Lab Exercise 2- Docker-Compose file

Objective:

Set up a WordPress environment using Docker Compose, including a MySQL database as the backend.

Prerequisites:

- Docker and Docker Compose installed on your system.

Step 1: Create a *docker-compose.yml* File

1. In the project directory, create a file named docker-compose.yml.

2. Add the following content to docker-compose.yml:

```
version: '3.8'

services:
 wordpress:
  image: wordpress:latest
  ports:
   - "8080:80"
  environment:
   WORDPRESS_DB_HOST: db:3306
   WORDPRESS_DB_USER: wp_user
   WORDPRESS_DB_PASSWORD: wp_pass
   WORDPRESS_DB_NAME: wp_database
  depends_on:
   - db

 db:
  image: mysql:5.7
  environment:
   MYSQL_ROOT_PASSWORD: root_password
   MYSQL_DATABASE: wp_database
   MYSQL_USER: wp_user
   MYSQL_PASSWORD: wp_pass
```

```
    volumes:
      - db_data:/var/lib/mysql

volumes:
  db_data:
```

```
 ~ ........................................
> cat << EOF > docker-compose.yml
version: '3.8'

services:
  wordpress:
    image: wordpress:latest
    ports:
      - "8080:80"
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wp_user
      WORDPRESS_DB_PASSWORD: wp_pass
      WORDPRESS_DB_NAME: wp_database
    depends_on:
      - db

  db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: root_password
      MYSQL_DATABASE: wp_database
      MYSQL_USER: wp_user
      MYSQL_PASSWORD: wp_pass
    volumes:
      - db_data:/var/lib/mysql

volumes:
  db_data:
EOF
```

Step 2: Start the Containers

1. Run the following command to start the containers:

```
docker-compose up -d
```

```
 ~ ........................................
> docker-compose up -d

[+] Running 35/35
 ✔ wordpress Pulled
   ✔ 6d29a096dd42 Already exists
   ✔ 29e9756e7bfc Pull complete
   ✔ 77e27785ead7 Pull complete
   ✔ 27a00cfa6daf Pull complete
   ✔ ca35e1d28ad7 Pull complete
   ✔ 410639cb967e Pull complete
   ✔ 1f9efafdf219 Pull complete
   ✔ a7946ba9a569 Pull complete
   ✔ 67f1de0deede Pull complete
   ✔ d96ecd01e3f5 Pull complete
   ✔ c470e029cbda Pull complete
   ✔ 77cbd2f3c900 Pull complete
   ✔ 646aa471737e Pull complete
   ✔ 4f4fb700ef54 Pull complete
   ✔ 3ab30a6e19b4 Pull complete
   ✔ e1635648fee7 Pull complete
   ✔ 721605ec8cbc Pull complete
   ✔ 6bf2a3e9ff28 Pull complete
   ✔ 3dce276ad38f Pull complete
   ✔ fce7fd52b7bb Pull complete
   ✔ 3d27decb1554 Pull complete
   ✔ 7a759b92dfbc Pull complete
 ✔ db Pulled
   ✔ d0164e3ac7ac Pull complete
   ✔ 7bb8c6c53fd8 Pull complete
   ✔ 9c31f49f8fe3 Pull complete
   ✔ 4af3af4e8f76 Pull complete
   ✔ e65be80e519a Pull complete
   ✔ 4ccb4fbac127 Pull complete
   ✔ 23ac9411b2f7 Pull complete
   ✔ 7f5b19b98e13 Pull complete
   ✔ 9b744ce874ce Pull complete
   ✔ ce5f8bfef834 Pull complete
   ✔ 6dd3a178e40b Pull complete
[+] Running 4/4
 ✔ Network ansh_default       Created
 ✔ Volume "ansh_db_data"      Created
 ✔ Container ansh-db-1        Started
 ✔ Container ansh-wordpress-1 Started
```

2. Docker Compose will download the necessary images (WordPress and MySQL) and start both services.

Step 4: Access WordPress

1. Open your web browser and go to http://localhost:8080.
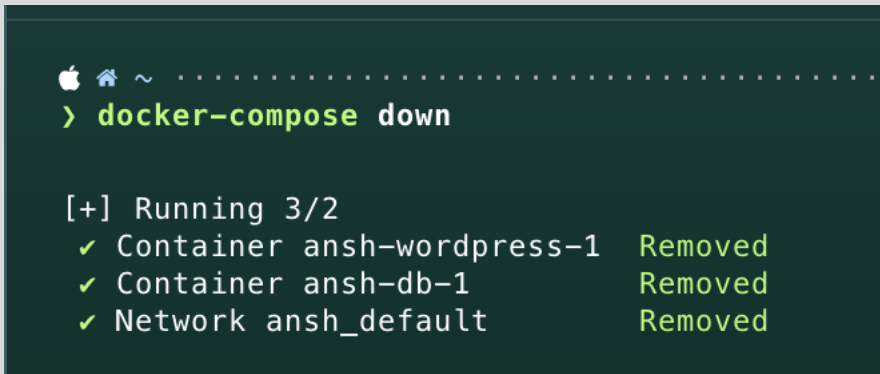
2. Follow the WordPress installation steps to set up your site.

Step 5: Stop and Remove Containers

To stop the containers and remove the associated resources, run:

docker-compose down

```
 ~ ..........................................
> docker-compose down


[+] Running 3/2
 ✔ Container ansh-wordpress-1   Removed
 ✔ Container ansh-db-1          Removed
 ✔ Network ansh_default         Removed
```

Explanation of docker-compose.yml:

- **wordpress**: Sets up the WordPress container, mapping port 80 inside the container to port 8080 on your local machine.

- **db**: Sets up the MySQL container with a volume (db_data) for persistent storage.

Additional Notes:

- Modify the environment variables as needed for different configurations.

- To view logs, use docker-compose logs -f.

This setup allows you to quickly start a WordPress site locally and experiment with configurations.