

# Deploying Flask Application Using Docker and AWS EC2

## 1. Introduction

- "In this demo, I will showcase how I deployed a Flask application using Docker, AWS Elastic Container Registry (ECR), and EC2. We will build the Docker image, push it to AWS ECR, and run it on an EC2 instance."
- 

## 2. Local Flask Application Setup

- "Step one is to ensure the Flask application runs locally. I tested the endpoint in Postman on localhost (127.0.0.1:5001/recommend)."

The screenshot shows the Postman application interface. The header bar includes 'Postman' and the date 'Fri Dec 27 4:21PM'. The main window displays a collection named 'My Workspace' with a single item 'http://127.0.0.1:5001/recommend'. A POST request is selected, targeting 'http://127.0.0.1:5001/recommend'. The 'Body' tab is active, showing a JSON payload with a key 'title' and value 'spiderman'. The 'Pretty' tab shows the expanded JSON response:

```
1 [  
2   {  
3     "director": "Sam Raimi",  
4     "genres": [  
5       "Action",  
6       "Adventure",  
7       "Fantasy"  
8     ],  
9     "main_cast": [  
10       "Tobey Maguire",  
11       "Kirsten Dunst",  
12       "James Franco",  
13       "Alfred Molina"  
14     ],  
15     "title": "Spider-Man 2",  
16     "w_ratings": 6.64929953353328,  
17     "year": "2004"  
18   },  
19   {  
20     "director": "Sam Raimi",  
21     "genres": [  
22       "Fantasy",  
23       "Action",  
24       "Adventure"  
25     ],  
26     "main_cast": [  
27       "Tobey Maguire",  
28       "Kirsten Dunst",  
29       "James Franco",  
30       "Alfred Molina"  
31     ]  
32   ]
```

The status bar at the bottom indicates a '200 OK' response with 167 ms duration and 1.71 KB size.

### 3. Build a Docker Image

- "Next, I created a Dockerfile to containerize the application. After that, we need to build the Docker image. To do this, navigate to the project directory and run the command to build the Docker image (multi-architecture compatible):"
  - Verify or display the built image using the `docker images` command.
  - You can also open Docker Desktop to showcase the image.

```

fnupratichi@FNUs-Laptop Desktop %
fnupratichi@FNUs-Laptop Desktop % cd "Movie_Recommendation Deployment"
fnupratichi@FNUs-Laptop Movie_Recommendation Deployment % docker build --platform linux/amd64 -t movie_reco .
[+] Building 94.7s (11/11) FINISHED
   => [internal] load build definition from Dockerfile
   => => transferring dockerfile: 960B
   => [internal] load metadata for docker.io/library/python:3.9-slim-buster
   => [auth] library/python:pull token for registry-1.docker.io
   => [internal] load .dockerignore
   => => transferring context: 113B
   => [1/5] FROM docker.io/library/python:3.9-slim-buster@sha256:320a7a4250ba4249f458872adecf92eee88dc6abd2d76dc5c0f01cac9b53990
   => => resolve docker.io/library/python:3.9-slim-buster@sha256:320a7a4250ba4249f458872adecf92eee88dc6abd2d76dc5c0f01cac9b53990
   => sha256:2e1c130fa3ec177a82123374b4c500623959f7983c1dd731ee4a83e1f1b3ff2 3.14MB
   => sha256:84c8c79126f669beec1dcf0f34cd8894471745579c1972d1f4d2226ac14ebfb5363f5fb789ff24d 11.04MB / 11.04MB
   => sha256:8d53da2604a0835f622508d7762fd14d2226ac14ebfb5363f5fb789ff24d 11.04MB / 11.04MB
   => sha256:824416e23a237961c9c5d4f1dfb5295a3c35a7ee528989bfbb8d8e257ec4c 2.78MB / 2.78MB
   => sha256:8b91b8d8687765c6db82668755a3f6dc437bce15a7e4857139e5fc964a7 27.14MB / 27.14MB
   => sha256:2e1c130fa3ec177a82123374b4c500623959f7983c1dd731ee4a83e1f1b3ff2 3.14MB
   => extracting sha256:84c8c79126f669beec1dcf0f34cd8894471745579c1972d1f4d2226ac14ebfb5363f5fb789ff24d
   => extracting sha256:8d53da2604a0835f622508d7762fd14d2226ac14ebfb5363f5fb789ff24d
   => extracting sha256:8b91b8d8687765c6db82668755a3f6dc437bce15a7e4857139e5fc964a7
   => extracting sha256:84c8c79126f669beec1dcf0f34cd8894471745579c1972d1f4d2226ac14ebfb5363f5fb789ff24d
   => extracting sha256:8d53da2604a0835f622508d7762fd14d2226ac14ebfb5363f5fb789ff24d
   => extracting sha256:2e1c130fa3ec177a82123374b4c500623959f7983c1dd731ee4a83e1f1b3ff2 0.0s
   => [internal] load build context
   => => transferring context: 2.33kB
   => [2/5] WORKDIR /app
   => [3/5] RUN apt-get update && apt-get install -y libblas-dev liblapack-dev gfortran build-essential && rm -rf /var/lib/apt/list 21.9s
   => [4/5] COPY . /app
   => [5/5] RUN pip install --no-cache-dir --upgrade pip && pip install --no-cache-dir -r requirements.txt 19.3s
   => exporting to image
   => => exporting layers
   => => exporting manifest sha256:1923ffc8f4e3f2eddb53d9879a9dd9639a4a836e7f4994e3d12e2637c5be122
   => => exporting config sha256:09ee69741lbf7fc03aaeb78cbfce0e8268e3e8e795896786b483e81cfca98766e
   => => exporting attestation manifest sha256:21121a04e9dc53d21db71430fdas38683e6bfbb2663da77a8181379ee7813
   => => exporting manifest list sha256:c72dbfc4e475de42046ba092c3342beac07e298bec177335a34d7794f0a3c
   => => naming to docker.io/library/movie_reco:latest 0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/tvjavaap88jwe18s5a9om0i68
fnupratichi@FNUs-Laptop Movie_Recommendation Deployment %
fnupratichi@FNUs-Laptop Movie_Recommendation Deployment %
fnupratichi@FNUs-Laptop Movie_Recommendation Deployment % docker images
REPOSITORY          TAG      IMAGE ID      CREATED             SIZE
movie_reco          latest   c72bdbfc4e75  About a minute ago  524MB
your-image-name     latest   5c9c9ac42941  5 hours ago       2.22GB
public.ecr.aws/s909g2a9/recommendation    latest   5c9c9ac42941  5 hours ago       2.22GB
flask-app           latest   58c73d664ba7  5 hours ago       1.87GB
moby/buildkit       buildx-stable-1  86c8ad91137  10 days ago      304MB
fnupratichi@FNUs-Laptop Movie_Recommendation Deployment %

```

## 4. Push Docker Image to AWS ECR

- "After building the Docker image, I created an ECR repository to store it. I used the AWS Management Console to create the repository and then authenticated Docker to ECR.

The next steps are:"

- Tag the image
  - Push the image to ECR

```

fnupratichi@FNUs-Laptop Movie_Recommendation Deployment %
fnupratichi@FNUs-Laptop Movie_Recommendation Deployment % aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin public.ecr.aws/s9u9g2a9
Login Succeeded
fnupratichi@FNUs-Laptop Movie_Recommendation Deployment % docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
movie_reco          latest   c72bdbfc4e75  13 minutes ago  524MB
your-image-name     latest   5cc9caa4c2941  5 hours ago   2.22GB
public.ecr.aws/s9u9g2a9/recommendation  latest   5cc9caa4c2941  5 hours ago   2.22GB
flask-app           latest   58c73d664ba7  6 hours ago   1.87GB
moby/buildkit       buildx-stable-1  86c0ad9d1137  18 days ago   304MB
fnupratichi@FNUs-Laptop Movie_Recommendation Deployment % docker tag movie_reco:tag public.ecr.aws/s9u9g2a9/recommendation:latest
Error response from daemon: No such image: movie_reco:tag
fnupratichi@FNUs-Laptop Movie_Recommendation Deployment %
fnupratichi@FNUs-Laptop Movie_Recommendation Deployment % docker tag movie_reco:latest public.ecr.aws/s9u9g2a9/recommendation:latest
fnupratichi@FNUs-Laptop Movie_Recommendation Deployment %
fnupratichi@FNUs-Laptop Movie_Recommendation Deployment % docker push public.ecr.aws/s9u9g2a9/recommendation:latest
The push refers to repository [public.ecr.aws/s9u9g2a9/recommendation]
8b91b8d5577: Pushed
2c9c0587c572: Pushed
8d53da269408: Pushed
84c8c79126f6: Pushed
c1d7e9d31f1: Pushed
3d5f94fa484b: Pushed
824416e23423: Pushed
2e1c130fa3ec: Pushed
500dd7914cb8: Layer already exists
a69a2efb19cc: Pushed
latest: digest: sha256:c72bdbfc4e75de42046ba0c92c33424beac07e298bec177335a34d7794f0a33c size: 856
fnupratichi@FNUs-Laptop Movie_Recommendation Deployment %

```

---

## 5. EC2 Setup

"Now, I'll set up an EC2 instance to pull and run the Docker image. Follow these steps:"

- Launch an EC2 instance with appropriate security groups (allow traffic on the desired port, e.g., 5001)
  - Connect to the instance
  - Configure AWS CLI
- 

## 6. Pull and Run the Docker Image on EC2

"Now, let's pull and run the Docker image on EC2:"

- Pull the image from AWS ECR
- Run the container
- Verify and show the running container
- Test the endpoint in a browser
- Check logs for GET and POST calls

```

Desktop — ec2-user@ip-172-31-47-9:---zsh—195x55
Last login: Thu Dec 26 23:24:01 on ttys000
fnupratichi@FNUS-Laptop ~ % cd Desktop
fnupratichi@FNUS-Laptop Desktop % ssh -i "keypair.pem" ec2-user@ec2-3-80-184-62.compute-1.amazonaws.com
The authenticity of host 'ec2-3-80-184-62.compute-1.amazonaws.com (3.80.184.62)' can't be established.
ED25519 key fingerprint is SHA256:1130caSbgndayTf0L2C1P0zknSFtJGSAiN0hAwA04.
This key is not known by any other name.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-80-184-62.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
#_
~\ _###_ Amazon Linux 2
~~ \####\ 
~~ \##| AL2 End of Life is 2025-06-30.
~~ \#/ -->
~~ V+`-->
~~ / A newer version of Amazon Linux is available!
~~ /`/
~/`/ Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-47-9 ~]$ clear

[ec2-user@ip-172-31-47-9 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-47-9 ~]$ sudo amazon-linux-extras enable docker
 2 httpd_modules           available      [ =1.0 =stable ]
 3 memcached1.5            available      \
  [ =1.5.1 =1.5.16 =1.5.17 ]
 9 R3.4                     available      [ =3.4.3 =stable ]
10 rust1                   available      \
  [ =1.22.1 =1.26.0 =1.26.1 =1.27.2 =1.31.0 =1.38.0
   =stable ]
18 libnvidia-cfg             available      \
  [ =5.0.6.2.15 =5.3.6.1 =stable ]
19 gimp                     available      [ =2.8.22 ]
20 tdocker=latest            enabled       \
  [ =17.12.1 =18.03.1 =18.06.1 =18.09.9 =stable ]
21 mate=desktop1.x           available      \
  [ =1.19.0 =1.20.0 =stable ]
22 GraphicsMagick1.3        available      \
  [ =1.3.29 =1.3.32 =1.3.34 =stable ]
24 epel                     available      [ =7.11 =stable ]
25 testing                  available      [ =1.0 =stable ]
26 ecs                      available      [ =stable ]
27 rcorretto8                available      \
  [ =1.8.0.192 =1.8.0.202 =1.8.0.212 =1.8.0.222 =1.8.0.232
   =1.8.0.242 =stable ]
32 lustre2.10                available      \
  [ =2.18.5 =2.19.8 =stable ]
34 lynis                    available      [ =stable ]
36 BCC                      available      [ =0.x =stable ]

```

```

Desktop — ec2-user@ip-172-31-47-9:---zsh—195x55
[ec2-user@ip-172-31-47-9 ~]$ 
[ec2-user@ip-172-31-47-9 ~]$ sudo docker pull public.ecr.aws/s9u9g2a9/recommendation
Using default tag: latest
latest: Pulling from s9u9g2a9/recommendation
8b91b8d5577: Pull complete
824416e23423: Pull complete
8d53da260488: Pull complete
84c87126f6: Pull complete
2e1c130f93ec: Pull complete
2c9c0587c572: Pull complete
a69ae2fb19c: Pull complete
508dd71e6cbf: Pull complete
508dd71e6cbf: Pull complete
Digest: sha256:c72b0bf4c675de42046ba0e92c33a24beac07e298bec177335a34d779af8a33c
Status: Downloaded newer image for public.ecr.aws/s9u9g2a9/recommendation:latest
public.ecr.aws/s9u9g2a9/recommendation:latest
[ec2-user@ip-172-31-47-9 ~]$ docker images
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/docker.sock: connect: permission denied
[ec2-user@ip-172-31-47-9 ~]$ sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED     SIZE
public.ecr.aws/s9u9g2a9/recommendation    latest      09ee69741bf    17 minutes ago   1.72GB
public.ecr.aws/s9u9g2a9/recommendation    latest      d2c94e258dc8b   2 months ago   13.3KB
[ec2-user@ip-172-31-47-9 ~]$ 
[ec2-user@ip-172-31-47-9 ~]$ 
[ec2-user@ip-172-31-47-9 ~]$ sudo docker run -d -p 8080:8080 public.ecr.aws/s9u9g2a9/recommendation:latest
66ac3fa03fd7e62478df4da49e959b3163109fb9a204a138adade67978e0a
[ec2-user@ip-172-31-47-9 ~]$ sudo docker run -d -p 5001:5001 public.ecr.aws/s9u9g2a9/recommendation:latest
65fea7552777ebf0c61406a9a1cd1be677189b41636f6878941071badc1ee6f4
[ec2-user@ip-172-31-47-9 ~]$ 
[ec2-user@ip-172-31-47-9 ~]$ 
[ec2-user@ip-172-31-47-9 ~]$ sudo docker logs 65fea7652777ebf0c61406a9a1cd1be677189b41636f6878941071badc1ee6f4
[ec2-user@ip-172-31-47-9 ~]$ 
[ec2-user@ip-172-31-47-9 ~]$ sudo docker ps
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.44/containers/json": dial unix /var/run/docker.sock: connect: permission denied
[ec2-user@ip-172-31-47-9 ~]$ sudo docker ps
CONTAINER ID  IMAGE               COMMAND      CREATED     STATUS      PORTS          NAMES
65fea7552777  public.ecr.aws/s9u9g2a9/recommendation:latest "flask run"   48 seconds ago Up 39 seconds  0.0.0.0:5001->5001/tcp, :::5001->5001/tcp   adoring_nash
66ac3fa03fd  public.ecr.aws/s9u9g2a9/recommendation:latest "flask run"   58 seconds ago Up 57 seconds  5001/tcp, 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   sad_hellman
[ec2-user@ip-172-31-47-9 ~]$ sudo docker logs 65fea7552777ebf0c61406a9a1cd1be677189b41636f6878941071badc1ee6f4
2024-12-27 05:21:13,125 - INFO - Starting preprocessing
2024-12-27 05:21:13,153 - INFO - Preprocessing completed
  id ... normalized_title
0  862 ... toystry
1  8844 ... jumanji
2  1862 ... fatherofthebridepartii
3   949 ... heat
4  9091 ... sunderneath

[5 rows x 14 columns]
(0, 2)   0.6844044102260063
(0, 3)   0.4001094201451891
(0, 7)   0.6898105845643373
(0, 6943) 0.24048530579592217

```

```

● ● ● Desktop — ec2-user@ip-172-31-47-9:~ - zsh - 195x55
(4553, 3894) 0.3037894214846056
(4553, 3532) 0.3037894214846056
(4553, 492) 0.3037894214846056
(4553, 3957) 0.3037894214846056
(4553, 7326) 0.6874188429692112
(4553, 9519) 0.6449561977617007
(4553, 18032) 0.76422195384631187
(4554, 1) 0.10000000000000002
(4554, 3458) 0.24296046530885175
(4554, 1347) 0.2757172464921127
(4554, 5822) 0.3284377430839536
(4554, 1388) 0.34009135776867165
(4554, 3679) 0.38866977229549435
(4554, 7166) 0.405520868491821
(4554, 2396) 0.381131608797268615
(4554, 1959) 0.396865908451150806
(4554, 8784) 0.40936091816495286
(4554, 9854) 0.6166546486915114
(4554, 1071) 0.6166546486915114
* Serving Flask app 'app.py'
* Debug mode: off
2024-12-27 05:21:13.623 - INFO - [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://127.0.0.1:5001
* Running on http://172.17.0.2:5001
2024-12-27 05:21:13.623 - INFO - Press CTRL+C to quit
[ec2-user@ip-172-31-47-9 ~]$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
65fe7652777 public.ecr.aws/s9u9g2a9/recommendation:latest "flask run" About a minute ago Up About a minute 0.0.0.0:5001->5001/tcp, :::5001->5001/tcp adoring_nash
66ac3faf03fd public.ecr.aws/s9u9g2a9/recommendation:latest "flask run" About a minute ago Up About a minute 5001/tcp, 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp sad_hellman
[ec2-user@ip-172-31-47-9 ~]$ sudo docker stop 66ac3faf03fd
[ec2-user@ip-172-31-47-9 ~]$ sudo docker rm 66ac3faf03fd
[ec2-user@ip-172-31-47-9 ~]$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
65fe7652777 public.ecr.aws/s9u9g2a9/recommendation:latest "flask run" About a minute ago Up About a minute 0.0.0.0:5001->5001/tcp, :::5001->5001/tcp adoring_nash
[ec2-user@ip-172-31-47-9 ~]$ sudo docker logs 65fe7652777
2024-12-27 05:21:32.079 - INFO - Preprocessing
2024-12-27 05:21:32.186 - INFO - Preprocessing completed
      id ... normalized_title
0  862 ... toystory
1  8844 ... jumanji
2  11862 ... fatherofthebridepartii
3   949 ... heat
4  9091 ... suddendeath
[5 rows x 14 columns]
(0, 2) 0.6844844132460803
(0, 3) 0.695105645453373
(0, 7) 0.695105645453373
(0, 6943) 0.24048538679592217
(0, 2934) 0.3249925722556408
(0, 6916) 0.2996498663097371
(0, 191) 0.30827459906026736

```

```

● ● ● Desktop — ec2-user@ip-172-31-47-9:~ - zsh - 195x55
(4554, 9382) 0.6166546406915114
* Serving Flask app 'app.py'
* Debug mode: off
2024-12-27 05:21:32.481 - INFO - [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://127.0.0.1:5001
* Running on http://172.17.0.3:5001
2024-12-27 05:21:32.481 - INFO - Press CTRL+C to quit
2024-12-27 05:26:17.364 - INFO - 172.17.0.1 - [27/Dec/2024 05:26:48] "GET /api HTTP/1.1" 404 -
2024-12-27 05:26:54.811 - INFO - 172.17.0.1 - [27/Dec/2024 05:26:54] "GET /recommend HTTP/1.1" 400 -
2024-12-27 05:27:25.361 - INFO - Normalized Input Title: inception
2024-12-27 05:27:25.363 - INFO - Movie Found at index: 2817
[[3639, 0.630056545508521], (2184, 0.5212238378645927), (1400, 0.4946571623293473), (2398, 0.4403608929002686)]
2024-12-27 05:27:25.680 - INFO - 172.17.0.1 - [27/Dec/2024 05:27:18] "GET /recommend?movie_title=inception HTTP/1.1" 400 -
2024-12-27 05:33:25.527 - INFO - Normalized Input Title: spiderman
2024-12-27 05:33:25.528 - INFO - Movie Found at index: 1395
[[1762, 0.850356085644348], (2244, 0.840207823237872), (3391, 0.5586992106541213), (348, 0.5078741598998614), (387, 0.5078741598998614)]
2024-12-27 05:33:25.845 - INFO - 73.132.11.138 - [27/Dec/2024 05:33:25] "GET /recommend?title=spiderman HTTP/1.1" 200 -
2024-12-27 05:33:25.193 - INFO - 73.132.11.138 - [27/Dec/2024 05:33:35] "POST /recommend/ HTTP/1.1" 404 -
2024-12-27 05:34:17.519 - INFO - Normalized Input Title: joker
2024-12-27 05:34:17.519 - INFO - Movie Found at index: 185
2024-12-27 05:34:17.522 - INFO - Normalized Input Title: batman
2024-12-27 05:34:23.947 - INFO - Normalized Input Title: batman
2024-12-27 05:34:23.949 - INFO - Movie Found at index: 185
[[444, 0.7396697772287434], (689, 0.62505653081206594), (3197, 0.5582266949384866), (453, 0.5499302638028892), (1650, 0.5216863429592632)]
2024-12-27 05:34:24.265 - INFO - 73.132.11.138 - [27/Dec/2024 05:34:24] "POST /recommend HTTP/1.1" 200 -
2024-12-27 05:34:28.967 - INFO - Normalized Input Title: spiderman
2024-12-27 05:34:38.968 - INFO - Movie Found at index: 1395
[[1762, 0.850356085644348], (2244, 0.840207823237872), (3391, 0.5586992106541213), (348, 0.5078741598998614), (387, 0.5078741598998614)]
2024-12-27 05:34:39.281 - INFO - 73.132.11.138 - [27/Dec/2024 05:34:39] "GET /recommend?title=spiderman HTTP/1.1" 200 -
2024-12-27 05:34:39.288 - INFO - 73.132.11.138 - [27/Dec/2024 05:34:39] "GET /favicon.ico HTTP/1.1" 404 -
2024-12-27 05:34:49.559 - INFO - 73.132.11.138 - [27/Dec/2024 05:34:49] "GET /recommend HTTP/1.1" 400 -
2024-12-27 05:34:47.573 - INFO - 73.132.11.138 - [27/Dec/2024 05:34:47] "GET / HTTP/1.1" 200 -
2024-12-27 05:34:53.784 - INFO - Normalized Input Title: ironman
2024-12-27 05:34:53.784 - INFO - Movie Found at index: 2415
[[2787, 0.7485933847782422], (2023, 0.55996408589285643), (2929, 0.5269702964617863), (3389, 0.43326266764366483), (1509, 0.4310288781599071)]
2024-12-27 05:36:54.182 - INFO - 73.132.11.138 - [27/Dec/2024 05:36:54] "POST /recommend HTTP/1.1" 200 -
2024-12-27 05:37:04.646 - INFO - Normalized Input Title: batman
2024-12-27 05:37:04.646 - INFO - Movie Found at index: 185
[[444, 0.7396697772287434], (689, 0.62505653081206594), (3197, 0.5582266949384866), (453, 0.5499302638028892), (1650, 0.5216863429592632)]
2024-12-27 05:37:35.839 - INFO - 73.132.11.138 - [27/Dec/2024 05:37:35] "GET / HTTP/1.1" 200 -
2024-12-27 05:38:50.319 - INFO - 73.132.11.138 - [27/Dec/2024 05:38:50] "GET /recommend HTTP/1.1" 200 -
2024-12-27 05:43:45.974 - INFO - Normalized Input Title: ironman
2024-12-27 05:43:45.975 - INFO - Movie Found at index: 2415
[[2787, 0.7485933847782422], (2023, 0.55996408589285643), (2929, 0.5269702964617863), (3389, 0.43326266764366483), (1509, 0.4310288781599071)]
2024-12-27 05:43:46.292 - INFO - 73.132.11.138 - [27/Dec/2024 05:43:46] "GET /recommend?title=ironman HTTP/1.1" 200 -
2024-12-27 05:43:50.213 - INFO - Normalized Input Title: ironman
2024-12-27 05:43:50.213 - INFO - Movie Found at index: 2415
[[2787, 0.7485933847782422], (2023, 0.55996408589285643), (2929, 0.5269702964617863), (3389, 0.43326266764366483), (1509, 0.4310288781599071)]

```

## 7. Conclusion

- "We've successfully deployed the Flask-based Movie Recommendation API using Docker and AWS services. The service is now live and accessible via the EC2 public IP."
  - "This setup ensures scalability and ease of deployment, making it ideal for production-ready applications."
-