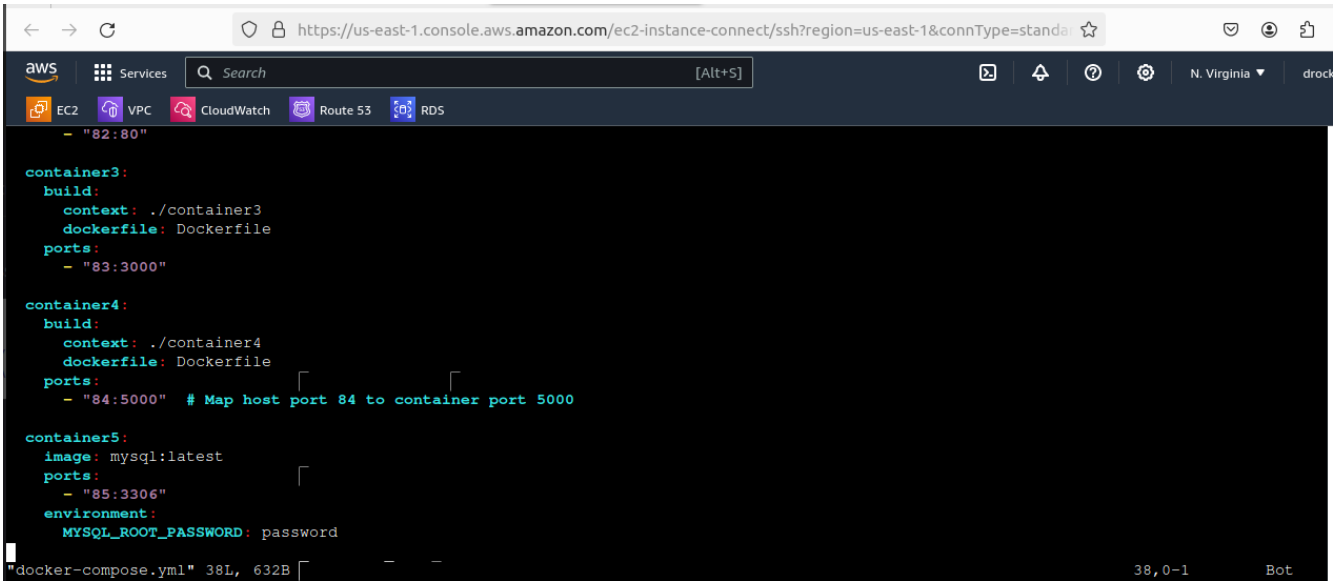
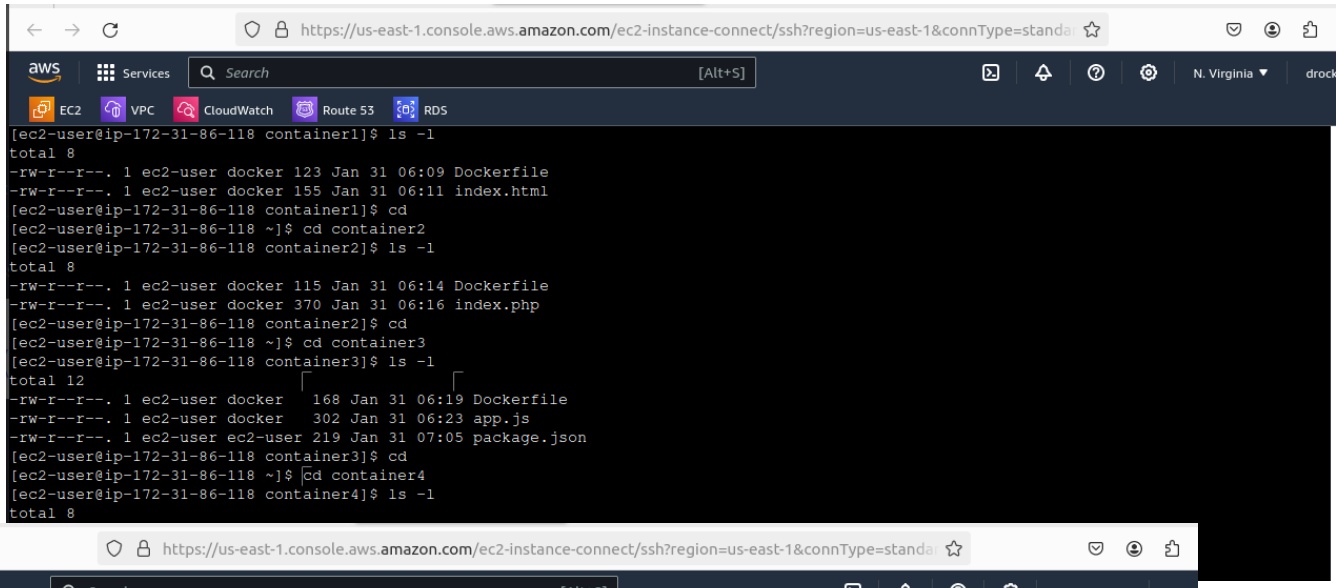
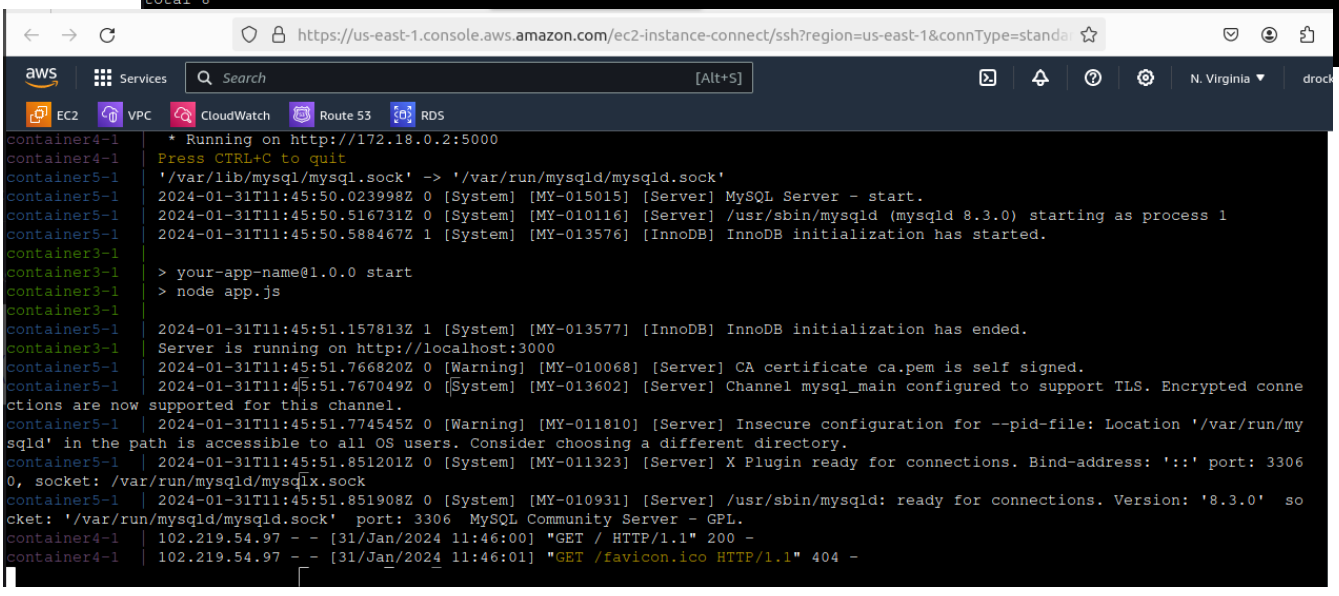
1. Create 5 custom container with 5 different default pages
2. Using Docker Compose deploy these 5 containers on port 81, 82, 83, 84 and 85 respectively

Dockerfile



In the directory container1 container2 container3 container4 container5





mkdir container1 container2 container3 container4 container5

cd container1

vi Dockerfile

```
# Container 1: Nginx serving a static website
FROM nginx:latest
WORKDIR /usr/share/nginx/html
COPY index.html .
EXPOSE 80
```

vi index.html

```
<!-- index.html -->
<!DOCTYPE html>
<html>
<head>
    <title>Welcome to Container 1</title>
</head>
<body>
    <h1>Hello from Nginx!</h1>
</body>
</html>
```

…….

cd container2
vi Dockerfile

```
# Container 2: Apache serving a PHP application
FROM php:apache
WORKDIR /var/www/html
COPY index.php .
EXPOSE 80
```

vi index.php

```
<!-- index.php -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>PHP Application</title>
</head>
```

```
<body>
    <h1>Welcome to Container 2!</h1>
    <p>This is a PHP application served by Apache.</p>
    <?php echo "PHP version: " . phpversion(); ?>
</body>
</html>
```

……..

```
cd container3
vi Dockerfile
```

```
# Container 3: Node.js application using Express framework
FROM node:latest
WORKDIR /app
COPY package.json .
RUN npm install
COPY . .
EXPOSE 3000
CMD ["npm", "start"]
```

```
vi app.js
```

```
// app.js
const express = require('express');
const app = express();

const PORT = 3000;

app.get('/', (req, res) => {
  res.send('Welcome to Container 3! This is a Node.js application using Express.');
});

app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```

```
vi package.json
```

```
{
  "name": "your-app-name",
  "version": "1.0.0",
  "description": "Your application description",
  "main": "app.js",
```

```
  "scripts": {
    "start": "node app.js"
  },
  "dependencies": {
    "express": "^4.17.1"
  }
}
```
……

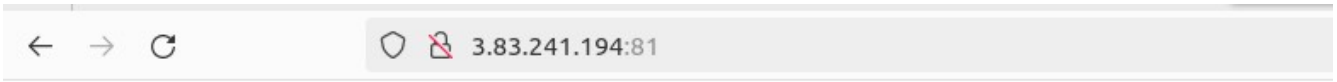
```
cd container4
vi Dockefile

# Container 4: Flask application
FROM python:latest
WORKDIR /app

# Install Flask
RUN pip install Flask

# Copy application files
COPY . .

EXPOSE 5000
CMD ["python", "app.py"]



vi app.py

# app.py
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello():
    return 'Welcome to Container 4! This is a Flask application.'

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

vi docker-compose.yml

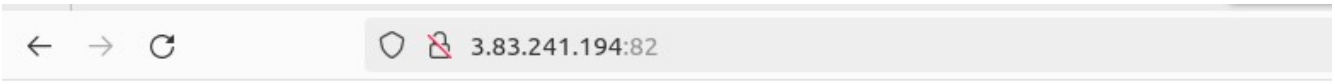
```
version: '3'

services:
  container1:
    build:
      context: ./container1
      dockerfile: Dockerfile
    ports:
      - "81:80"

  container2:
    build:
      context: ./container2
      dockerfile: Dockerfile
    ports:
      - "82:80"

  container3:
    build:
      context: ./container3
      dockerfile: Dockerfile
    ports:
      - "83:3000"

  container4:
    build:
      context: ./container4
      dockerfile: Dockerfile
    ports:
      - "84:5000"  # Map host port 84 to container port 5000

  container5:
    image: mysql:latest
    ports:
      - "85:3306"
    environment:
      MYSQL_ROOT_PASSWORD: password
```

……
docker-compose build

docker-compose up

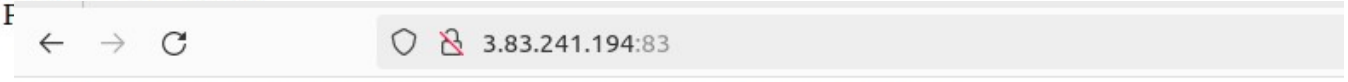


Hello from Nginx!



Welcome to Container 2!

This is a PHP application served by Apache.



Welcome to Container 3! This is a Node.js application using Express.