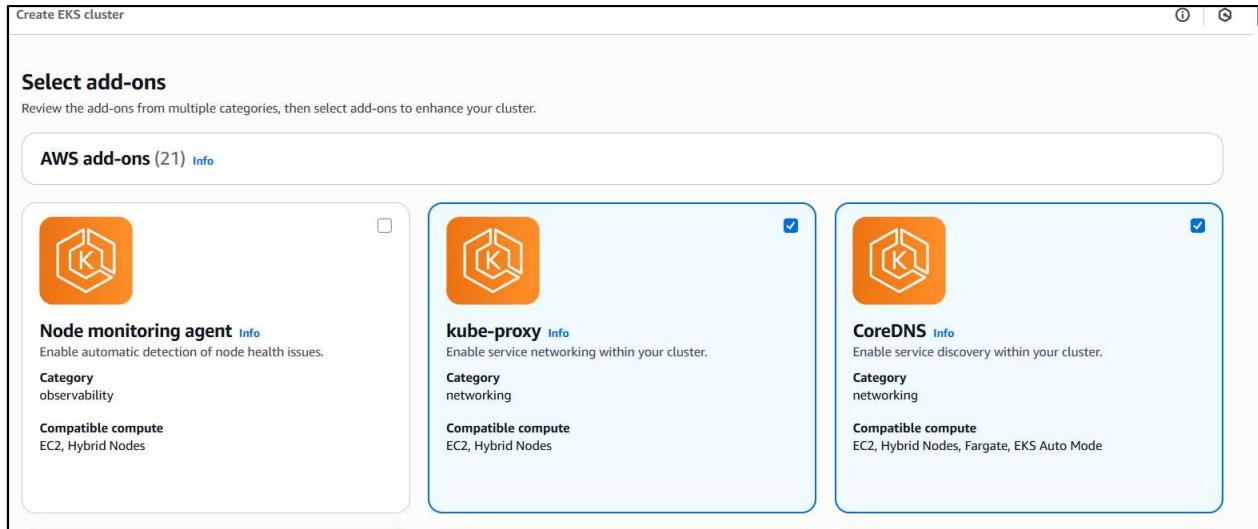
Send audit and diagnostic logs from the Amazon EKS control plane to CloudWatch Logs.

- ☒ **API server**  
Logs pertaining to API requests to the cluster.
- ☒ **Audit**  
Logs pertaining to cluster access via the Kubernetes API.
- ☒ **Authenticator**  
Logs pertaining to authentication requests into the cluster.
- ☒ **Controller manager**  
Logs pertaining to state of cluster controllers.
- ☒ **Scheduler**  
Logs pertaining to scheduling decisions.

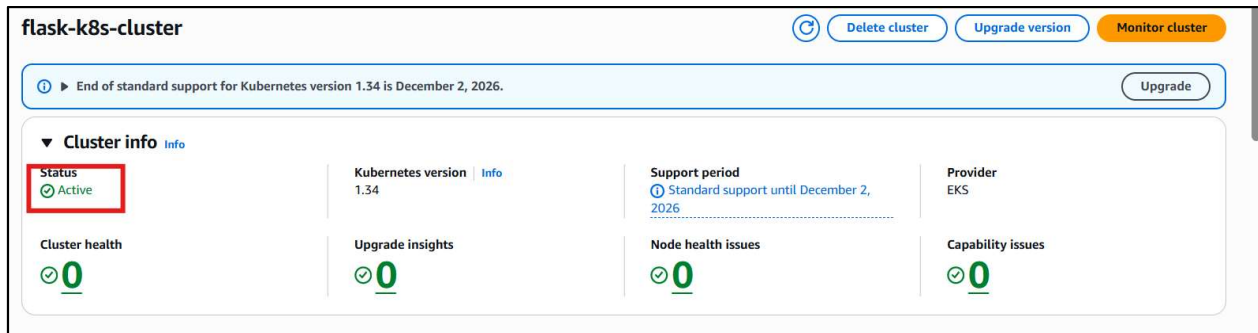
[Cancel](#)

[Previous](#)

[Next](#)



- Cluster is created and active .



## Step 7 : Create Node Group.

- Click on Compute
- Add nodes group.
- Create recommended role.

**flask-k8s-cluster**

🔄 Delete cluster Upgrade version Monitor cluster

🔔 End of standard support for Kubernetes version 1.34 is December 2, 2026. Upgrade

▼ Cluster info [Info](#)

Status  
🟢 Active

Kubernetes version  
1.34 [Info](#)

Support period  
🕒 Standard support until December 2, 2026

Provider  
EKS

Cluster health  
🟢 0

Upgrade insights  
🟢 0

Node health issues  
🟢 0

Capability issues  
🟢 0

Overview Resources **Compute** Networking Add-ons 1 Capabilities Access Observability Update history & backups Tags

Details

API server endpoint  
<https://015C313E3CE70EB49839C61EE9648B41.gr7.us-east-2.eks.amazonaws.com>

OpenID Connect provider URL  
<https://oidc.eks.us-east-2.amazonaws.com/id/015C313E3CE70EB49839C61EE9648B41>

Created  
🕒 11 minutes ago

Cluster ARN

**Node groups (0)** [Info](#)

Edit Delete Add

Node groups implement basic compute scaling through EC2 Auto Scaling groups.

🔍 Filter node groups by property or value

< 1 >

Group name	Desired size	AMI release version	Launch template	Status
No node groups This cluster does not have any node groups. Nodes that are not part of an Amazon EKS managed node group are not shown in the AWS console.				

Add node group

**Eks-nodes-role** [Info](#)

Delete

Allows EC2 instances to call AWS services on your behalf.

Summary 

Edit

Creation date  
February 04, 2026, 17:16 (UTC+05:30)

ARN  
[arn:aws:iam::891146182066:role/Eks-nodes-role](#)

Instance profile ARN  
[arn:aws:iam::891146182066:instance-profile/Eks-nodes-role](#)

Last activity  
-

Maximum session duration  
1 hour

Permissions Trust relationships Tags Last Accessed Revoke sessions

Permissions policies (3) [Info](#)

🔄 Simulate 📄 Remove Add permissions ▼

You can attach up to 10 managed policies.

🔍 Search Filter by Type All types < 1 > ⚙️

<input type="checkbox"/>	Policy name	Type	Attached entities
<input type="checkbox"/>	<a href="#">AmazonEC2ContainerRegistryReadOnly</a>	AWS managed	2
<input type="checkbox"/>	<a href="#">AmazonEKS_CNI_Policy</a>	AWS managed	2
<input type="checkbox"/>	<a href="#">AmazonEKSWorkerNodePolicy</a>	AWS managed	2

- Step 1
- Configure node group**
- Step 2
- Set compute and scaling configuration
- Step 3
- Specify networking
- Step 4
- Review and create

## Configure node group [Info](#)

A node group is a group of EC2 instances that supply compute capacity to your Amazon EKS cluster. You can add multiple node groups to your cluster.

### Node group configuration

These properties cannot be changed after the node group is created.

#### Name

Assign a unique name for this node group.

flask-nodes

The node group name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 63.

#### Node IAM role [Info](#)

Select the IAM role that will be used by the nodes. To create a new role, go to the [IAM console](#).

Eks-nodes-role



[Create recommended role](#)



The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.

[Learn more](#)

- AMI Type : Default
- Capacity type : On-Demand
- Instance types : t3.medium

Clusters > flask-k8s-cluster > Add node group

### Set compute and scaling configuration

#### Node group compute configuration

These properties cannot be changed after the node group is created.

**AMI type** [Info](#)  
Select the EKS-optimized Amazon Machine Image for nodes.

Amazon Linux 2023 (x86\_64) Standard (AL2023\_x86\_64\_STANDARD) ▼

**Capacity type**  
Select the capacity purchase option for this node group.

On-Demand ▼

**Instance types** [Info](#)  
Select instance types you prefer for this node group.

Q Enter an instance type

t3.medium  
vCPU: 2 vCPUs Memory: 4 GiB Network: Up to 5 Gigabit Max ENI: 3 Max IPs: 18 ✕

**Disk size**  
Select the size of the attached EBS volume for each node.

20 GiB

- Set
  - Desired Size : 2
  - Minimum : 2
  - Maximum : 2

### Node group scaling configuration

**Desired size**  
Set the desired number of nodes that the group should launch with initially.

2 nodes  
Desired node size must be greater than or equal to 0

**Minimum size**  
Set the minimum number of nodes that the group can scale in to.

1 nodes  
Minimum node size must be greater than or equal to 0

**Maximum size**  
Set the maximum number of nodes that the group can scale out to.

2 nodes  
Maximum node size must be greater than or equal to 1 and cannot be lower than the minimum size

- Set maximum unavailability :1

**Node group update configuration** [Info](#)

**Maximum unavailable**  
Set the maximum number or percentage of unavailable nodes to be tolerated during the node group version update.

☒ **Number**  
Enter a number

☐ **Percentage**  
Specify a percentage

**Value**  
 node  
Node count must be greater than 0.

**Update strategy**  
☒ **Default**  
☐ **Minimal**

- Select private subnets.

**Specify networking**

**Node group network configuration**  
These properties cannot be changed after the node group is created.

**Subnets** [Info](#)  
Specify the subnets in your VPC where your nodes will run. To create a new subnet, go to the corresponding page in the [VPC console](#).

Select subnets

subnet-01786fd5529f79144  
us-east-2b 172.31.16.0/20

subnet-07aa8d3d3b90b12b2  
us-east-2a 172.31.0.0/20

subnet-08df3d810ef26634b  
us-east-2c 172.31.32.0/20

☐ **Configure remote access to nodes** [Info](#)

Cancel

Previous

Next

- Node group is created.

**flask-nodes** [Refresh](#) [Edit](#) [Delete](#)

**Node group configuration** [Info](#)

**Kubernetes version**  
1.34

**AMI type** [Info](#)  
Amazon Linux 2023 (x86\_64) Standard

**Status**  
Creating

**AMI release version** [Info](#)  
1.34.2-20260129

**Instance types**  
t3.medium

**Disk size**  
20 GiB

Details

Nodes

Health issues 0

Kubernetes labels

Update config

Kubernetes taints

Update history

Tags

**Details**

**Node group ARN**  
[arn:aws:eks:us-east-2:891146182066:nodegroup/flask-k8s-cluster/flask-nodes/50ce143e-84f1-348c-a48c-cb477d1da3bb](#)

**Created**  
a few seconds ago

**Autoscaling group name**  
-

**Node IAM role ARN**  
-

**Capacity type**  
On-Demand

**Desired size**  
2 nodes

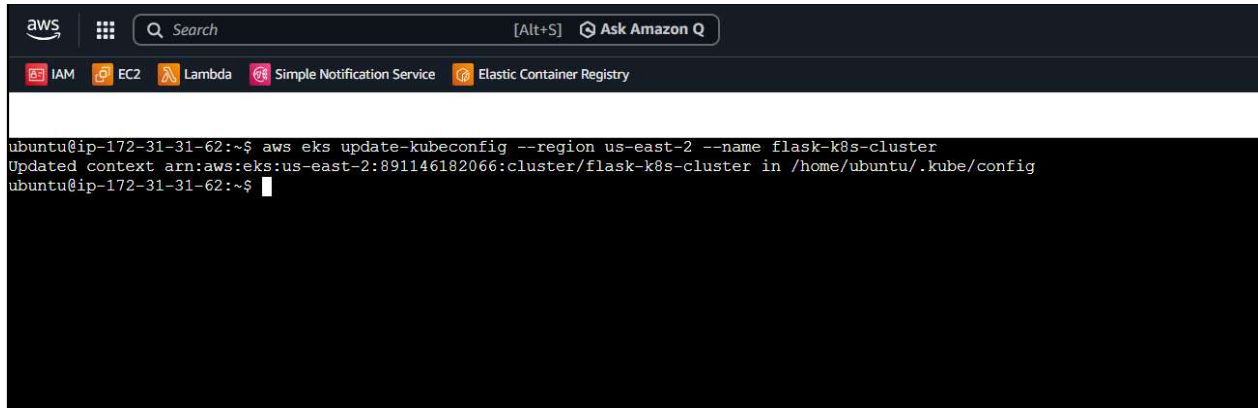
**Minimum size**  
1 node

**Maximum size**  
-

**Subnets**  
[subnet-01786fd5529f79144](#)  
[subnet-07aa8d3d3b90b12b2](#)  
[subnet-08df3d810ef26634b](#)

**Configure remote access to nodes**  
off

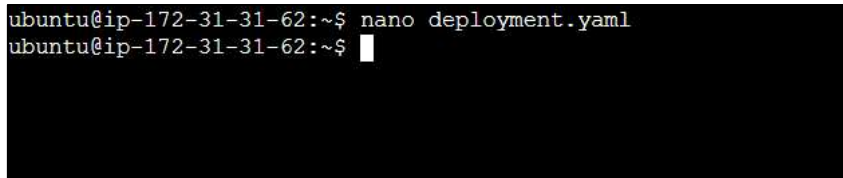
- Connect to EKS cluster
  - `aws eks update-kubeconfig --region <region_name> --name <cluster_name>`



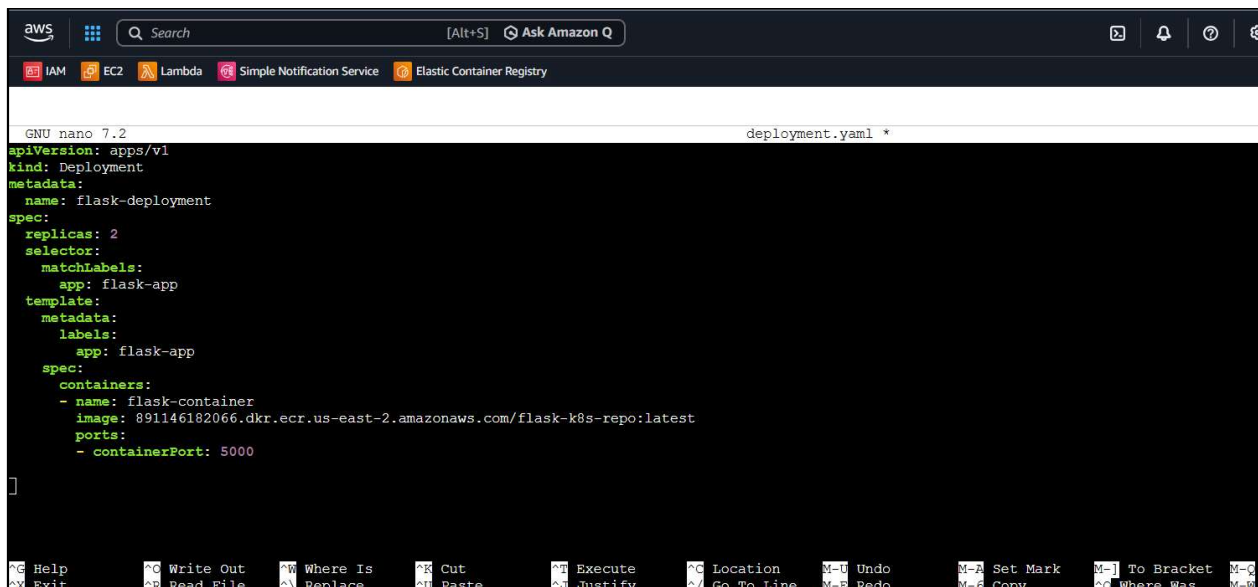
```
aws
IAM EC2 Lambda Simple Notification Service Elastic Container Registry

ubuntu@ip-172-31-31-62:~$ aws eks update-kubeconfig --region us-east-2 --name flask-k8s-cluster
Updated context arn:aws:eks:us-east-2:891146182066:cluster/flask-k8s-cluster in /home/ubuntu/.kube/config
ubuntu@ip-172-31-31-62:~$
```

- Create Deployment YAML
  - `nano deployment.yaml`



```
ubuntu@ip-172-31-31-62:~$ nano deployment.yaml
ubuntu@ip-172-31-31-62:~$
```



```
aws
IAM EC2 Lambda Simple Notification Service Elastic Container Registry

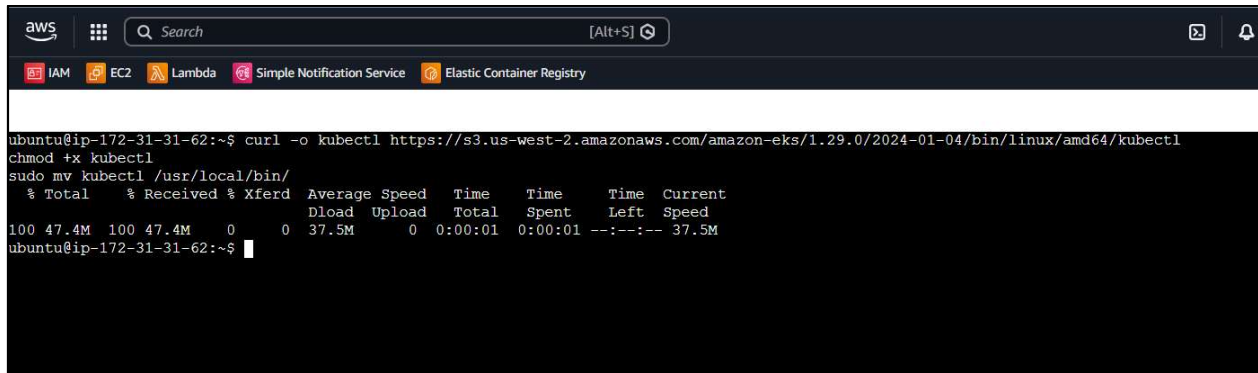
GNU nano 7.2 deployment.yaml *
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: flask-app
  template:
    metadata:
      labels:
        app: flask-app
    spec:
      containers:
      - name: flask-container
        image: 891146182066.dkr.ecr.us-east-2.amazonaws.com/flask-k8s-repo:latest
        ports:
        - containerPort: 5000
]
```

- Create Service YAML
  - `nano service.yaml`

```
GNU nano 7.2 service.yaml *
apiVersion: v1
kind: Service
metadata:
  name: flask-service
spec:
  type: NodePort
  selector:
    app: flask-app
  ports:
    - port: 80
      targetPort: 5000
      nodePort: 30007
```



- Install kubectl
  - `curl -o kubectl https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.0/2024-01-04/bin/linux/amd64/kubectl`
  - `chmod +x kubectl`
  - `sudo mv kubectl /usr/local/bin/`



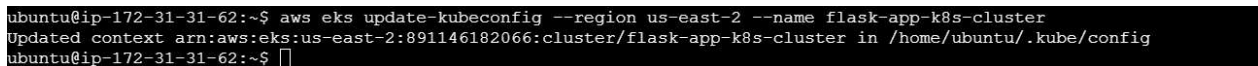
```

aws
IAM EC2 Lambda Simple Notification Service Elastic Container Registry

ubuntu@ip-172-31-31-62:~$ curl -o kubectl https://s3.us-west-2.amazonaws.com/amazon-eks/1.29.0/2024-01-04/bin/linux/amd64/kubectl
chmod +x kubectl
sudo mv kubectl /usr/local/bin/
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100 47.4M  100 47.4M    0     0  37.5M      0  0:00:01  0:00:01 --:--:-- 37.5M
ubuntu@ip-172-31-31-62:~$

```

- Connect to EKS cluster
  - `aws eks update-kubeconfig --region <region_name> --name <cluster_name>`

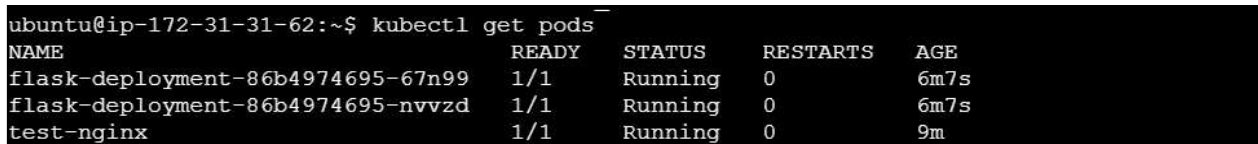


```

ubuntu@ip-172-31-31-62:~$ aws eks update-kubeconfig --region us-east-2 --name flask-app-k8s-cluster
Updated context arn:aws:eks:us-east-2:891146182066:cluster/flask-app-k8s-cluster in /home/ubuntu/.kube/config
ubuntu@ip-172-31-31-62:~$

```

- Test:
  - `kubectl get nodes`

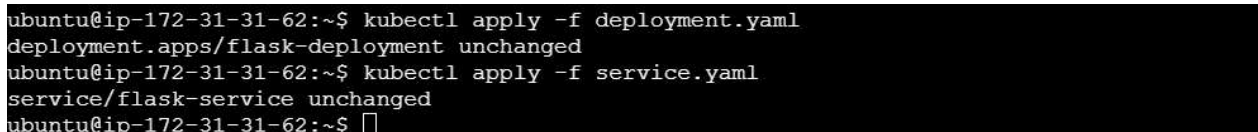


```

ubuntu@ip-172-31-31-62:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
flask-deployment-86b4974695-67n99   1/1     Running   0           6m7s
flask-deployment-86b4974695-nvvzd   1/1     Running   0           6m7s
test-nginx                           1/1     Running   0           9m

```

- Apply configs
  - `kubectl apply -f deployment.yaml`
  - `kubectl apply -f service.yaml`



```

ubuntu@ip-172-31-31-62:~$ kubectl apply -f deployment.yaml
deployment.apps/flask-deployment unchanged
ubuntu@ip-172-31-31-62:~$ kubectl apply -f service.yaml
service/flask-service unchanged
ubuntu@ip-172-31-31-62:~$

```

- Check pods:
  - `kubectl get pods`

```
ubuntu@ip-172-31-31-62:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
flask-deployment-86b4974695-67n99  1/1     Running   0           53m
flask-deployment-86b4974695-nvvzd  1/1     Running   0           53m
test-nginx                          1/1     Running   0           56m
ubuntu@ip-172-31-31-62:~$
```

- Check service:
  - `kubectl get svc`

```
ubuntu@ip-172-31-31-62:~$ kubectl get svc
NAME            TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
flask-service   NodePort    172.20.30.141 <none>         80:30007/TCP     43s
kubernetes      ClusterIP   172.20.0.1    <none>         443/TCP          147m
```

- Get node public IP:
  - `kubectl get nodes -o wide`

```
ubuntu@ip-172-31-31-62:~$ kubectl get nodes -o wide
NAME                                STATUS    ROLES    AGE   VERSION    INTERNAL-IP    EXTERNAL-IP    OS-IMAGE
i-038a3db49ded215b9                Ready    <none>   136m  v1.34.3-eks-3c60543  10.1.11.149    <none>         Bottlerocket (EKS Auto, Standard) 2026.1.28 (aws-k8s-1.
34-standard) 6.12.64
ip-10-1-12-58.us-east-2.compute.internal Ready    <none>   132m  v1.34.2-eks-ecaa3a6  10.1.12.58    <none>         Amazon Linux 2023.10.20260120
6.12.64-87.122.amzn2023.x86_64 containerd://2.1.5
```

- For running application on ELB
- need to correctly map **5000** → **80** → **30007** and then access it via **Node IP**.
- Edit service.yaml properly:

```
apiVersion: v1
kind: Service
metadata:
  name: flask-service
spec:
  type: LoadBalancer
  selector:
    app: flask-app
  ports:
    - port: 80
      targetPort: 5000
```

- Apply:
  - `kubectl apply -f service.yaml`

```
ubuntu@ip-172-31-31-62:~$ kubectl apply -f service.yaml
service/flask-service unchanged
```

- `kubectl get svc flask-service`

```
ubuntu@ip-172-31-31-62:~$ kubectl get svc flask-service
NAME            TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
flask-service   LoadBalancer 172.20.30.141  afa92a4dcee094763aa626d90f4ac029-1408905745.us-east-2.elb.amazonaws.com 80:30007/TCP 60m
ubuntu@ip-172-31-31-62:~$
```

- Check Application on Browser

- [Http://<external-IP>link](http://<external-IP>link)

