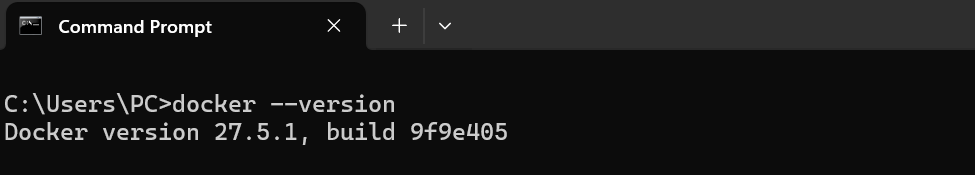
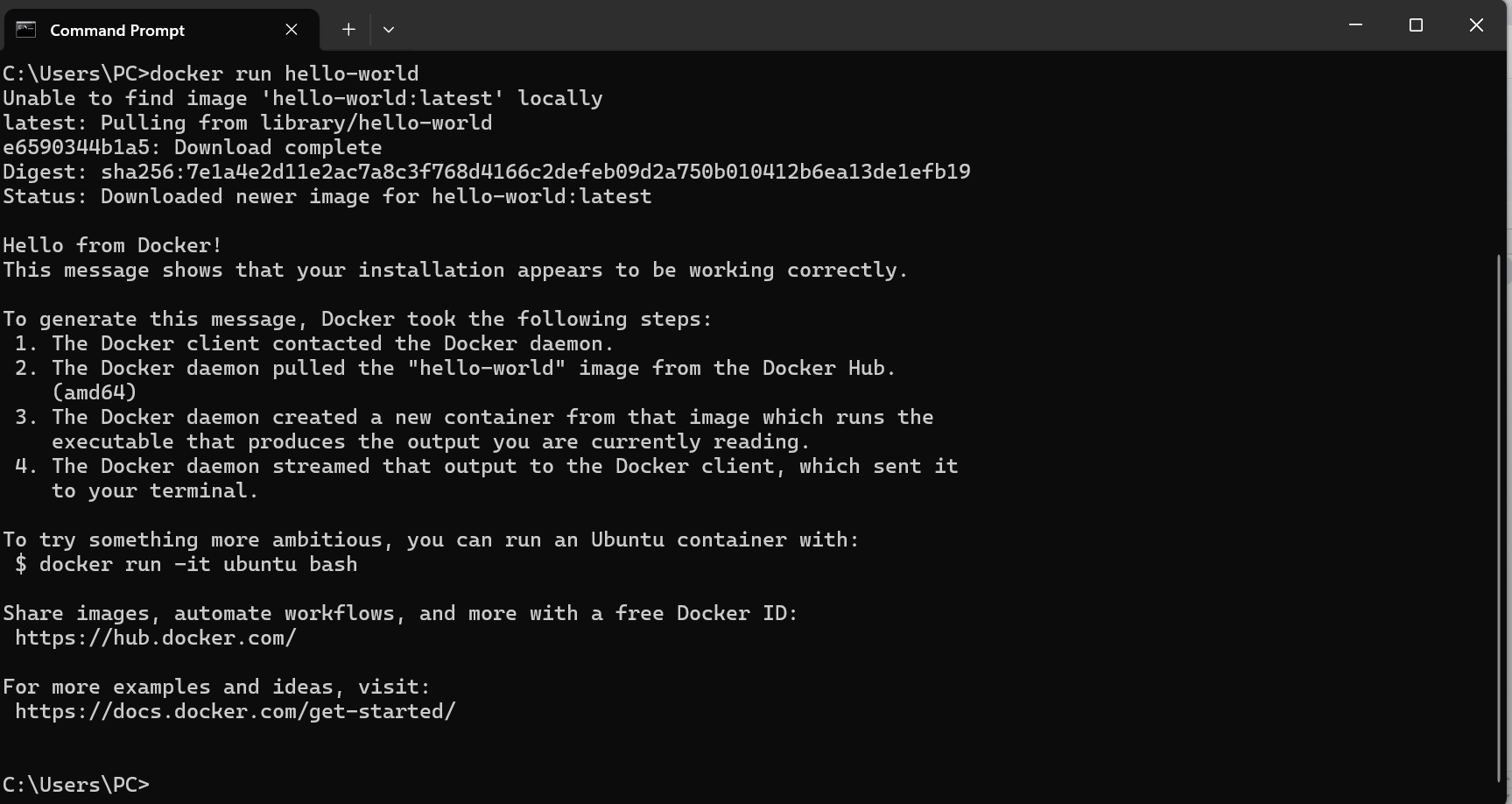
**Phần 1: Các lệnh cơ bản thao tác với Docker**

1. docker --version

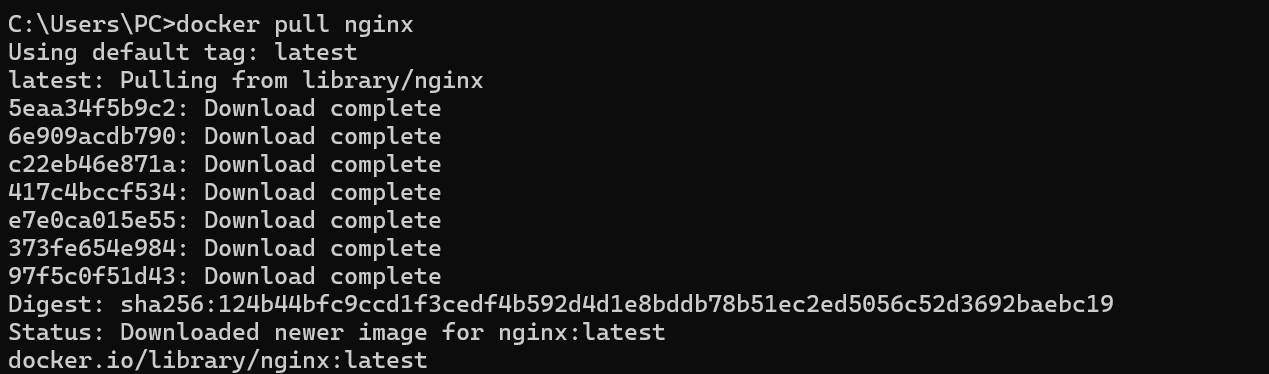


Kiểm tra phiên bản docker. Hiện tại là version 27.5.1

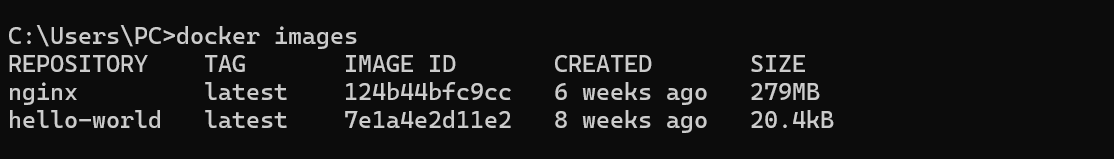
1. docker run hello-world



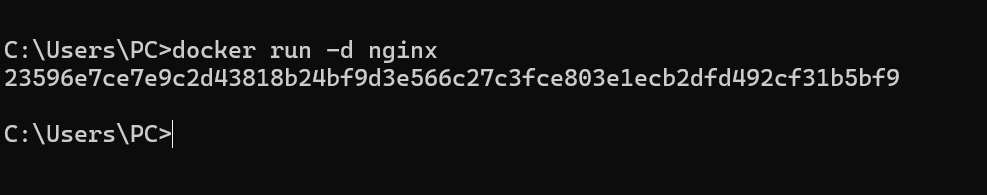
1. docker pull nginx



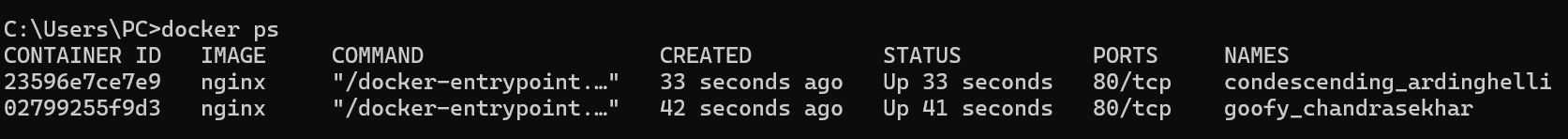
1. docker images

Kiểm tra xem có những images nào trên máy

1. docker run -d nginx

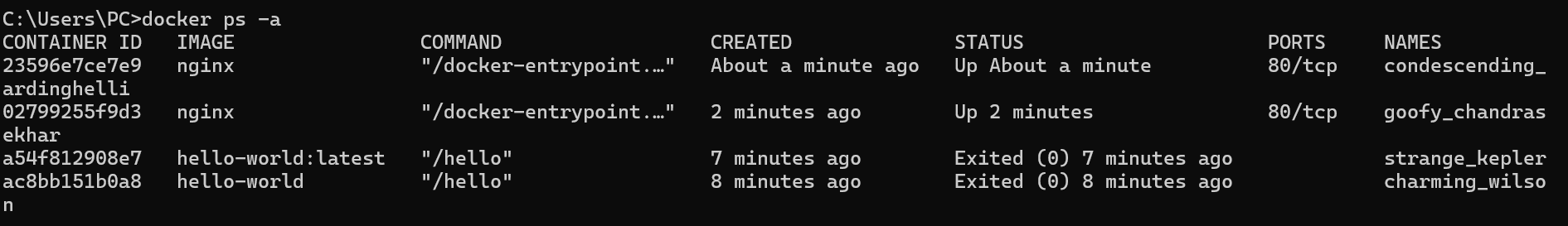


1. docker ps



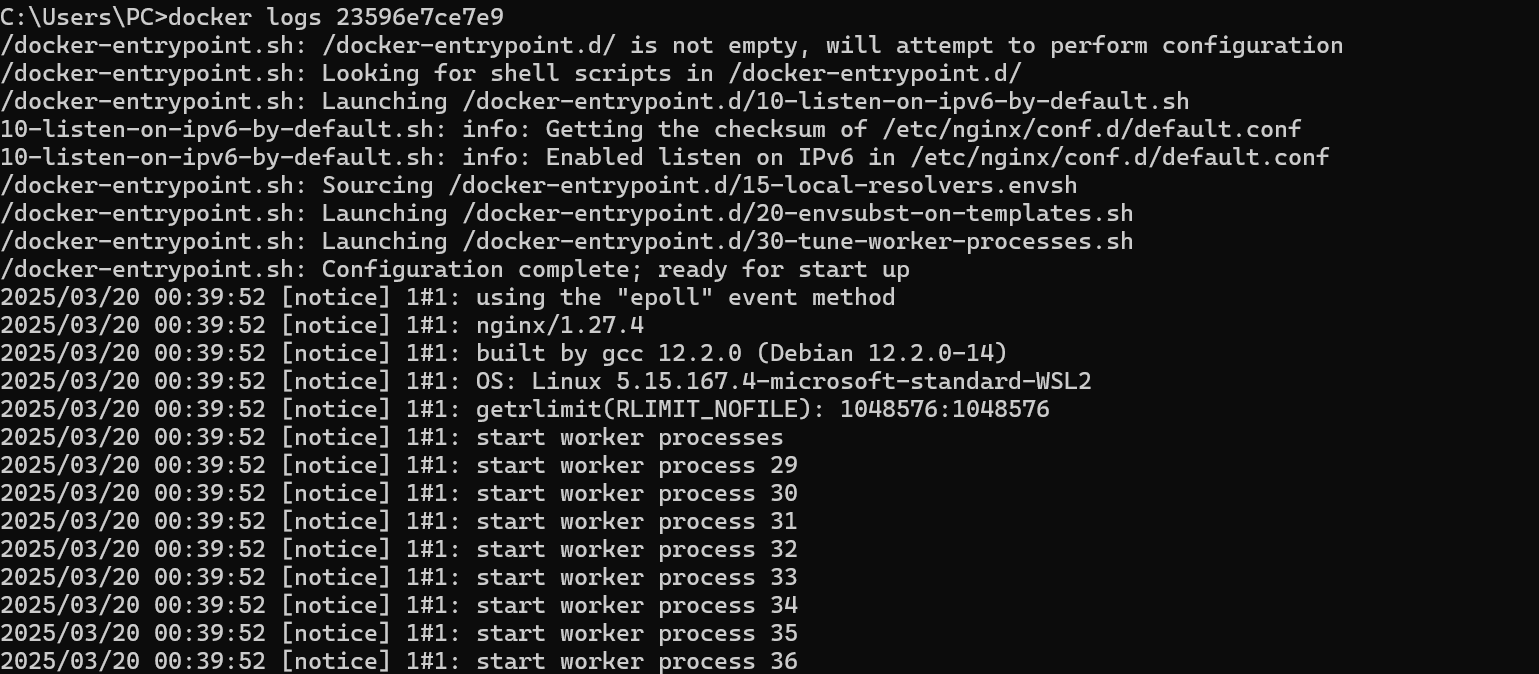
Liệt kê các container đang chạy trong docker

1. docker ps -a

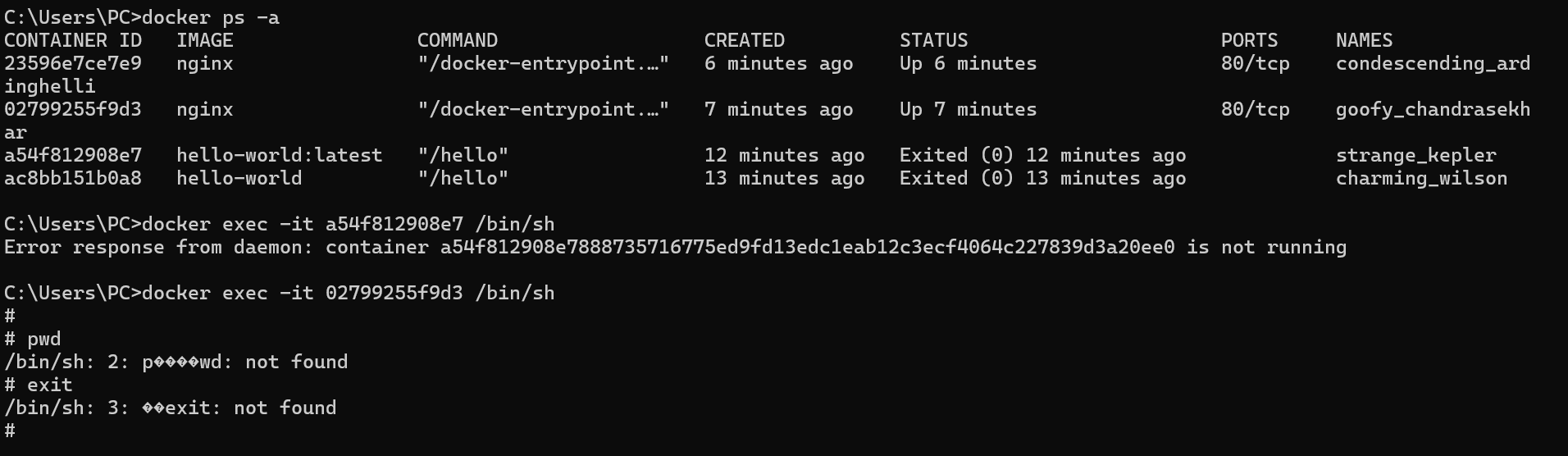


Dòng lệnh để xem những container nào đang chạy và đã dừng

1. docker logs <container\_id>

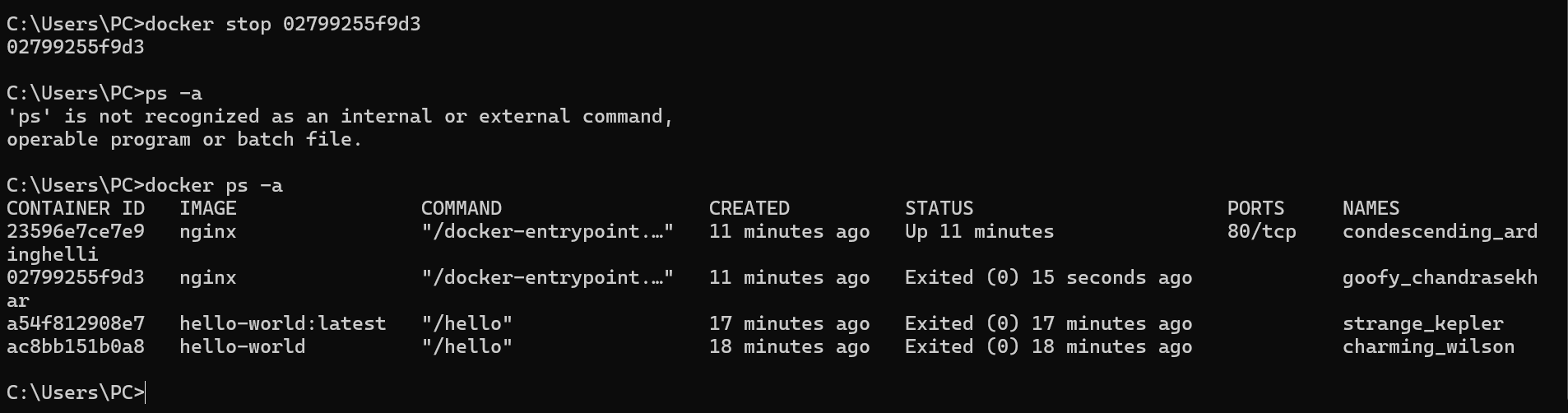


1. docker exec -it <container\_id> /bin/sh

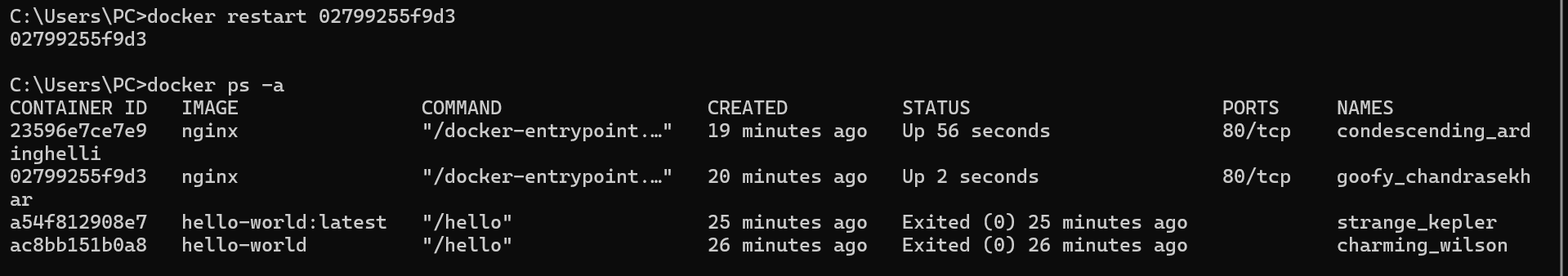


Tương tác với container

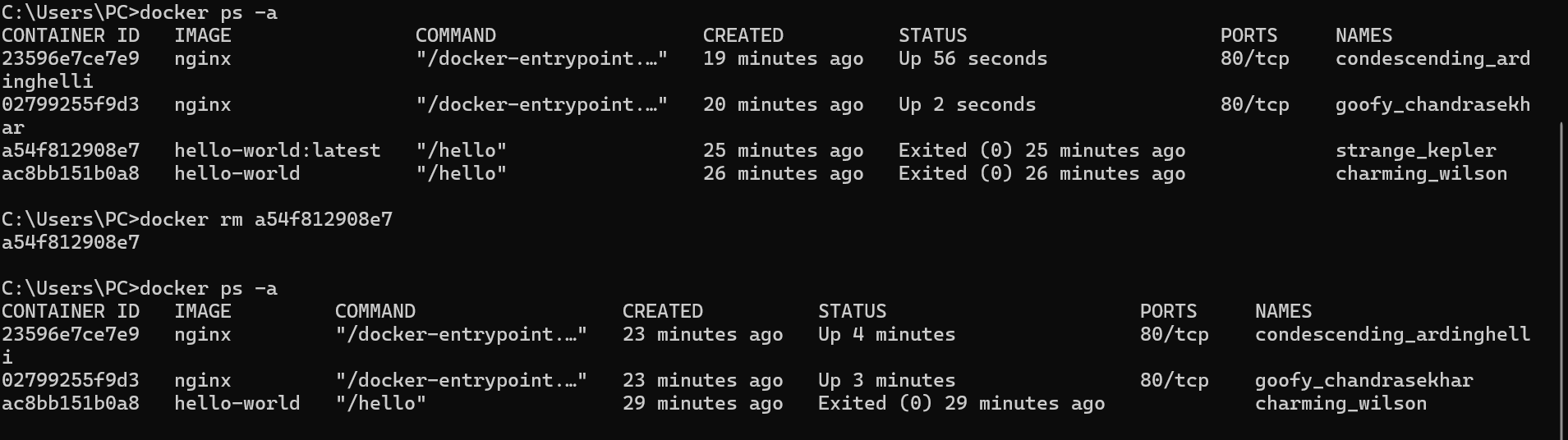
1. docker stop <container\_id>

Dừng docker đang chạy

1. docker restart <container\_id>

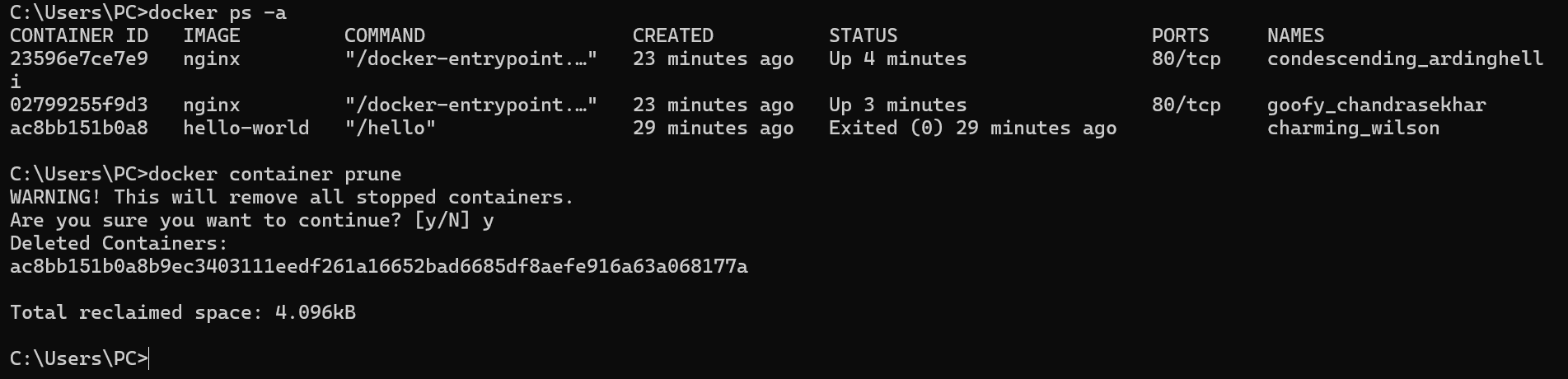


1. docker rm <container\_id>



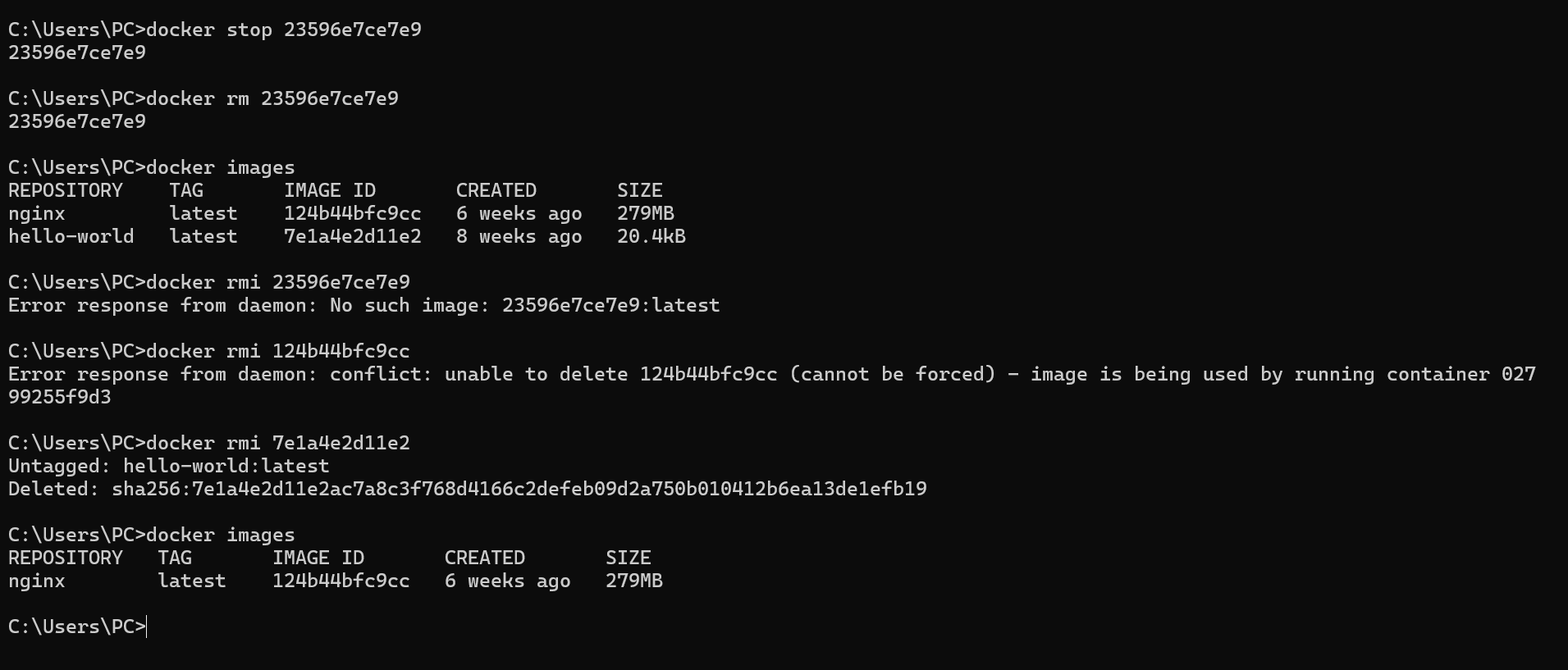
Xóa 1 container đã dừng

1. docker container prune

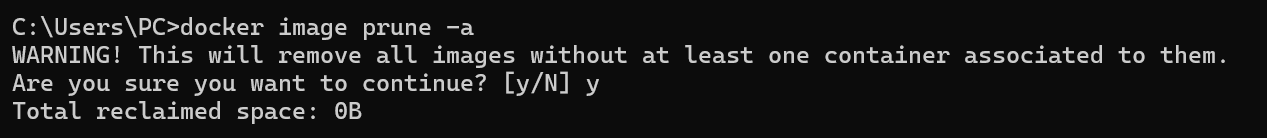


Dừng các container không chạy để giải phóng tài nguyên

1. docker rmi <image\_id>

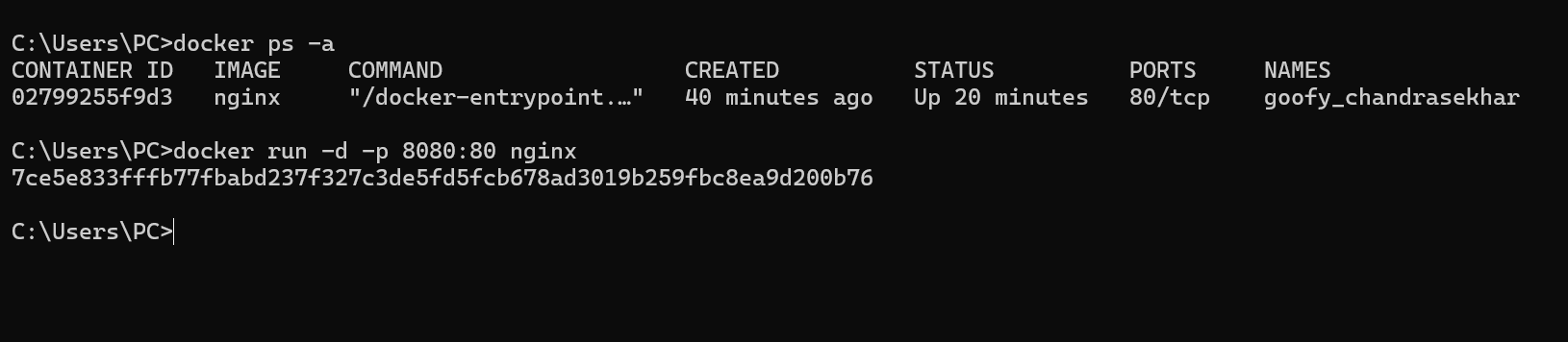
Remove img

1. docker image prune -a

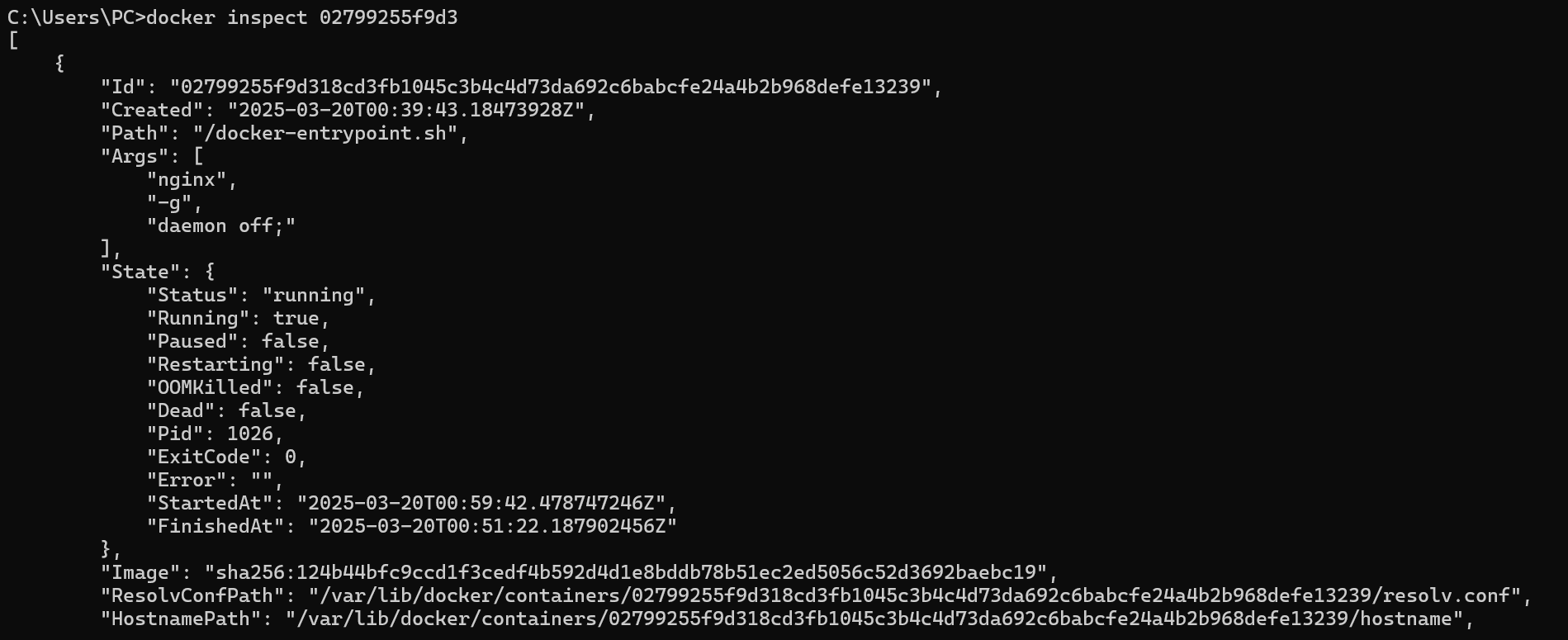


Dừng các img không hoạt động để giải phóng tài nguyên

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



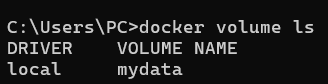
1. docker inspect <container\_id>



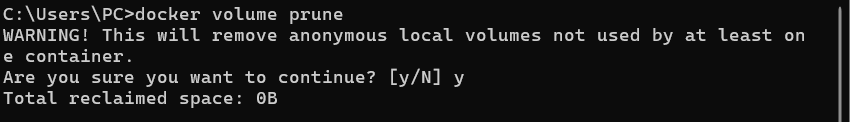
1. docker run -d -v mydata:/data nginx



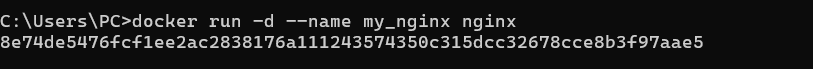
1. docker volume ls



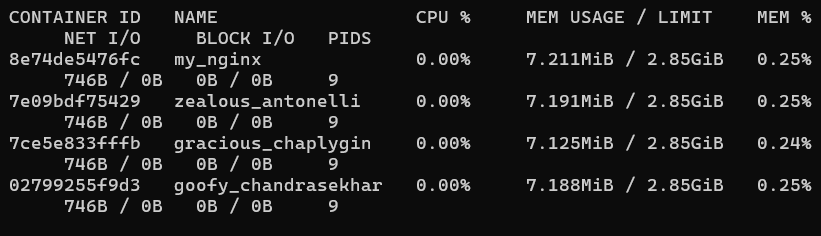
1. docker volume prune



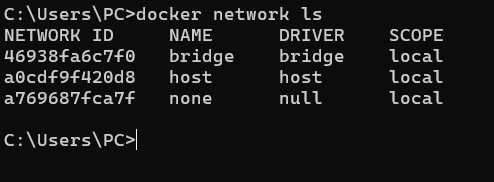
1. docker run -d --name my\_nginx nginx



1. docker stats



1. docker network ls



1. docker network create my\_network

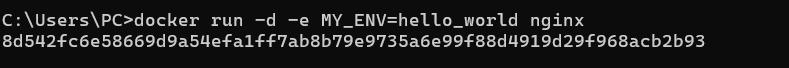


25 docker run -d --network my\_network --name my\_container nginx 

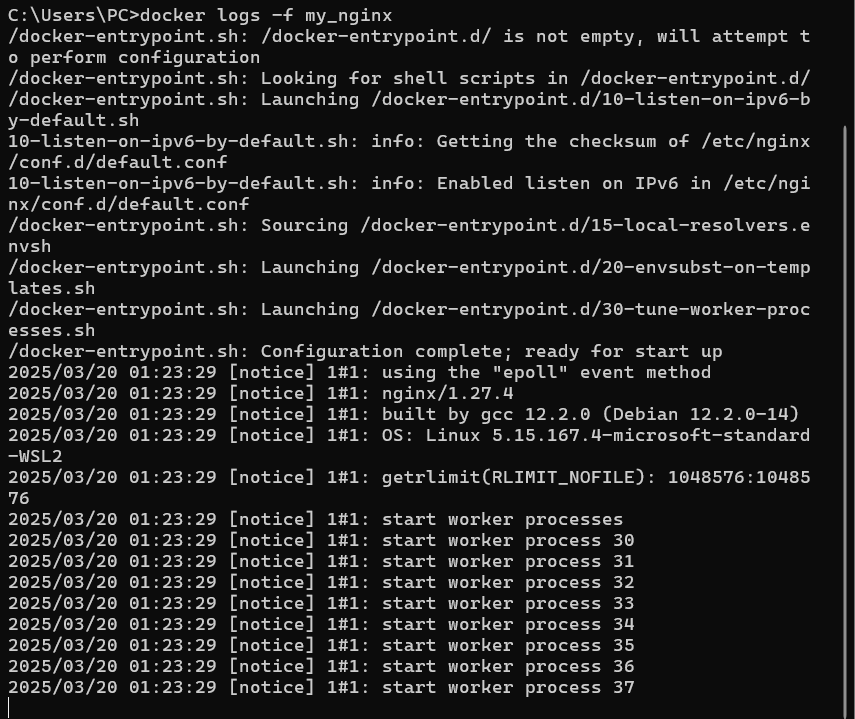
26 docker network connect my\_network my\_nginx



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

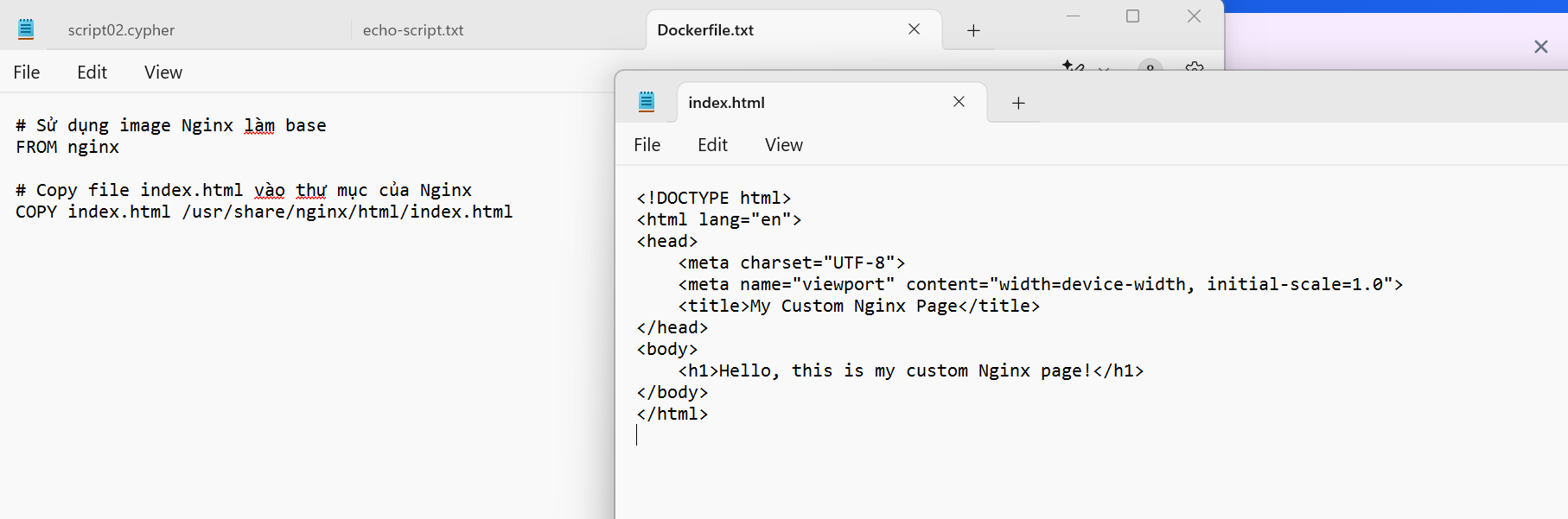


1. docker logs -f my\_nginx

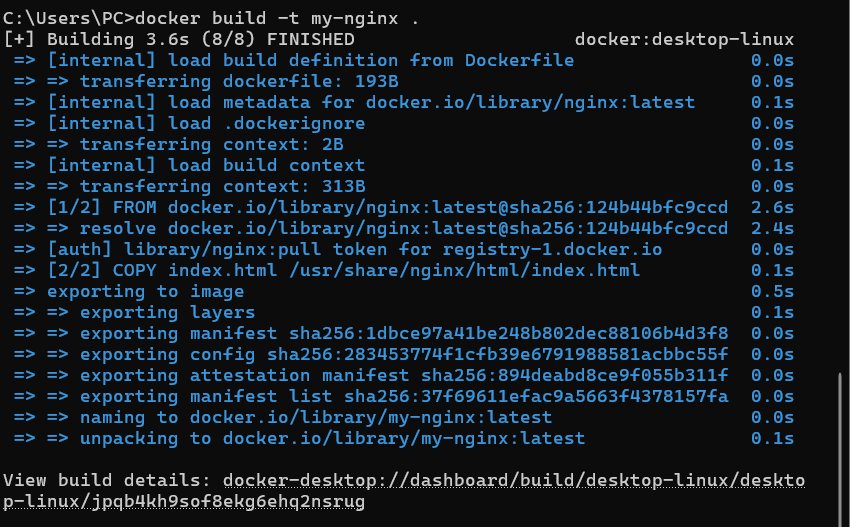


29 FROM nginx

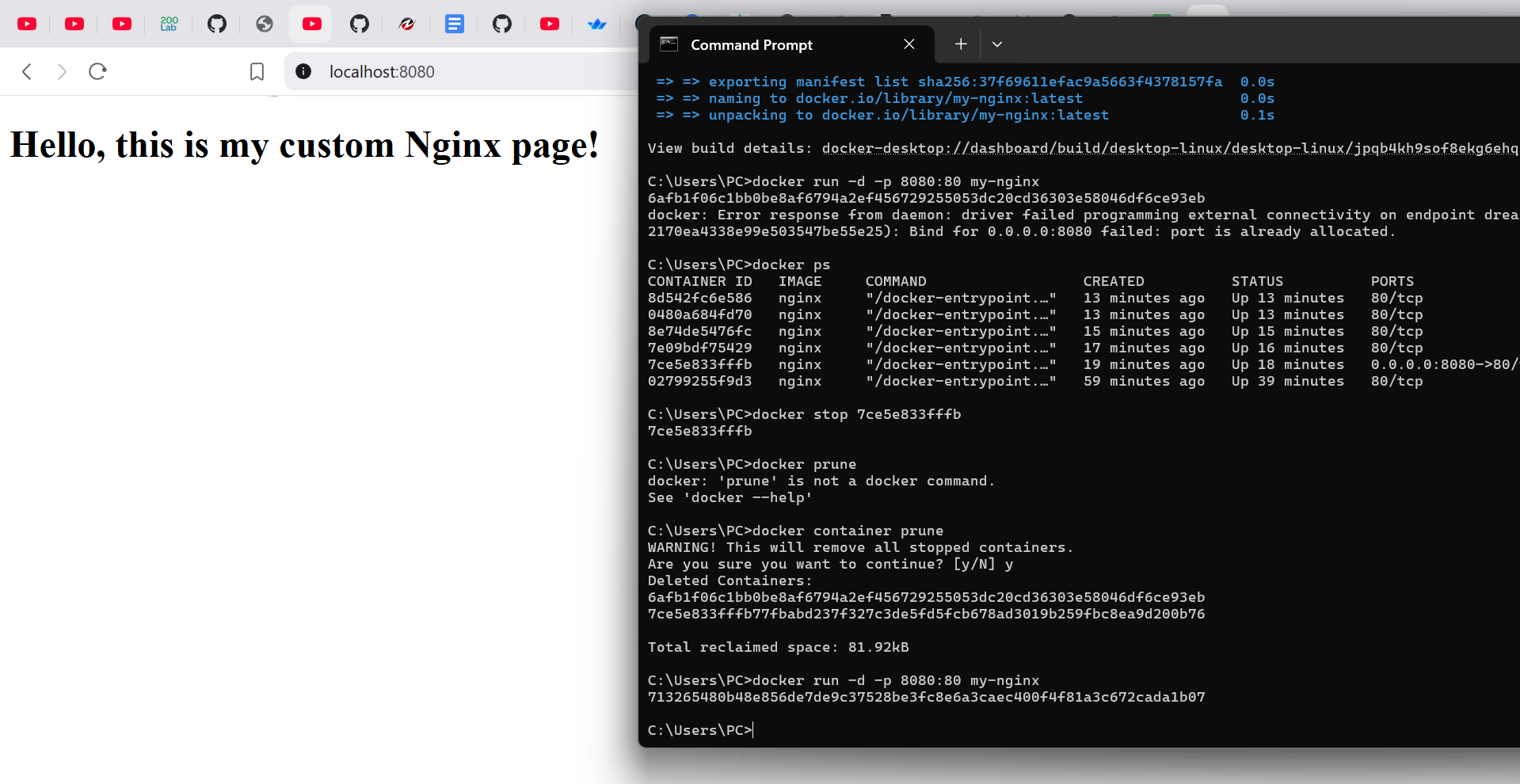
COPY index.html /usr/share/nginx/html/index.html



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



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



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.

Bài 4: Tạo Dockerfile chạy một trang web tĩnh bằng Nginx

📌 Yêu cầu:

Tạo một file index.html đơn giản.

Sử dụng nginx:latest để phục vụ trang web.

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

📌 Yêu cầu:

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.

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 dòng lệnh:

dockerfile

Copy

Edit

ENV APP\_ENV=development

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.

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.