

## Docker Networking and Volumes Assessment

### Objective:

You are tasked with setting up a Docker environment where:

1. A **bridge network** is created to enable communication between two containers.
  2. A **database container** is created using Docker volumes for persistent data.
  3. A **web application container** is deployed using **Nginx** as the web server to serve the application.
  4. The two containers need to communicate with each other on the bridge network.
- 

### Instructions:

#### Step 1: Create a Docker Bridge Network

Create a **bridge network** in Docker, which will allow two containers to communicate with each other on a private network.

- Command:
  - Create a custom bridge network for container communication.

#### Step 2: Create a Docker Volume for Persistent Database Storage

Create a **Docker volume** to store database data persistently. This volume will ensure that even if the database container is stopped or removed, the data will be retained.

- Command:
  - Create a named volume for storing database data.

#### Step 3: Deploy a Database Container

Deploy a database container (e.g., MySQL or PostgreSQL) and connect it to the custom bridge network created in Step 1.

- Ensure that the **database data** is persisted using the volume created in Step 2.
- Command:
  - Deploy the database container using the image of your choice (e.g., MySQL) and mount the volume for persistent storage.

#### Step 4: Deploy a Web Application with Nginx

Deploy a web application container that uses **Nginx** to serve a simple HTML page. This web application will need to communicate with the database container.

- Command:
  - Deploy the web application container, ensuring it's connected to the same bridge network.

### Step 5: Establish Communication Between Containers

Ensure that both containers (the database and the web application) can communicate with each other over the custom bridge network. For example, the web application should be able to access the database using the database container's name or IP address.

- Command:
  - Verify that both containers can communicate with each other.

### Step 6: Verify the Setup

After deployment:

1. Ensure both containers are connected to the bridge network.
  2. Confirm the **web application** is accessible through **Nginx**.
  3. Test if the web application can **interact with the database**.
- Command:
    - Use Docker commands to inspect the network and verify the containers' status.
    - Optionally, use `docker exec` to verify the web container can access the database