SOFTWARE ENGINEERING LAB

EXERCISE – 7

TOPIC – 4

RUNNING MULTIPLE CONTAINERS USING DOCKER COMPOSE

By following the steps in the Docker Compose exercise, you will learn how to:

- **Set up a WordPress and MySQL multi-container environment** using a simple configuration file (docker-compose.yml).
- **Create and configure Docker services** for a web application and a database, including setting environment variables and persistent data storage.
- **Run multiple containers together** with a single command, simplifying deployment.
- **Access the WordPress application locally** through a web browser and link it to the MySQL database for data storage.
- **Use Docker Compose commands** to start, stop, and remove containers easily.
- **Persist and share data** between containers using Docker volumes.
- **Understand basic networking in Docker Compose** to enable communication between containers.
- **Reuse the setup on any system** by copying the docker-compose.yml file and running it.

- **Note: At every step take screenshots and save in a document**

1. What is a docker-compose.yml File?

A docker-compose.yml file is a configuration file that defines:

What containers (services) to create.

How they should work together (like connecting WordPress to MySQL).

Settings for each container (such as ports, environment variables, etc.).

This file is written in YAML format, which is a simple, human-readable way to structure data.

2. Where to Find Sample Configurations

You can find examples of docker-compose.yml files:

Docker Documentation: The official Docker website has guides and sample configurations.

GitHub: Many developers share sample files. For example, you can visit Brad Traversy's Docker Example and scroll to version 3 for a beginner-friendly example.

- https://gist.github.com/bradtraversy/faa8de544c62eef3f31de406982f1d42

Docker Hub: Each image (like MySQL or WordPress) often includes example configurations.

For simplicity, let's build a basic docker-compose.yml file for WordPress and MySQL from scratch.

3. Writing a Basic docker-compose.yml File

Step 1: Create a Folder

Go to your desktop or any folder you prefer.

Right-click and select New Folder.

Name the folder my_docker_project.

Step 2: Open a Text Editor

Open Visual Studio Code, Notepad, or any other text editor.

Create a new file.

Step 3: Write the YAML Configuration

Here's a simple example of a docker-compose.yml file for WordPress and MySQL:

```
version: '3.1'

services:

  db:
```

```
    image: mysql:5.7
    container_name: mysql_container
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: wordpress_db
      MYSQL_USER: wordpress_user
      MYSQL_PASSWORD: wordpress_pass
    volumes:
      - db_data:/var/lib/mysql

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    container_name: wordpress_container
    ports:
      - "8000:80"
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress_user
      WORDPRESS_DB_PASSWORD: wordpress_pass
      WORDPRESS_DB_NAME: wordpress_db
    volumes:
      - ./wordpress_data:/var/www/html

volumes:
```

```
  db_data:
```

4. Step-by-Step Explanation

1. File Format

version: '3.1': This specifies the Docker Compose file version. Version 3 is widely supported.

2. Services

Under services, we define the containers we want to run:

- db (MySQL Container):

image: Specifies the MySQL version (here, 5.7).

container_name: Gives the container a name (mysql_container).

environment: Sets environment variables for the database:

MYSQL_ROOT_PASSWORD: Root user password.

MYSQL_DATABASE: Name of the database WordPress will use.

MYSQL_USER and MYSQL_PASSWORD: Credentials WordPress will use to connect.

volumes: Persists database data to the db_data volume.

- wordpress (WordPress Container):

depends_on: Ensures the MySQL container starts first.

image: Specifies the WordPress version (here, latest).

container_name: Gives the container a name (wordpress_container).

ports: Maps port 8000 on your computer to port 80 in the container. You'll access WordPress at http://localhost:8000.

environment: Provides settings to connect WordPress to the MySQL database.

volumes: Connects your local directory (./wordpress_data) to the container for file storage.

3. Volumes

volumes help save data even if the container stops. For example:

db_data: Stores MySQL database data.

5. Saving the File

Save the file as docker-compose.yml.

Place it in the my_docker_project folder.

6. Running the Setup

Step 1: Open Command Line

Open PowerShell or Command Prompt.

Navigate to the my_docker_project folder:

cd path_to_my_docker_project

Step 2: Start the Containers

Run:

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

This command reads the docker-compose.yml file and creates both the WordPress and MySQL containers.

-d runs the containers in the background.

7. Accessing the Application

Open your web browser.

Go to `http://localhost:8000`.

Follow the WordPress setup wizard to complete the installation:

Site Name.

Admin Username and Password.

Email Address.

8. Managing Containers

Stop the Containers

To stop the containers without removing them:

```
docker-compose stop
```

Start Again

To restart the containers:

```
docker-compose start
```

Remove Containers

To stop and remove everything:

```
docker-compose down
```