Họ tên: Nguyễn Thanh Nhứt

MSSV: 21140001

Môn: Kiến trúc và Thiết kế phần mềm

Thực hành Tuần 6

Link github: <https://github.com/NguyenThanhNhut13/KTTKPM/tree/main/Tuan6>

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

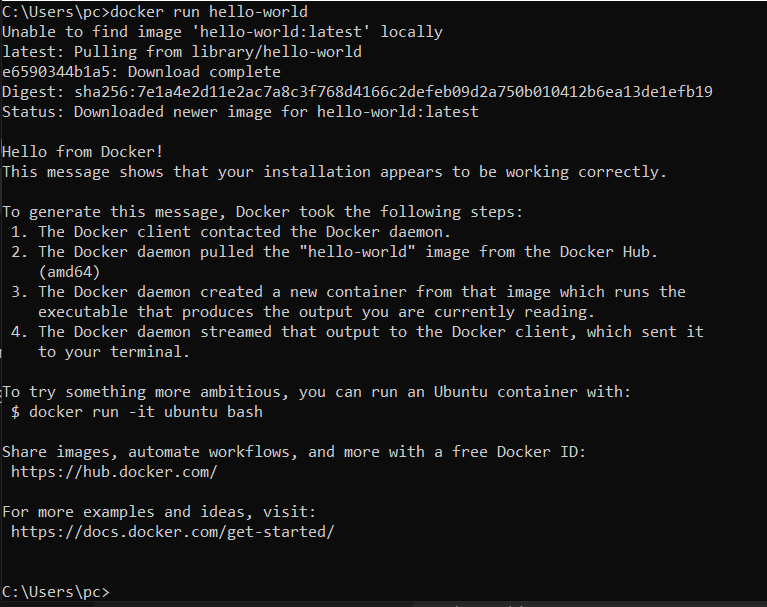
1. docker –version

Hiển thị phiên bản Docker đang được cài đặt trên máy.



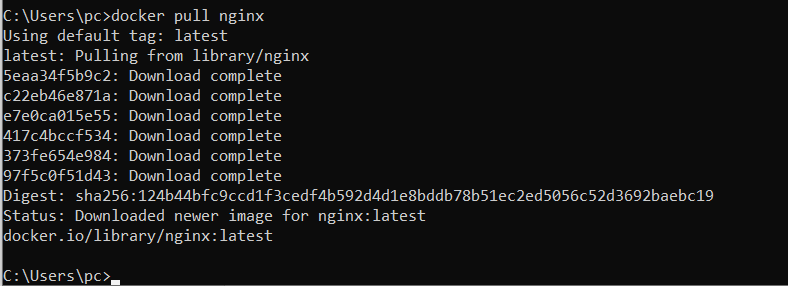
1. docker run hello-world

Tải xuống (nếu chưa có) và chạy container hello-world để kiểm tra Docker hoạt động đúng hay không.



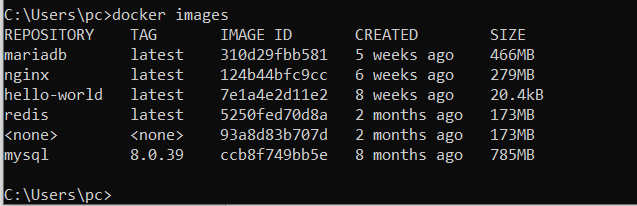
1. docker pull nginx

Tải xuống (pull) image nginx từ Docker Hub về máy.



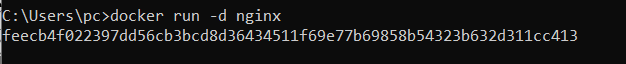
1. docker images

Hiển thị danh sách các image Docker có trên máy.



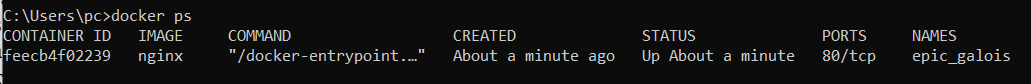
1. docker run -d nginx

Chạy một container từ image nginx ở chế độ nền (-d tức là detached mode).



1. docker ps

Hiển thị danh sách các container đang chạy.



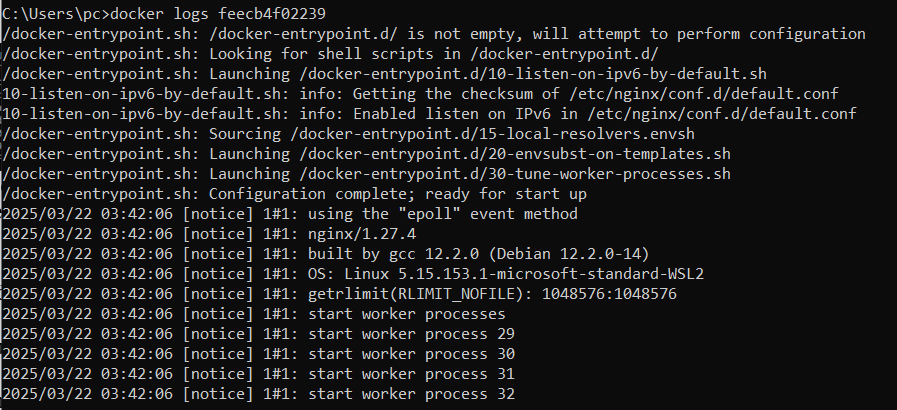
1. docker ps -a

Hiển thị danh sách tất cả container (bao gồm container đã dừng).



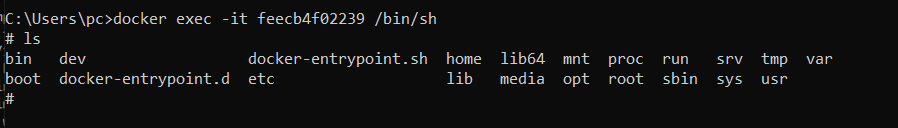
1. docker logs <container\_id>

Xem log của một container cụ thể.



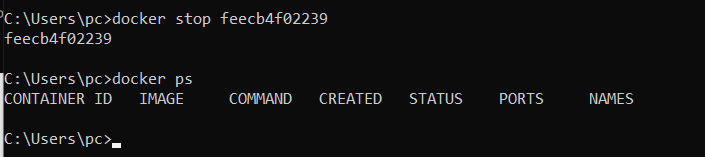
1. docker exec -it <container\_id> /bin/sh

Mở terminal (shell) vào trong container để thao tác.



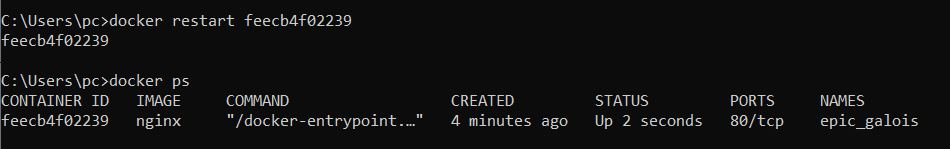
1. docker stop <container\_id>

Dừng container theo container\_id.



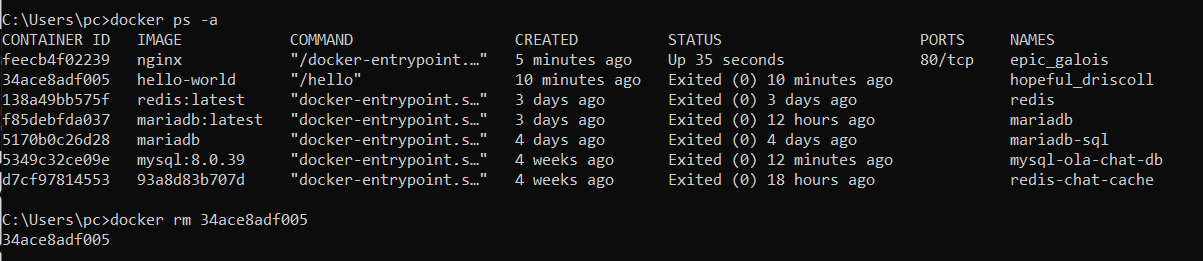
1. docker restart <container\_id>

Khởi động lại container.



1. docker rm <container\_id>

Xóa container đã dừng.



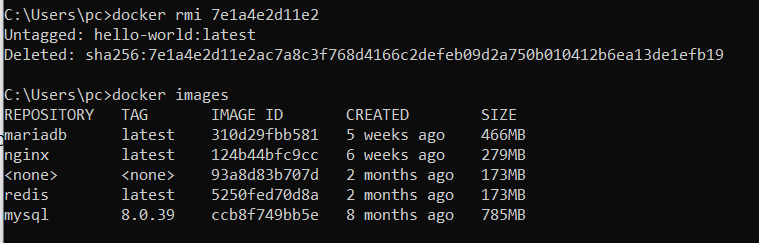
1. docker container prune

Xóa tất cả container đã dừng.

Không xóa vì nó sẽ mất hết container cần thiết đang sử dụng

1. docker rmi <image\_id>

Xóa một image theo image\_id.



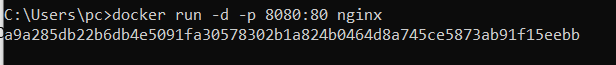
1. docker image prune -a

Xóa tất cả image không sử dụng.

Không xóa vì nó sẽ mất hết image cần thiết đang sử dụng

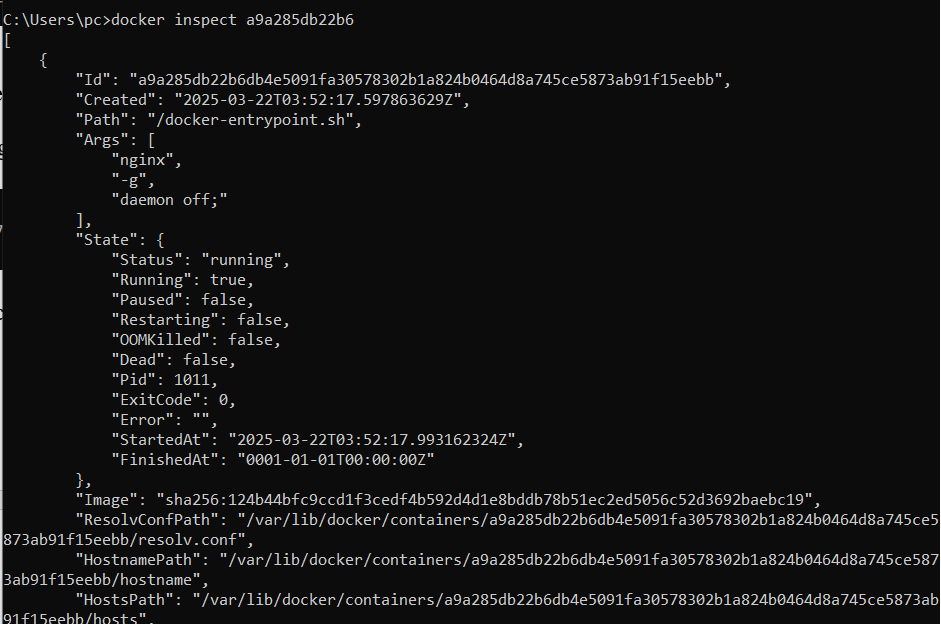
1. docker run -d -p 8080:80 nginx

Chạy container nginx, ánh xạ cổng 8080 của máy host với cổng 80 của container.



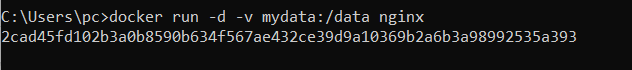
1. docker inspect <container\_id>

Hiển thị thông tin chi tiết về một container.



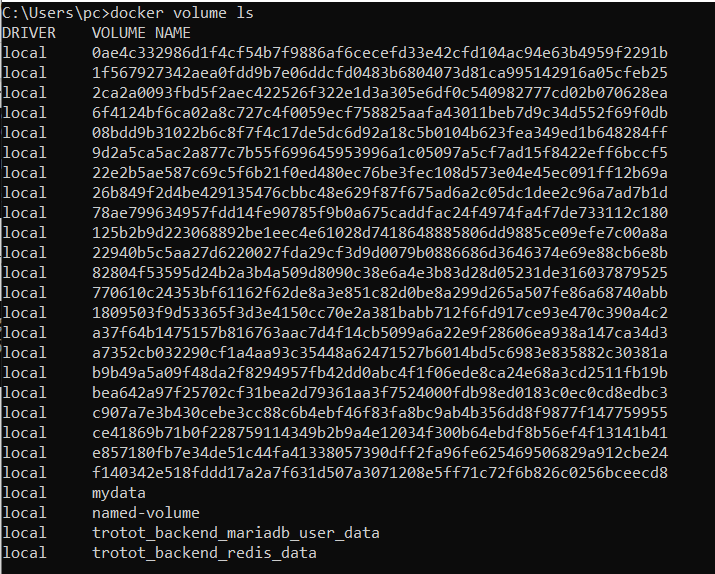
1. docker run -d -v mydata:/data nginx

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



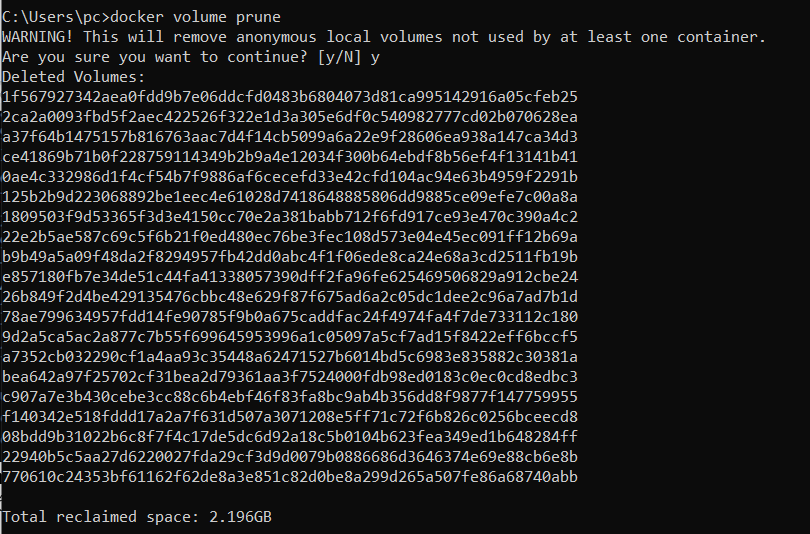
1. docker volume ls

Liệt kê danh sách các volume hiện có.



1. docker volume prune

Xóa tất cả volume không sử dụng.



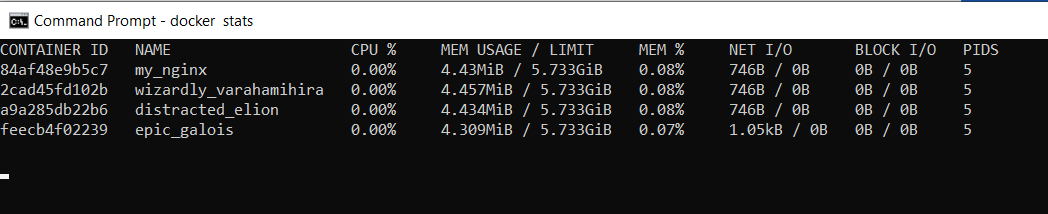
1. docker run -d --name my\_nginx nginx

Chạy container nginx với tên cụ thể my\_nginx.



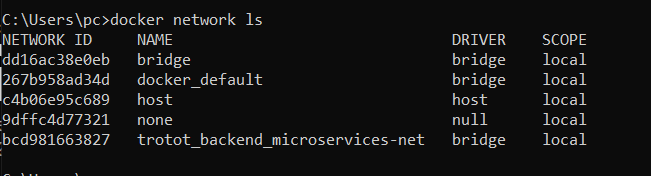
1. docker stats

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



1. docker network ls

Liệt kê các network trong Docker.



1. docker network create my\_network

Tạo một network có tên my\_network.



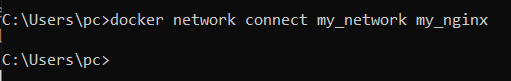
1. docker run -d --network my\_network --name my\_container nginx

Chạy container nginx và gán vào network my\_network.



1. docker network connect my\_network my\_nginx

Kết nối container my\_nginx vào network my\_network.



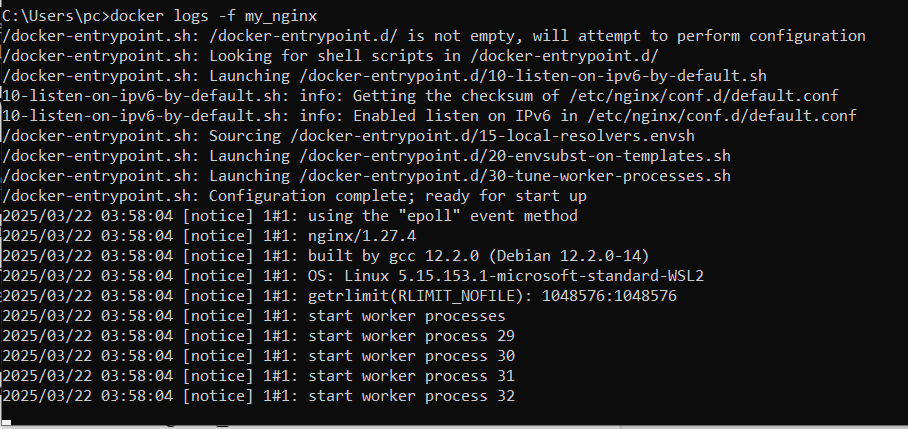
1. docker run -d -e MY\_ENV=hello\_world nginx

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



1. docker logs -f my\_nginx

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

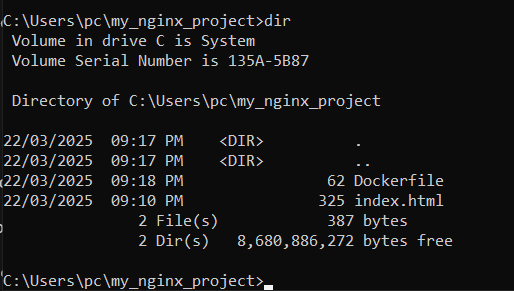


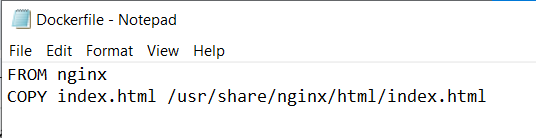
1. Dockerfile (Build Image)

FROM nginx

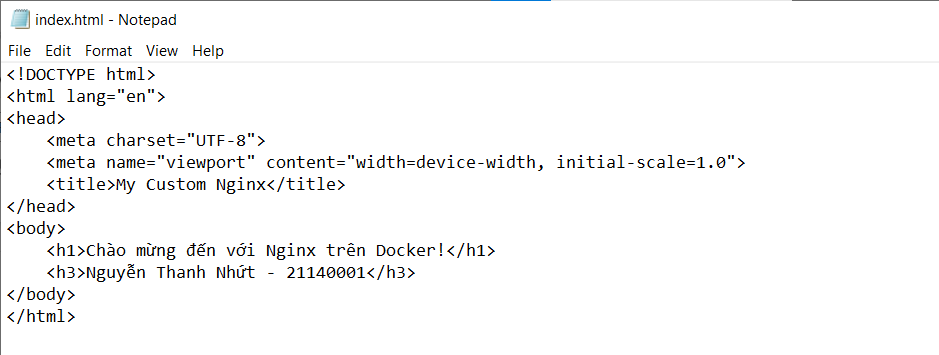
COPY index.html /usr/share/nginx/html/index.html

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



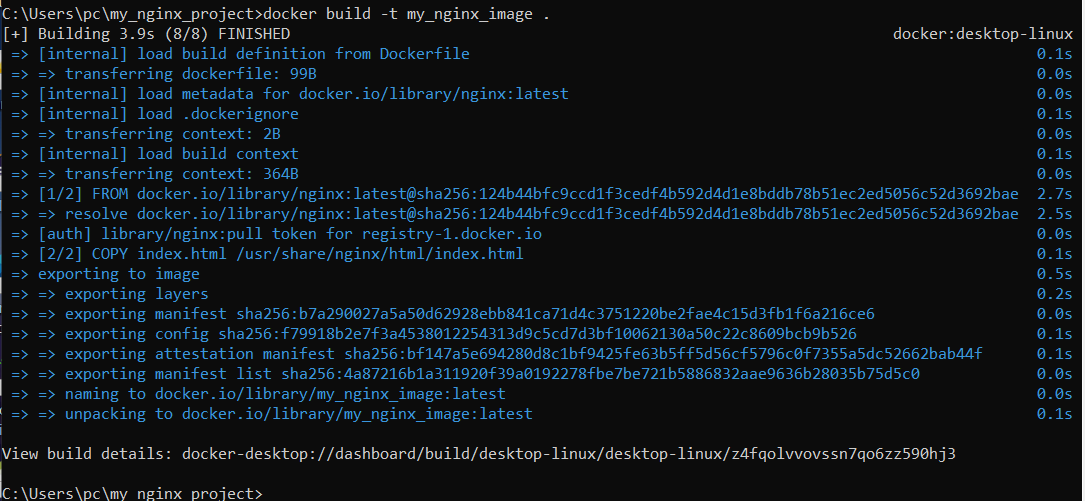
Docker file: 

File index.html:



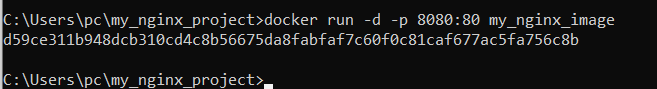
1. docker build -t my\_nginx\_image .

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

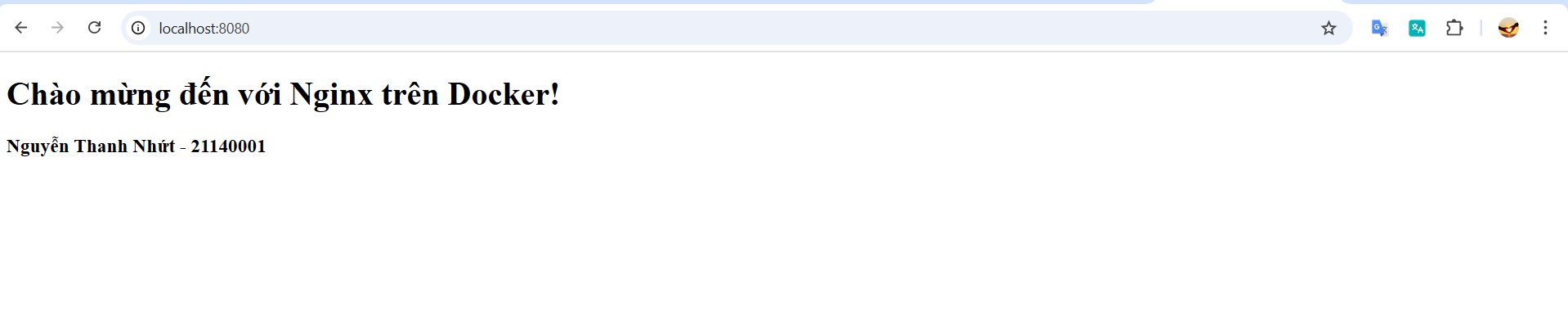


1. docker run -d -p 8080:80 my\_nginx\_image

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



Kết quả trên trình duyệt:

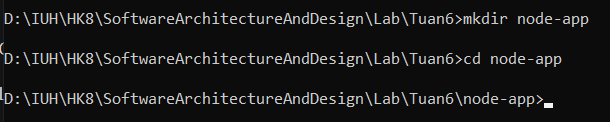


# Phần 2: Thao tác với Dockerfile

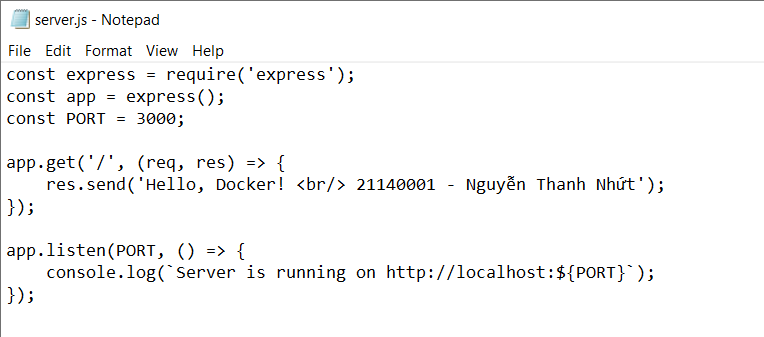
## Bài 1: Tạo Dockerfile chạy một ứng dụng Node.js đơn giản

Yêu cầu: Viết Dockerfile để chạy một ứng dụng Node.js hiển thị "Hello, Docker!" trên cổng 3000. Sử dụng node:18 làm base image.

Tạo thư mục dự án

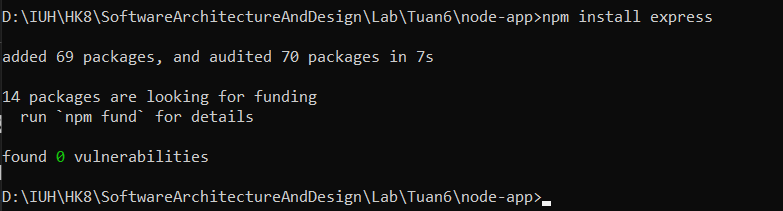


Tạo file server.js

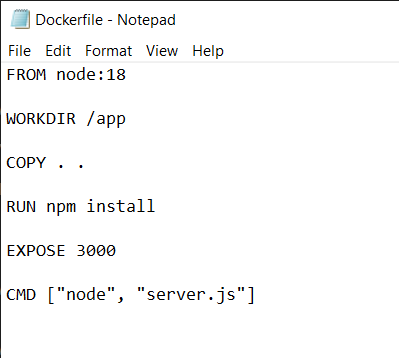


Tạo file package.json và cài express

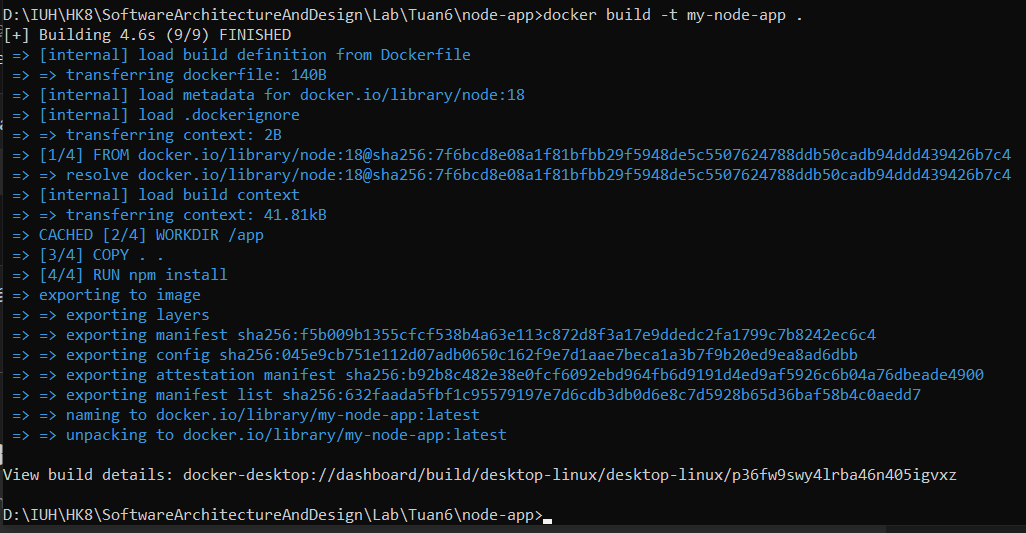


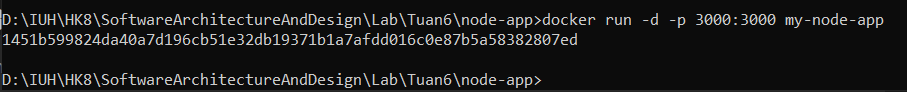


Viết Dockerfile

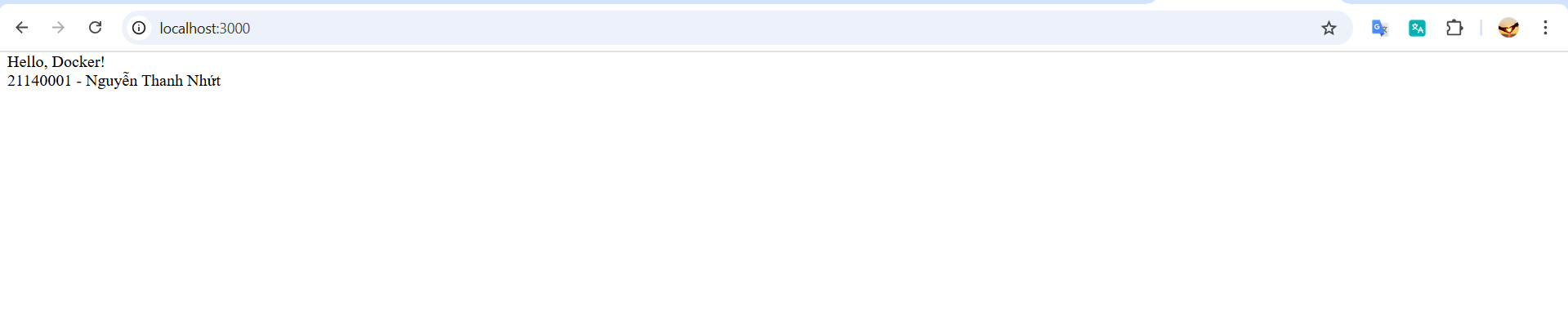


Build & Run Docker Container





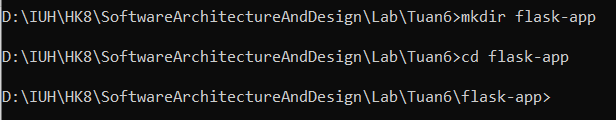
Kết quả trên trình duyệt:



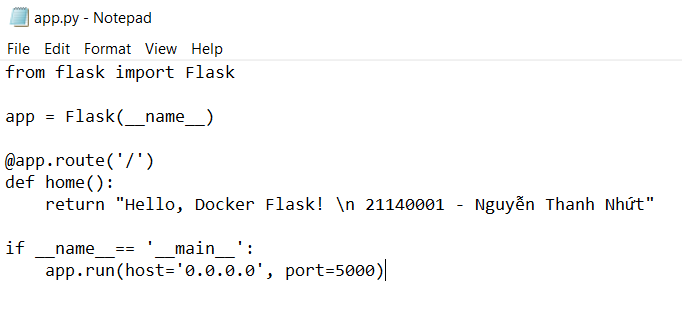
## Bài 2: Tạo Dockerfile chạy một ứng dụng Python Flask

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.

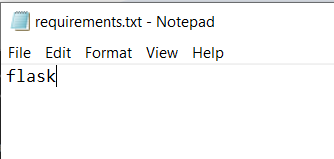
Tạo thư mục dự án



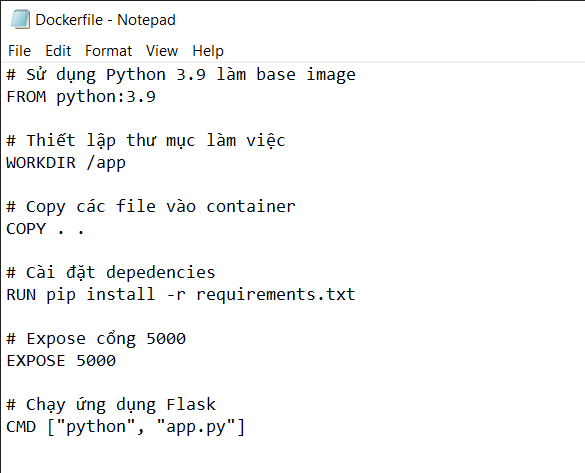
Tạo file app.py



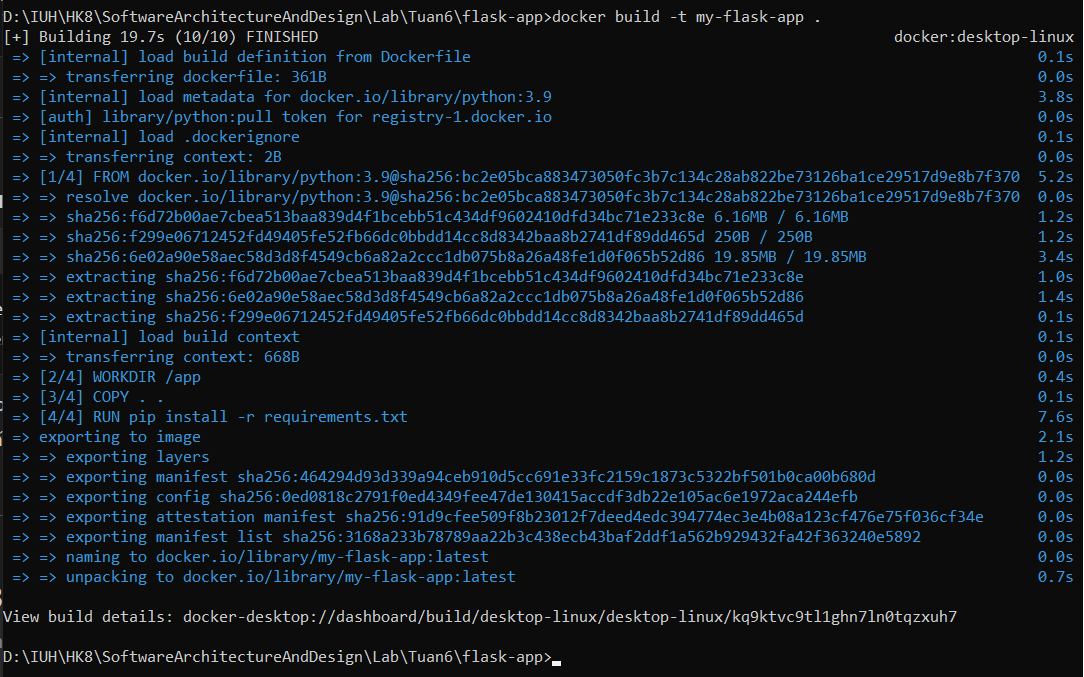
Tạo file requirements.txt

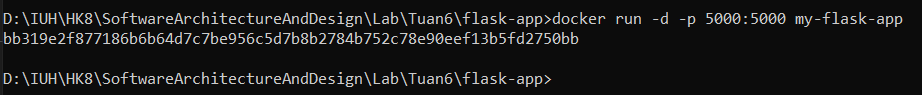


Viết Dockerfile

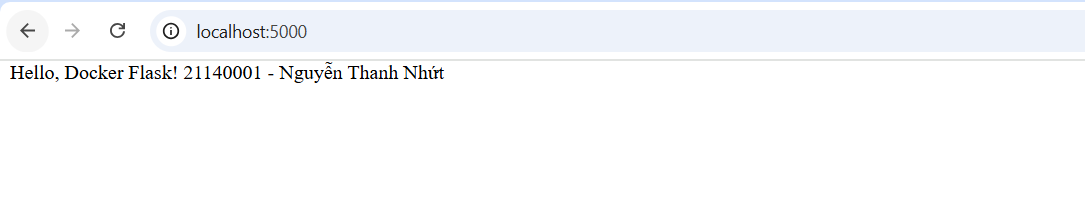


Build & Run Docker Container





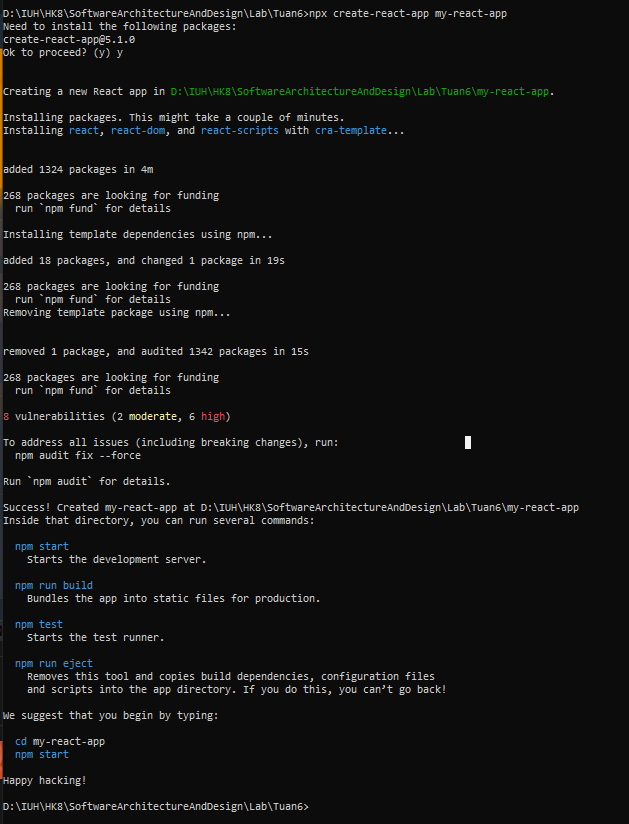
Kết quả trên trình duyệt:



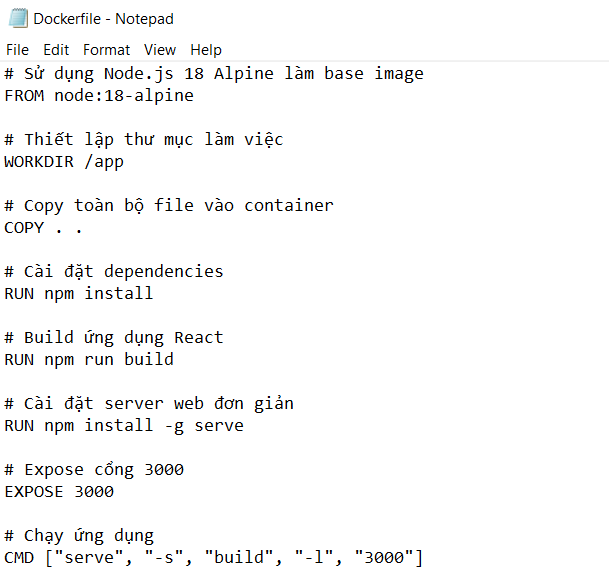
## Bài 3: Tạo Dockerfile chạy ứng dụng React

Yêu cầu: Dùng node:18-alpine để build và chạy một ứng dụng React.

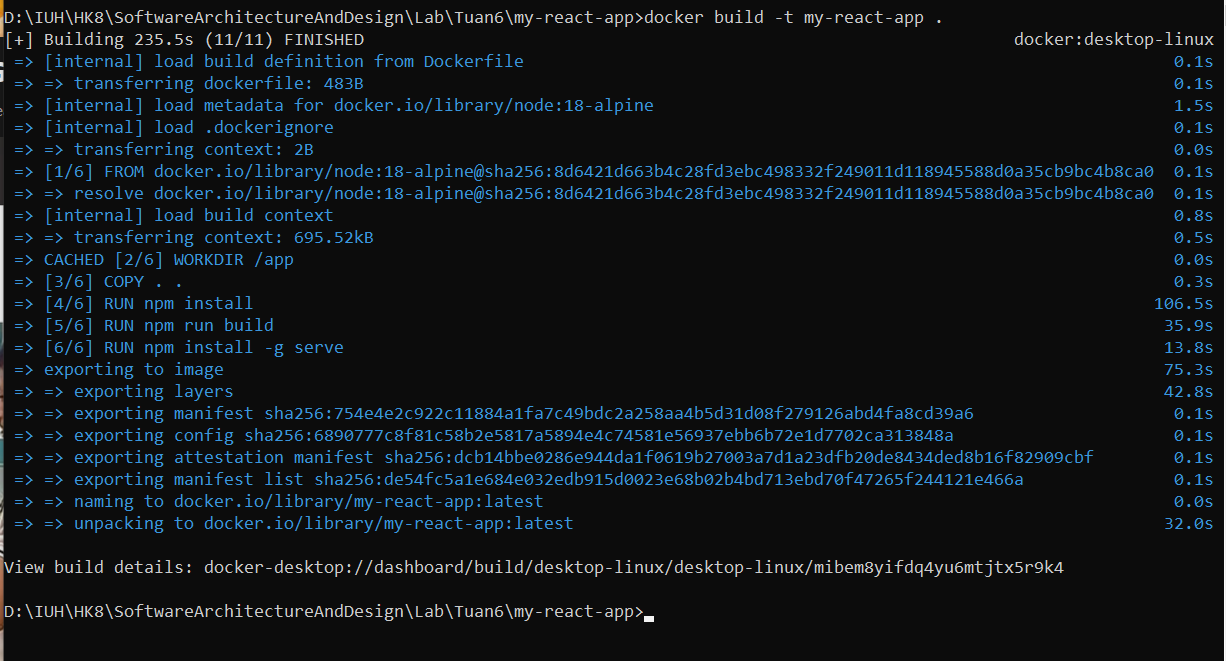
Tạo thư mục dự án

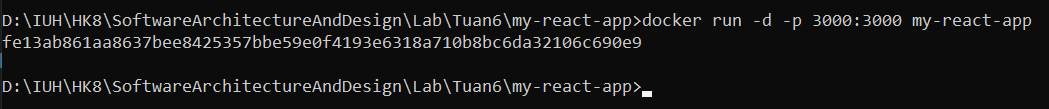


Viết Dockerfile

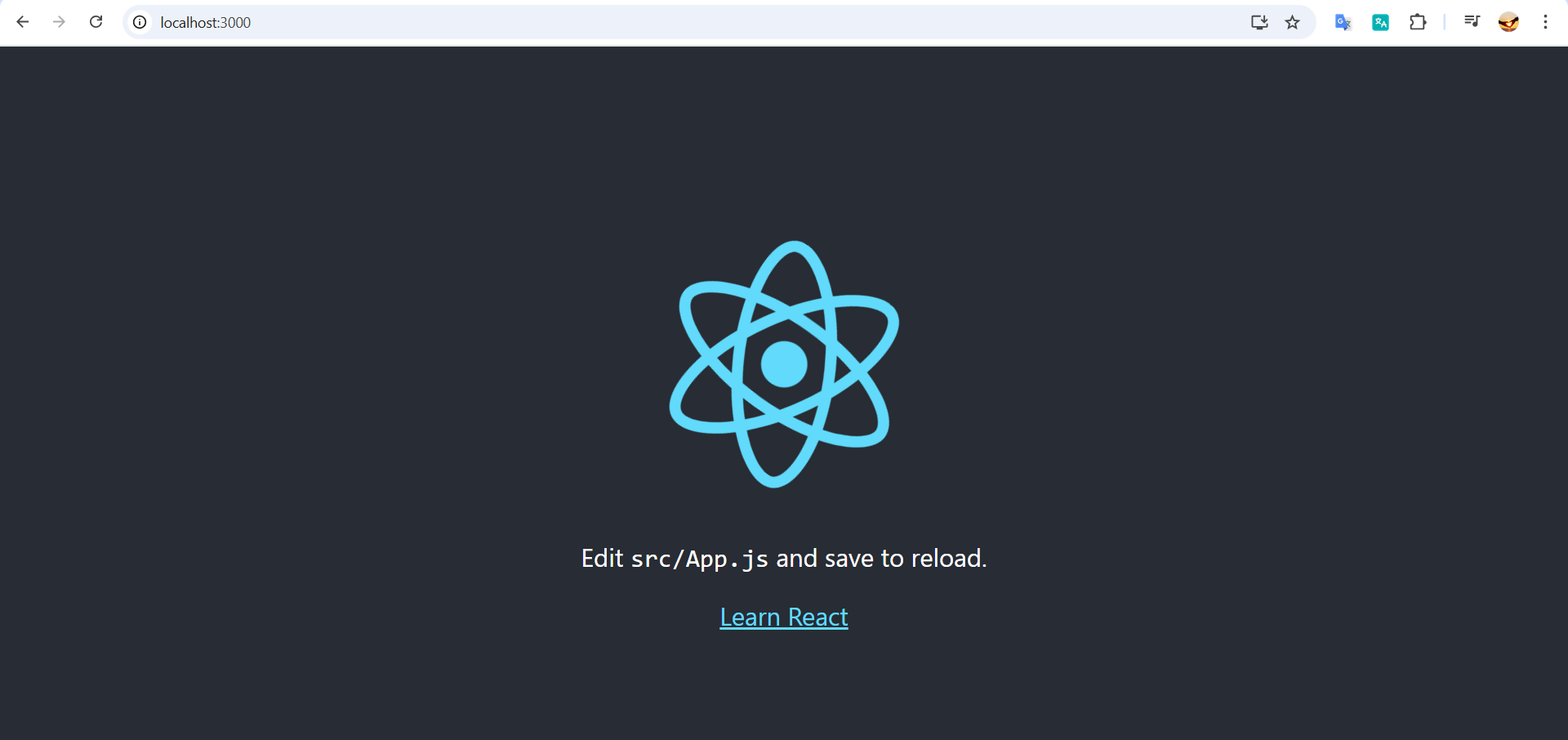


Build & Run Docker Container





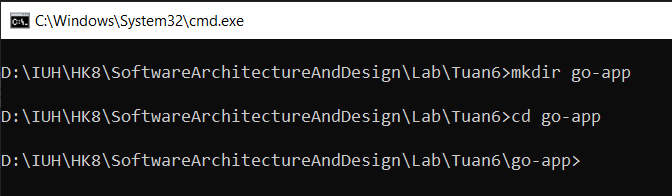
Kết quả trên trình duyệt:



## Bài 5: Tạo Dockerfile cho ứng dụng Go

Yêu cầu: Viết Dockerfile để build và chạy một ứng dụng Go đơn giản.

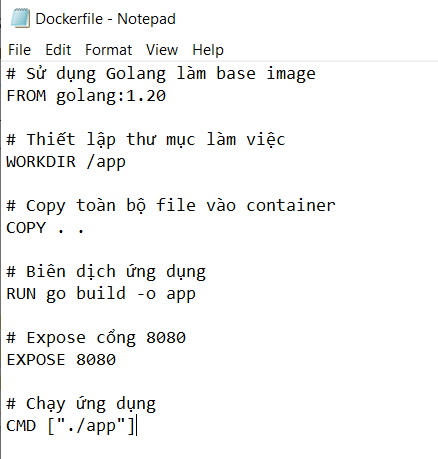
Tạo thư mục dự án



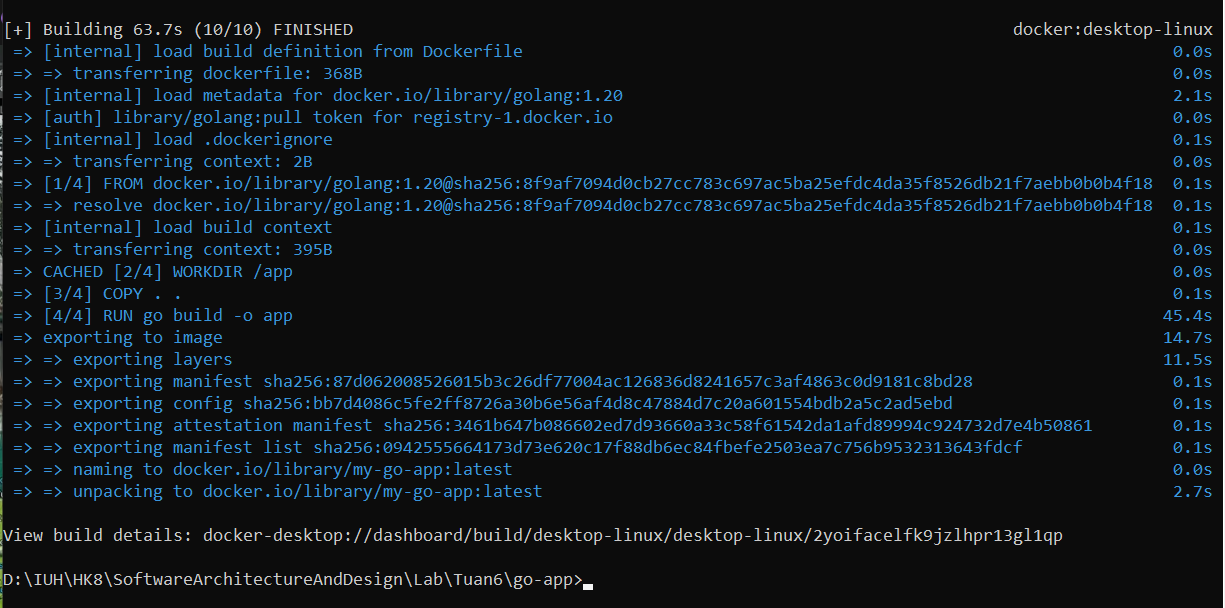
Tạo file main.go



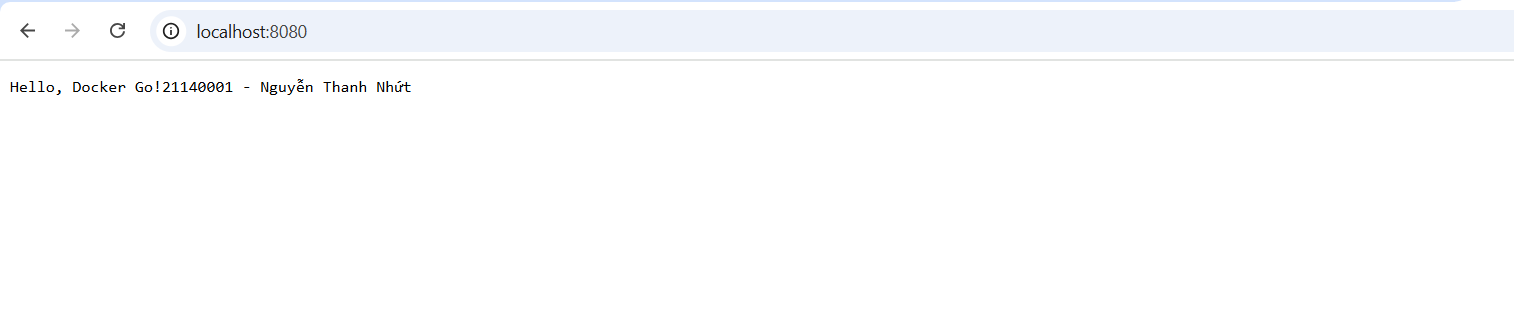
Viết Dockerfile:



Build & Run Docker Container



Kết quả trên trình duyệt:

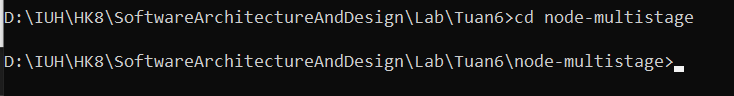


## Bài 6: Sử dụng Multi-stage Build trong Dockerfile (Node.js)

Yêu cầu:

* Dùng multi-stage build để tối ưu Dockerfile.
* Stage 1: Dùng node:18 để build code.
* Stage 2: Dùng node:18-alpine để chạy ứng dụng đã build.

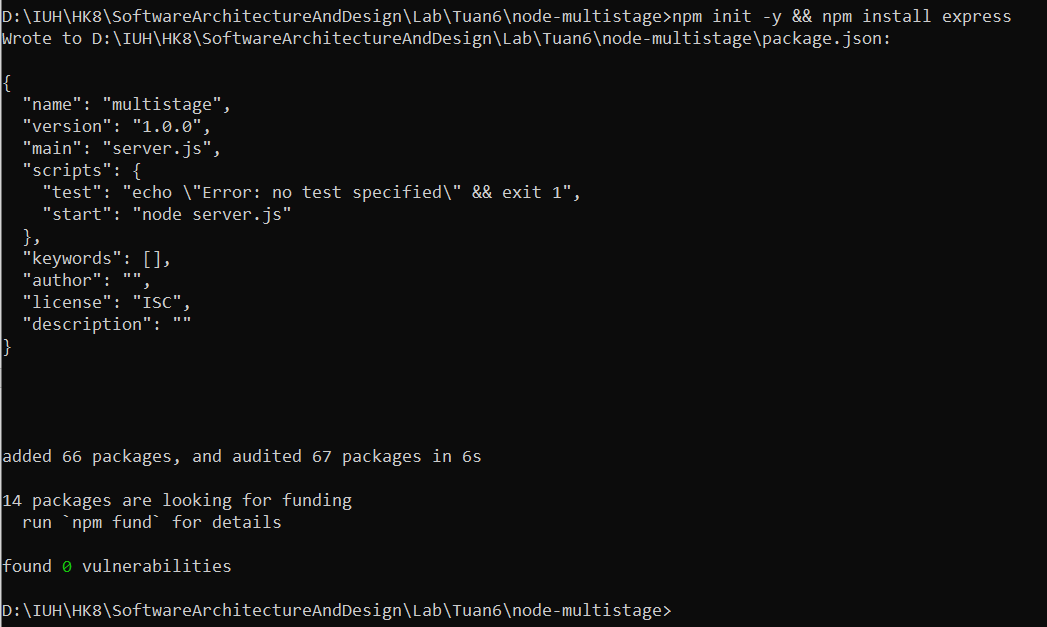
Tạo thư mục dự án



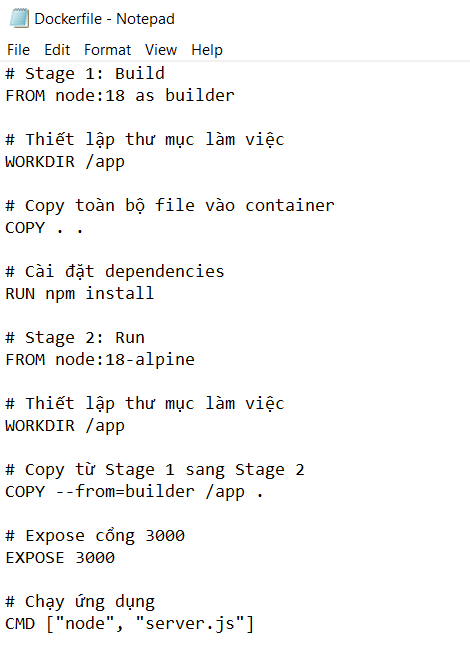
Tạo file server.js



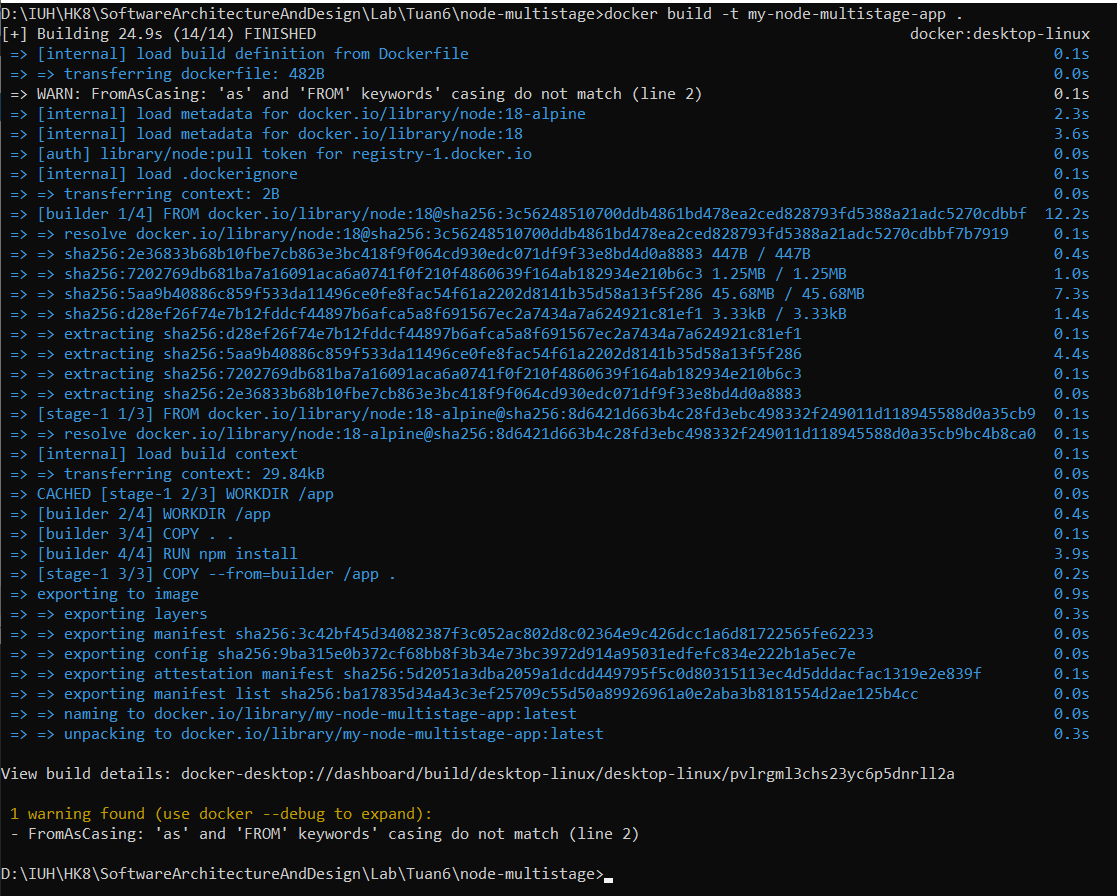
Tạo file package.json và cài express

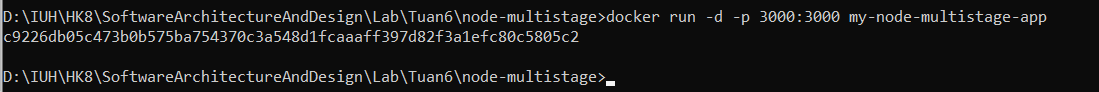


Viết Dockerfile với Multi-stage Build



Build & Run Docker Container





Kết quả trên trình duyệt:

