LET'S SIMPLIFY HEALTHCARE. TOGETHER

www.equipohealth.com

equipo Team Up

DEVOPS ASSESSMENT

EQUIPO HEALTH INC

Dockerize Python API + MySQL + React + Nginx Proxy

Project Overview

You must create a multi-container Docker setup with these components:

- **Python API:**
  - Use Python 3.11 slim
  - Connects to MySQL DB using env vars (settings.py)
  - Runs a simple REST API (e.g., CRUD tasks or simple hello endpoint) (Optional)
- **MySQL Database:**
  - Use official MySQL image
  - Use environment variables for root user, password, and database name
  - Use Docker volume to persist data
- **React App:**
  - Simple React app served by its own container (use `node:18-alpine or LTS` to build, serve with `nginx`)
- **Nginx Proxy:**
  - Reverse proxy on localhost to route requests:
    - `/api` → to Python API container
    - `/` → to React app container

Requirements

1. **Dockerfiles**
    - One for the Python API (using python:3.11-slim)
    - One for React app (build + serve static files)
2. **docker-compose.yml** with 4 services:
    - `api` (Python API)
    - `db` (MySQL)
    - `app` (React app)
    - `proxy` (Nginx reverse proxy)
3. **Environment Variables:**
    - DB host, user, password, database passed to API via env vars
    - MySQL root password, user, password, database set in docker-compose
4. **Networking:**
    - All containers on the same custom Docker network
    - Nginx routes `/api` to `api:port`
    - Nginx routes `/` to `app:port`
5. **Volumes:**
    - Persistent volume for MySQL data
6. **README:**
    - How to build and run
    - How to test endpoints and React app
    - Env variables explanation

Expected Deliverables

- `Dockerfile` for API
- `Dockerfile` for React app
- `docker-compose.yml`
- `nginx.conf` for proxy setup
- `README.md`
- Source code (minimal, e.g., a simple REST API and a React app that fetches `/api` data)

Create a public repository in github with result and share.