Postiefs

Containerizing a Node.js Application with MySQL Using Docker and Docker Compose

Overview

This project demonstrates how to use **Docker** and **Docker Compose** to containerize a simple **Node.js** application that connects to a **MySQL** database. The Node.js app queries the database for the current time and displays it through an HTTP endpoint.

Key Components:

- **Node.js Application**: A simple Express.js app that connects to a MySQL database and fetches the current time.
- **MySQL Database**: A MySQL database that stores and provides data to the Node.js application.

Docker

Docker is a platform that allows you to run applications in isolated environments called containers. It helps ensure that your app runs the same way on any machine, making it easier to develop, deploy, and manage.

Docker Compose

Docker Compose is a tool that allows you to define and manage multi-container applications. It lets you specify how different services (e.g., the Node.js app and the MySQL database) should work together in a single file (**docker-compose.yml**).

Workflow

1. **Node.js Application**:
    - The Node.js app runs an **Express.js** server that connects to a **MySQL database** and queries for the current time using the SQL command **SELECT NOW()**.
    - The app is containerized using Docker. The **Dockerfile** builds an image for the Node.js application, installs dependencies, and starts the server.
2. **MySQL Database**:
    - The MySQL database is set up inside a Docker container. It is initialized with a database (**myapp_db**) and a user (**myapp_user**), which the Node.js application uses to access the database.
    - The MySQL container is configured through **Docker Compose**.
3. **Docker Compose**:
    - **Docker Compose** is used to manage both the Node.js and MySQL containers. It defines both services, ensuring they can communicate with each other by using Docker's internal networking.
    - The **docker-compose.yml** file specifies the configuration for both the Node.js app and MySQL database.

Steps to Run the Project:

1. **Clone the Project**:
    - First, clone the project repository to your local machine.

2. **Build and Start the Containers**:
    - Navigate to the project directory and run the following command: **docker-compose up --build**
    - This command builds the containers and starts both the Node.js app and MySQL database.

3. **Access the Application**:
    - Open your browser and go to **http://localhost:3000**. You should see a message with the current time fetched from the MySQL database.

4. **Stop the Containers**:
    - To stop the containers run: **docker-compose down**

Conclusion

This project uses **Docker** and **Docker Compose** to simplify the process of setting up and running a Node.js application with a MySQL database. By containerizing both the app and the database, you can easily run and deploy the application across different environments, ensuring consistency and ease of maintenance.