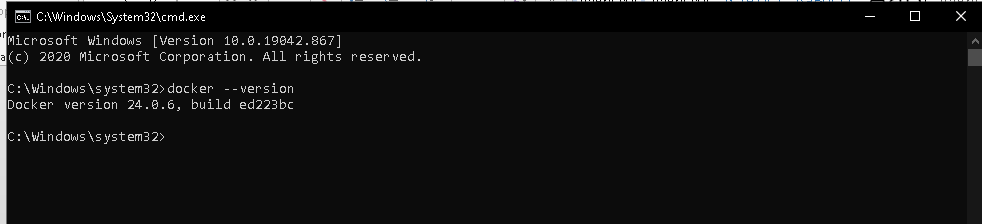
Phần 1: Các lệnh cơ bản thao tác với Docker

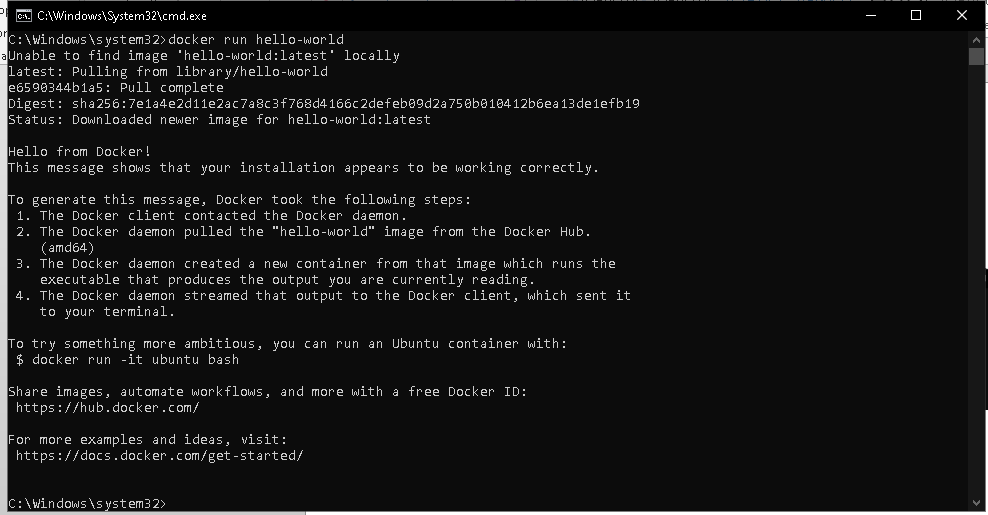
**1. docker –version**

Hiển thị phiên bản Docker hiện tại đang cài đặt.



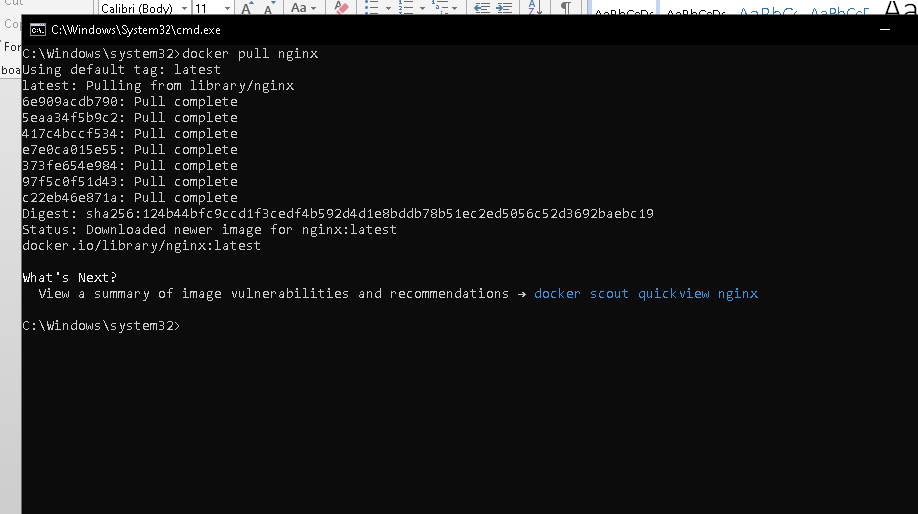
**docker run hello-world**

Chạy container thử nghiệm đơn giản để kiểm tra Docker hoạt động bình thường.



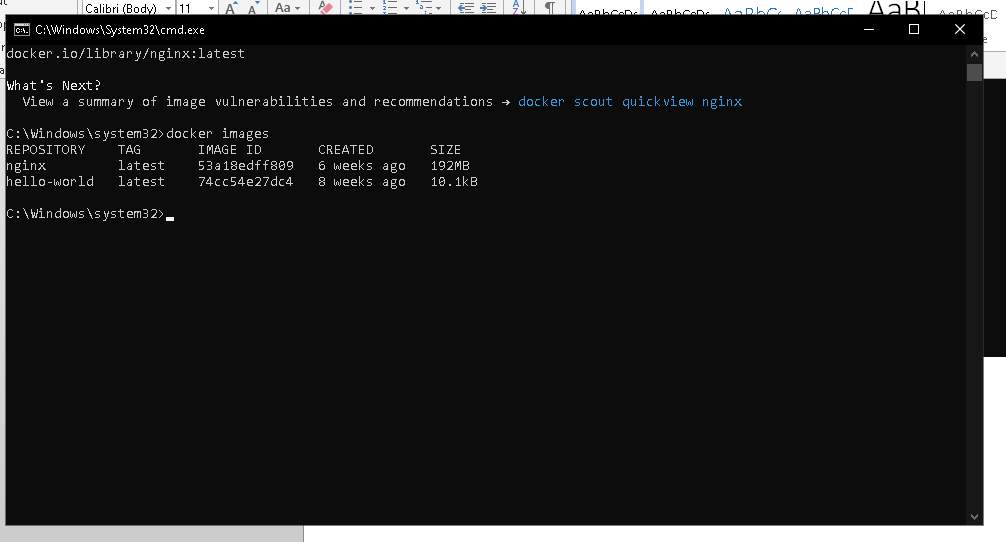
**docker pull nginx**

Tải image Nginx từ Docker Hub về máy.



**docker images**

Liệt kê tất cả các image đã được tải về và có trên hệ thống.



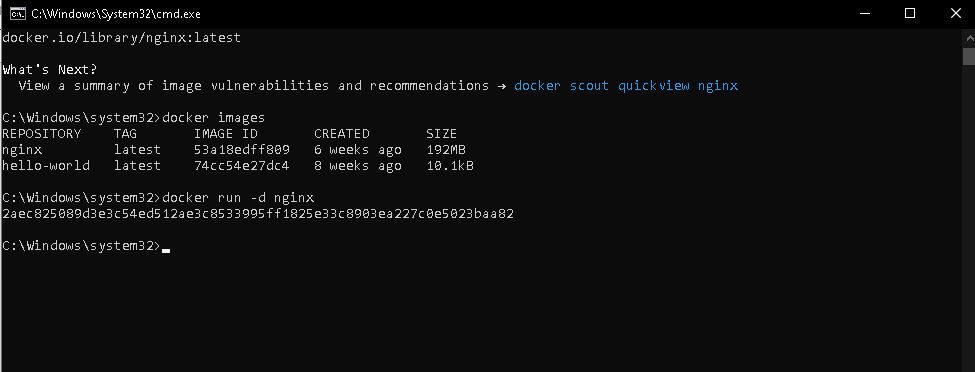
 **nginx**: Đây là một image Nginx, có tag là latest, image ID là 53a18edff809, được tạo ra 6 tuần trước với kích thước là 192MB.

 **hello-world**: Đây là một image thử nghiệm để kiểm tra Docker, có tag là latest, image ID là 74cc54e27dc4, được tạo ra 8 tuần trước với kích thước là 10.1kB.

Các lệnh **docker images** sẽ giúp bạn xem tất cả các image đã tải về và có sẵn trên máy tính của bạn.

**docker run -d nginx**

Chạy một container Nginx trong chế độ tách biệt (-d), không hiển thị đầu ra.

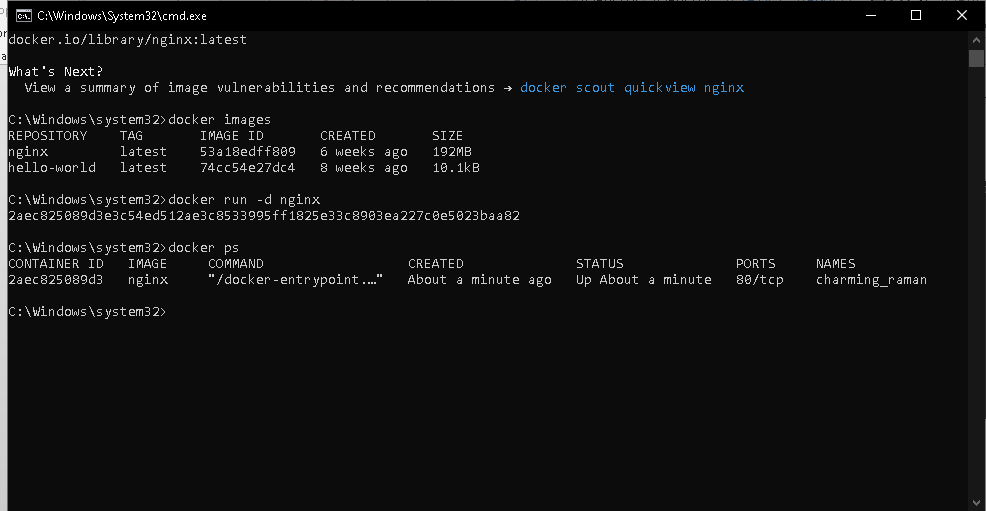


Sau khi chạy lệnh, Docker sẽ trả về một ID container (trong trường hợp của bạn là: 2aec825089d3e3c54ed512ae3c8533995ff1825e33c8903ea227c0e5023baa82). Đây là mã nhận dạng duy nhất của container bạn vừa tạo.

Để kiểm tra xem container có đang chạy không, bạn có thể sử dụng lệnh **docker ps** để liệt kê tất cả các container đang chạy

**docker ps**

Hiển thị danh sách các container đang chạy.

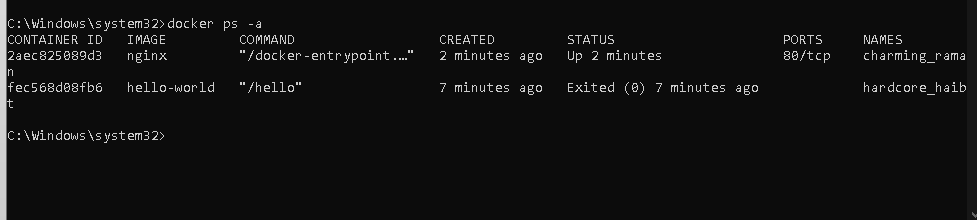


**Giải thích các cột**:

* **CONTAINER ID**: 2aec825089d3 là ID của container.
* **IMAGE**: Image mà container đang chạy từ đó là nginx.
* **COMMAND**: Lệnh mà container đang chạy là /docker-entrypoint... (lệnh khởi động Nginx).
* **CREATED**: Container được tạo ra cách đây khoảng 1 phút.
* **STATUS**: Container đang hoạt động, trạng thái là "Up About a minute" (đang chạy được khoảng 1 phút).
* **PORTS**: Container mở cổng 80 (mặc định của Nginx), nhưng hiện tại chưa có ánh xạ cổng với máy chủ của bạn.
* **NAMES**: Docker đã tự động tạo tên cho container là charming\_raman

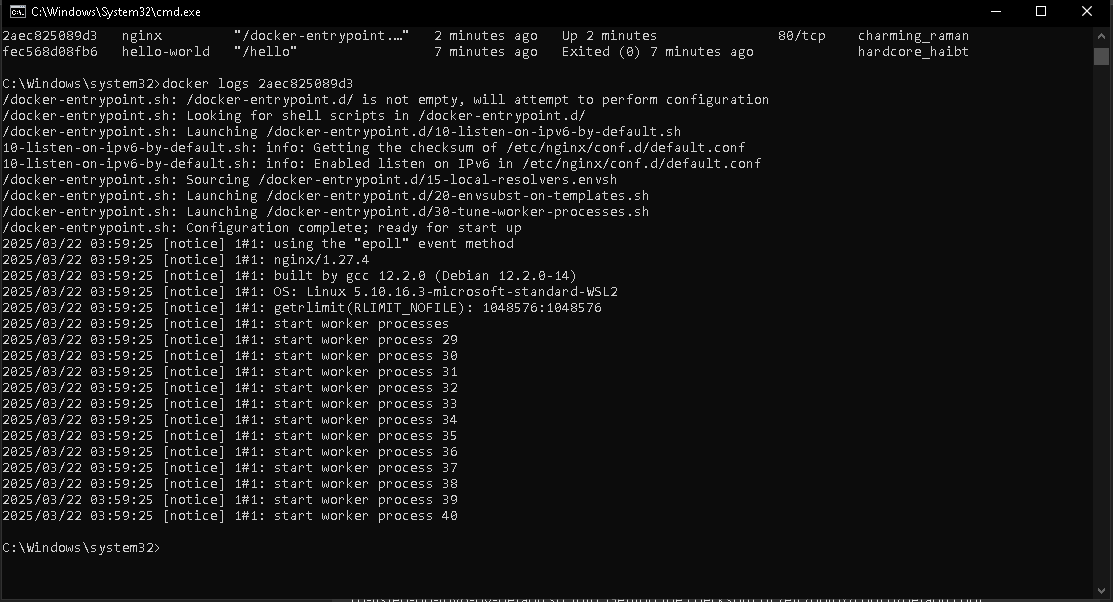
**docker ps -a**

Hiển thị danh sách tất cả các container, bao gồm cả các container đã dừng.



**docker logs <container\_id>**

Xem log đầu ra của một container dựa trên container\_id.



Container Nginx của bạn đã được khởi động thành công và hiện đang chạy. Các script cấu hình đã được thực thi và Nginx đang lắng nghe các kết nối (bao gồm cả IPv6).

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

Truy cập vào shell của container đang chạy để thực hiện các thao tác bên trong container.

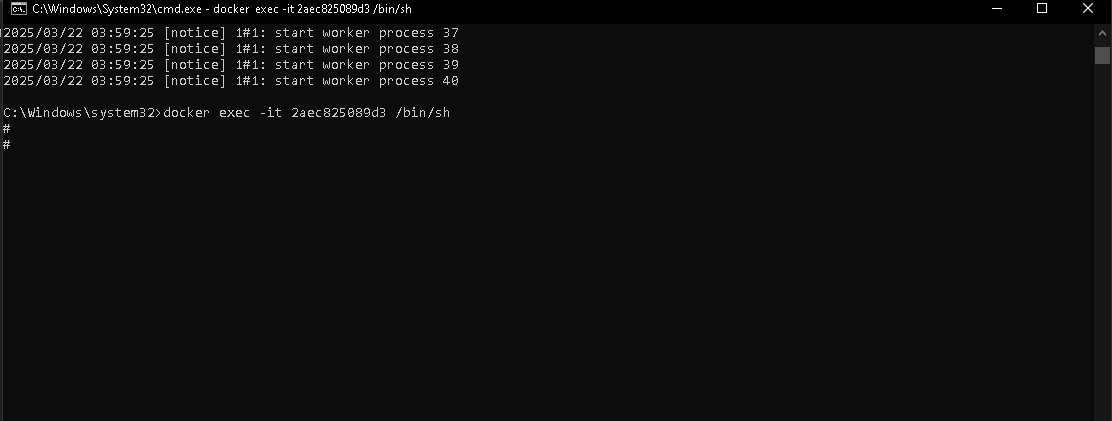
 **ocker exec**: Lệnh này cho phép bạn thực thi một lệnh trong một container đang chạy.

 **-it**:

* **-i** (interactive): Cho phép bạn tương tác với container.
* **-t** (tty): Cung cấp một terminal ảo, giúp bạn có thể làm việc với shell như trên môi trường dòng lệnh.

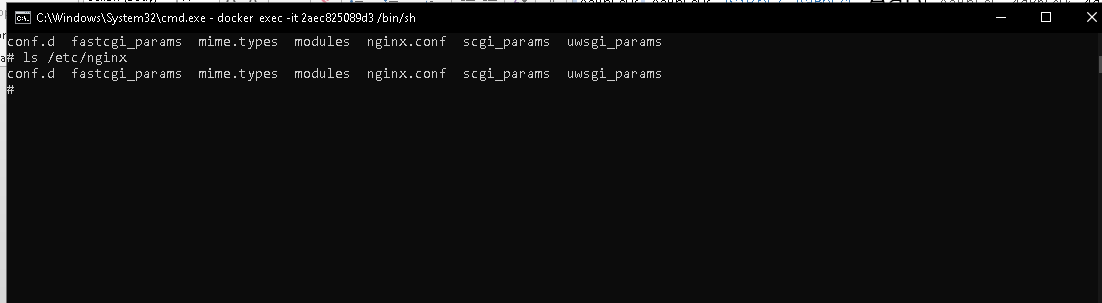
 **<container\_id>**: ID hoặc tên của container mà bạn muốn truy cập vào.

 **/bin/sh**: Lệnh để mở một shell (hoặc một trình shell khác, ví dụ như bash nếu có) bên trong container.



Điều này nghĩa là bạn đã vào trong môi trường shell của container và có thể thực hiện các thao tác như kiểm tra file, sửa đổi cấu hình, hoặc chạy các lệnh như trên hệ điều hành Linux.

kiểm tra các tệp bên trong container. Ví dụ, bạn muốn kiểm tra các tệp cấu hình Nginx trong thư mục /etc/nginx:



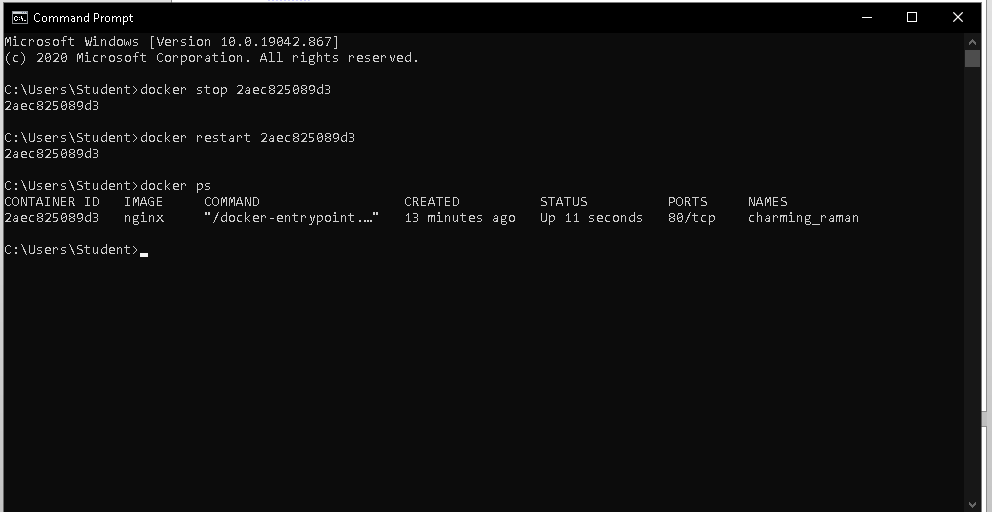
**docker stop <container\_id>**

Dừng một container đang chạy.



docker restart <container\_id>

Khởi động lại một container.



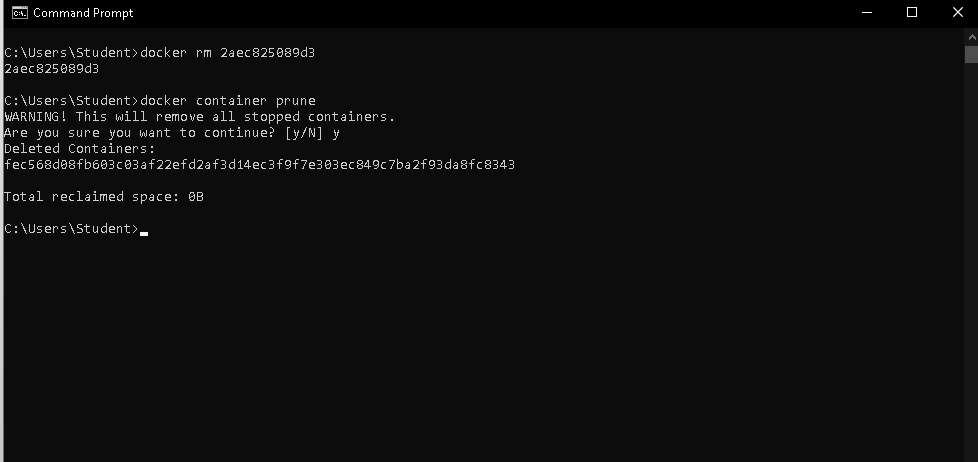
docker rm <container\_id>

Xóa một container đã dừng.



docker container prune

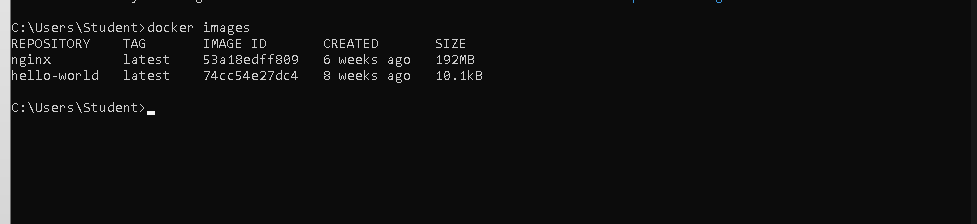
Xóa tất cả các container đã dừng.

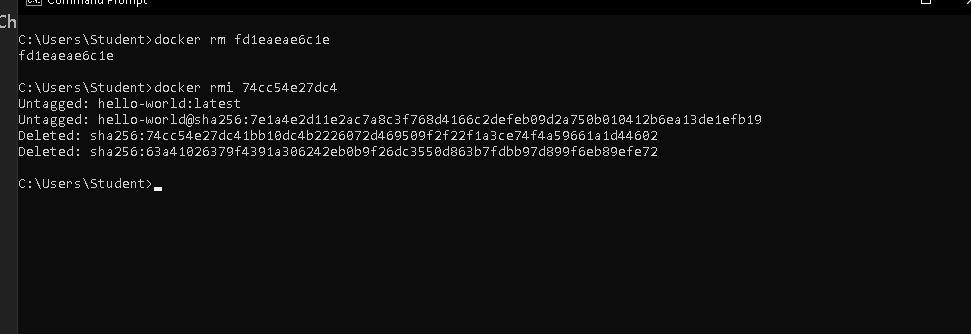


docker rmi <image\_id>

Xóa một image Docker không còn sử dụng.

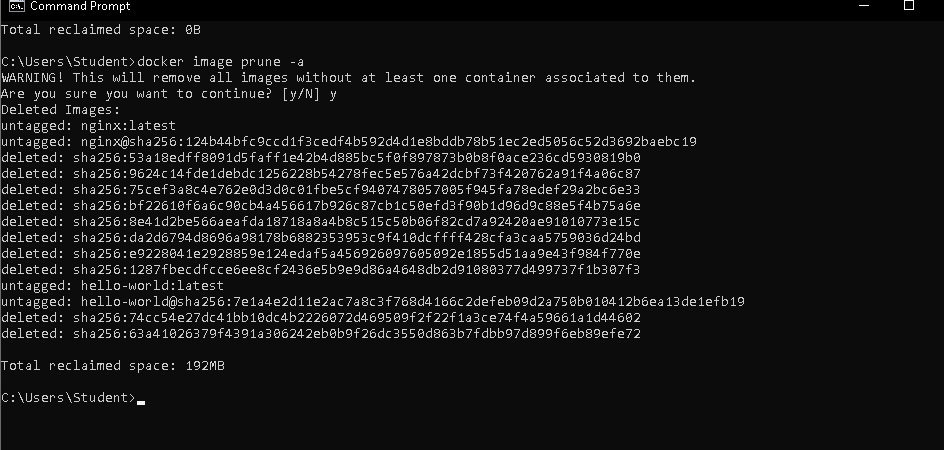
docker images





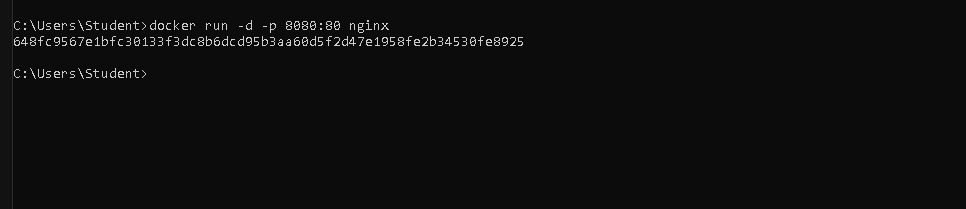
docker image prune -a

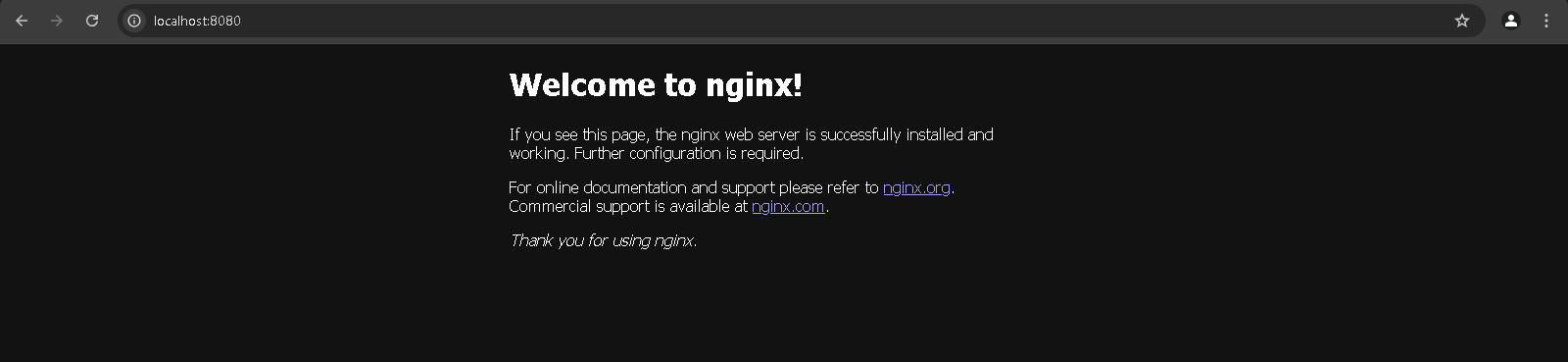
Xóa tất cả các image không còn được sử dụng bởi bất kỳ container nào.



docker run -d -p 8080:80 nginx

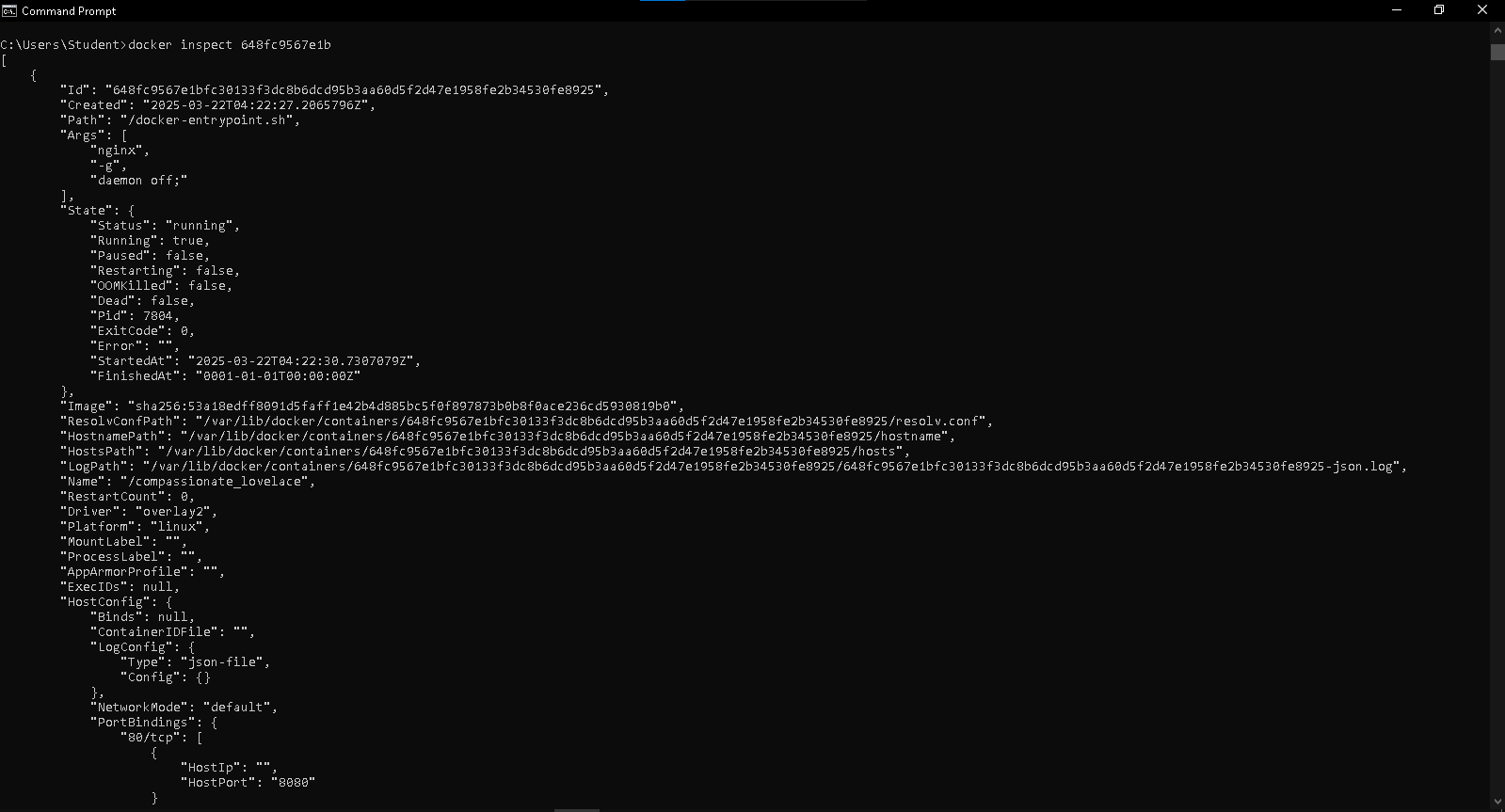
Chạy container Nginx và ánh xạ cổng 80 của container ra cổng 8080 trên máy chủ.





docker inspect <container\_id>

Kiểm tra thông tin chi tiết về một container.



docker run -d -v mydata:/data nginx

Chạy một container với một volume (mydata) được gắn vào thư mục /data trong container.

A black screen with white text

AI-generated content may be incorrect.

docker volume ls

Liệt kê tất cả các volumes đã tạo trên Docker.

docker volume prune

Xóa tất cả các volume không còn được sử dụng.

docker run -d --name my\_nginx nginx

Chạy container Nginx và gán tên là my\_nginx.

docker stats

Hiển thị thông tin thống kê về các container đang chạy, như CPU, bộ nhớ, I/O.

docker network ls

Liệt kê tất cả các mạng Docker.

docker network create my\_network

Tạo một mạng Docker mới có tên là my\_network.

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

Chạy một container Nginx và kết nối nó với mạng my\_network.

docker network connect my\_network my\_nginx

Kết nối một container đang chạy (my\_nginx) vào một mạng (my\_network).

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

Chạy container Nginx và thiết lập biến môi trường MY\_ENV với giá trị hello\_world.

docker logs -f my\_nginx

Xem log của container my\_nginx và theo dõi liên tục (tương tự như tail -f).

FROM nginx

Dùng trong Dockerfile, chỉ định image gốc để tạo Docker image mới.

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

Dùng trong Dockerfile, sao chép tệp index.html từ máy chủ vào container tại thư mục chỉ định.

docker build -t my\_nginx\_image .

Xây dựng một image Docker từ Dockerfile trong thư mục hiện tại (.) và gán tên là my\_nginx\_image.

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

Chạy container từ image my\_nginx\_image và ánh xạ cổng 80 của container ra cổng 8080 trên máy chủ.