

## **Placement Empowerment Program**

### ***Cloud Computing and DevOps Centre***

Set Up Docker Compose for Multi-Container Applications Create a docker-compose.yml file to define and run a multi-container application (e.g., a web server and database).

Name: Gopinath M

Department: ADS

## Objective

The goal of this Proof of Concept (PoC) is to demonstrate how to create and use a docker-compose.yml file to define and run a multi-container application. This example will set up an Nginx web server and a MySQL database using Docker Compose.

## What is Docker Compose?

Docker Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure application services, networks, and volumes, then start everything with a single command.

## Prerequisites

Before proceeding, ensure the following prerequisites are met:

- **Docker Installed:** [Get Docker](#)
  - **Docker Compose Installed:** [Install Docker Compose](#)
  - Basic knowledge of **Docker**, **Containers**, and **YAML Syntax**
- 

## Step-by-Step Guide

### Step 1: Verify Docker and Docker Compose Installation

Check if Docker is installed:

```
docker --version
```

Check if Docker Compose is installed:

```
docker-compose --version
```

```
C:\Users\gopin>docker-compose --version  
Docker Compose version v2.32.4-desktop.1
```

If any of these are missing, install them using the links in the prerequisites.

---

### Step 2: Create a Project Directory

Open a terminal and create a directory for your project:

```
mkdir multi-container-app && cd multi-container-app
```

```
C:\Users\gopin>mkdir multi-container-app && cd multi-container-app
C:\Users\gopin\multi-container-app>
```

Inside this directory, create a docker-compose.yml file:

**touch docker-compose.yml**

---

### Step 3: Define Services in docker-compose.yml

Open the file in a text editor and add the following content:

```
version: '3.8'

services:
  web:
    image: nginx
    ports:
      - "8080:80"
    depends_on:
      - db

  db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: mydatabase
      MYSQL_USER: user
      MYSQL_PASSWORD: password
```

#### Explanation of the YAML Configuration:

- version: '3.8' → Specifies the Docker Compose version.
- services: → Defines the containers in this application.
- web: → Nginx service:
  - Uses the official Nginx image.
  - Maps container port 80 to local port 8080.
  - Depends on the database service (db).
- db: → MySQL service:

- Uses MySQL version 5.7.
- Sets up environment variables for root password, database, user, and user password.

---

## Step 4: Start the Application

Run the following command to start the services:

**docker-compose up -d**

```
C:\Users\gopin\multi-container-app>docker-compose up -d
time="2025-02-23T19:34:16+05:30" level=warning msg="C:\Users\gopin\multi-container-app\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
db [pull] 46.88MB / 137.9MB Pulling 18.6s
db95a948e7b6 Download complete 0.4s
1c56c3d4ce74 Download complete 0.4s
20e4dca4c69 Downloading [=====] 28.97MB/59.5MB 2.0s
43d85e938198 Downloading [=====] 11.53MB/56.29MB 13.6s
df9a4d85569b Download complete 13.6s
db Pulled 2.2s
db95a948e7b6 Download complete 51.1s
1c56c3d4ce74 Download complete 2.0s
20e4dca4c69 Download complete 25.0s
43d85e938198 Download complete 37.4s
df9a4d85569b Download complete 2.2s
e9f83a1c24c Download complete 3.6s
964b2d298fba Download complete 1.9s
ffc89e9fd88 Download complete 2.2s
ae71319d9779 Download complete 31.4s
68c1898c2015 Download complete 8.4s
90986bb8de6e Download complete 2.0s
web Pulled 3.1s
[+] Running 3/3
Network multi-container-app_default Created 0.4s
Container multi-container-app-db-1 Started 4.7s
Container multi-container-app-web-1 Started 1.9s
C:\Users\gopin\multi-container-app>
```

This command will:

- Pull the required Docker images (if not available locally).
- Create and start the containers in the background (detached mode -d).

---

## Step 5: Verify the Running Containers

To check running containers:

**docker ps**

To view the logs for the services:

**docker-compose logs**

```
C:\Users\gopin\multi-container-app>docker-compose logs
time="2025-02-23T19:35:58+05:30" level=warning msg="C:\Users\gopin\multi-container-app\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
db-1 | 2025-02-23 14:05:12+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.44-1.el7 started.
db-1 | 2025-02-23 14:05:12+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
db-1 | 2025-02-23 14:05:12+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.44-1.el7 started.
db-1 | 2025-02-23 14:05:12+00:00 [Note] [Entrypoint]: Initializing database files
db-1 | 2025-02-23T14:05:12.764767Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
db-1 | 2025-02-23T14:05:13.391446Z 0 [Warning] InnoDB: New log files created, LSN=457990
db-1 | 2025-02-23T14:05:13.486654Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
db-1 | 2025-02-23T14:05:13.512734Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: 337dae4e-f1ef-11ef-83b2-0242ac120902.
db-1 | 2025-02-23T14:05:13.518866Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
web-1 | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
web-1 | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
web-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
web-1 | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
web-1 | 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
web-1 | /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
web-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
web-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
web-1 | /docker-entrypoint.sh: Configuration complete; ready for start up
web-1 | 2025/02/23 14:05:12 [notice] 181: using the "epoll" event method
web-1 | 2025/02/23 14:05:12 [notice] 181: nginx/1.27.4
web-1 | 2025/02/23 14:05:12 [notice] 181: built by gcc 12.2.0 (Debian 12.2.0-14)
web-1 | 2025/02/23 14:05:12 [notice] 181: OS: Linux 5.15.167.0-microsoft-standard-WSL2
web-1 | 2025/02/23 14:05:12 [notice] 181: getloadavg(0.01): 1948576.1948576
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker processes
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker process 29
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker process 30
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker process 31
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker process 32
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker process 33
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker process 34
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker process 35
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker process 36
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker process 37
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker process 38
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker process 39
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker process 40
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker process 41
web-1 | 2025/02/23 14:05:12 [notice] 181: start worker process 42
web-1 | 2025-02-23T14:05:12.764767Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
web-1 | 2025-02-23T14:05:12.764767Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: 337dae4e-f1ef-11ef-83b2-0242ac120902.
```

To check a specific service log (e.g., MySQL):

**docker-compose logs db**

## Step 6: Stop and Remove Containers

To stop the application:

**docker-compose down**

```
C:\Users\gopin\multi-container-app>docker-compose down
time="2025-02-23T19:36:21+05:30" level=warning msg="C:\Users\gopin\multi-container-app\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 3/3
✔ Container multi-container-app-web-1 Removed 1.1s
✔ Container multi-container-app-db-1 Removed 2.2s
✔ Network multi-container-app_default Removed 0.9s
```

This command will:

- Stop all running containers.
- Remove the containers and networks created by Docker Compose.

## Conclusion

By following these steps, you have successfully set up a multi-container application using Docker Compose. This setup can be extended to include more services like Redis, Node.js, or PostgreSQL.

## References

- [Docker Compose Documentation](#)
- [Docker Hub Nginx](#)

- [Docker Hub MySQL](#)