Objective

The main objective of this assignment is to demonstrate hands-on knowledge of:

- AWS EC2
- Docker & Docker Compose
- Application and Database deployment
- Basic DevOps workflow

I deployed a **Python Flask application** connected to a **MySQL database** on an **AWS EC2 instance** using containerization.

This setup ensures:

- Portability
- Isolation of services
- Easy deployment and scalability

Why Docker Was Used

Docker allows applications to run in isolated environments called containers.

Benefits:

- No dependency issues
- Same behavior in all environments
- Easy deployment on cloud servers

In this project:

- Flask runs in one container
- MySQL runs in another container

Why Docker Compose Was Used

Docker Compose is used when multiple containers are required.

Advantages:

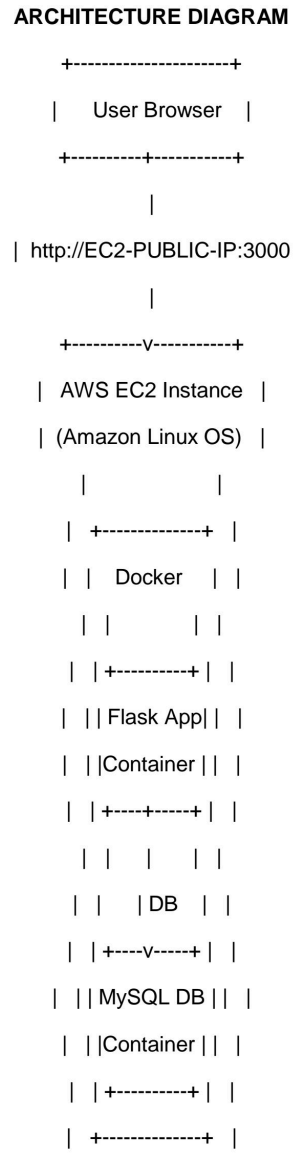
- Single command to start everything
- Automatic networking between containers
- Dependency management (DB starts before app)

Command used:

```
docker compose up -d
```

Architecture Overview

User accesses the application through a browser using the EC2 public IP. The EC2 instance runs Docker, which hosts two containers: one for the Flask application and one for the MySQL database. Both containers communicate through a Docker network.



AWS EC2 Setup Explanation

An **Amazon Linux EC2 instance** was created.

Configuration:

- Instance type: Free-tier eligible
- Public IP enabled
- Security group rules:
  - Port 22 → SSH access
  - Port 3000 → Application access

This allows users to access the application from a bro

Application & Database Communication

The Flask application connects to the MySQL database using **Docker service name**.

Why not `localhost`?

- Each container has its own network
- `localhost` refers to the container itself
- Docker provides internal DNS

Correct approach:

```
host="db"
```

Deployment Process

Steps followed:

1. Built Docker image for Flask application
2. Pulled MySQL image
3. Created Docker network automatically using Compose
4. Started containers in detached mode
5. Verified services using logs and browser

Verification & Testing

Deployment was verified by:

- Checking running containers:

```
docker ps
```

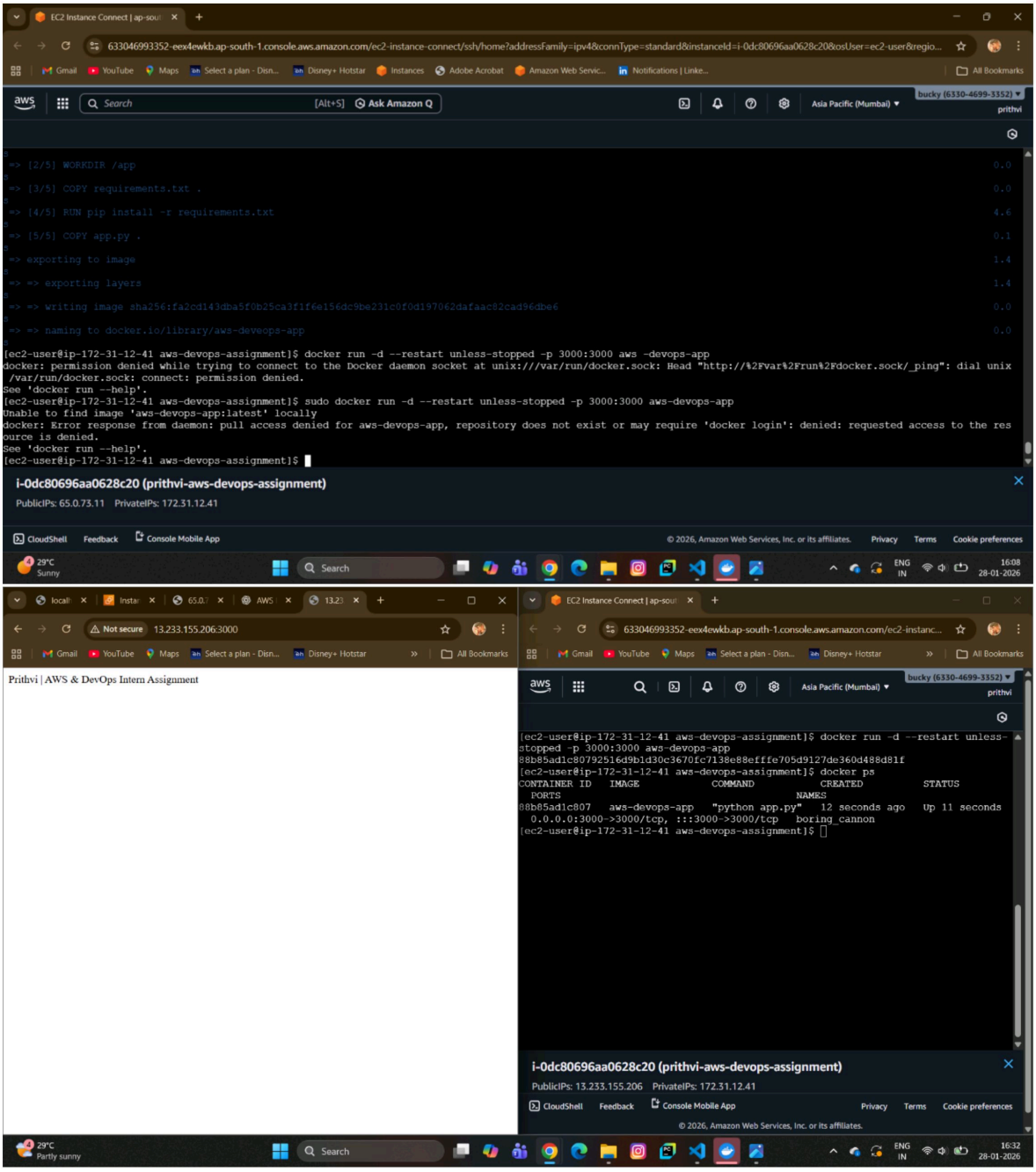
- Checking application logs
- Accessing the application via browser:

Implementation Steps

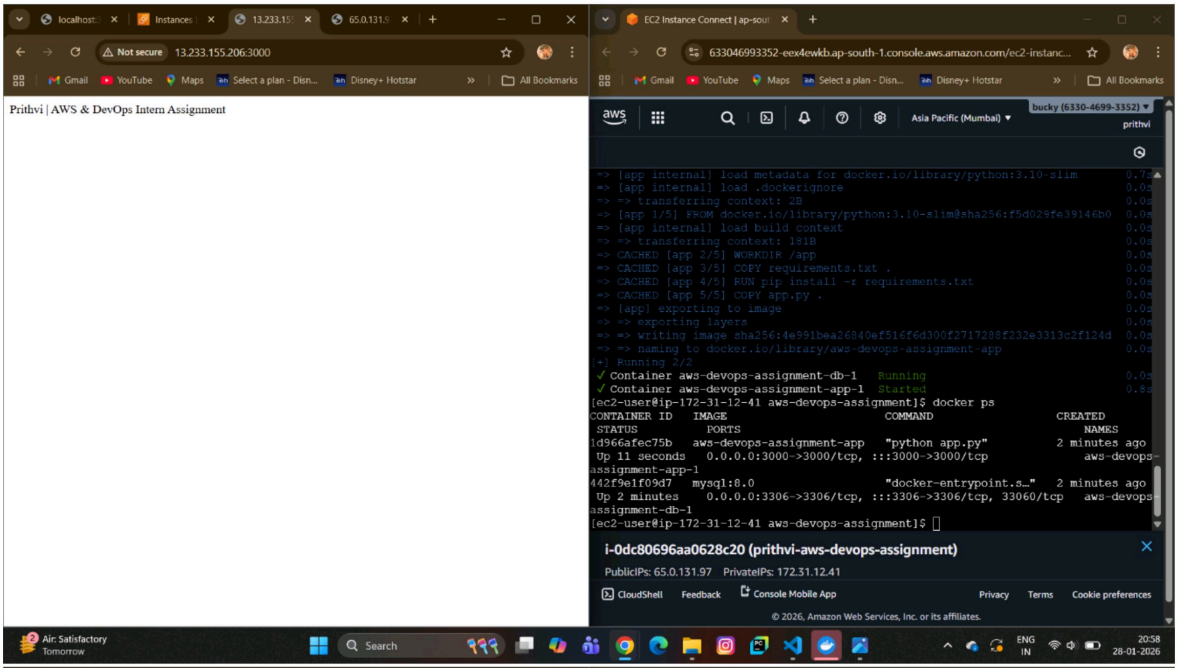
1. Created an AWS EC2 instance and configured security group rules for required ports.
2. Installed Docker and Docker Compose on the EC2 instance.
3. Developed a Flask application to connect with a MySQL database.
4. Containerized the Flask application using a Dockerfile.
5. Used Docker Compose to run Flask and MySQL containers together.
6. Verified successful database connectivity.
7. Accessed the application using EC2 public IP.

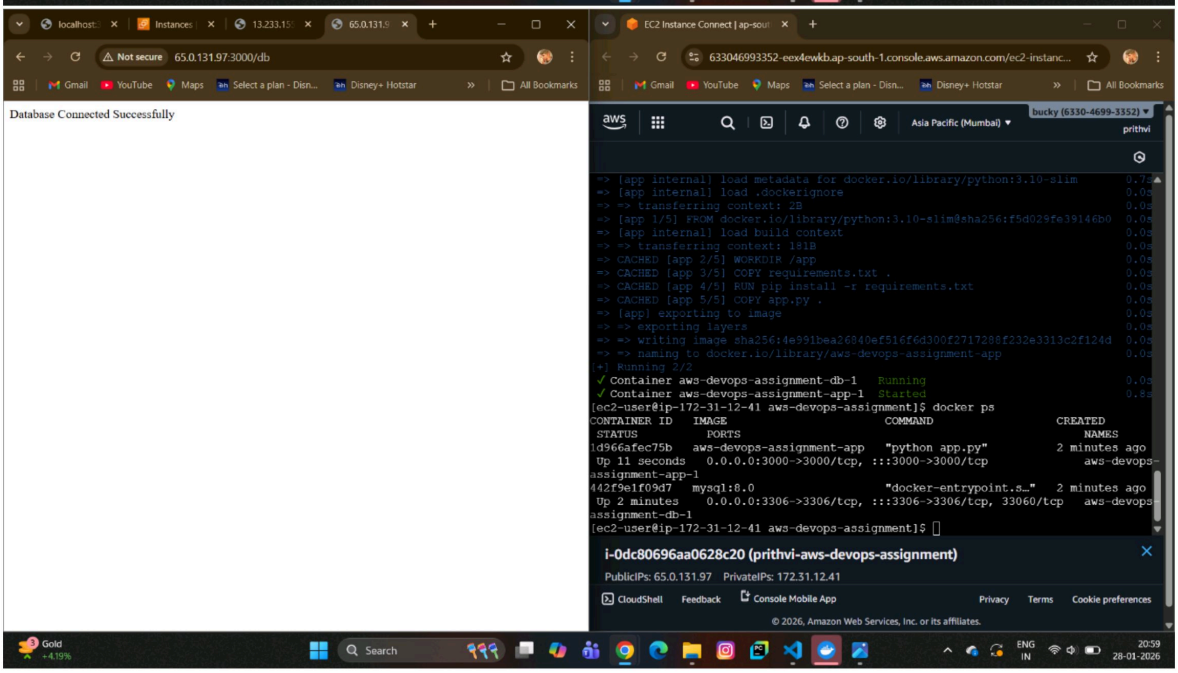
Deployment Screenshots

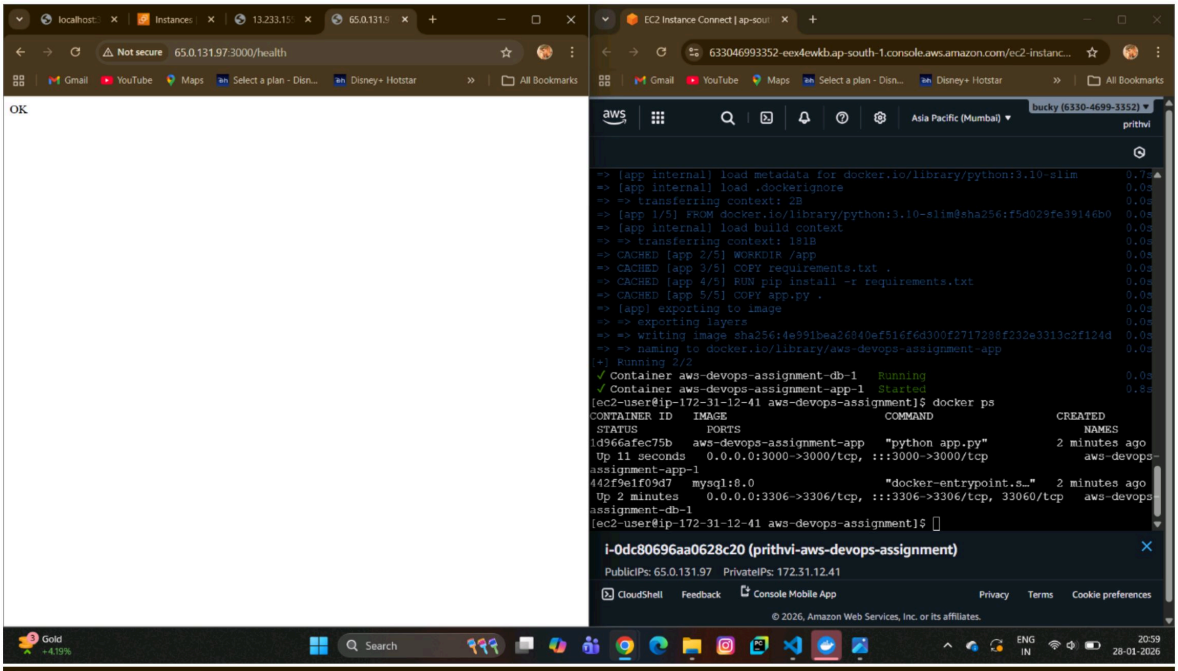
Docker containers running successfully on AWS EC2 instance.

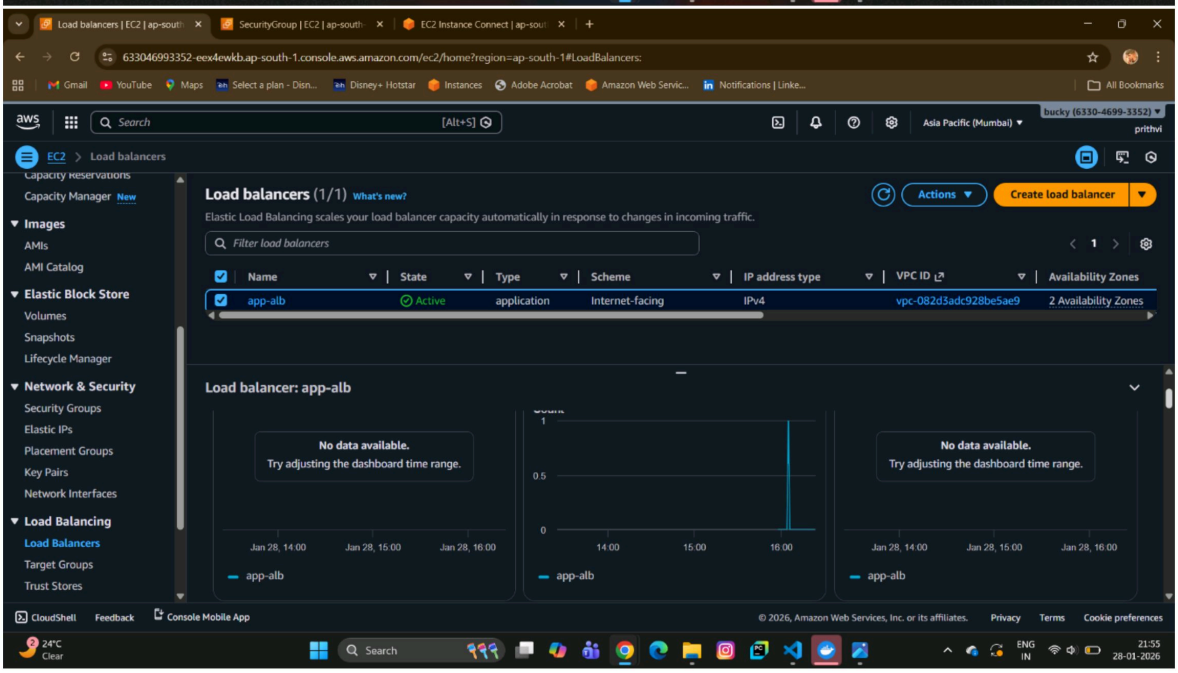

EC2 Instance Connect | ap-south-1
633046993352-eex4ewkb.ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0dc80696aa0628c20&osUser=ec2-user&regio...
Search [Alt+S] Ask Amazon Q
Asia Pacific (Mumbai)
bucky (6330-4699-3352)
prithvi
=> [2/5] WORKDIR /app 0.0
=> [3/5] COPY requirements.txt . 0.0
=> [4/5] RUN pip install -r requirements.txt 4.6
=> [5/5] COPY app.py . 0.1
=> exporting to image 1.4
=> => exporting layers 1.4
=> => writing image sha256:fa2cd143dba5f0b25ca3f1f6e156dc9be231c0f0d197062dafaac82cad96dbe6 0.0
=> => naming to docker.io/library/aws-deveops-app 0.0
[ec2-user@ip-172-31-12-41 aws-devops-assignment]$ docker run -d --restart unless-stopped -p 3000:3000 aws -devops-app
docker: 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.
See 'docker run --help'.
[ec2-user@ip-172-31-12-41 aws-devops-assignment]$ sudo docker run -d --restart unless-stopped -p 3000:3000 aws-devops-app
Unable to find image 'aws-devops-app:latest' locally
docker: Error response from daemon: pull access denied for aws-devops-app, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.
[ec2-user@ip-172-31-12-41 aws-devops-assignment]$
i-0dc80696aa0628c20 (prithvi-aws-devops-assignment)
PublicIPs: 65.0.73.11 PrivateIPs: 172.31.12.41
CloudShell Feedback Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
29°C Sunny
Search
ENG IN
16:08 28-01-2026

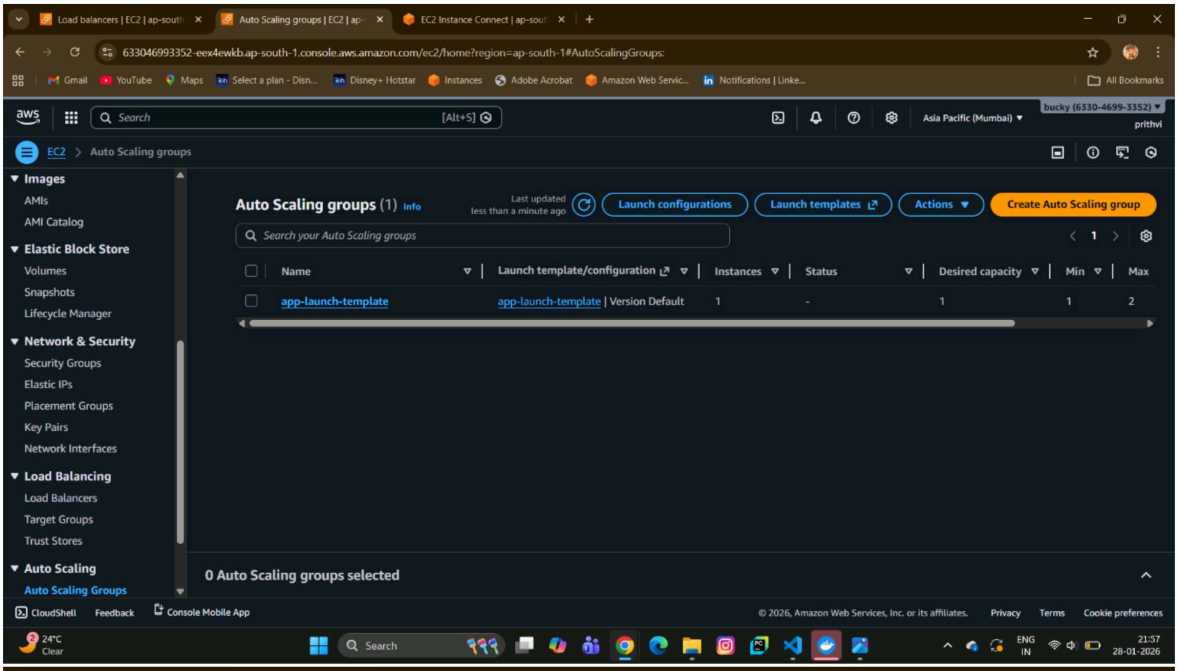
Not secure 13.233.155.206:3000
Prithvi | AWS & DevOps Intern Assignment
EC2 Instance Connect | ap-south-1
633046993352-eex4ewkb.ap-south-1.console.aws.amazon.com/ec2-instanc...
Asia Pacific (Mumbai)
bucky (6330-4699-3352)
prithvi
[ec2-user@ip-172-31-12-41 aws-devops-assignment]$ docker run -d --restart unless-stopped -p 3000:3000 aws-devops-app
88b85ad1c80792516d9b1d30c3670fc7138e88efffe705d9127de360d488d81f
[ec2-user@ip-172-31-12-41 aws-devops-assignment]$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
88b85ad1c807 aws-devops-app "python app.py" 12 seconds ago Up 11 seconds 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp boring_cannon
[ec2-user@ip-172-31-12-41 aws-devops-assignment]$
i-0dc80696aa0628c20 (prithvi-aws-devops-assignment)
PublicIPs: 13.233.155.206 PrivateIPs: 172.31.12.41
CloudShell Feedback Console Mobile App
Privacy Terms Cookie preferences
© 2026, Amazon Web Services, Inc. or its affiliates.
29°C Partly sunny
Search
ENG IN
16:32 28-01-2026

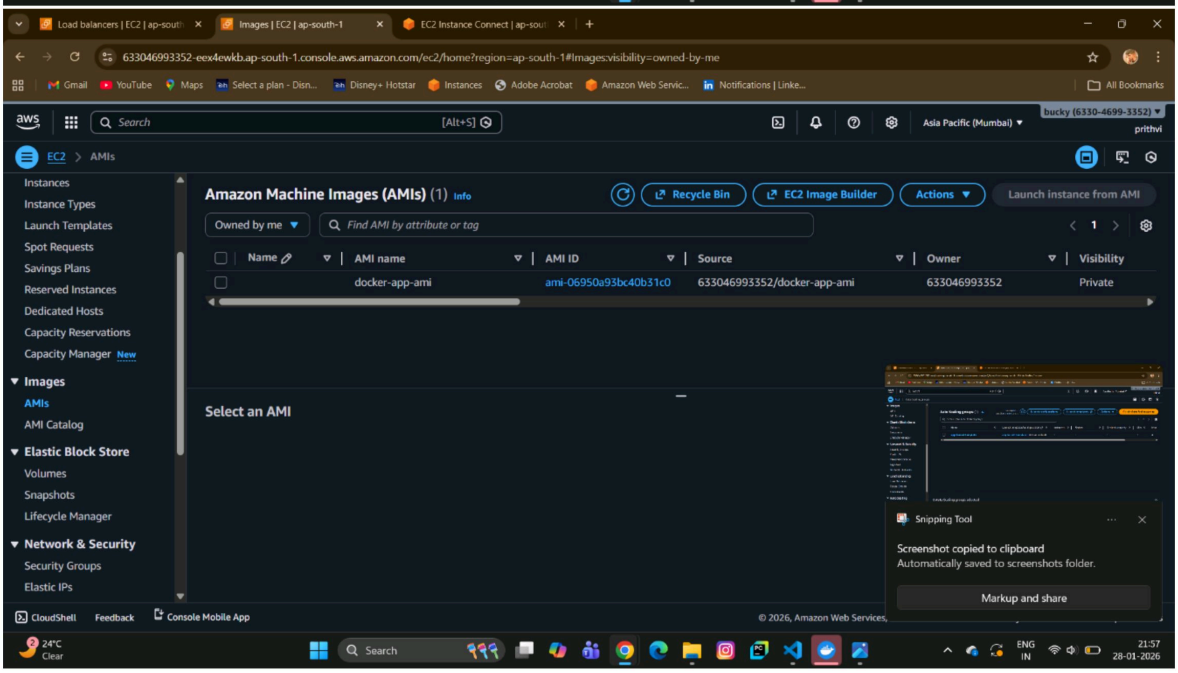

Not secure 13.233.155.206:3000
Prithvi | AWS & DevOps Intern Assignment
EC2 Instance Connect | ap-south-1
633046993352-eex4ewkb.ap-south-1.console.aws.amazon.com/ec2-instanc...
Asia Pacific (Mumbai)
bucky (6330-4699-3352)
prithvi
=> [app internal] load metadata for docker.io/library/python:3.10-slim
=> [app internal] load .dockerignore
=> => transferring context: 2B
=> [app 1/5] FROM docker.io/library/python:3.10-slim@sha256:f5d029fe39146b0
=> [app internal] load build context
=> => transferring context: 181B
=> CACHED [app 2/5] WORKDIR /app
=> CACHED [app 3/5] COPY requirements.txt .
=> CACHED [app 4/5] RUN pip install -r requirements.txt
=> CACHED [app 5/5] COPY app.py .
=> [app] exporting to image
=> => exporting layers
=> => writing image sha256:4e991bea26040ef516f6d300f2717288f232e3313c2f124d
=> => naming to docker.io/library/aws-devops-assignment-app
[+] Running 2/2
✓ Container aws-devops-assignment-db-1 Running
✓ Container aws-devops-assignment-app-1 Started
[ec2-user@ip-172-31-12-41 aws-devops-assignment]$ docker ps
CONTAINER ID IMAGE COMMAND CREATED
STATUS PORTS NAMES
1d966afec75b aws-devops-assignment-app "python app.py" 2 minutes ago
Up 11 seconds 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp aws-devops-assignment-app-1
442f9e1f09d7 mysql:8.0 "docker-entrypoint.s…" 2 minutes ago
Up 2 minutes 0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp aws-devops-assignment-db-1
[ec2-user@ip-172-31-12-41 aws-devops-assignment]$
i-0dc80696aa0628c20 (prithvi-aws-devops-assignment)
PublicIPs: 65.0.131.97 PrivateIPs: 172.31.12.41
CloudShell Feedback Console Mobile App Privacy Terms Cookie preferences
© 2026, Amazon Web Services, Inc. or its affiliates.
Air: Satisfactory Tomorrow
Search
ENG IN
20:58 28-01-2026

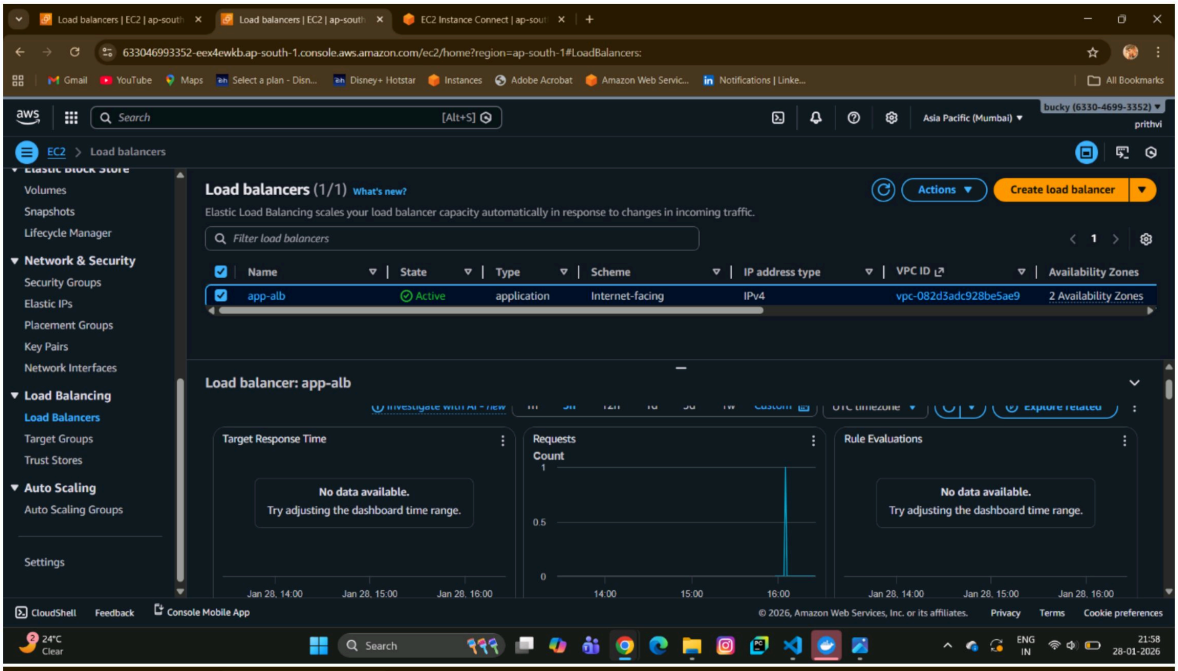

Not secure 65.0.131.97:3000/db
Database Connected Successfully
EC2 Instance Connect | ap-south-1
633046993352-eex4ewkb.ap-south-1.console.aws.amazon.com/ec2-instanc...
Asia Pacific (Mumbai)
bucky (6330-4699-3352)
prithvi
=> [app internal] load metadata for docker.io/library/python:3.10-slim
=> [app internal] load .dockerignore
=> => transferring context: 2B
=> [app 1/5] FROM docker.io/library/python:3.10-slim@sha256:f5d029fe39146b0
=> [app internal] load build context
=> => transferring context: 181B
=> CACHED [app 2/5] WORKDIR /app
=> CACHED [app 3/5] COPY requirements.txt .
=> CACHED [app 4/5] RUN pip install -r requirements.txt
=> CACHED [app 5/5] COPY app.py .
=> [app] exporting to image
=> => exporting layers
=> => writing image sha256:4e991bea26040ef516f6d300f2717288f232e3313c2f124d
=> => naming to docker.io/library/aws-devops-assignment-app
[+] Running 2/2
✓ Container aws-devops-assignment-db-1 Running
✓ Container aws-devops-assignment-app-1 Started
[ec2-user@ip-172-31-12-41 aws-devops-assignment]$ docker ps
CONTAINER ID IMAGE COMMAND CREATED
STATUS PORTS NAMES
1d966afec75b aws-devops-assignment-app "python app.py" 2 minutes ago
Up 11 seconds 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp aws-devops-assignment-app-1
442f9e1f09d7 mysql:8.0 "docker-entrypoint.s…" 2 minutes ago
Up 2 minutes 0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp aws-devops-assignment-db-1
[ec2-user@ip-172-31-12-41 aws-devops-assignment]$
i-0dc80696aa0628c20 (prithvi-aws-devops-assignment)
PublicIPs: 65.0.131.97 PrivateIPs: 172.31.12.41
CloudShell Feedback Console Mobile App Privacy Terms Cookie preferences
© 2026, Amazon Web Services, Inc. or its affiliates.
Gold +4.19%
Search
ENG IN
20:59 28-01-2026

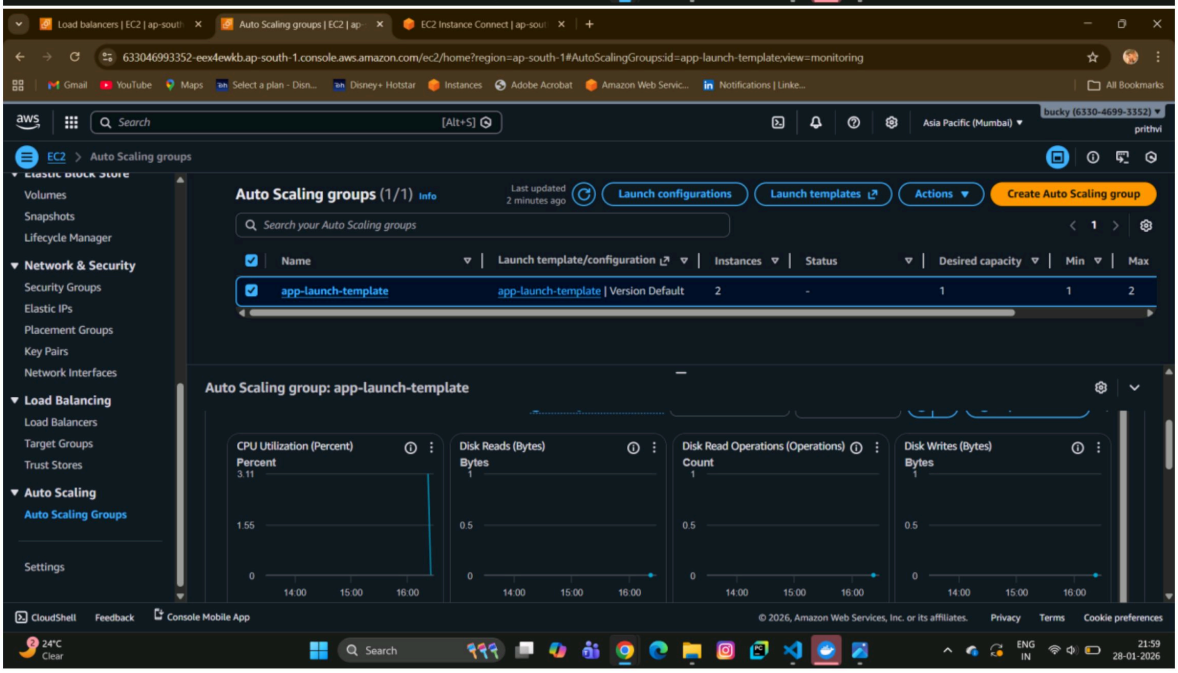

Not secure 65.0.131.97:3000/health
OK
EC2 Instance Connect | ap-south-1
633046993352-eex4ewkb.ap-south-1.console.aws.amazon.com/ec2-instanc...
Asia Pacific (Mumbai)
bucky (6330-4699-3352)
prithvi
=> [app internal] load metadata for docker.io/library/python:3.10-slim
=> [app internal] load .dockerignore
=> => transferring context: 2B
=> [app 1/5] FROM docker.io/library/python:3.10-slim@sha256:f5d029fe39146b0
=> [app internal] load build context
=> => transferring context: 181B
=> CACHED [app 2/5] WORKDIR /app
=> CACHED [app 3/5] COPY requirements.txt .
=> CACHED [app 4/5] RUN pip install -r requirements.txt
=> CACHED [app 5/5] COPY app.py .
=> [app] exporting to image
=> => exporting layers
=> => writing image sha256:4e991bea26040ef516f6d300f2717208f232e3313c2f124d
=> => naming to docker.io/library/aws-devops-assignment-app
+] Running 2/2
✓ Container aws-devops-assignment-db-1 Running
✓ Container aws-devops-assignment-app-1 Started
[ec2-user@ip-172-31-12-41 aws-devops-assignment]$ docker ps
CONTAINER ID IMAGE COMMAND CREATED
STATUS PORTS NAMES
1d966afec75b aws-devops-assignment-app "python app.py" 2 minutes ago
Up 11 seconds 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp aws-devops-
assignment-app-1
442f9e1f09d7 mysql:8.0 "docker-entrypoint.s…" 2 minutes ago
Up 2 minutes 0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp aws-devops
assignment-db-1
[ec2-user@ip-172-31-12-41 aws-devops-assignment]$
i-0dc80696aa0628c20 (prithvi-aws-devops-assignment)
PublicIPs: 65.0.131.97 PrivateIPs: 172.31.12.41
CloudShell Feedback Console Mobile App
Privacy Terms Cookie preferences
© 2026, Amazon Web Services, Inc. or its affiliates.
Gold +4.19%
Search
ENG IN
20:59 28-01-2026

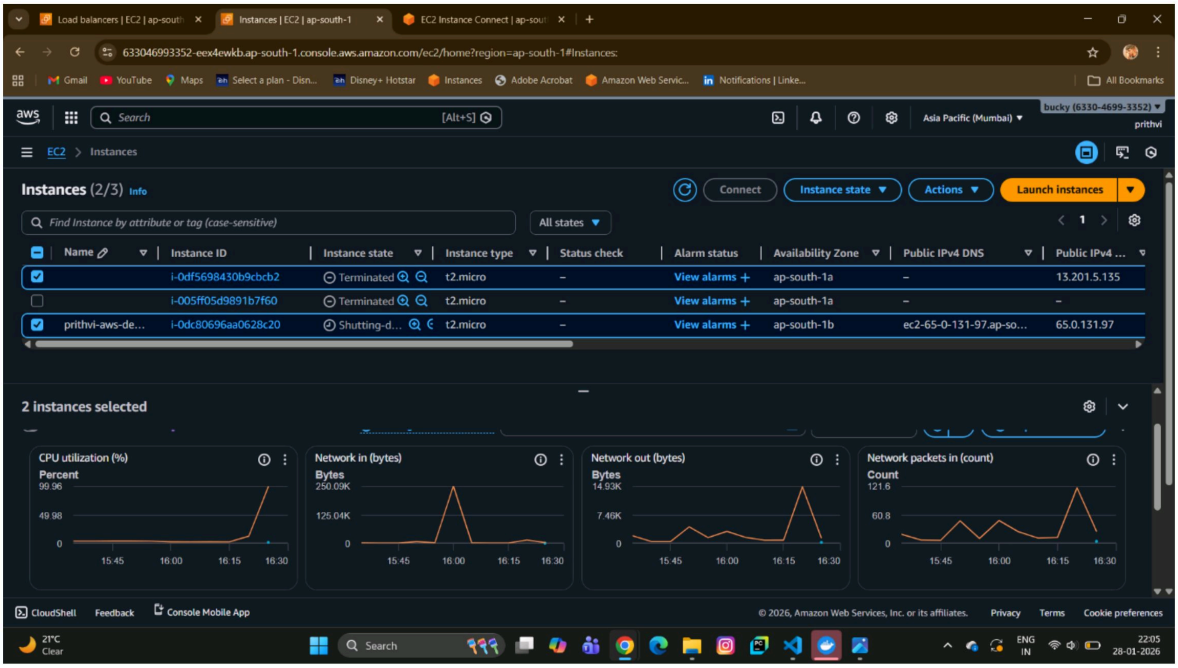

Load balancers | EC2 | ap-south-1
SecurityGroup | EC2 | ap-south-1
EC2 Instance Connect | ap-south-1
633046993352-eex4ewkb.ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LoadBalancers:
Search [Alt+S]
Asia Pacific (Mumbai)
bucky (6330-4699-3352)
prithvi
EC2 > Load balancers
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores
Load balancers (1/1) What's new?
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.
Actions
Create load balancer
Filter load balancers
Name State Type Scheme IP address type VPC ID Availability Zones
app-alb Active application Internet-facing IPv4 vpc-082d3adc928be5ae9 2 Availability Zones
Load balancer: app-alb
No data available.
Try adjusting the dashboard time range.
Jan 28, 14:00
Jan 28, 15:00
Jan 28, 16:00
app-alb
1
0.5
0
14:00
15:00
16:00
app-alb
No data available.
Try adjusting the dashboard time range.
Jan 28, 14:00
Jan 28, 15:00
Jan 28, 16:00
app-alb
CloudShell Feedback Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
24°C Clear
Search
ENG IN
21:55 28-01-2026


Auto Scaling groups | EC2 | ap-
EC2 Instance Connect | ap-sout
633046993352-eex4ewkb.ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#AutoScalingGroups:
Search
[Alt+S]
Asia Pacific (Mumbai)
bucky (6330-4699-3352)
prithvi
EC2 > Auto Scaling groups
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores
Auto Scaling
Auto Scaling Groups
Auto Scaling groups (1) Info
Last updated less than a minute ago
Launch configurations
Launch templates
Actions
Create Auto Scaling group
Search your Auto Scaling groups
Name
Launch template/configuration
Instances
Status
Desired capacity
Min
Max
app-launch-template
app-launch-template | Version Default
1
-
1
1
2
0 Auto Scaling groups selected
CloudShell
Feedback
Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
24°C
Clear
Search
ENG
IN
21:57
28-01-2026


Images | EC2 | ap-south-1
EC2 Instance Connect | ap-sout
633046993352-eex4ewkb.ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Images:visibility=owned-by-me
Search
[Alt+S]
Asia Pacific (Mumbai)
bucky (6330-4699-3352)
prithvi
EC2 > AMIs
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Amazon Machine Images (AMIs) (1) Info
Recycle Bin
EC2 Image Builder
Actions
Launch instance from AMI
Owned by me
Find AMI by attribute or tag
Name
AMI name
AMI ID
Source
Owner
Visibility
docker-app-ami
ami-06950a93bc40b31c0
633046993352/docker-app-ami
633046993352
Private
Select an AMI
Snipping Tool
Screenshot copied to clipboard
Automatically saved to screenshots folder.
Markup and share
CloudShell
Feedback
Console Mobile App
24°C
Clear
Search
ENG
IN
21:57
28-01-2026


Load balancers | EC2 | ap-south-1
633046993352-eex4ewkb.ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LoadBalancers:
Search
Asia Pacific (Mumbai)
bucky (6330-4699-3352)
prithvi
EC2 > Load balancers
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores
Auto Scaling
Auto Scaling Groups
Settings
Load balancers (1/1) What's new?
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.
Actions
Create load balancer
Filter load balancers
Name
State
Type
Scheme
IP address type
VPC ID
Availability Zones
app-alb
Active
application
Internet-facing
IPv4
vpc-082d3adc928be5ae9
2 Availability Zones
Load balancer: app-alb
Target Response Time
No data available.
Try adjusting the dashboard time range.
Requests
Count
Rule Evaluations
No data available.
Try adjusting the dashboard time range.
Jan 28, 14:00
Jan 28, 15:00
Jan 28, 16:00
CloudShell
Feedback
Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
21:58
28-01-2026


Auto Scaling groups | EC2 | ap-south-1
633046993352-eex4ewkb.ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#AutoScalingGroups:id=app-launch-template;view=monitoring
EC2 > Auto Scaling groups
Auto Scaling groups (1/1) Info
Last updated 2 minutes ago
Launch configurations
Launch templates
Actions
Create Auto Scaling group
Search your Auto Scaling groups
Name
Launch template/configuration
Instances
Status
Desired capacity
Min
Max
app-launch-template
app-launch-template | Version Default
2
-
1
1
2
Auto Scaling group: app-launch-template
CPU Utilization (Percent)
Percent
Disk Reads (Bytes)
Bytes
Disk Read Operations (Operations)
Count
Disk Writes (Bytes)
Bytes
14:00
15:00
16:00
CloudShell
Feedback
Console Mobile App
© 2026, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
21:59
28-01-2026



EC2 instance and security group configuration.

Challenges Faced & Fixes

❌ Port already in use

✔ Stopped unused containers

❌ MySQL connection error

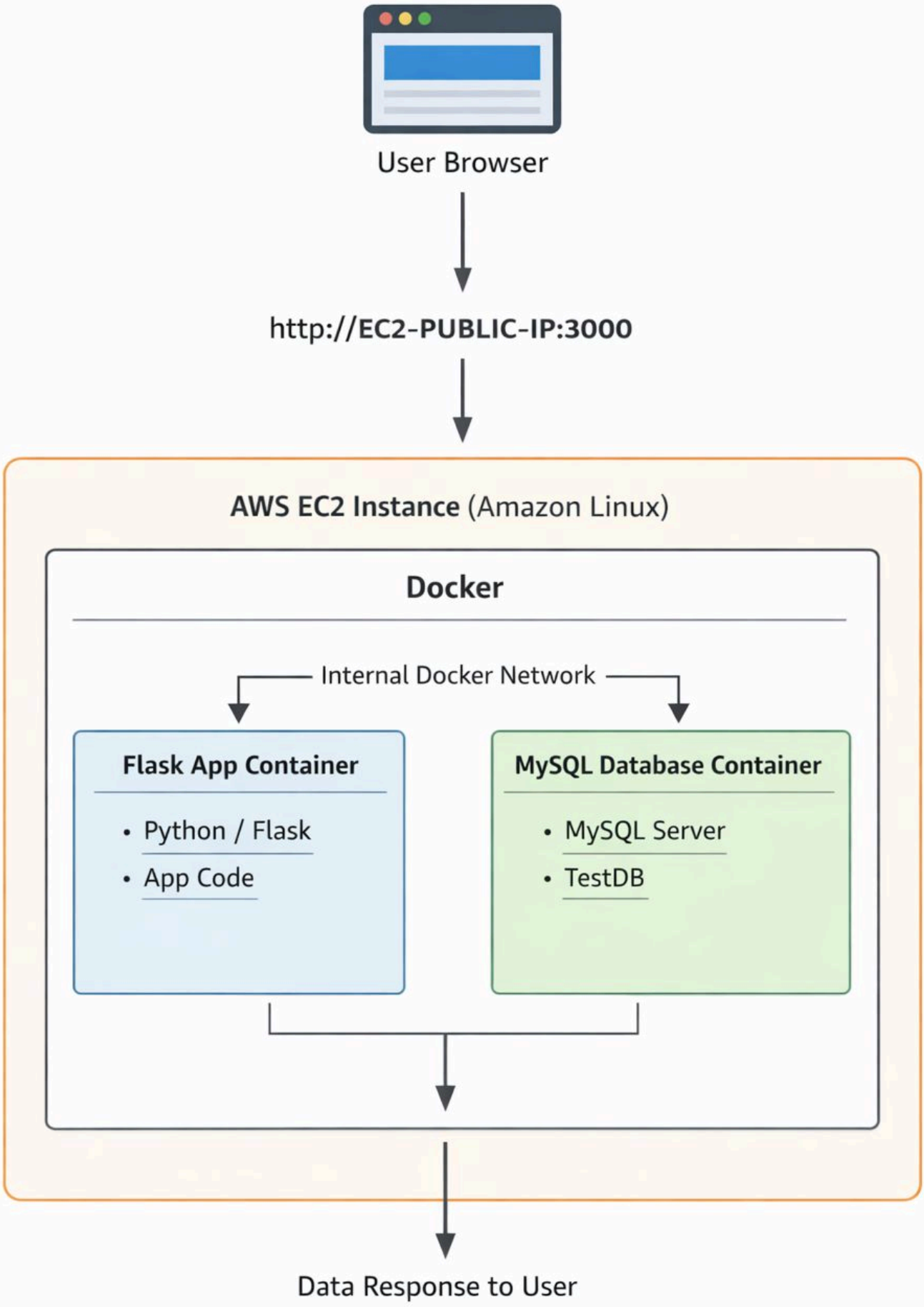
✔ Used correct service name instead of localhost

❌ Application not accessible

✔ Fixed security group rules

Outcome

The assignment was successfully completed by deploying a containerized Flask application with a MySQL database on AWS EC2 using Docker and Docker Compose. The application is accessible via the public IP, demonstrating a complete DevOps workflow.


User Browser
http://EC2-PUBLIC-IP:3000
AWS EC2 Instance (Amazon Linux)
Docker
Internal Docker Network
Flask App Container
• Python / Flask
• App Code
MySQL Database Container
• MySQL Server
• TestDB
Data Response to User