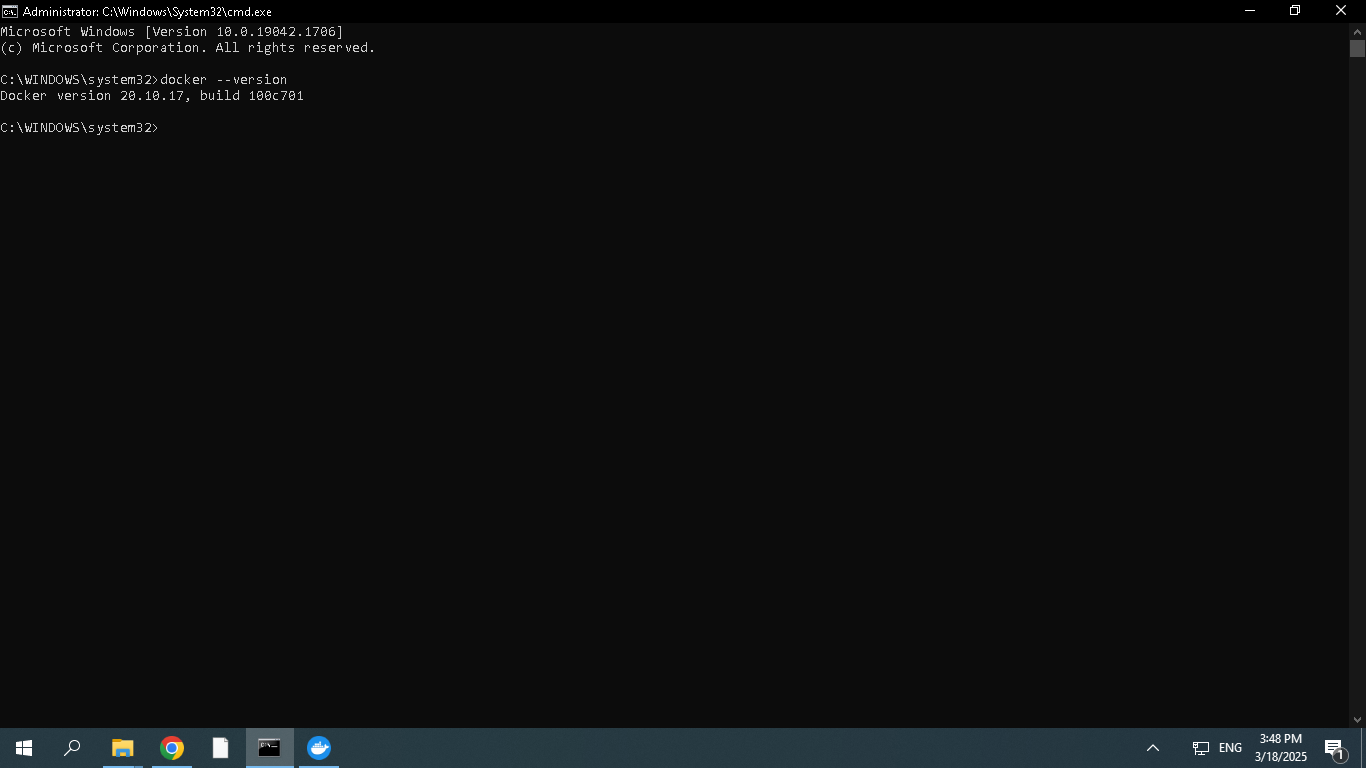
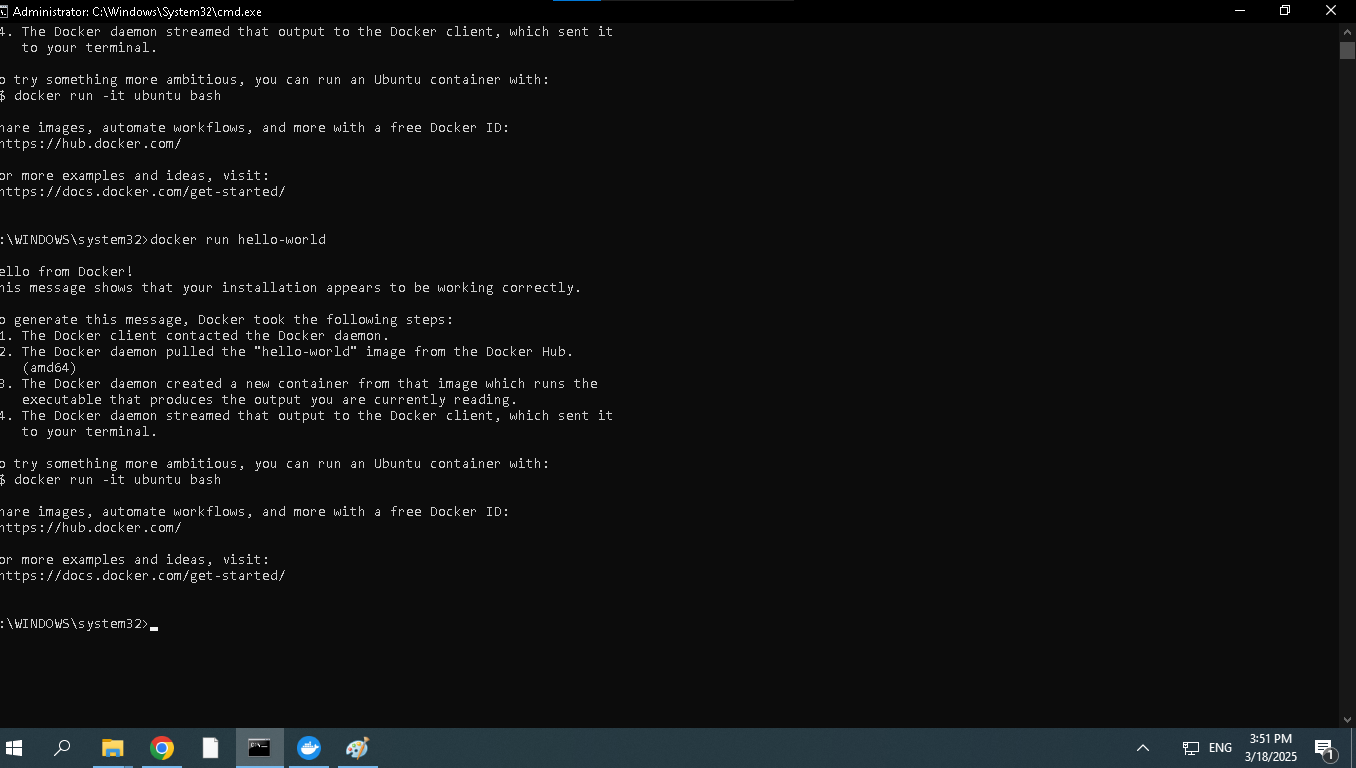
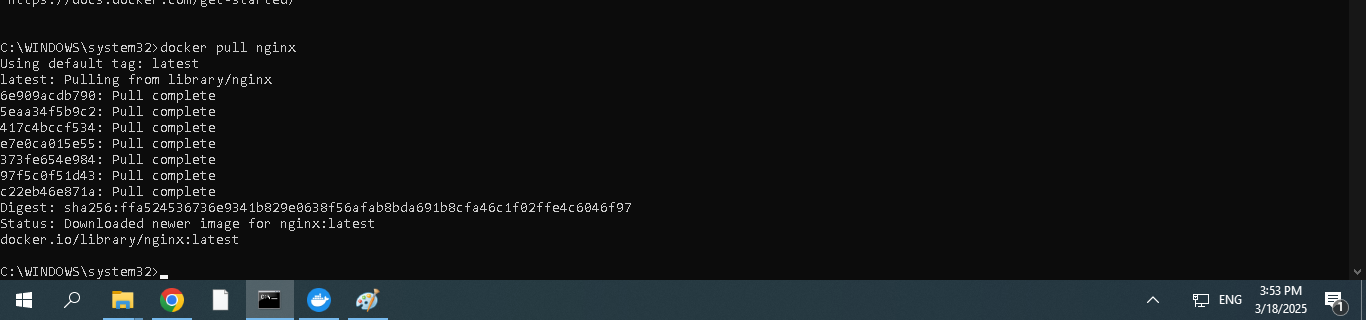
1. **docker --version**: Hiển thị phiên bản Docker hiện đang được cài đặt trên hệ thống. Lệnh này giúp bạn xác định phiên bản Docker để đảm bảo tính tương thích với các ứng dụng hoặc môi trường cụ thể.

****

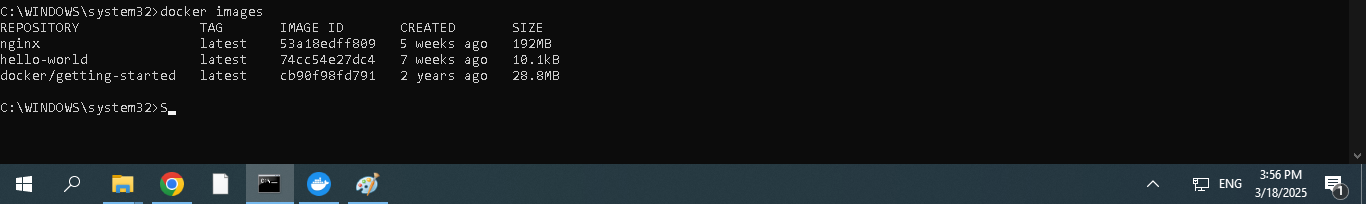
1. **docker run hello-world**: Chạy container từ image hello-world để kiểm tra xem Docker đã được cài đặt và hoạt động đúng hay chưa. Khi lệnh này được thực thi, Docker sẽ tải image hello-world (nếu chưa có) và chạy nó. Container sẽ in ra một thông báo chào mừng, xác nhận rằng Docker hoạt động bình thường.



1. **docker pull nginx**: Tải image Nginx mới nhất từ Docker Hub về máy local. Lệnh này hữu ích khi bạn muốn chuẩn bị sẵn image để sử dụng sau này hoặc để đảm bảo bạn có phiên bản mới nhất của image đó.



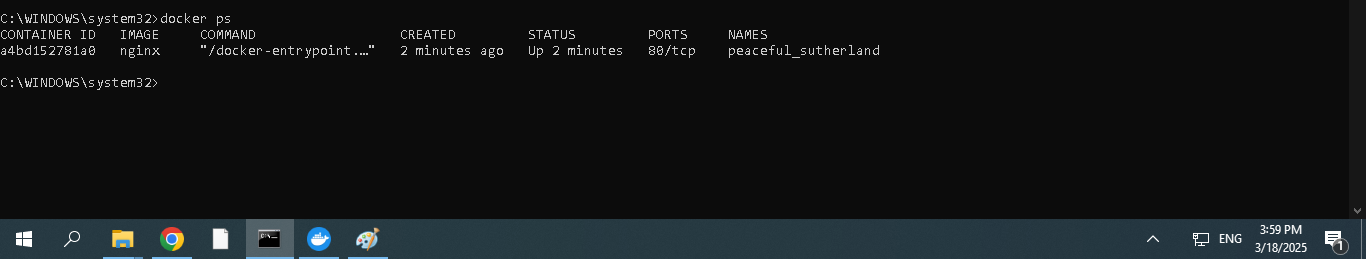
1. **docker images**: Liệt kê tất cả các image Docker hiện có trên máy local. Lệnh này giúp bạn quản lý các image đã tải về, bao gồm thông tin về tên image, thẻ (tag), ID image, thời gian tạo và kích thước.



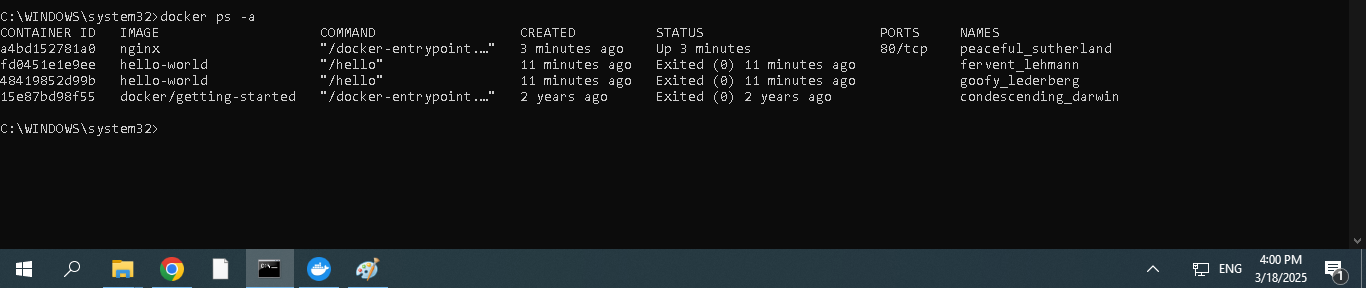
1. **docker run -d nginx**: Chạy một container từ image Nginx trong chế độ nền (detached mode). Lệnh này sẽ khởi chạy Nginx dưới dạng một container chạy ngầm, cho phép bạn tiếp tục sử dụng terminal mà không bị chiếm dụng.



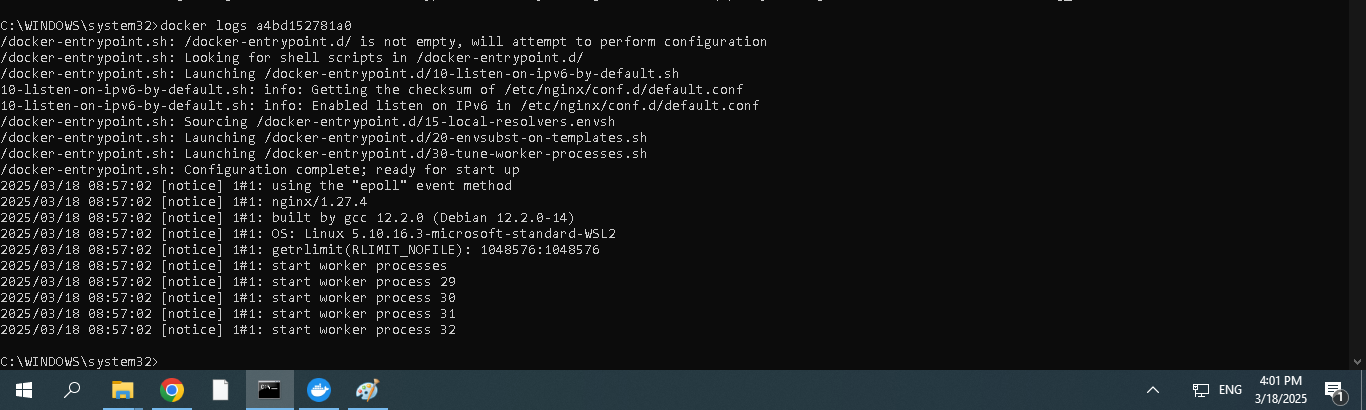
1. **docker ps**: Liệt kê các container đang chạy. Lệnh này cung cấp thông tin về các container đang hoạt động, bao gồm ID container, image sử dụng, lệnh được thực thi, thời gian khởi chạy, trạng thái và tên container.



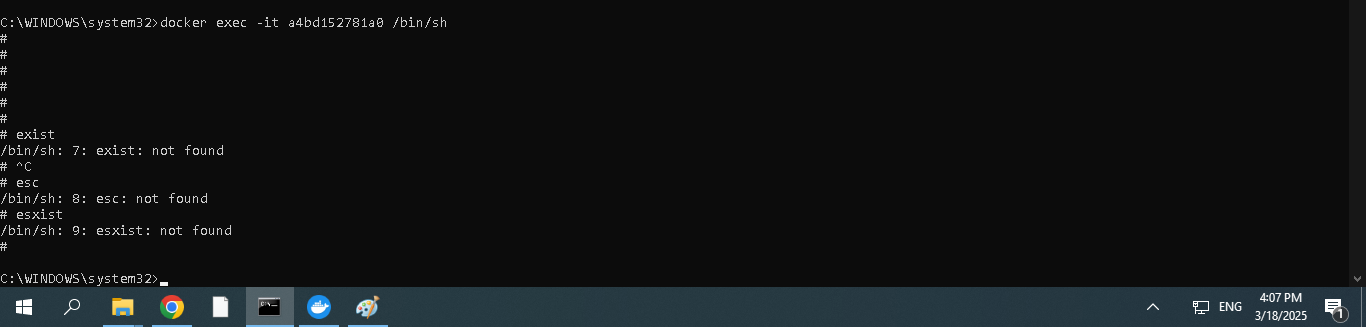
1. **docker ps -a**: Liệt kê tất cả các container, bao gồm cả những container đã dừng. Điều này giúp bạn theo dõi lịch sử các container đã chạy trên hệ thống.



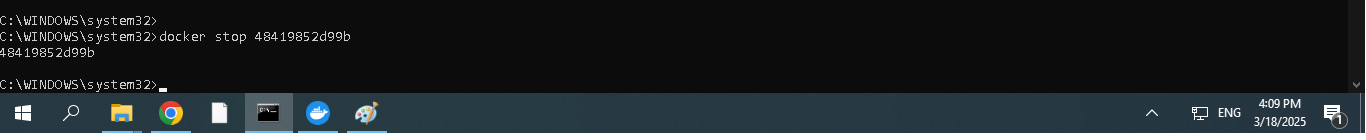
1. **docker logs <container\_id>**: Xem log (nhật ký) của một container cụ thể. Lệnh này hữu ích để kiểm tra output hoặc debug ứng dụng chạy bên trong container.



1. **docker exec -it <container\_id> /bin/sh**: Mở một phiên shell tương tác bên trong container đang chạy. Lệnh này cho phép bạn truy cập vào môi trường bên trong container để thực hiện các thao tác như kiểm tra file, chạy lệnh hoặc debug.



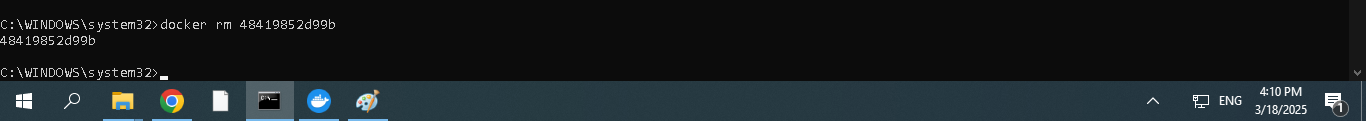
1. **docker stop <container\_id>**: Dừng một container đang chạy. Lệnh này gửi tín hiệu dừng (SIGTERM) đến container, cho phép nó thực hiện các thao tác dọn dẹp trước khi tắt.



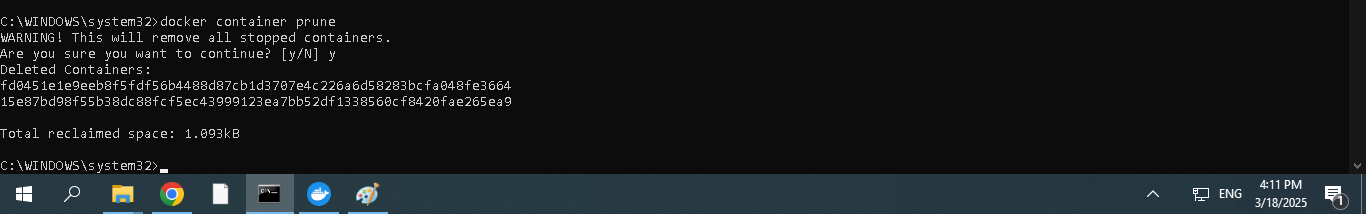
1. **docker restart <container\_id>**: Khởi động lại một container. Lệnh này tương đương với việc dừng và sau đó khởi động lại container, hữu ích khi bạn muốn áp dụng các thay đổi cấu hình hoặc cập nhật.



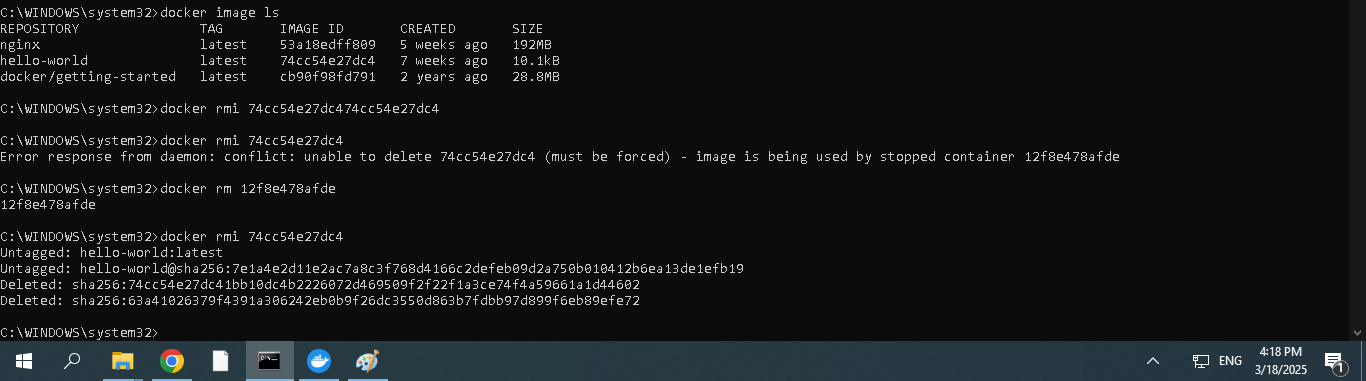
1. **docker rm <container\_id>**: Xóa một container đã dừng. Lệnh này giúp dọn dẹp các container không còn cần thiết, giải phóng tài nguyên hệ thống.



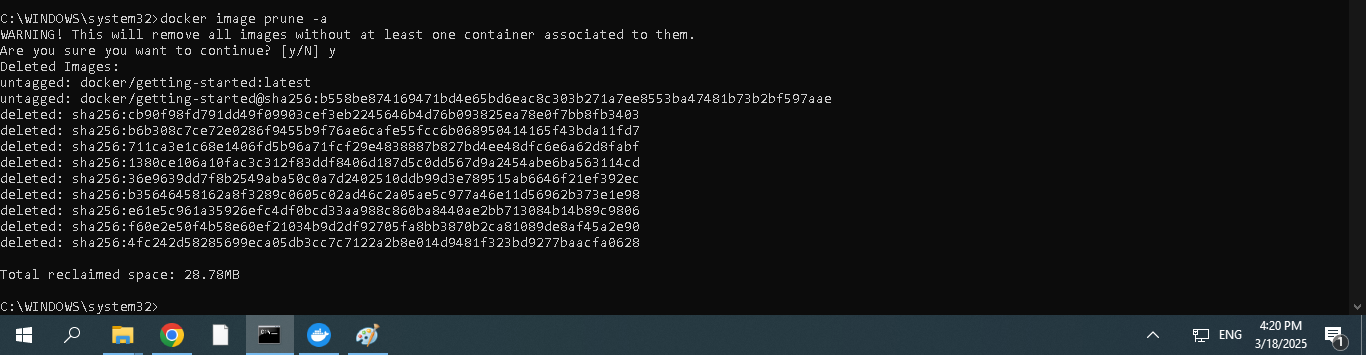
1. **docker container prune**: Xóa tất cả các container đã dừng để giải phóng dung lượng. Lệnh này hữu ích khi bạn muốn dọn dẹp nhanh chóng mà không cần xóa từng container một.



1. **docker rmi <image\_id>**: Xóa một image Docker khỏi máy local. Lệnh này giúp quản lý dung lượng lưu trữ bằng cách loại bỏ các image không còn sử dụng.



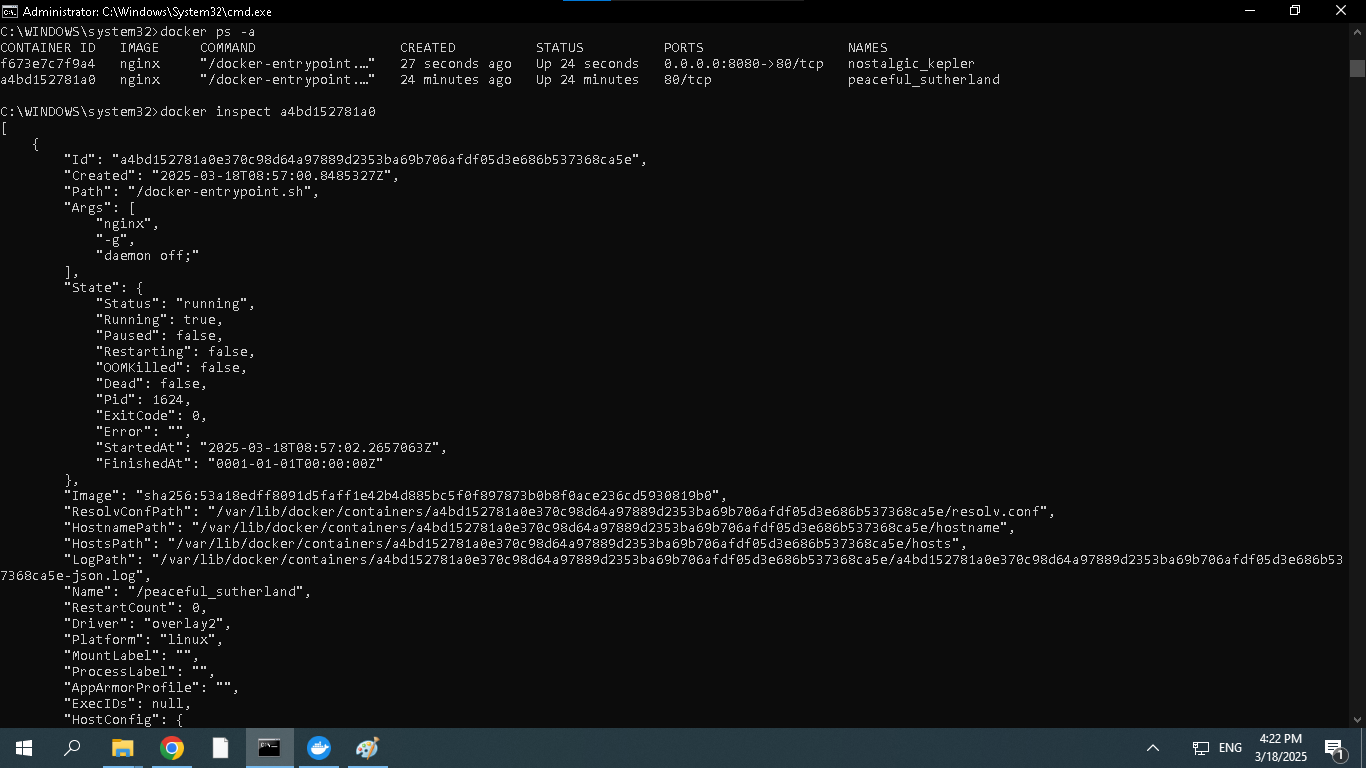
1. **docker image prune -a**: Xóa tất cả các image không sử dụng để giải phóng dung lượng. Lệnh này sẽ xóa các image không được gắn với bất kỳ container nào, kể cả các image chưa được gắn thẻ (dangling images).



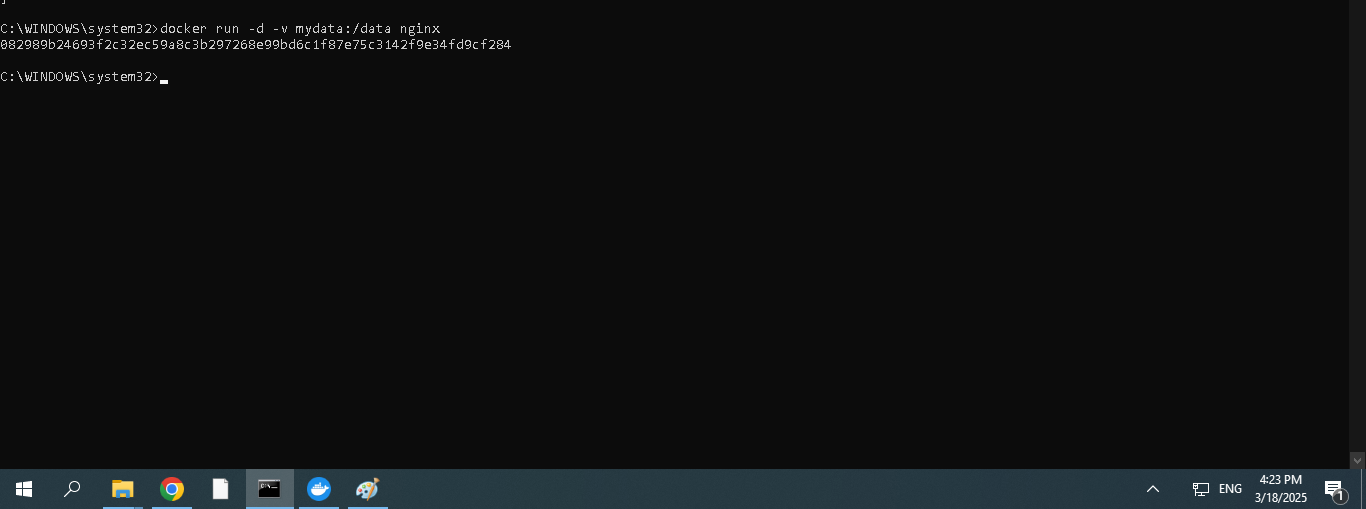
1. **docker run -d -p 8080:80 nginx**: Chạy một container Nginx trong chế độ nền và ánh xạ cổng 80 của container tới cổng 8080 trên máy host. Điều này cho phép bạn truy cập Nginx thông qua cổng 8080 trên máy host.



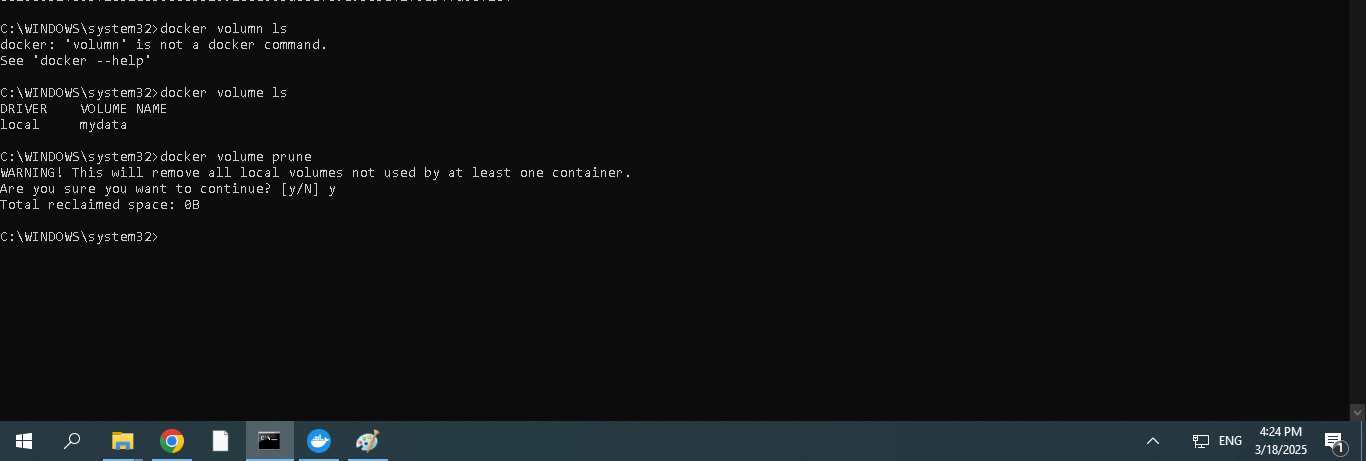
1. **docker inspect <container\_id>**: Hiển thị thông tin chi tiết về một container hoặc image. Lệnh này trả về dữ liệu ở định dạng JSON, bao gồm cấu hình, trạng thái, mạng và các thông tin khác.



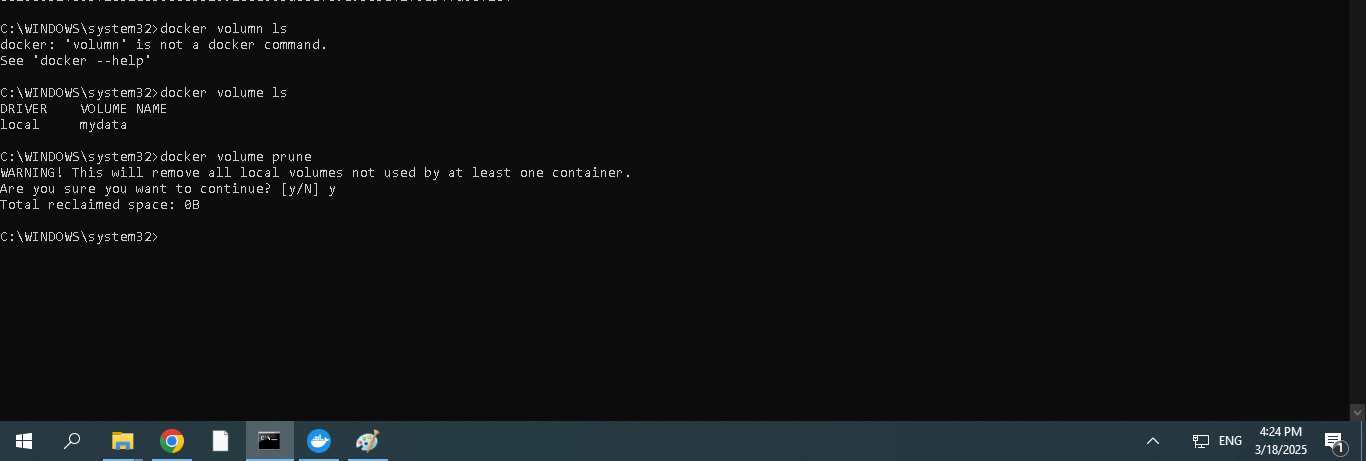
1. **docker run -d -v mydata:/data nginx**: Chạy một container Nginx trong chế độ nền và gắn (mount) volume mydata từ máy host vào thư mục /data trong container. Điều này giúp lưu trữ dữ liệu bên ngoài container, duy trì dữ liệu ngay cả khi container bị xóa.



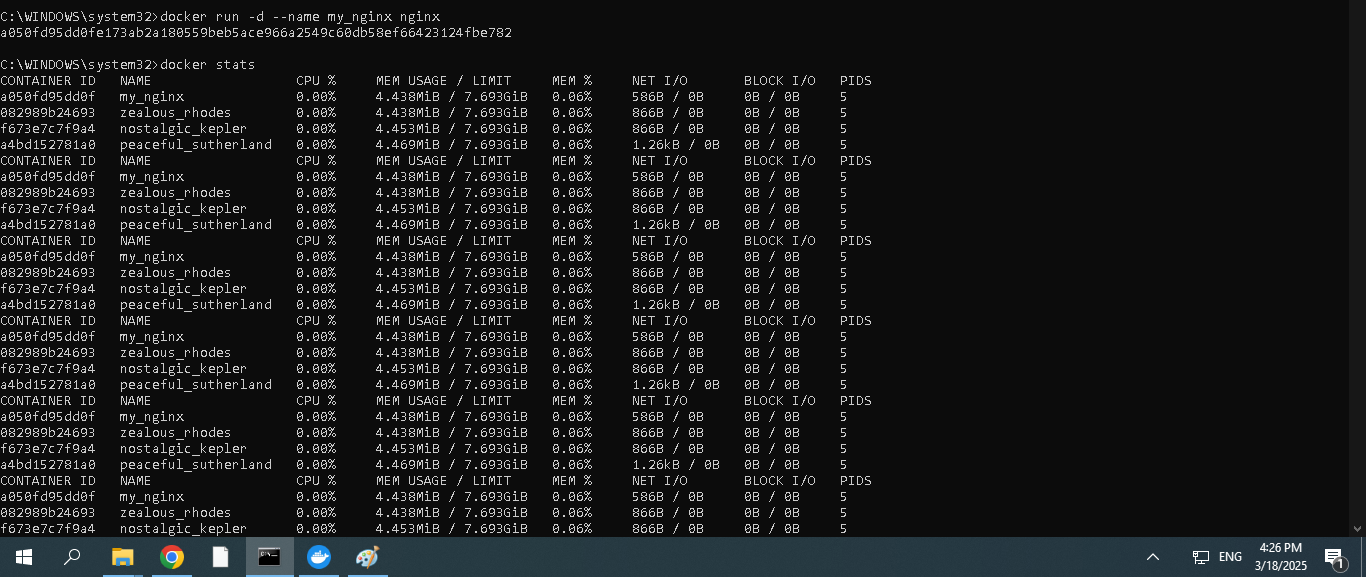
1. **docker volume ls**: Liệt kê tất cả các volume Docker hiện có. Lệnh này giúp bạn quản lý các volume được sử dụng để lưu trữ dữ liệu cho các container.



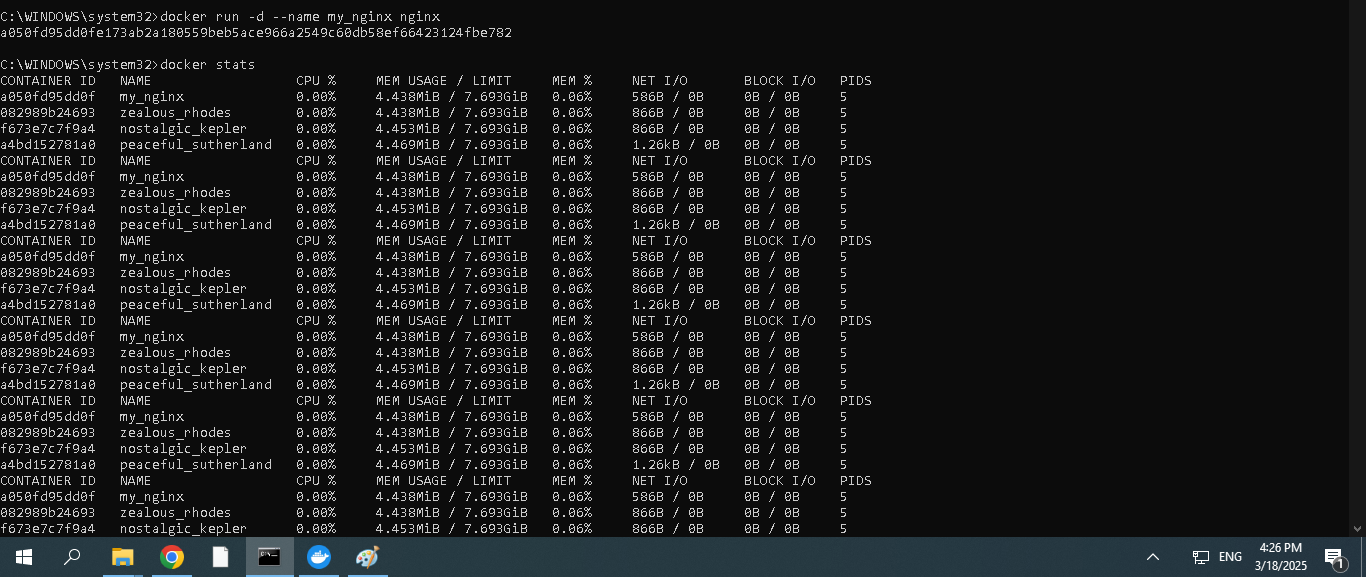
1. **docker volume prune**: Xóa tất cả các volume không sử dụng để giải phóng dung lượng. Lệnh này sẽ loại bỏ các volume không được gắn kết với bất kỳ container nào.



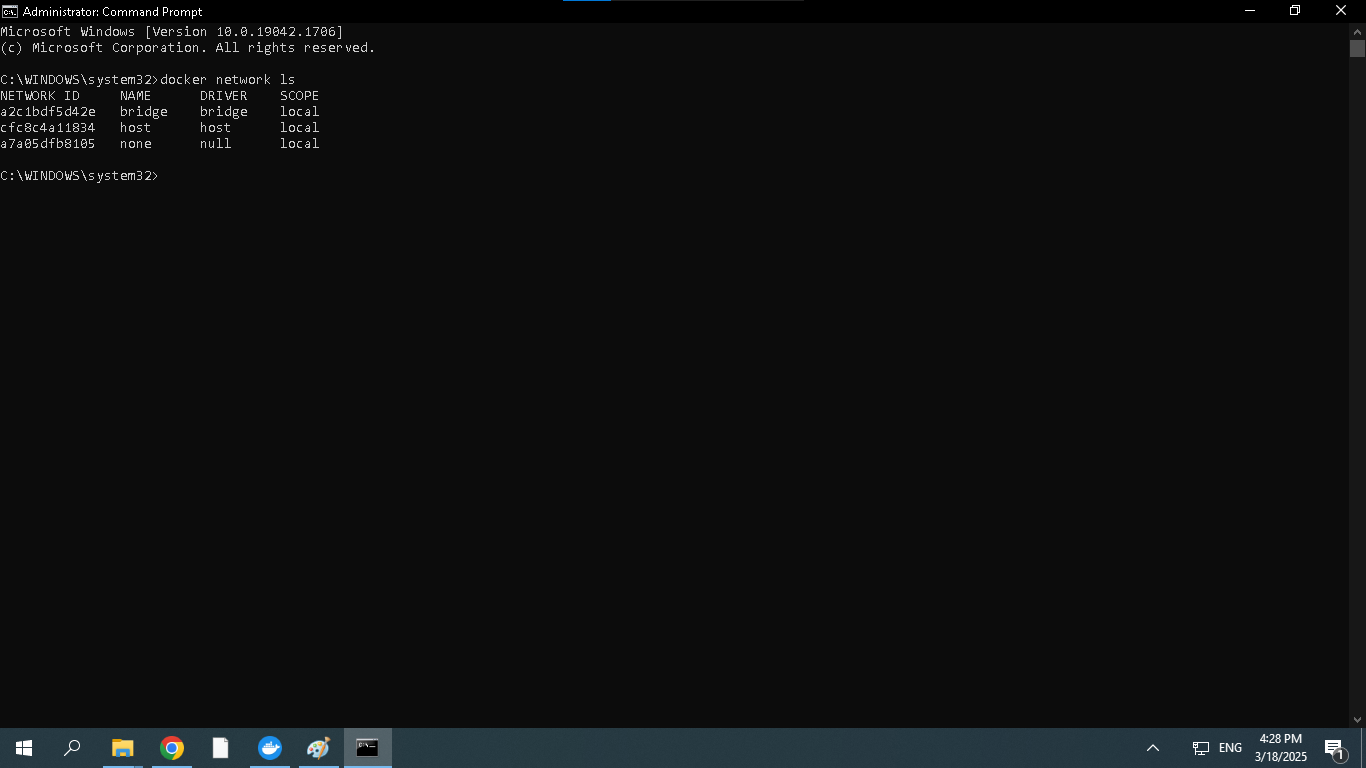
1. **docker run -d --name my\_nginx nginx**: Chạy một container Nginx trong chế độ nền và đặt tên cho container là my\_nginx. Việc đặt tên giúp dễ dàng quản lý và tham chiếu đến container sau này.



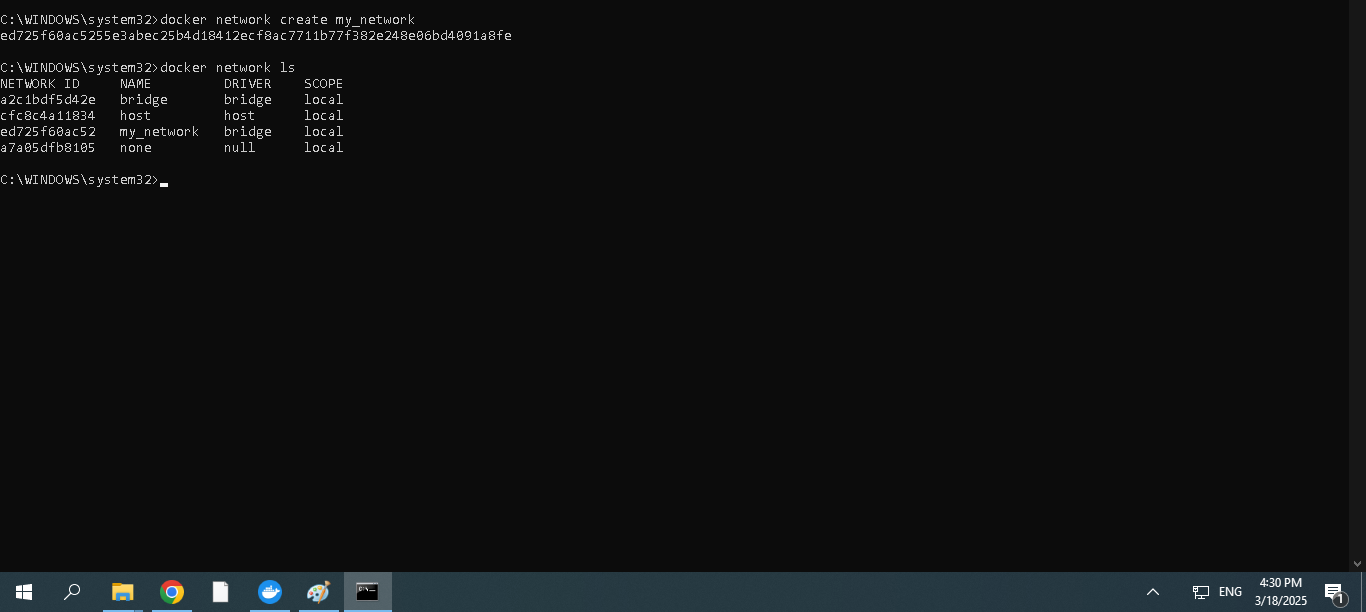
1. **docker stats**: Hiển thị thống kê tài nguyên (CPU, bộ nhớ, I/O) của các container đang chạy. Lệnh này giúp bạn giám sát hiệu suất và tài nguyên mà các container đang sử dụng.



1. **docker network ls**: Liệt kê tất cả các mạng (network) Docker hiện có. Lệnh này giúp bạn quản lý và xem các mạng mà container có thể kết nối.



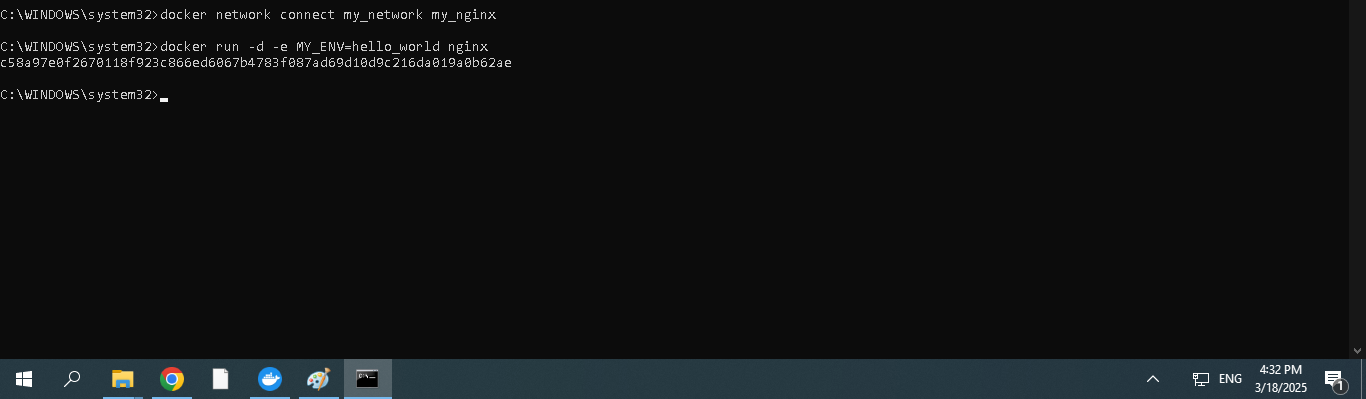
1. **docker network create my\_network**: Tạo một mạng Docker mới với tên my\_network. Mạng này giúp các container kết nối và giao tiếp với nhau một cách dễ dàng.



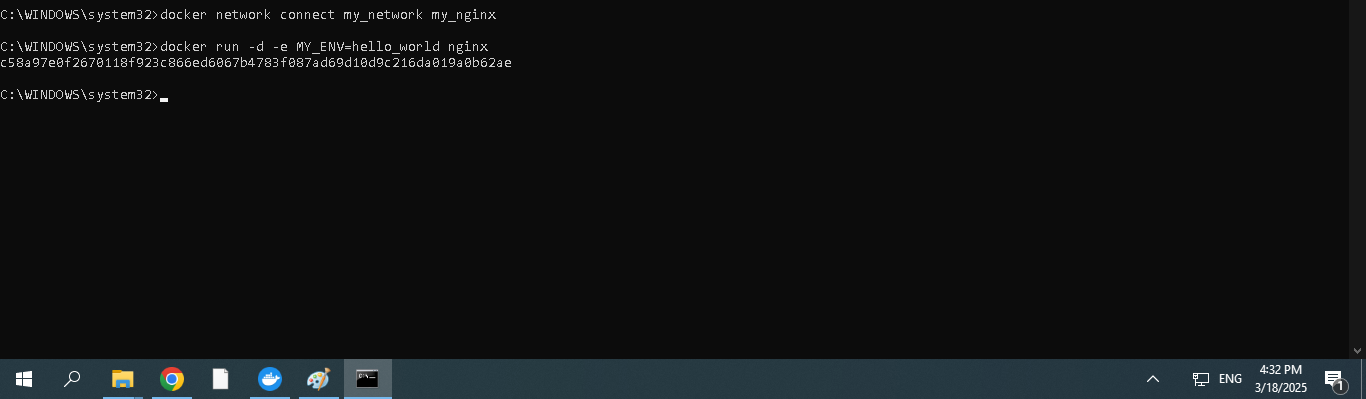
1. **docker run -d --network my\_network --name my\_container nginx**: Chạy một container Nginx trong chế độ nền, kết nối nó vào mạng my\_network và đặt tên là my\_container. Điều này cho phép container giao tiếp với các container khác trong cùng mạng.



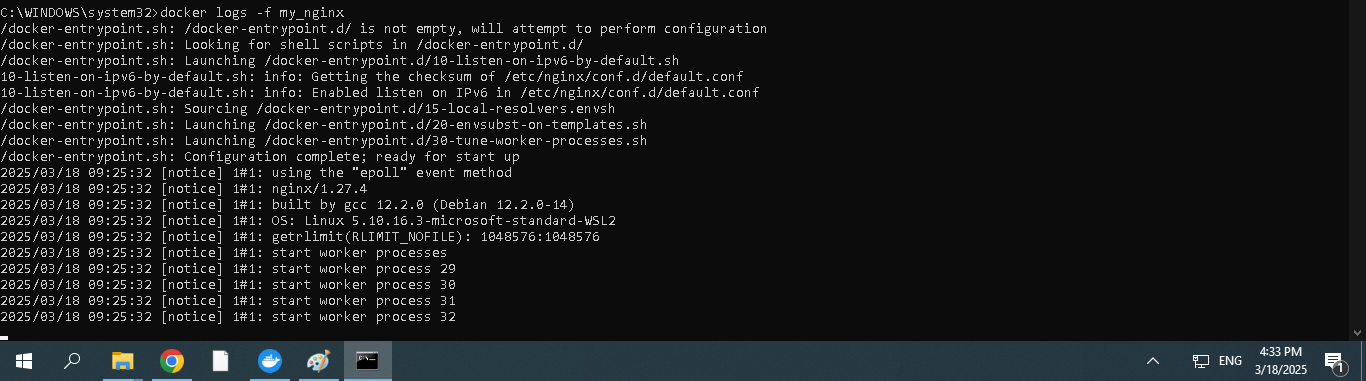
1. **docker network connect my\_network my\_nginx**: Kết nối container my\_nginx vào mạng my\_network. Lệnh này hữu ích khi bạn muốn thêm một container đang chạy vào một mạng cụ thể.



1. **docker run -d -e MY\_ENV=hello\_world nginx**: Chạy một container Nginx trong chế độ nền và thiết lập biến môi trường MY\_ENV với giá trị hello\_world. Việc này giúp cấu hình hành vi của ứng dụng bên trong container dựa trên các biến môi trường.



1. **docker logs -f my\_nginx**: Theo dõi log (nhật ký) của container my\_nginx theo thời gian thực. Lệnh này hữu ích để giám sát hoạt động và debug ứng dụng đang chạy trong container.



29. **FROM nginx COPY index.html /usr/share/nginx/html/index.html**: Định nghĩa một Dockerfile bắt đầu từ image Nginx và sao chép file index.html từ máy chủ vào thư mục /usr/share/nginx/html/index.html trong container. Điều này cho phép bạn tùy chỉnh nội dung được phục vụ bởi Nginx.

30. **docker build -t my\_nginx\_image .**: Xây dựng một image Docker mới từ Dockerfile trong thư mục hiện tại và gán thẻ (tag) là my\_nginx\_image. Lệnh này tạo ra một image có thể được sử dụng để khởi chạy các container với cấu hình và ứng dụng cụ thể.

31. **docker run -d -p 8080:80 my\_nginx\_image**: Chạy một container từ image my\_nginx\_image trong chế độ nền và ánh xạ cổng 80 của container tới cổng 8080 trên máy host. Điều này cho phép truy cập ứng dụng web trong container thông qua cổng 8080 trên máy host.