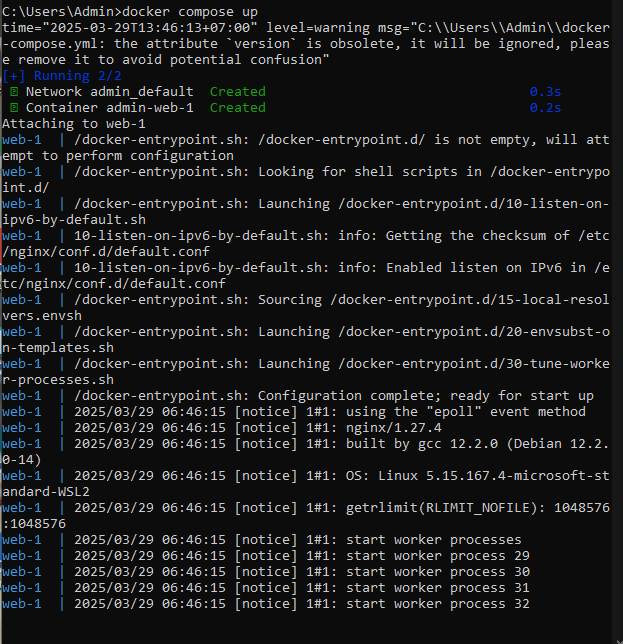
Phần 1: Một số lệnh Docker Compose cơ bản

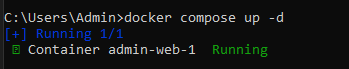
1 docker compose version



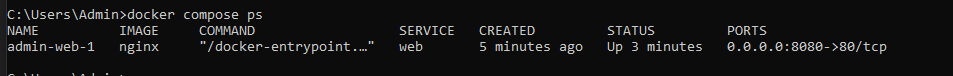
2 docker compose up



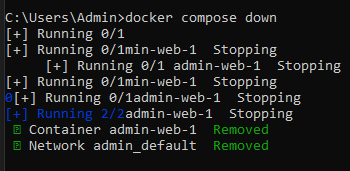
3 docker compose up -d



4 docker compose ps

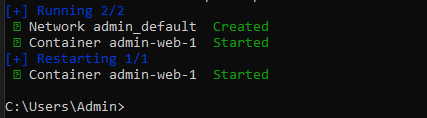


5 docker compose down



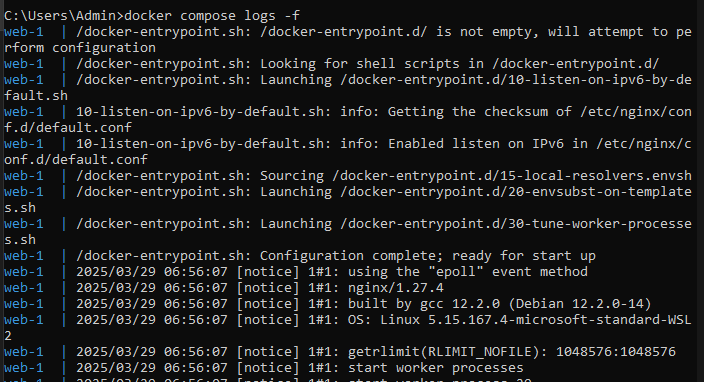
6 docker compose restart

=> khởi động lại tất cả các container đang chạy

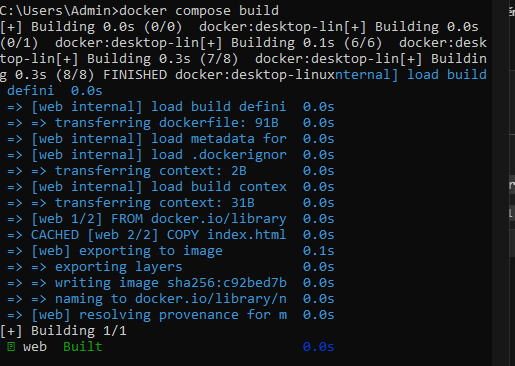


7 docker compose logs -f

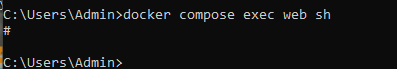
=> Hiển thị log đầu ra của tất cả container trong docker-compose.yml



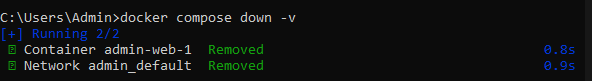
8 docker compose build



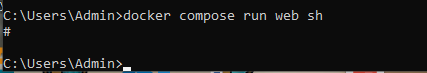
9 docker compose exec <service\_name> <command>



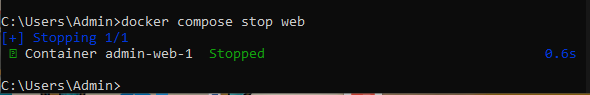
10 docker compose down -v



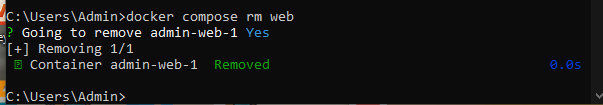
11 docker compose run <service\_name> <command>



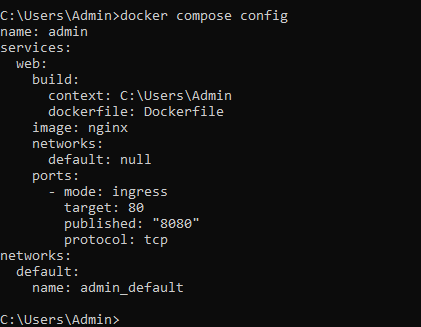
12 docker compose stop <service\_name>



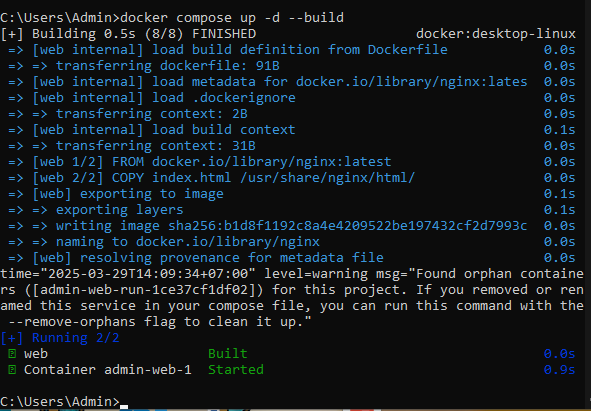
13 docker compose rm <service\_name>



14 docker compose config



15 docker compose up -d –build



Phần 2: Docker Compose file

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.

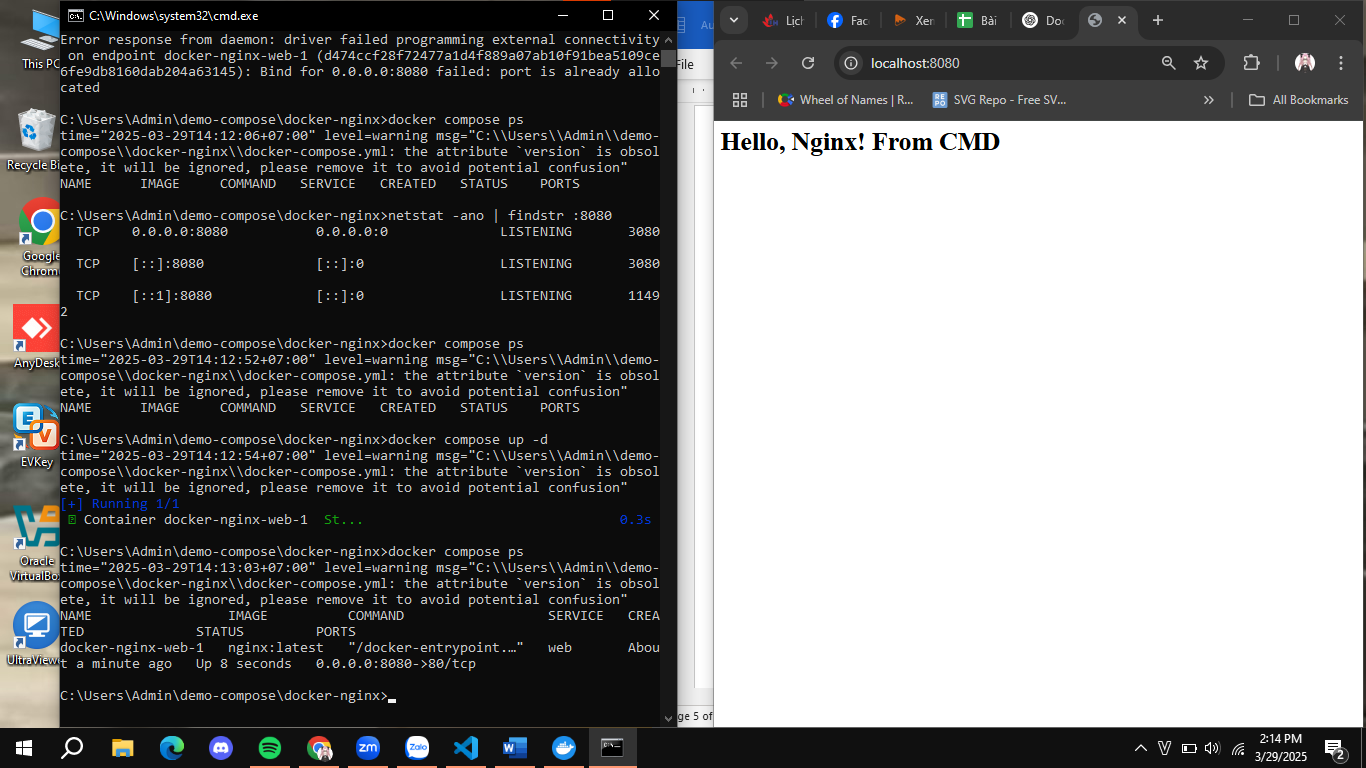
B1: tạo thư mục chứa project -> mkdir docker-nginx -> cd docker-nginx

B2: tạo docker-compose.yml -> notepad docker-compose.yml

B3: chạy container -> docker compose up -d

B4: kiểm tra -> docker compose ps/vào localhost:8080

Xong có thể chạy “docker compose down” để xoá container



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

B1: tạo thư mục chứa project -> mkdir docker-mysql-> cd docker-mysql

B2: tạo docker-compose.yml -> notepad docker-compose.yml

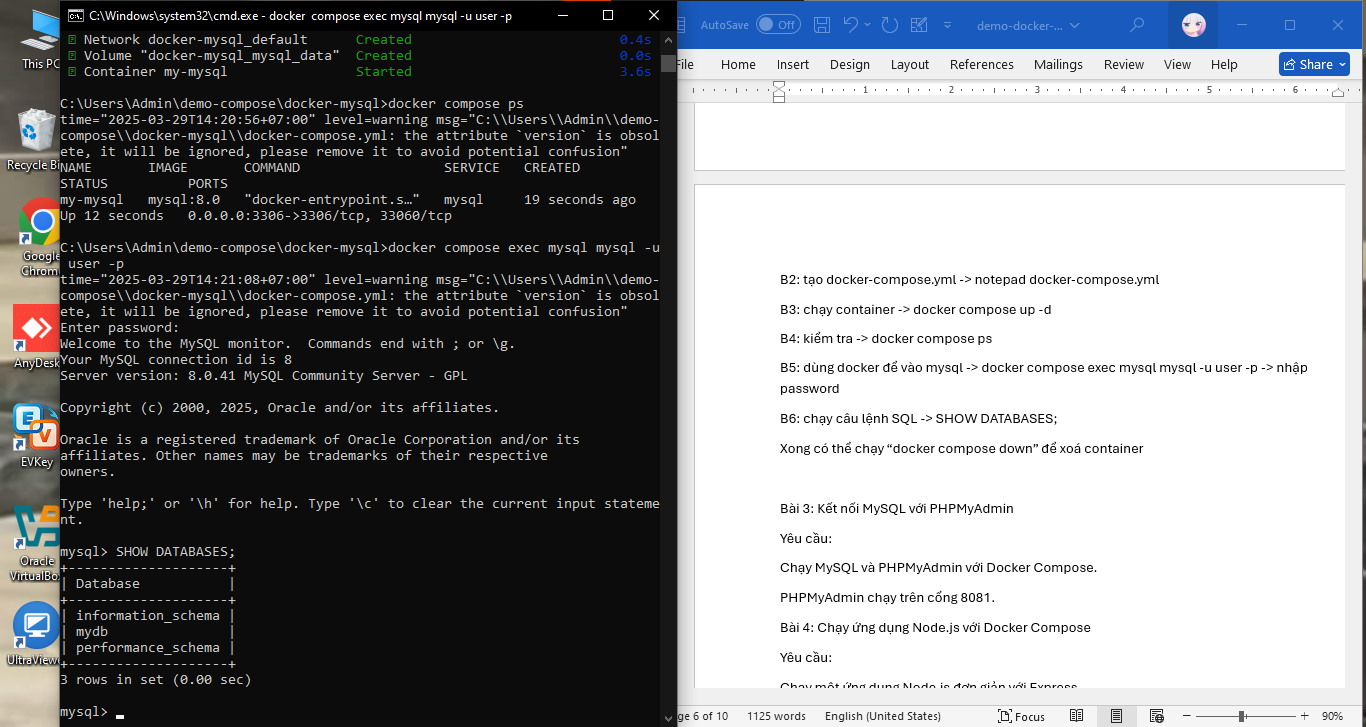
B3: chạy container -> docker compose up -d

B4: kiểm tra -> docker compose ps

B5: dùng docker để vào mysql -> docker compose exec mysql mysql -u user -p -> nhập password

B6: chạy câu lệnh SQL -> SHOW DATABASES;

Xong có thể chạy “docker compose down” để xoá container



Bài 3: Kết nối MySQL với PHPMyAdmin

Yêu cầu:

Chạy MySQL và PHPMyAdmin với Docker Compose.

PHPMyAdmin chạy trên cổng 8081.

B1: tạo thư mục chứa project -> mkdir docker-mysql-phpmyadmin-> cd

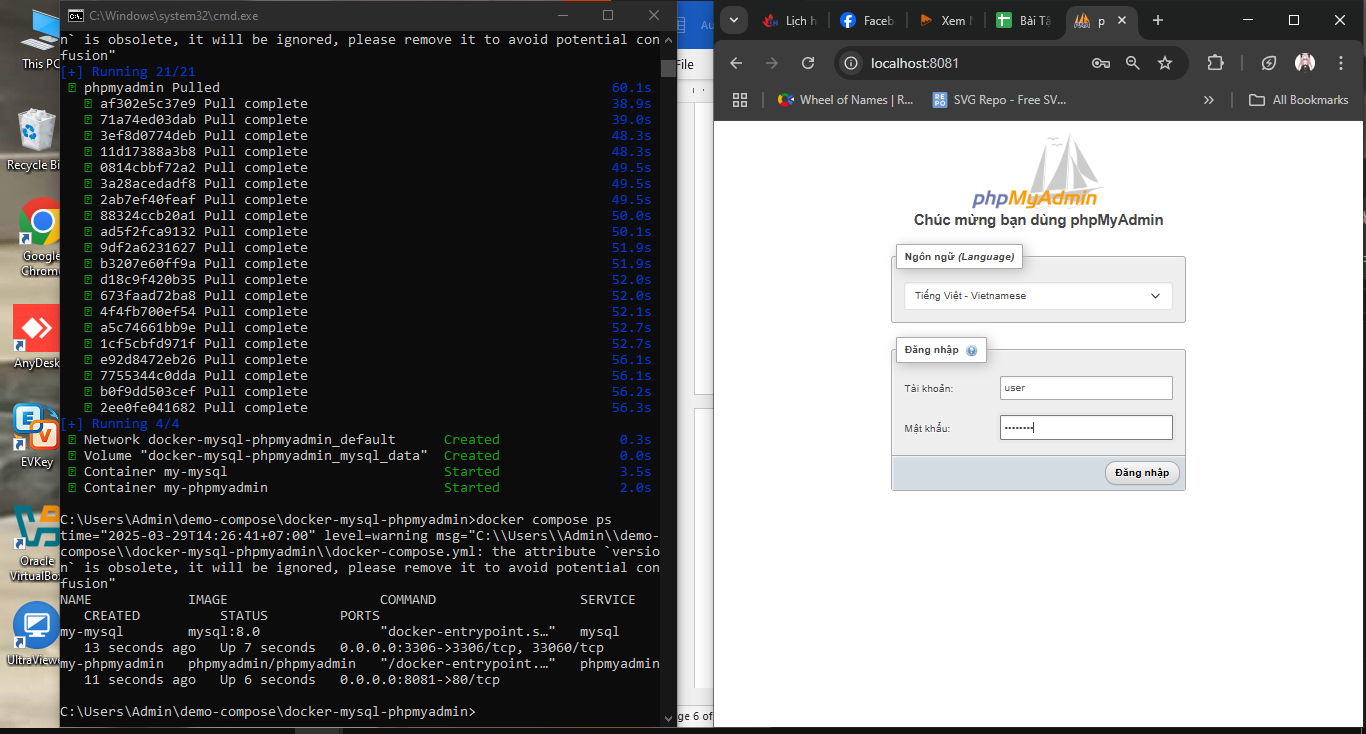
B2: tạo docker-compose.yml -> notepad docker-compose.yml

B3: chạy container -> docker compose up -d

B4: kiểm tra -> docker compose ps/vào localhost:8081

B5: đăng nhập -> nhập user và password

Xong có thể chạy “docker compose down” để xoá container



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

Yêu cầu:

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

B1: tạo thư mục chứa project -> mkdir docker-node-> cd

B2: tạo node project, install express

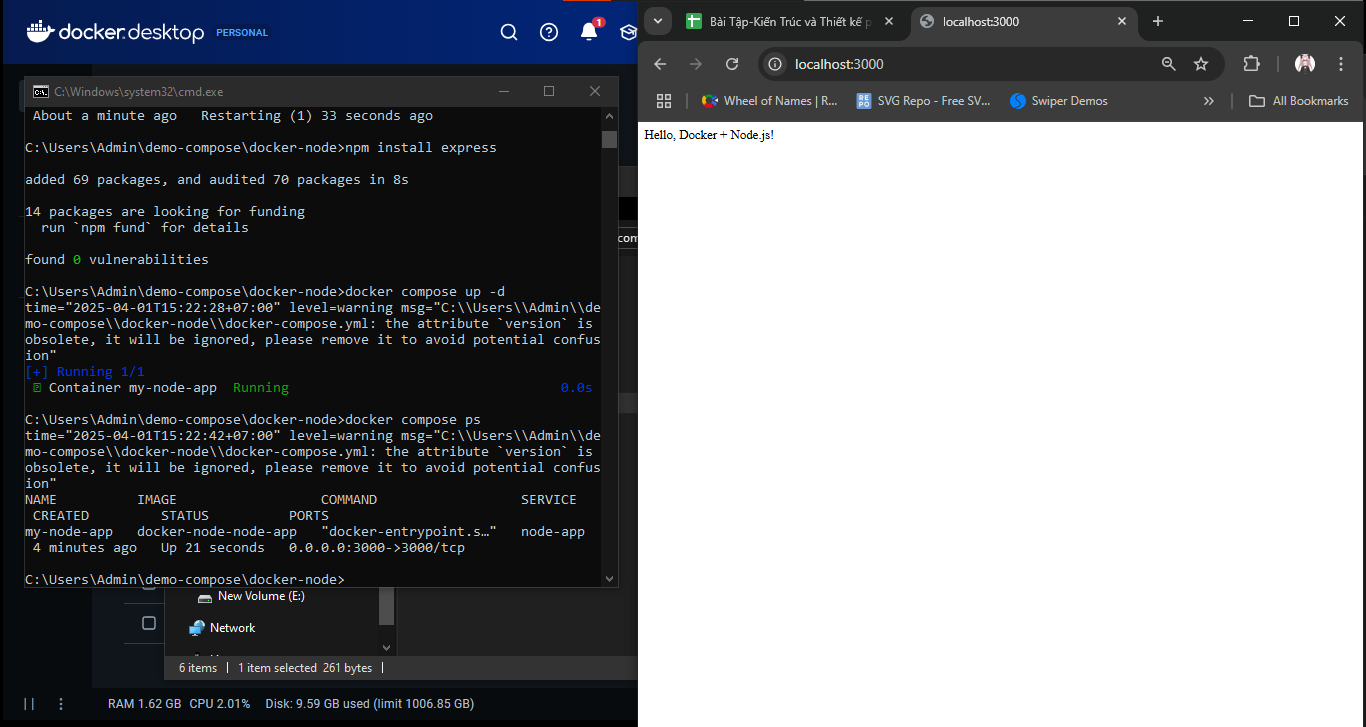
B3: tạo docker-compose.yml -> notepad docker-compose.yml

B4: tạo Dockerfile -> notepad Dockerfile

B5: chạy container -> docker compose up -d

B4: kiểm tra -> docker compose ps/vào localhost:3000

Xong có thể chạy “docker compose down” để xoá container



Bài 5: Chạy Redis với Docker Compose

Yêu cầu:

Chạy một container Redis trên cổng 6379

B1: tạo thư mục chứa project -> mkdir docker-redis-> cd

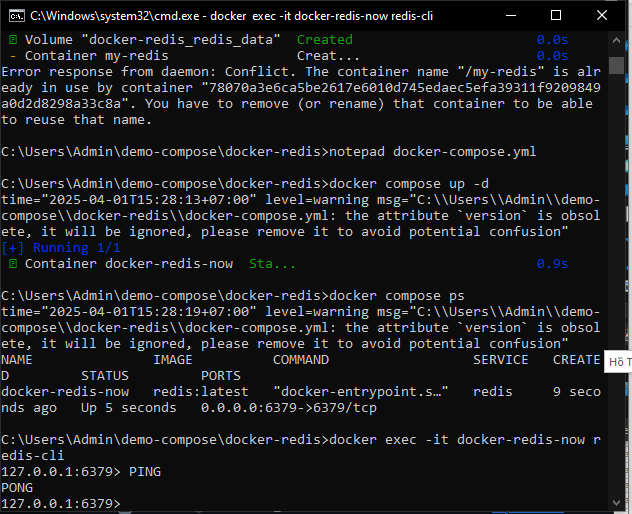
B2: tạo docker-compose.yml -> notepad docker-compose.yml

B3: chạy container -> docker compose up -d

B4: kết nối vào redis -> docker exec -it [container-name] redis-cli

B5: kiểm tra -> PING -> nhận lại PONG

Xong có thể chạy “docker compose down” để xoá container



Bài 6: Chạy WordPress với MySQL

Yêu cầu:

Chạy WordPress với MySQL bằng Docker Compose.

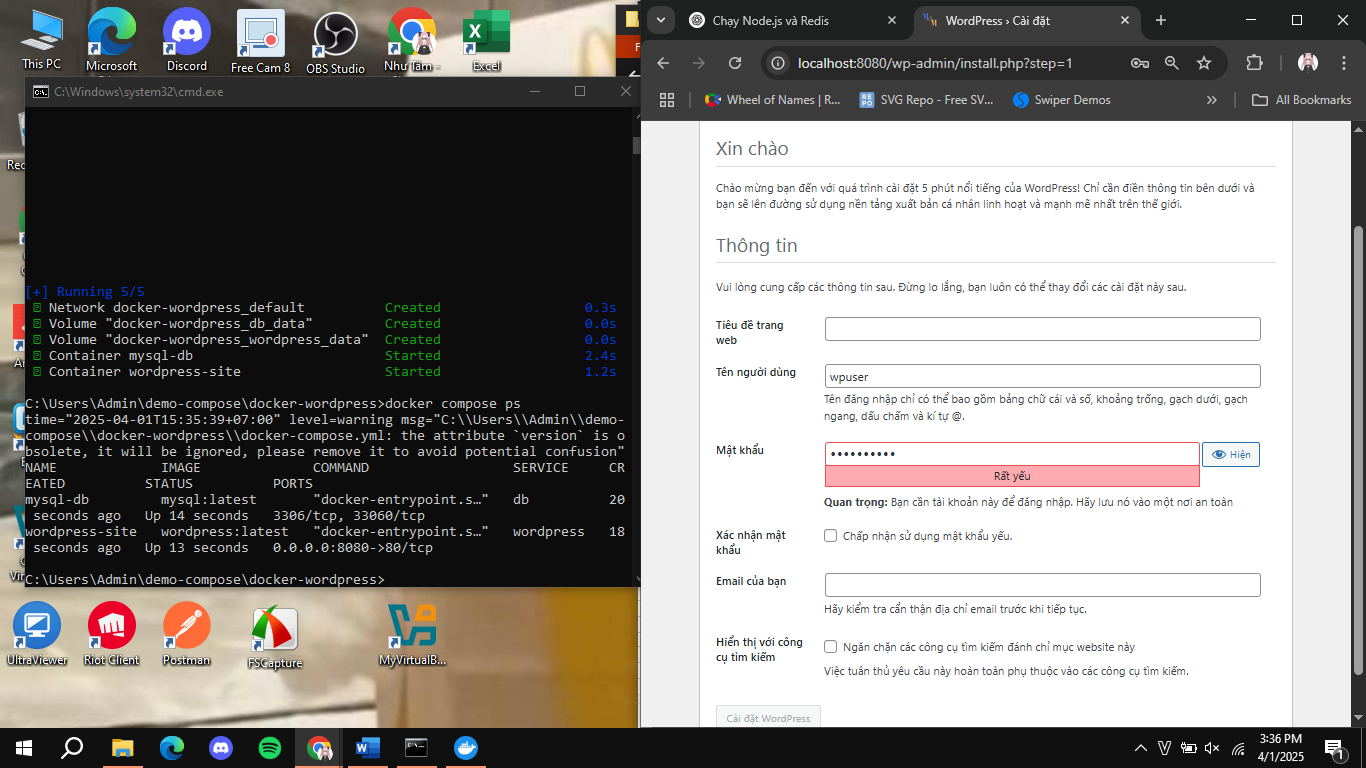
B1: tạo thư mục chứa project -> mkdir docker-wordpress-> cd

B2: tạo docker-compose.yml -> notepad docker-compose.yml

B3: chạy container -> docker compose up -d

B4: vào localhost:8080 đăng nhập và sử dụng

Xong có thể chạy “docker compose down” để xoá container



Bài 7: Chạy MongoDB với Docker Compose

Yêu cầu:

Chạy MongoDB và Mongo Express để quản lý.

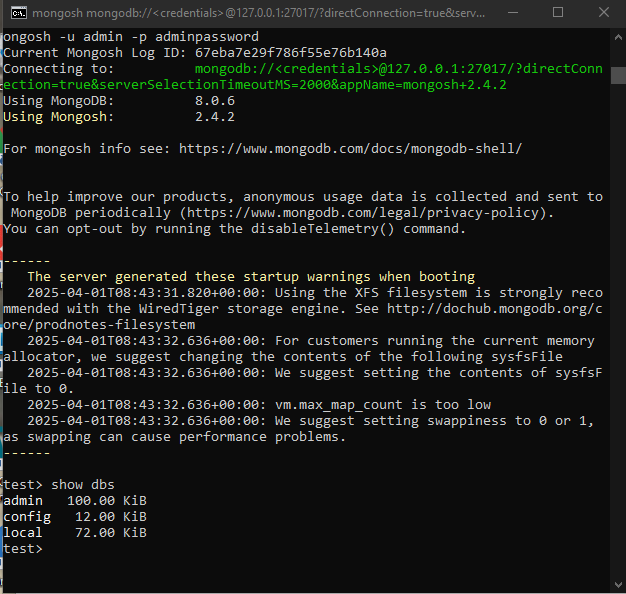
B1: tạo thư mục chứa project -> mkdir docker-mongo-express-> cd

B2: tạo docker-compose.yml -> notepad docker-compose.yml

B3: chạy container -> docker compose up -d

B4: kết nối vào mongo db -> docker exec -it mongo-db mongosh -u [username] -p [password]

Xong có thể chạy “docker compose down” để xoá container



Bài 8: Kết nối nhiều dịch vụ với Docker Compose

Yêu cầu:

Chạy Node.js kết nối với MySQL.

B1: tạo thư mục chứa project -> mkdir docker-node-mysql-> cd

B2: tạo node project, install các thư viện cần thiết

B3: tạo docker-compose.yml -> notepad docker-compose.yml

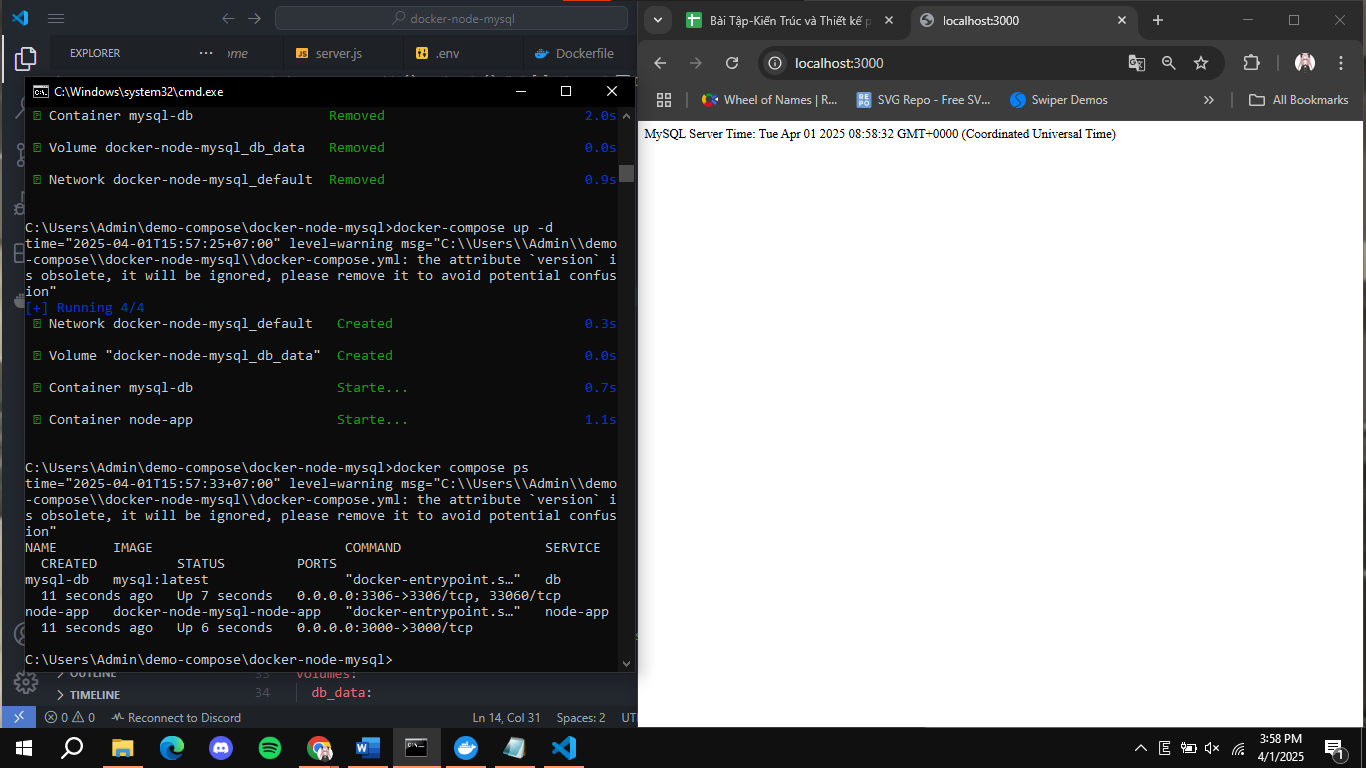
B4: tạo Dockerfile -> notepad Dockerfile

B5: chạy container -> docker compose up -d

B4: kiểm tra -> docker compose ps/vào localhost:3000

* Kết nối thành công sẽ hiện giờ hiện tại

Xong có thể chạy “docker compose down” để xoá container



Bài 9: Chạy ứng dụng Python Flask với Docker Compose

Yêu cầu:

Chạy ứng dụng Flask đơn giản với Docker Compose.

B1: tạo thư mục chứa project -> mkdir docker-node-mysql-> cd

B2: tạo flask project

B3: tạo docker-compose.yml -> notepad docker-compose.yml

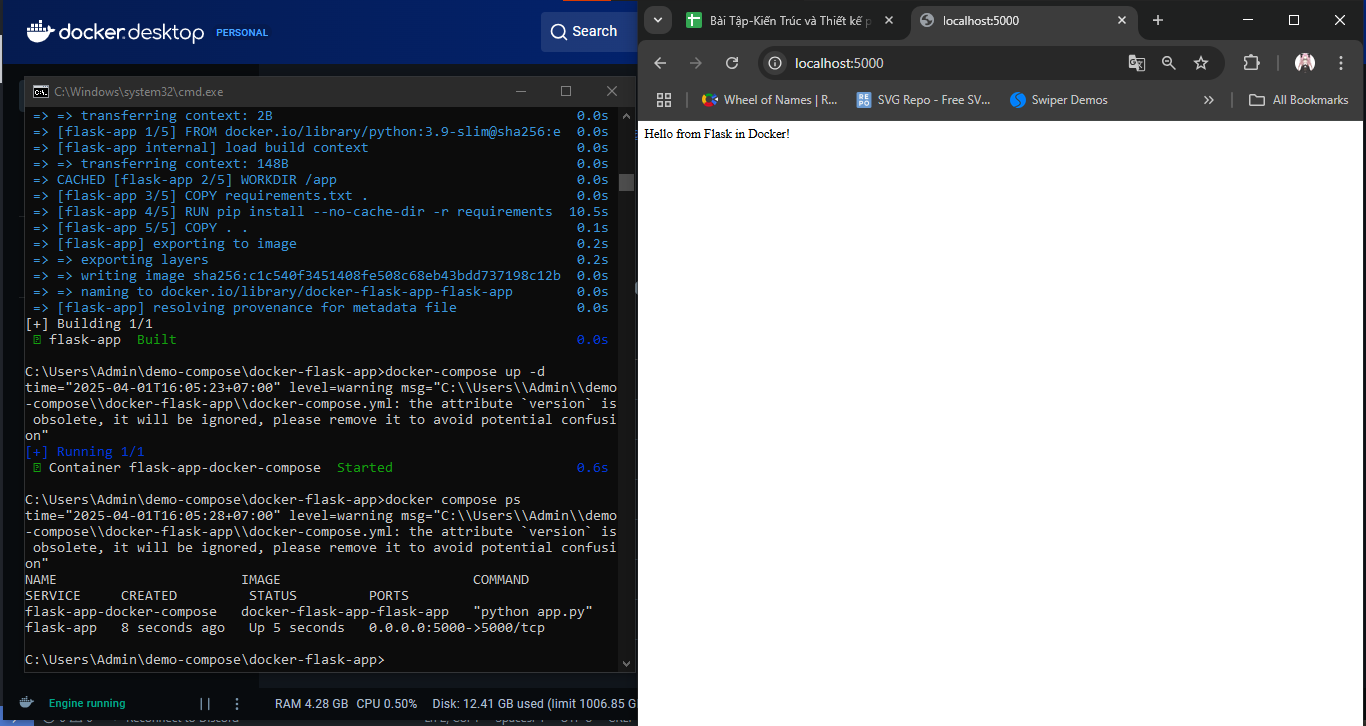
B4: tạo Dockerfile -> notepad Dockerfile

B5: build -> docker-compose build --no-cache

B5: chạy container -> docker compose up -d

B4: kiểm tra -> docker compose ps/vào localhost:5000

Xong có thể chạy “docker compose down” để xoá container



Bài 10: Lưu trữ dữ liệu với Docker Volumes

Yêu cầu:

Chạy MySQL và gắn volume để dữ liệu không bị mất.

B1: tạo thư mục chứa project -> mkdir docker-mysql-volume-> cd

B2: tạo docker-compose.yml -> notepad docker-compose.yml

B3: chạy container -> docker compose up -d

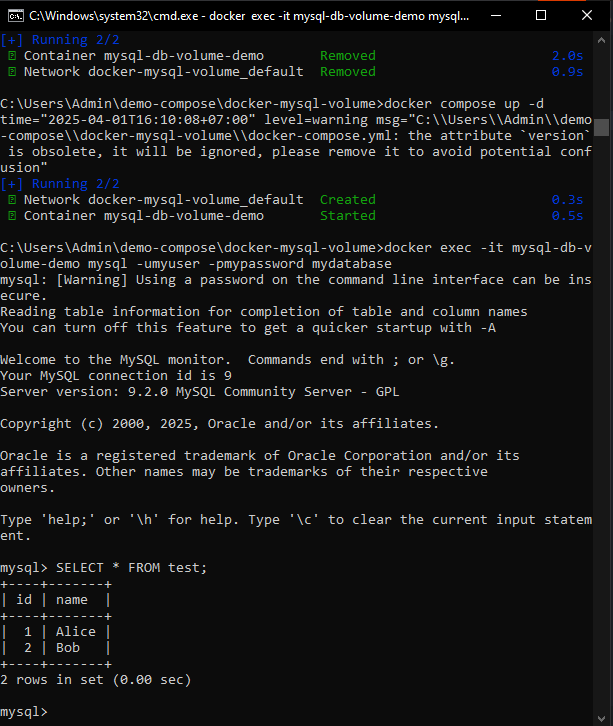
B4: kết nối mysql -> docker exec -it mysql-db-volume-demo mysql -umyuser -pmypassword mydatabase

B5: thêm dữ liệu

B6: dừng container -> docker compose down

B7: chạy lại và vào mysql kiểm tra xem dữ liệu thêm ở B5 còn không

Xong có thể chạy “docker compose down” để dừng container



Phần 3: Docker Compose file

Bài tập 1: Triển khai WordPress với MySQL

Mục tiêu: Tạo stack WordPress kết nối với MySQL, sử dụng volumes để lưu trữ dữ liệu.

Yêu cầu:

1. Sử dụng image wordpress:latest (port 80).

2. Sử dụng image mysql:5.7 (port 3306)

3. Volume cho database (/var/lib/mysql).

4. Biến môi trường cho MySQL

MYSQL\_ROOT\_PASSWORD, MYSQL\_DATABASE, MYSQL\_USER, MYSQL\_PASSWORD

Gợi ý:

WordPress cần khai báo depends\_on MySQL.

Sử dụng network tùy chỉnh để kết nối giữa 2 service.

Bài tập 2: Ứng dụng Node.js + MongoDB

Mục tiêu: Triển khai ứng dụng Node.js (lưu dữ liệu vào MongoDB) và MongoDB với volume.

Yêu cầu:

1. Viết Dockerfile cho ứng dụng Node.js (ví dụ: REST API đơn giản).

2. Sử dụng image mongo:latest (port 27017).

3. Volume cho MongoDB (/data/db).

4. Đảm bảo Node.js service khởi động sau MongoDB (depends\_on + healthcheck).

Bài tập 3: Load Balancing với Nginx + Flask

Mục tiêu: Cân bằng tải giữa 2 instance Flask dùng Nginx.

Yêu cầu:

1. 2 service Flask (sử dụng app.py từ bài tập trước, port 5000).

2. 1 service Nginx (port 8080) cấu hình làm reverse proxy:

Chuyển request / đến các Flask instance (round-robin).

Bài tập 4: Prometheus + Grafana Monitoring

Mục tiêu: Giám sát Docker containers dùng Prometheus và Grafana.

Yêu cầu:

1. Service Prometheus (port 9090) với file cấu hình thu thập metrics từ Docker.

2. Service Grafana (port 3000) kết nối đến Prometheus.

3. Volume để lưu dữ liệu Prometheus và Grafana.

Bài tập 5: Multi-tier Voting App

Mục tiêu: Triển khai ứng dụng voting gồm 5 services (Tham khảo từ Docker Docs).

Yêu cầu:

1. Frontend: vote (Python, port 5000).

2. Backend: result (Node.js, port 5001).

3. Redis (lưu tạm vote).

4. Worker (Java) xử lý vote từ Redis sang DB.

5. Postgres (lưu kết quả).