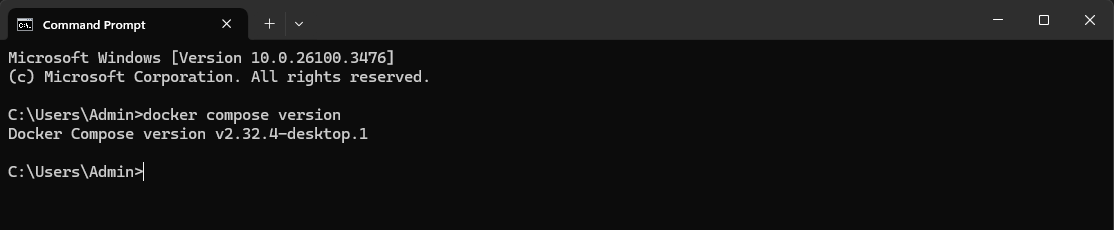
**Nguyễn Đức Nhật\_21059221**

**Phần 1: Một số lệnh Docker Compose cơ bản**

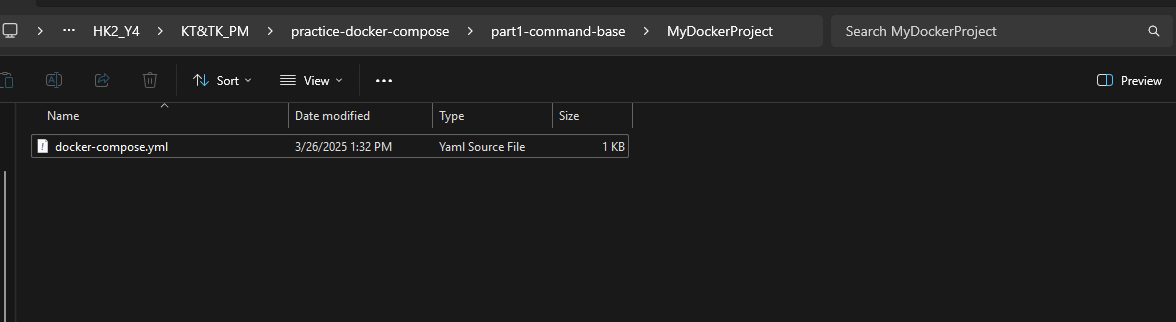
1. **docker compose version**



1. **docker compose up**

**-- setup thư mục**

**B1: tạo file docker-compose.yml**



**B2: Cấu hình file**

- 3.8 là phiên bản được sử dụng, tương thích với Docker Engine mới nhất.

- services là phần khai báo các container sẽ chạy trong Docker Compose.

(web và db).

- ports: - "8080:80"

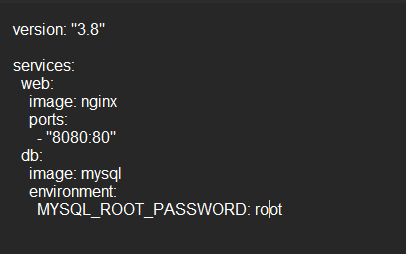
- db: Tên của dịch vụ (container MySQL).

image: mysql: Sử dụng image MySQL từ Docker Hub.

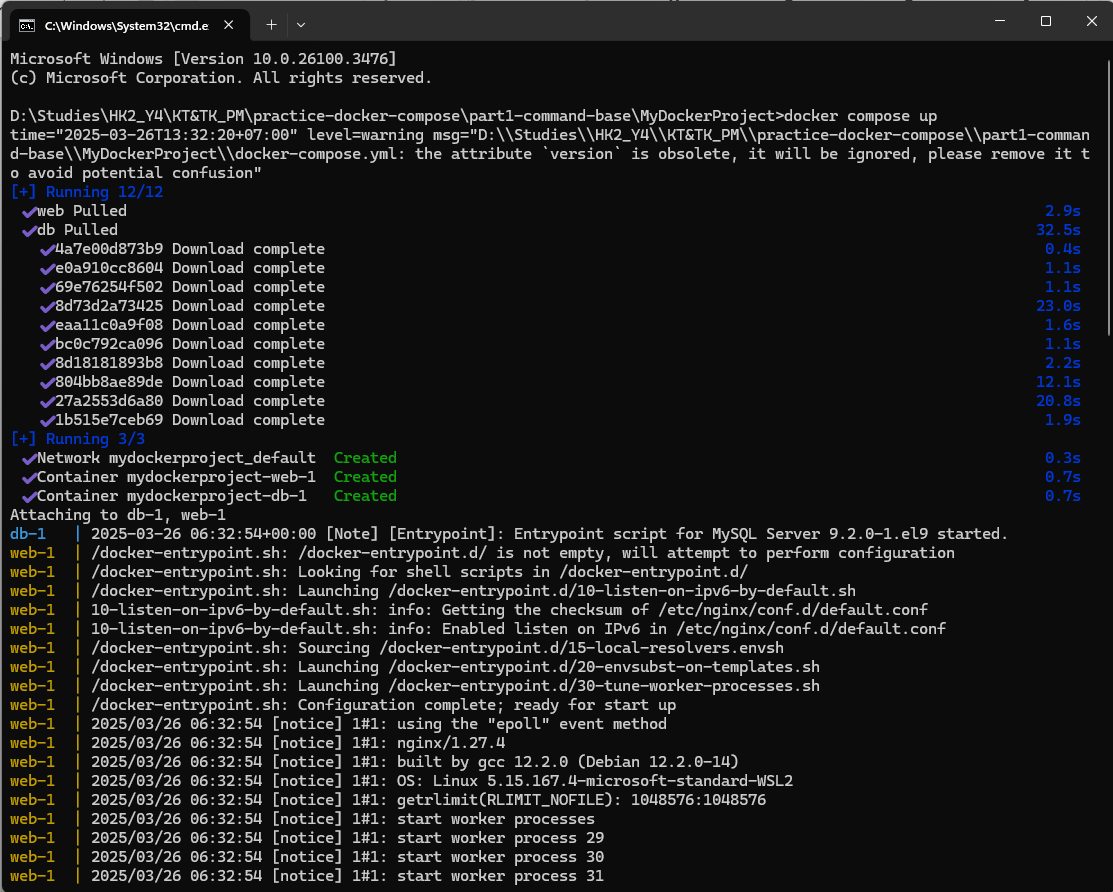
environment::MYSQL\_ROOT\_PASSWORD: root:

Thiết lập mật khẩu cho user root của MySQL.

Giá trị mật khẩu là root.

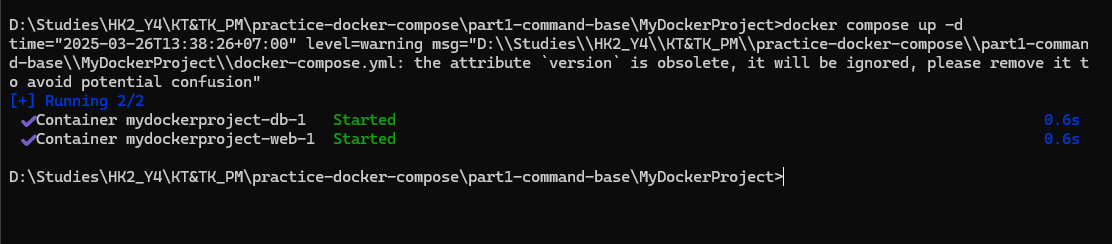


**--** khởi chạy Docker compose bằng lệnh **“docker compose up”**



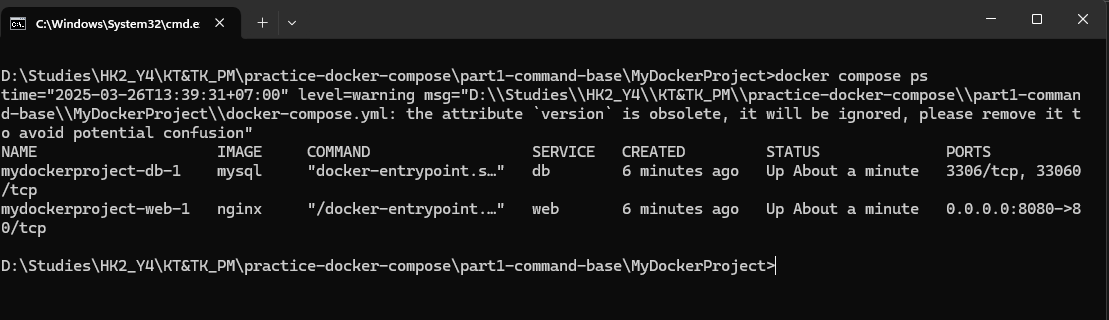
1. **docker compose up -d**

**-** Chạy container nhưng không hiển thị log trong CMD.



1. **docker compose ps**

**-** Xem danh sách container đang chạy.



- có 2 container đang chạy:

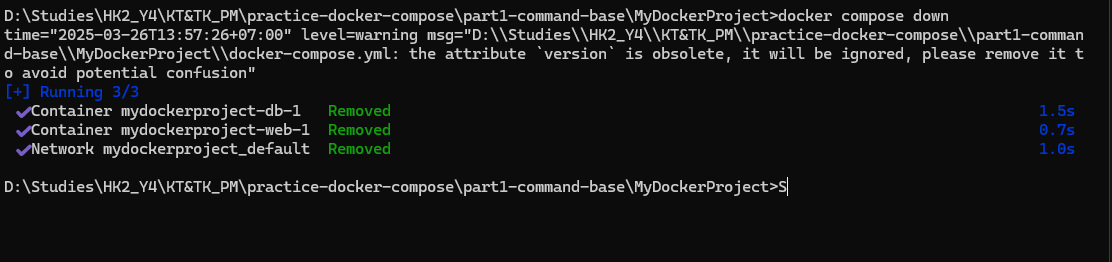
mydockerproject-db-1: Container cho dịch vụ MySQL.

mydockerproject-web-1: Container cho dịch vụ Nginx.

1. **docker compose down**

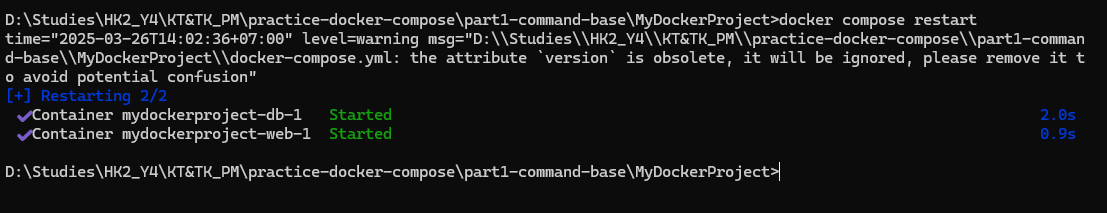
**-** Dùng để **Dừng** và **xóa** container

- Nó chỉ Xóa container nhưng giữ lại volume



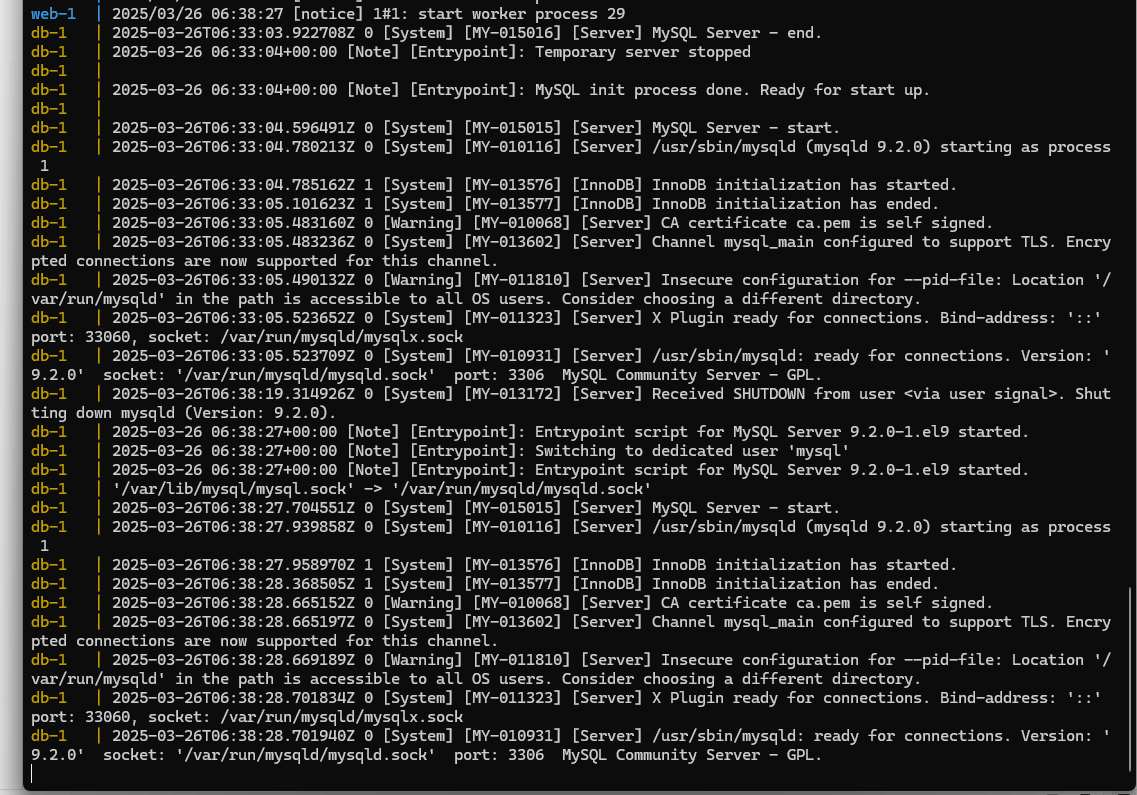
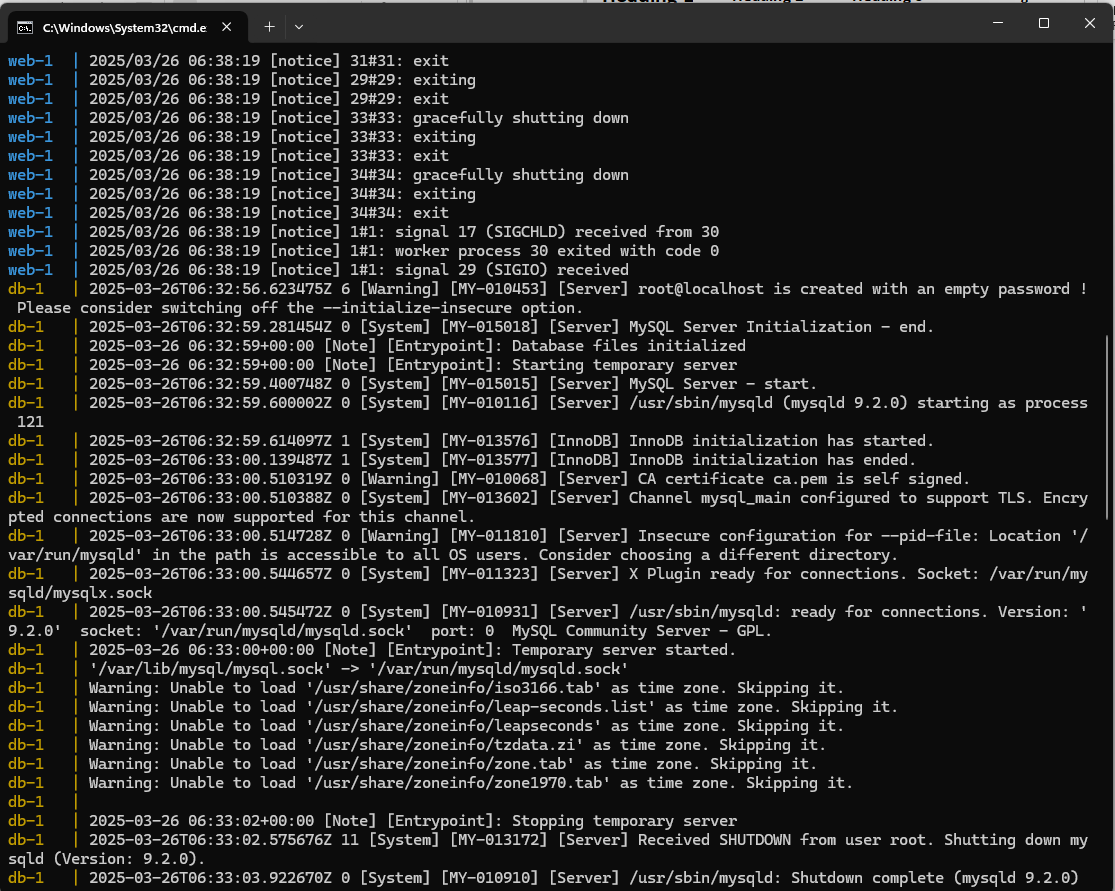
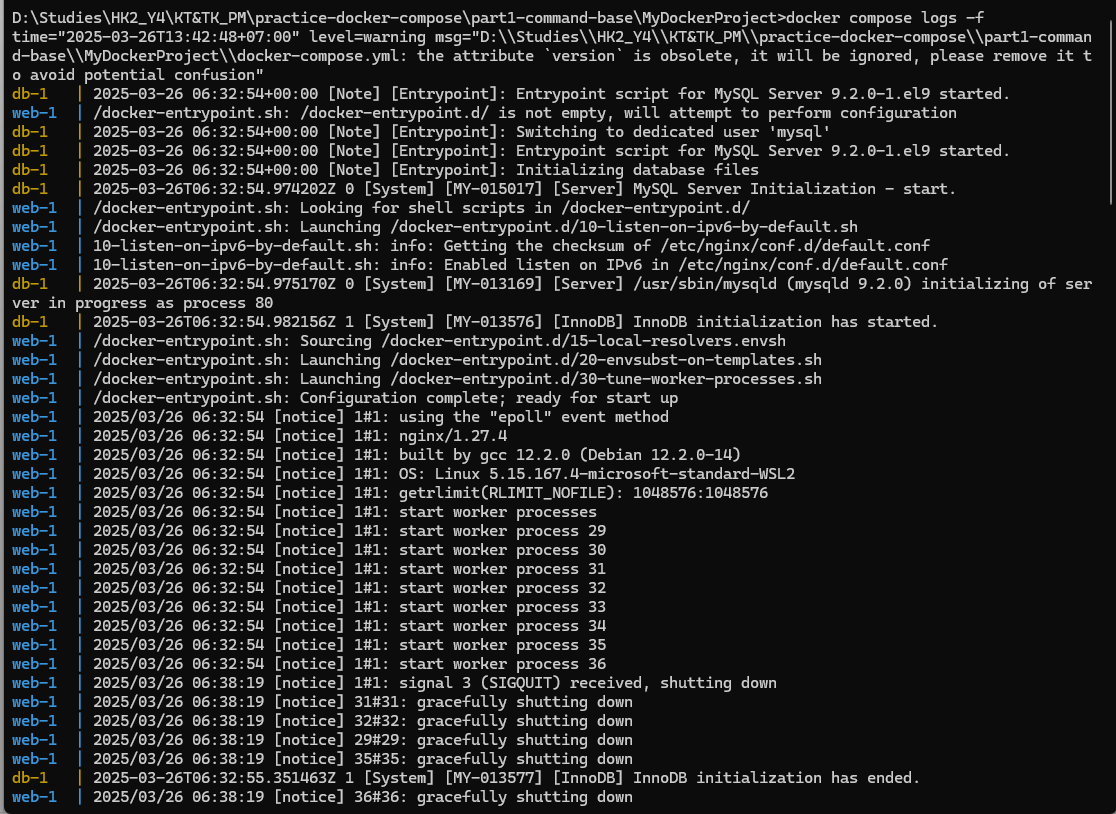
1. **docker compose restart**

- Dùng để khởi động lại container



1. **docker compose logs -f**

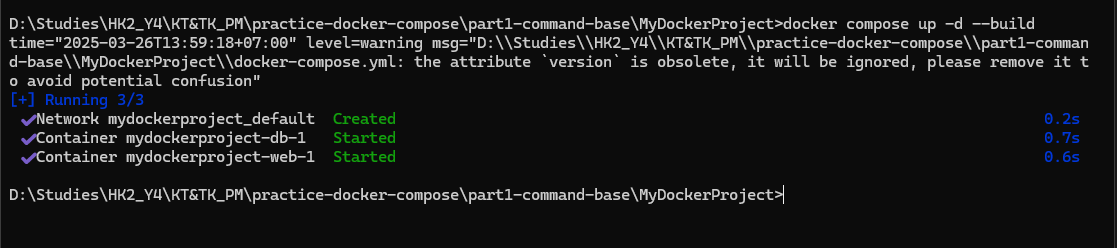
- Xem log theo thời gian thực.



1. **docker compose build**

- **docker compose build**: dùng để Build lại image

- **docker compose up -d --build**: dùng để Build lại image và chạy container



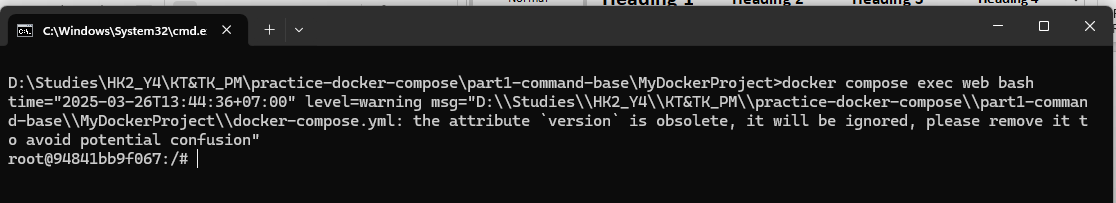
1. **docker compose exec <service\_name> <command>**

**-> docker compose exec web bash**

- docker compose exec: Thực thi lệnh bên trong container đang chạy.

- web: Tên của dịch vụ (container) mà bạn muốn truy cập.

- bash: Mở terminal shell bên trong container.



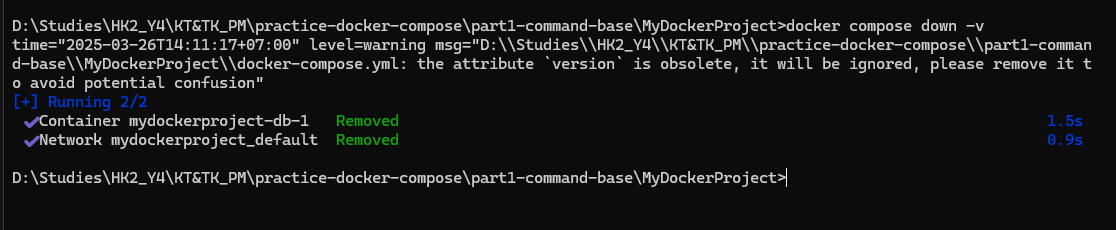
- display xem một vài thông tin



- nhập **‘exit’** để thoát

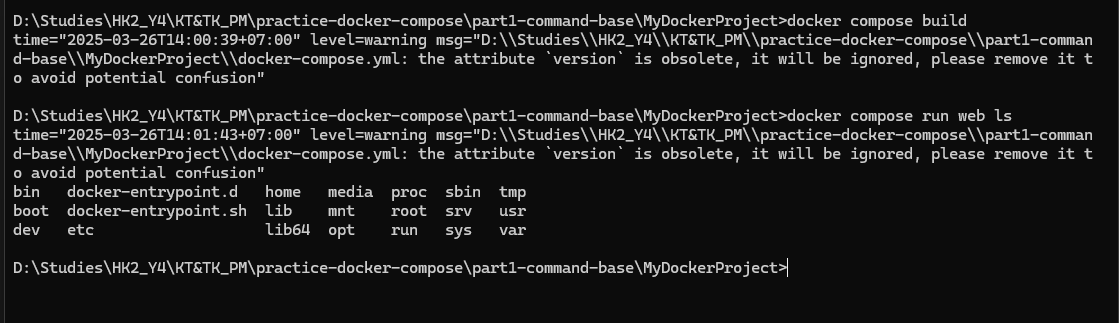
1. **docker compose down -v**

- dừng và xóa tất cả container, và nếu muốn xóa cả volume (dữ liệu MySQL)



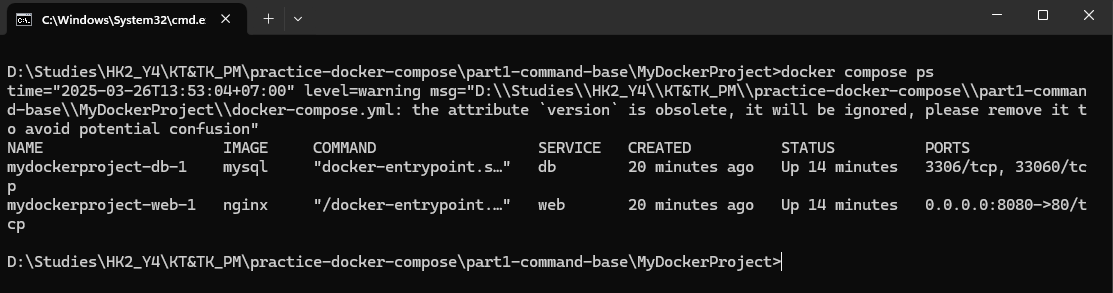
1. **docker compose run <service\_name> <command>**

**- docker compose run web ls:** Chạy lệnh ls trong container web.

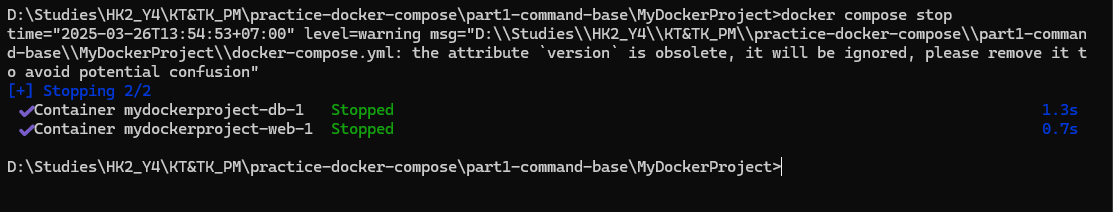


1. **docker compose stop <service\_name>**

**- docker compose ps** để xem danh sách

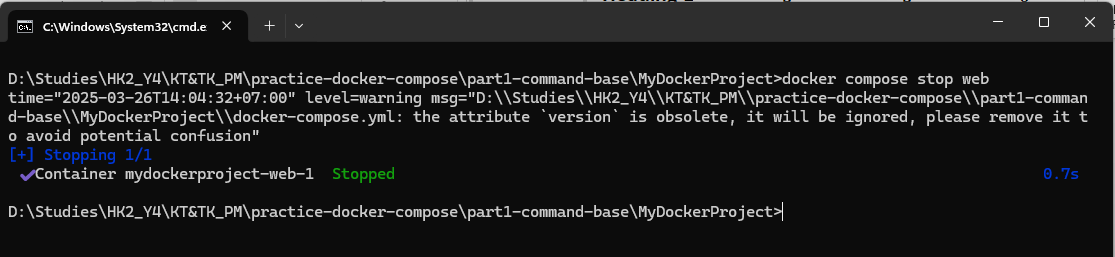


**- docker compose stop:** Dừng container nhưng không xóa

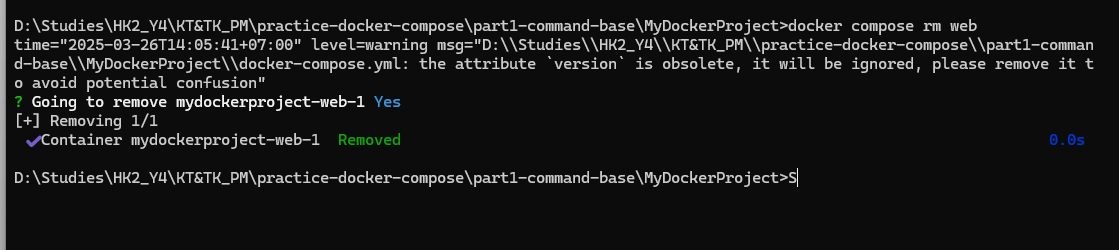


1. **docker compose rm <service\_name>**

- b1: dừng container bằng: docker compose stop web



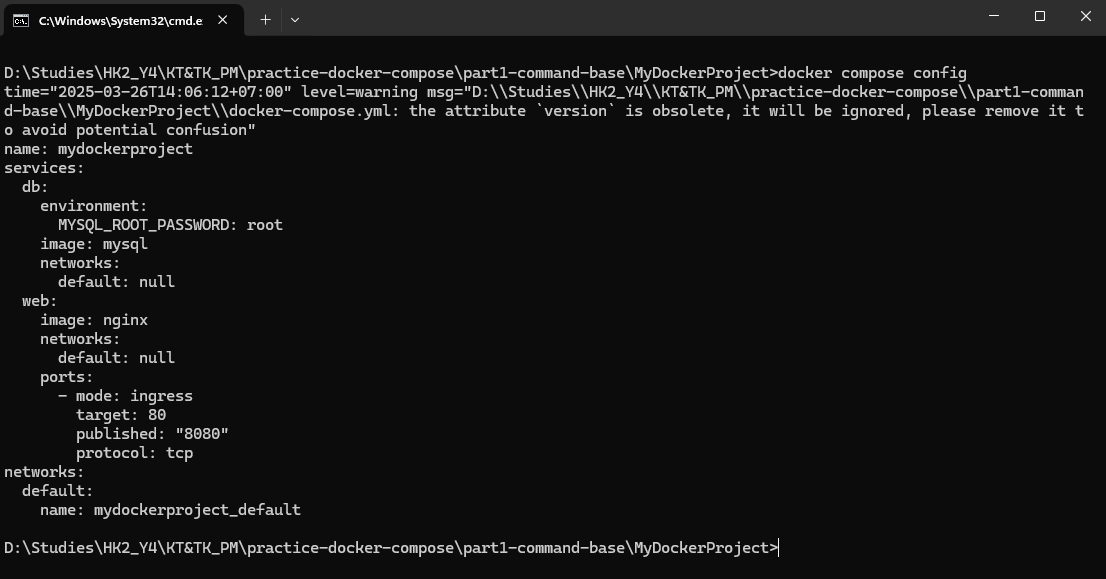
- b2: run lệnh **“docker compose rm web”**: Xóa container nhưng không xóa volume



1. **docker compose config**

- Kiểm tra và hiển thị file docker-compose.yml sau khi Docker Compose phân tích.

- Giúp phát hiện lỗi cấu hình nếu có.



**-> Không có lỗi, chỉ có cảnh báo** version **đã lỗi thời:"the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"**

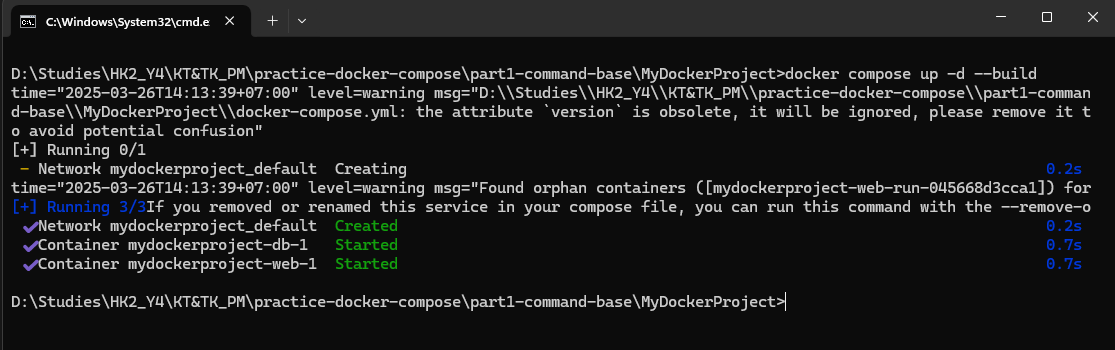
**->** Cách khắc phục: Mở file docker-compose.yml và xóa dòng ‘**version: "3.8"’**

1. **docker compose up -d --build**

**docker compose up:**Khởi động tất cả container theo docker-compose.yml

**-d:** Chạy container ở chế độ nền (detached mode)

**--build:** Build lại image trước khi chạy (nếu có thay đổi)



**Phần 2: Docker Composefile**

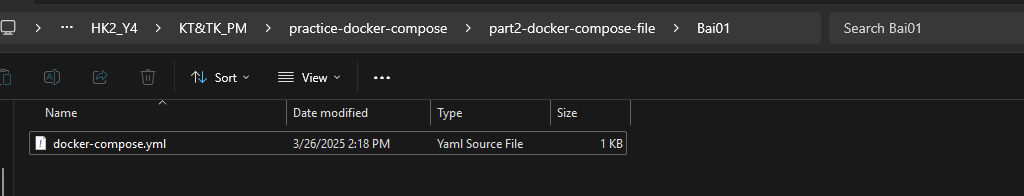
**Bài 1:** Chạy một container đơn giản với Docker Compose

**Yêu cầu:**

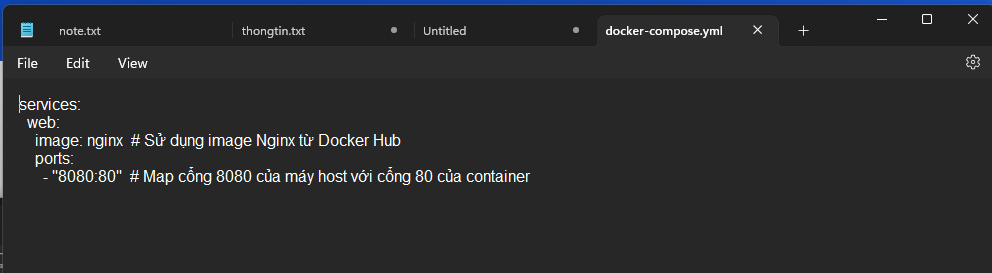
Tạo một container chạy Nginx bằng Docker Compose.

Map cổng 8080 của máy host với cổng 80 của container.

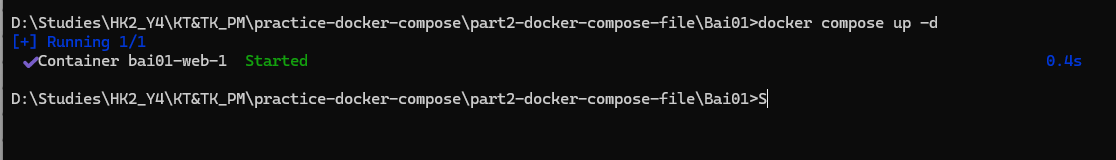
-- setup



-- nội dung file docker-compose.yml



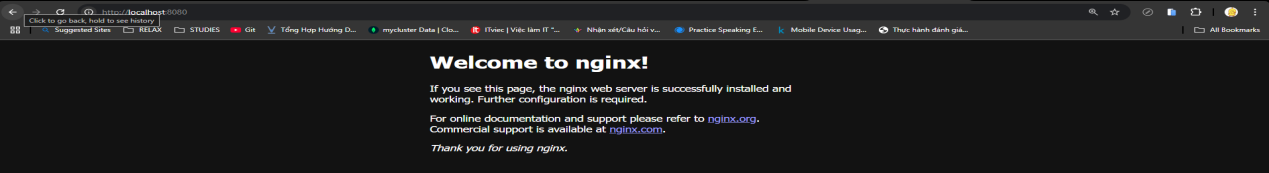
-- Chạy container với Docker Compose: **docker compose up -d**



-- Kiểm tra container có chạy không ? **docker compose ps**



-- Truy cập: **<http://localhost:8080>** để kiểm tra



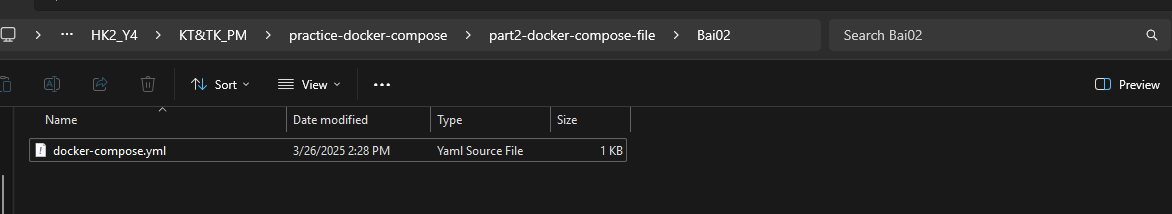
**Bài 2:** Chạy MySQL với Docker Compose

**Yêu cầu:**

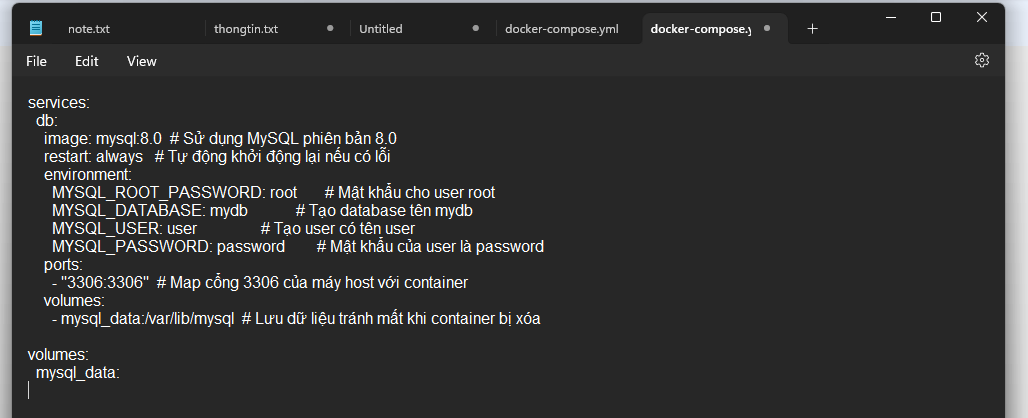
Tạo một container chạy MySQL phiên bản 8.0.

Đặt username là user, password là password và database là mydb

**-- setup**



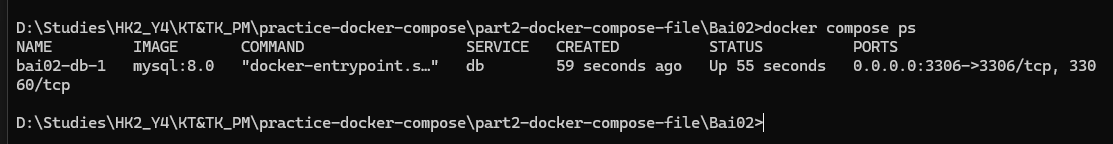
-- nội dung file docker-compose.yml



-- Chạy container với Docker Compose: **docker compose up -d**



-- Kiểm tra container có chạy không ? **docker compose ps**

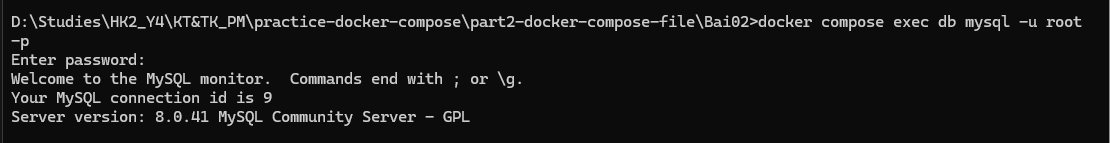


--Kiểm tra kết quả:

**B1: dùng docker CLI:** docker compose exec db mysql -u user -p

**B2:** Nhập mật khẩu: root

**B3:** Kết quả:



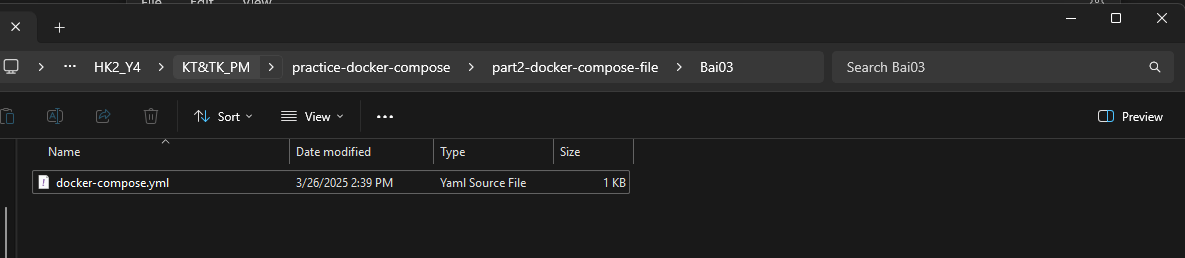
**Bài 3:** Kết nối MySQL với PHPMyAdmin

**Yêu cầu:**

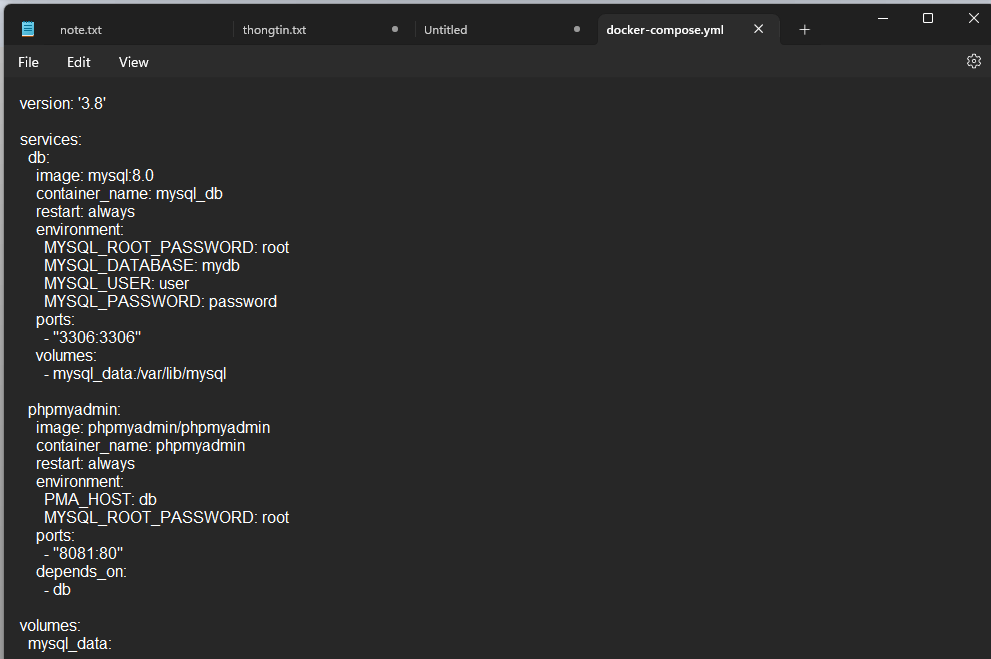
Chạy MySQL và PHPMyAdmin với Docker Compose.

PHPMyAdmin chạy trên cổng 8081.

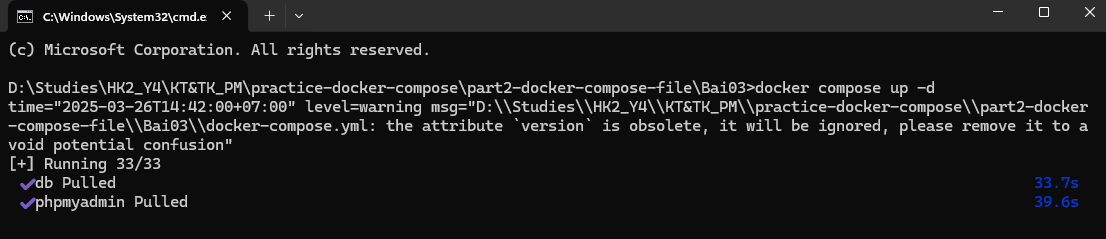
-- setup

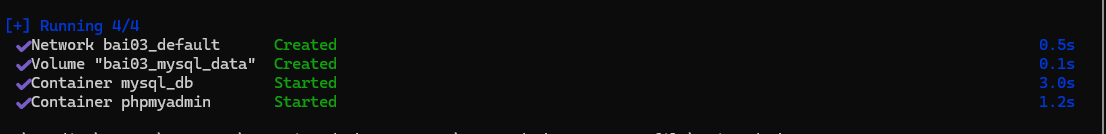


-- docker compose file

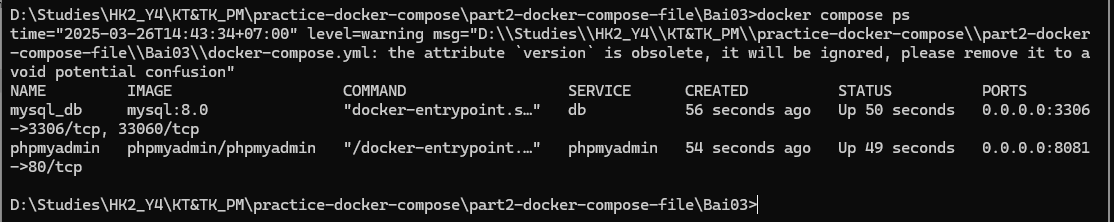


-- run docker compose





-- Kiểm tra container có chạy không ? **docker compose ps**



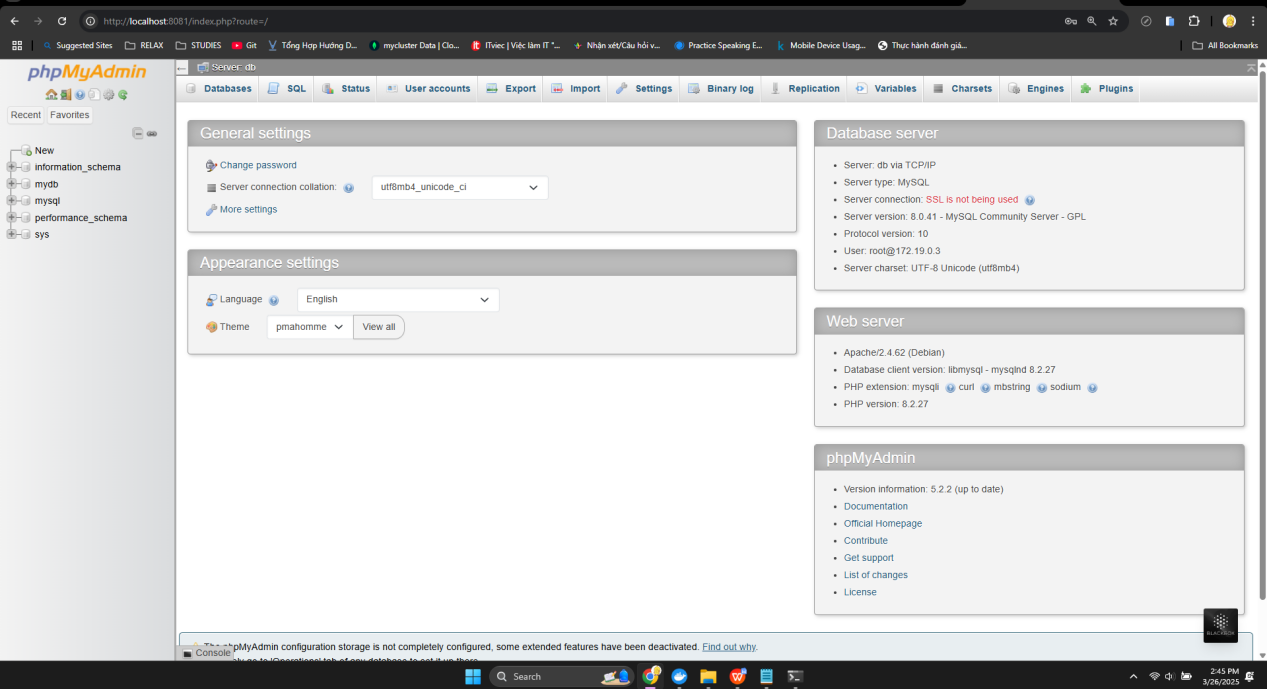
-- Mở trình duyệt và truy cập: http://localhost:8081

Thông tin đăng nhập PHPMyAdmin:

Server: db

User: root

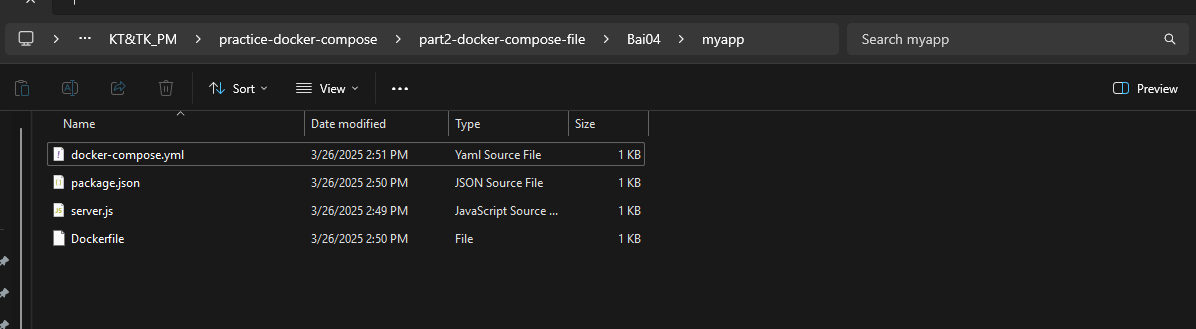
Password: root

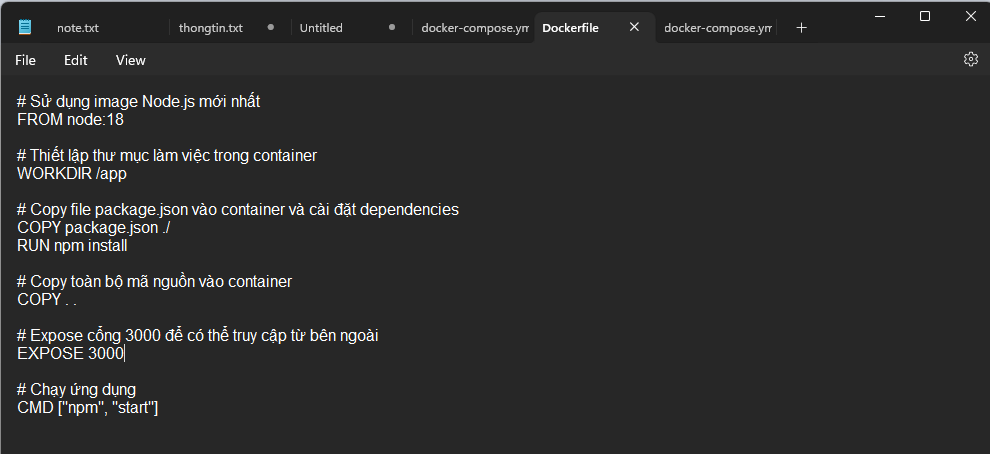


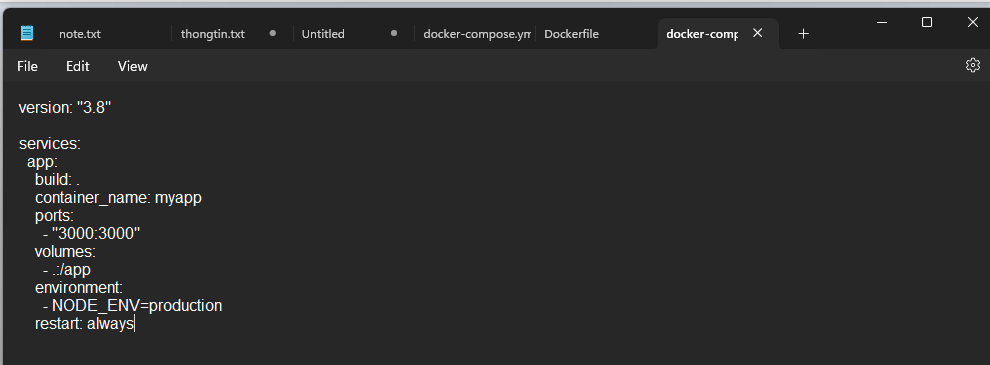
**Bài 4:** Chạy ứng dụng Node.js với Docker Compose

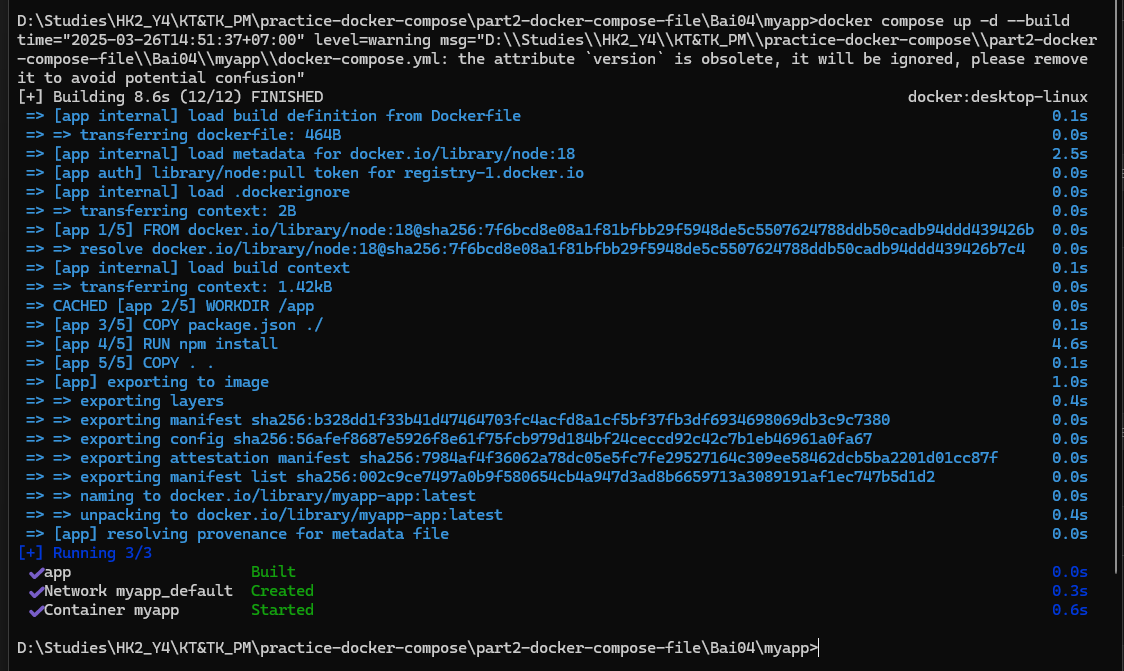
**Yêu cầu:**

Chạy một ứng dụng Node.js đơn giản với Express.



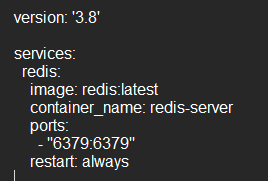






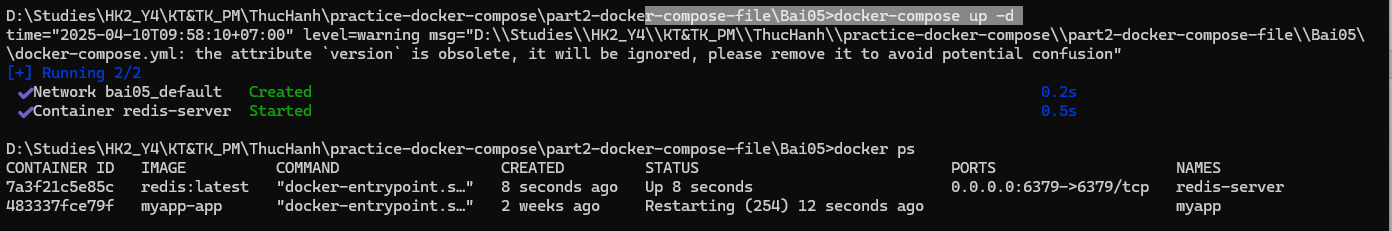
**Bài 5:** Chạy ứng dụng Redis với Docker Compose

- Tạo một file docker-compose.yml



- Run lệnh: docker-compose up -d

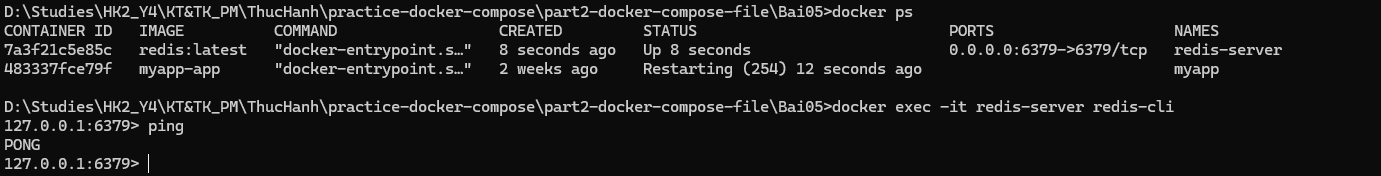
- Kiểm tra container Redis: docker ps



- Kiểm tra Redis: đã chạy chưa:

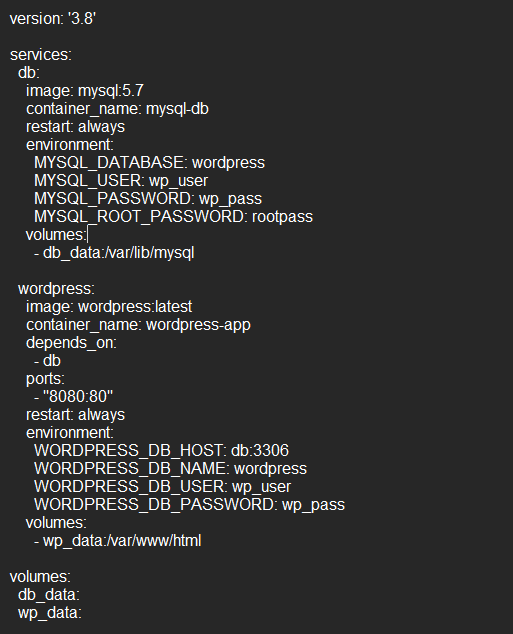
+ Kết nội vào Redis Container: **docker exec -it redis-server redis-cli**

+ Test: gõ “ping” --> kết quả thành công sẽ là: “**PONG”**

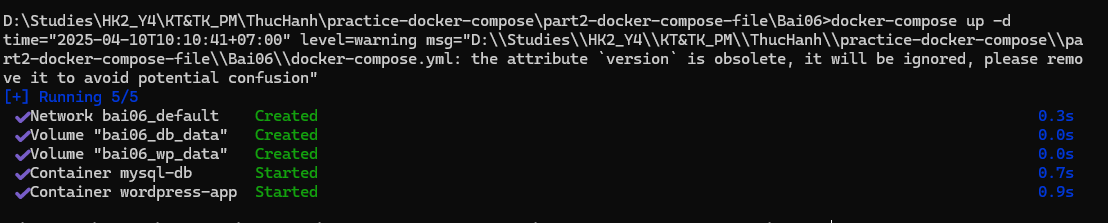


**Bài 6:** Chạy ứng dụng WordPress với mySQL

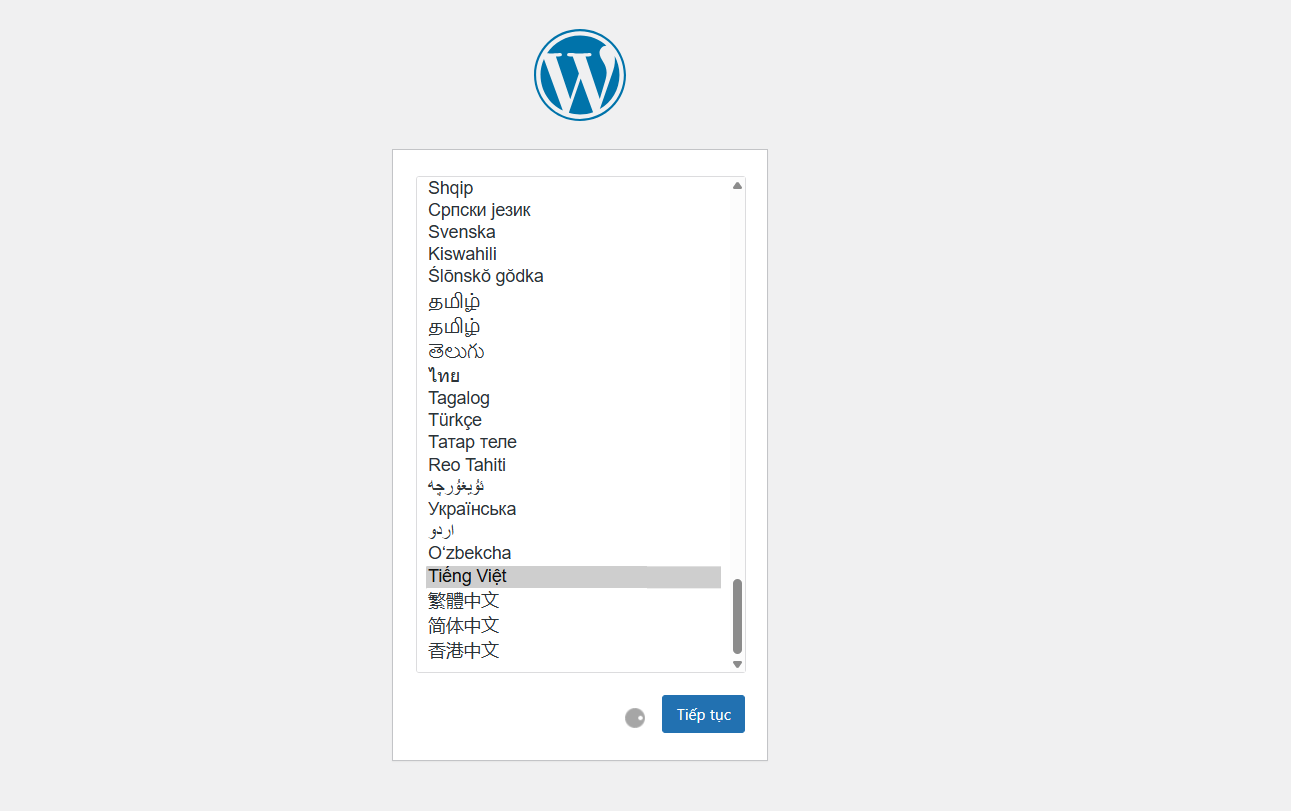
- Tạo một file docker-compose.yml

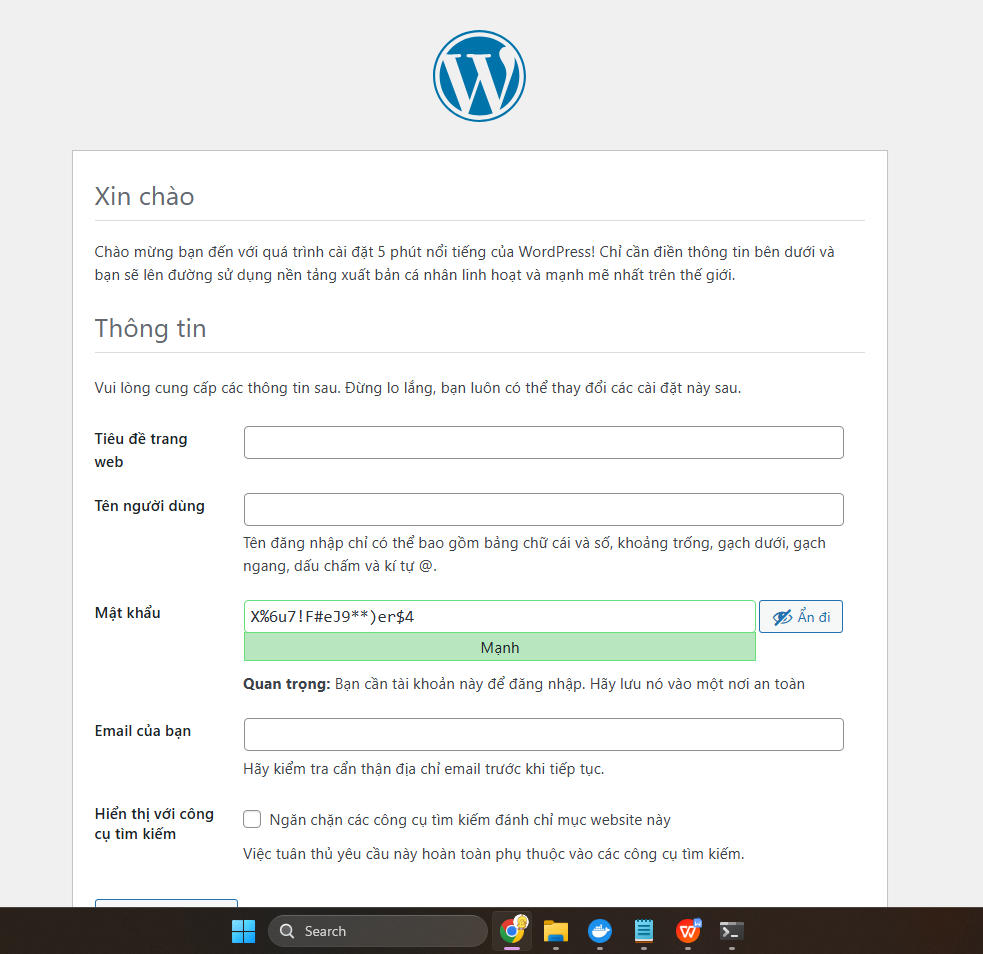


- Run lệnh: docker-compose up -d



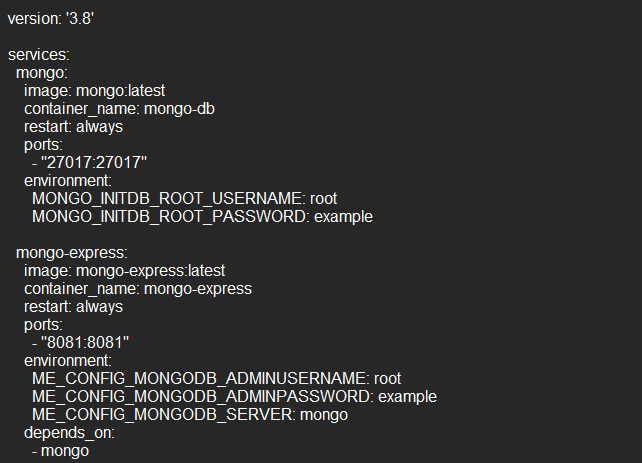
- Truy cập http://localhost:8080 để thiết lập WordPress (chọn ngôn ngữ, tài khoản admin...).



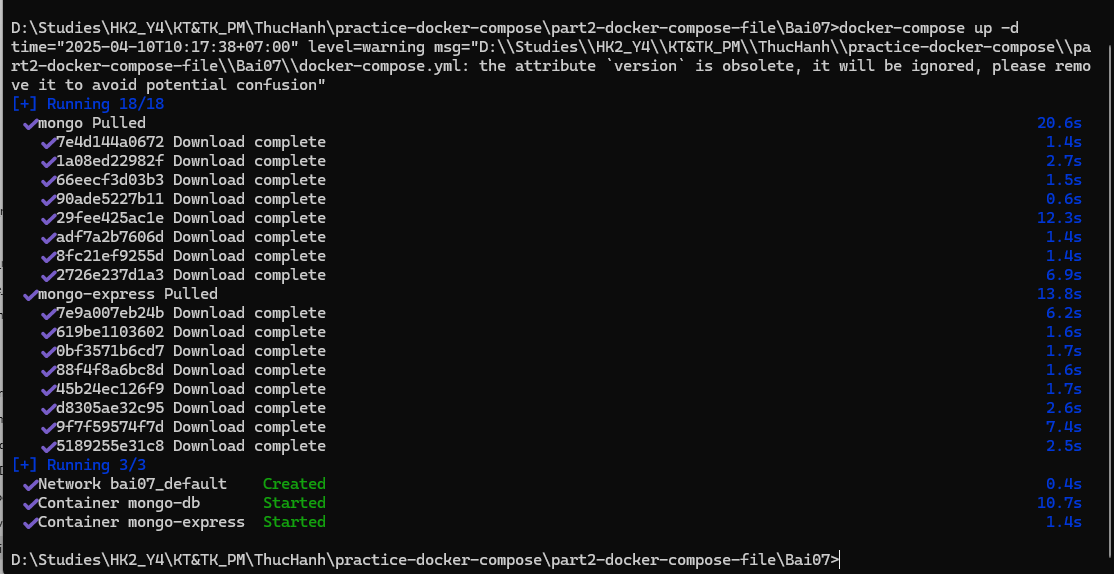


**Bài 7:** Chạy ứng dụng MongoDB với Docker Compose

- Tạo một file docker-compose.yml

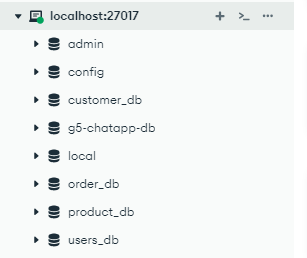


- Run lệnh: docker-compose up -d

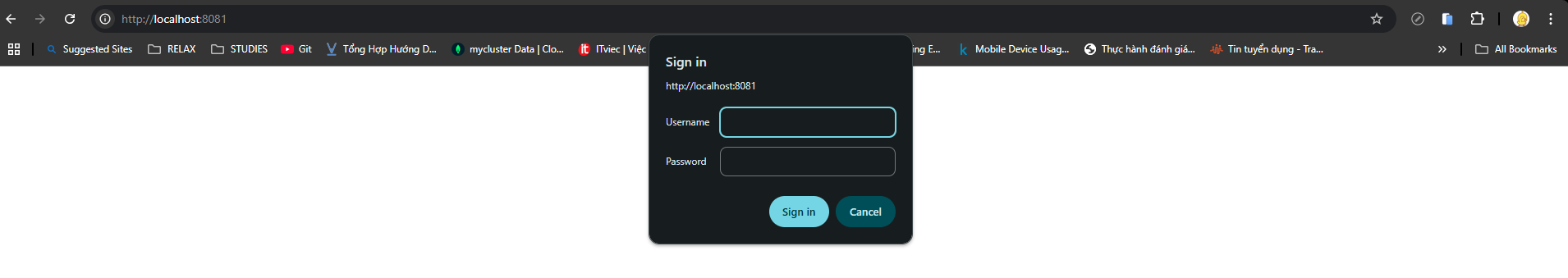


- Truy cập http://localhost:8080 để thiết lập WordPress (chọn ngôn ngữ, tài khoản admin...).

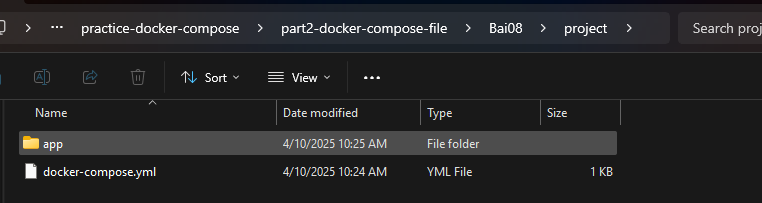
MongoDB server: mongodb://localhost:27017



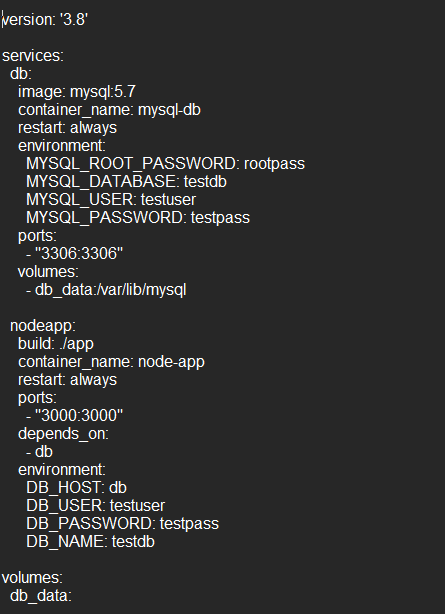
Giao diện Mongo Express: 👉 http://localhost:8081



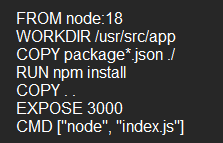
**Bài 8:** Kết nối nhiều dịch vụ với Docker Compose



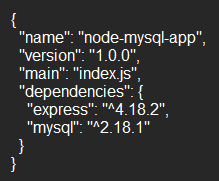
- Tạo một file docker-compose.yml



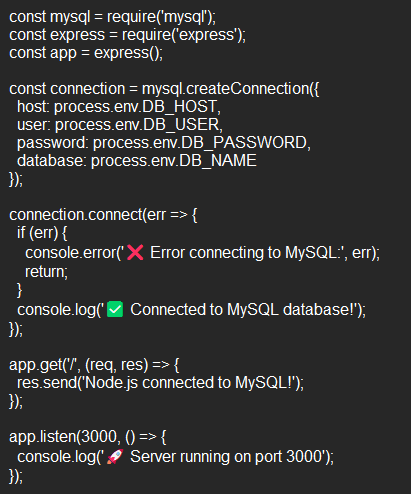
- Tạo một Dockerfile



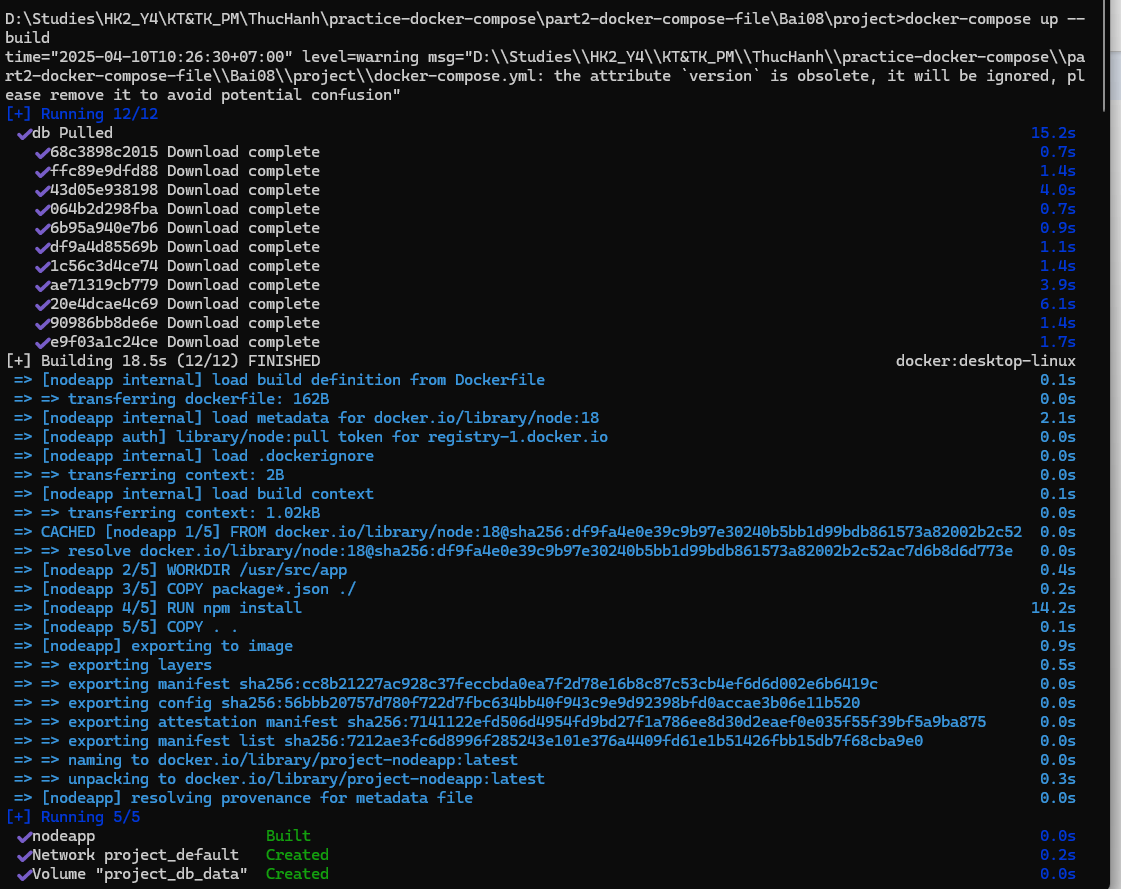
- Tạo một file package.json



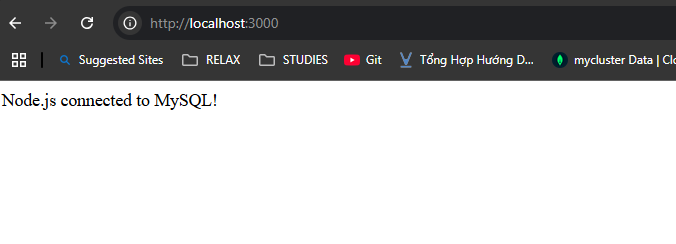
- Tạo một fle index.js



- Run lệnh: docker-compose up -d

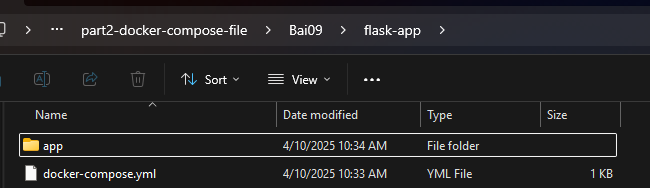


- Kết quả:

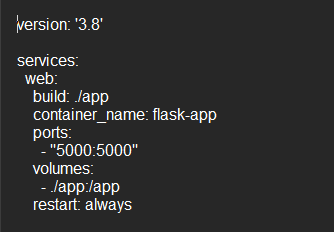


**Bài 9:** Chạy ứng dụng Python Flask với Docker Compose

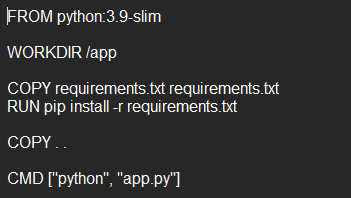
- Tạo cấu trúc thư mục như sau:



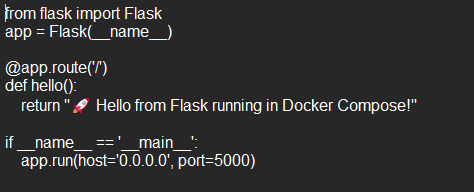
- Tạo docker-compose.yml



- Tạo Dockerfile

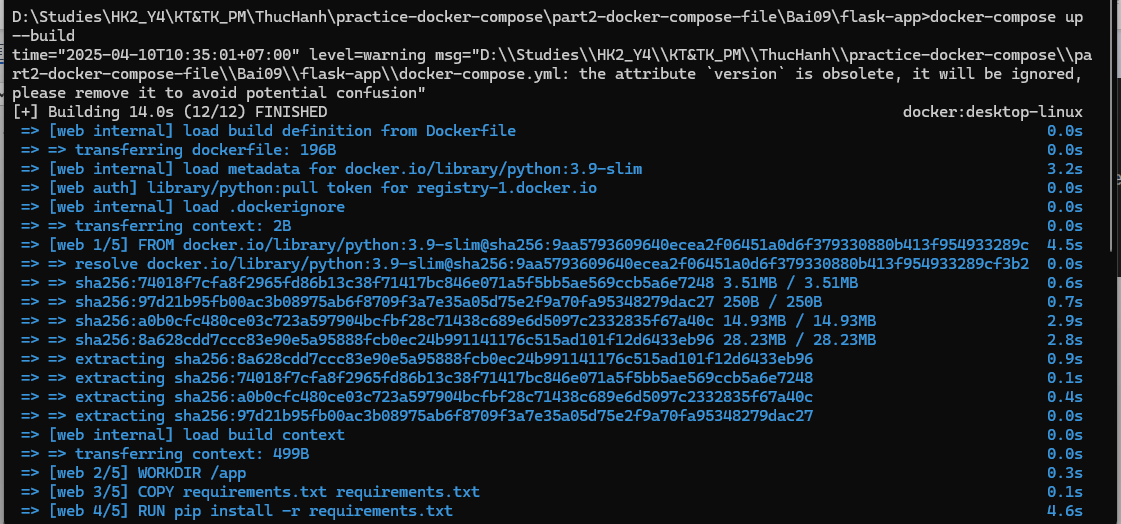


- Tạo app.py



- Tạo requirements.txt: nội dung **“flask”**

- Run lệnh: **docker-compose up -d**

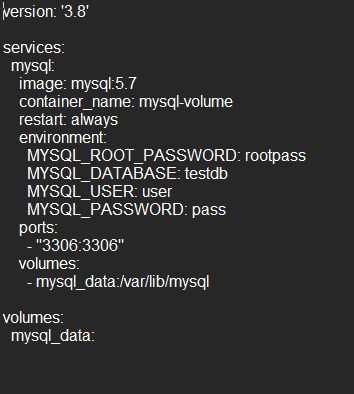


- Truy cập địa chỉ: **<http://localhost:5000>**

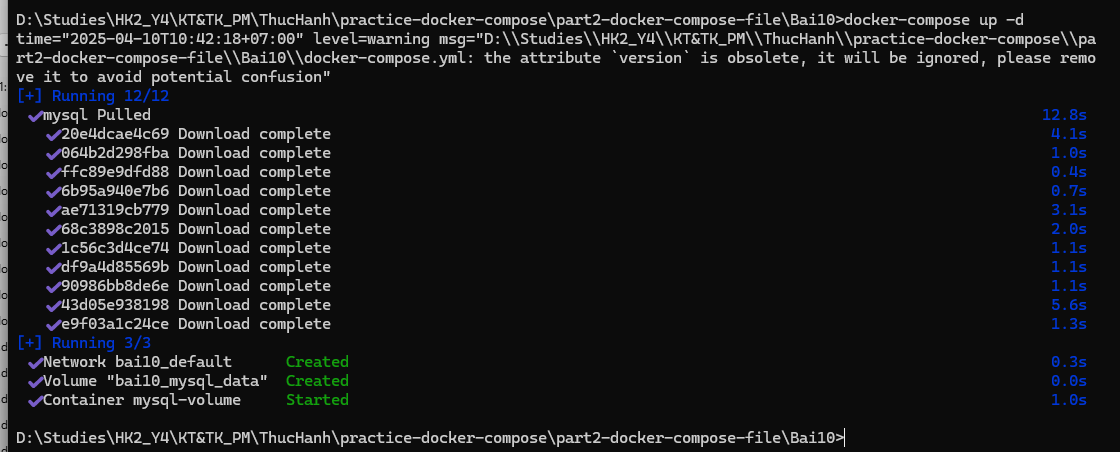


**Bài 10:** Lưu trữ dữ liệu với Docker Volumes

- Tạo file docker-compose.yml



- Run lệnh: **docker-compose up -d**



- Bước 1: Vào container: **docker exec -it mysql-volume bash**

- Bước 2: Vào MySQL: mysql -u root -p

# nhập: rootpass

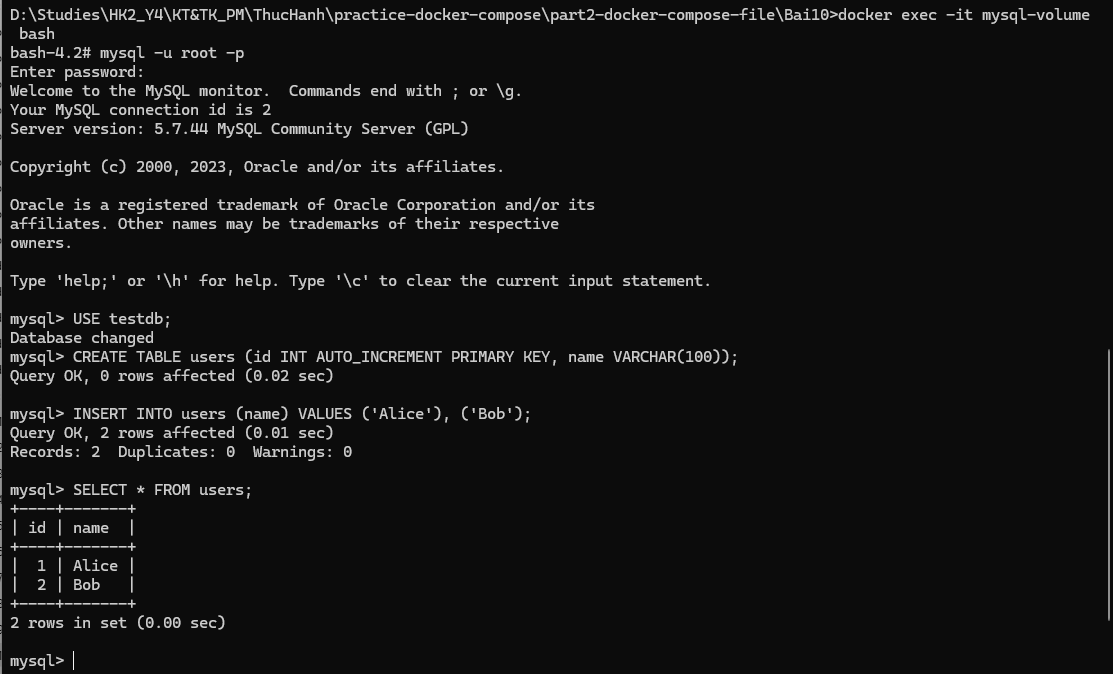
- Bước 3: Tạo bảng và chèn dữ liệu:

USE testdb;

CREATE TABLE users (id INT AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(100));

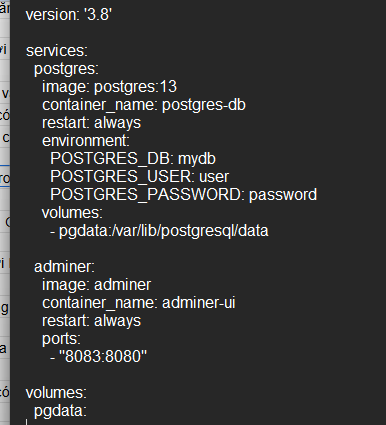
INSERT INTO users (name) VALUES ('Alice'), ('Bob');

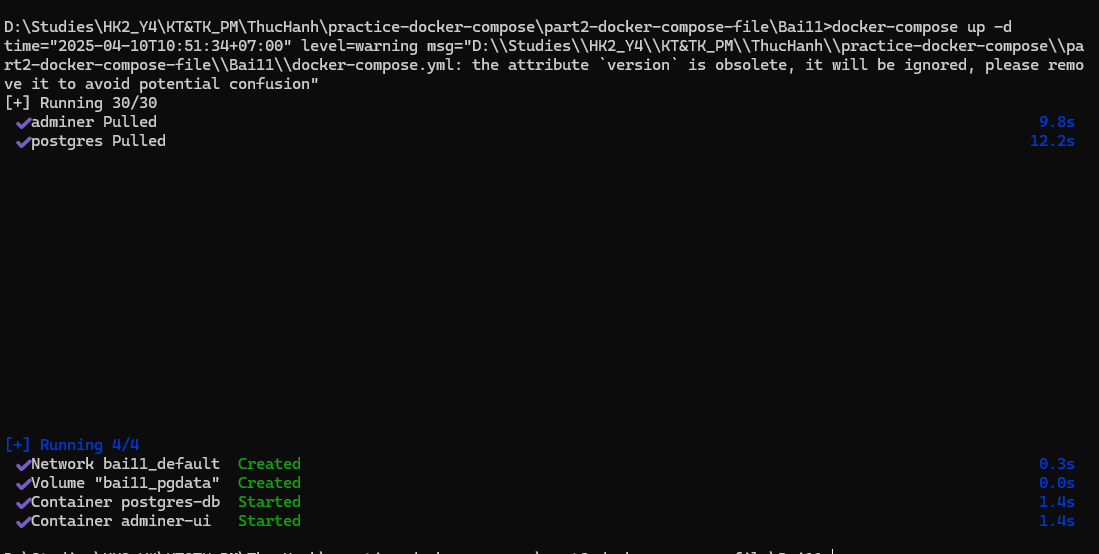
SELECT \* FROM users;



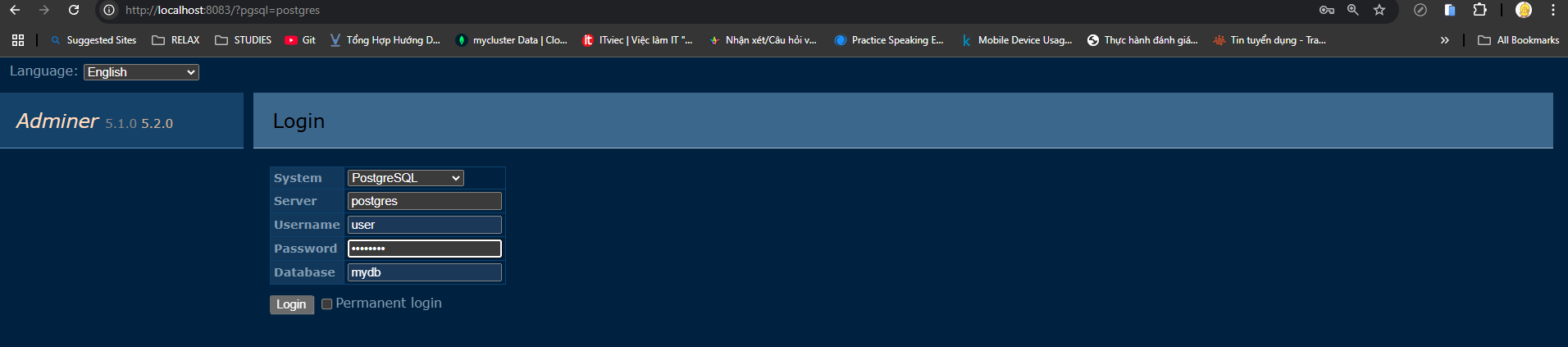
**Bài 11:** Chạy dịch vụ Postgres với Adminer

- Tạo file docker-compose.yml



- Run lệnh: **docker-compose up -d**

- Truy cập địa chỉ: **http://localhost:8083**



- Thông tin đăng nhập

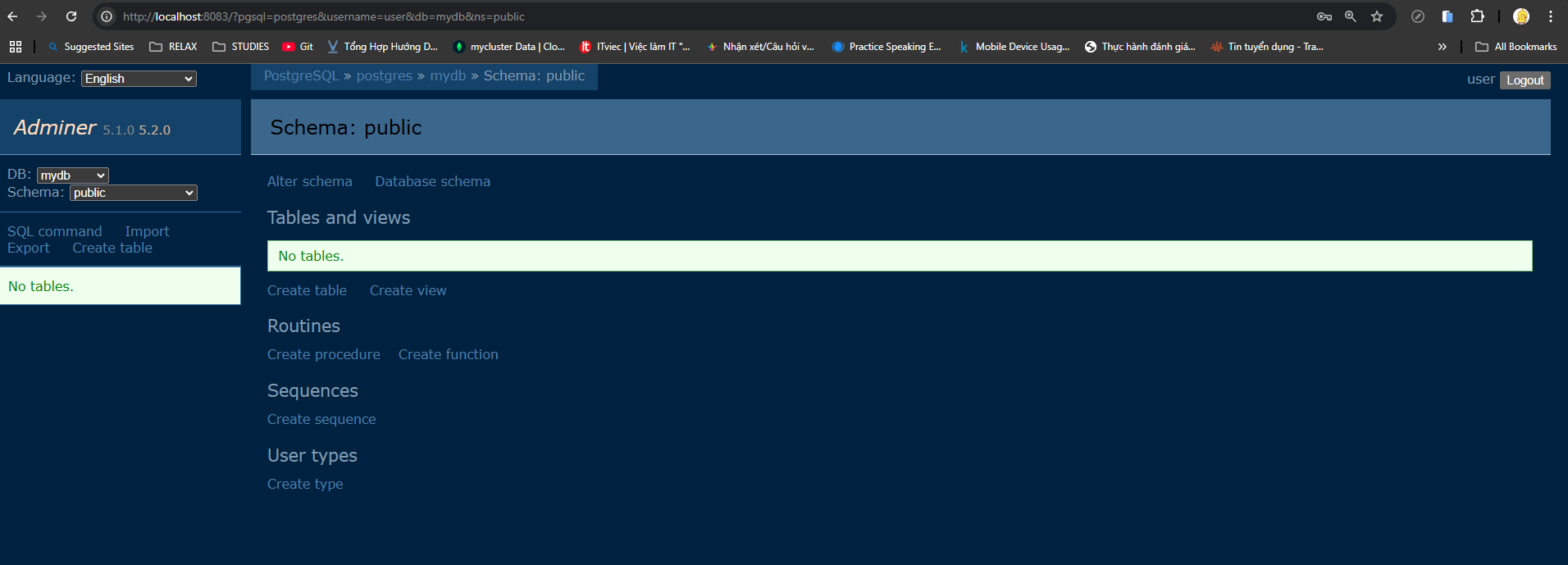
System: **PostgreSQL**

Server: **postgres**

Username **user**

Password **password**

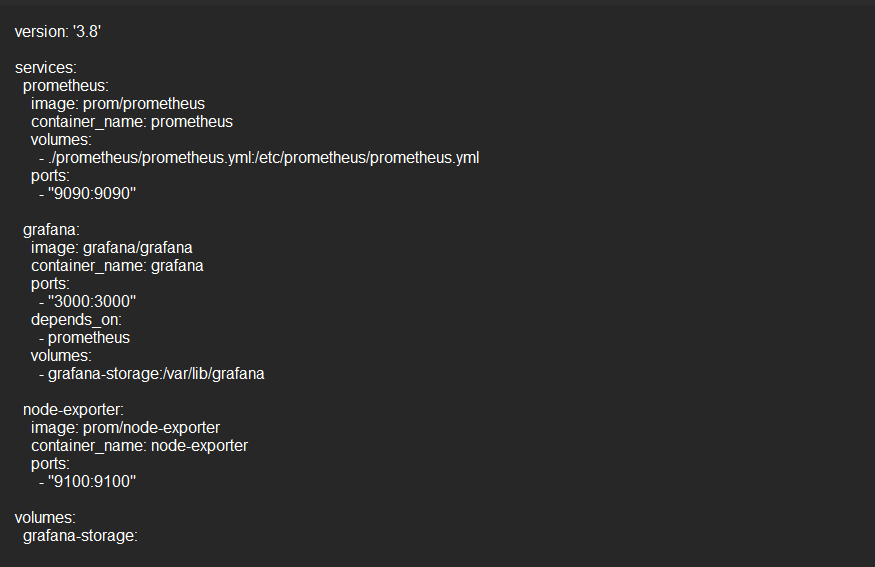
Database **mydb**



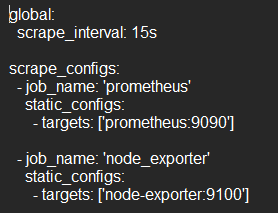
- Kết quả.

**Bài 12:** Giám sát container với Prometheus và Grafana

- Tạo file **docker-compose.yml**



- Tạo file **prometheus.yml**

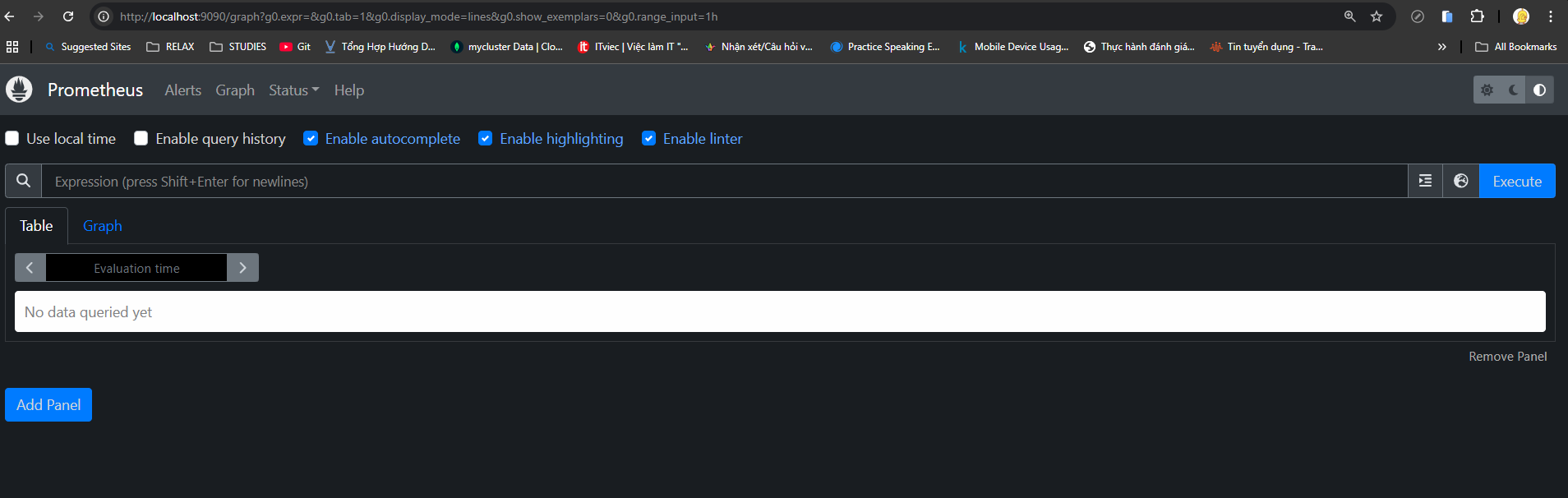


- Run lệnh: **docker-compose up -d**

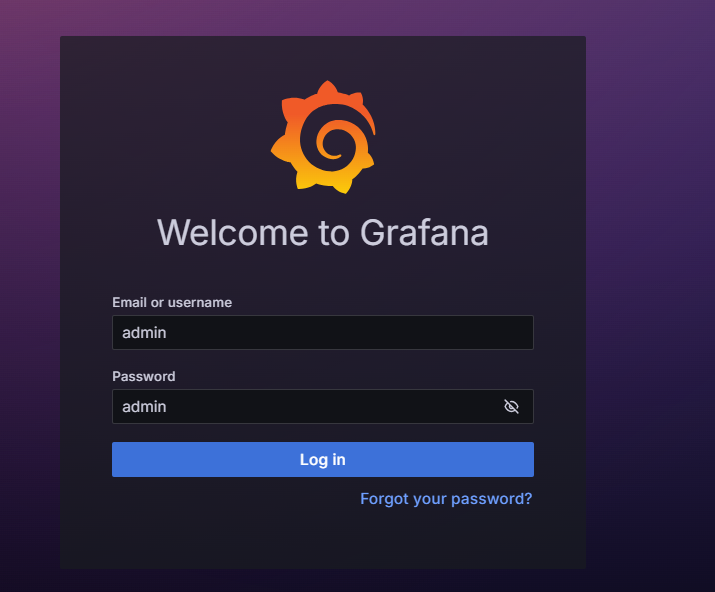


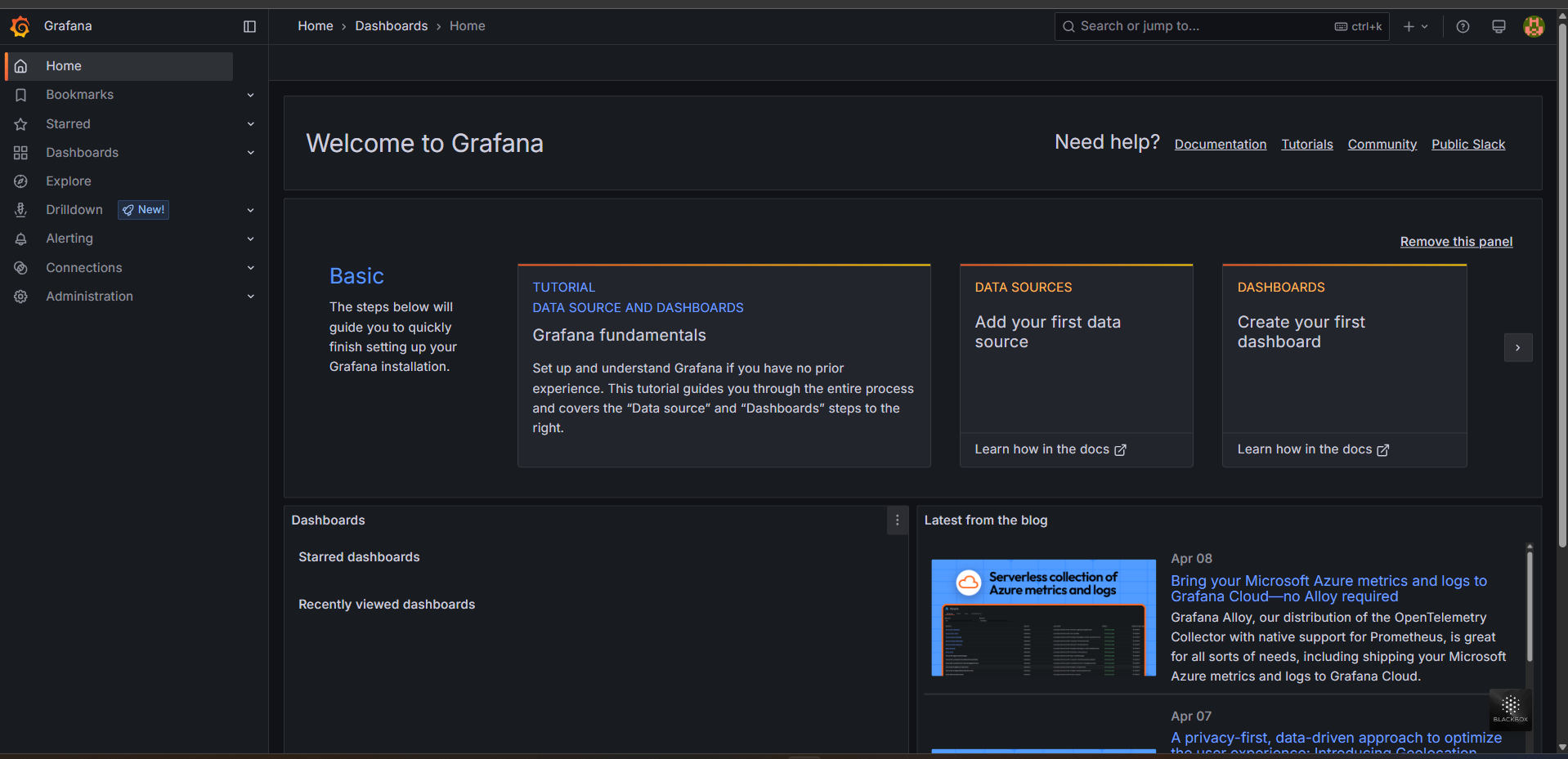
- Kết quả:

+ Prometheus [http:](http://localhost:9090)**[//localhost:9090](http://localhost:9090)**



+ Grafana **<http://localhost:3000>**





+ Node Exporter **<http://localhost:9100/metrics>**

