

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

```
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
```

In the directory container1 container2 container3 container4 container5

```
[ec2-user@ip-172-31-86-118 container1]$ ls -l
total 8
-rw-r--r--. 1 ec2-user docker 123 Jan 31 06:09 Dockerfile
-rw-r--r--. 1 ec2-user docker 155 Jan 31 06:11 index.html
[ec2-user@ip-172-31-86-118 container1]$ cd
[ec2-user@ip-172-31-86-118 ~]$ cd container2
[ec2-user@ip-172-31-86-118 container2]$ ls -l
total 8
-rw-r--r--. 1 ec2-user docker 115 Jan 31 06:14 Dockerfile
-rw-r--r--. 1 ec2-user docker 370 Jan 31 06:16 index.php
[ec2-user@ip-172-31-86-118 container2]$ cd
[ec2-user@ip-172-31-86-118 ~]$ cd container3
[ec2-user@ip-172-31-86-118 container3]$ ls -l
total 12
-rw-r--r--. 1 ec2-user docker 168 Jan 31 06:19 Dockerfile
-rw-r--r--. 1 ec2-user docker 302 Jan 31 06:23 app.js
-rw-r--r--. 1 ec2-user ec2-user 219 Jan 31 07:05 package.json
[ec2-user@ip-172-31-86-118 container3]$ cd
[ec2-user@ip-172-31-86-118 ~]$ cd container4
[ec2-user@ip-172-31-86-118 container4]$ ls -l
total 8
```

```
* Running on http://172.18.0.2:5000
Press CTRL+C to quit
'/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.sock'
2024-01-31T11:45:50.023998Z 0 [System] [MY-015015] [Server] MySQL Server - start.
2024-01-31T11:45:50.516731Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.3.0) starting as process 1
2024-01-31T11:45:50.588467Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
> your-app-name@1.0.0 start
> node app.js
2024-01-31T11:45:51.157813Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
Server is running on http://localhost:3000
2024-01-31T11:45:51.766820Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2024-01-31T11:45:51.767049Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.
2024-01-31T11:45:51.774545Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.
2024-01-31T11:45:51.851201Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 3306
0, socket: /var/run/mysqld/mysqld.sock
2024-01-31T11:45:51.851908Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.3.0' socket: '/var/run/mysqld/mysqld.sock' port: 3306 MySQL Community Server - GPL.
102.219.54.97 - - [31/Jan/2024 11:46:00] "GET / HTTP/1.1" 200 -
102.219.54.97 - - [31/Jan/2024 11:46:01] "GET /favicon.ico HTTP/1.1" 404 -
```

```
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:
```

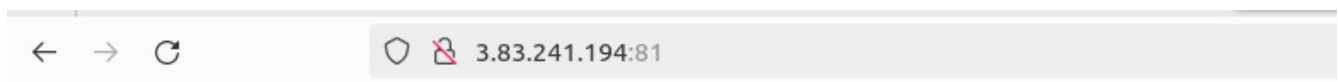
```
      - "85:3306"
```

```
    environment:
```

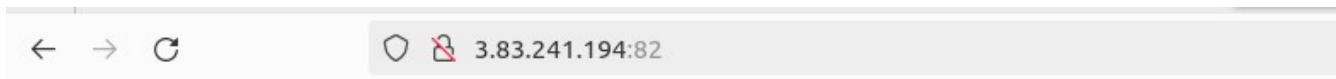
```
      MYSQL_ROOT_PASSWORD: password
```

```
.....  
docker-compose build
```

```
docker-compose up
```

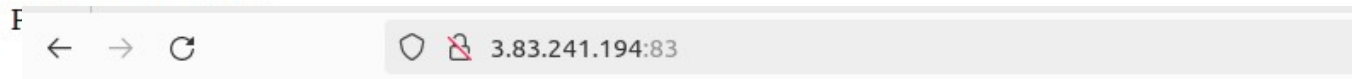


**Hello from Nginx!**



**Welcome to Container 2!**

This is a PHP application served by Apache.



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



Welcome t