**BÁO CÁO THỰC HÀNH**

**LAB 04:** Tìm hiểu về Docker

**Họ và tên:** Lê Hoàng Khang **Mã số sinh viên:** 21083791

# Phần 1: Các lệnh cơ bản thao tác với Docker

## docker --version

A screenshot of a computer

AI-generated content may be incorrect.

**Mô tả :** Kiểm tra phiên bản của Docker hiện đang được cài đặt trên hệ thống.

**Kết quả :** Hiển thị phiên bản Docker

## docker run hello-world

A screenshot of a computer screen

AI-generated content may be incorrect.

**Mô tả :** Chạy một container từ image hello-world. Đây là một image đơn giản dùng để kiểm tra xem Docker có hoạt động đúng không.

**Quá trình :**

Nếu image hello-world chưa tồn tại cục bộ, Docker sẽ tải nó từ Docker Hub.

Sau đó, container sẽ chạy và in ra thông báo "Hello from Docker!".

## docker pull nginx

A computer screen with white text

AI-generated content may be incorrect.

**Mô tả :** Tải xuống (pull) image nginx từ Docker Hub mà không chạy nó.

**Kết quả :** Image nginx sẽ được lưu trữ cục bộ trên máy tính của bạn.

## docker images

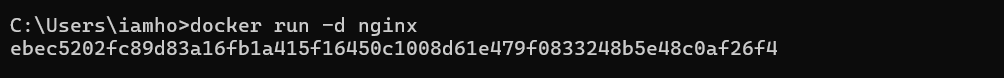
A computer screen shot of a computer code

AI-generated content may be incorrect.

**Mô tả :** Liệt kê tất cả các Docker images hiện có trên hệ thống.

**Kết quả :** Hiển thị danh sách các images với thông tin như REPOSITORY, TAG, IMAGE ID, CREATED, và SIZE.

## docker run -d nginx

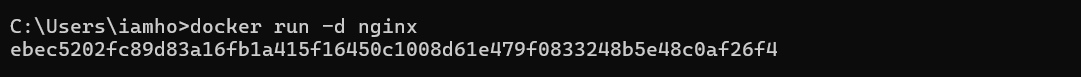


**Mô tả**: Chạy một container từ image nginx ở chế độ nền (detached mode).

**Tham số -d :** Container sẽ chạy ở chế độ nền mà không hiển thị log ra terminal.

**Kết quả:** Một container mới sẽ được khởi tạo và chạy ở nền.

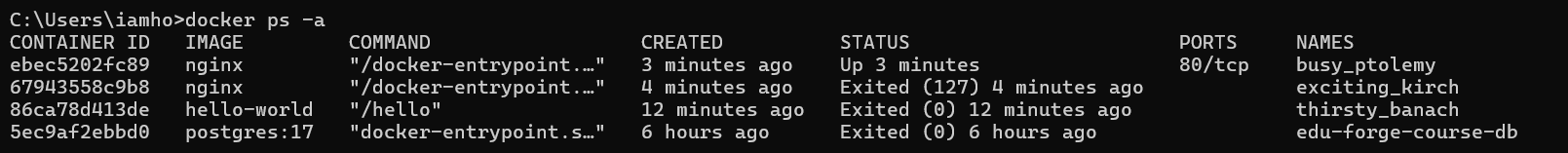
## docker ps



**Mô tả:** Liệt kê các container đang chạy hiện tại.

**Kết quả:** Hiển thị danh sách các container đang hoạt động với thông tin như CONTAINER ID, IMAGE, STATUS, PORTS, và NAMES.

## docker ps -a



**Mô tả:** Liệt kê tất cả các container (bao gồm cả đang chạy và đã dừng).

**Kết quả:** Hiển thị danh sách đầy đủ các container trên hệ thống.

## docker logs <container\_id>

A black screen with white text

AI-generated content may be incorrect.

**Mô tả:** Xem log đầu ra của một container cụ thể (được xác định bằng <container\_id> hoặc tên).

**Kết quả:** Hiển thị các thông điệp log mà container đã ghi lại trong quá trình hoạt động.

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

A screen shot of a computer

AI-generated content may be incorrect.

**Mô tả:** Truy cập vào shell của một container đang chạy để thực hiện các lệnh tương tác.

**Tham số:**

-it: Kết hợp giữa chế độ tương tác (-i) và thiết bị đầu cuối (-t).

/bin/sh: Chỉ định shell sẽ sử dụng bên trong container.

**Kết quả:** Bạn có thể thực thi các lệnh trực tiếp trong contai

## docker stop <container\_id>

A black screen with white text

AI-generated content may be incorrect.

**Mô tả:** Dừng một container đang chạy.

**Kết quả:** Container sẽ dừng hoạt động nhưng vẫn tồn tại trên hệ thống.

## docker restart <container\_id>

A black screen with white text

AI-generated content may be incorrect.

**Mô tả:** Khởi động lại một container.

**Kết quả:** Container sẽ được dừng và khởi động lại ngay lập tức.

## docker rm <container\_id>

A black screen with white text

AI-generated content may be incorrect.

**Mô tả:** Xóa một container đã dừng.

**Kết quả:** Container bị xóa hoàn toàn khỏi hệ thống.

## docker container prune

A black screen with white text

AI-generated content may be incorrect.

**Mô tả:** Xóa tất cả các container đã dừng.

**Kết quả:** Giải phóng tài nguyên bằng cách xóa các container không còn cần thiết.

## docker rmi <image\_id>

A screenshot of a computer screen

AI-generated content may be incorrect.

## docker image prune -a

A screenshot of a computer screen

AI-generated content may be incorrect.

**Mô tả:** Xóa một image cụ thể.

**Lưu ý:** Image chỉ có thể bị xóa nếu không có container nào đang sử dụng nó.

## docker run -d -p 8080:80 nginx

A computer screen with white text

AI-generated content may be incorrect.

**Mô tả:** Chạy một container từ image nginx ở chế độ nền và ánh xạ cổng 8080 trên host tới cổng 80 của container.

**Kết quả:** Bạn có thể truy cập ứng dụng Nginx bằng cách mở trình duyệt và nhập địa chỉ <http://localhost:8080>.

A screenshot of a computer

AI-generated content may be incorrect.

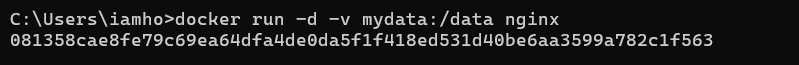
## docker inspect <container\_id>



**Mô tả:** Xem thông tin chi tiết về một container (hoặc image).

**Kết quả:** Hiển thị thông tin cấu hình, mạng, và trạng thái của container dưới dạng JSON.

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



**Mô tả:** Chạy một container từ image nginx và gắn một volume có tên mydata vào đường dẫn /data trong container.

**Kết quả:** Dữ liệu trong thư mục /data của container sẽ được lưu trữ bền vững trong volume mydata.

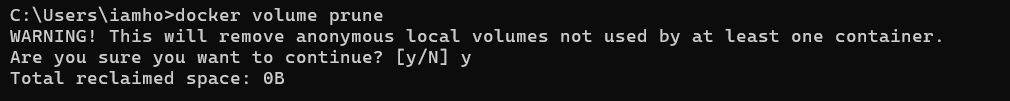
## docker volume ls

## A black background with white text AI-generated content may be incorrect.

**Mô tả:** Liệt kê tất cả các volumes hiện có trên hệ thống.

**Kết quả:** Hiển thị danh sách các volumes cùng với thông tin liên quan.

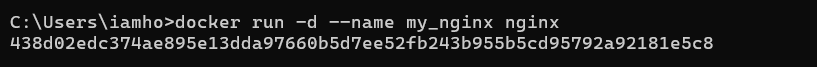
## docker volume prune



**Mô tả:** Xóa tất cả các volumes không được sử dụng bởi bất kỳ container nào.

**Kết quả:** Giải phóng không gian đĩa bằng cách loại bỏ các volumes dư thừa.

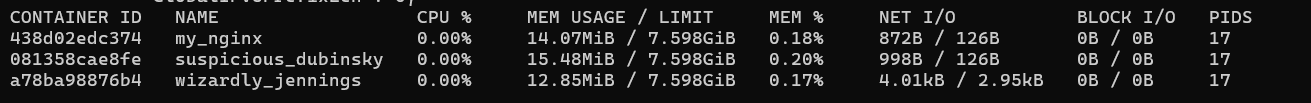
## docker run -d --name my\_nginx nginx



**Mô tả:** Chạy một container từ image nginx ở chế độ nền và đặt tên cho container là my\_nginx.

**Kết quả:** Container có tên dễ nhớ để quản lý.

## docker stats



**Mô tả:** Hiển thị thông tin thời gian thực về việc sử dụng tài nguyên (CPU, memory, network, I/O) của các container đang chạy.

**Kết quả:** Bảng thống kê tài nguyên theo thời gian thực.

## docker network ls

A black screen with white text

AI-generated content may be incorrect.

**Mô tả:** Liệt kê tất cả các mạng Docker hiện có.

**Kết quả:** Hiển thị danh sách các mạng cùng với thông tin như NETWORK ID, NAME, và DRIVER.

## docker network create my\_network



**Mô tả:** Tạo một mạng Docker mới có tên my\_network.

**Kết quả:** Mạng mới được tạo để kết nối các container.

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



**Mô tả:** Chạy một container từ image nginx và gắn nó vào mạng my\_network với tên my\_container.

**Kết quả:** Container có thể giao tiếp với các container khác trong cùng mạng.

## docker network connect my\_network my\_nginx

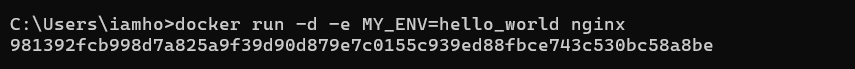
A computer screen with white text

AI-generated content may be incorrect.

**Mô tả:** Kết nối một container hiện có (my\_nginx) vào mạng my\_network.

**Kết quả:** Container có thể giao tiếp với các thành viên khác trong mạng.

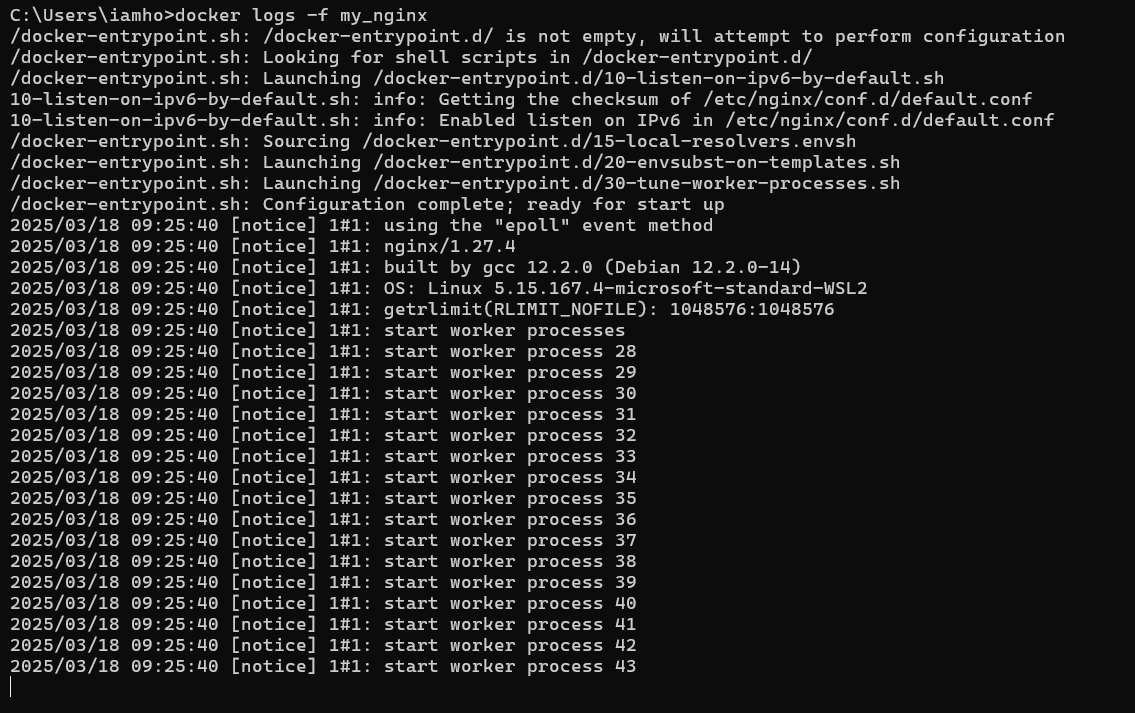
## docker run -d -e MY\_ENV=hello\_world nginx



**Mô tả:** Chạy một container từ image nginx và thiết lập biến môi trường MY\_ENV với giá trị hello\_world.

**Kết quả:** Biến môi trường có thể được sử dụng bên trong container.

## docker logs -f my\_nginx



**Mô tả:** Xem log của container my\_nginx theo thời gian thực.

Tham số -f : Theo dõi log liên tục (giống như lệnh tail -f).

## Tạo file với nội dung:

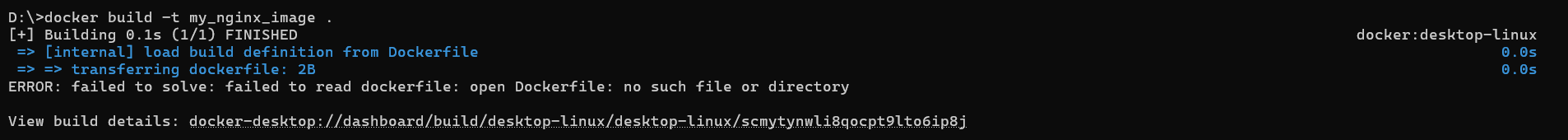
FROM nginx

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

A black screen with white text

AI-generated content may be incorrect.

## docker build -t my\_nginx\_image .



Build docker image với nội dung vừa tạo

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

A black screen with white text

AI-generated content may be incorrect.

Chạy docker image vừa tạo ở bước trên

**Bài 1:** Tạo Dockerfile chạy một ứng dụng Node.js đơn giản

Các bước thực hiện:

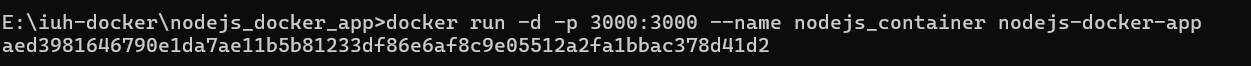
**Bước 1:** Tạo file app.js và Dockerfile

**Bước 2:** Chạy lệnh docker build -t nodejs-docker-app .

A screenshot of a computer program

AI-generated content may be incorrect.

**Bước 3:** Chạy lệnh docker run -d -p 3000:3000 --name nodejs\_container nodejs-docker-app



**Bước 4:** Truy cập <http://localhost:3000> để xem kết quả

Bài 2: Tạo Dockerfile chạy một ứng dụng Python Flask

Yêu cầu:

Viết Dockerfile để chạy một ứng dụng Flask hiển thị "Hello, Docker Flask!" trên cổng 5000.

Sử dụng python:3.9 làm base image.

**Bài 2:** Tạo Dockerfile chạy một ứng dụng Python Flask

Bước 1: Tạo các file cần thiết

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Bước 2:** Viết ứng dụng và cấu hình docker file

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**Bước 3:** Chạy lệnh build ứng dụng và start container

docker build -t flask-app .

A screen shot of a computer

AI-generated content may be incorrect.

docker run -p 5000:5000 flask-app

A screen shot of a black screen

AI-generated content may be incorrect.

(Cảnh báo để cho chúng ta biết việc sử dụng lệnh này không nên dùng trong môi trường Product)

Kết quả

A screenshot of a computer

AI-generated content may be incorrect.

**Bài 3: Chạy 1 ứng dụng React đơn giản bằng docker file**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Bước 2:** Tạo docker file và cấu hình cho dockerfile

A screenshot of a computer program

AI-generated content may be incorrect.

**Bước 3:** Build và run ứng dụng

**A screen shot of a computer

AI-generated content may be incorrect.**

A screenshot of a computer program

AI-generated content may be incorrect.

Kết quả:

A screen shot of a computer

AI-generated content may be incorrect.

**Bài 4: Tạo ứng dụng với Nginx**

**Bước 1:** Tạo thư mục và các file cần thiết

**Bước 2:** Tạo file index.html và cấu hình docker file

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Bước 3**: Build ứng dụng và start container.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

**Kết quả:**

A close up of a text

AI-generated content may be incorrect.

**Bài 5: Tạo ứng dụng Go**

**Bước 1:** Tạo file và cấu hình docker file

A screen shot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**Bước 2:** Build và start

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

**Kết quả:**

A close-up of a computer screen

AI-generated content may be incorrect.**Bài 6:** Sử dụng Multi-stage Build trong Dockerfile

Viết Dockerfile để build một ứng dụng Node.js với hai stage:

Stage 1: Dùng node:18 để build code.

Stage 2: Dùng node:18-alpine để chạy ứng dụng đã build.

**Bước 1:** Khởi tạo ứng dụng ExpressJS

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Bước 2: Viết nội dung cho index.js

A screen shot of a computer program

AI-generated content may be incorrect.

**Bước 3:** Viết Dockerfile với Multi-stage Build

A screenshot of a computer program

AI-generated content may be incorrect.

**Bước 4:** Build và Start

docker build -t multi-node-app .

A screenshot of a computer program

AI-generated content may be incorrect.

docker run -p 3000:3000 multi-node-app

A black screen with white text

AI-generated content may be incorrect.

**Kết quả:**

**A close-up of a sign

AI-generated content may be incorrect.**

**Bài 7:** Sử dụng biến môi trường trong Dockerfile

**Bước 1:** Tạo ứng dụng python

A screen shot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Bước 2:** Build và Start

A screen shot of a computer

AI-generated content may be incorrect.



**Kết quả:**