Phần 1: Các lệnh cơ bản thao tác với Docker

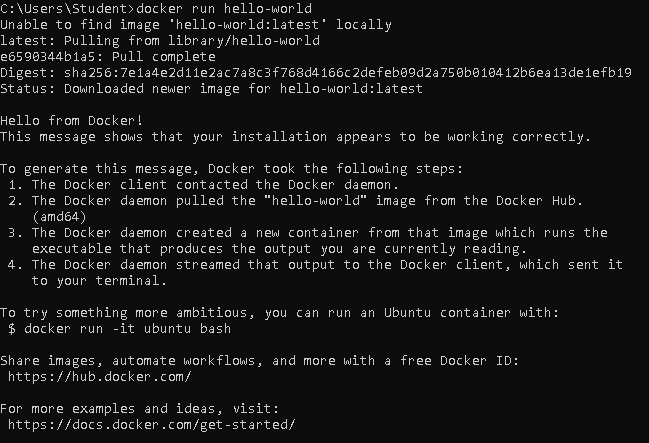
1 docker --version

Kiểm tra phiên bản của Docker



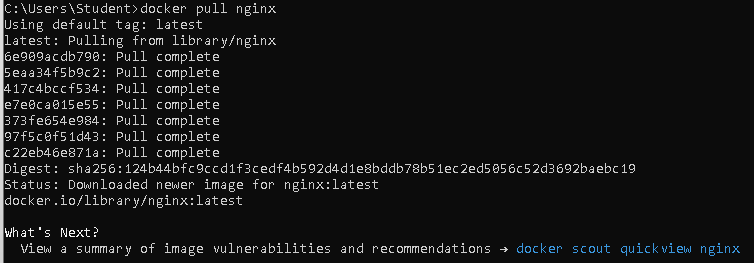
2 docker run hello-world

Kiểm tra xem image hello-word có trên máy chưa nếu chưa thì sẽ tải trên Docker Hub về . Sau khi tải xong image, Docker sẽ tạo một container mới từ image đó và chạy nó.



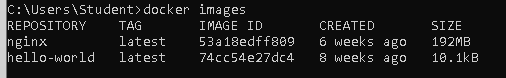
3 docker pull nginx

Yêu cầu tải Image nginx từ Docker Hub về. Sau khi tải thành công, bạn sẽ thấy thông báo



4 docker images

Liệt kê tất cả các **images** hiện có trên máy của bạn



5 docker run -d nginx

Nếu image nginx chưa được tải về trên máy của bạn, Docker sẽ tự động tải image nginx từ Docker Hub. Docker sẽ tạo một container mới từ image nginx và chạy nó. Với tùy chọn -d, container sẽ được chạy ở chế độ nền



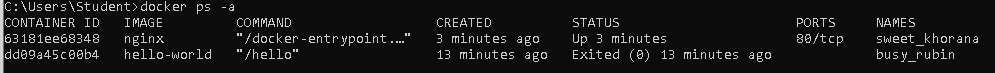
6 docker ps

Liệt kê các container đang chạy trên máy của bạn



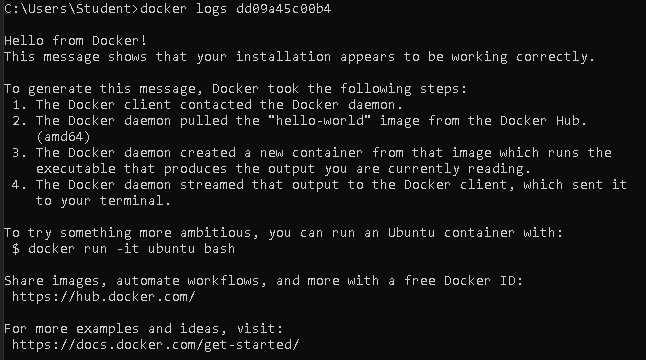
7 docker ps -a

Liệt kê tất cả các container trên máy của bạn, bao gồm cả các container đang chạy và các container đã dừng



8 docker logs <container\_id>

Sử dụng để xem log (nhật ký) của một container đang chạy hoặc đã dừng. Log này sẽ giúp bạn kiểm tra các thông tin về hoạt động của container, chẳng hạn như lỗi, cảnh báo, hoặc các thông điệp từ ứng dụng bên trong container.



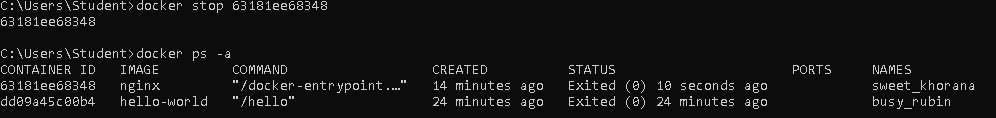
9 docker exec -it <container\_id> /bin/sh

Sử dụng để vào bên trong một container đang chạy và mở một terminal interactive shell (shell tương tác) để bạn có thể thực hiện các lệnh trực tiếp trong môi trường của container. Để thoát khỏi shell trong container, bạn chỉ cần gõ lệnh exit hoặc nhấn Ctrl + D.



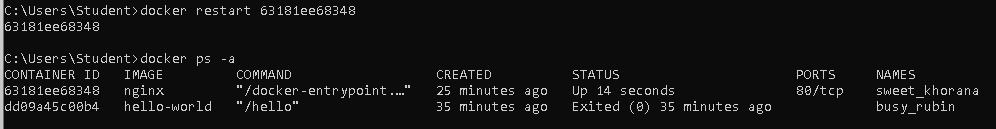
10 docker stop <container\_id>

Sử dụng để dừng một container đang chạy.



11 docker restart <container\_id>

Sử dụng để khởi động lại một container đang chạy hoặc đã dừng.



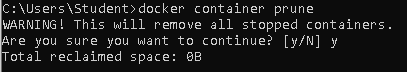
12 docker rm <container\_id>

Sử dụng để xóa một container đã dừng.



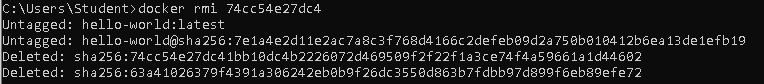
13 docker container prune

Sử dụng để xóa tất cả các container đã dừng trên hệ thống Docker



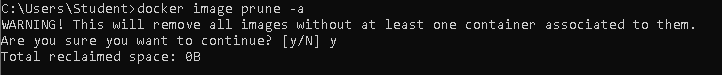
14 docker rmi <image\_id>

Sử dụng để xóa một Docker image khỏi hệ thống.



15 docker image prune -a

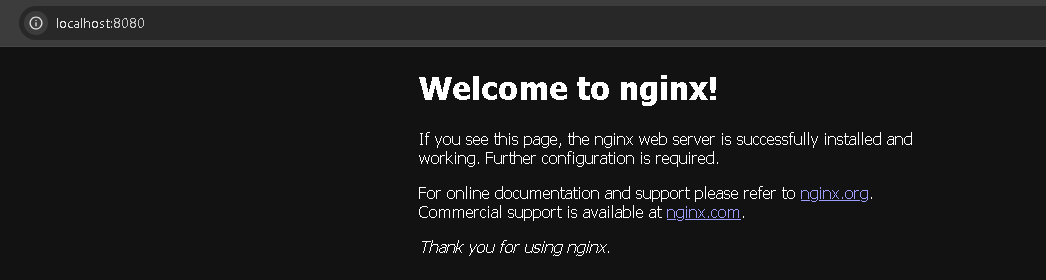
Sử dụng để xóa tất cả các image không còn được sử dụng trên hệ thống Docker



16 docker run -d -p 8080:80 nginx

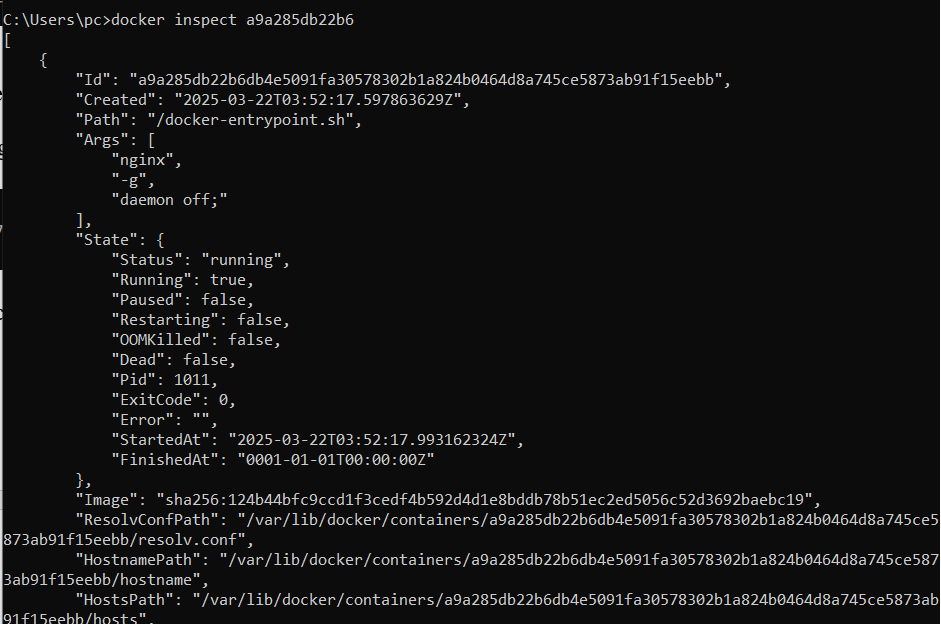
Sử dụng để chạy một container Nginx trong chế độ nền và mở cổng từ container ra hệ thống của bạn.





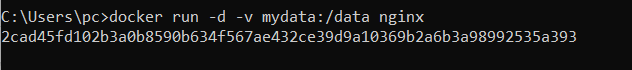
17 docker inspect <container\_id>

Hiển thị thông tin chi tiết về một container.

****

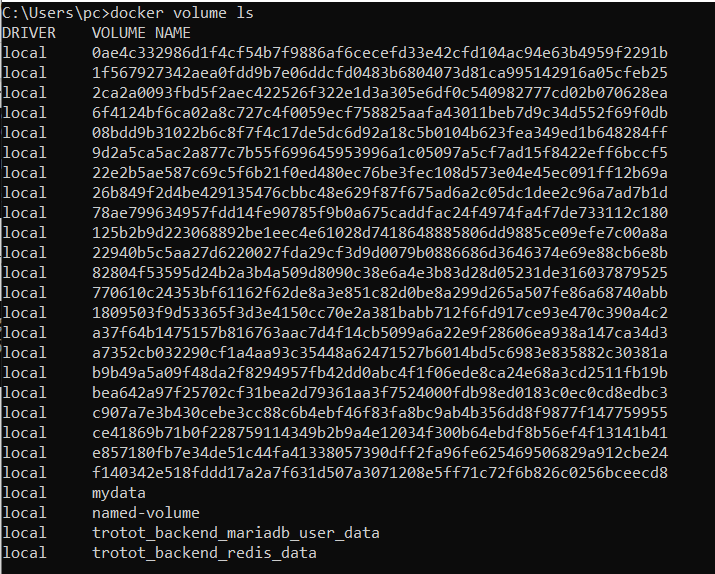
18 docker run -d -v mydata:/data nginx

Chạy container nginx với volume mydata được ánh xạ vào thư mục /data trong container.

****

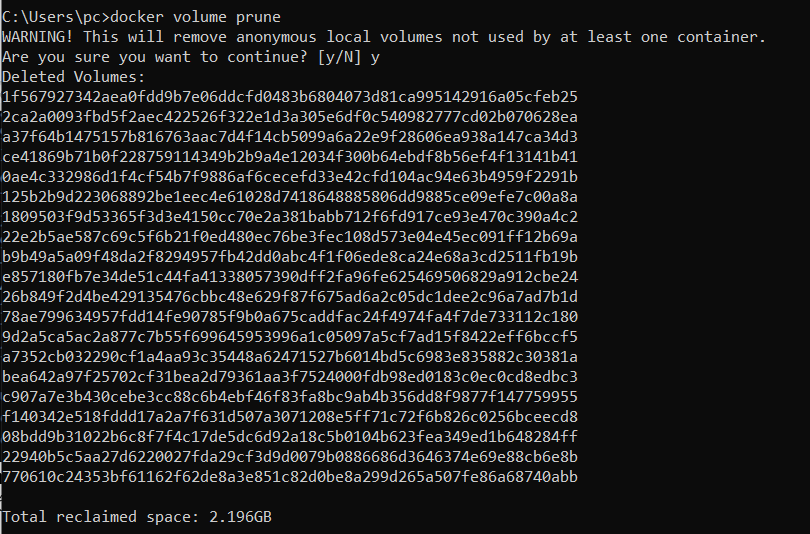
19 docker volume ls

Liệt kê danh sách các volume hiện có.

****

20 docker volume prune

Xóa tất cả volume không sử dụng.

****

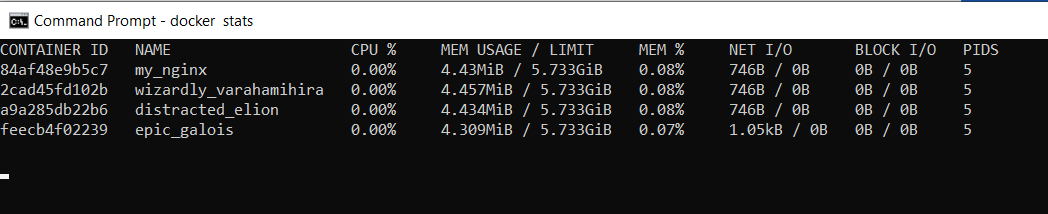
21 docker run -d --name my\_nginx nginx

Chạy container nginx với tên cụ thể my\_nginx.

****

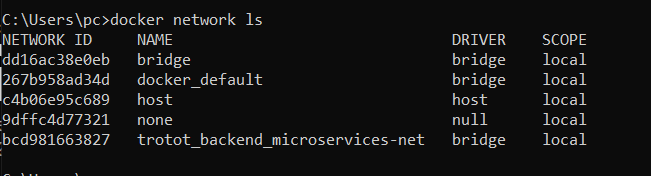
22 docker stats

Hiển thị thông tin sử dụng tài nguyên (CPU, RAM, I/O) của các container đang chạy.

****

23 docker network ls

Liệt kê các network trong Docker.

****

24 docker network create my\_network

Tạo một network có tên my\_network.

****

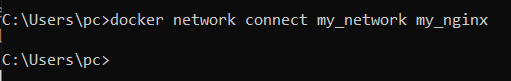
25 docker run -d --network my\_network --name my\_container nginx

Chạy container nginx và gán vào network my\_network.

****

26 docker network connect my\_network my\_nginx

Kết nối container my\_nginx vào network my\_network

****

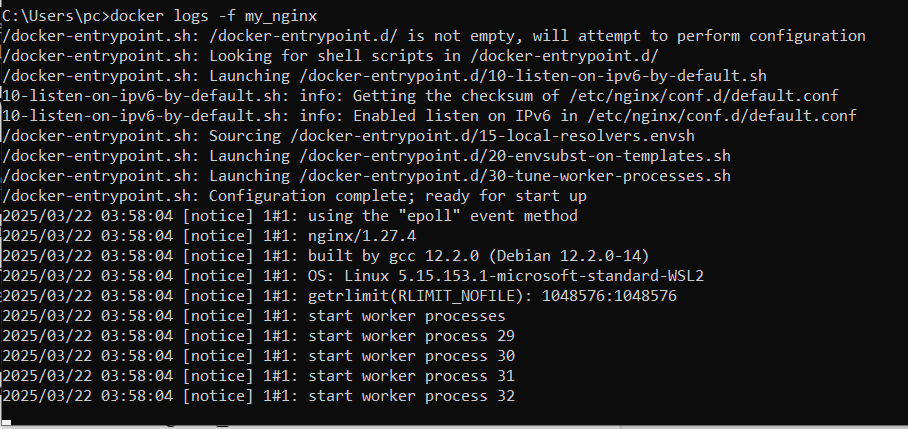
27 docker run -d -e MY\_ENV=hello\_world nginx

Chạy container nginx với biến môi trường MY\_ENV được đặt thành hello\_world.

****

28 docker logs -f my\_nginx

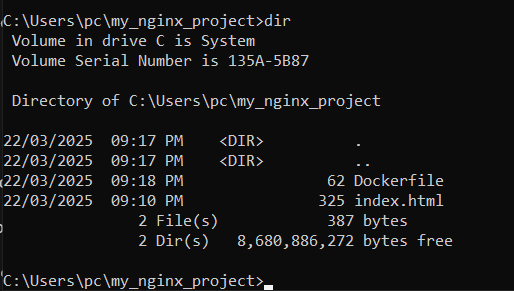
Theo dõi log của container my\_nginx theo thời gian thực

****

29 FROM nginx

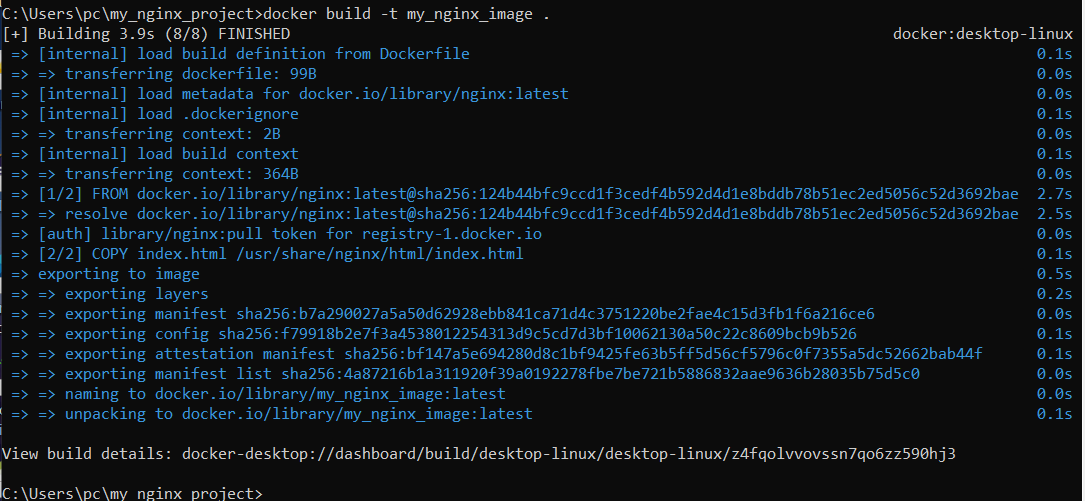
COPY index.html /usr/share/nginx/html/index.html

Tạo một Docker image từ nginx và sao chép file index.html vào thư mục chứa nội dung web.

****

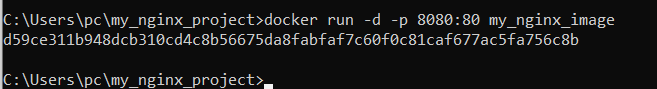
30 docker build -t my\_nginx\_image .

Xây dựng Docker image từ Dockerfile trong thư mục hiện tại với tên my\_nginx\_image.

****

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

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

****