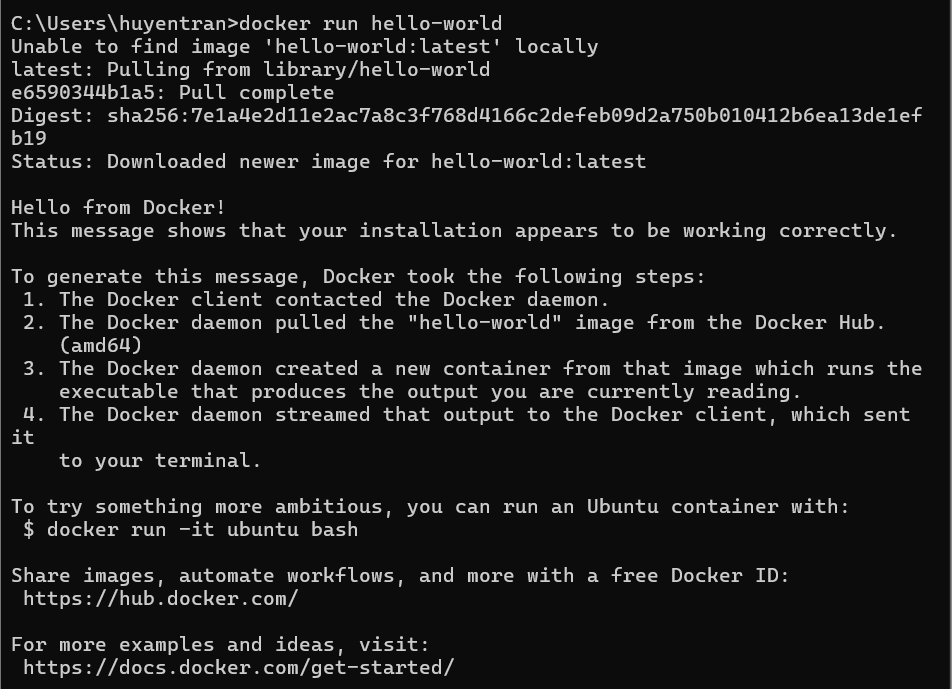
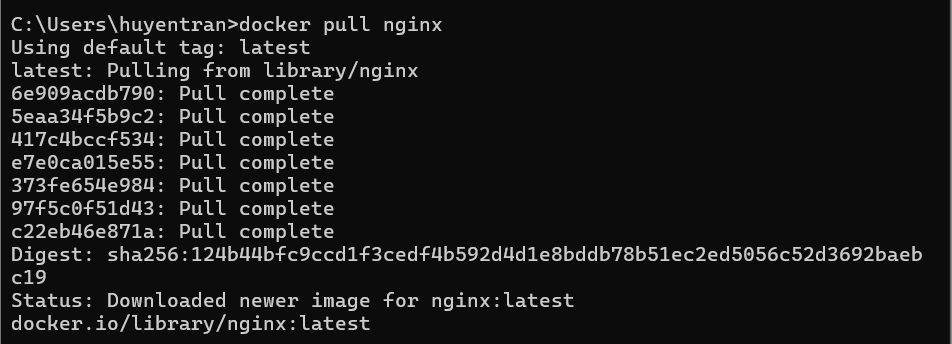
**Các Lệnh Docker Cơ Bản**

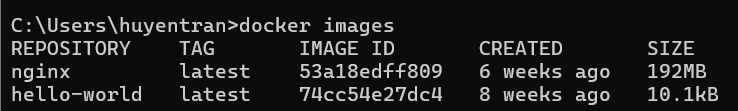
1. **docker --version**: Hiển thị phiên bản Docker hiện tại mà bạn đang sử dụng.
2. **docker run hello-world**: Chạy một container từ image hello-world để kiểm tra xem Docker có hoạt động đúng không. Docker sẽ tải image về và in thông báo chào mừng nếu mọi thứ hoạt động tốt



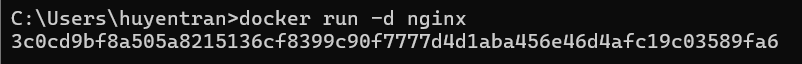
1. **docker pull nginx**: Tải image nginx từ Docker Hub về máy tính của bạn. Đây là image phổ biến của web server Nginx.



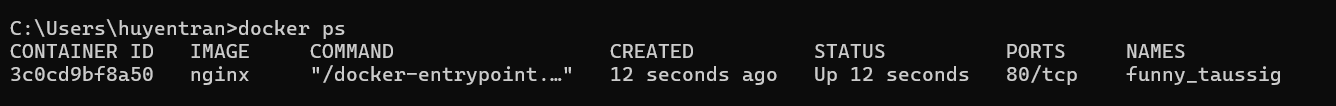
1. **docker images**: Liệt kê tất cả các image mà bạn đã tải xuống từ Docker Hub hoặc tạo ra từ các container.



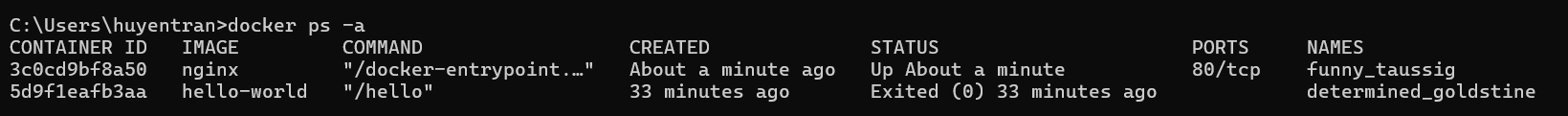
1. **docker run -d nginx**: Chạy container từ image nginx trong chế độ nền (-d là detached mode, tức là không tương tác trực tiếp với container).



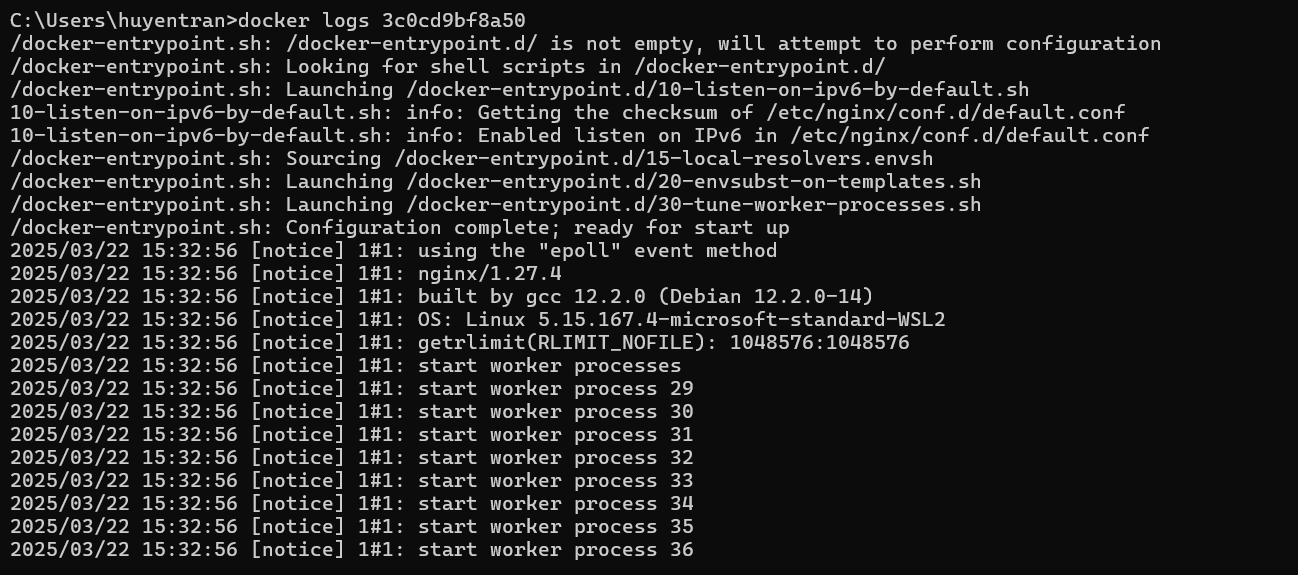
1. **docker ps**: Liệt kê các container đang chạy.



1. **docker ps –a**: Liệt kê tất cả các container (bao gồm các container đã dừng).

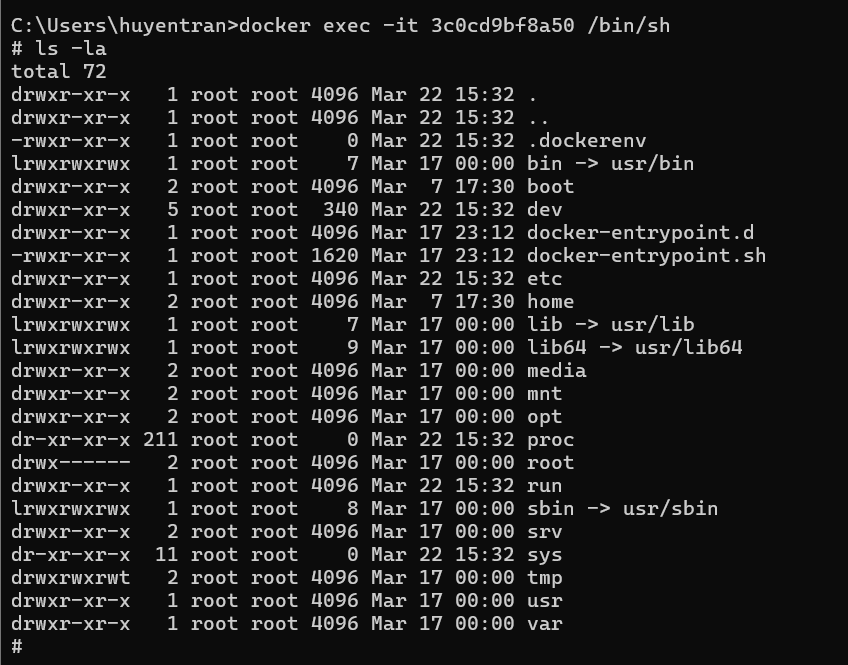


1. **docker logs <container\_id>**: Xem log của một container dựa trên ID của nó, hữu ích để kiểm tra lỗi hoặc thông tin container.



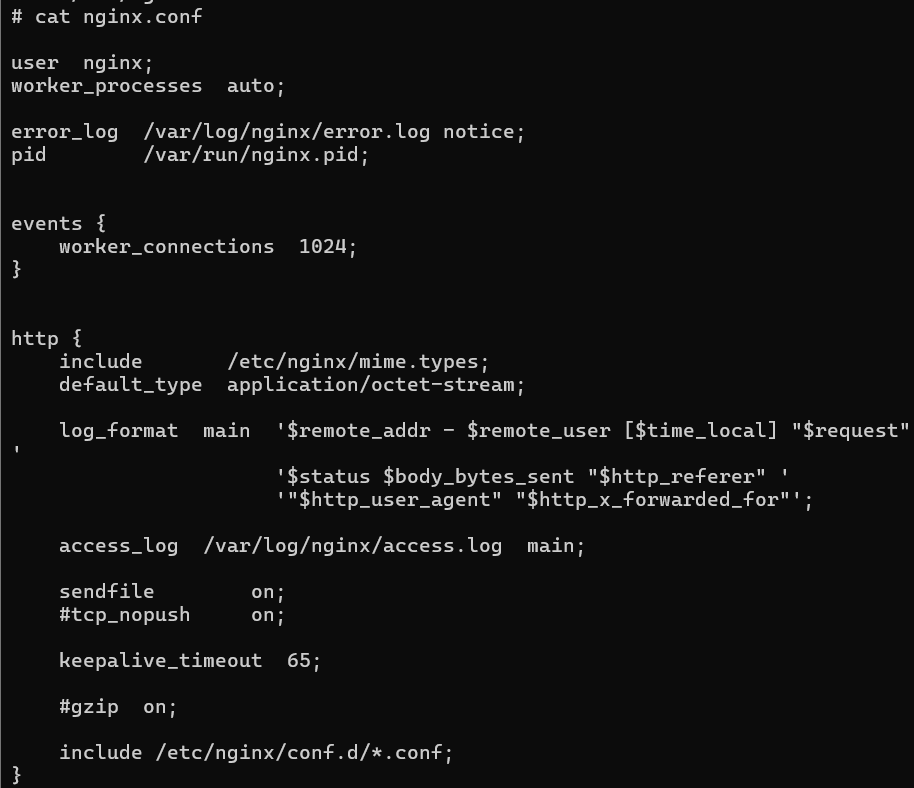
1. **docker exec -it <container\_id> /bin/sh**: Mở terminal (shell) bên trong một container đang chạy. -it giúp bạn tương tác với container, và /bin/sh là shell mà bạn có thể sử dụng trong container (nếu container hỗ trợ).

Xem tất cả file và thư mục hiện tại:

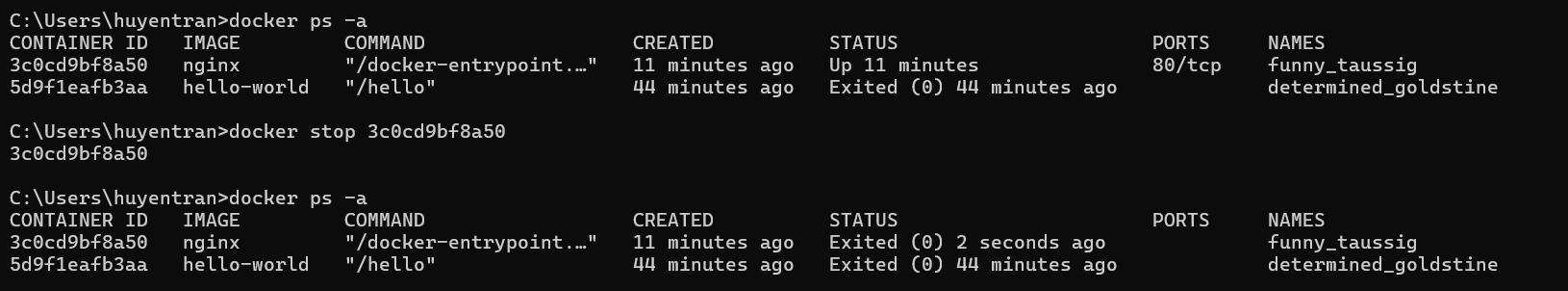


Chuyển đến thư mục cấu hình Nginx (nếu đang chạy một container Nginx):

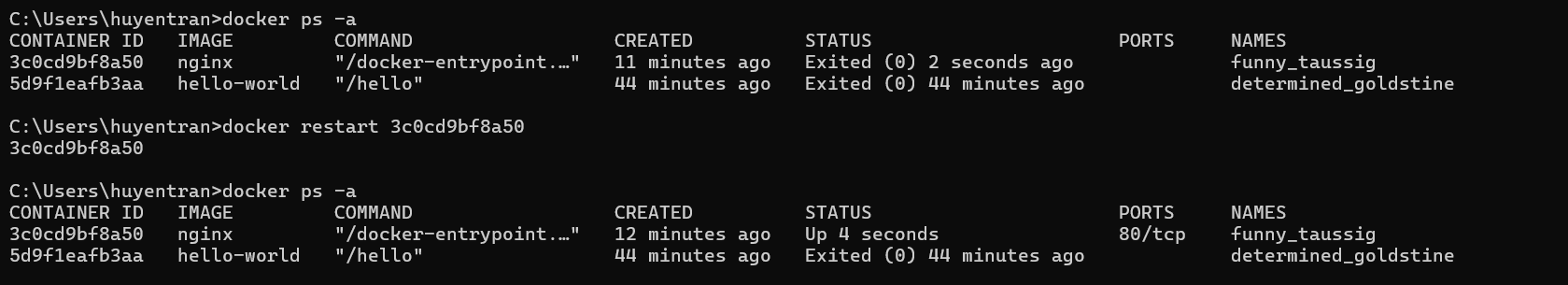
Xem nội dung file: 



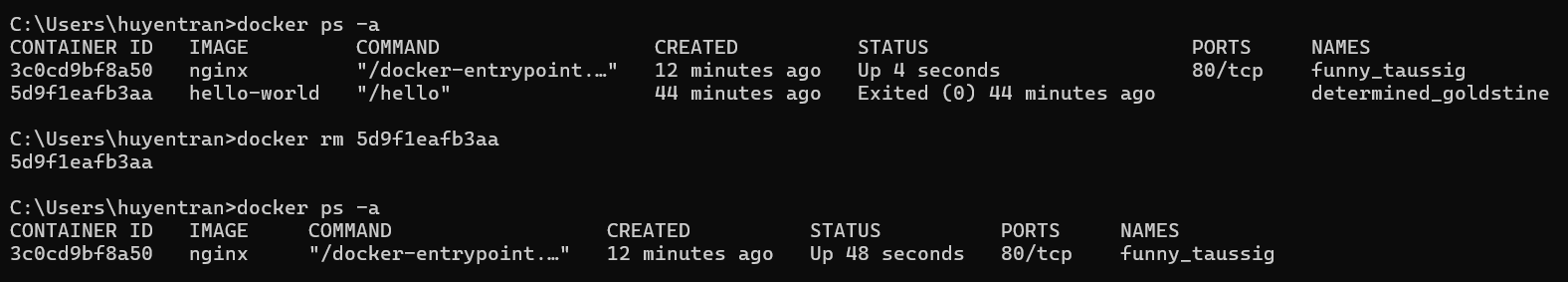
1. **docker stop <container\_id>**: Dừng một container đang chạy.



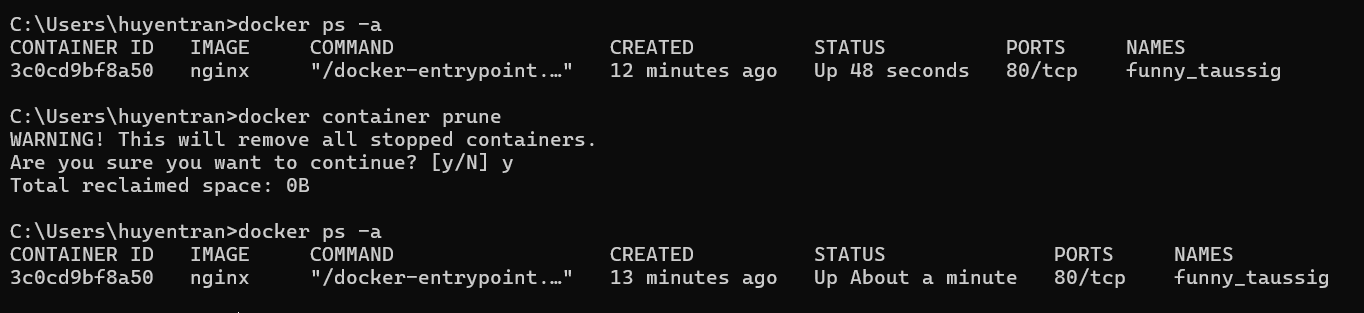
1. **docker restart <container\_id>**: Khởi động lại một container đang chạy.



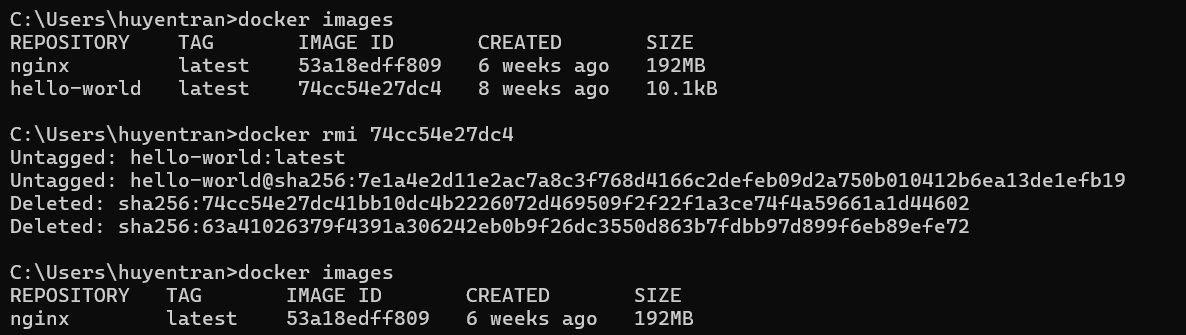
1. **docker rm <container\_id>**: Xóa một container đã dừng (không thể xóa container đang chạy).



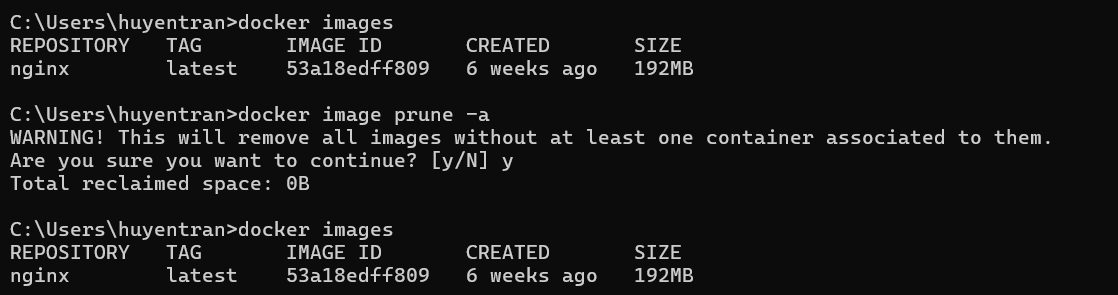
1. **docker container prune**: Xóa tất cả các container đã dừng. Đây là một cách để giải phóng không gian ổ đĩa.



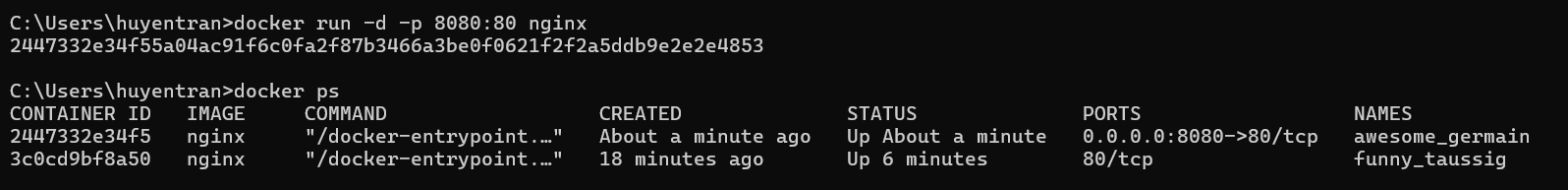
1. **docker rmi <image\_id>**: Xóa một image Docker khỏi hệ thống của bạn. Bạn phải chắc chắn rằng không có container nào đang sử dụng image này trước khi xóa.



1. **docker image prune -a:** Xóa tất cả các image không được sử dụng (không có container nào liên kết với chúng).



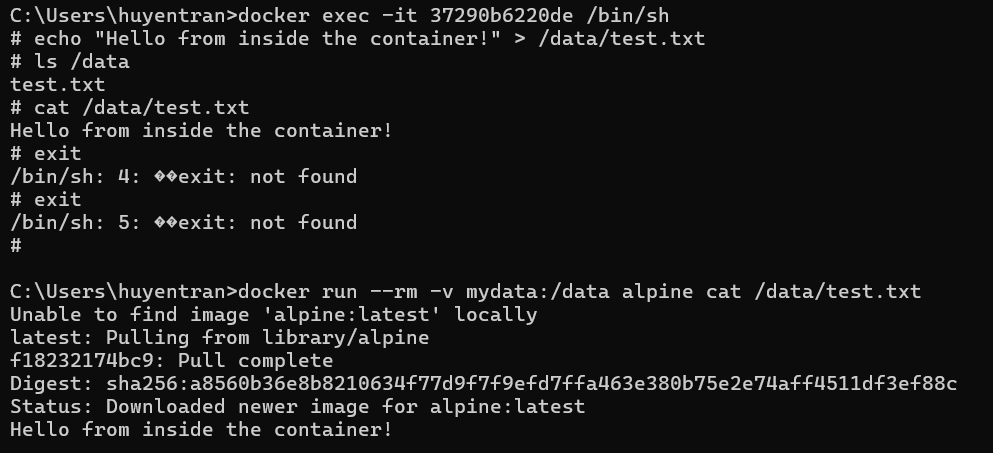
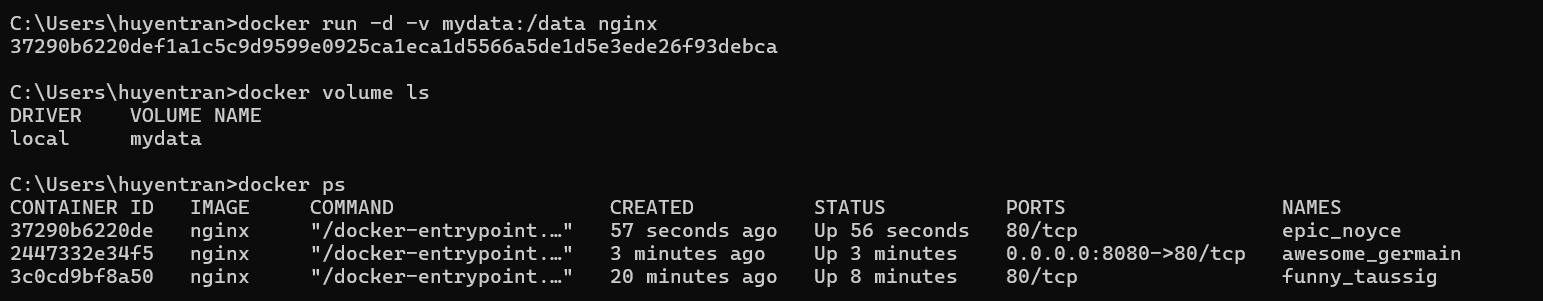
1. **docker run -d -p 8080:80 nginx:** Chạy một container Nginx trong chế độ nền và ánh xạ cổng 8080 trên máy chủ của bạn đến cổng 80 trong container (để truy cập Nginx qua cổng 8080).



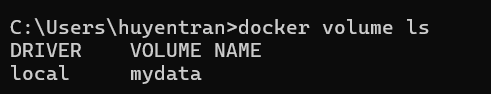
1. **docker inspect <container\_id>:** Hiển thị thông tin chi tiết về một container, bao gồm cấu hình, thông số kỹ thuật, và trạng thái.



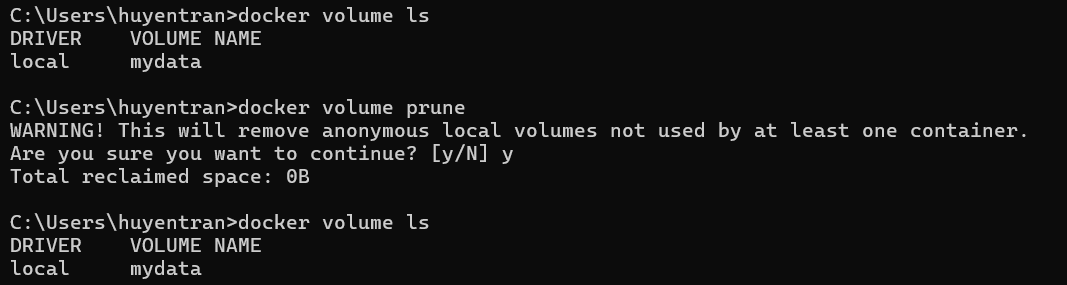
1. **docker run -d -v mydata:/data nginx:** Chạy một container Nginx trong chế độ nền và ánh xạ một volume có tên mydata vào thư mục /data trong container. Điều này giúp lưu trữ dữ liệu ngoài container, ngăn ngừa mất dữ liệu khi container dừng.



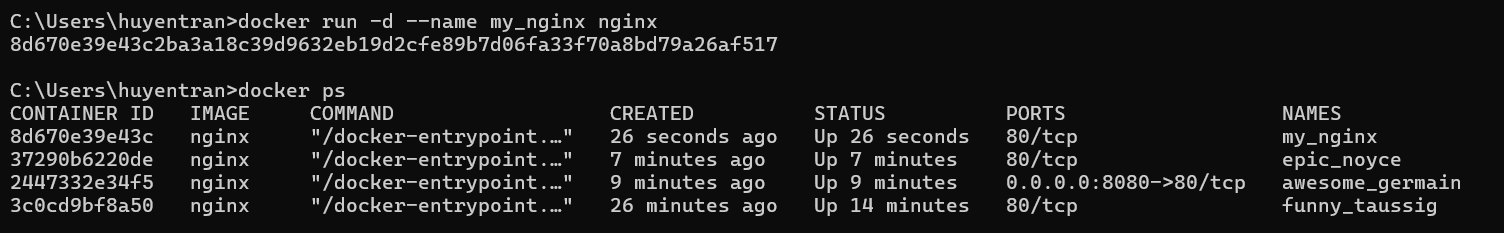
1. **docker volume ls:** Liệt kê tất cả các volumes Docker hiện có trên hệ thống.



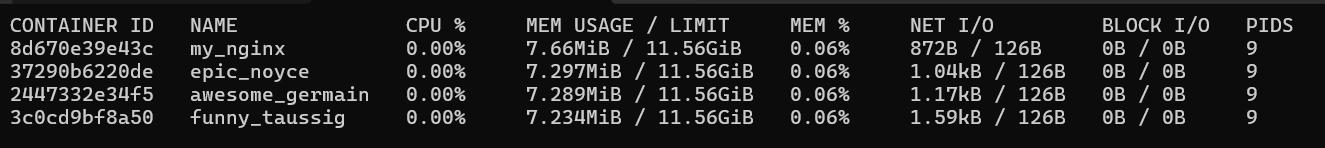
1. **docker volume prune:** Xóa tất cả các volume không còn được sử dụng bởi bất kỳ container nào.



1. **docker run -d --name my\_nginx nginx:** Chạy một container Nginx trong chế độ nền và gán tên my\_nginx cho container.



1. **docker stats:** Hiển thị thông tin thống kê thời gian thực về tài nguyên sử dụng của các container (CPU, bộ nhớ, mạng, v.v.).



Giải thích từng cột:

CONTAINER ID: Mã định danh duy nhất của container.

NAME: Tên container (do bạn đặt hoặc Docker tự đặt).

CPU %: Phần trăm CPU mà container đang sử dụng.

MEM USAGE / LIMIT: Bộ nhớ đang sử dụng / Giới hạn bộ nhớ tối đa.

MEM %: Phần trăm bộ nhớ được sử dụng.

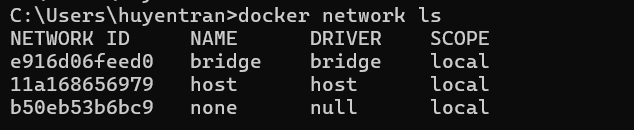
NET I/O: Lưu lượng dữ liệu truyền qua mạng (In/Out).

BLOCK I/O: Dữ liệu được đọc/ghi từ bộ nhớ (disk).

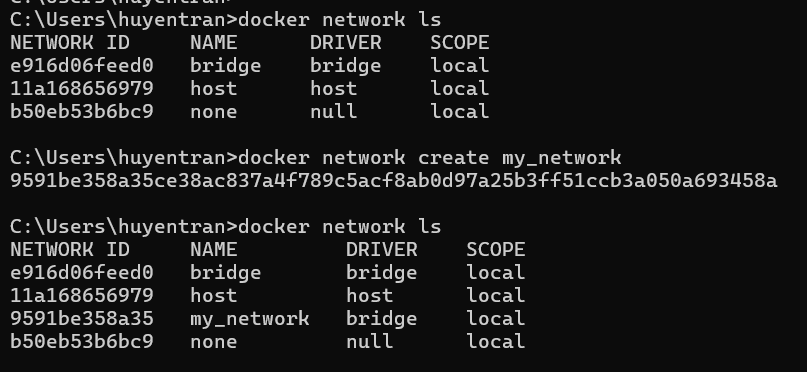
PIDS: Số lượng tiến trình đang chạy bên trong container.

Thoát bằng CTRL + C.

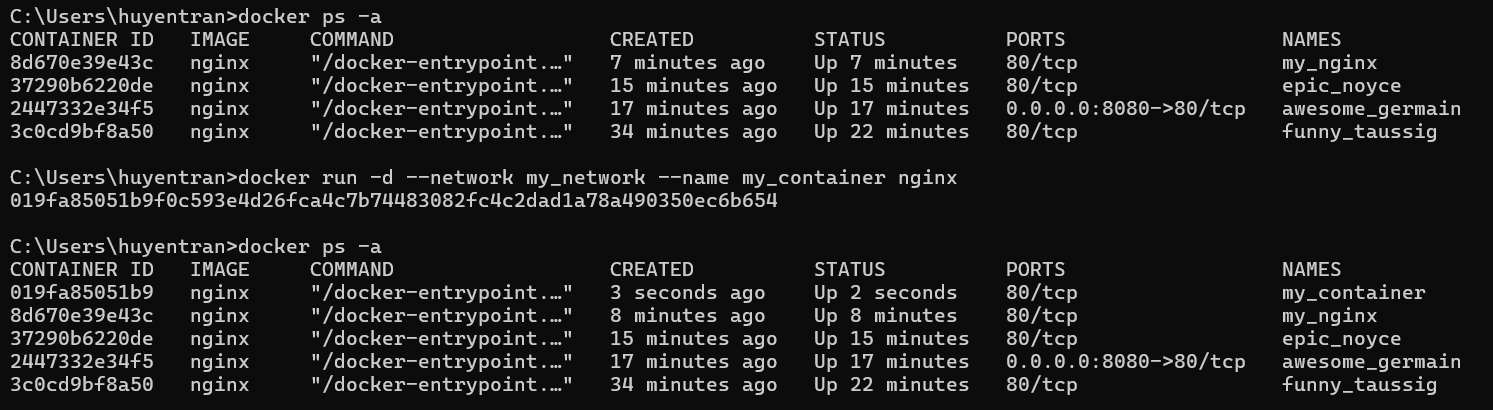
1. **docker network ls:** Liệt kê tất cả các mạng Docker hiện có.



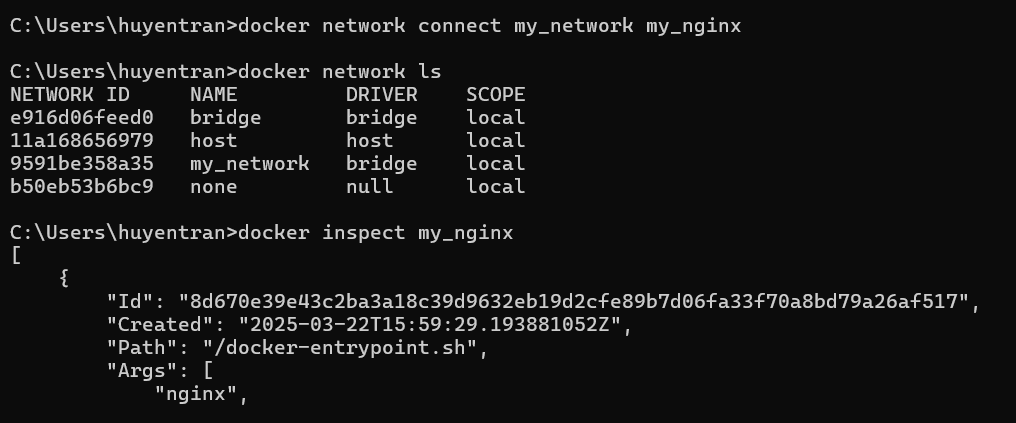
1. **docker network create my\_network:** Tạo một mạng Docker mới có tên my\_network.



1. **docker run -d --network my\_network --name my\_container nginx:** Chạy một container Nginx trong chế độ nền và kết nối nó vào mạng my\_network. Container này sẽ có tên là my\_container.

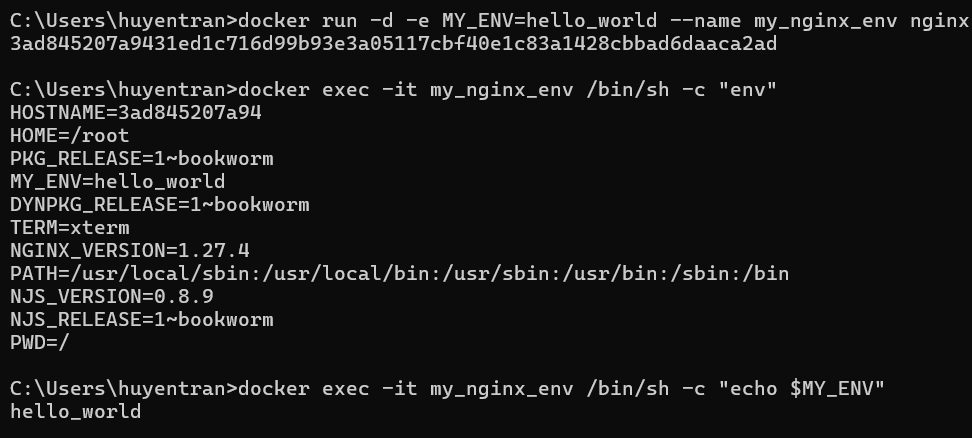


1. **docker network connect my\_network my\_nginx:** Kết nối một container (trong ví dụ này là my\_nginx) vào một mạng đã tạo trước đó (ở đây là my\_network).

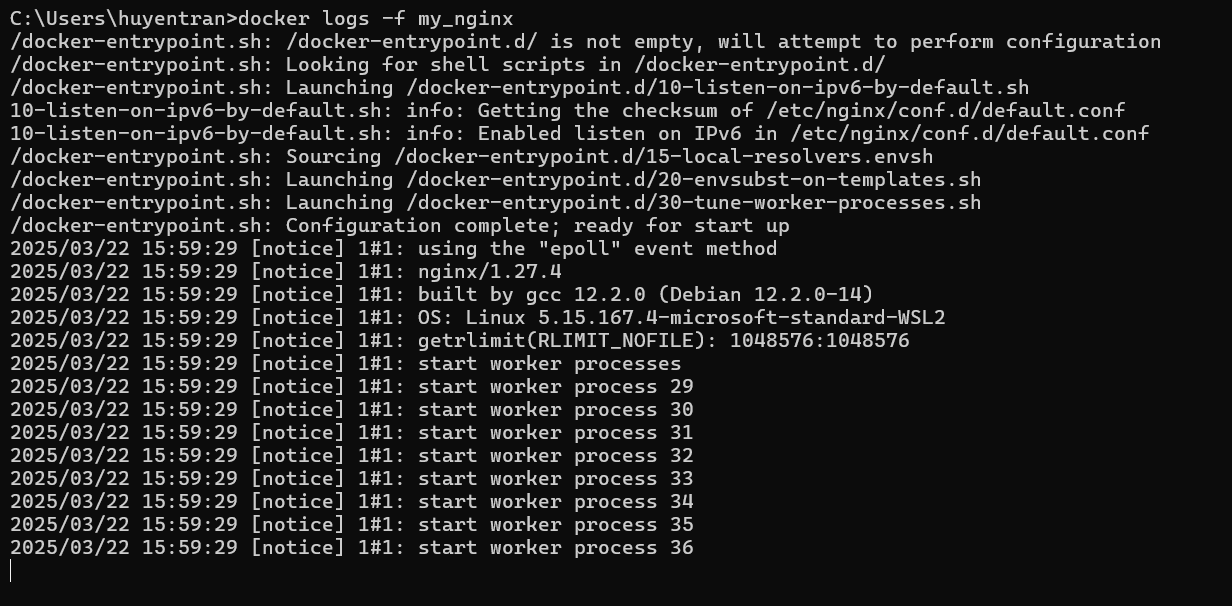


Nếu muốn ngắt kết nối container my\_nginx khỏi mạng my\_network: docker network disconnect my\_network my\_nginx

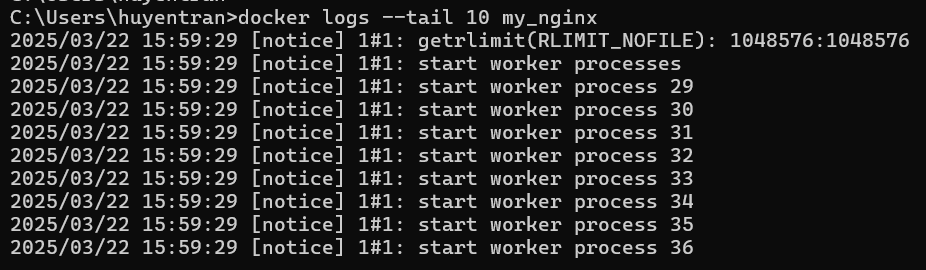
1. **docker run -d -e MY\_ENV=hello\_world nginx:** Chạy một container Nginx và thiết lập biến môi trường MY\_ENV với giá trị hello\_world trong container. Các biến môi trường có thể được sử dụng trong ứng dụng chạy trong container.



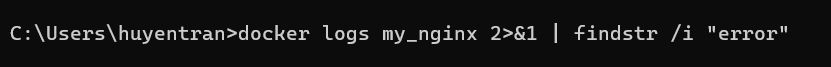
1. **docker logs -f my\_nginx:** Theo dõi log thời gian thực của container my\_nginx. Lệnh này sẽ tiếp tục xuất các log mới nhất khi có.



Xem một số dòng log cuối cùng (ví dụ 10 dòng): docker logs --tail 10 my\_nginx

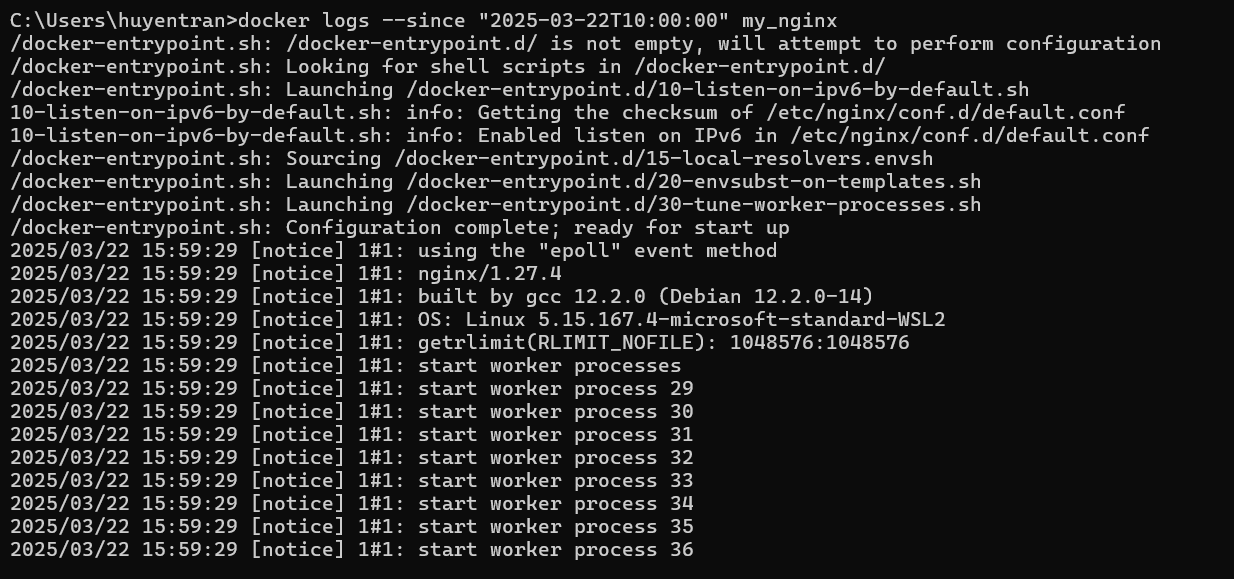


Chỉ xem lỗi (stderr) thay vì tất cả log: docker logs my\_nginx 2>&1 | findstr /i "error"



Lệnh của bạn không trả về gì, có nghĩa là không tìm thấy lỗi nào chứa từ "error" trong log của container my\_nginx.

Xem log từ một thời điểm cụ thể: docker logs --since "2025-03-22T10:00:00" my\_nginx

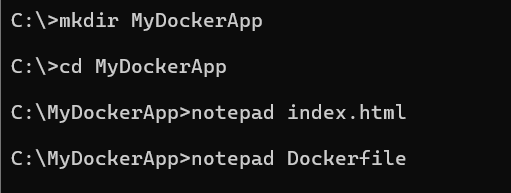


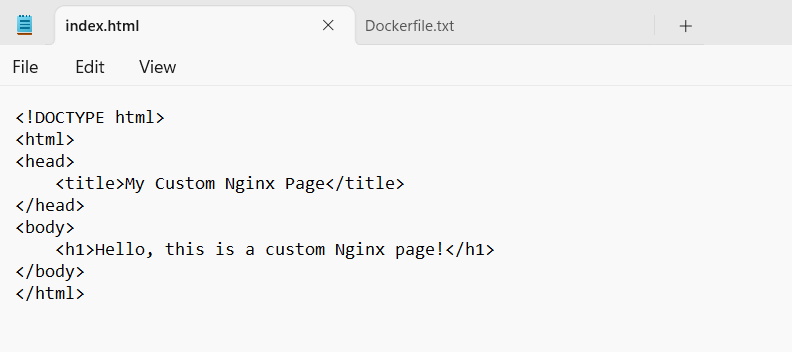
Xem log theo thời gian thực và thoát bằng CTRL + C.

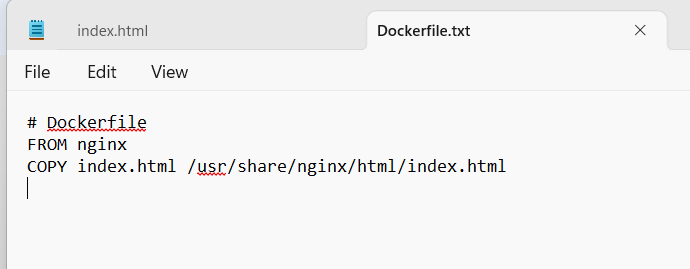
1. **Dockerfile**

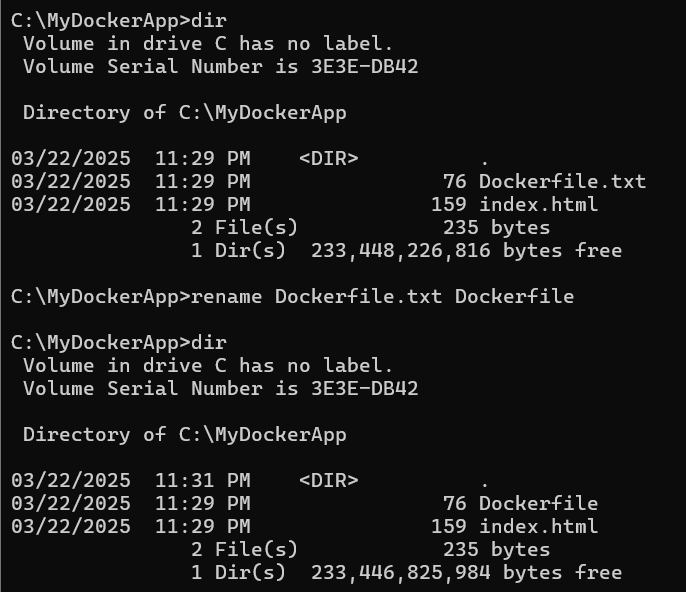
FROM nginx: Lệnh này trong Dockerfile chỉ định rằng image gốc cho Dockerfile này sẽ là nginx. Bạn có thể bắt đầu xây dựng một image mới từ Nginx.

COPY index.html /usr/share/nginx/html/index.html: Lệnh COPY sẽ sao chép file index.html từ máy chủ vào container Nginx, vào thư mục mà Nginx sử dụng để lưu trữ các file web (/usr/share/nginx/html/index.html).

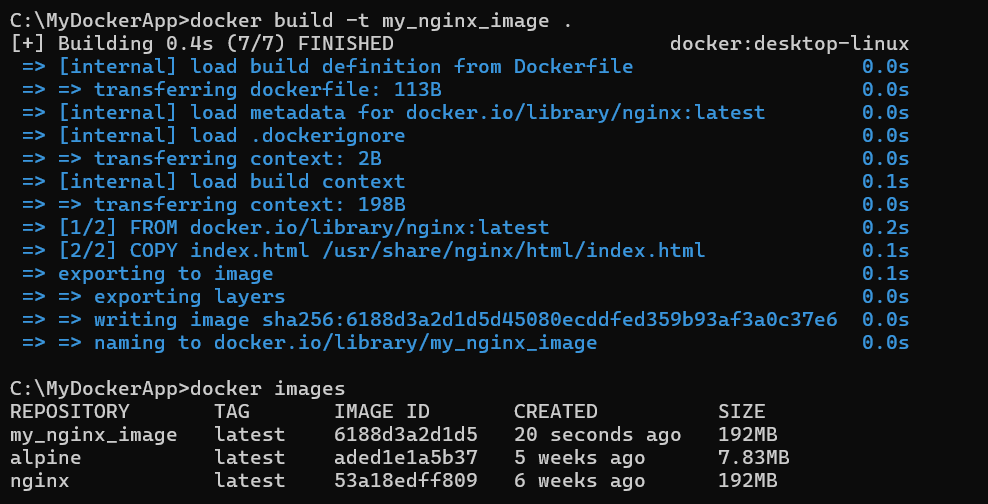




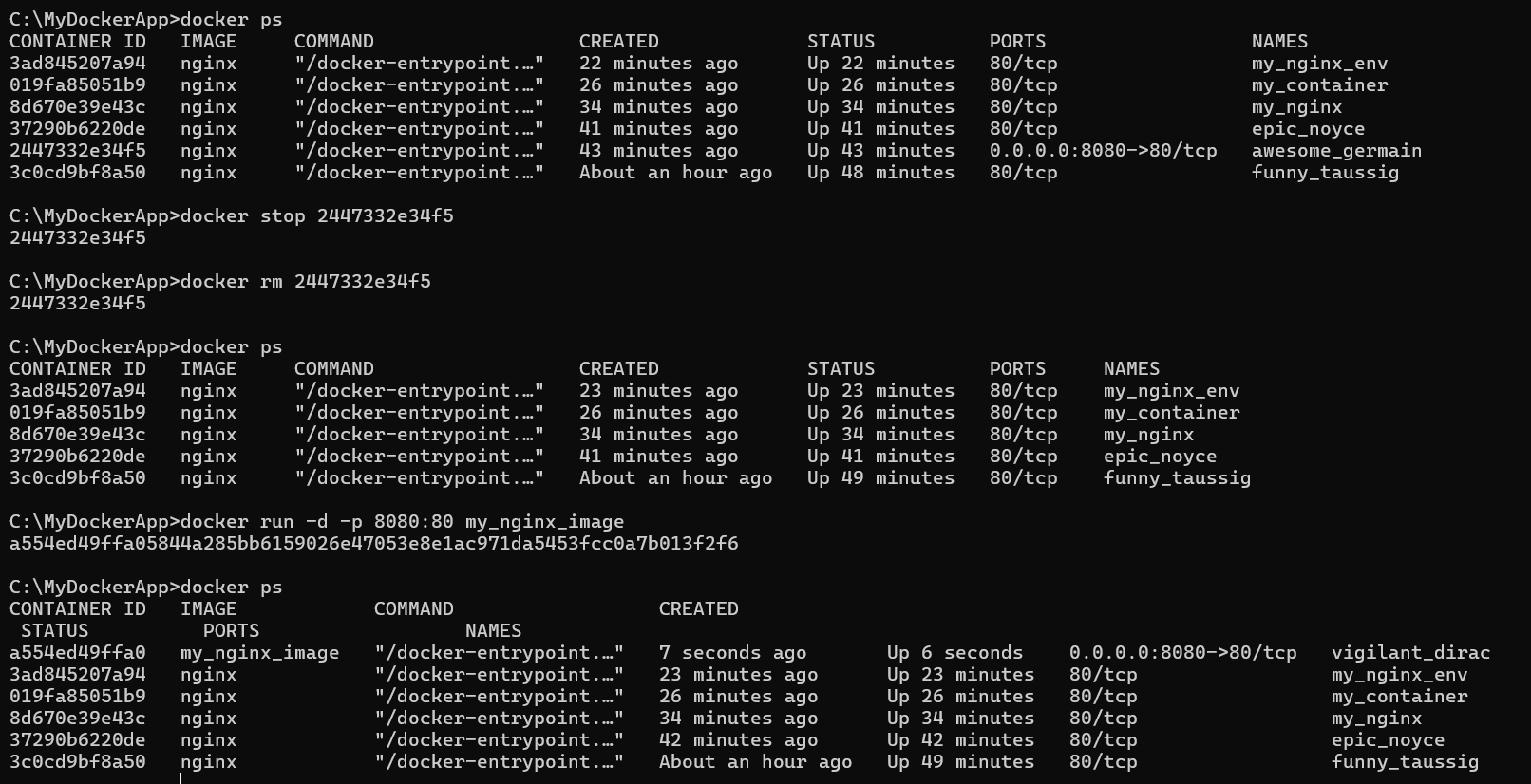




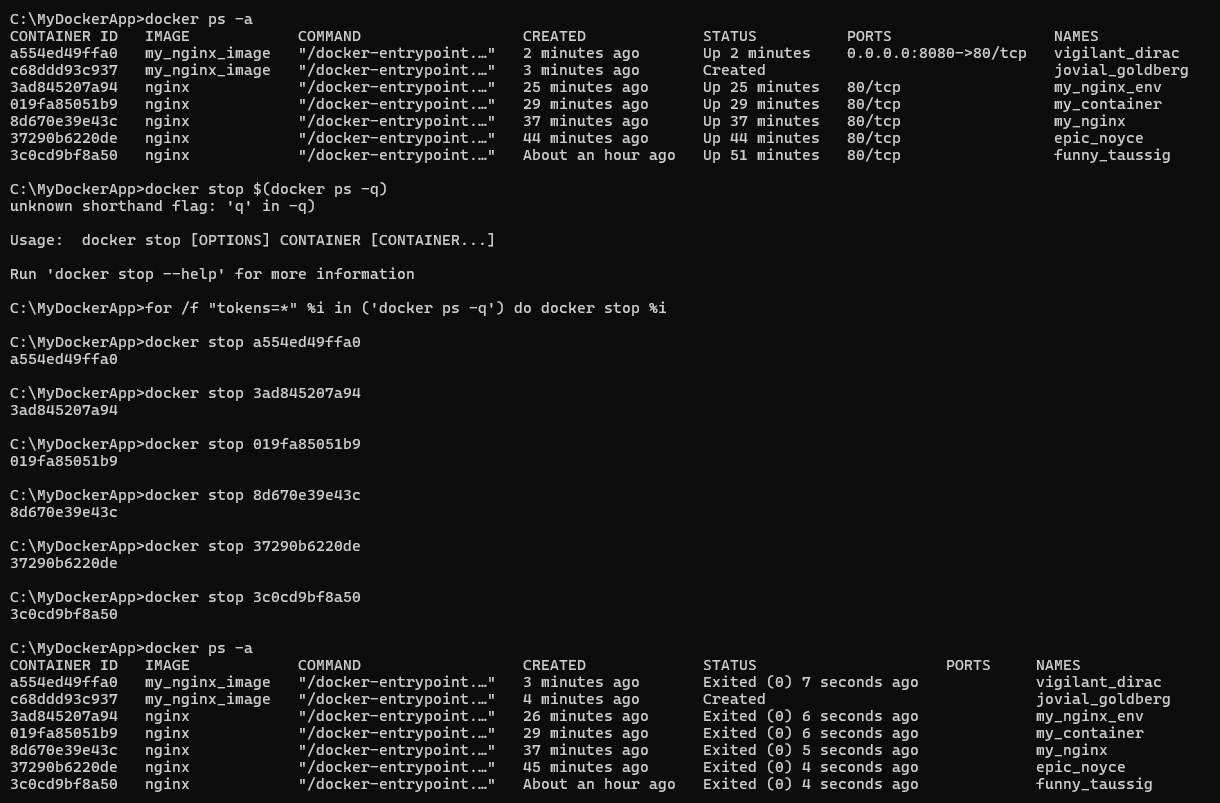
1. **docker build -t my\_nginx\_image:** Xây dựng một image mới từ Dockerfile trong thư mục hiện tại (.). Image mới này sẽ có tên là my\_nginx\_image.



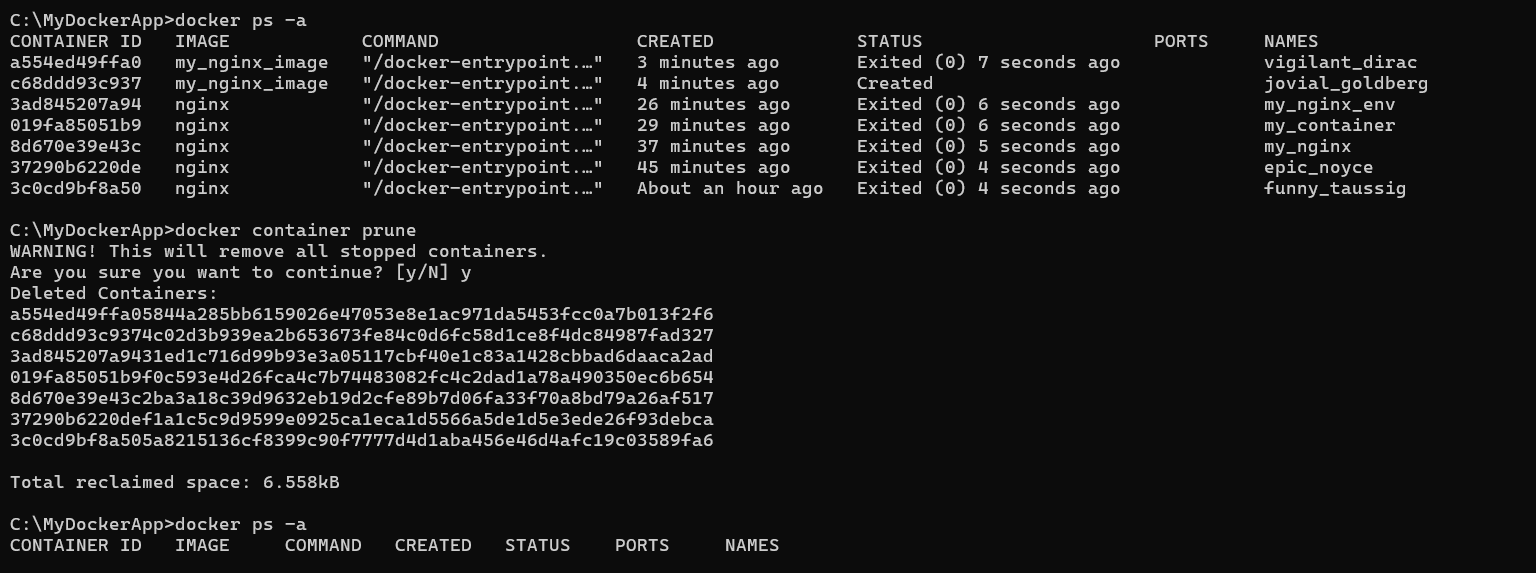
1. **docker run -d -p 8080:80 my\_nginx\_image:** Chạy một container từ image my\_nginx\_image trong chế độ nền, ánh xạ cổng 8080 trên máy chủ tới cổng 80 trong container. Container này sẽ chạy dựa trên image mà bạn vừa xây dựng.



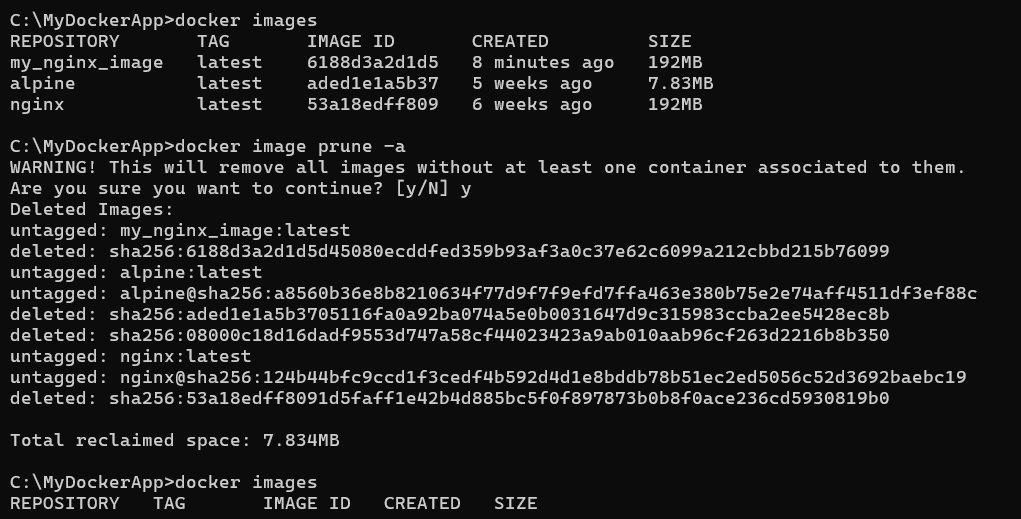
Dừng tất cả các container đang chạy: for /f "tokens=\*" %i in ('docker ps -q') do docker stop %I



Xóa tất cả các container đang dừng: docker container prune



Xóa tất cả images: docker image prune -a



Xóa volume:

Volume mydata không bị xóa vì nó được sử dụng bởi ít nhất một container đã tạo ra nó hoặc đã gắn kết nó trước đây.

🔍 Để xóa được volume mydata, bạn cần:

Dừng và xóa tất cả các container đang sử dụng nó.

Xóa volume thủ công.

