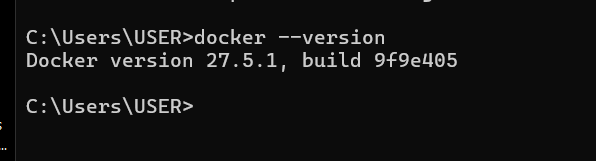
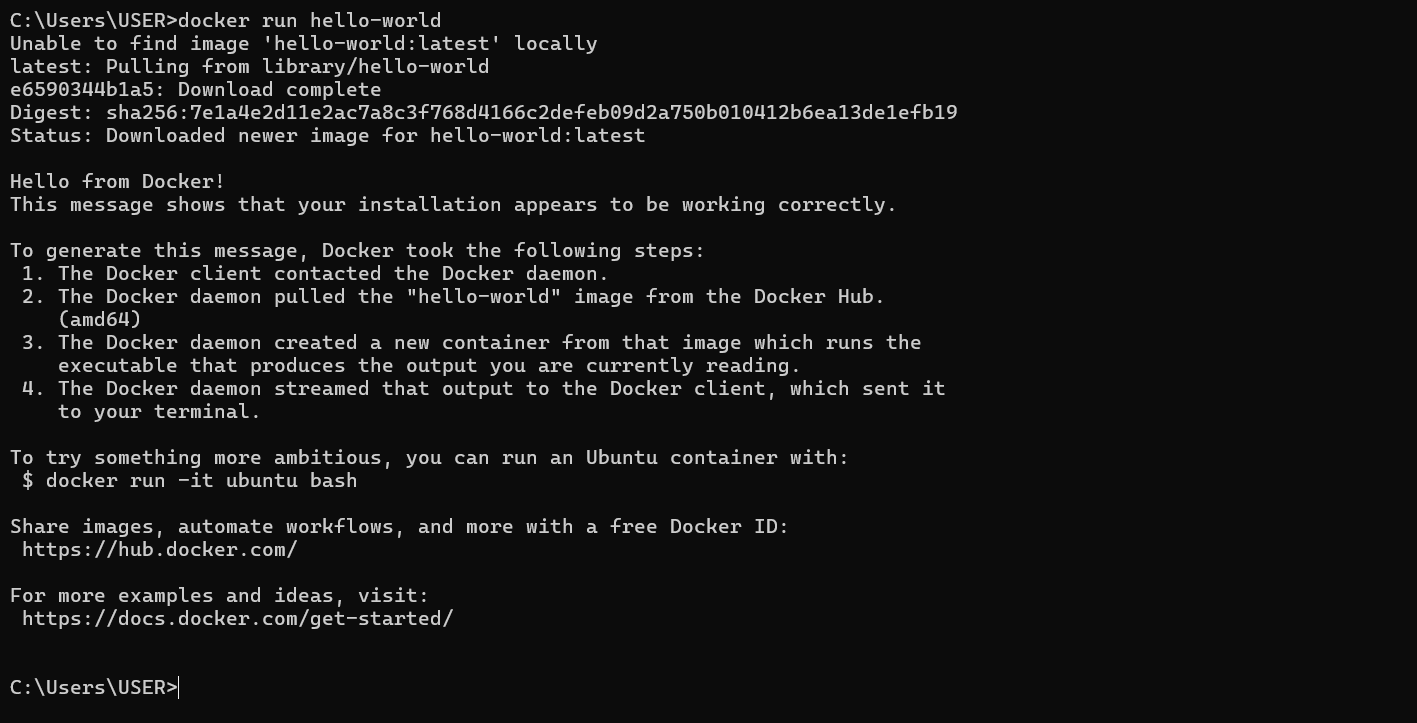
**Phần 1**

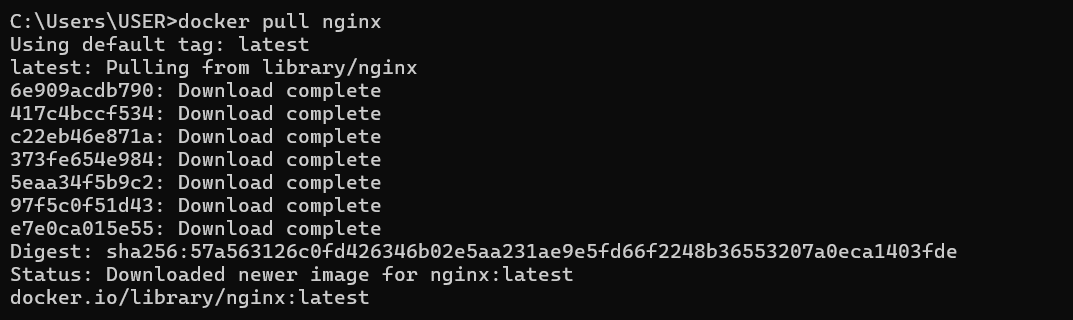
**1** docker --version



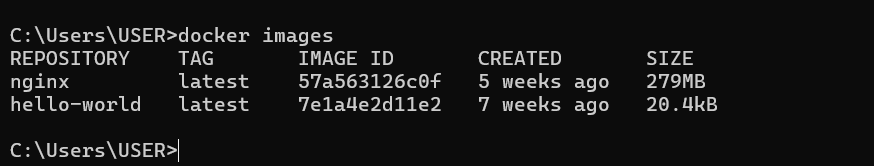
**2** docker run hello-world



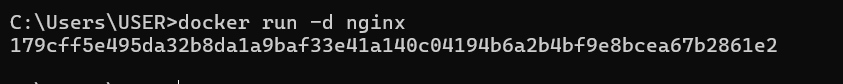
**3** docker pull nginx



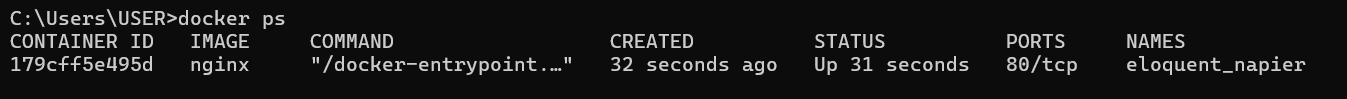
**4** docker images



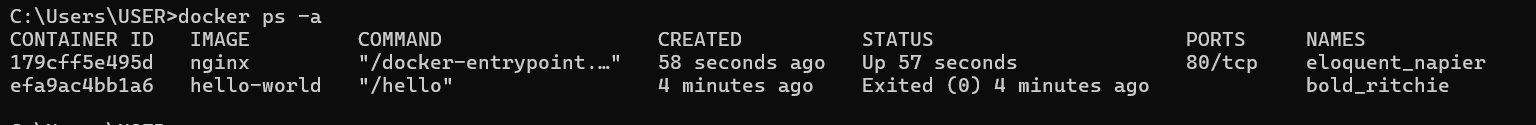
**5** docker run -d nginx



**6** docker ps

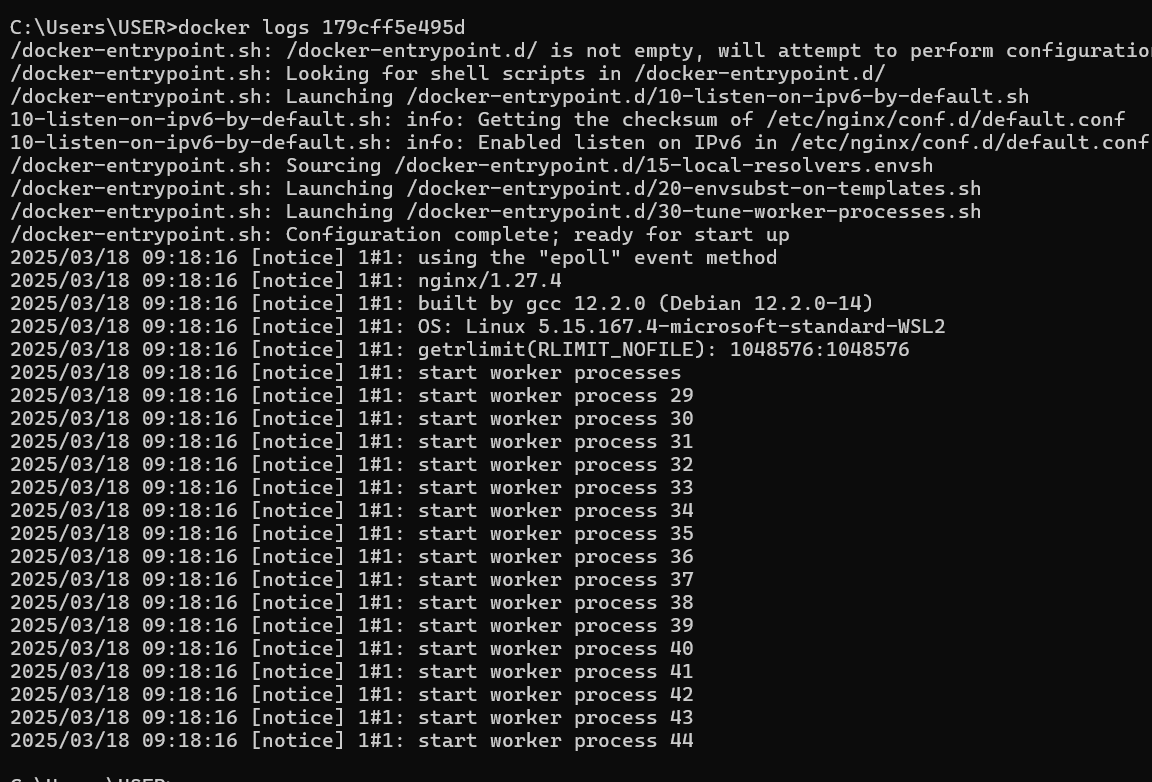


**7** docker ps -a



**8** docker logs <container\_id>

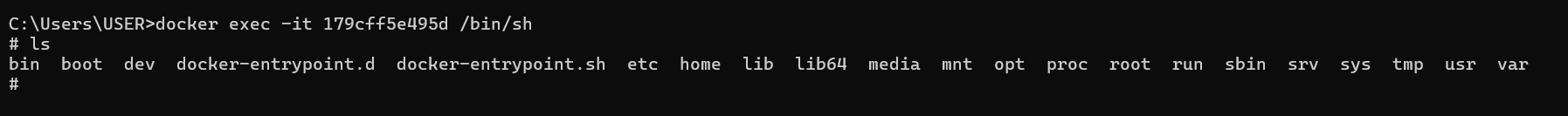
docker logs 179cff5e495d



**9** docker exec -it <container\_id> /bin/sh

Truy cập vào container đang chạy để thực thi lệnh shel

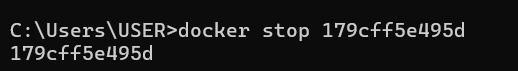
docker exec -it 179cff5e495d /bin/sh



**10** docker stop <container\_id>

Dừng container đang chạy

docker stop 179cff5e495d



**11** docker restart <container\_id>

Khởi động lại container.

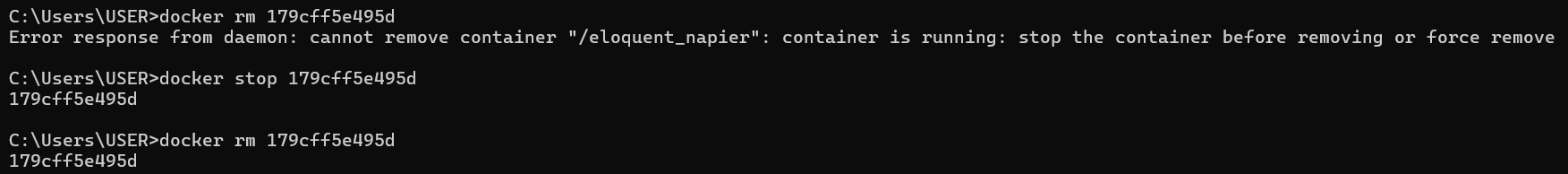
docker restart 179cff5e495d



**12** docker rm <container\_id>

Xóa container.

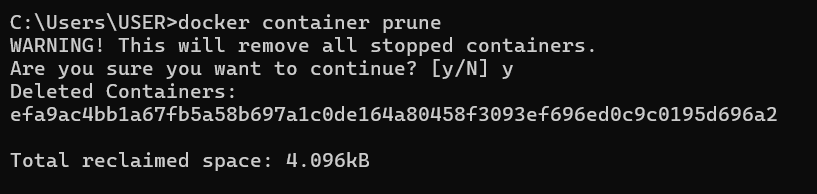
docker rm 179cff5e495d



**13** docker container prune

Xóa tất cả container đã dừng.

docker container prune

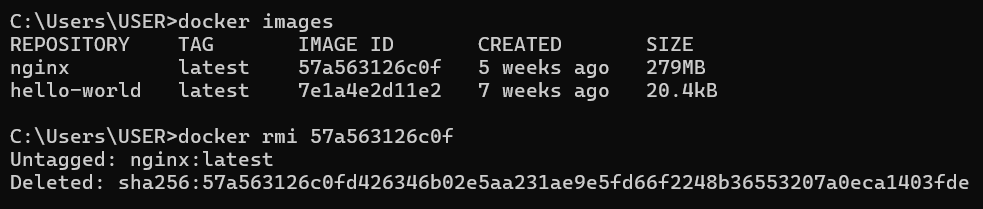




**14** docker rmi <image\_id>

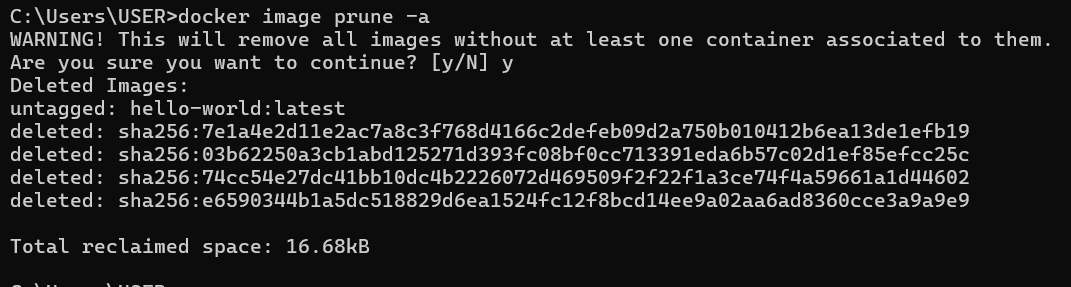
Xóa một image cụ thể.

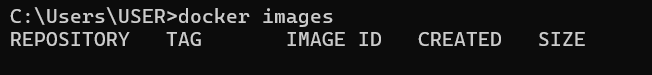
docker rmi 57a563126c0f



**15** docker image prune -a

Xóa tất cả image không được dùng.

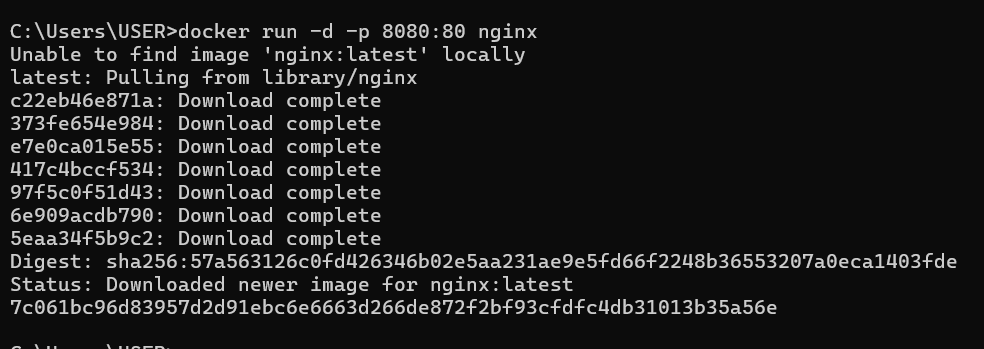




**16** docker run -d -data nginx

Chạy container Nginx, map cổng 8080 (host) với 80 (container)

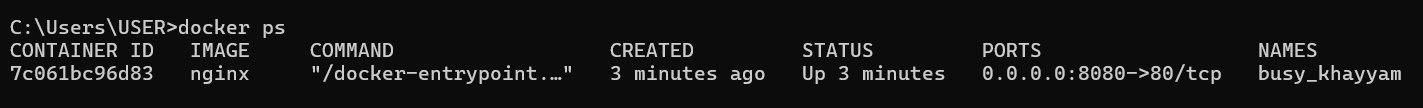
docker run -d -p 8080:80 nginx



**17** docker inspect <container\_id>

Xem thông tin chi tiết của container.

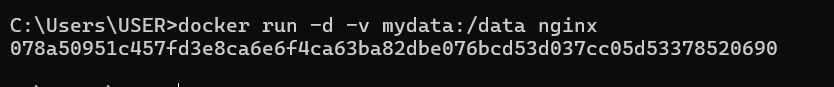
docker inspect 7c061bc96d83





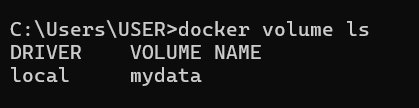
**18** docker run -d -v mydata:/data nginx

Chạy container Nginx với volume mydata gắn vào /data.



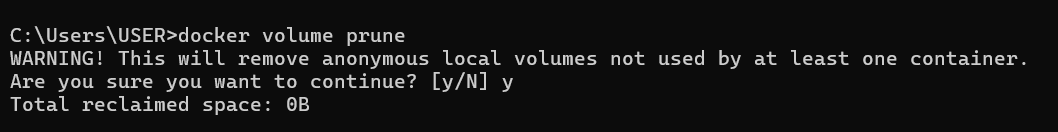
**19** docker volume ls

Liệt kê tất cả volume



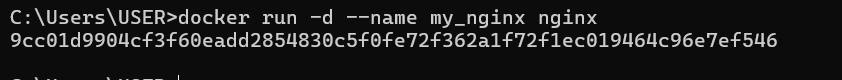
**20** docker volume prune

Xóa tất cả volume không được dùng.

****

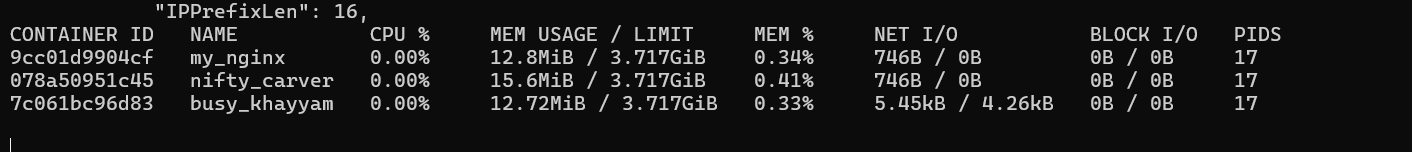
**21** docker run -d --name my\_nginx nginx

Chạy container Nginx với tên my\_nginx.



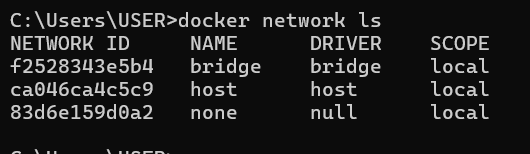
**22** docker stats

Xem tài nguyên (CPU, RAM,...) của container đang chạy



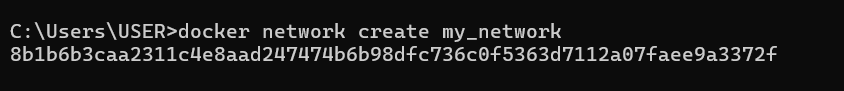
**23** docker network ls

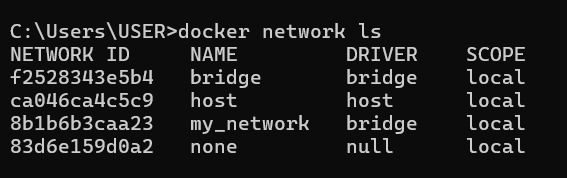
Liệt kê các network.



**24** docker network create my\_network

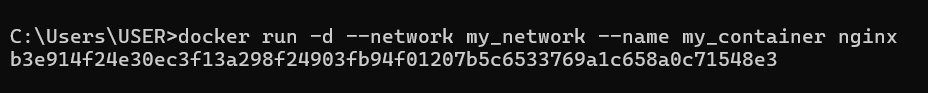
Tạo network mới tên my\_network





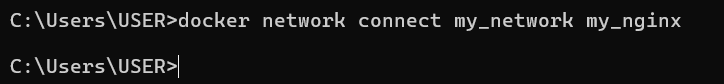
**25** docker run -d --network my\_network --name my\_container nginx

Chạy container my\_container trong network my\_network



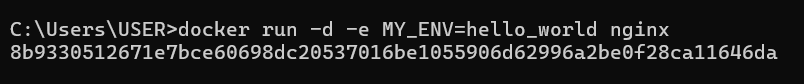
**26** docker network connect my\_network my\_nginx

Kết nối container my\_nginx vào network my\_network



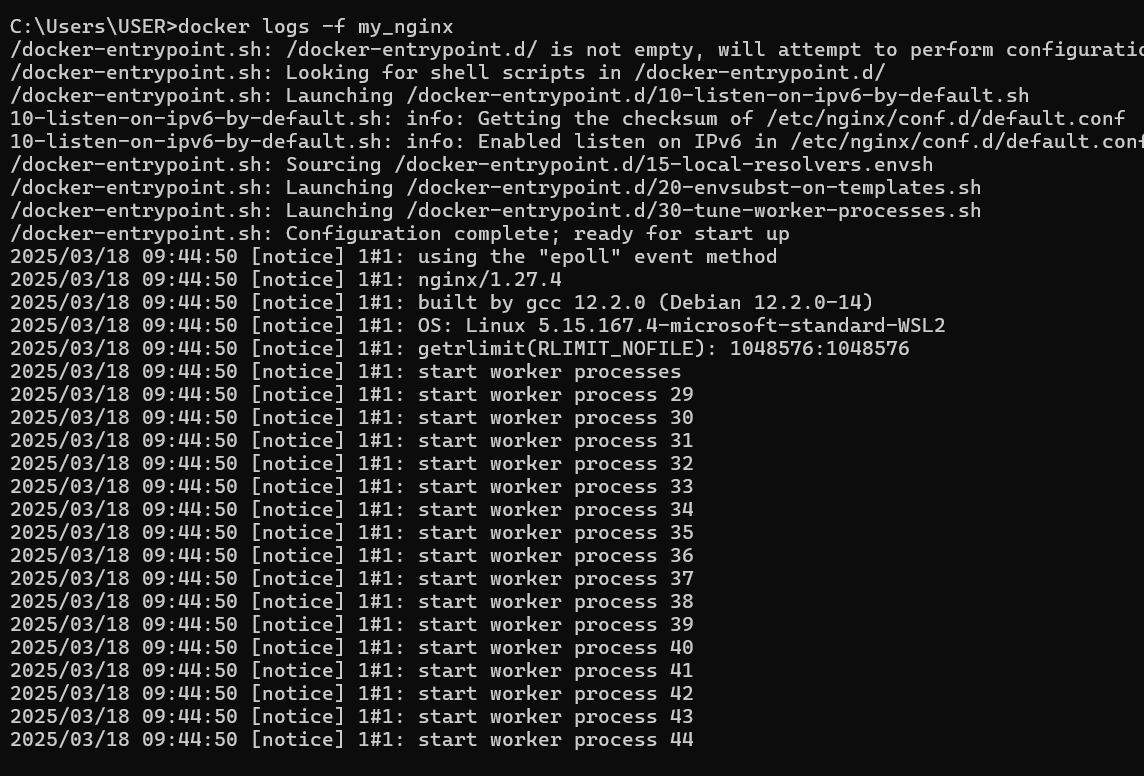
**27** docker run -d -e MY\_ENV=hello\_world nginx

Chạy container với biến môi trường MY\_ENV=hello\_world

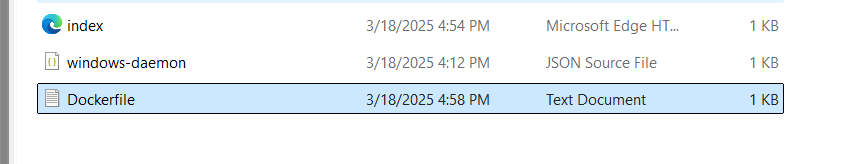


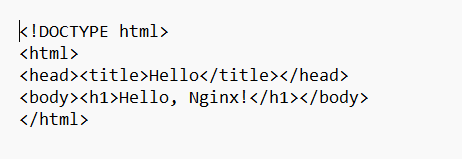
**28** docker logs -f my\_nginx

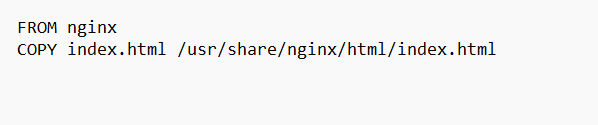
Theo dõi log của container my\_nginx theo thời gian thực



**29** FROM nginx COPY index.html /usr/share/nginx/html/index.html

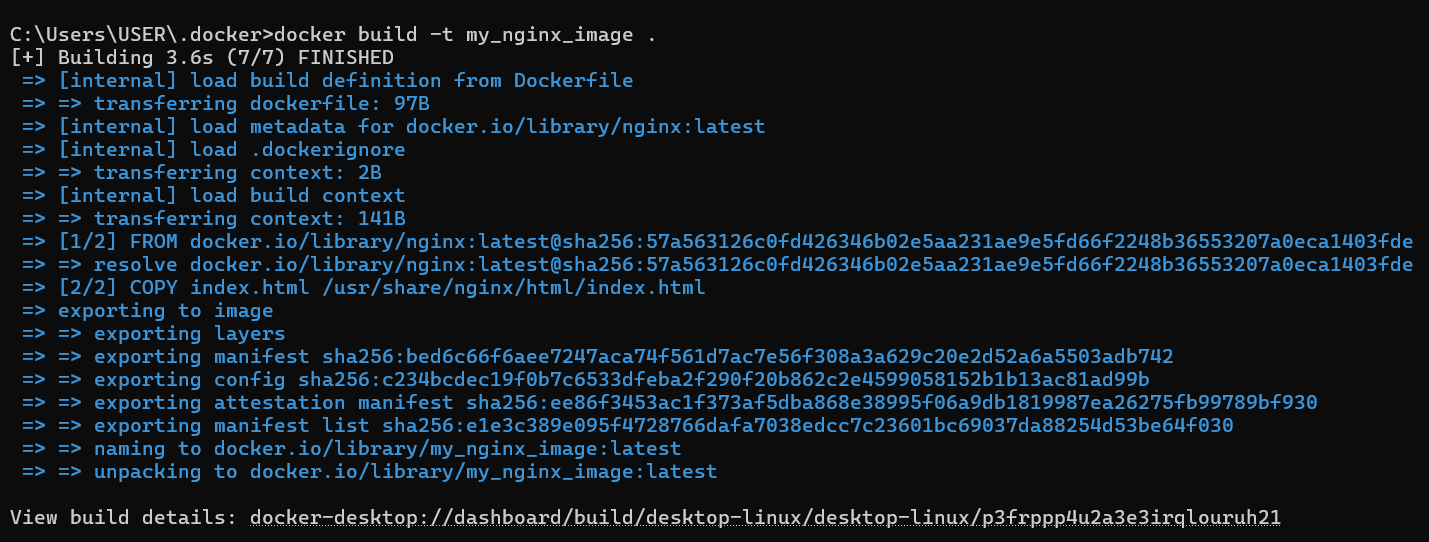






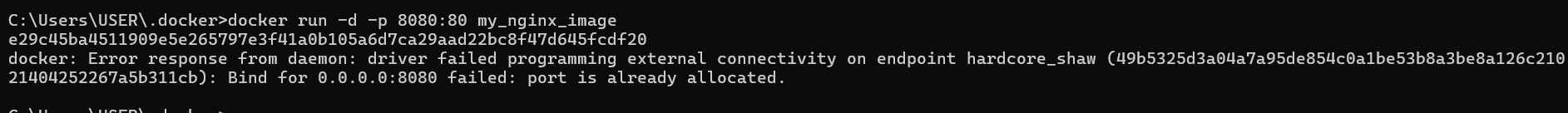
**30** docker build -t my\_nginx\_image .

Build image từ file Dockerfile với tên my\_nginx\_image.



**31** docker run -d -p 8080:80 my\_nginx\_image

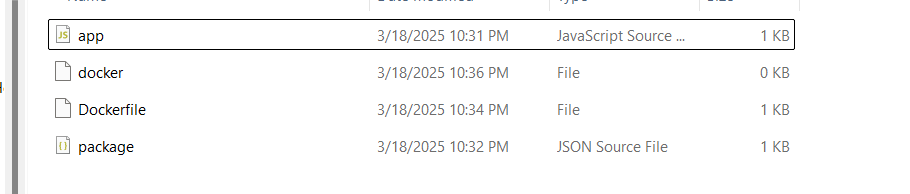
Chạy container từ image my\_nginx\_image, map cổng 8080 (host) với 80 (container).

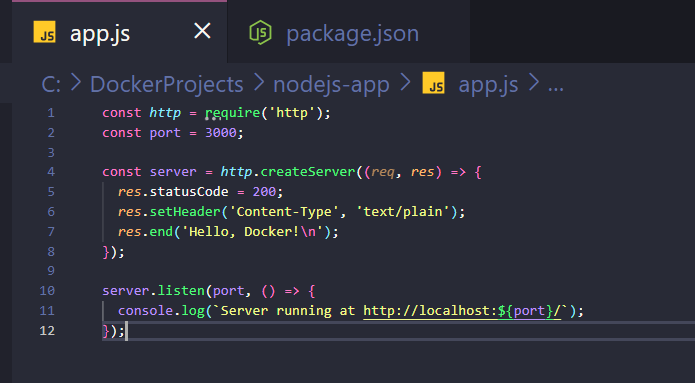


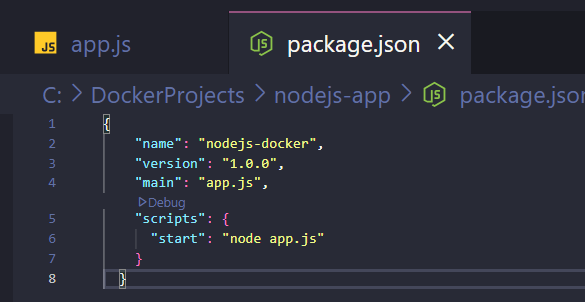
**Phần 2: Thao tác với Dockerfile**

**Bài 1:** Tạo Dockerfile chạy một ứng dụng Node.js đơn giản Yêu cầu: Viết Dockerfile để chạy một ứng dụng Node.js hiển thị "Hello, Docker!" trên cổng 3000. Sử dụng node:18 làm base image.

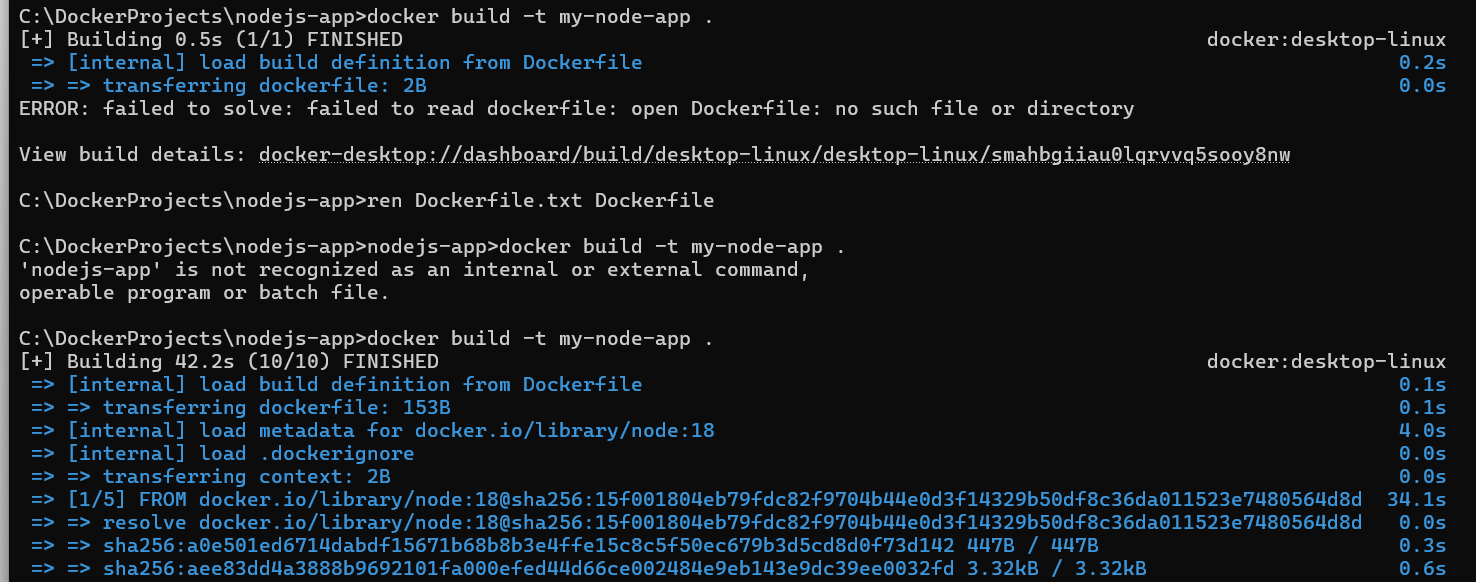
-Tạo file và mã nguồn

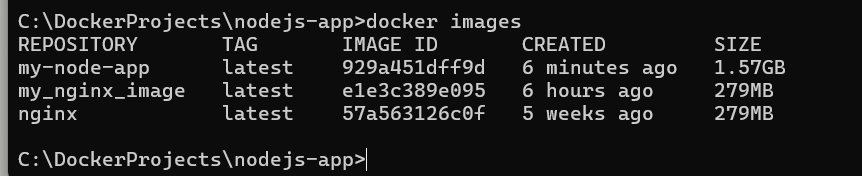






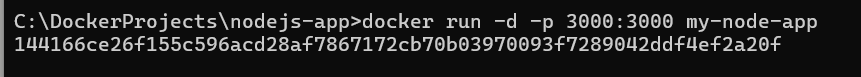
- Build image



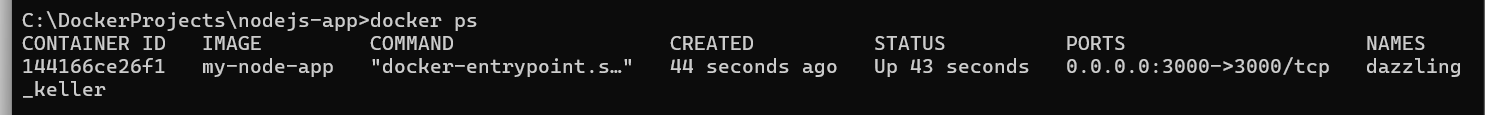


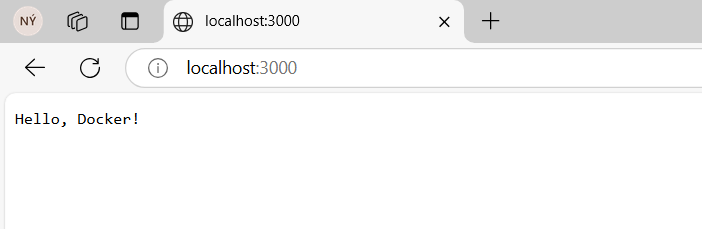
- Chạy container

docker run -d -p 3000:3000 my-node-app



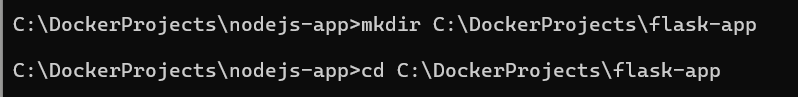
docker ps

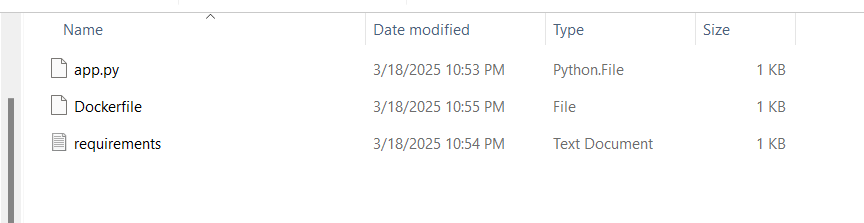


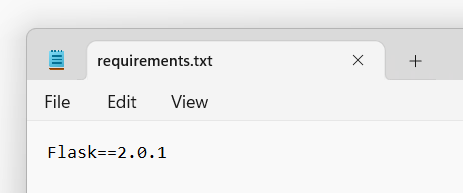


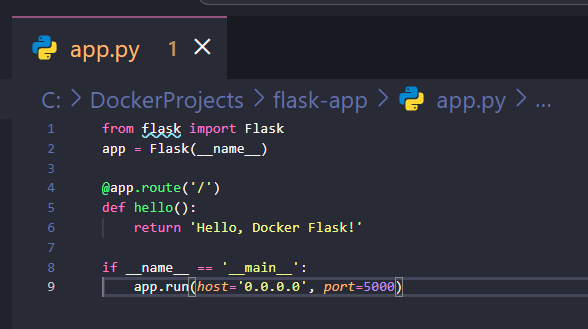
**Bài 2:** Tạo Dockerfile chạy một ứng dụng Python Flask Yêu cầu: Viết Dockerfile để chạy một ứng dụng Flask hiển thị "Hello, Docker Flask!" trên cổng 5000. Sử dụng python:3.9 làm base image.

- Tạo thư mục và mã nguồn



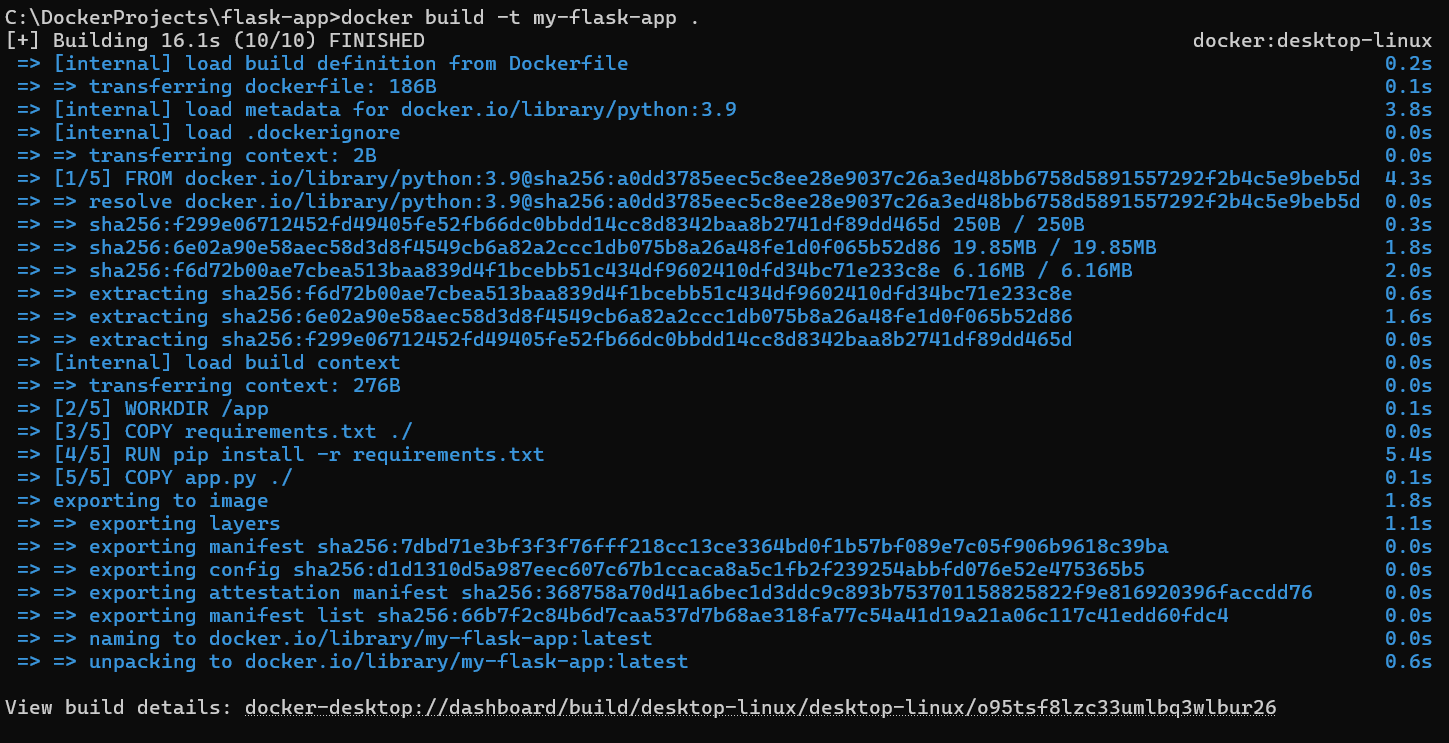




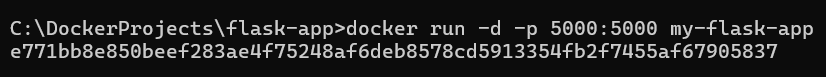


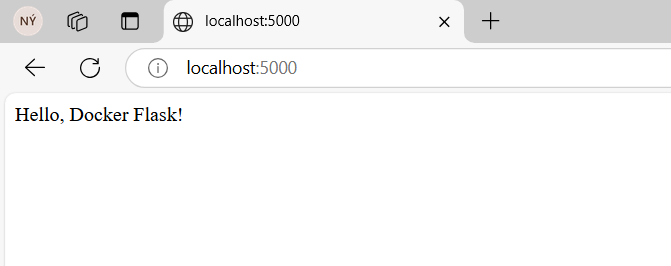
- Build và chạy

docker build -t my-flask-app .



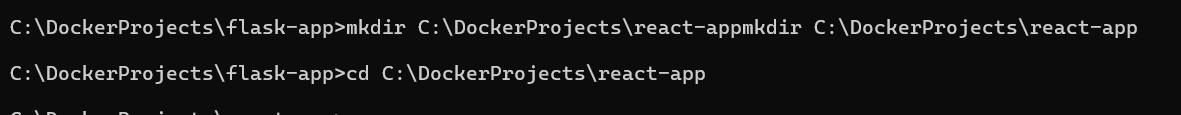
docker run -d -p 5000:5000 my-flask-app



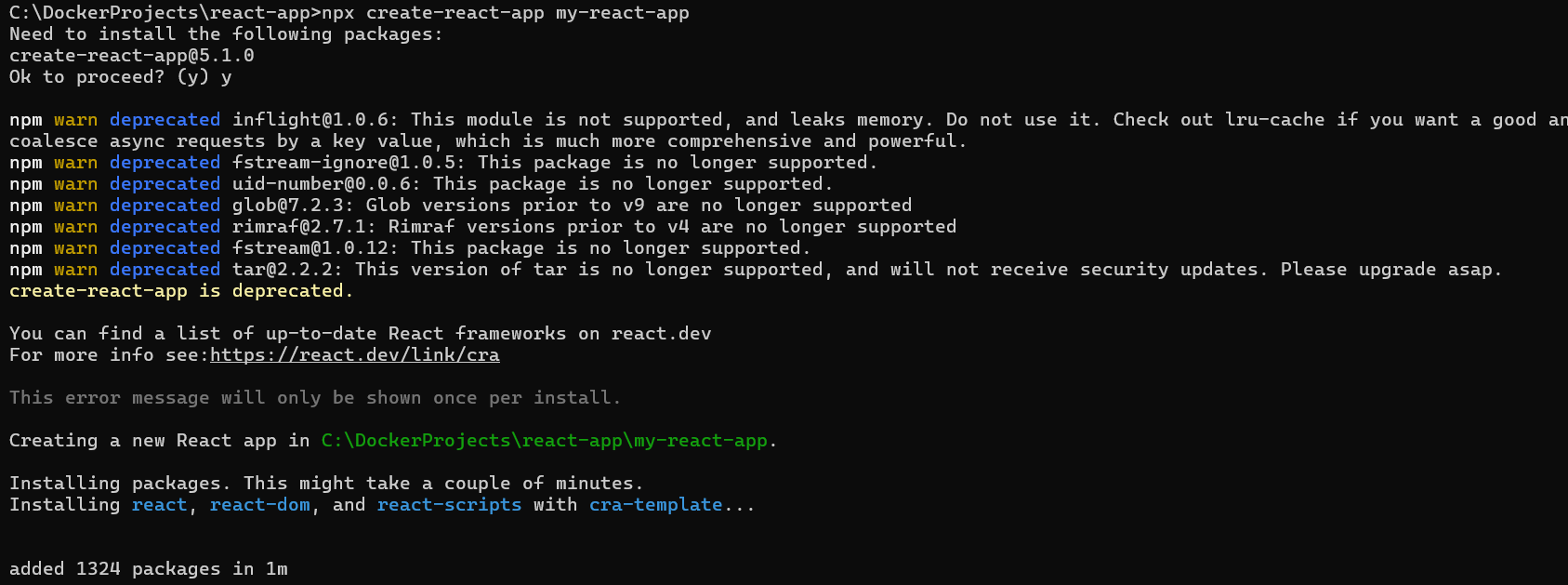


**Bài 3:** Tạo Dockerfile chạy một ứng dụng React Yêu cầu: Viết Dockerfile để build và chạy một ứng dụng React. Sử dụng node:18-alpine làm base image.

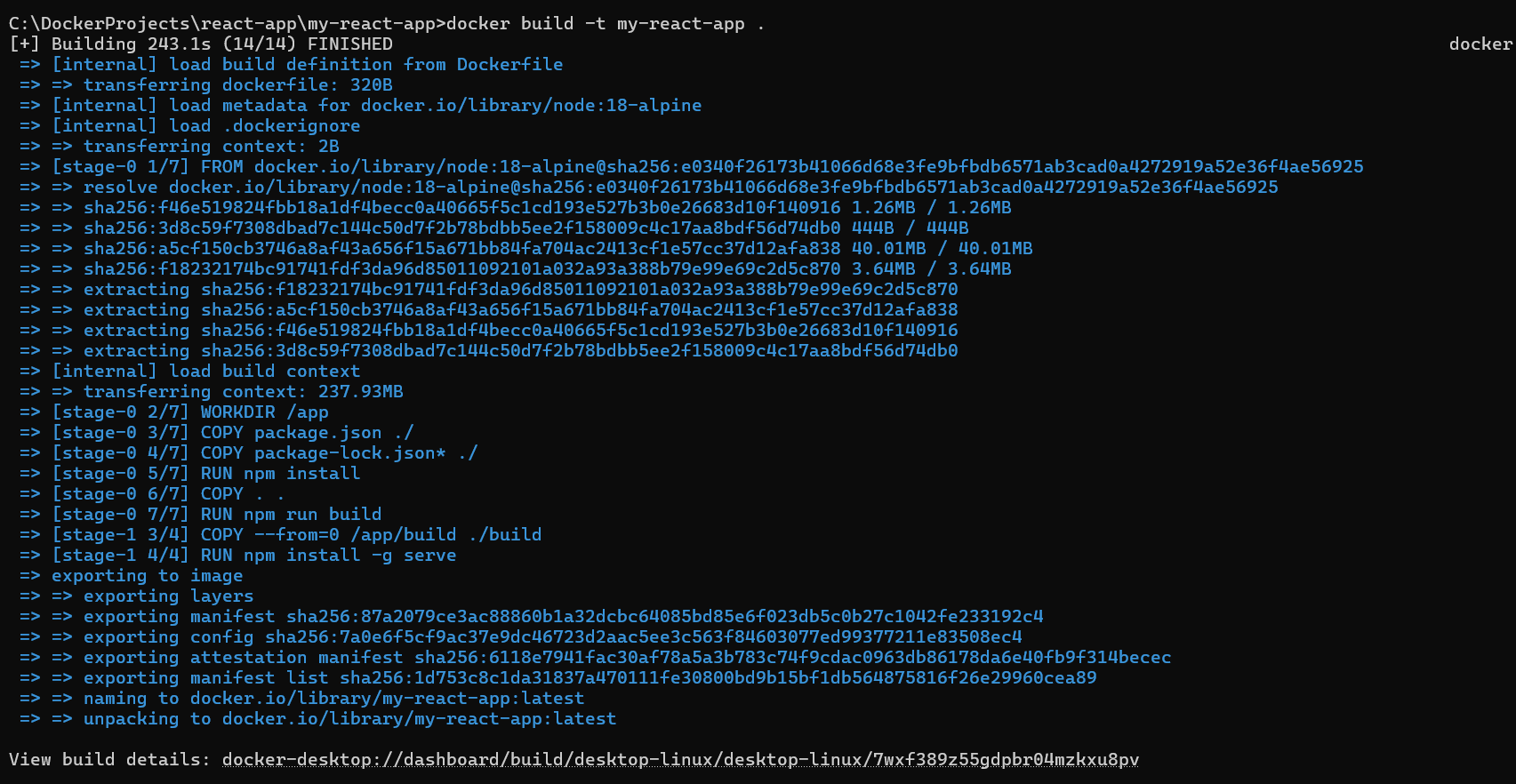
- Tạo thư mục và mã nguồn



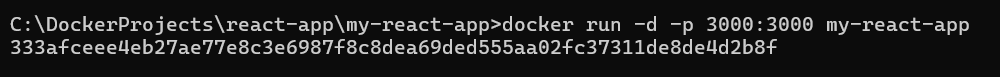
npx create-react-app my-react-app

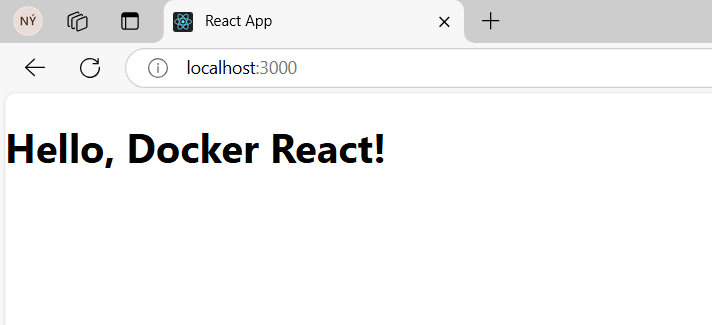


docker build -t my-react-app .



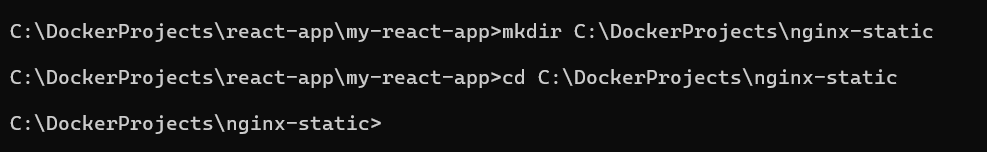
docker run -d -p 3000:3000 my-react-app

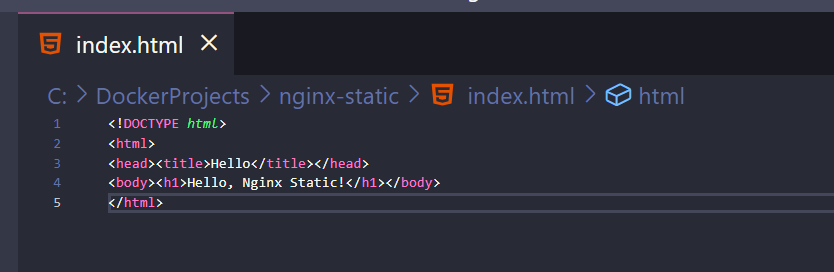




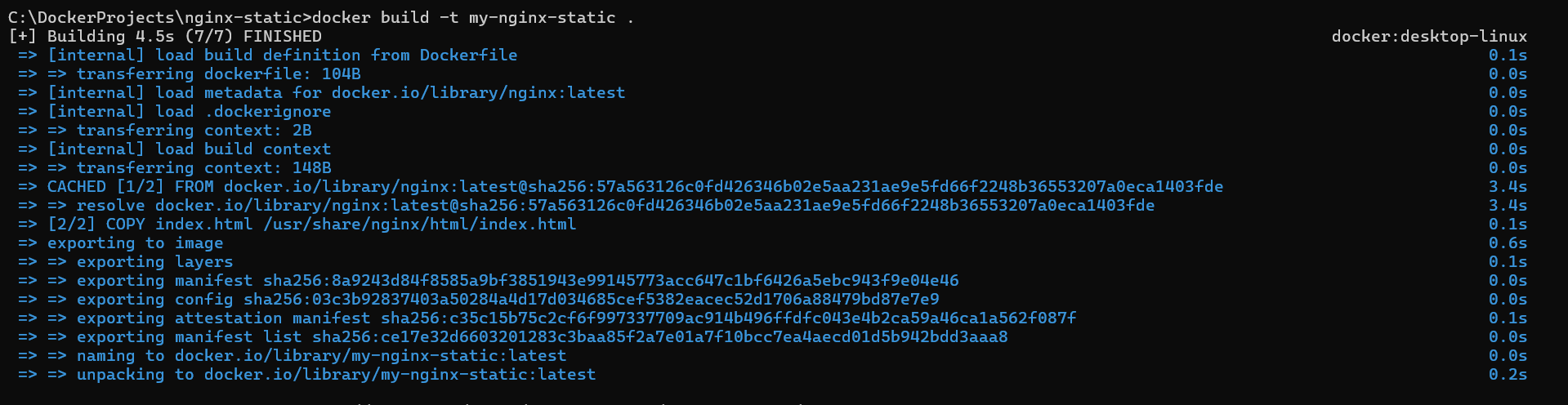
**Bài 4:** Tạo Dockerfile chạy một trang web tĩnh bằng Nginx Yêu cầu: Tạo một file index.html đơn giản và sử dụng nginx:latest để phục vụ trang web.

- Tạo thư mục và mã nguồn

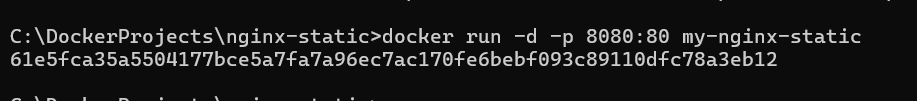




docker build -t my-nginx-static .



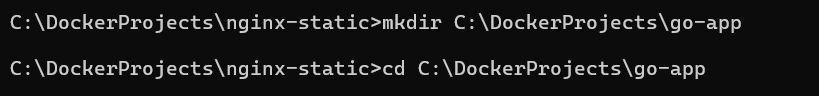
docker run -d -p 8080:80 my-nginx-static

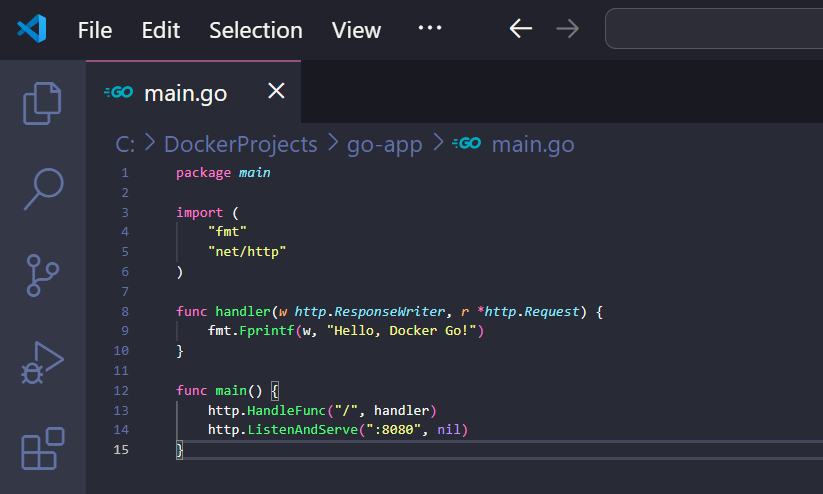


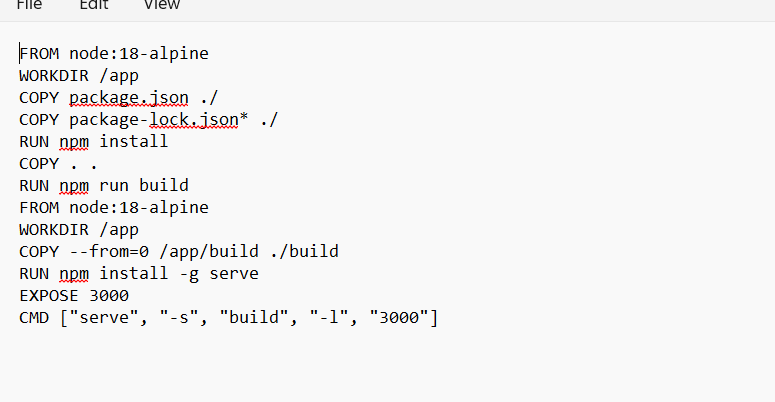


**Bài 5:** Tạo Dockerfile cho ứng dụng Go Yêu cầu: Viết Dockerfile để build và chạy một ứng dụng Go đơn giản.

-Tạo thư mục và mã nguồn

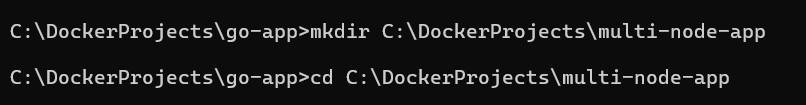


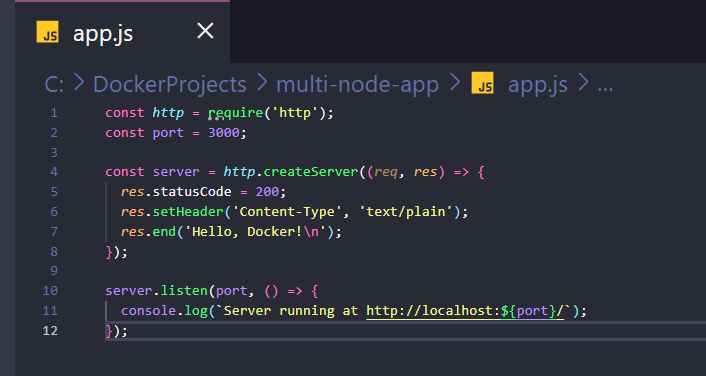




**Bài 6:** Sử dụng Multi-stage Build trong Dockerfile Viết Dockerfile để build một ứng dụng Node.js với hai stage: Stage 1: Dùng node:18 để build code. Stage 2: Dùng node:18-alpine để chạy ứng dụng đã build.

- Tạo thư mục và mã nguồn



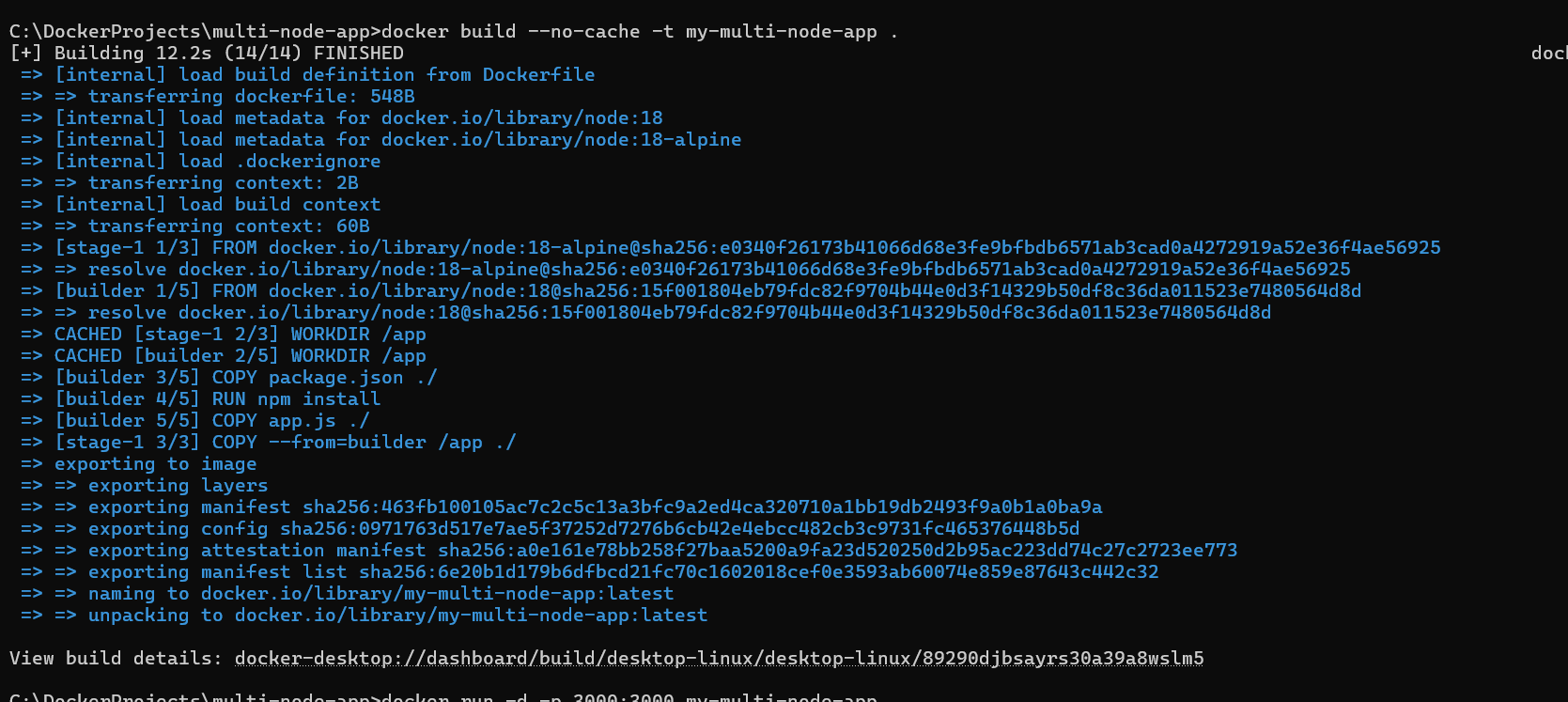


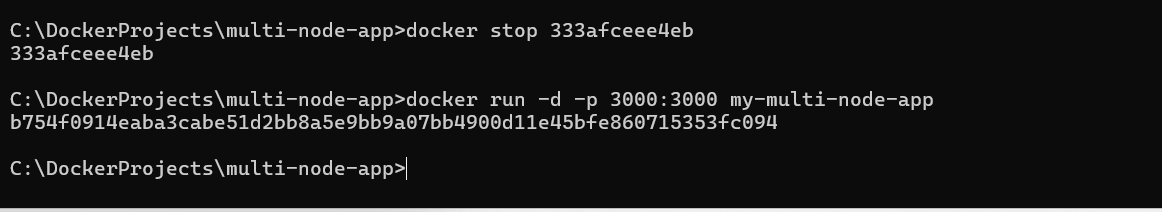


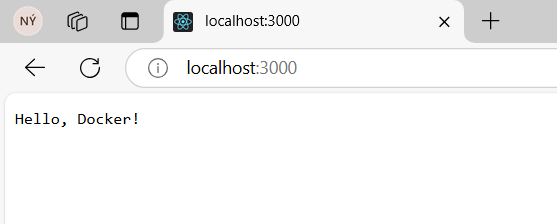
npm init -y



docker build --no-cache -t my-multi-node-app .

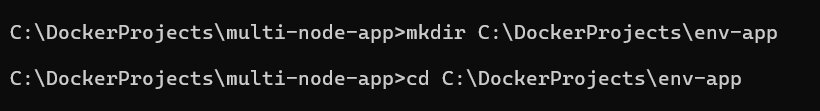


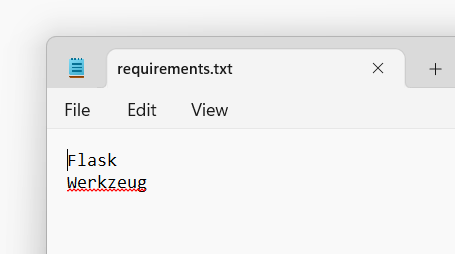


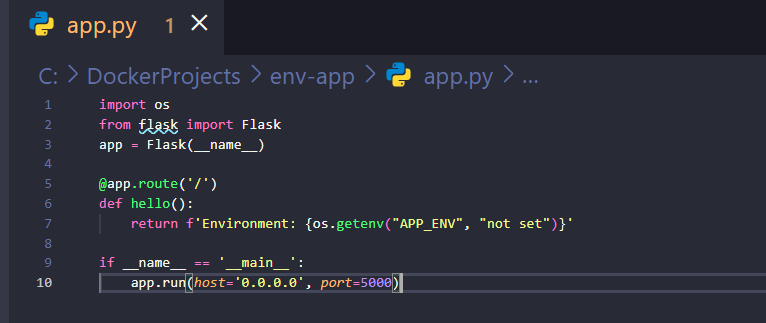


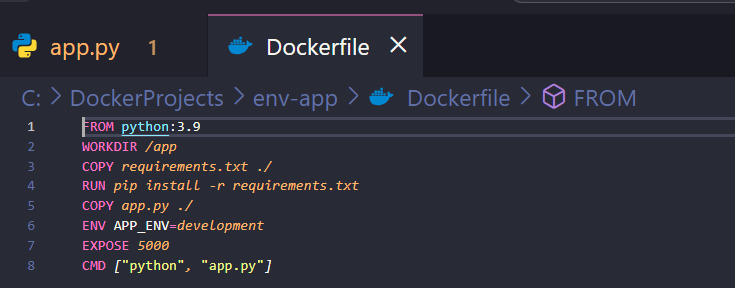
**Bài 7:** Sử dụng biến môi trường trong Dockerfile Yêu cầu: Viết Dockerfile cho ứng dụng Python đọc biến môi trường APP\_ENV và in ra màn hình. Sử dụng ENV APP\_ENV=development trong Dockerfile.

-Tạo thư mục và mã nguồn

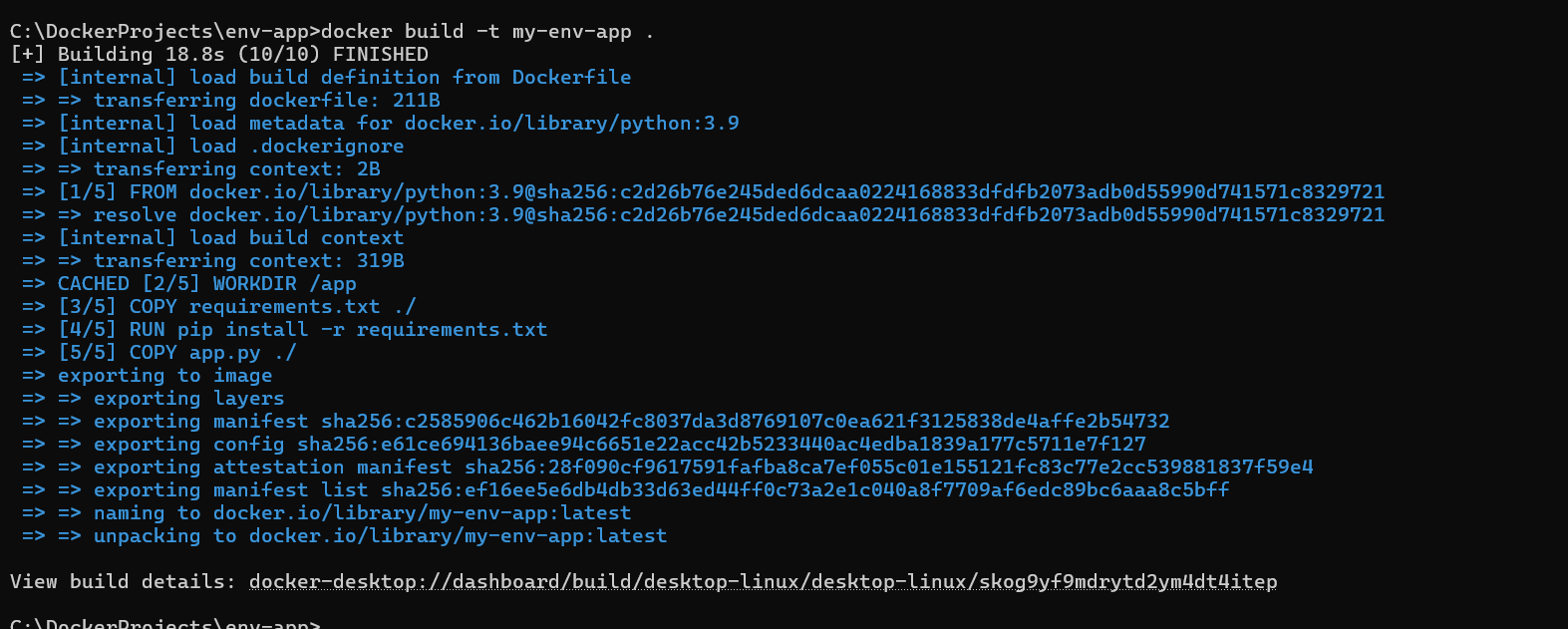




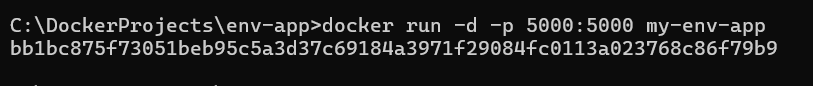


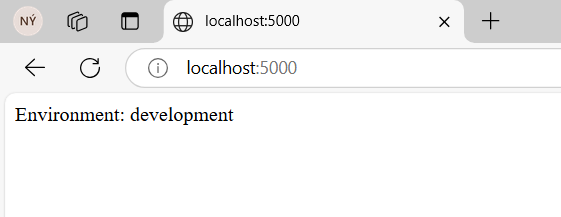


docker build -t my-env-app .



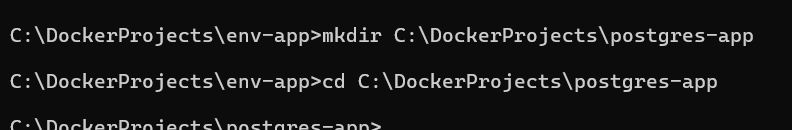
docker run -d -p 5000:5000 my-env-app

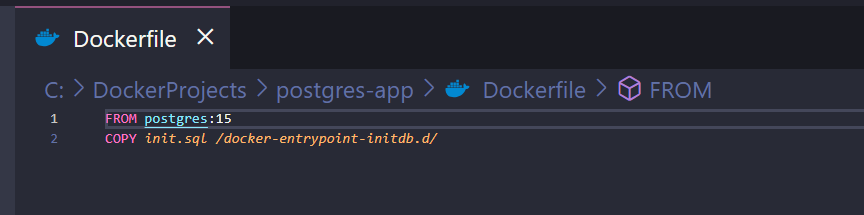




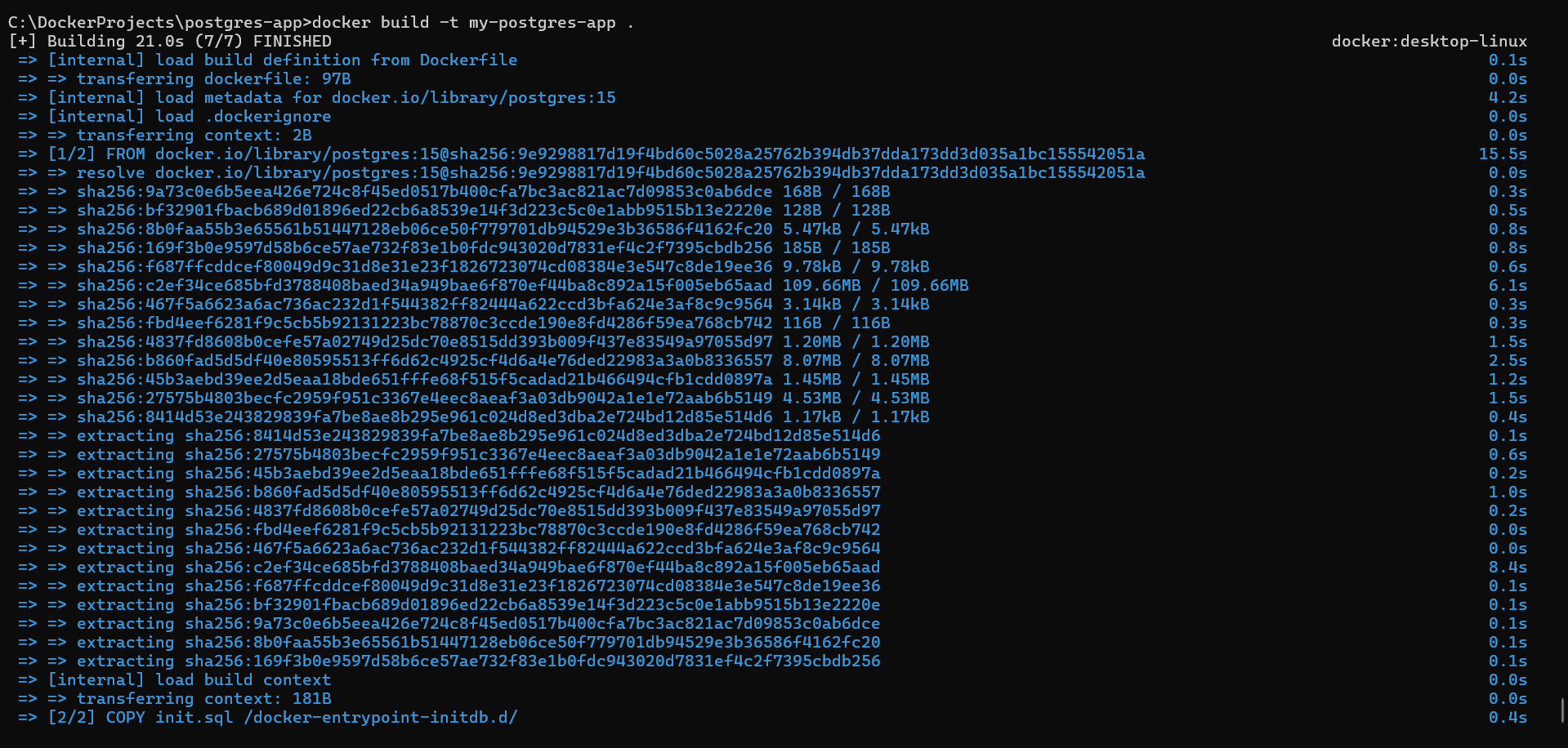
**Bài 8:** Tạo Dockerfile cho PostgreSQL tùy chỉnh Yêu cầu: Viết Dockerfile để chạy PostgreSQL (postgres:15). Thêm file SQL để tự động tạo database khi container chạy lần đầu tiên.

-Tạo thư mục và mã nguồn

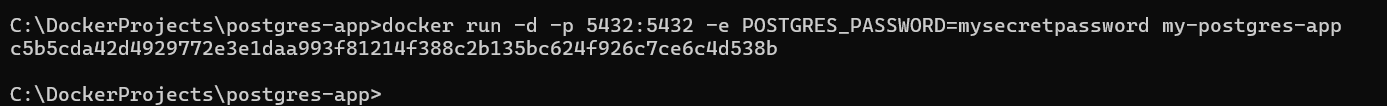




docker build -t my-postgres-app .

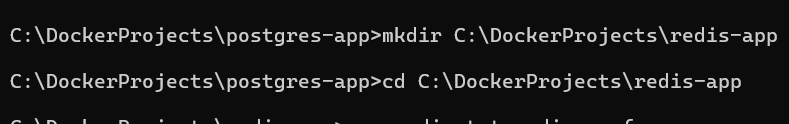


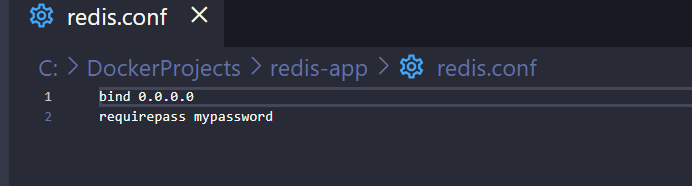
docker run -d -p 5432:5432 -e POSTGRES\_PASSWORD=mysecretpassword my-postgres-app

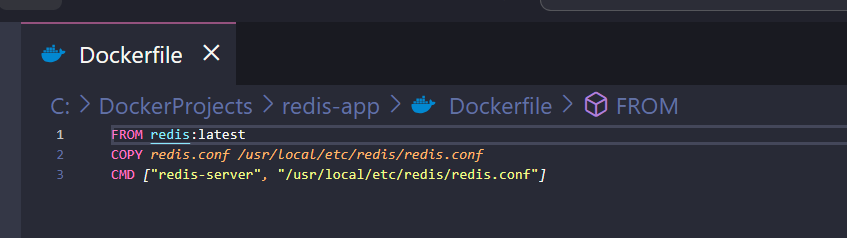


**Bài 9:** Tạo Dockerfile chạy Redis với cấu hình tùy chỉnh Yêu cầu: Viết Dockerfile sử dụng redis:latest. Thêm file redis.conf vào container.

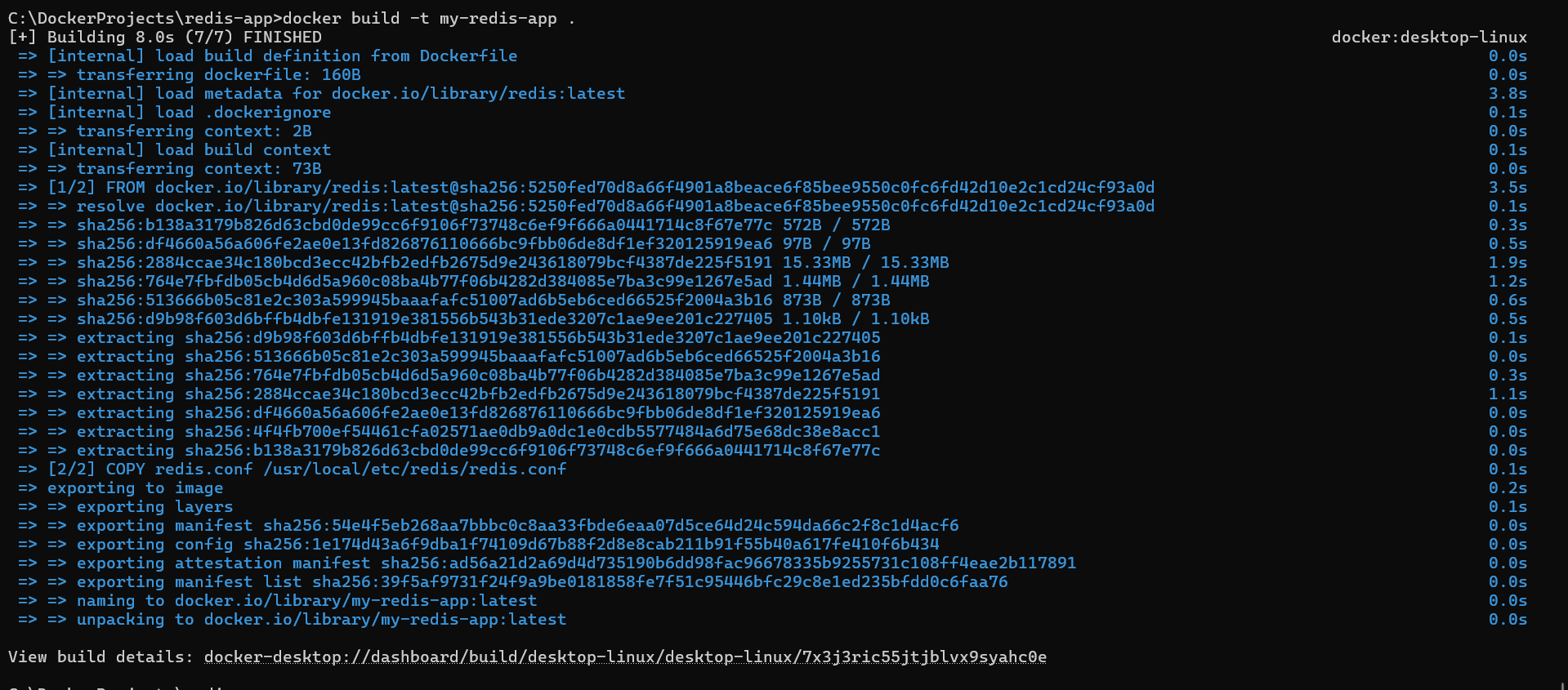
-Tạo thư mục và mã nguồn



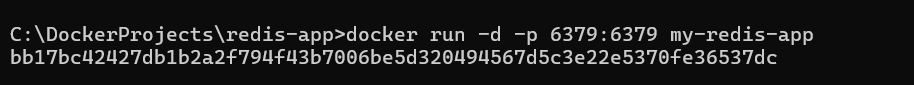




docker build -t my-redis-app .

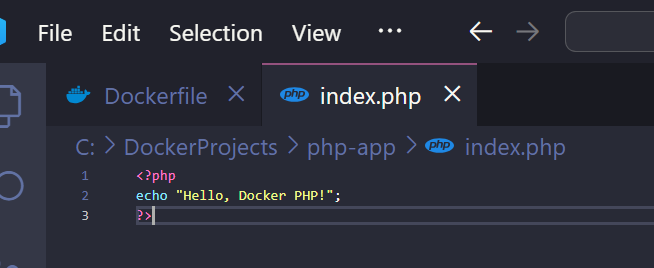


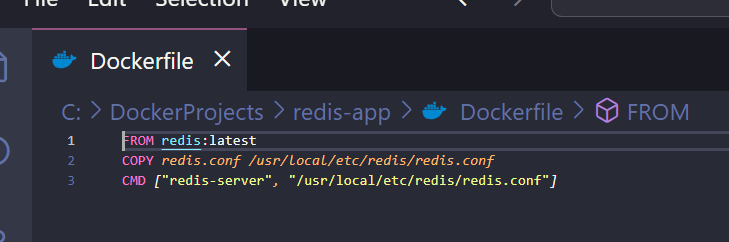
docker run -d -p 6379:6379 my-redis-app



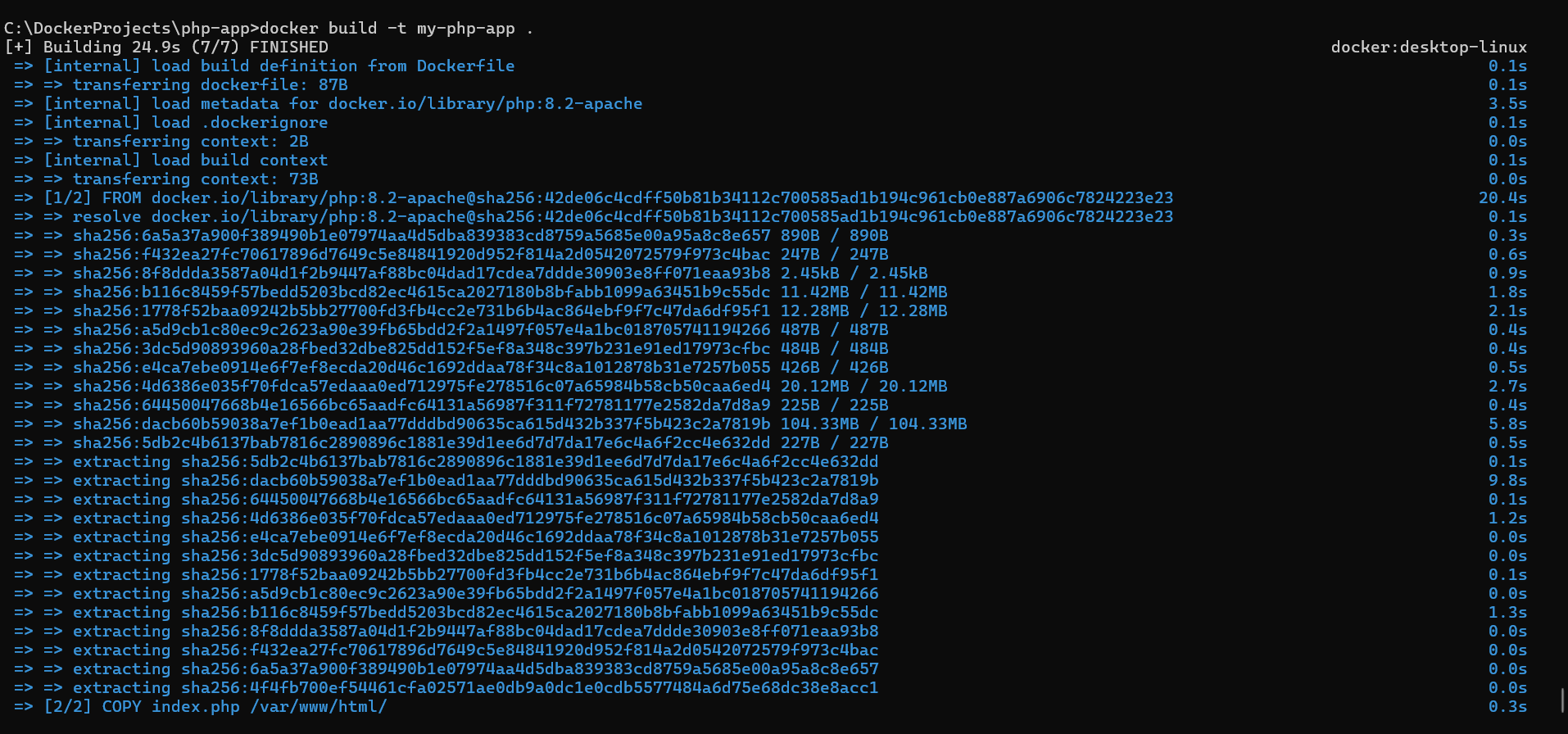
**Bài 10:** Chạy ứng dụng PHP với Apache Yêu cầu: Viết Dockerfile để chạy một ứng dụng PHP đơn giản (php:8.2-apache). Mount mã nguồn từ máy host vào container.

-Tạo thư mục và mã nguồn





docker build -t my-php-app .



docker run -d -p 80:80 -v C:\DockerProjects\php-app:/var/www/html my-php-app

