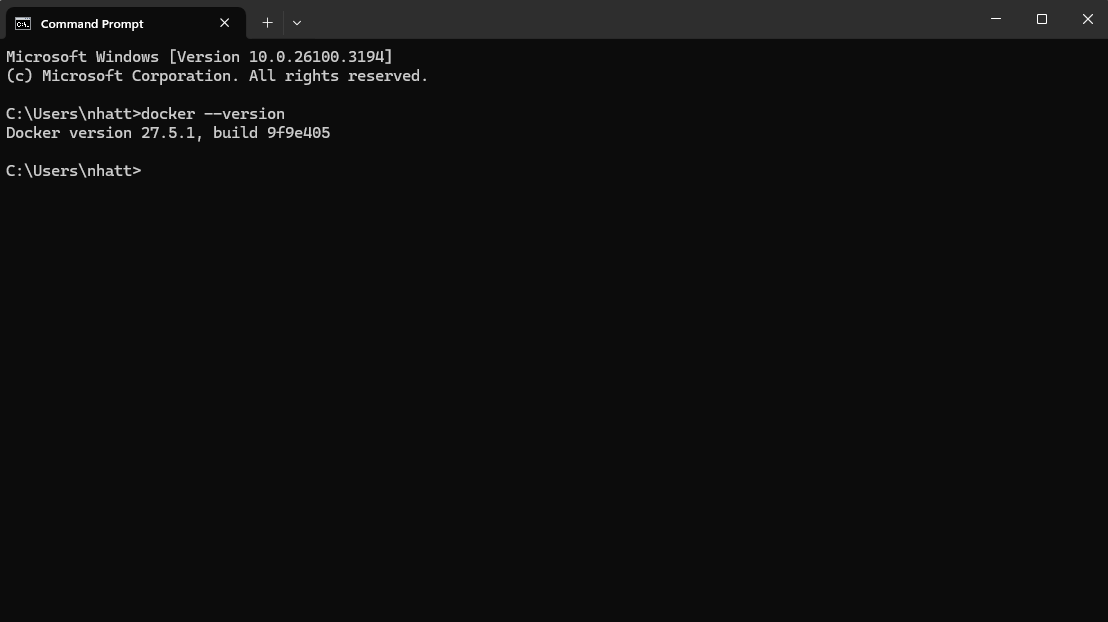
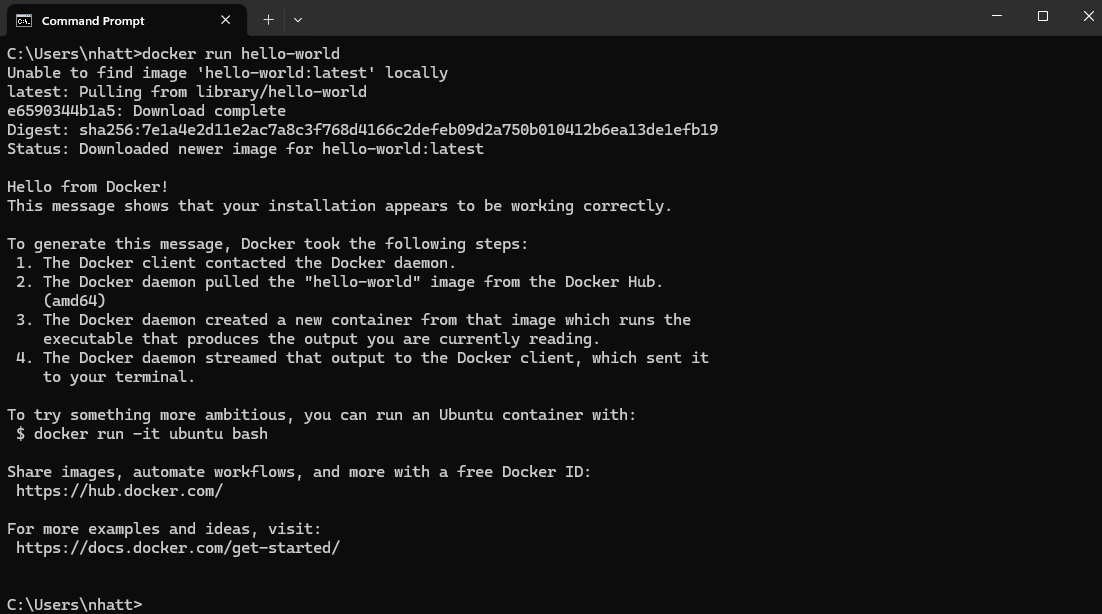
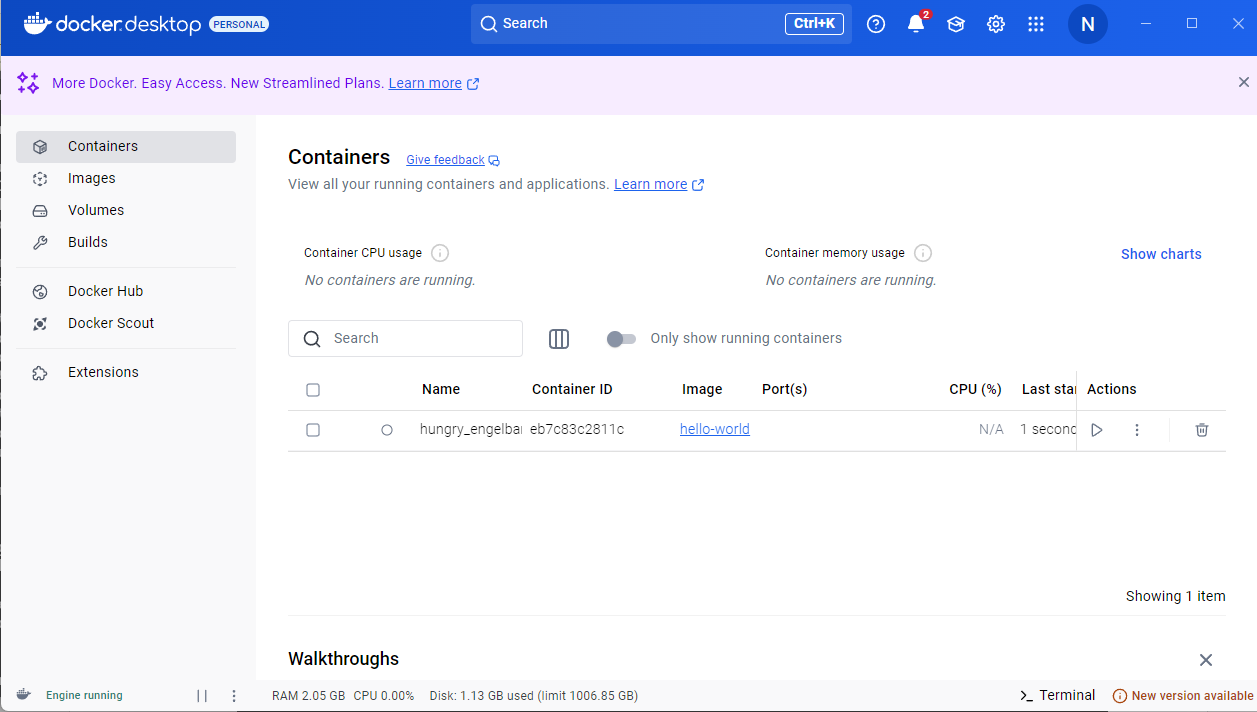
Docker –version : Kiểm tra phiên bản Docker đang cài trên máy.



docker run hello-world :

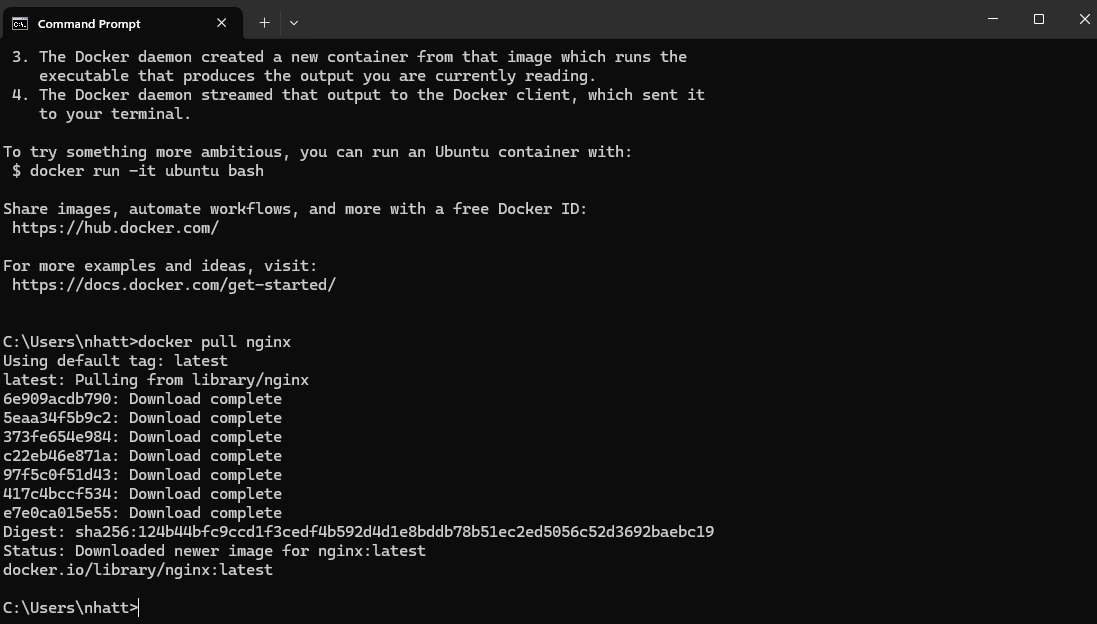
* Tải image hello-world từ Docker Hub (nếu chưa có).
* Tạo và chạy container từ image đó.
* Hiển thị thông báo "Hello from Docker!" để kiểm tra Docker hoạt động đúng.





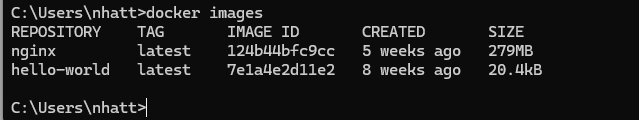
docker pull nginx

* Tải image nginx từ Docker Hub về local
* Nếu đã có image, Docker sẽ kiểm tra phiên bản mới nhất



docker images

* Hiển thị danh sách các image đã tải về local.



docker run -d nginx

* Chạy container từ image nginx ở chế độ nền (-d = detached mode).
* Container sẽ hoạt động mà không chiếm terminal.



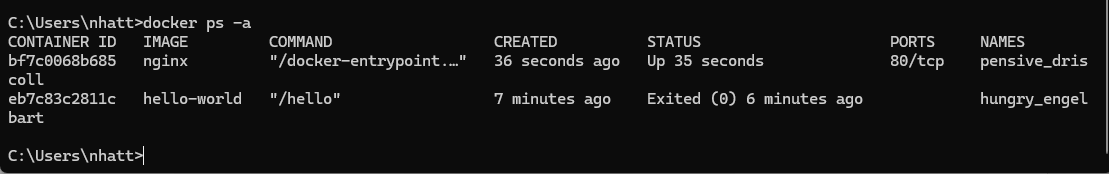
docker ps

* Hiển thị danh sách các container đang chạy.



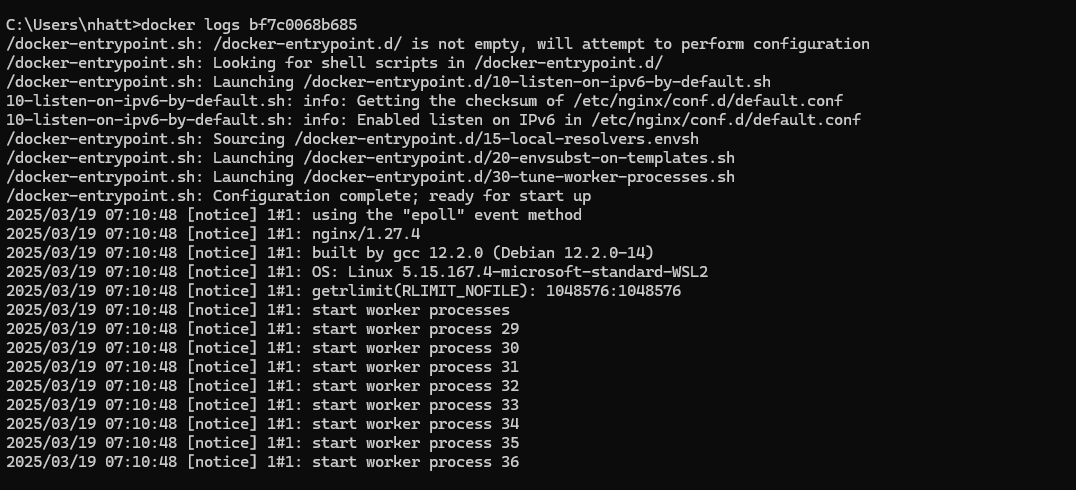
docker ps -a

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



docker logs <container\_id>

* Kiểm tra log đầu ra của container



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

* Truy cập vào **terminal của container** để chạy lệnh bên trong.
* Nếu container chạy Ubuntu/Debian, có thể dùng /bin/bash.



Thoát nhấn exit

docker stop <container\_id>

* Dừng container đang chạy.



docker restart <container\_id>

* Khởi động lại container mà không cần tạo lại



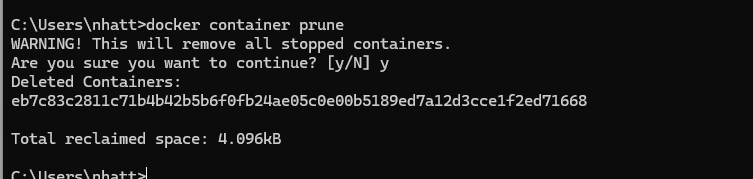
docker rm <container\_id>

* Xóa container khỏi hệ thống (chỉ xóa được nếu nó đã dừng).

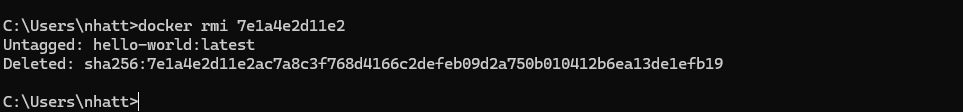


docker container prune

* Xóa tất cả container **đã dừng**, giúp dọn dẹp hệ thống

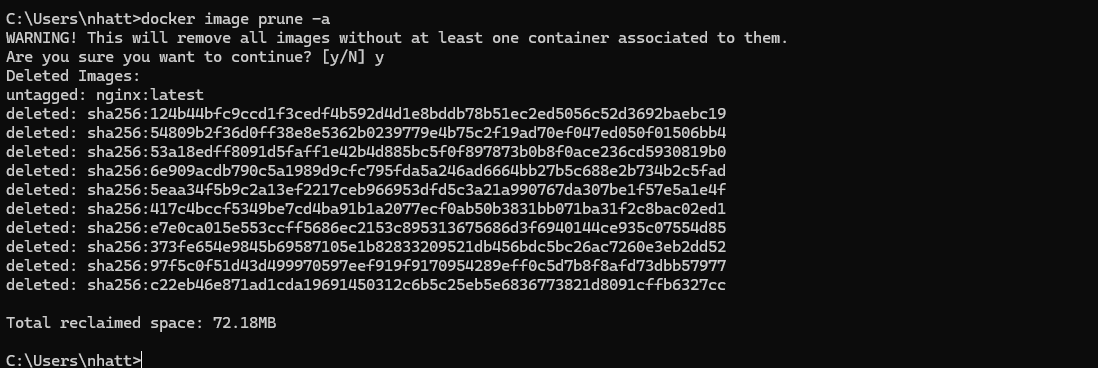


docker rmi <image\_id>

* Xóa image khỏi máy. 

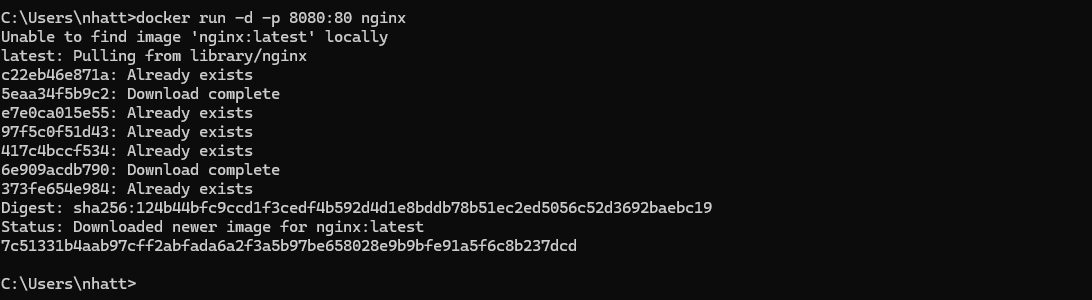
docker image prune -a

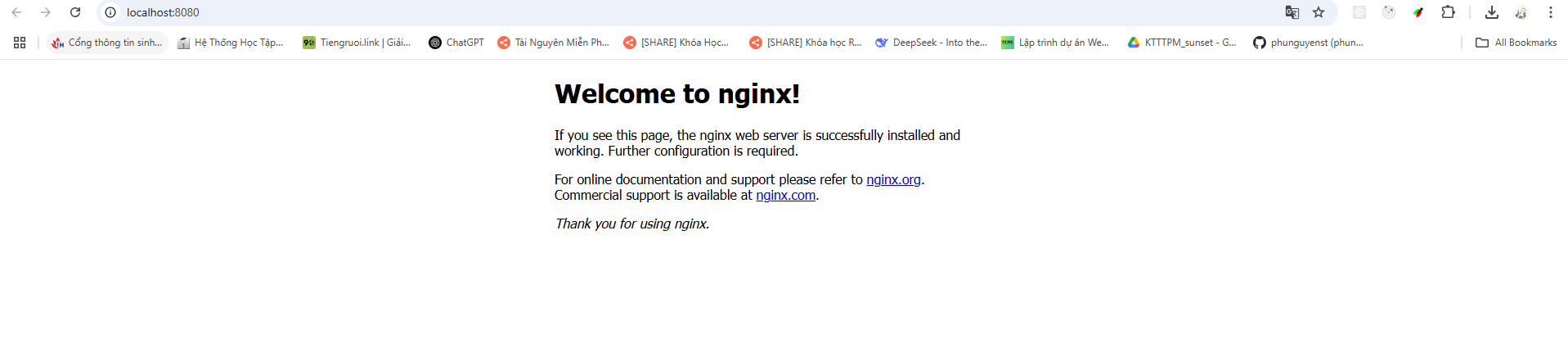
* Dọn dẹp tất cả image không còn sử dụng, giúp tiết kiệm dung lượng.



docker run -d -p 8080:80 nginx

* Chạy container từ image nginx.
* **Map cổng** 8080 của máy host với 80 của container.
* Truy cập http://localhost:8080 để kiểm tra.





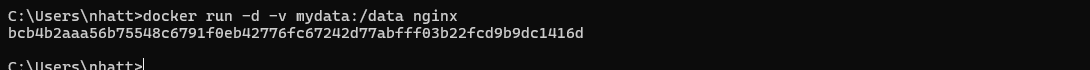
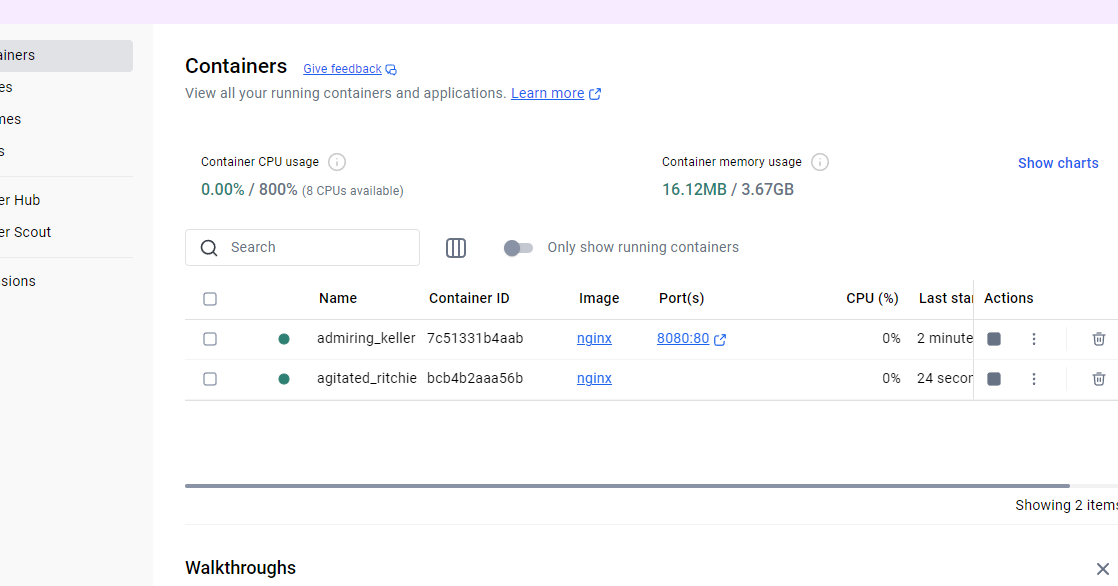
docker inspect <container\_id>

* Hiển thị thông tin chi tiết về container, bao gồm **IP**, **cấu hình mạng**, **mount volume**, v.v.



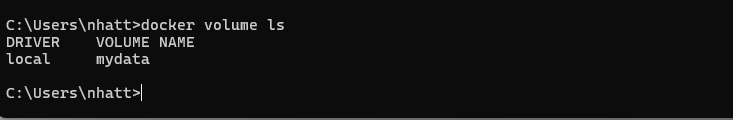
docker run -d -v mydata:/data nginx

* Tạo một volume tên mydata và gắn vào thư mục /data của container.
* Dữ liệu trong /data **vẫn được giữ lại** ngay cả khi container bị xóa.

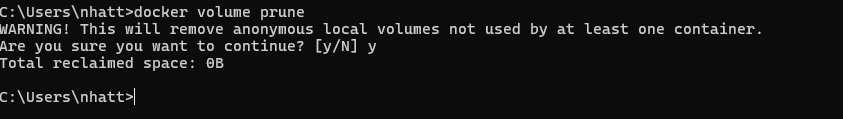
docker volume ls

* Hiển thị danh sách các volume trong Docker (dùng để lưu trữ dữ liệu giữa các container).



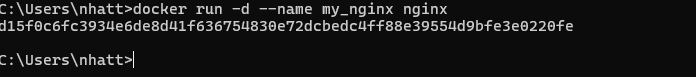
docker volume prune

* Xóa tất cả các volume **không còn được container nào sử dụng** để giải phóng dung lượng



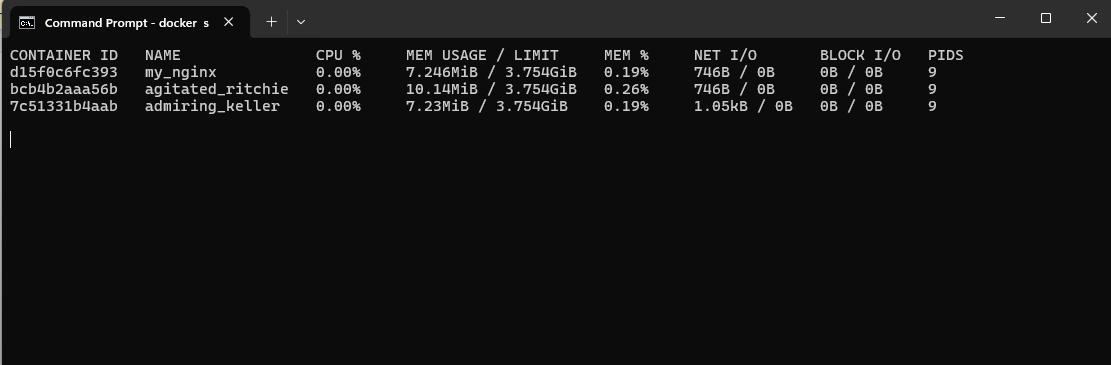
docker run -d --name my\_nginx nginx

* Chạy container từ image nginx với tên my\_nginx và chạy ở chế độ nền (-d).



docker stats

* Hiển thị thông tin tài nguyên (CPU, RAM, network) mà các container đang sử dụng theo thời gian thực.



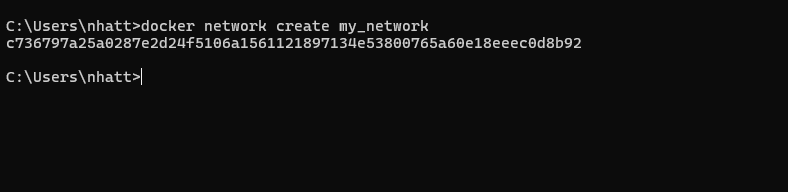
docker network ls

* Hiển thị danh sách các mạng Docker đang có



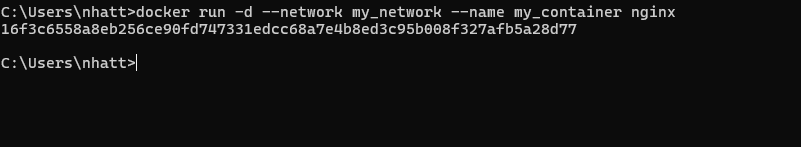
docker network create my\_network

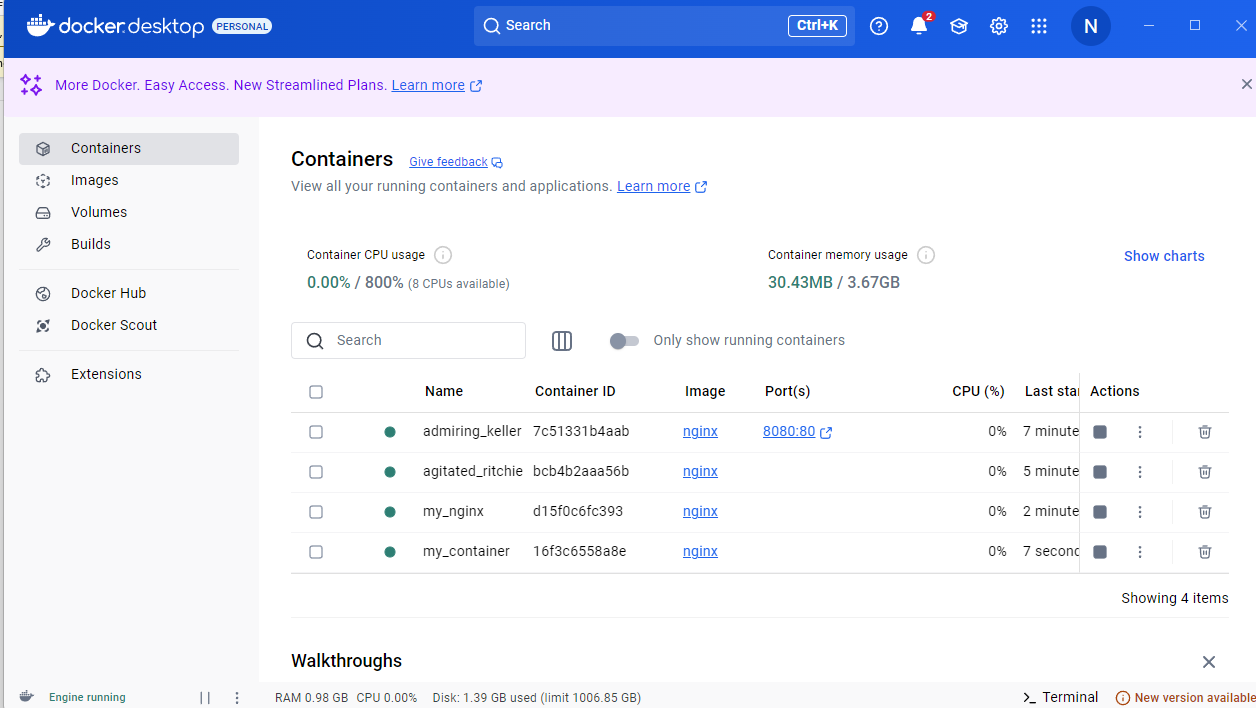
* Tạo một mạng Docker tên là my\_network.



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

* Tạo container my\_container từ image nginx và kết nối nó vào mạng my\_network.





docker network connect my\_network my\_nginx

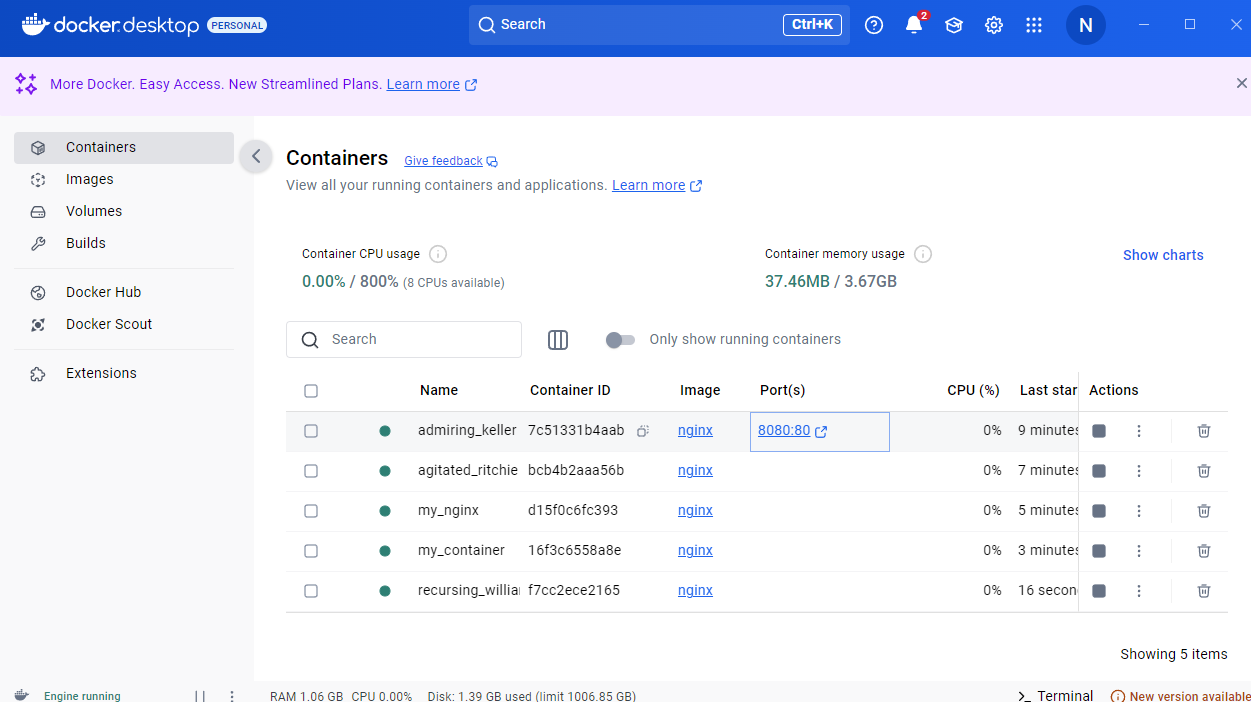
* Kết nối container my\_nginx vào mạng my\_network.



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

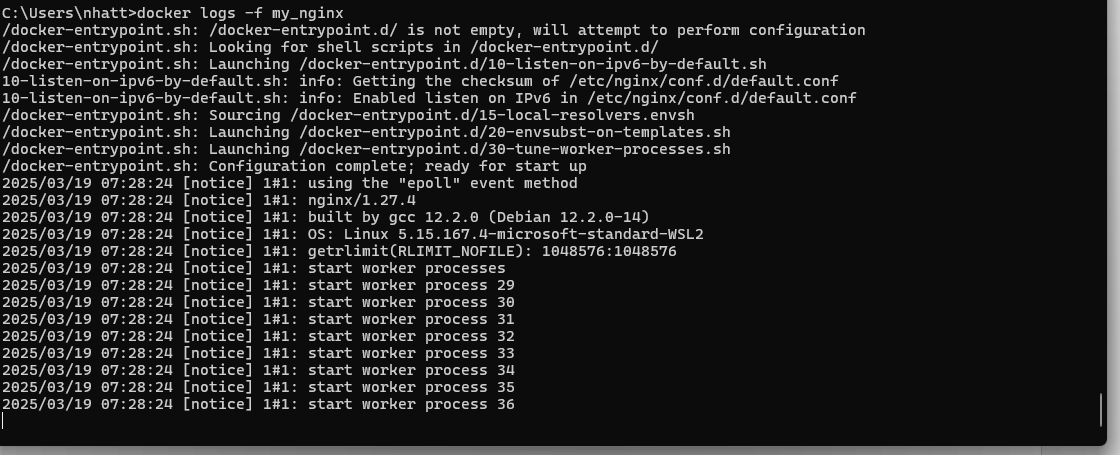
* Chạy container nginx với biến môi trường MY\_ENV=hello\_world.





docker logs -f my\_nginx

* Xem log của container my\_nginx theo thời gian thực (-f để theo dõi liên tục).



docker build -t my\_nginx\_image .

* Build một image mới từ Dockerfile trong thư mục hiện tại (.) và đặt tên (-t) là my\_nginx\_image

