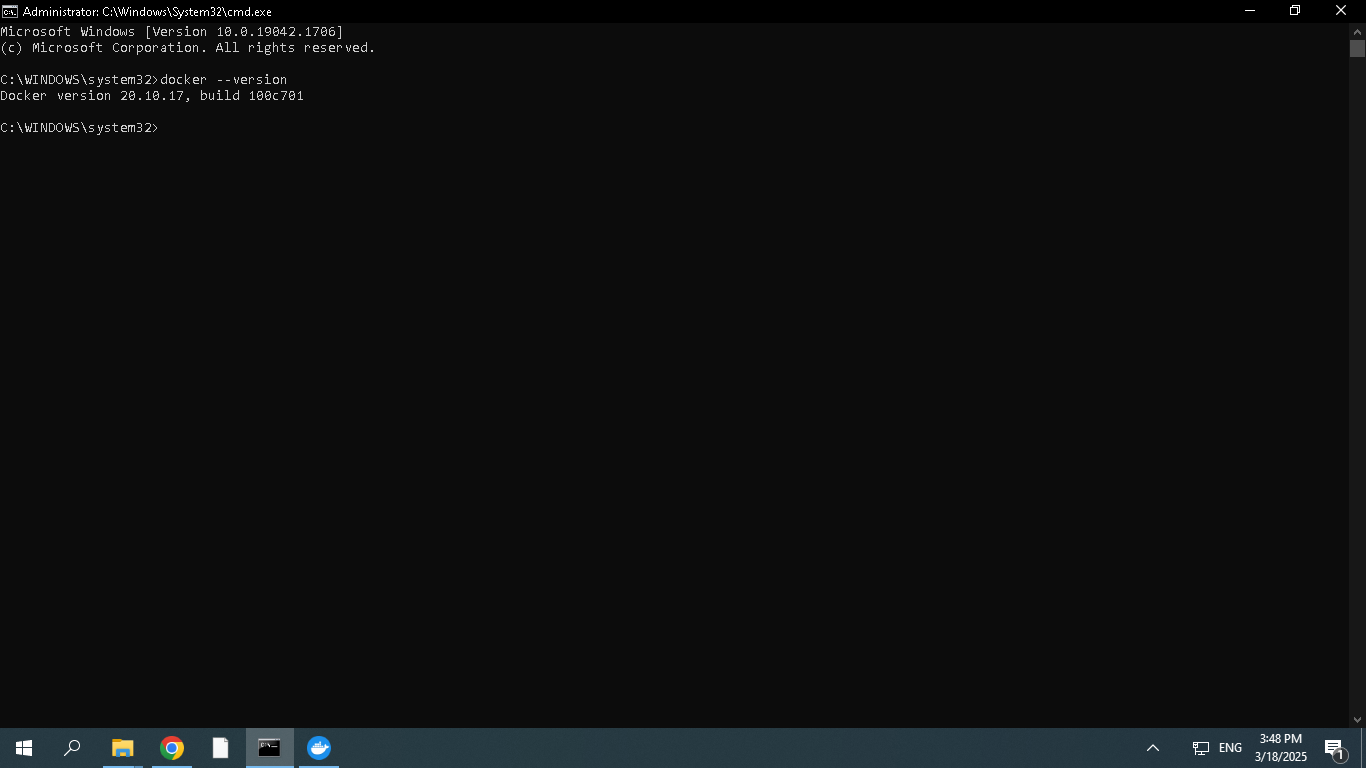
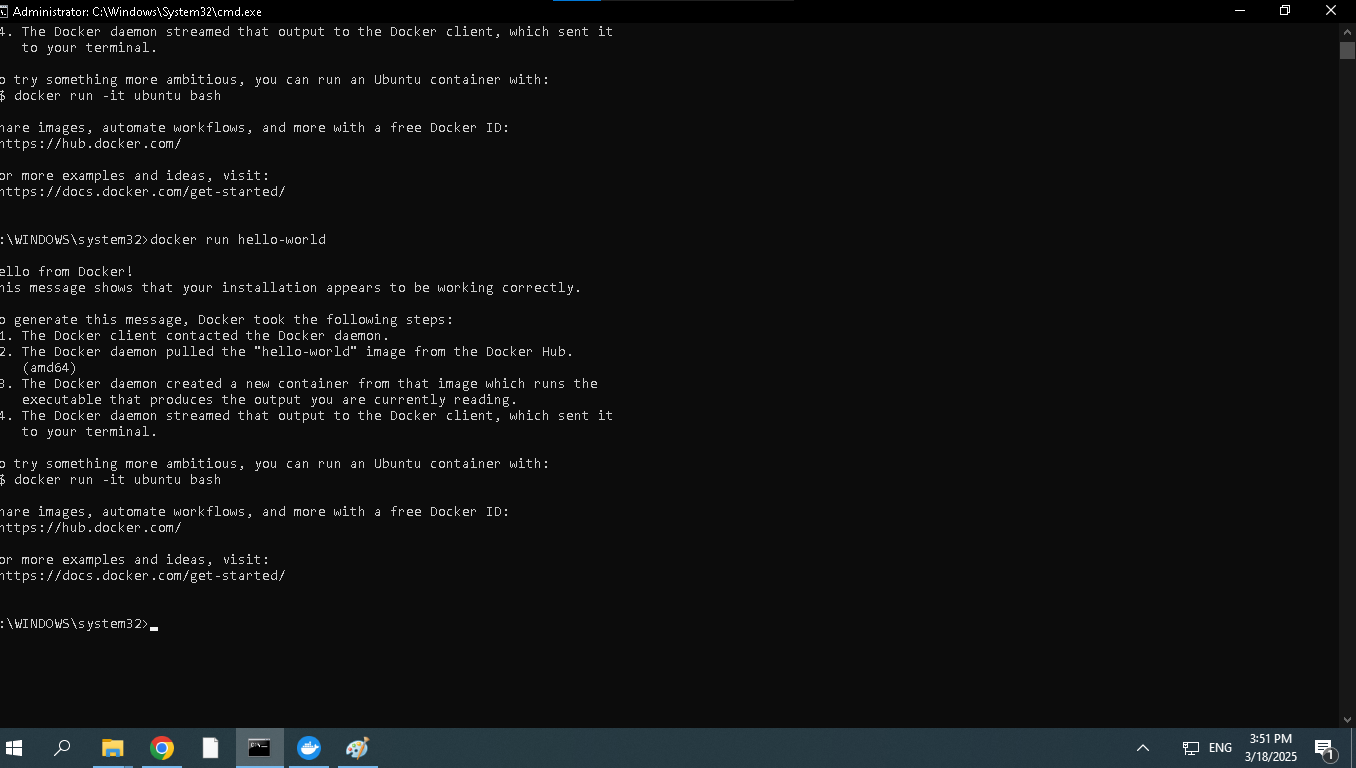
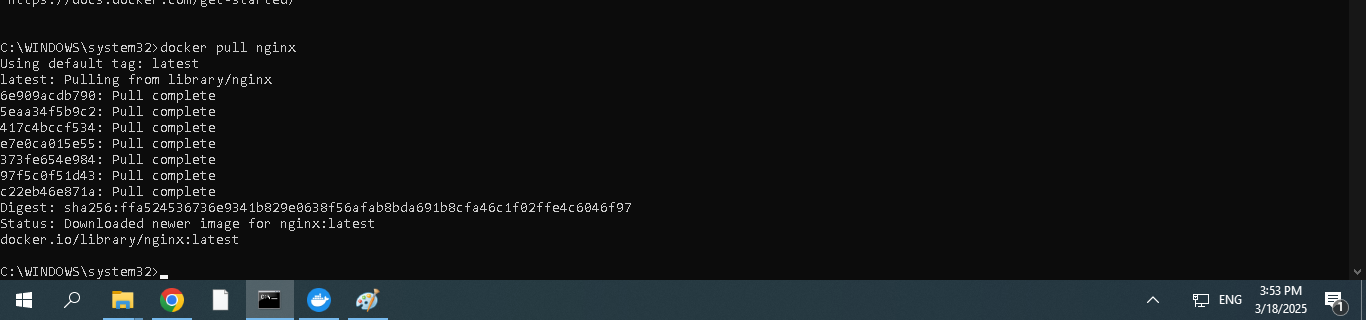
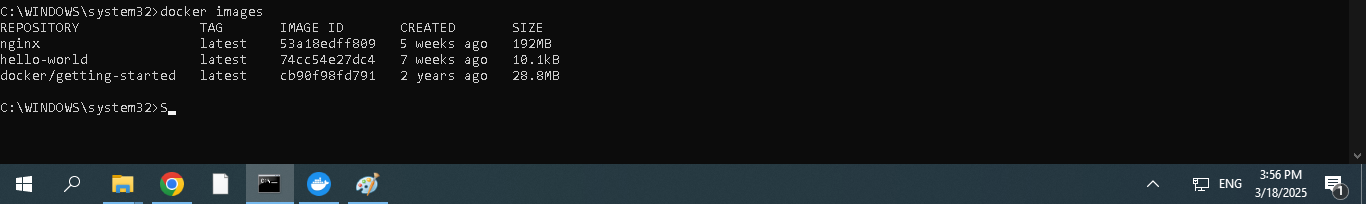
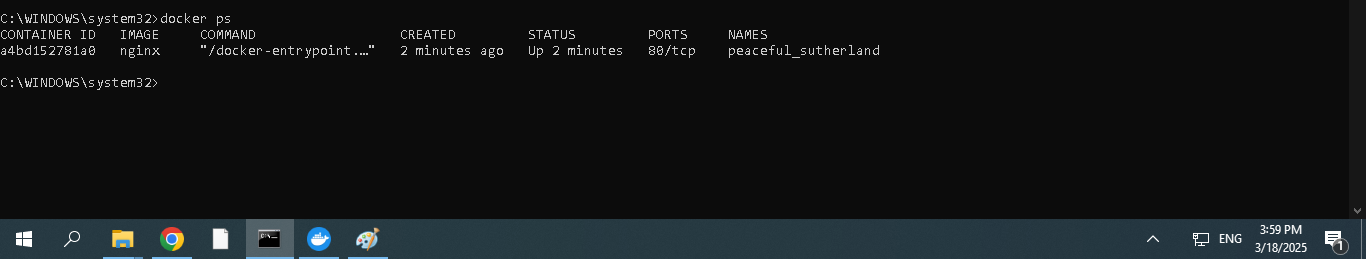
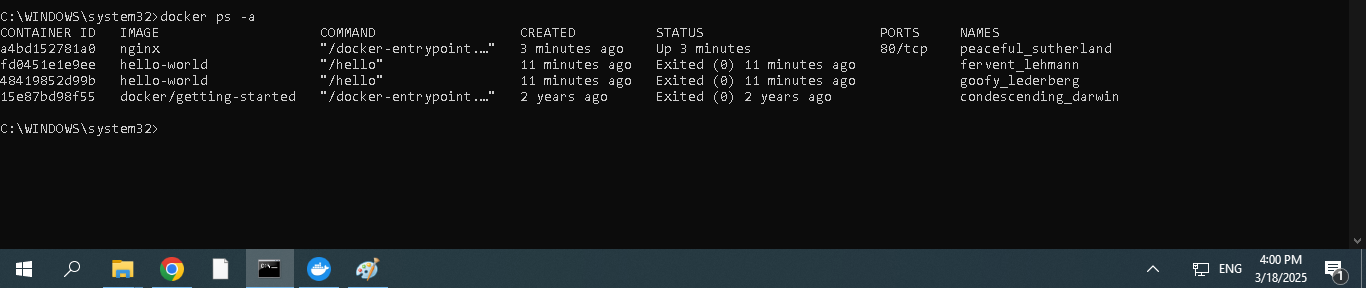
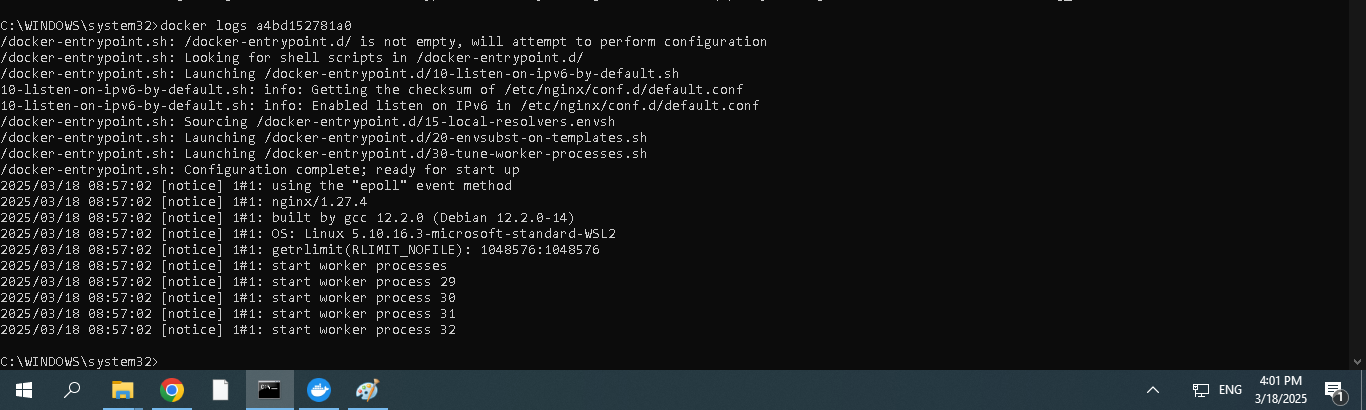
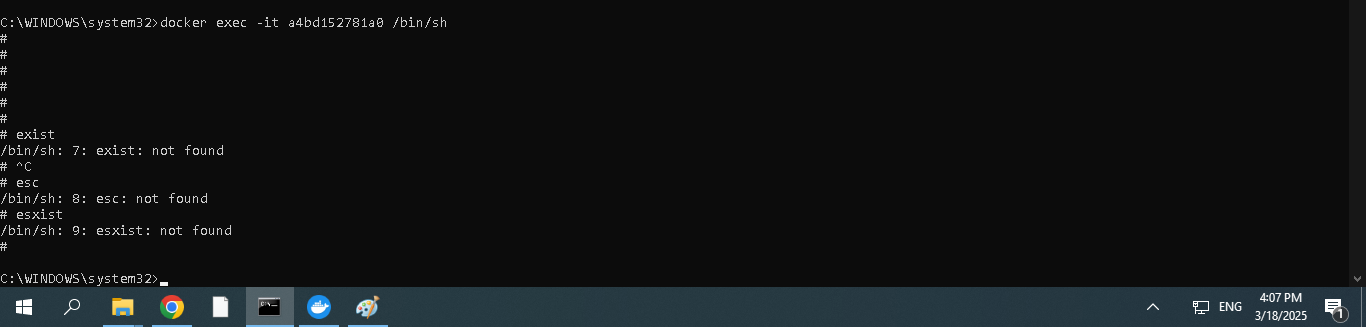
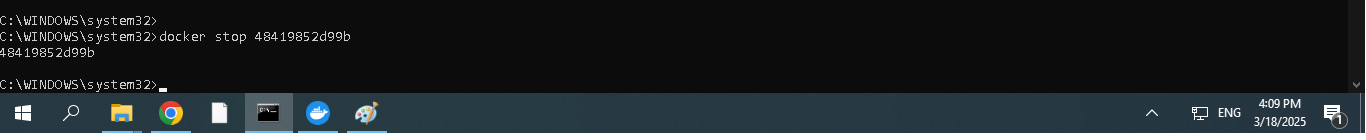
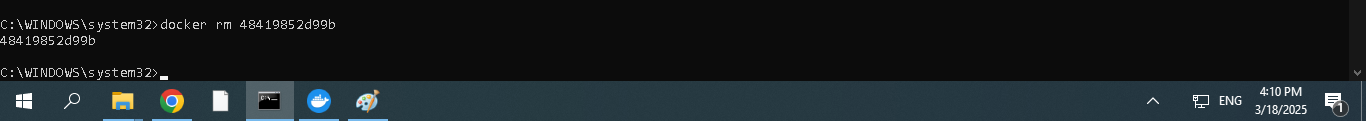
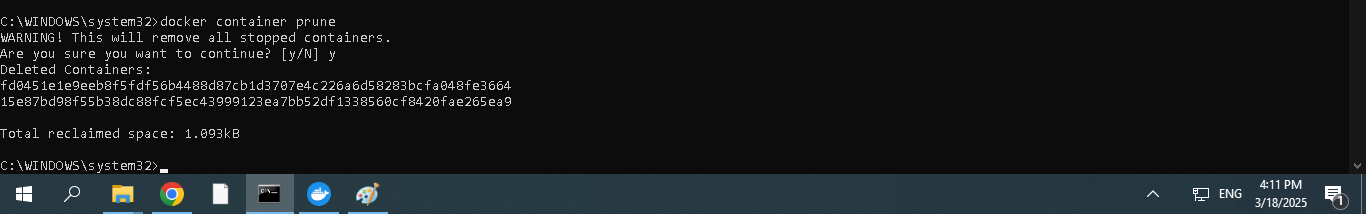
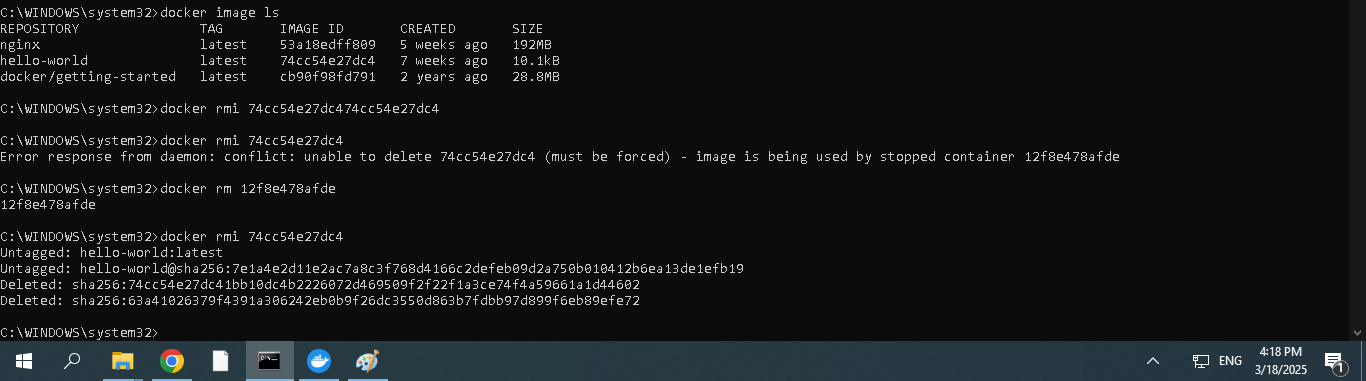
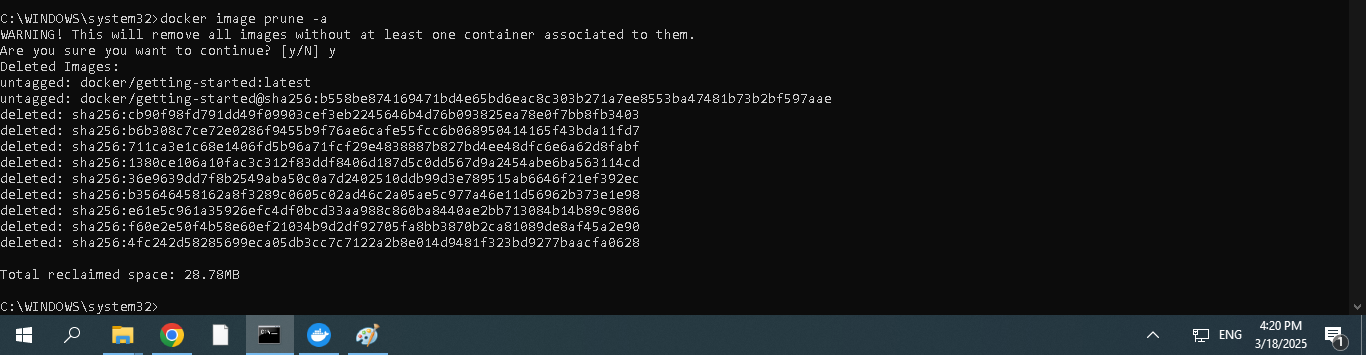
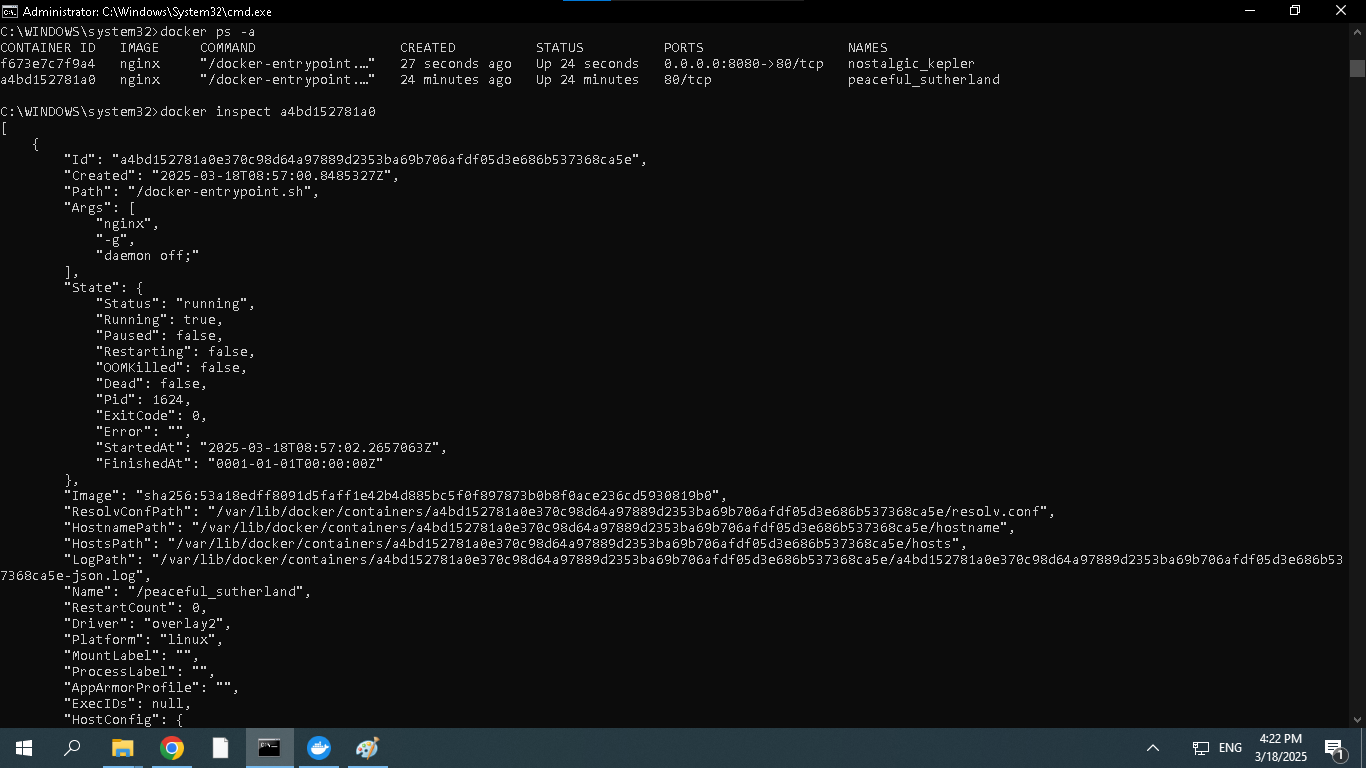
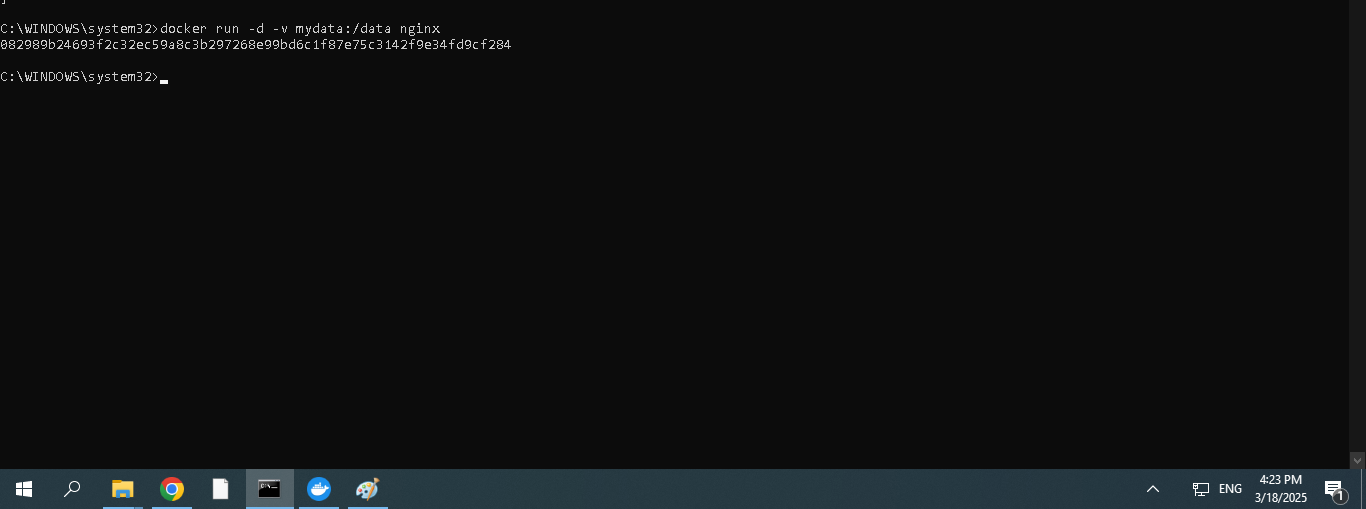
Nguyễn Minh Nhật - 21102931

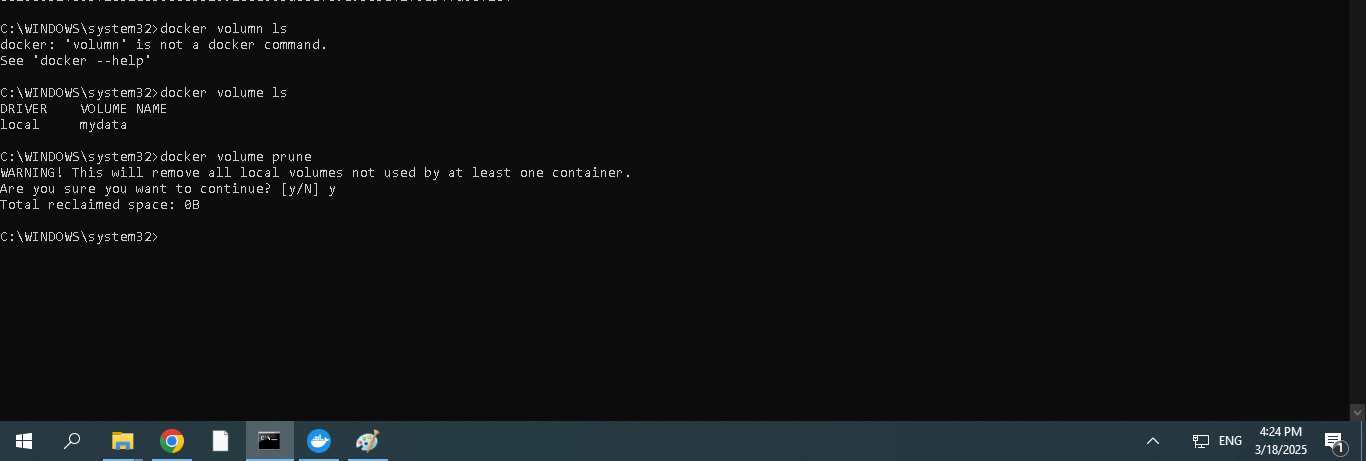
1. **docker --version**: Lệnh này kiểm tra phiên bản Docker được cài đặt trên hệ thống.****
2. **docker run hello-world**: Lệnh này tải và chạy container hello-world để kiểm tra xem Docker có hoạt động bình thường không.Container này in ra một thông báo đơn giản nếu Docker hoạt động đúng.



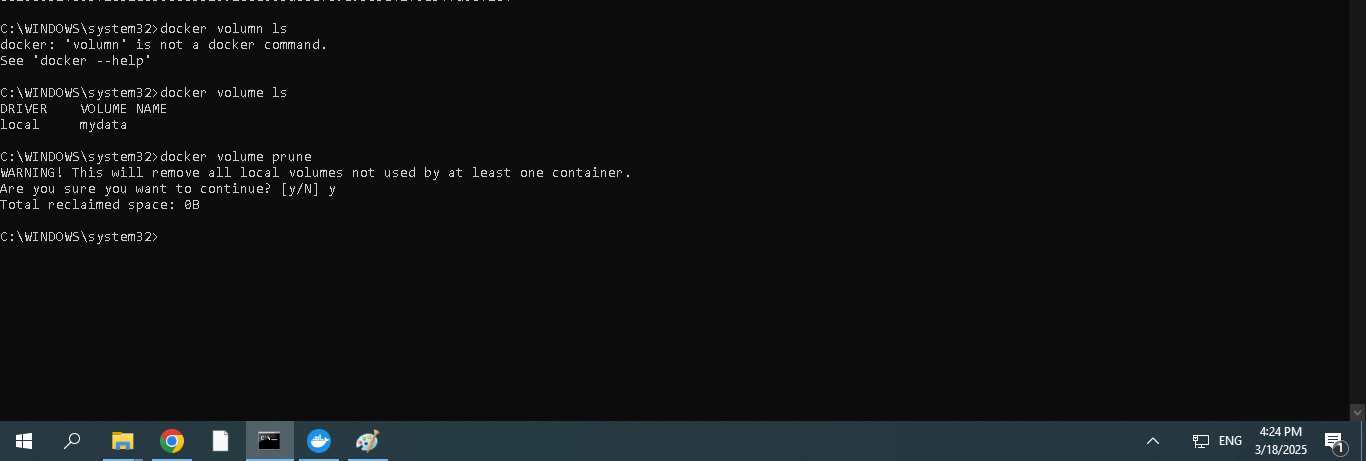
1. **docker pull nginx**: Lệnh này tải image nginx từ Docker Hub về máy. Nếu image đã có sẵn, Docker kiểm tra và tải phiên bản mới nhất (nếu có).
2. **docker images**: Hiển thị danh sách các image đã tải về máy



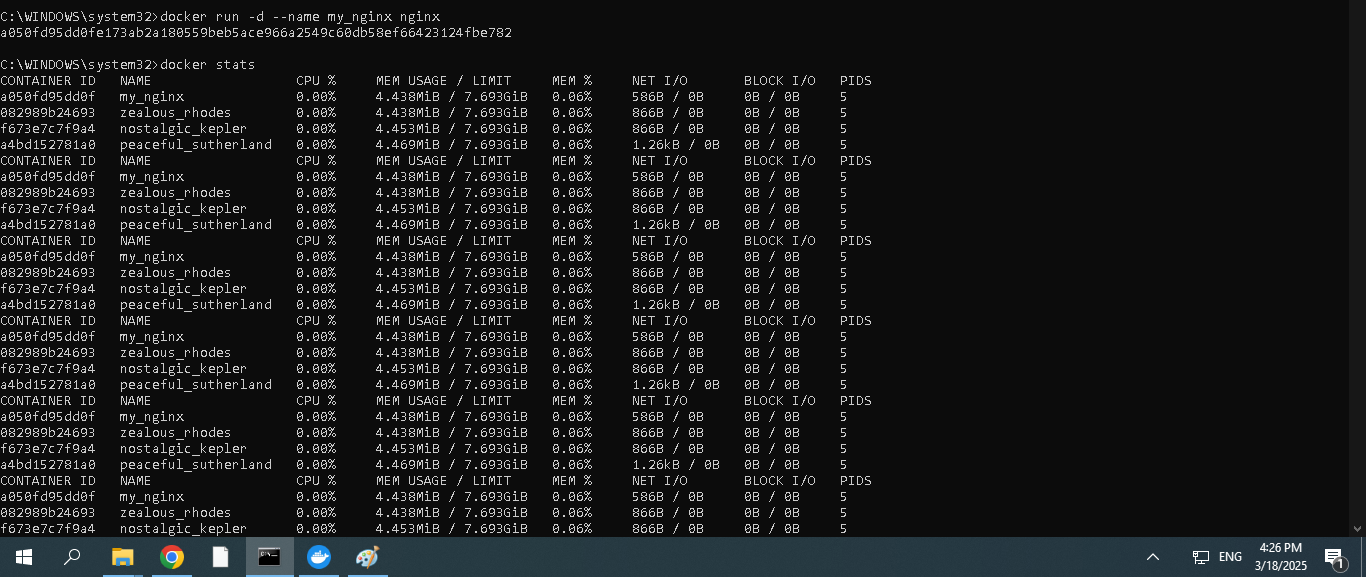
1. **docker run -d nginx**: -d: Chạy container ở chế độ nền (detached mode). Container chạy Nginx, nhưng không gán tên hoặc ánh xạ cổng.
2. **docker ps**: Hiển thị danh sách các container đang chạy.
3. **docker ps -a**: Hiển thị tất cả container, kể cả container đã dừng.
4. **docker logs <container\_id>**: Hiển thị log của container, giúp debug nếu gặp lỗi.
5. **docker exec -it <container\_id> /bin/sh**: Cho phép tương tác với container.
6. **docker stop <container\_id>**: Dừng contai ner đang chạy.
7. **docker restart <container\_id>**: Dừng và khởi động lại container.
8. **docker rm <container\_id>**: Xóa một container đã dừng
9. **docker container prune**: Xóa tất cả các container đã 
10. **docker rmi <image\_id>**: Xóa một image Docker khỏi máy local
11. **docker image prune -a**: Xóa tất cả các image không sử dụng để giải phóng dung lượng
12. **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. 
13. **docker inspect <container\_id>**: Hiển thị thông tin chi tiết về một container hoặc image. 
14. **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. 
15. **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.



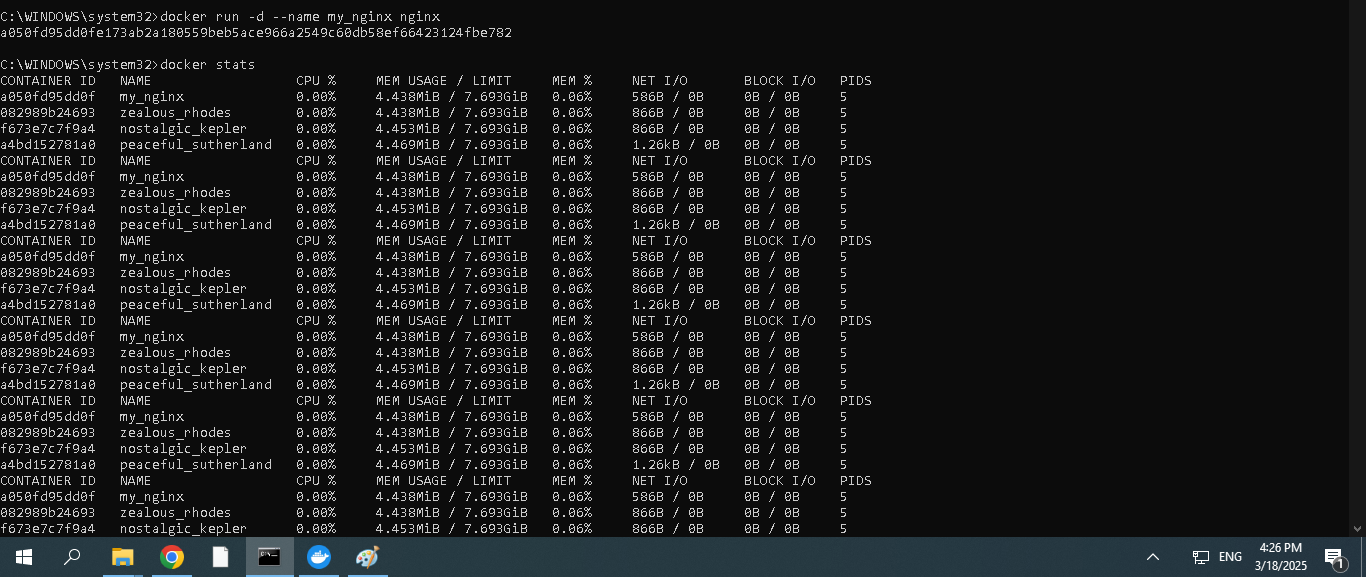
1. **docker volume prune**: Xóa tất cả các volume không sử dụng để giải phóng dung lượng. Lệnh này sẽ loại bỏ các volume không được gắn kết với bất kỳ container nào.



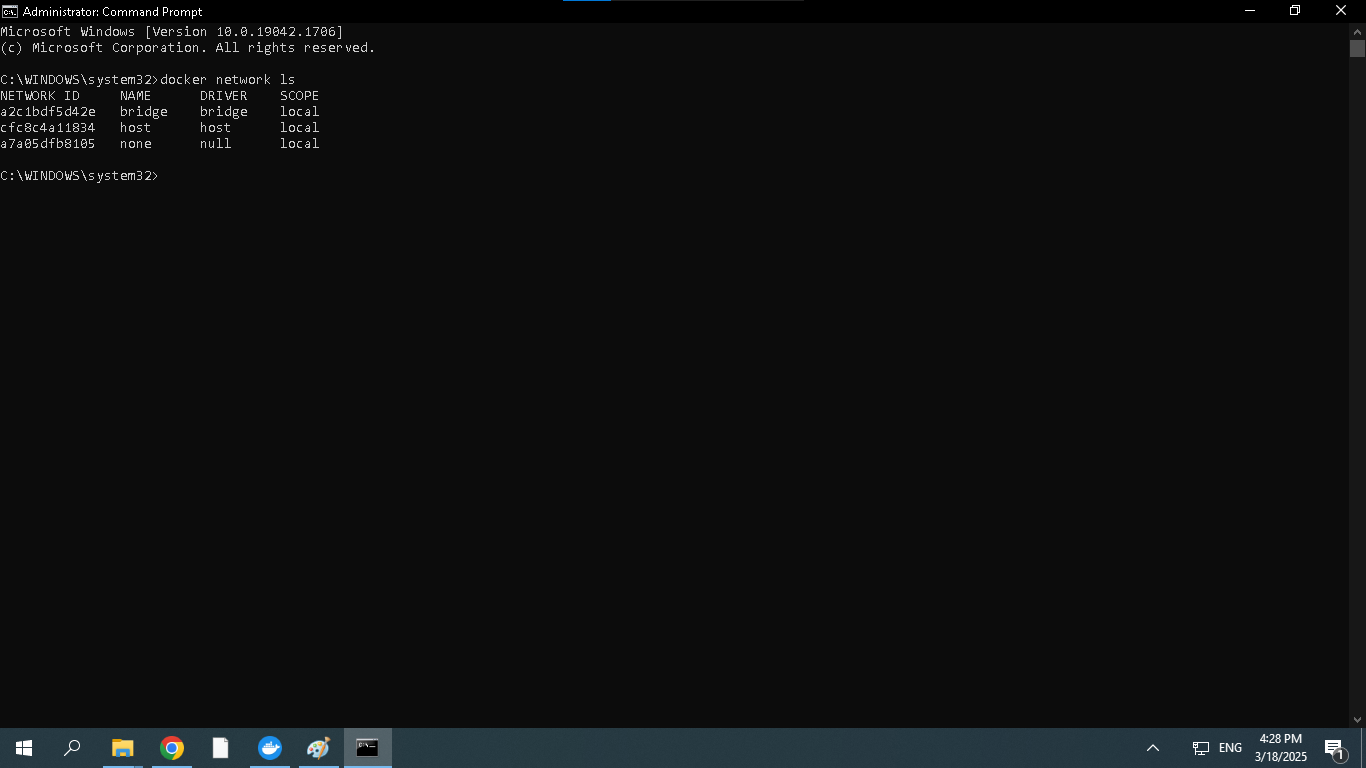
1. **docker run -d --name my\_nginx nginx**: Chạy một container Nginx trong chế độ nền và đặt tên cho container là my\_nginx. Việc đặt tên giúp dễ dàng quản lý và tham chiếu đến container sau này.



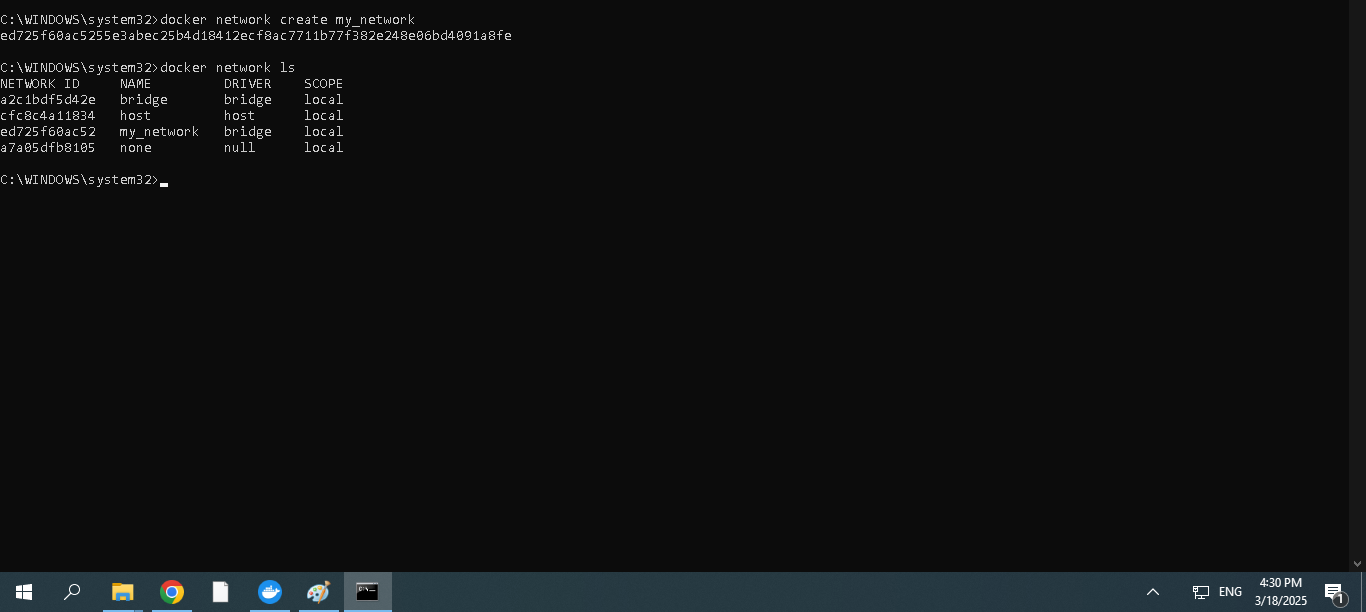
1. **docker stats**: Hiển thị thống kê tài nguyên (CPU, bộ nhớ, I/O) của các container đang chạy. Lệnh này giúp bạn giám sát hiệu suất và tài nguyên mà các container đang sử dụng.



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.



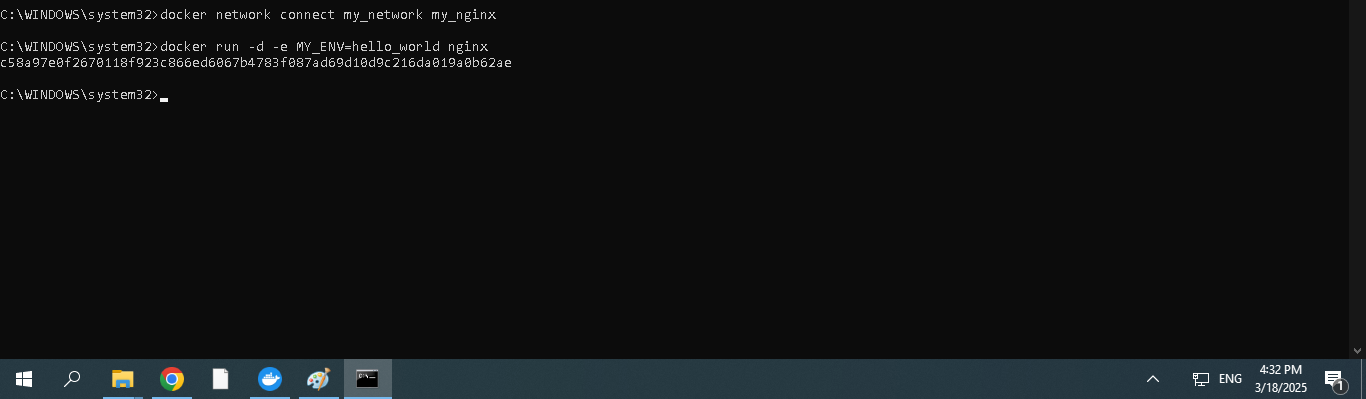
1. **docker network create my\_network**: Tạo một mạng Docker mới với tên my\_network. Mạng này giúp các container kết nối và giao tiếp với nhau một cách dễ dàng.



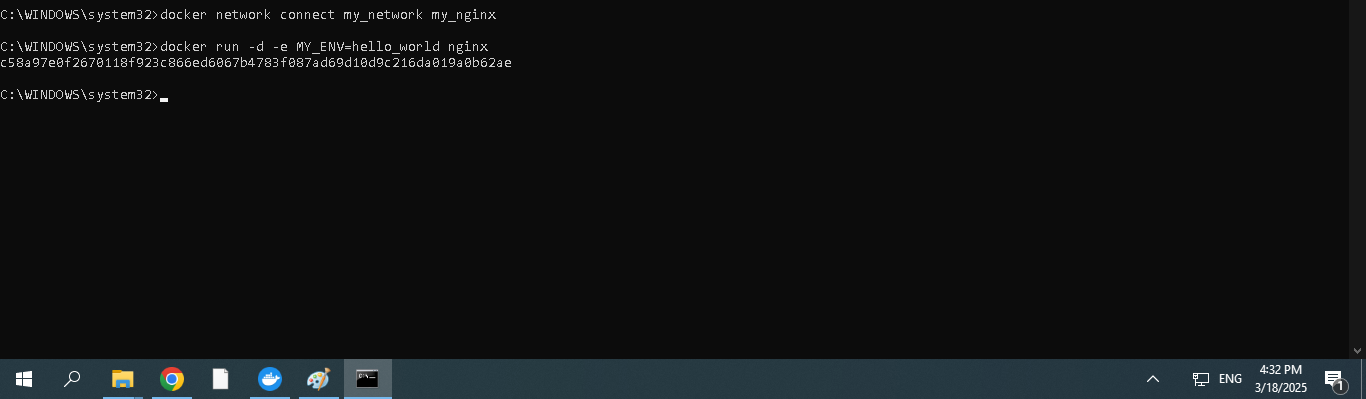
1. **docker run -d --network my\_network --name my\_container nginx**: Chạy một container Nginx trong chế độ nền, kết nối nó vào mạng my\_network và đặt tên là my\_container. Điều này cho phép container giao tiếp với các container khác trong cùng mạng.



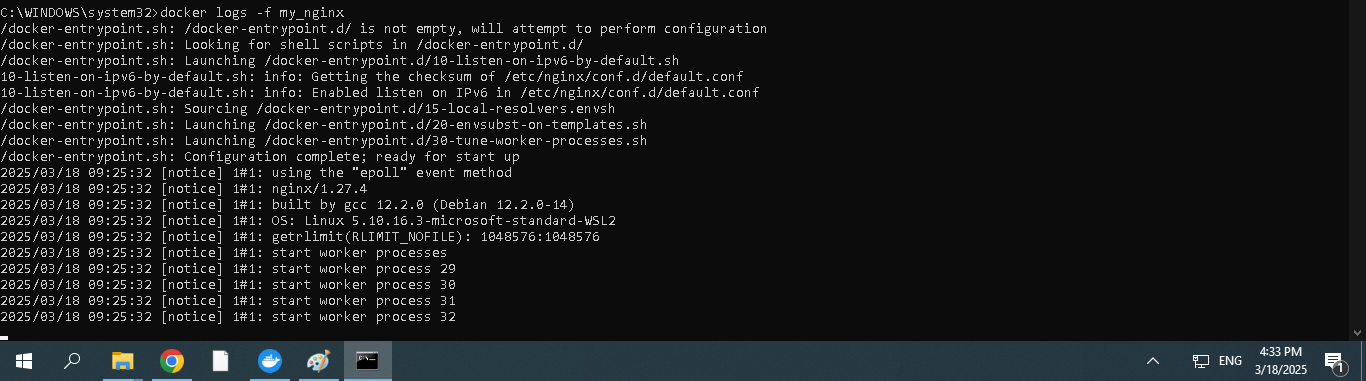
1. **docker network connect my\_network my\_nginx**: Kết nối container my\_nginx vào mạng my\_network. Lệnh này hữu ích khi bạn muốn thêm một container đang chạy vào một mạng cụ thể.



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.