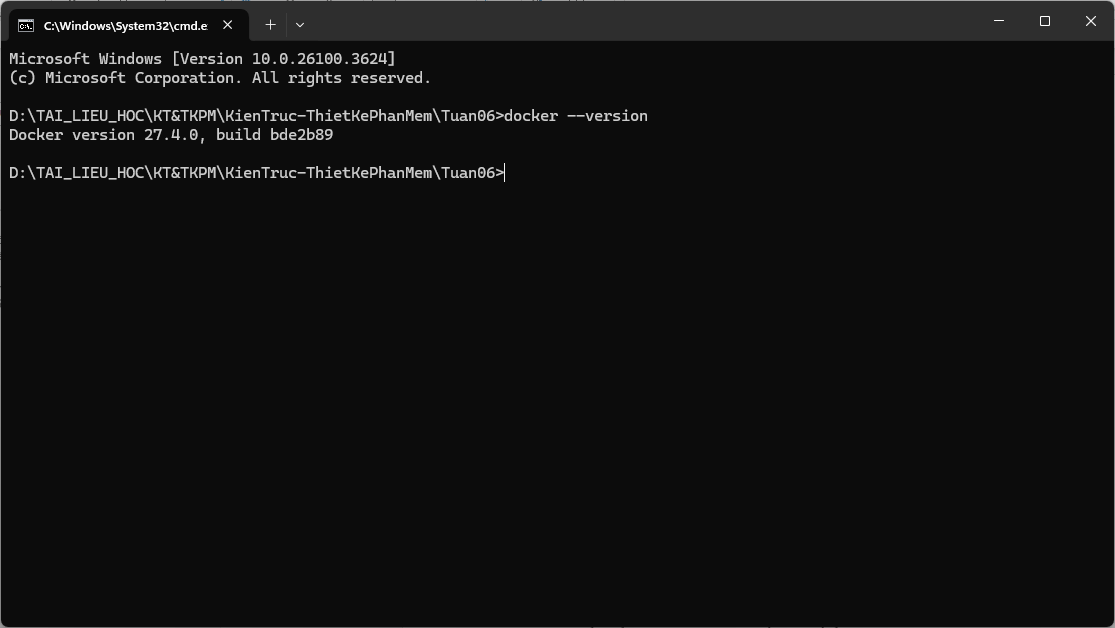
**Phần 1: Các lệnh cơ bản thao tác với Docker**

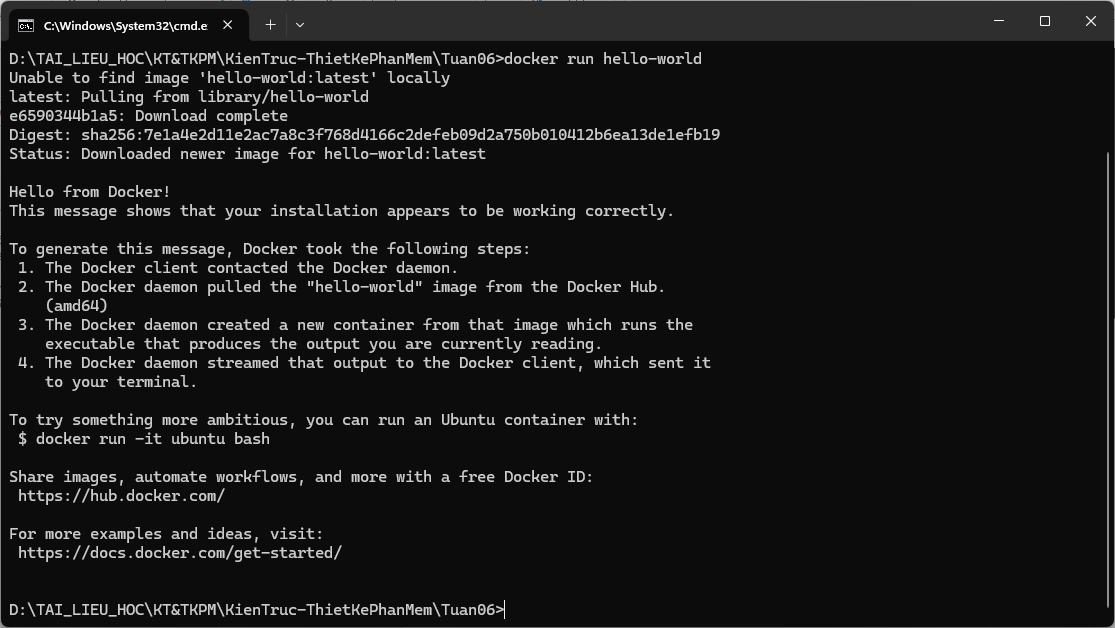
1. docker –version

Hiển thị phiên bản Docker hiện đang cài đặt trên hệ thống.



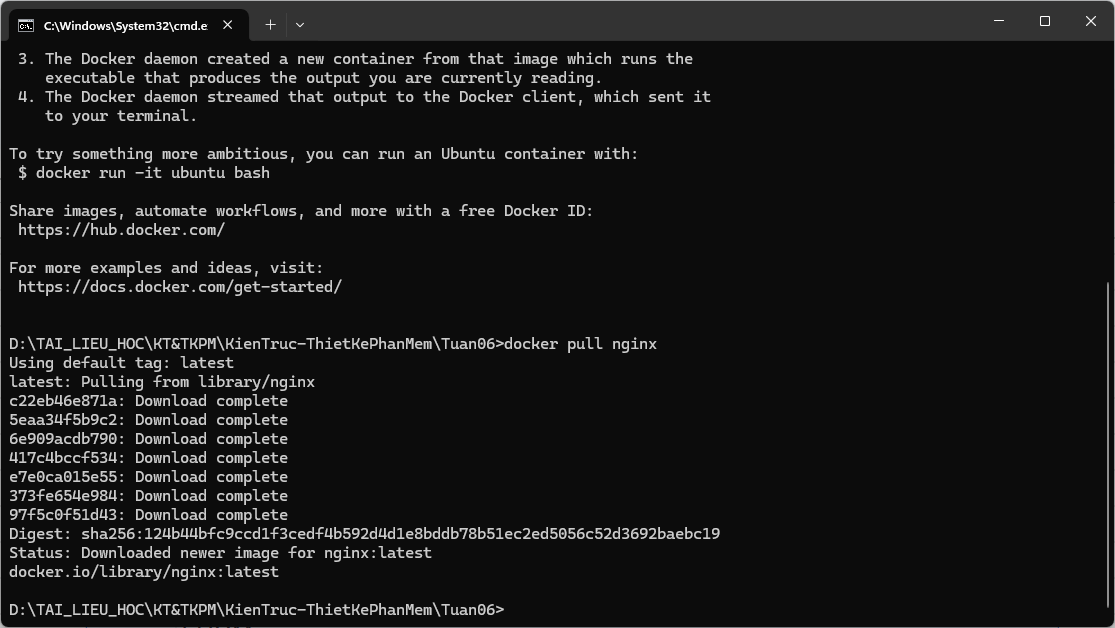
1. docker run hello-world

Tải và chạy một container hello-world, hiển thị thông điệp kiểm tra xem Docker có hoạt động đúng không.



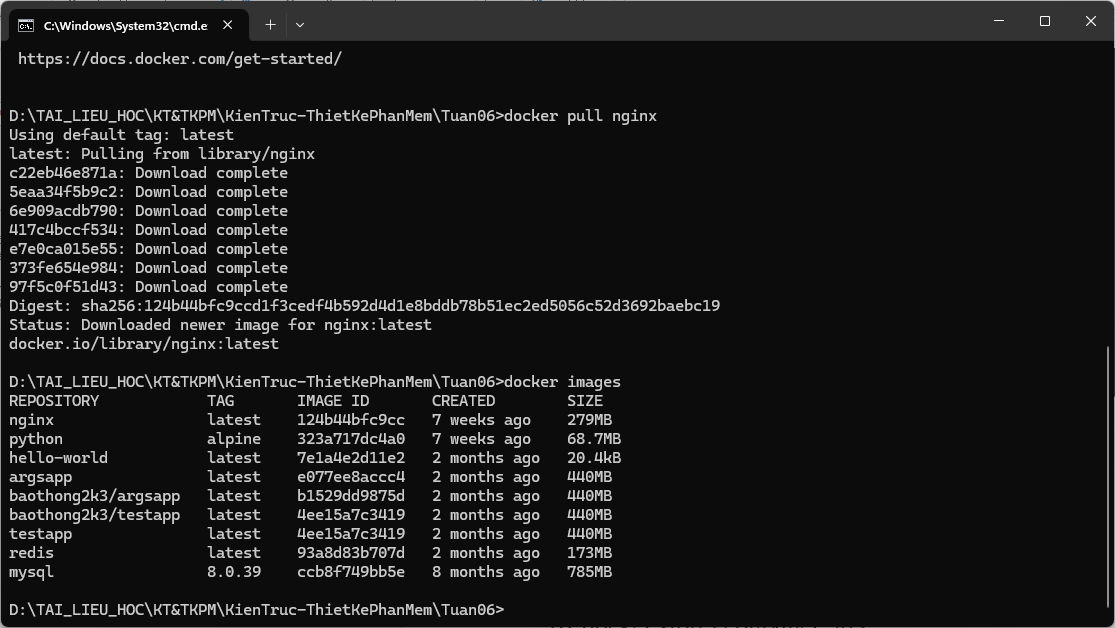
1. docker pull nginx

Tải image nginx mới nhất từ Docker Hub về máy.



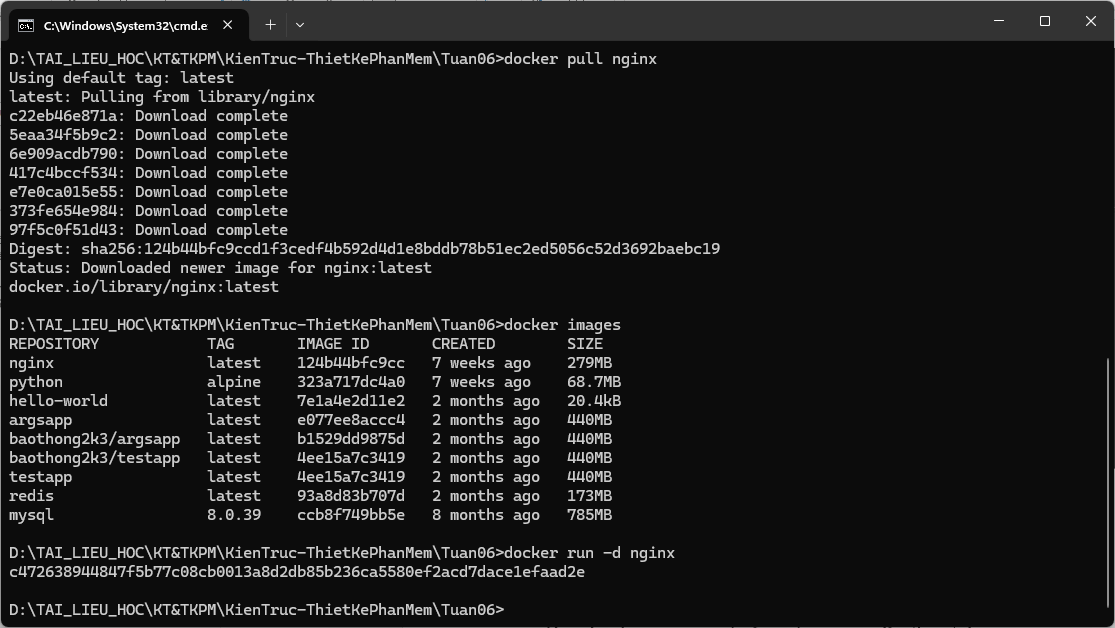
1. docker images

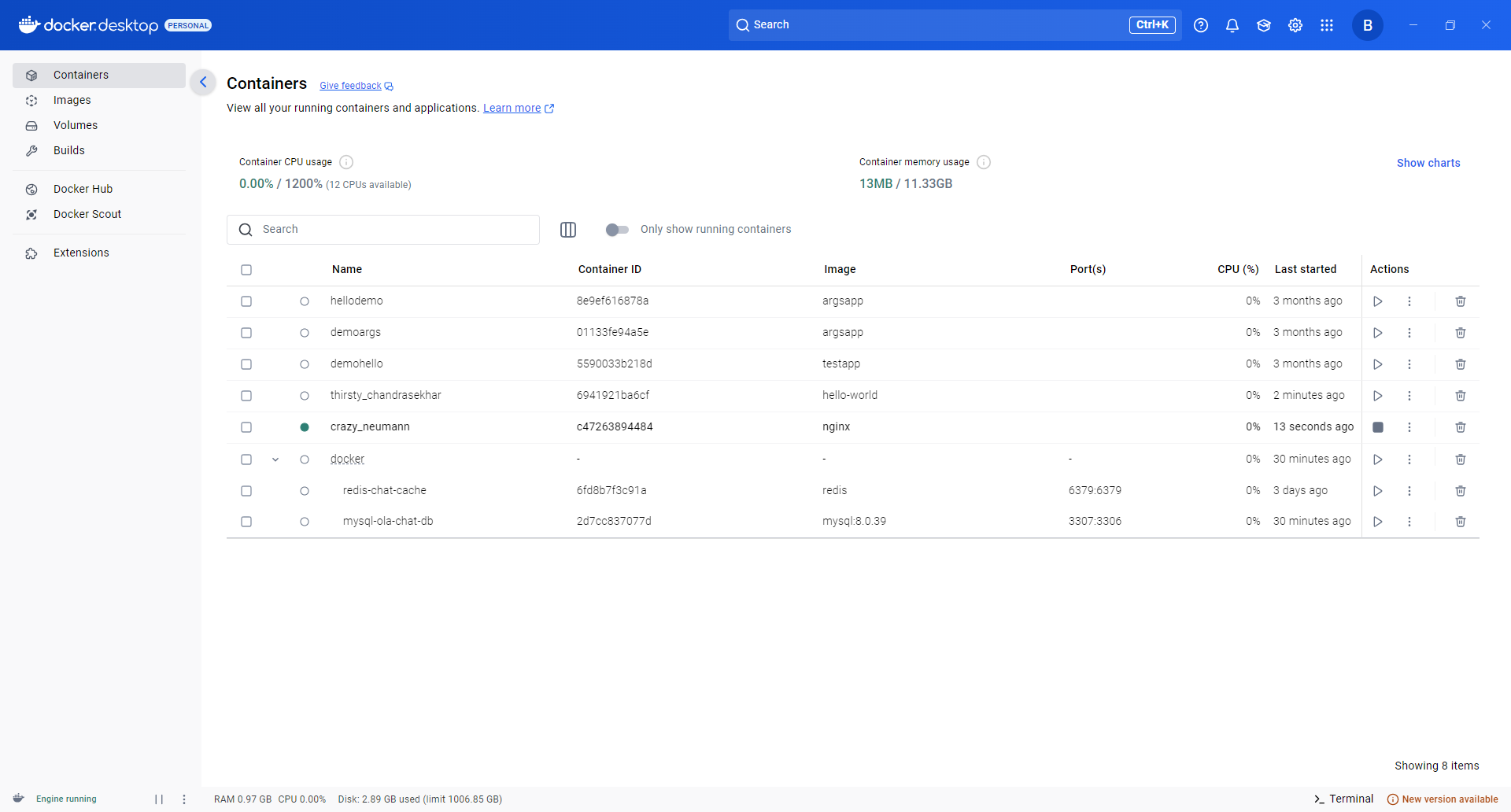
Hiển thị danh sách các image có sẵn trong máy.



1. docker run -d nginx

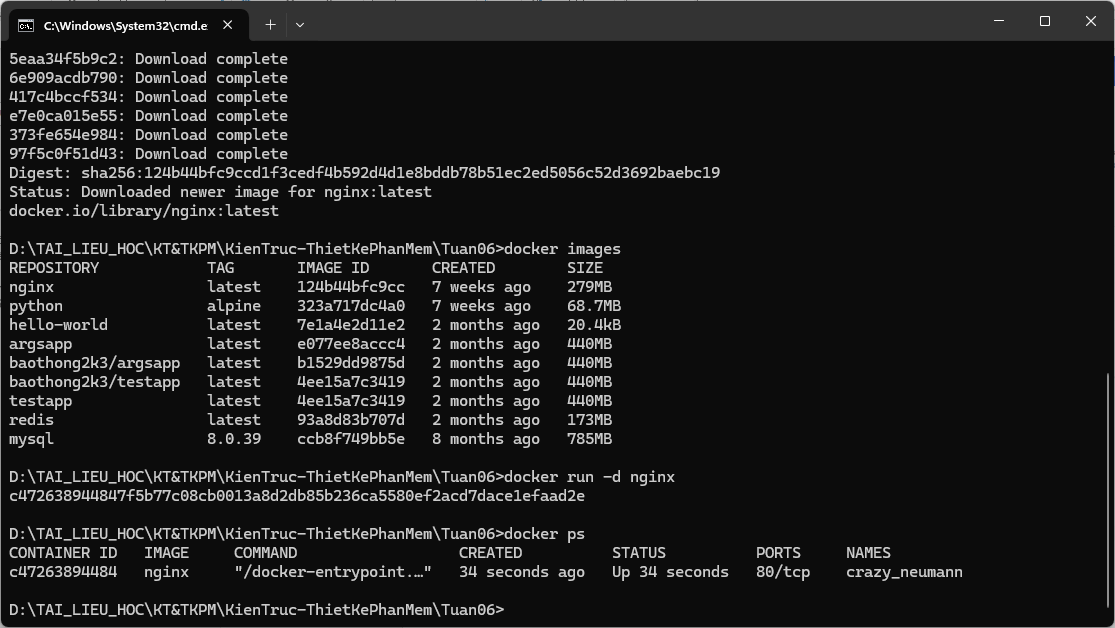
Chạy container từ image nginx ở chế độ detached (chạy nền).





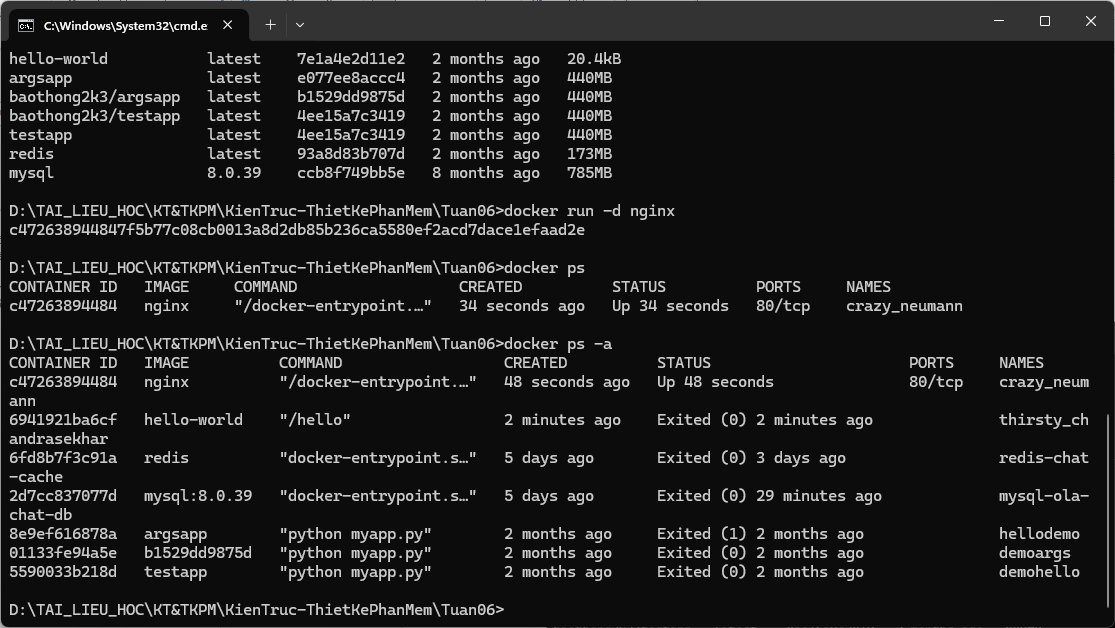
1. docker ps

Hiển thị danh sách các container đang chạy.



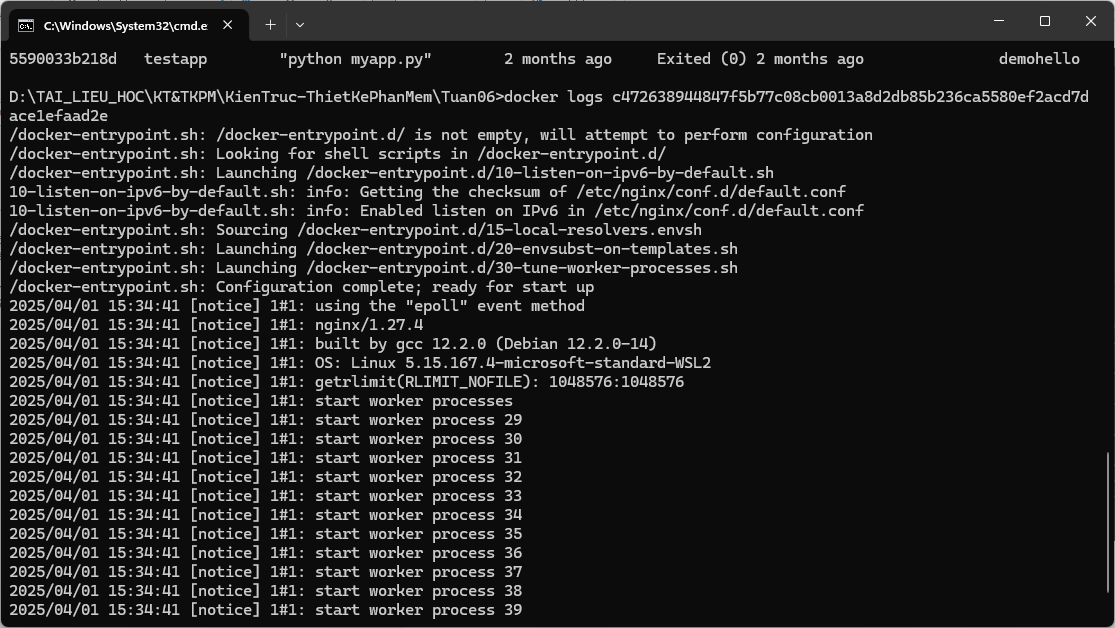
1. docker ps -a

Hiển thị tất cả container, kể cả container đã dừng.



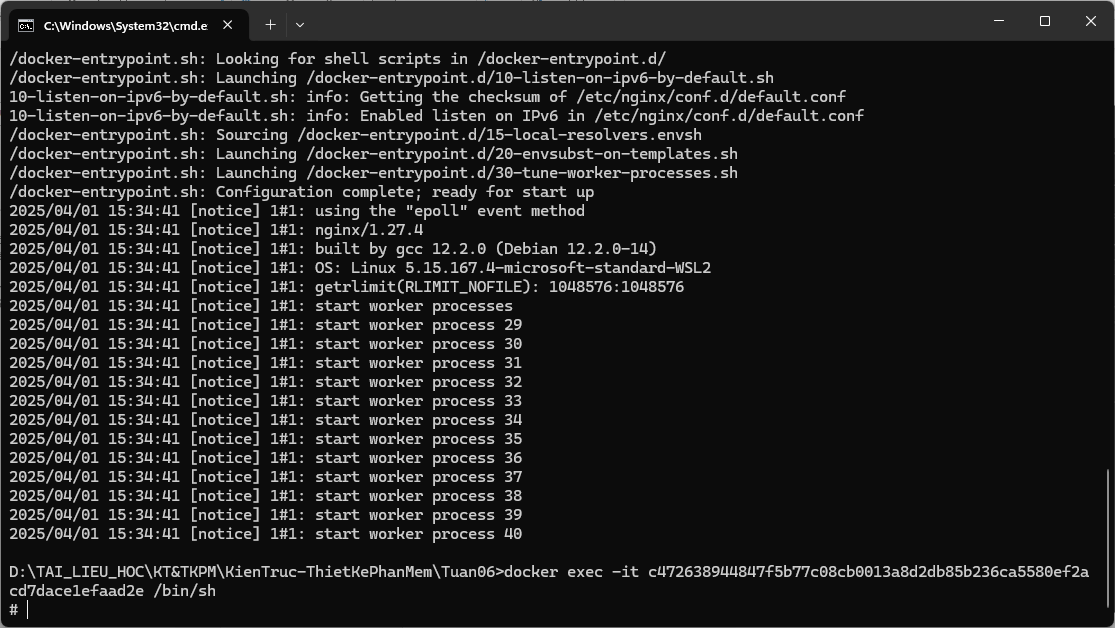
1. docker logs <container\_id>

Xem log của container có ID <container\_id>.



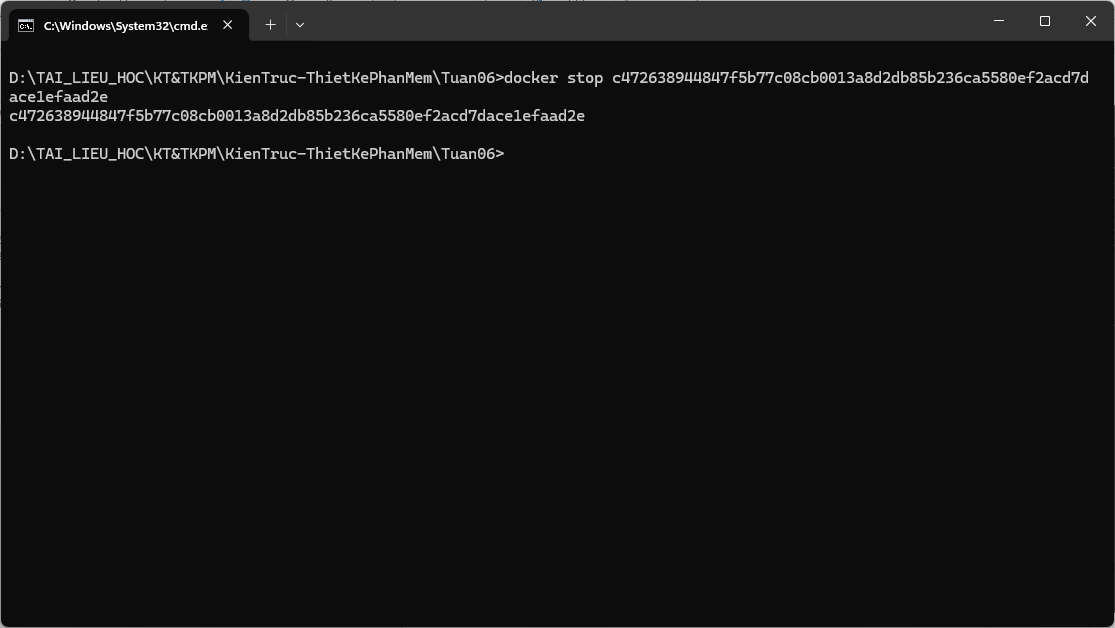
1. docker exec -it <container\_id> /bin/sh

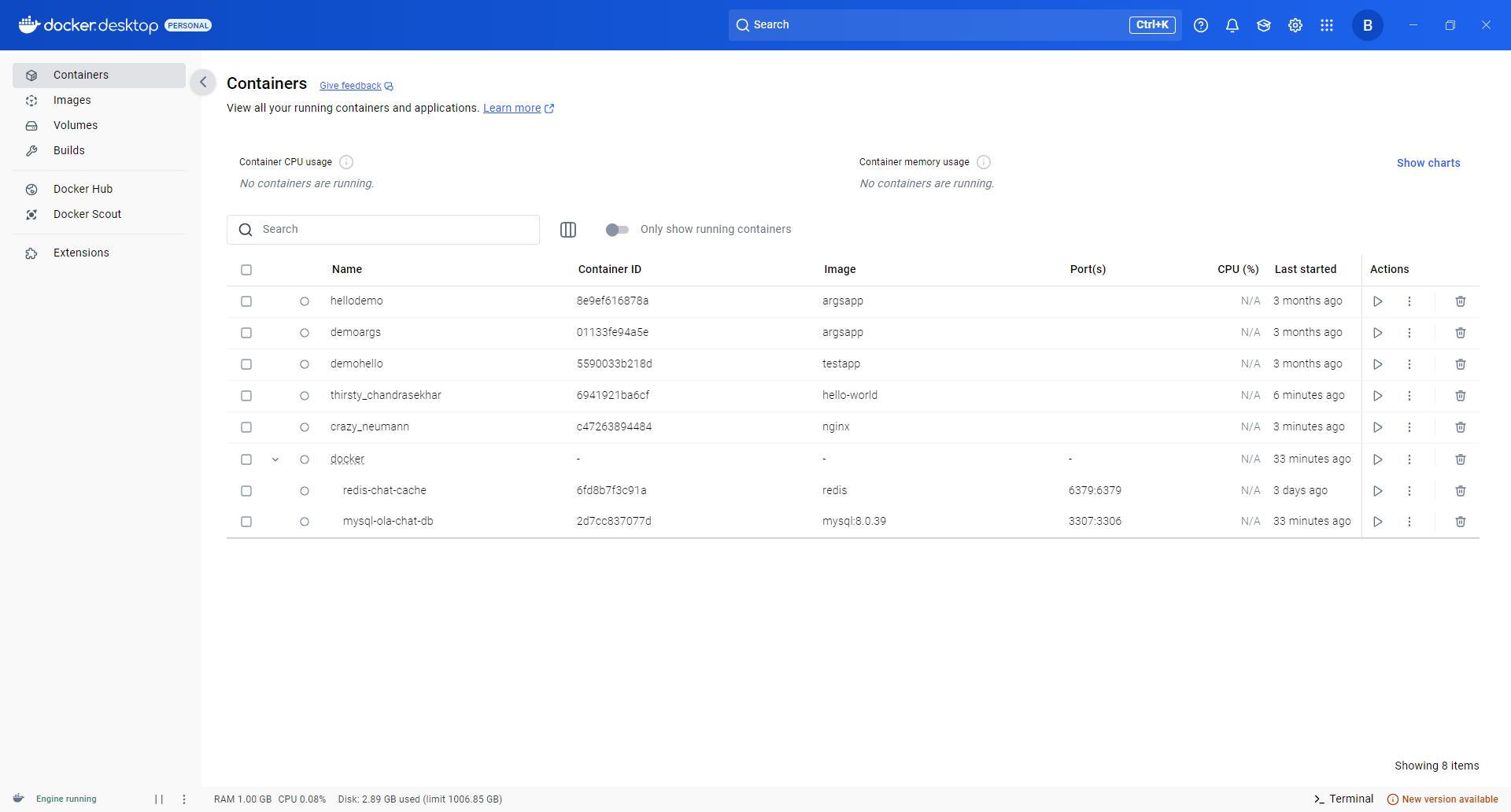
Mở một terminal (shell) trong container có ID <container\_id>.



1. docker stop <container\_id>

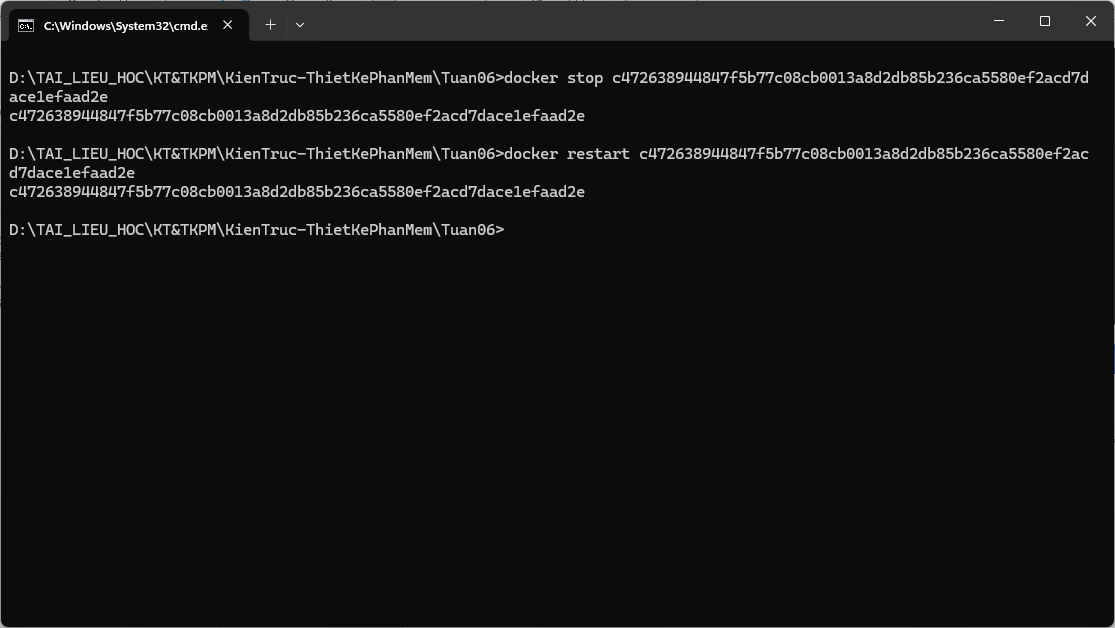
Dừng container đang chạy.

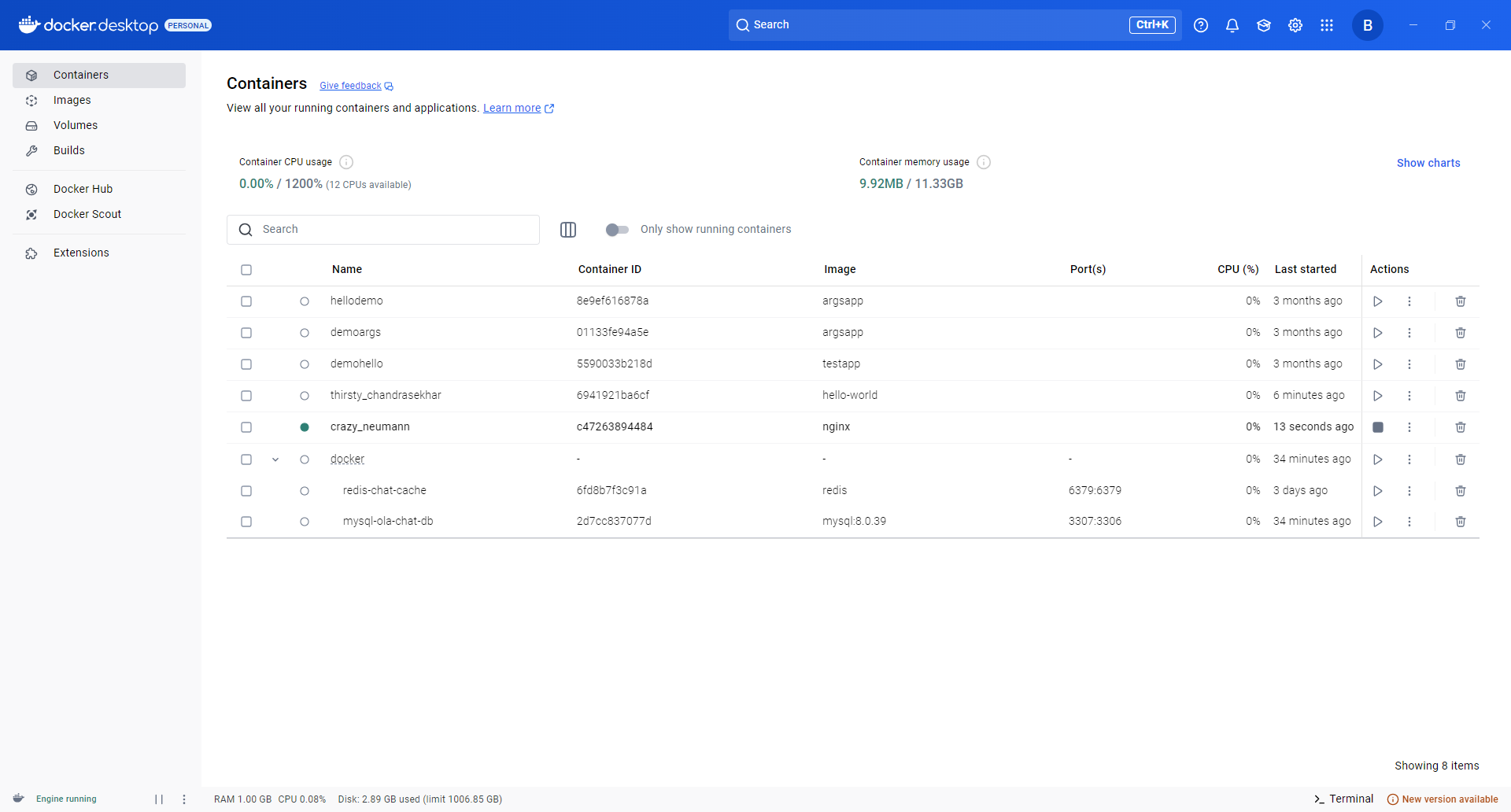




1. docker restart <container\_id>

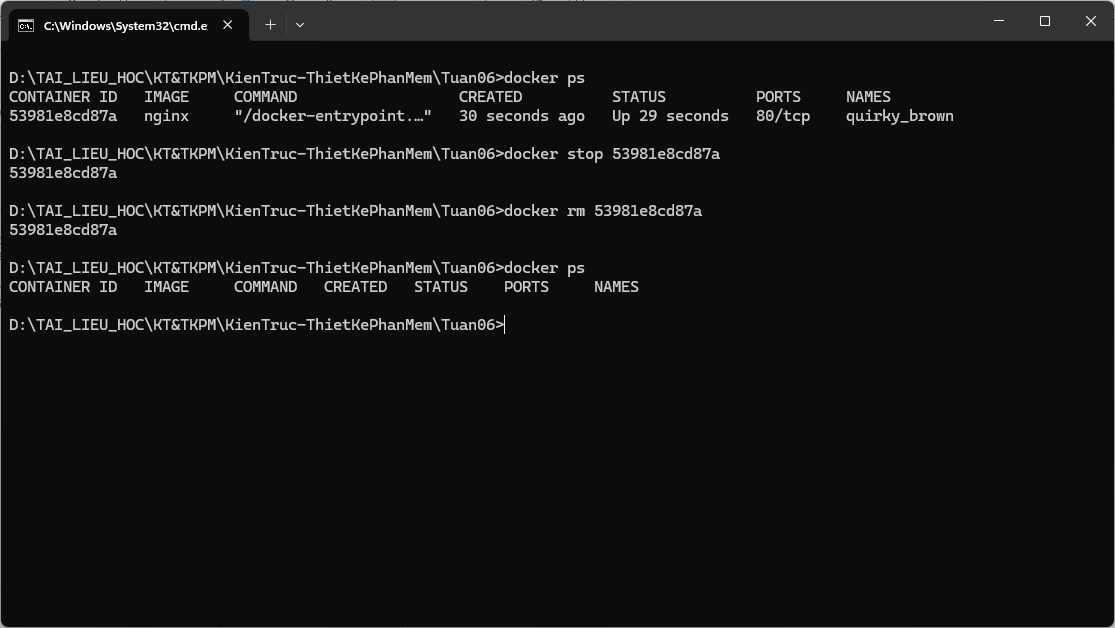
Khởi động lại container.





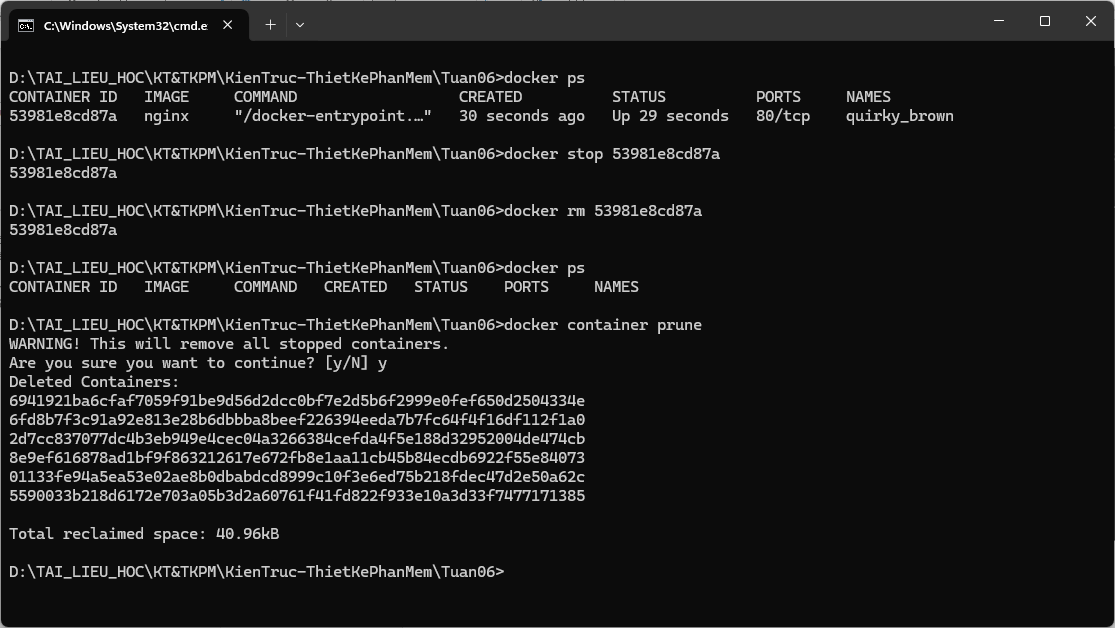
1. docker rm <container\_id>

Xóa container có ID <container\_id>.



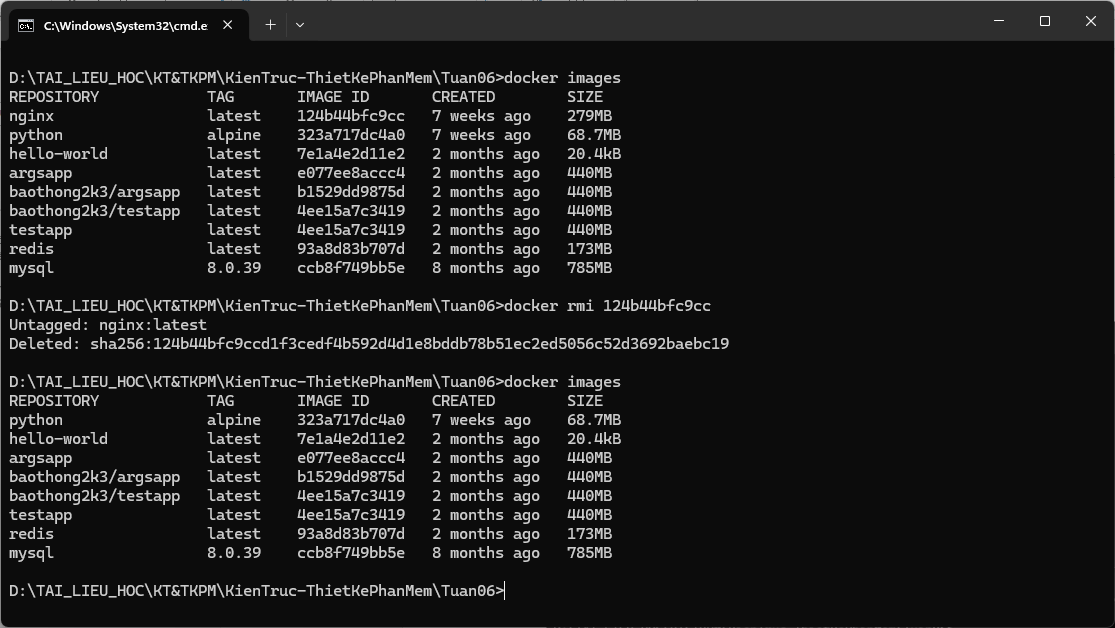
1. docker container prune

Xóa tất cả container đã dừng để giải phóng tài nguyên.



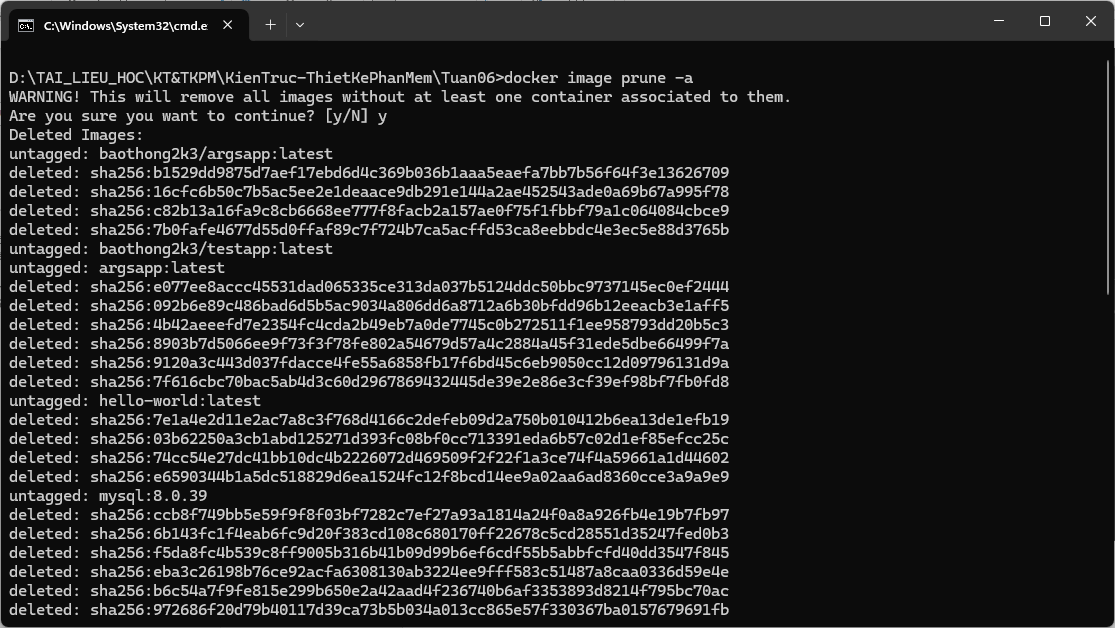
1. docker rmi <image\_id>

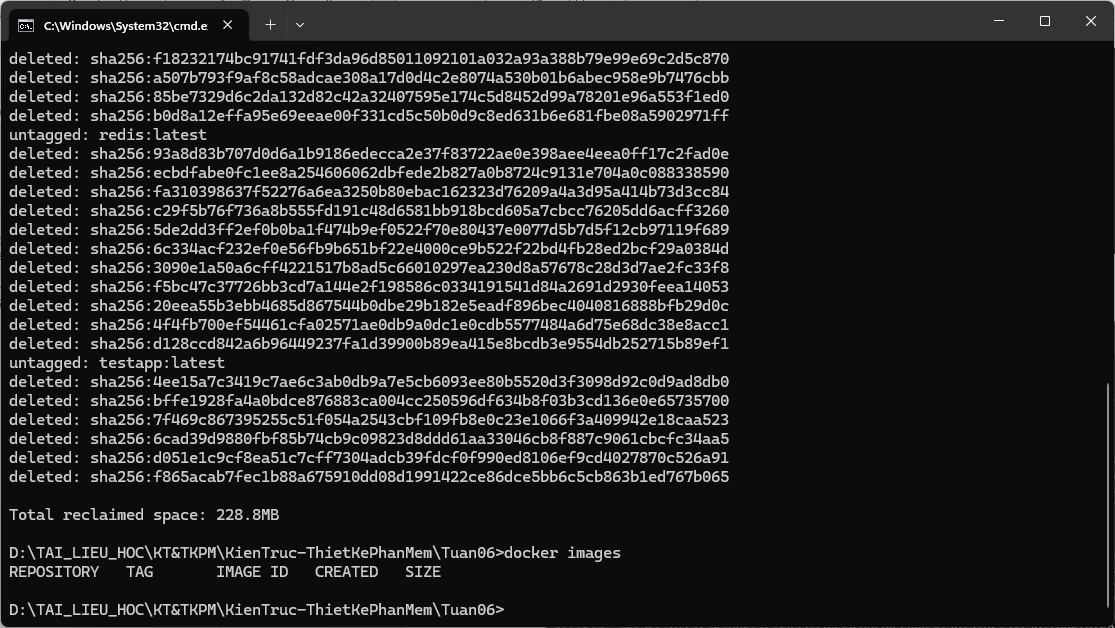
Xóa image có ID <image\_id>.



1. docker image prune -a

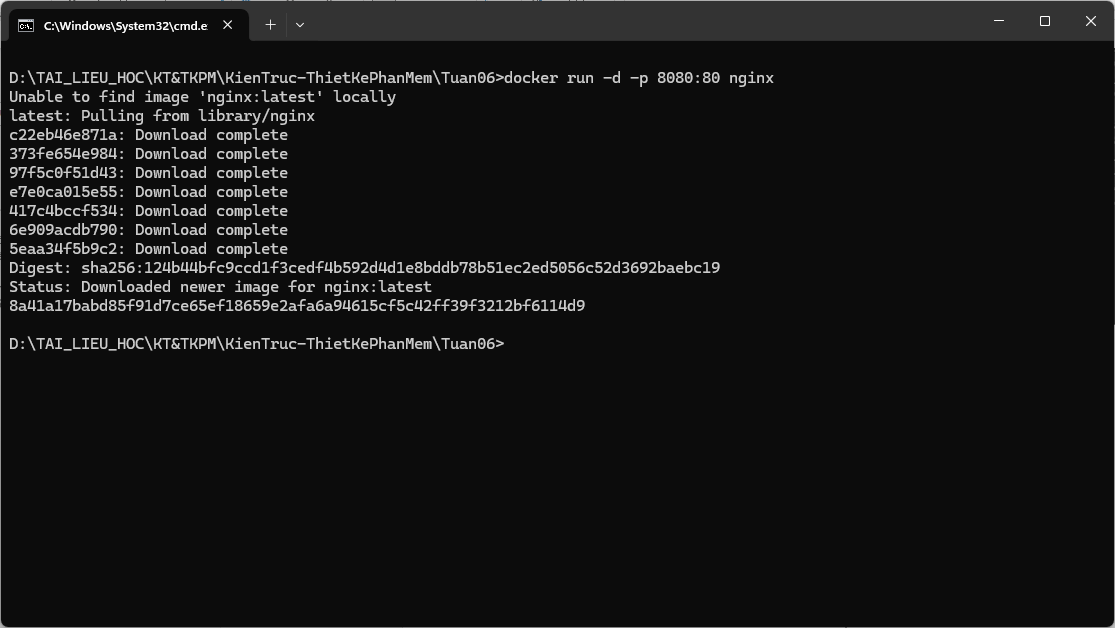
Xóa tất cả image không có container nào sử dụng.

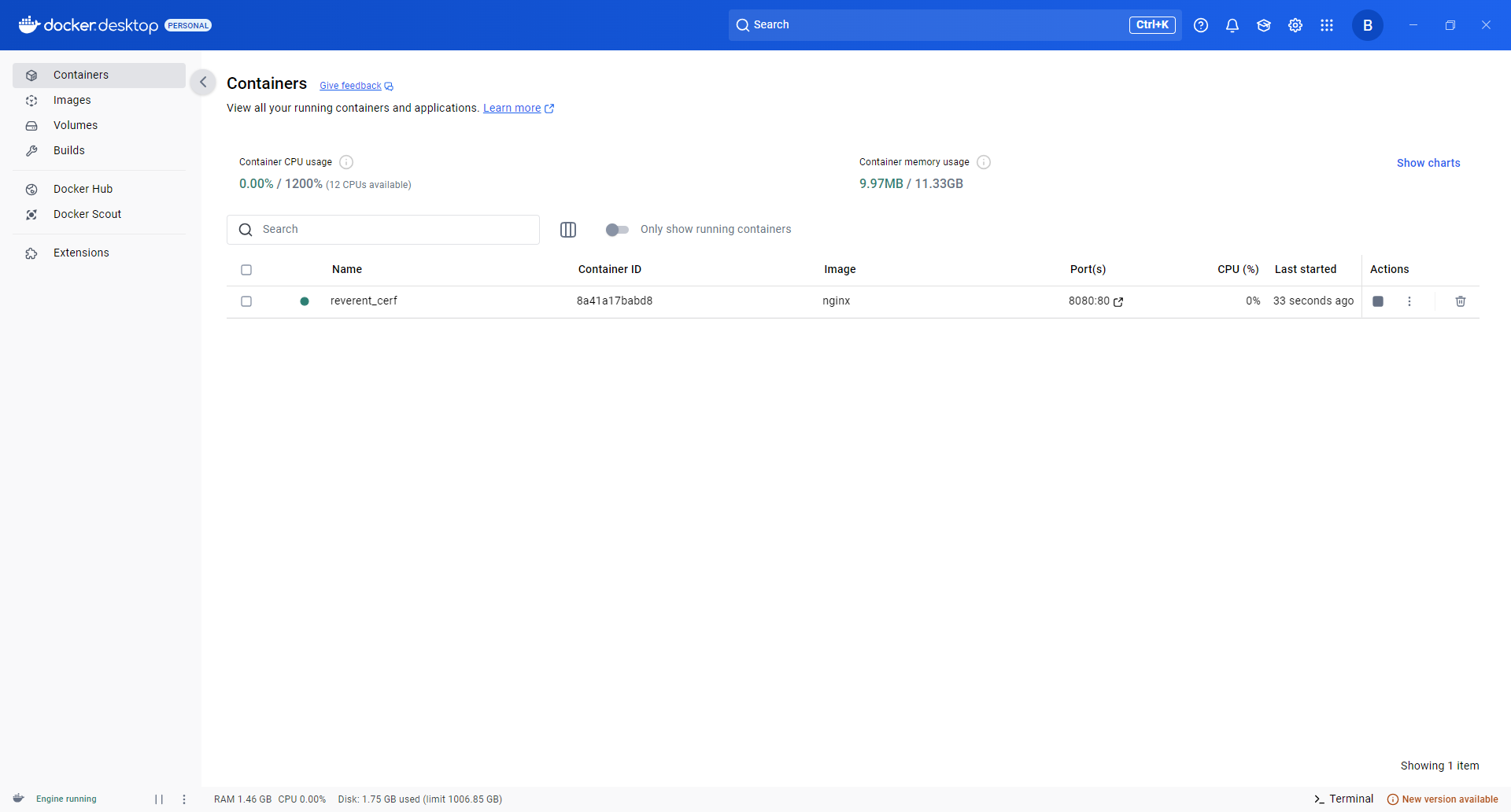




1. docker run -d -p 8080:80 nginx

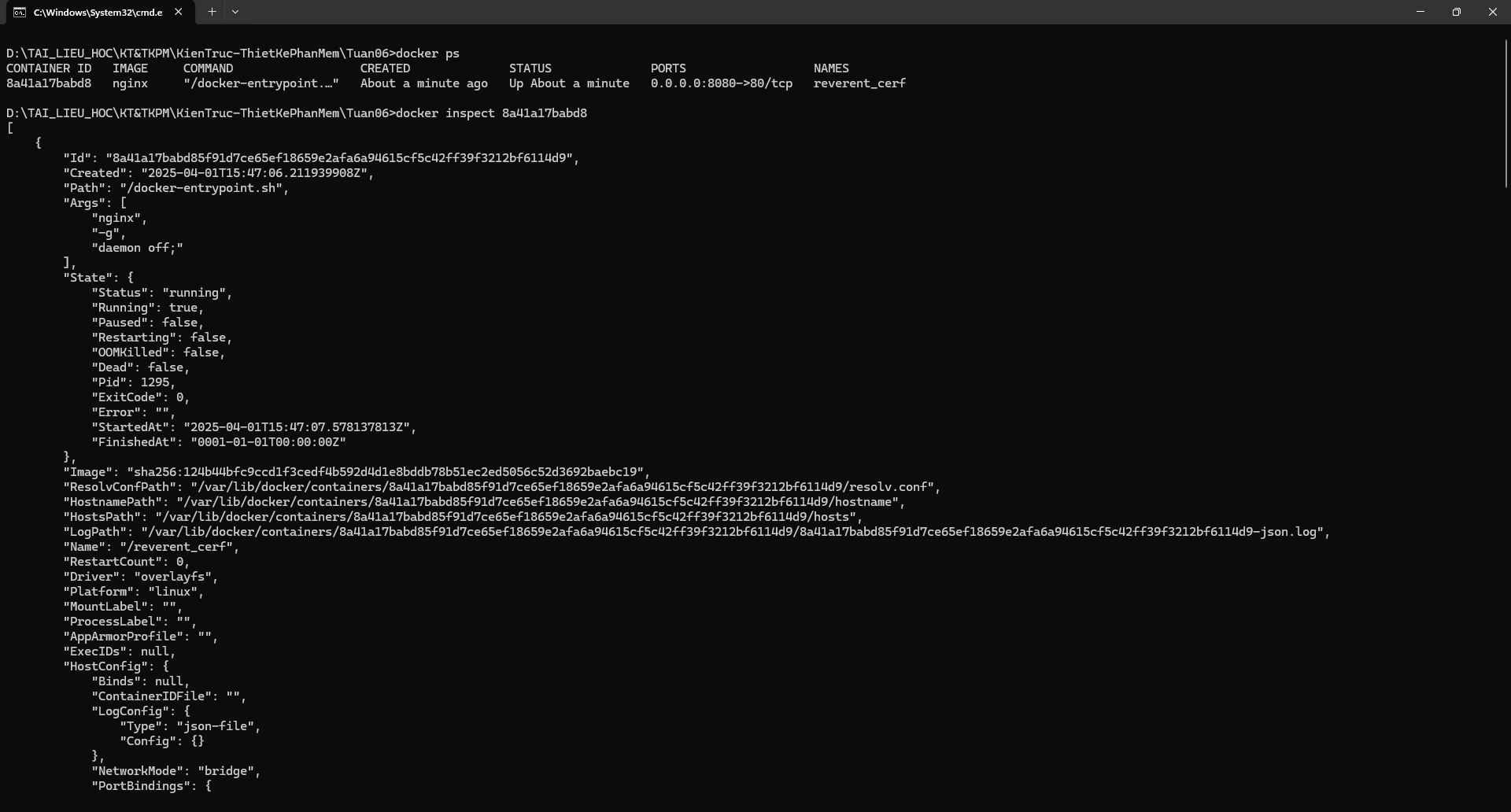
Chạy container Nginx và ánh xạ cổng 8080 của máy với cổng 80 của container.





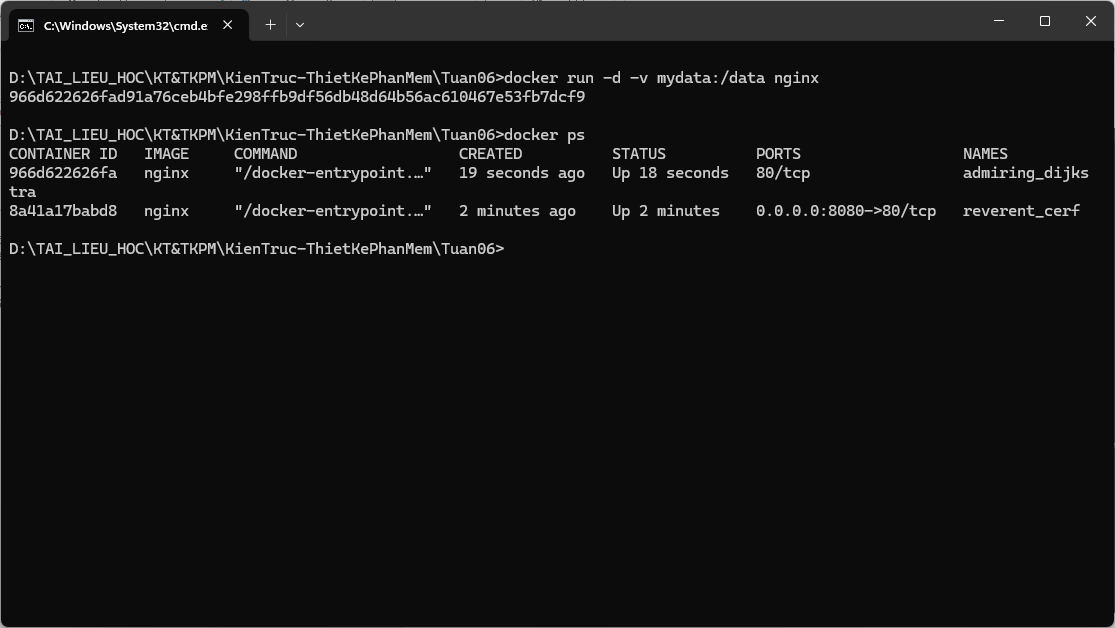
1. docker inspect <container\_id>

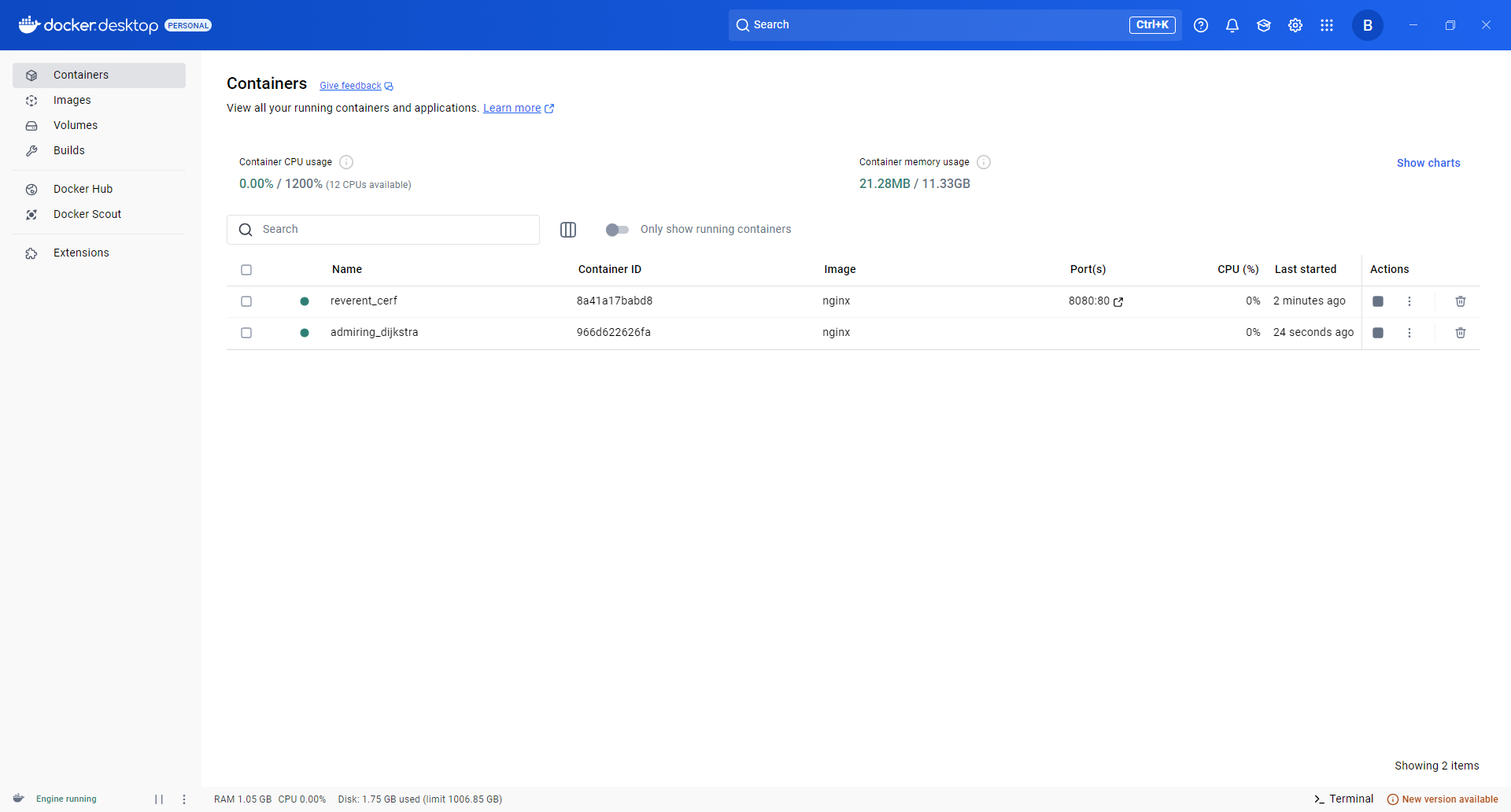
Hiển thị thông tin chi tiết về container, như IP, volumes, cấu hình môi trường, v.v.



1. docker run -d -v mydata:/data nginx

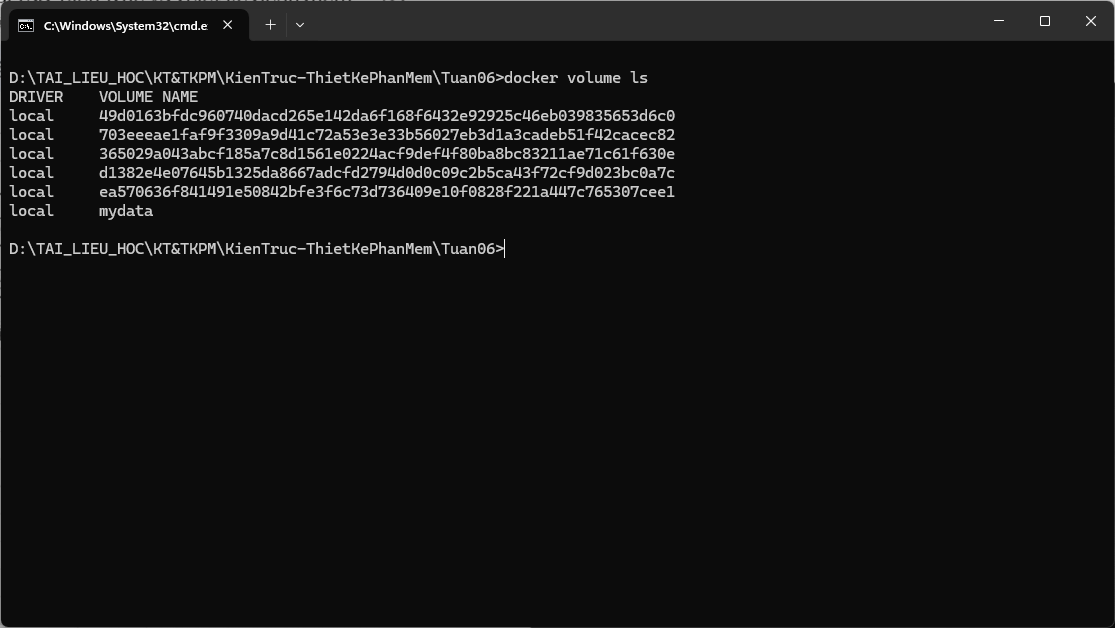
Tạo và gắn volume mydata vào thư mục /data của container Nginx.





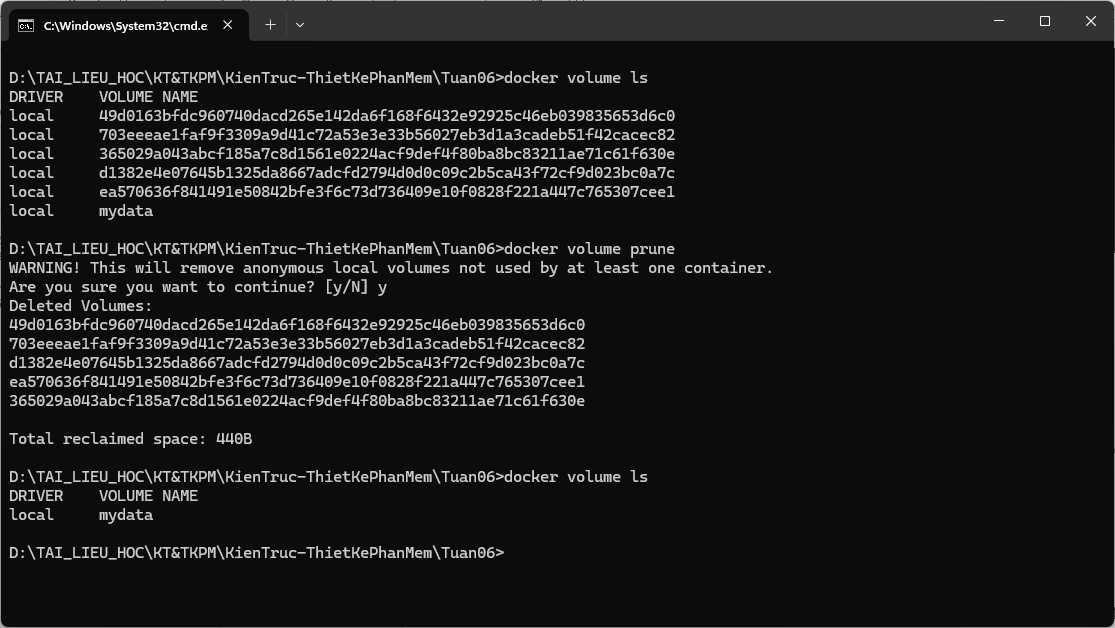
1. docker volume ls

Hiển thị danh sách các volumes.



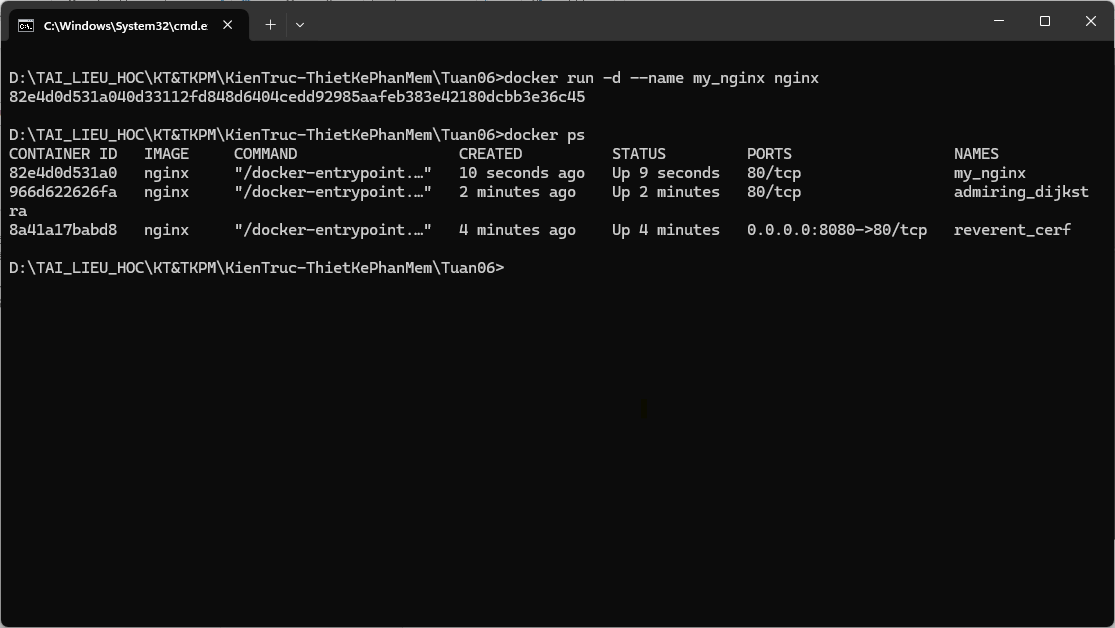
1. docker volume prune

Xóa tất cả volumes không gắn với container nào.



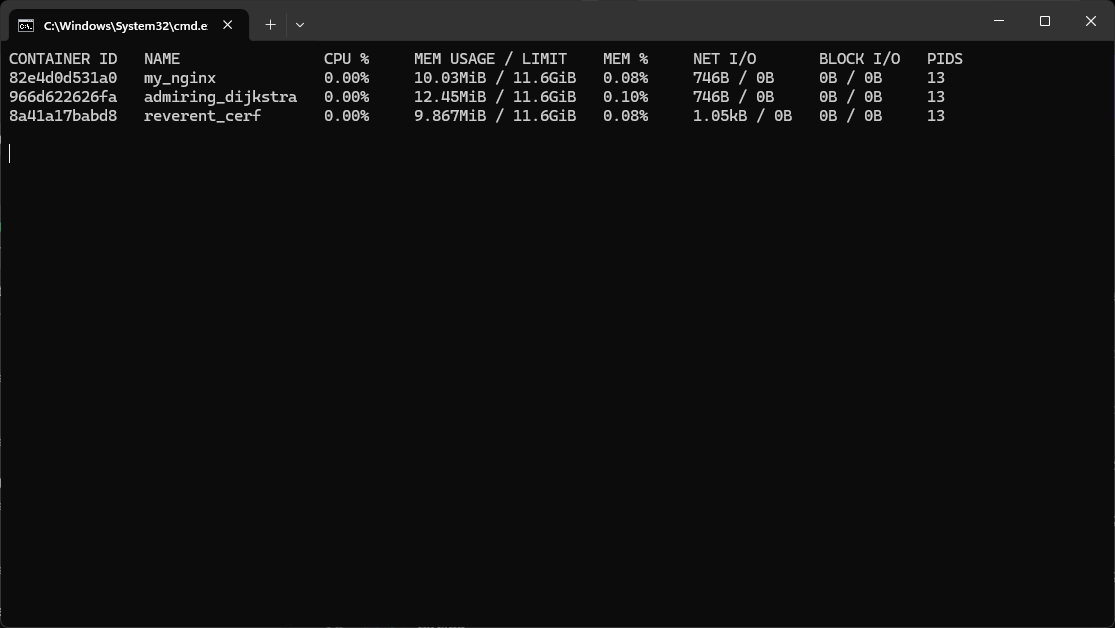
1. docker run -d --name my\_nginx nginx

Chạy container từ nginx với tên my\_nginx.



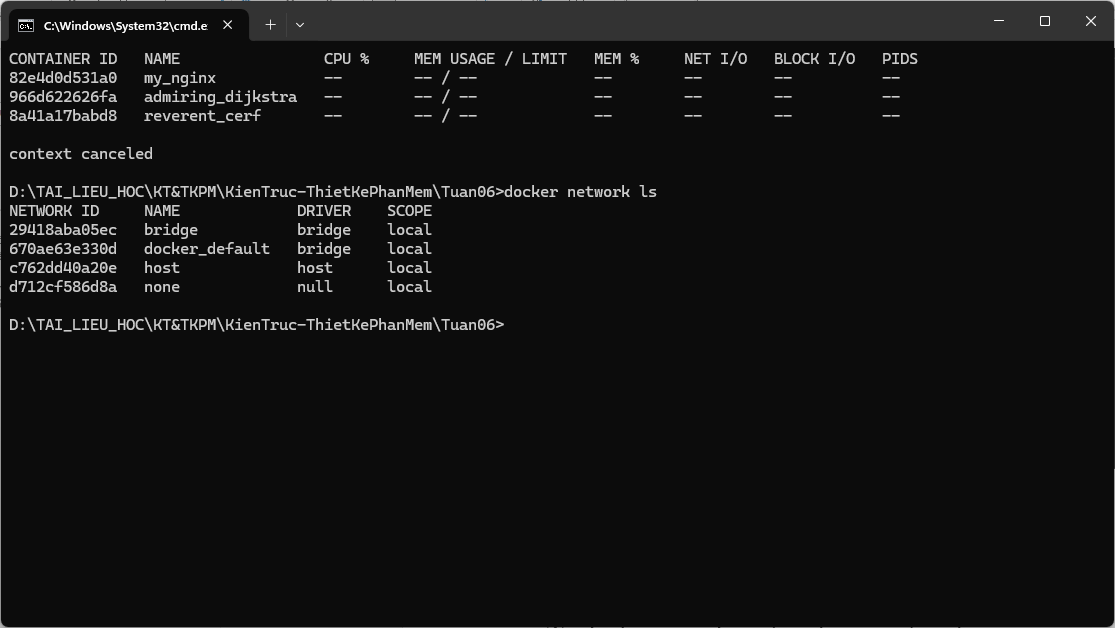
1. docker stats

Hiển thị CPU, RAM, và tài nguyên mà các container đang sử dụng.



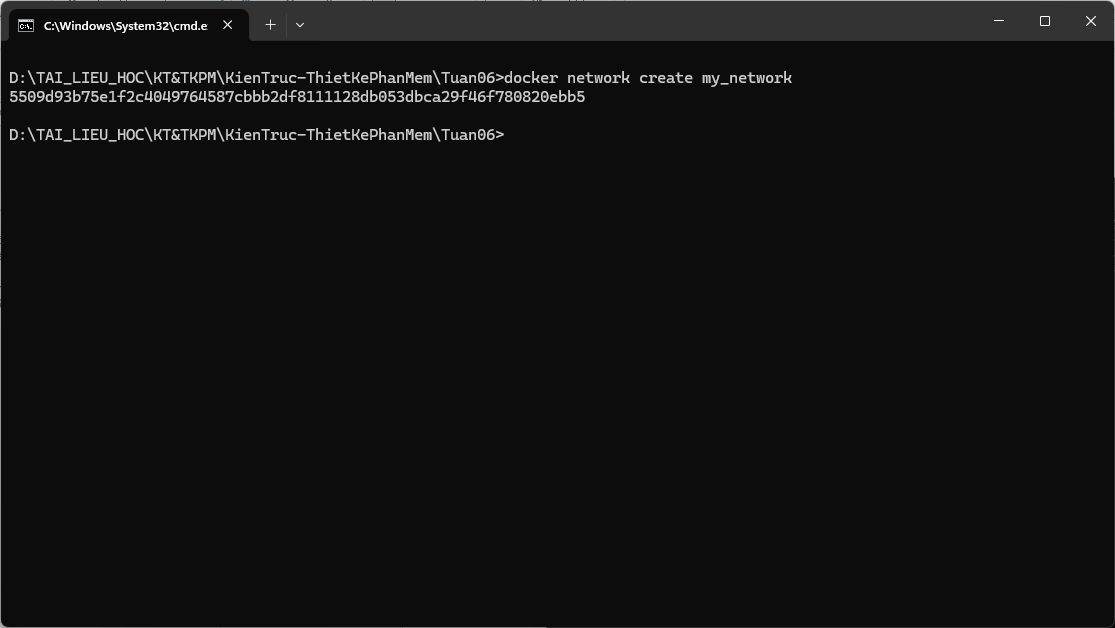
1. docker network ls

Hiển thị danh sách các networks trong Docker.



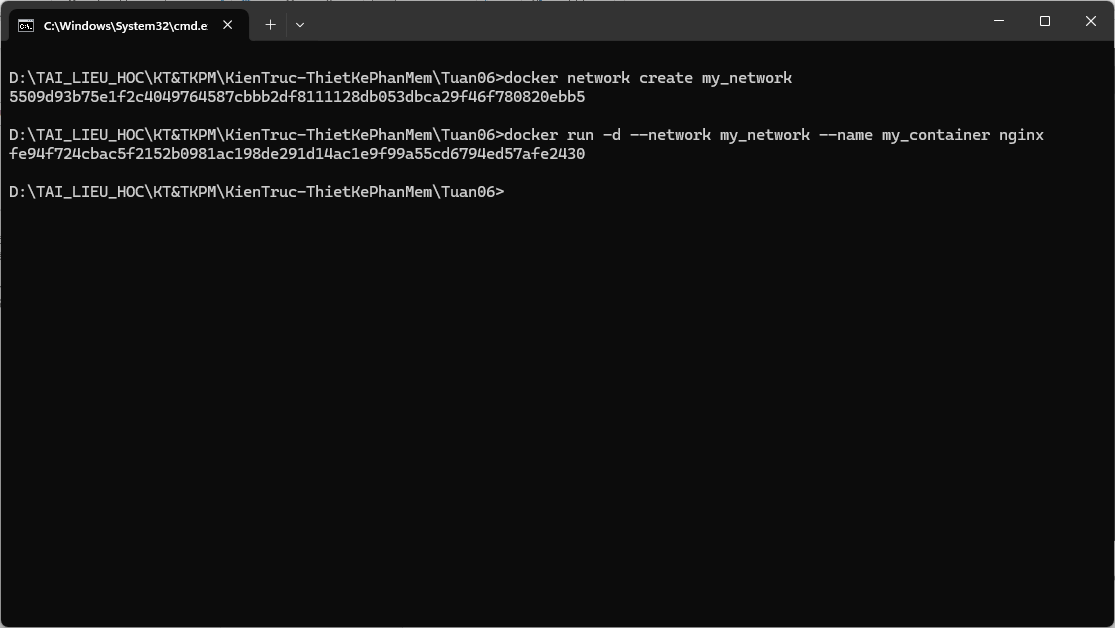
1. docker network create my\_network

Tạo một Docker network có tên my\_network.



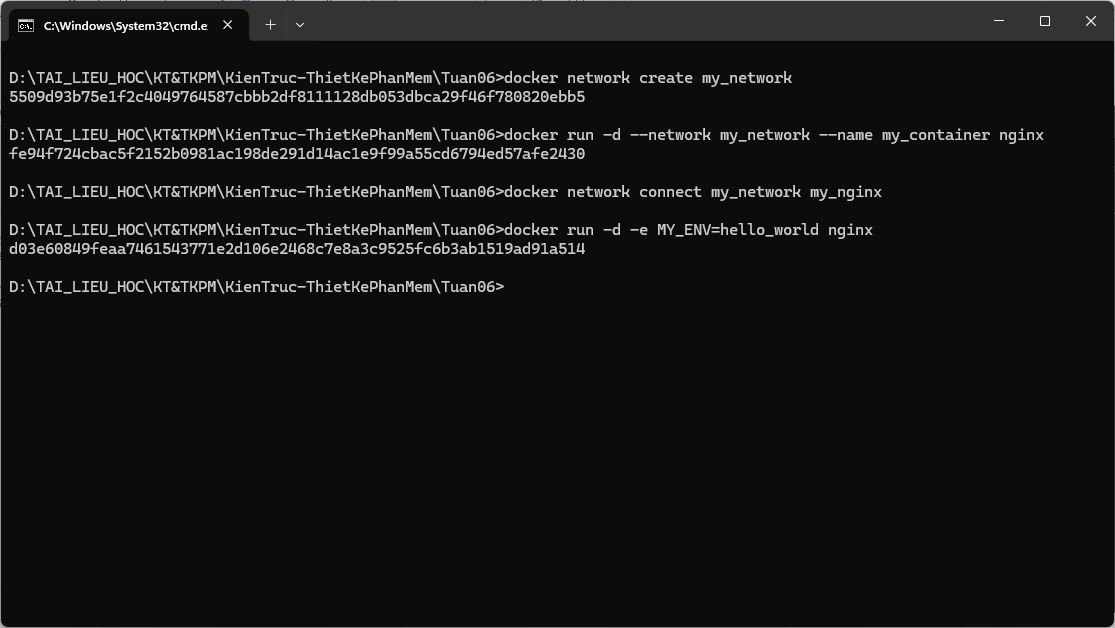
1. docker run -d --network my\_network --name my\_container nginx

Chạy container my\_container trong network my\_network.



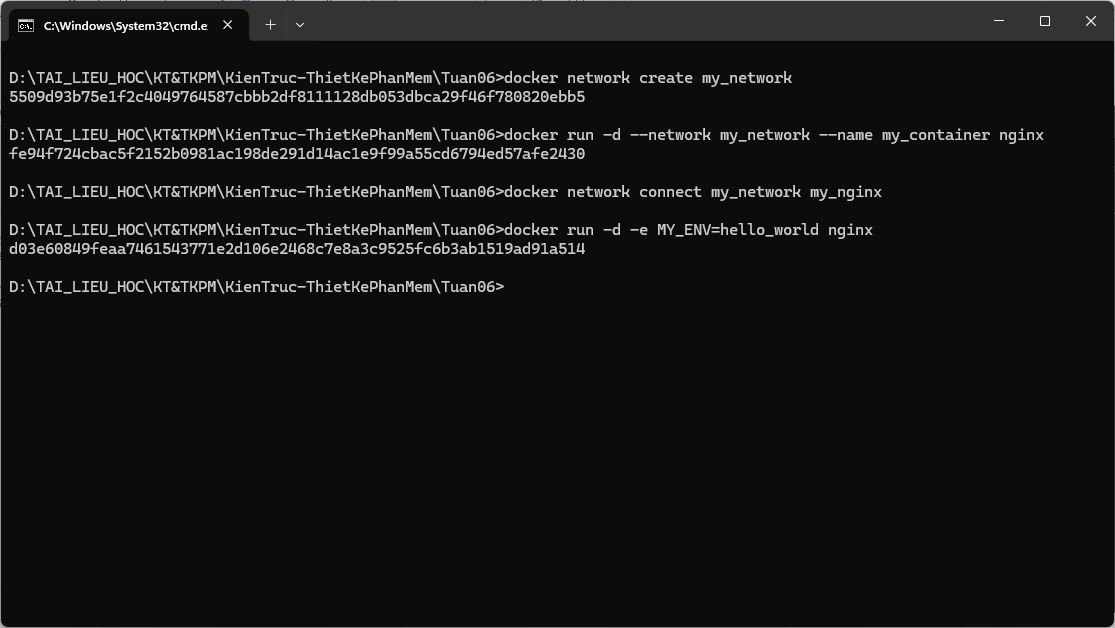
1. docker network connect my\_network my\_nginx

Kết nối container my\_nginx với network my\_network.



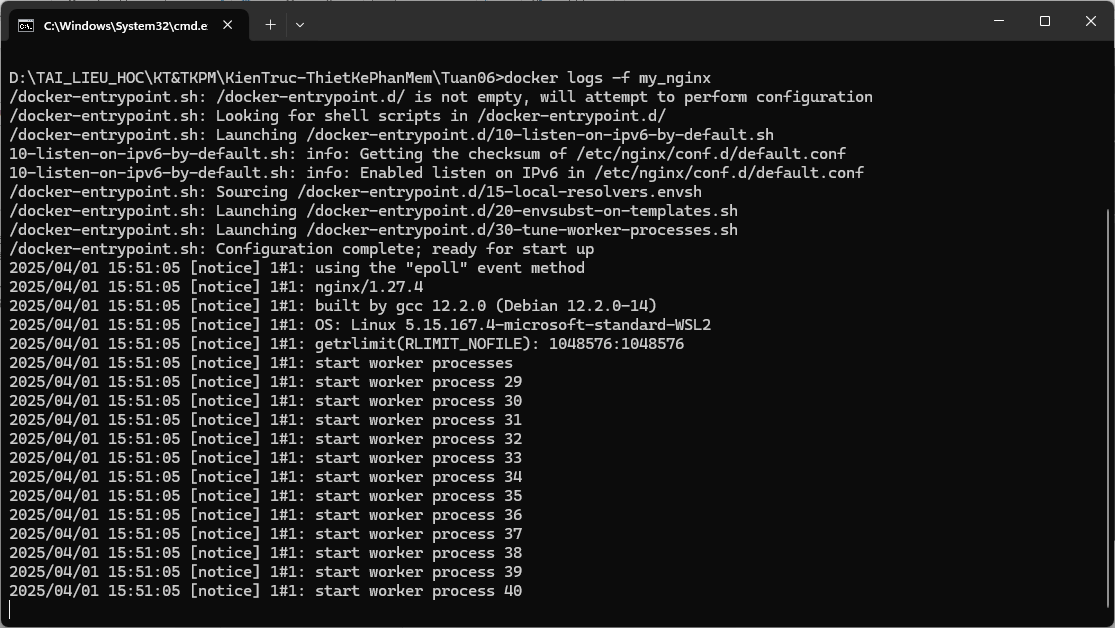
1. docker run -d -e MY\_ENV=hello\_world nginx

Chạy container Nginx và thiết lập biến môi trường MY\_ENV=hello\_world.



1. docker logs -f my\_nginx

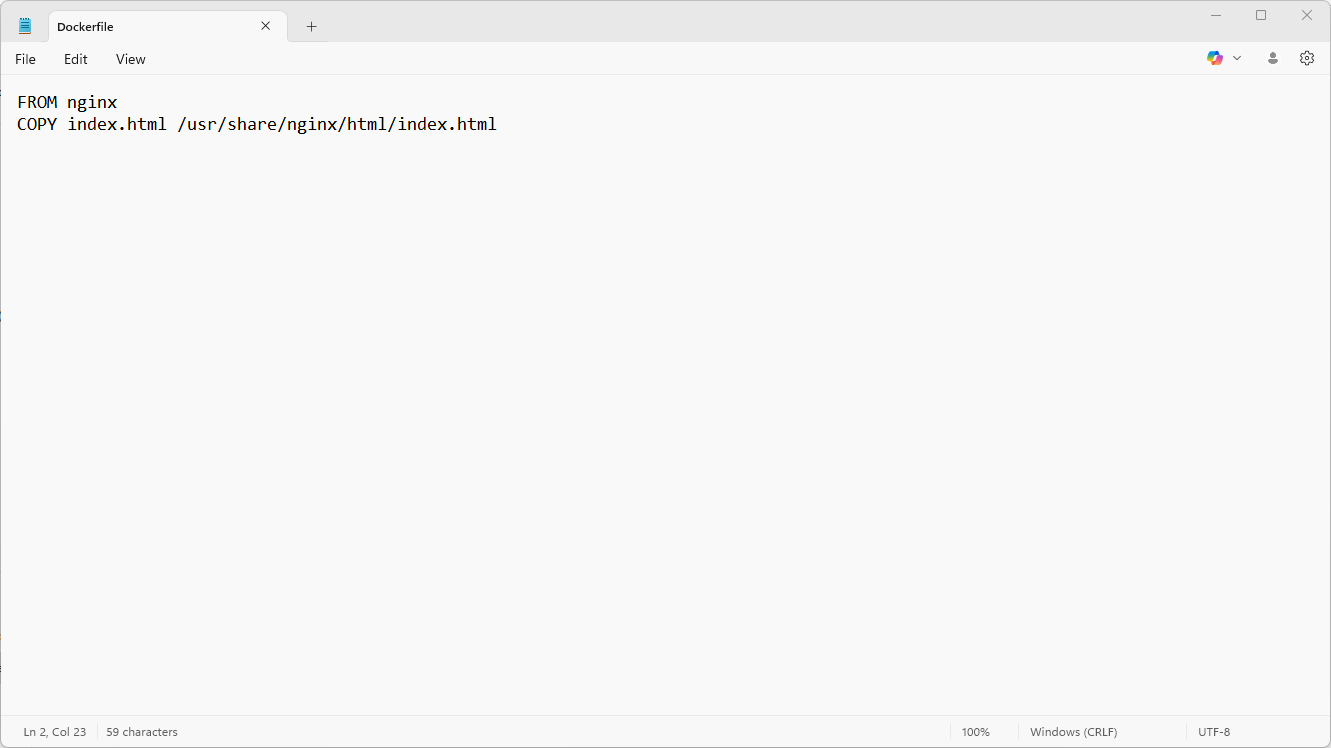
Theo dõi log của container my\_nginx theo thời gian thực.



1. FROM nginx

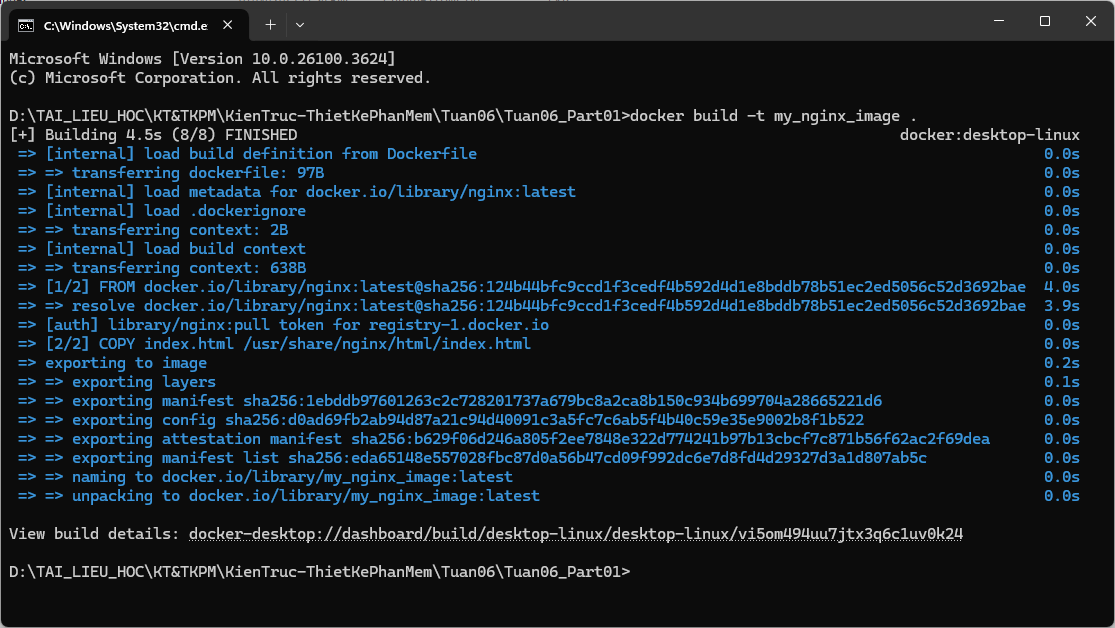
COPY index.html /usr/share/nginx/html/index.html

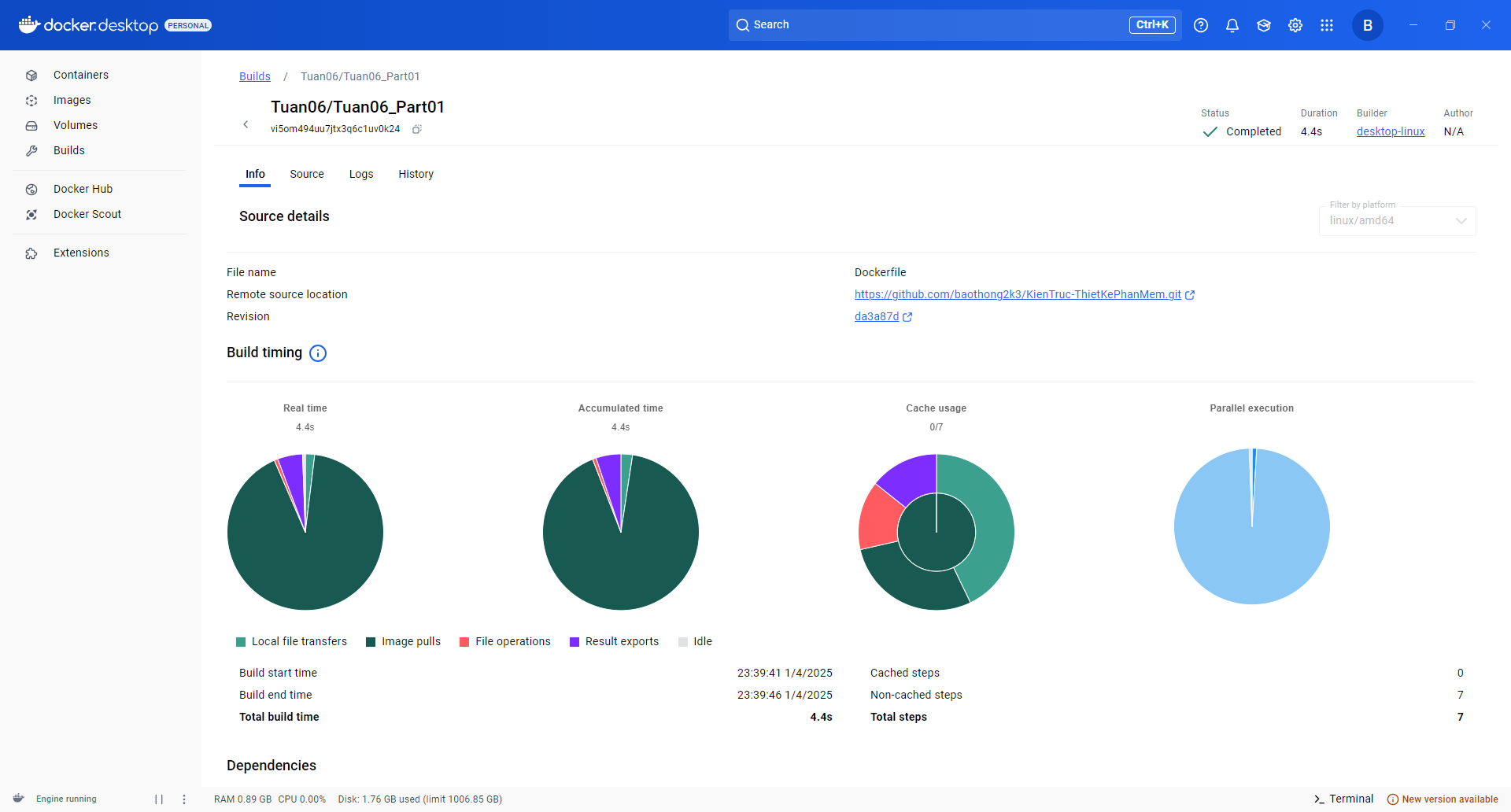
Viết Dockerfile với Nginx



1. docker build -t my\_nginx\_image .

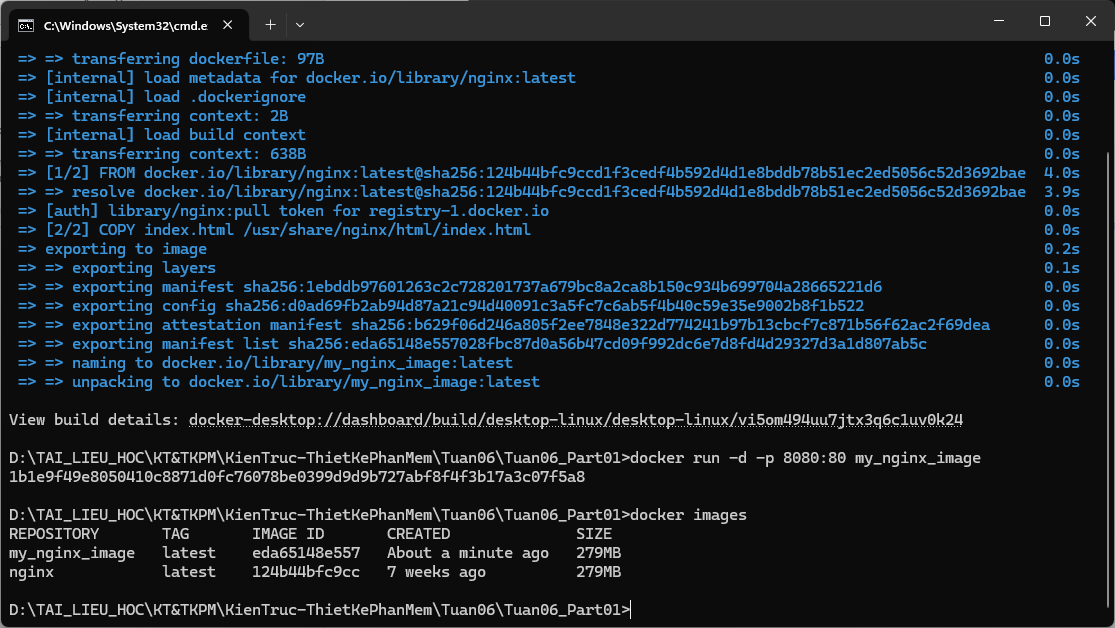
Build image từ Dockerfile và đặt tên my\_nginx\_image.



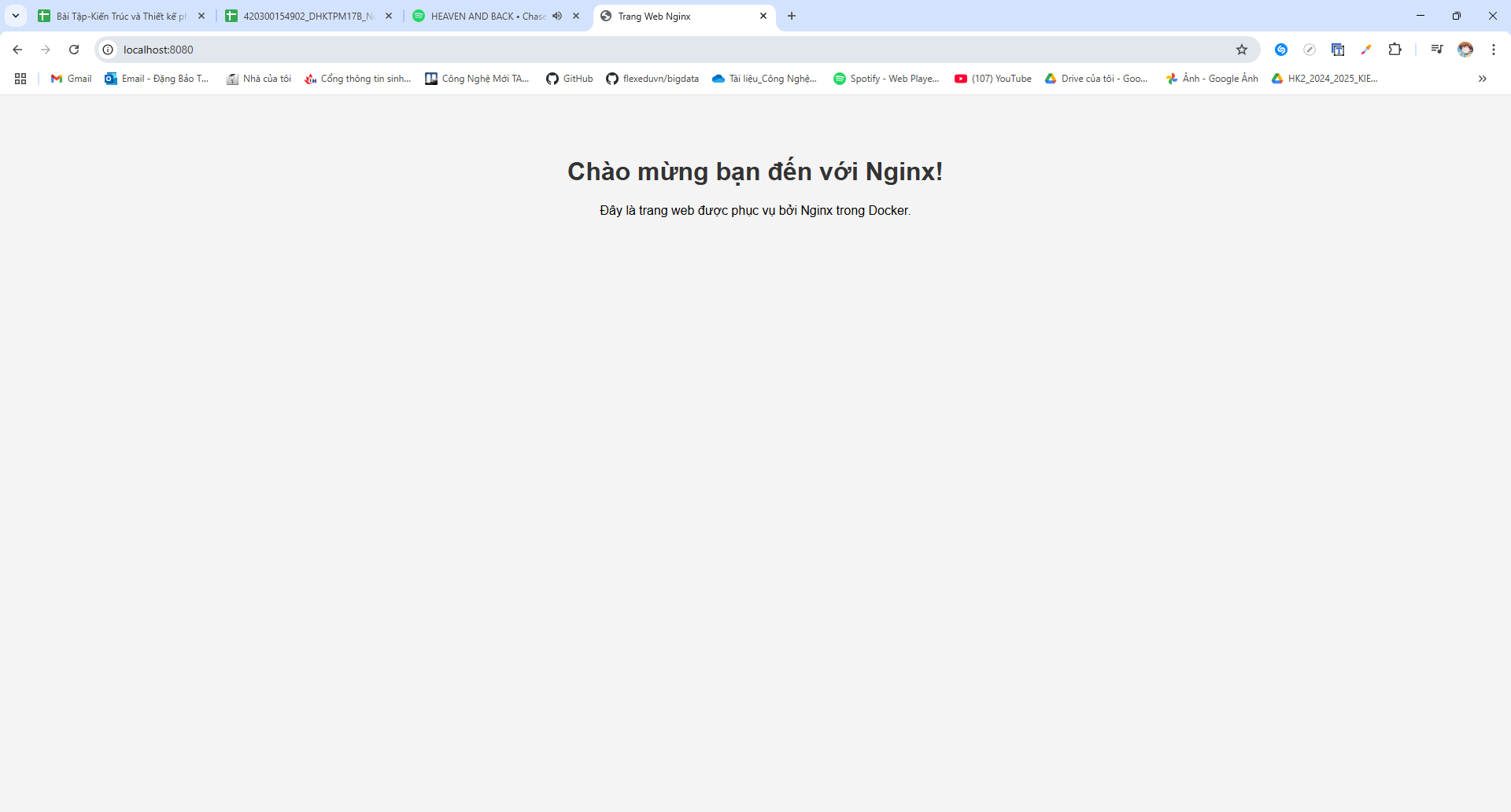


1. docker run -d -p 8080:80 my\_nginx\_image

Chạy container từ my\_nginx\_image, ánh xạ cổng 8080 của máy với cổng 80 của container.



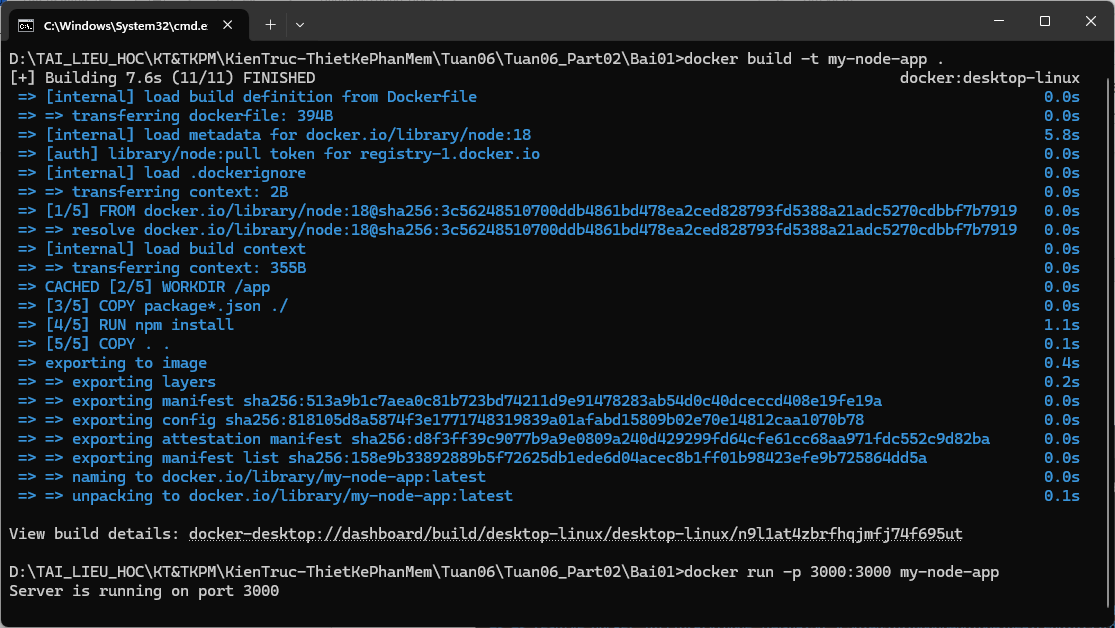
http://localhost:8080/

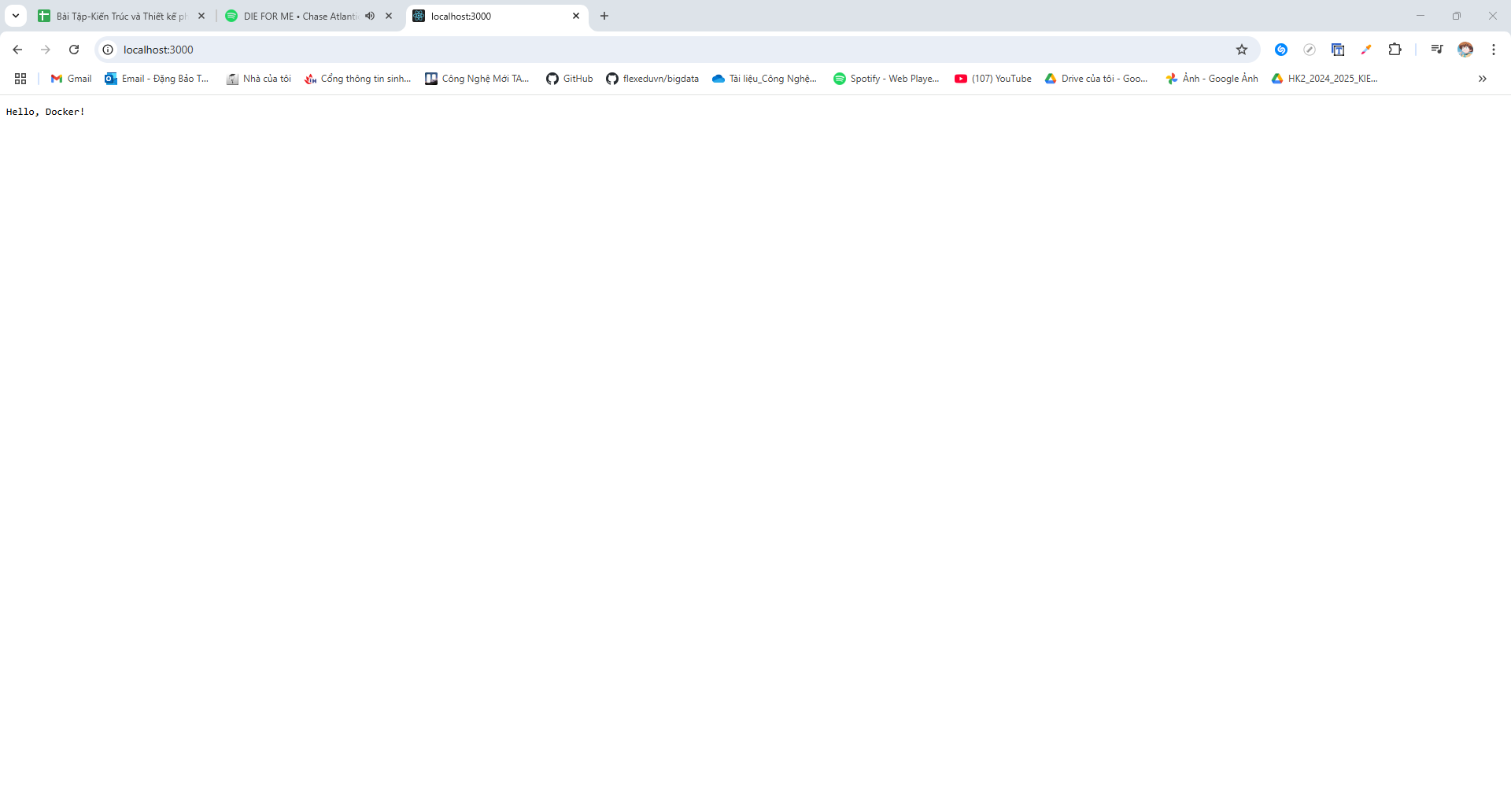


**Phần 2: Thao tác với Dockerfile**  
**Bài 1:** Tạo Dockerfile chạy một ứng dụng Node.js đơn giản

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.

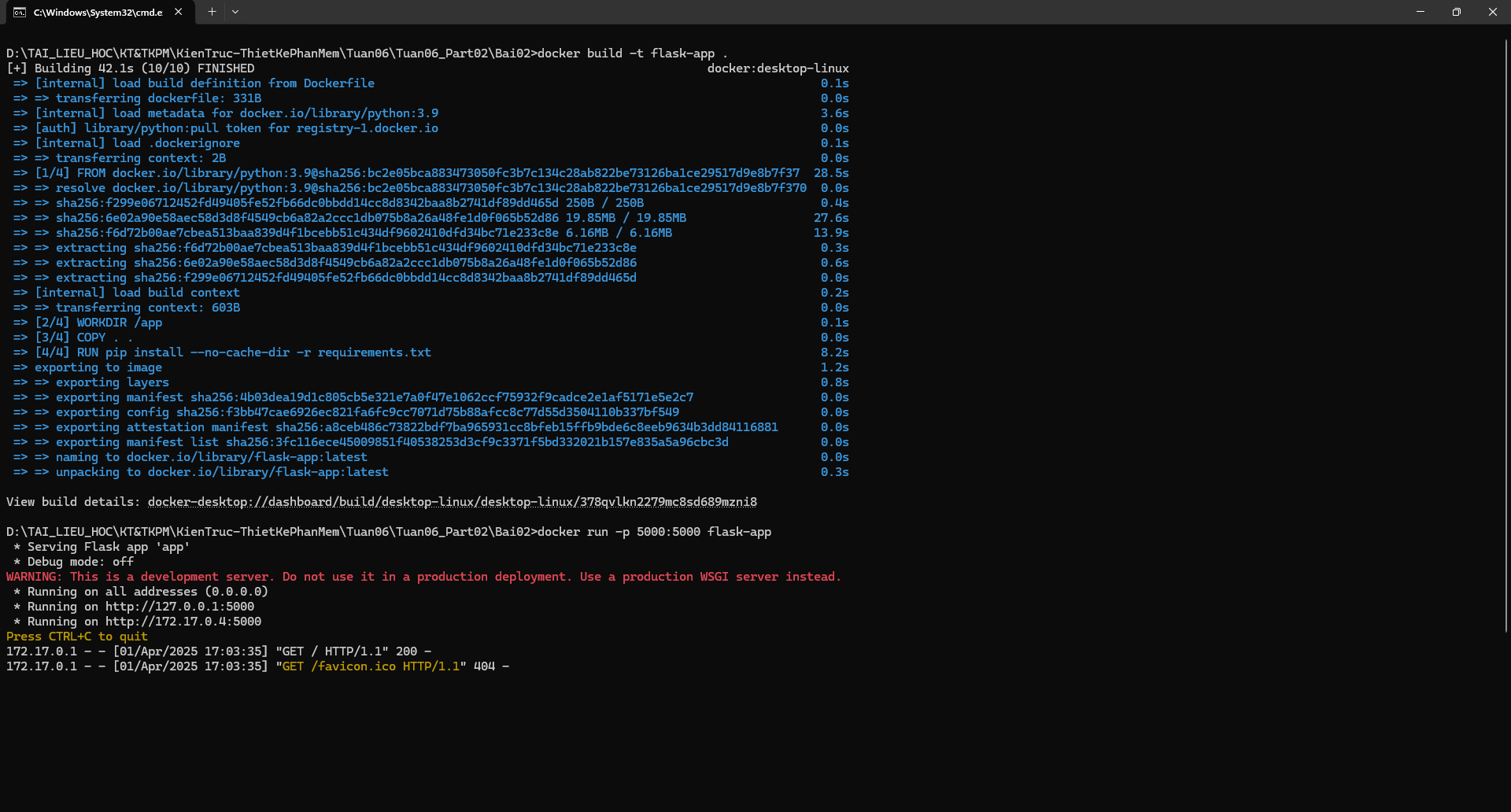


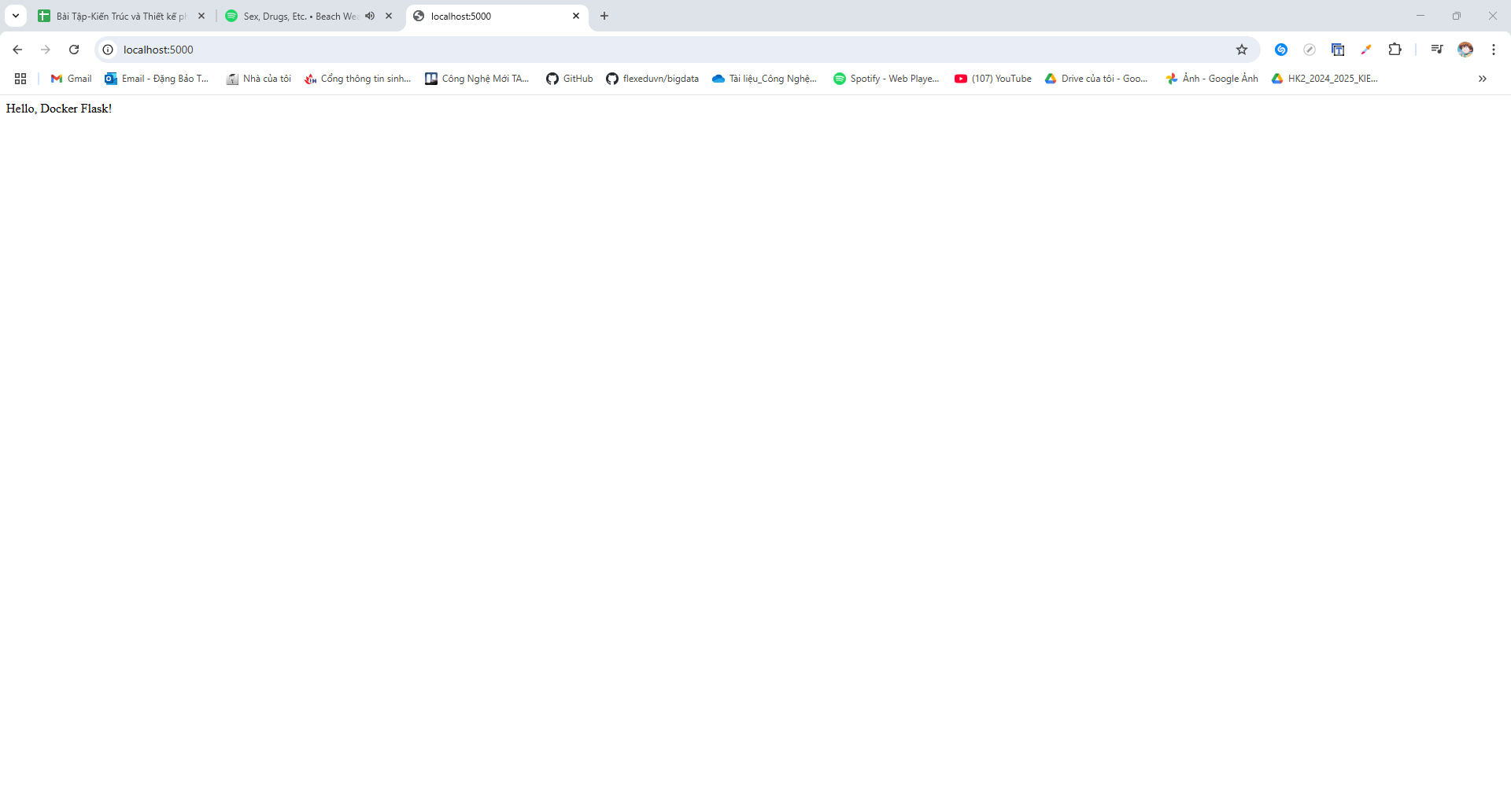


**Bài 2:** Tạo Dockerfile chạy một ứng dụng Python Flask

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.

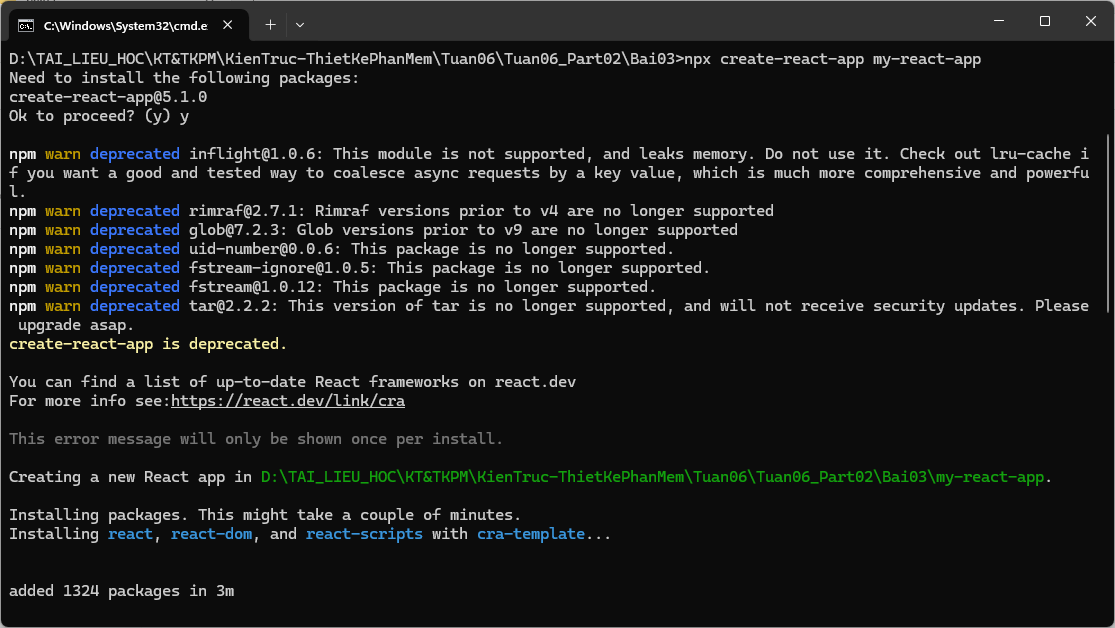


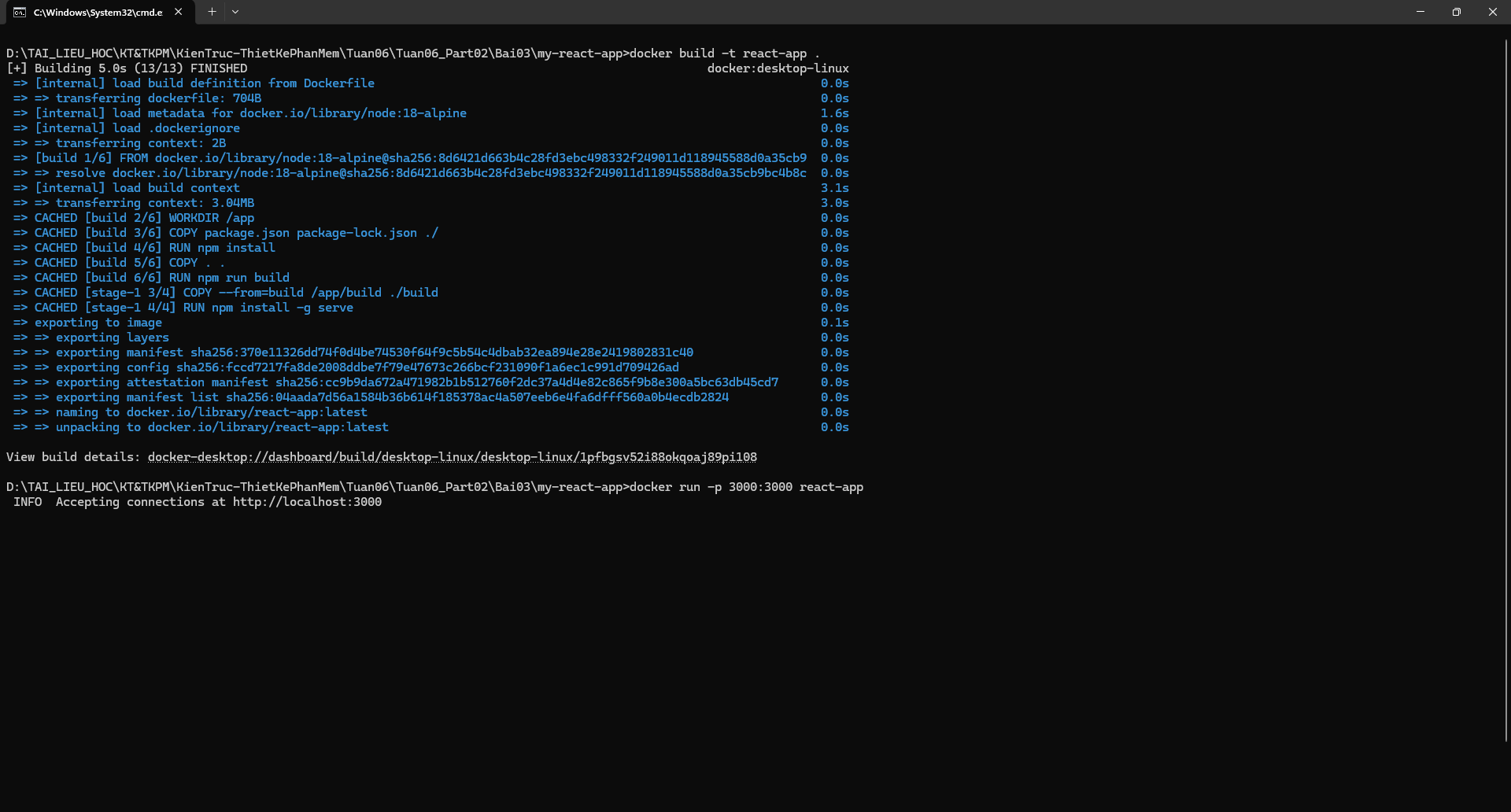


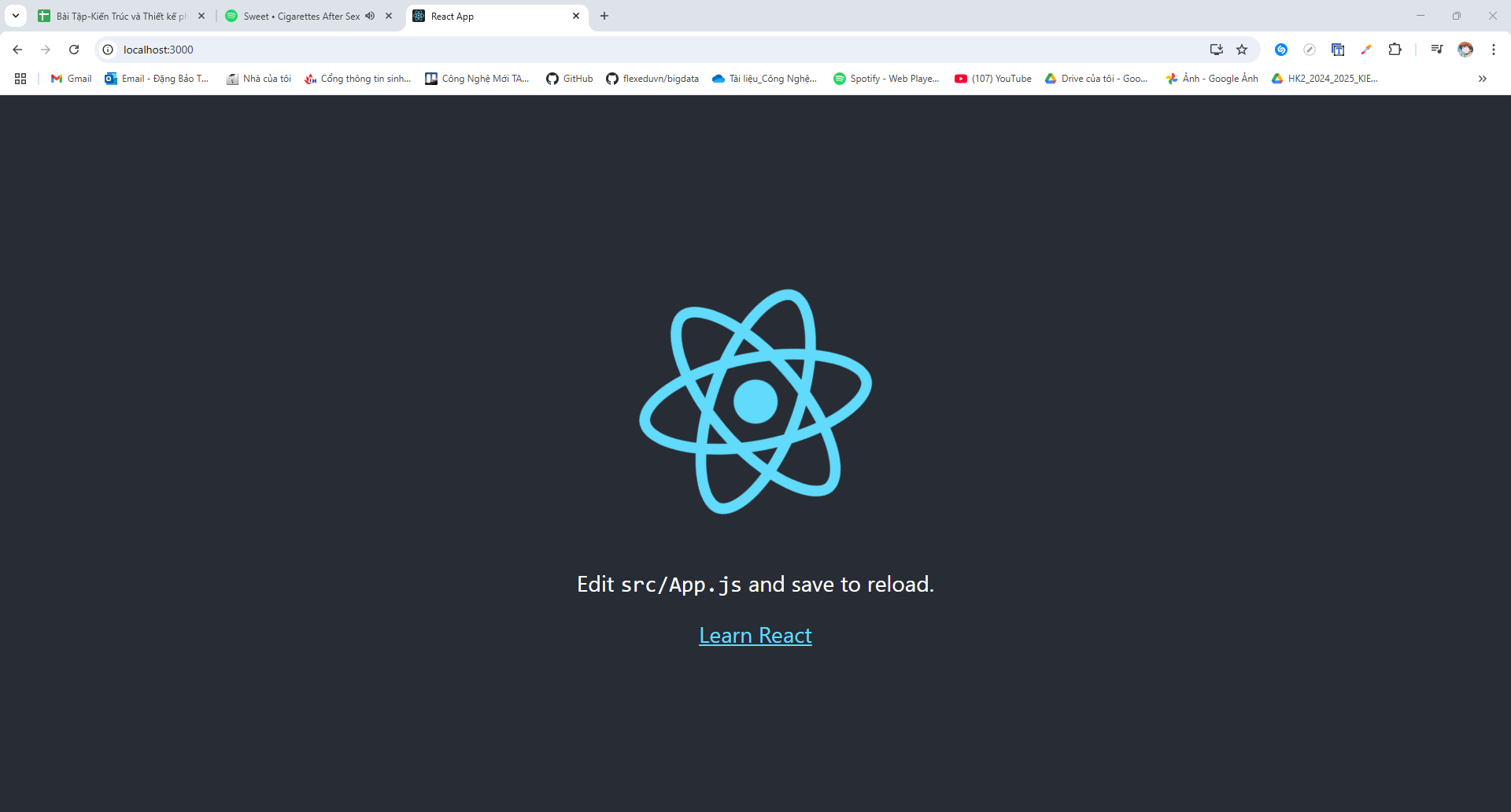
**Bài 3:** Tạo Dockerfile chạy một ứng dụng React

Yêu cầu: Viết Dockerfile để build và chạy một ứng dụng React.

Sử dụng node:18-alpine làm base image.



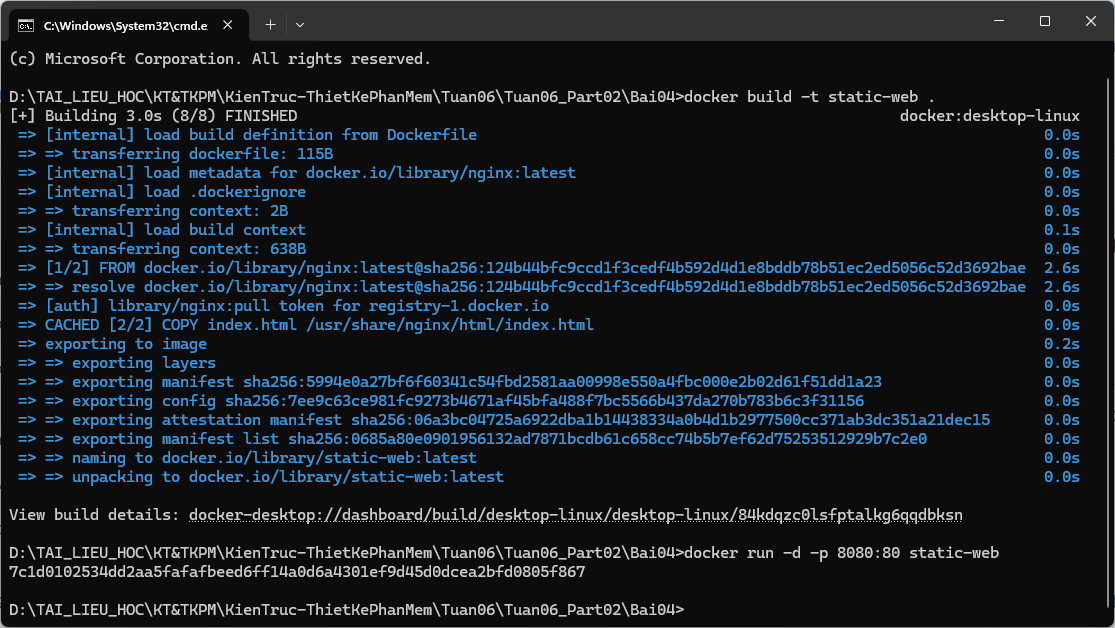


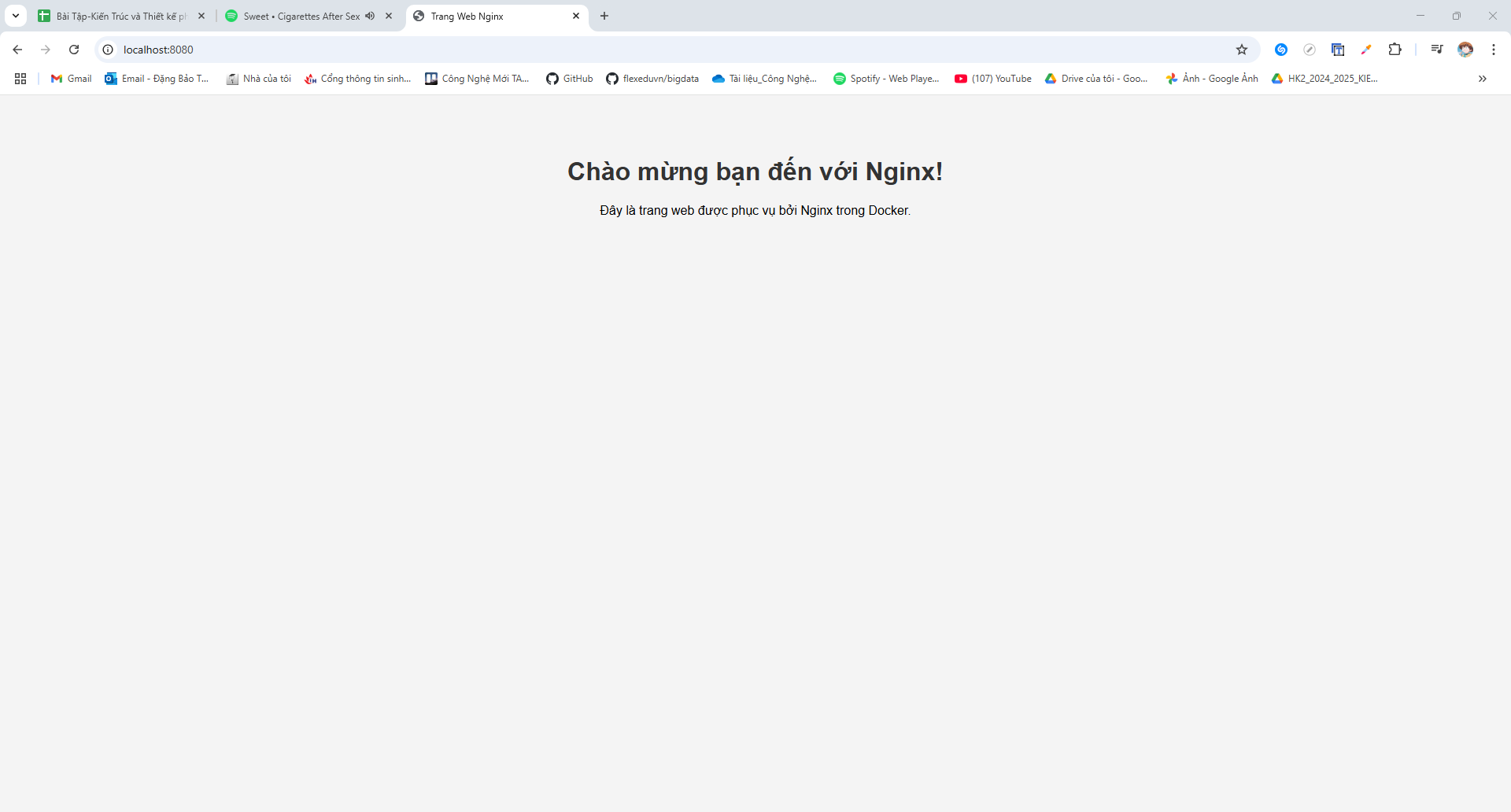


http://localhost:3000/

**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.

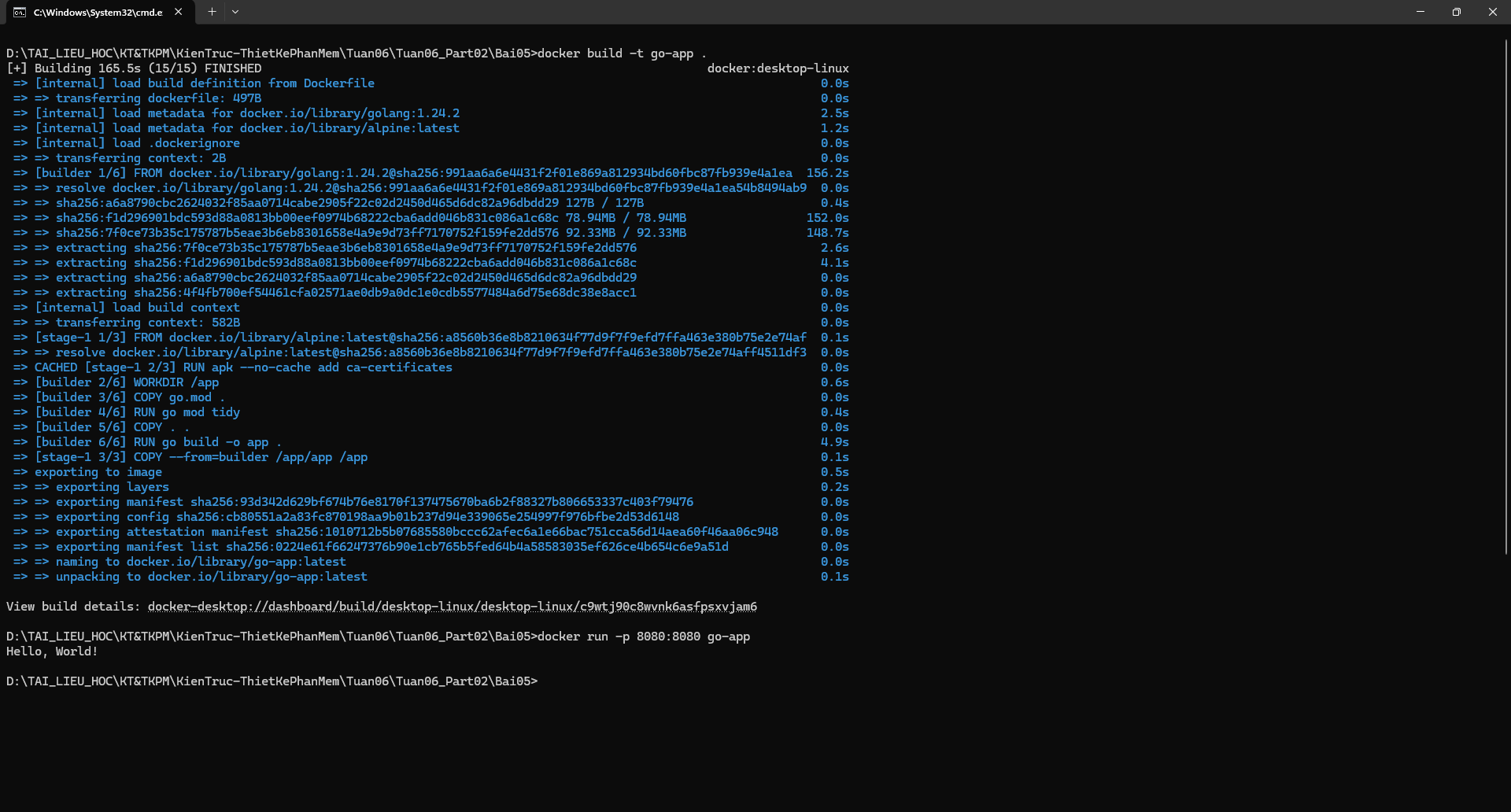




http://localhost:8080/

**Bài 5:** Tạo Dockerfile cho ứng dụng Go

Yêu cầu: Viết Dockerfile để build và chạy một ứng dụng Go đơn giản.

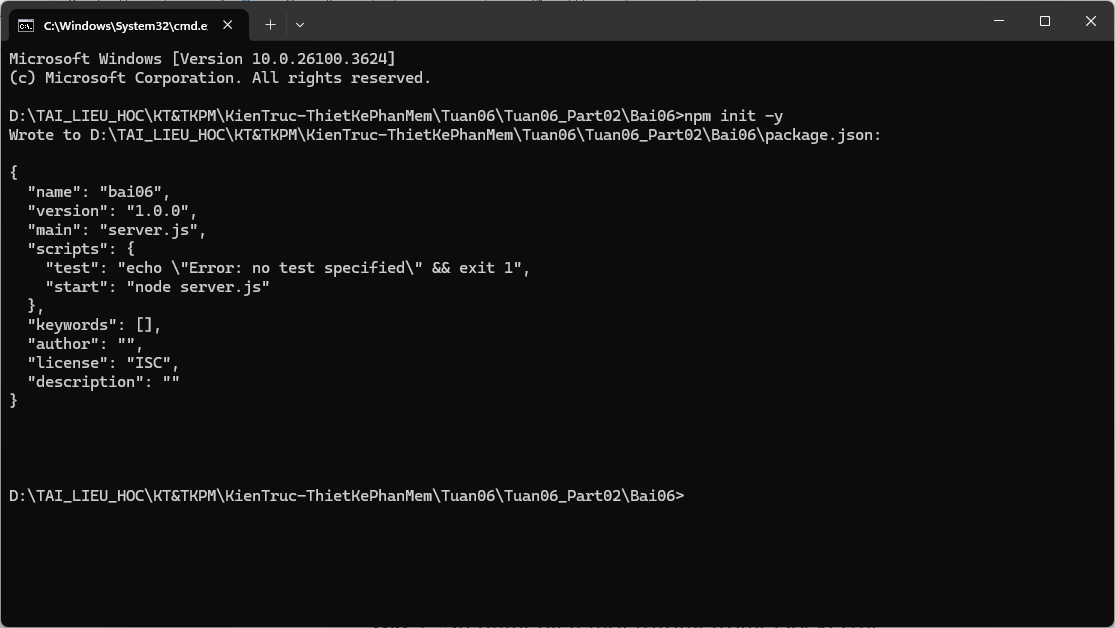


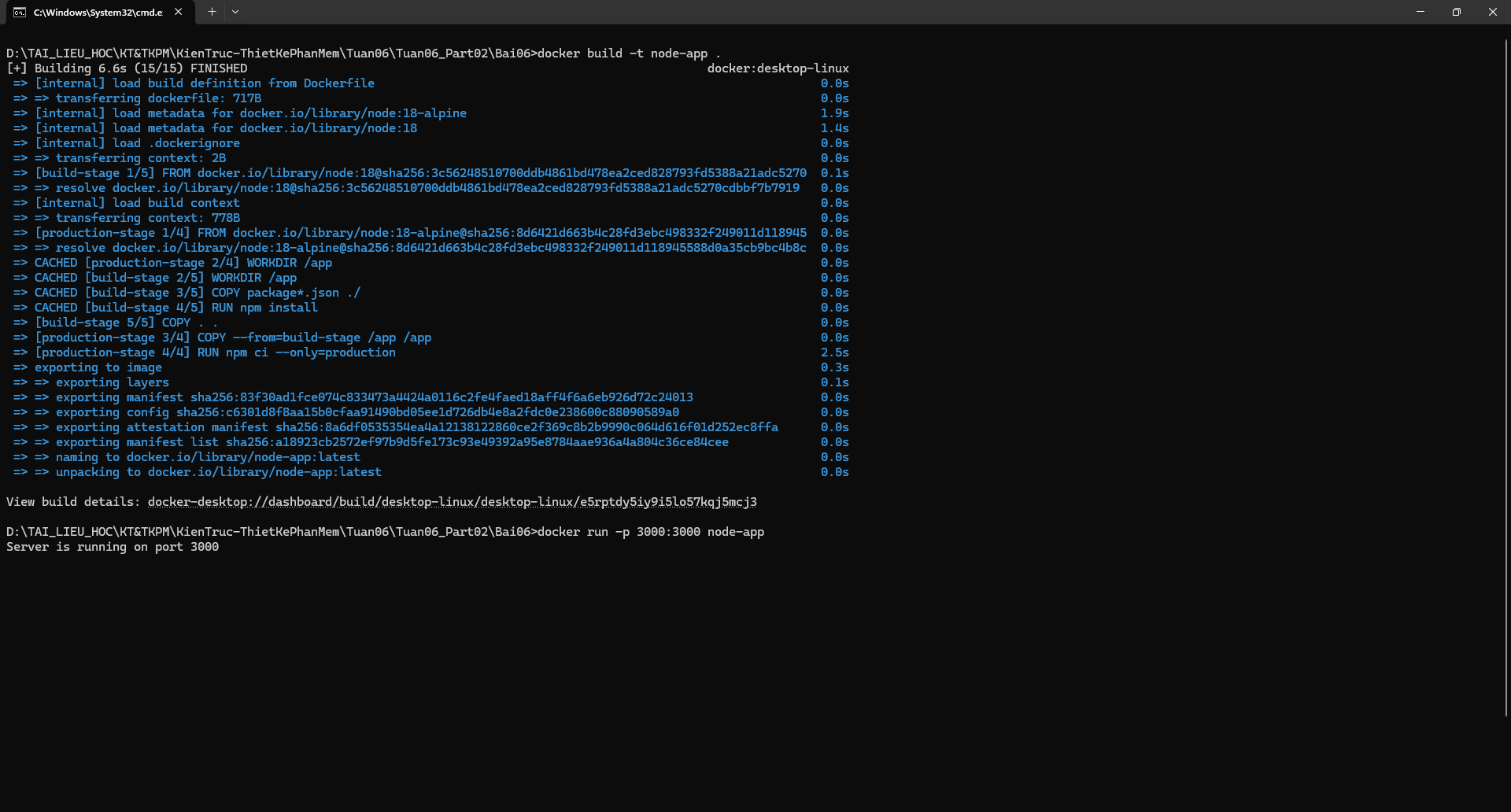
**Bài 6:** Sử dụng Multi-stage Build trong Dockerfile

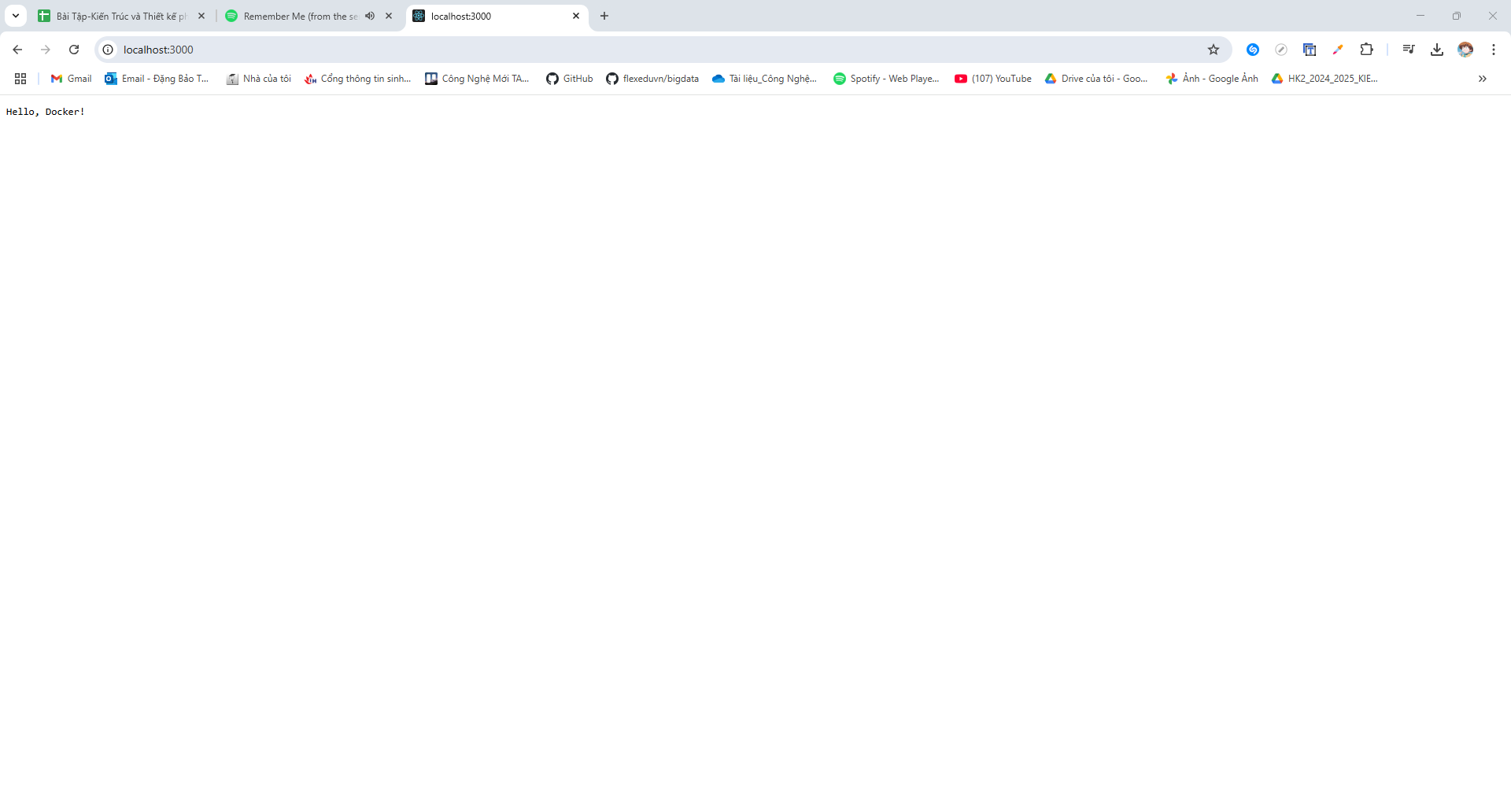
Viết Dockerfile để build một ứng dụng Node.js với hai stage:

* Stage 1: Dùng node:18 để build code.
* Stage 2: Dùng node:18-alpine để chạy ứng dụng đã build.

npm init -y



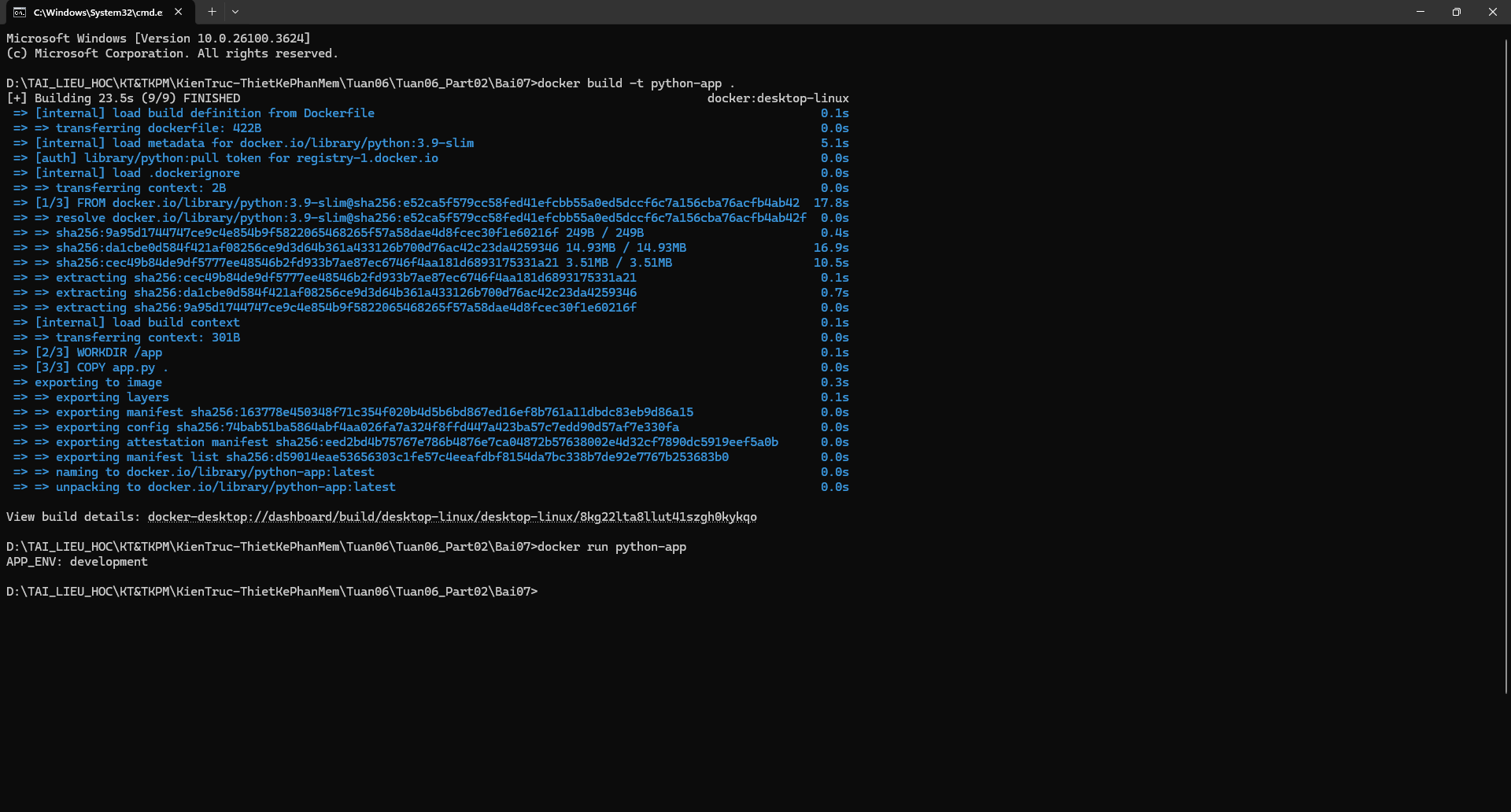




**Bài 7:** Sử dụng biến môi trường trong Dockerfile

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.



**Bài 8:** Tạo Dockerfile cho PostgreSQL tùy chỉnh

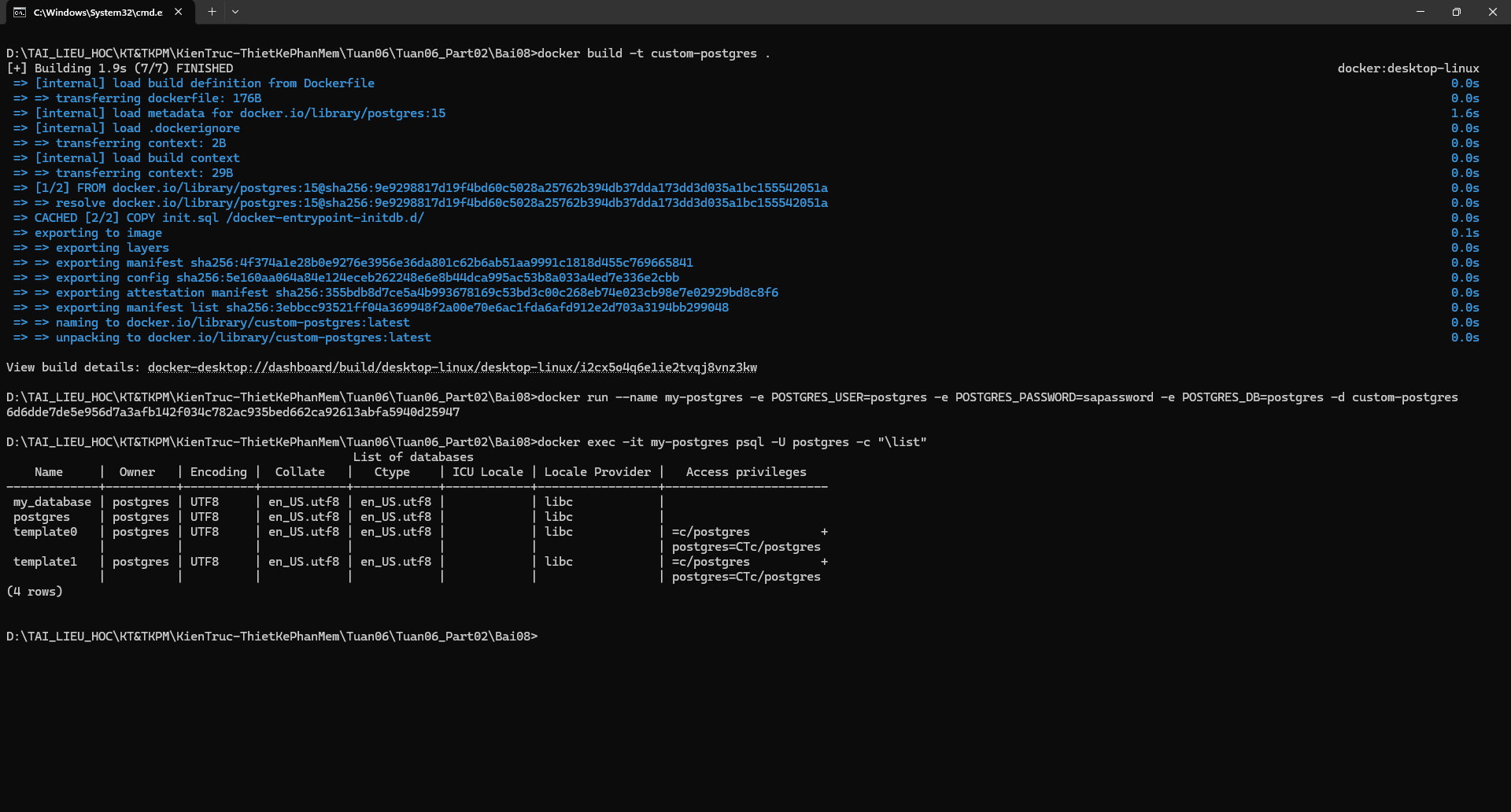
Yêu cầu: Viết Dockerfile để chạy PostgreSQL (postgres:15).

Thêm file SQL để tự động tạo database khi container chạy lần đầu tiên.

docker build -t custom-postgres .

docker run --name my-postgres -e POSTGRES\_USER=postgres -e POSTGRES\_PASSWORD=sapassword -e POSTGRES\_DB=postgres -d custom-postgres

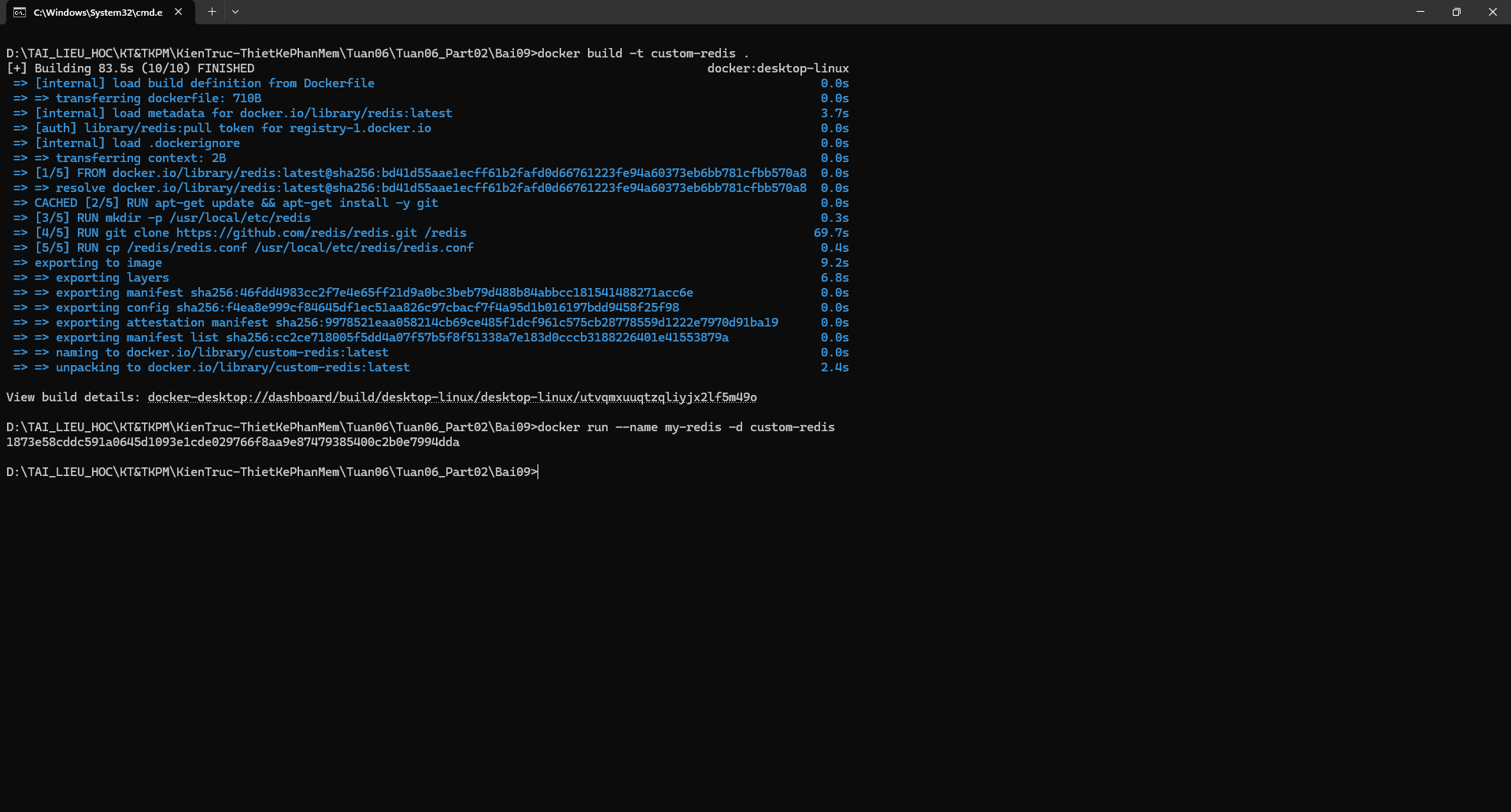
docker exec -it my-postgres psql -U postgres -c "\list"



**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.

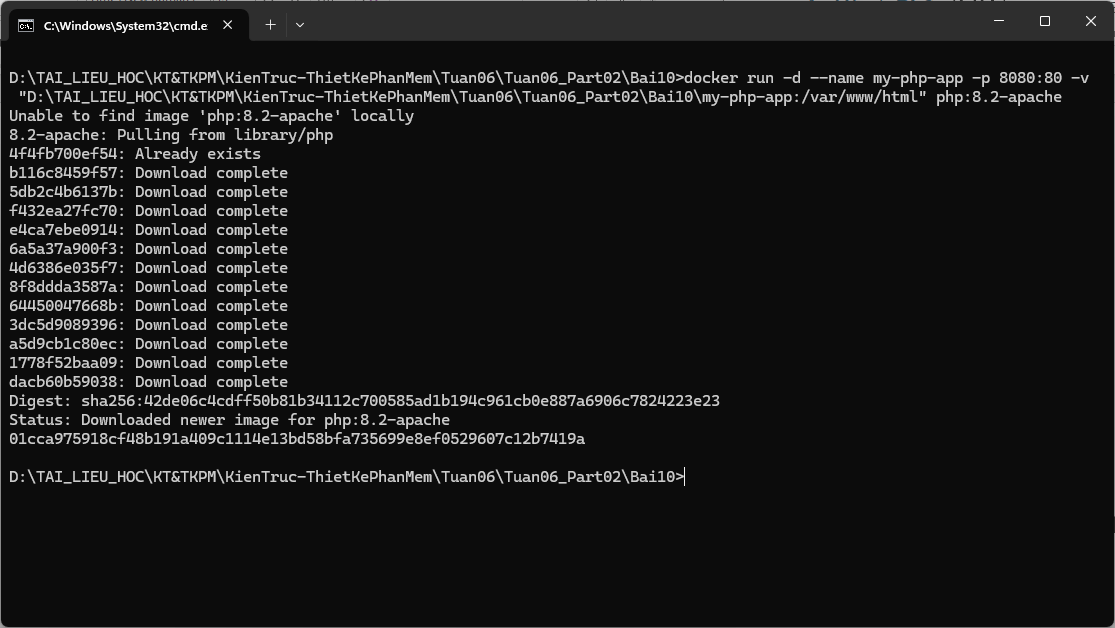


**Bài 10:** Chạy ứng dụng PHP với Apache

Yêu cầu: Viết Dockerfile để chạy một ứng dụng PHP đơn giản (php:8.2-apache).

Mount mã nguồn từ máy host vào container.

docker run -d --name my-php-app -p 8080:80 -v "D:\TAI\_LIEU\_HOC\KT&TKPM\KienTruc-ThietKePhanMem\Tuan06\Tuan06\_Part02\Bai10\my-php-app:/var/www/html" php:8.2-apache



http://localhost:8080/info.php

