**21004295-Phạm Hữu Thuận-DHKTPM17A**

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

1. docker --version -> kiểm tra version hiện tại của docker trong máy



1. Docker run hello-world -> khởi chạy docker trước khi run

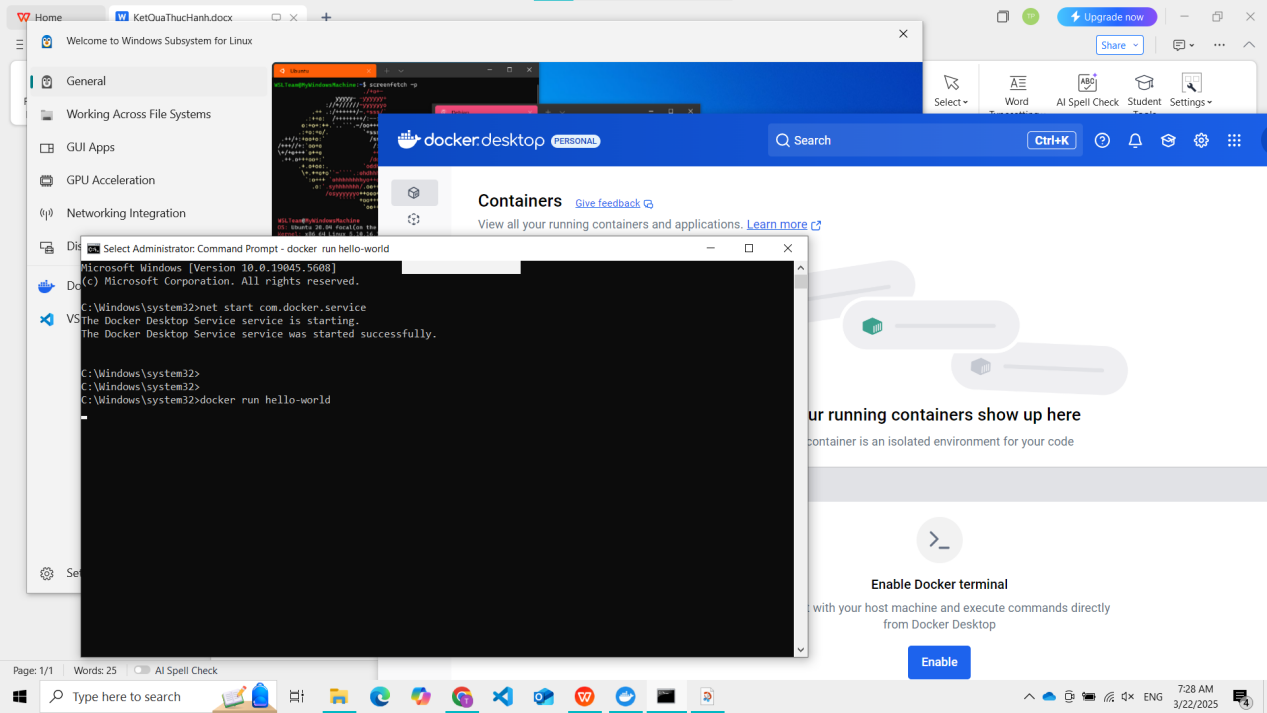


Image hello-world dc tải xuống và chạy



1. Docker pull nginx -> tìm kiếm và tải về image nginx từ Docker



1. Docker images -> liệt kê danh sách images hiện có gồm nginx và hello-world



1. docker run -d nginx : chạy container nginx và -d là chạy chế độ nền (kh chiếm terminal)

Kết quả là id của container



1. docker ps : show cotainer đang chạy



1. docker ps -a: hiển thị danh sách container bao gồm cả container đã stop



1. docker logs 1fd931550ef5: xem logs của container có id là 1fd931550ef5(nginx)



1. docker exec -it 1fd931550ef5 /bin/sh: truy cập và thao tác trong container tại /bin/sh



1. docker stop 1fd931550ef5: stop 1 container đang running theo id



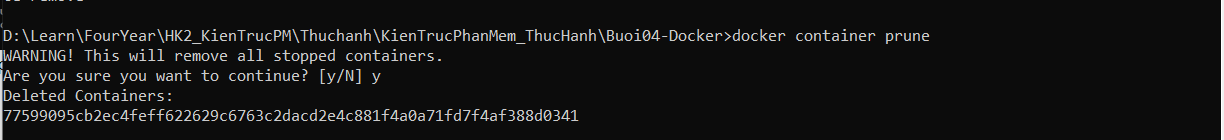
1. docker restart 1fd931550ef5: restart lại container theo id



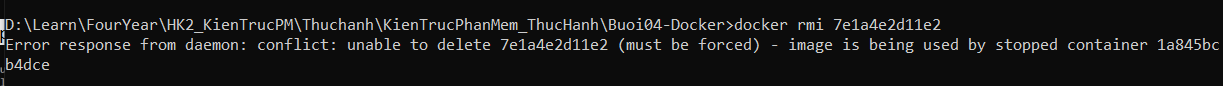
1. docker rm 1fd931550ef5: remove 1 container ra khỏi docker. Nhưng container này đang chạy nên Error -> phải stop container mới remove được



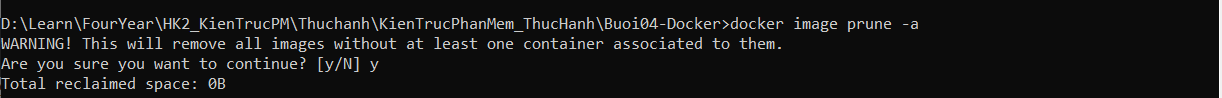
1. docker container prune: remove all container stopped



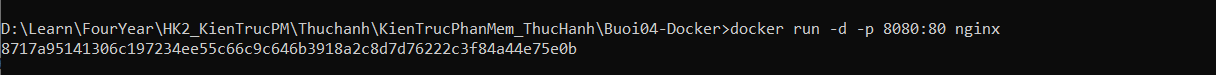
1. docker rmi 7e1a4e2d11e2: lệnh xóa img theo id -> không thể xóa do img đang được sử dụng bởi 1 container.



1. docker image prune -a: xóa image không sử dung -a(all)



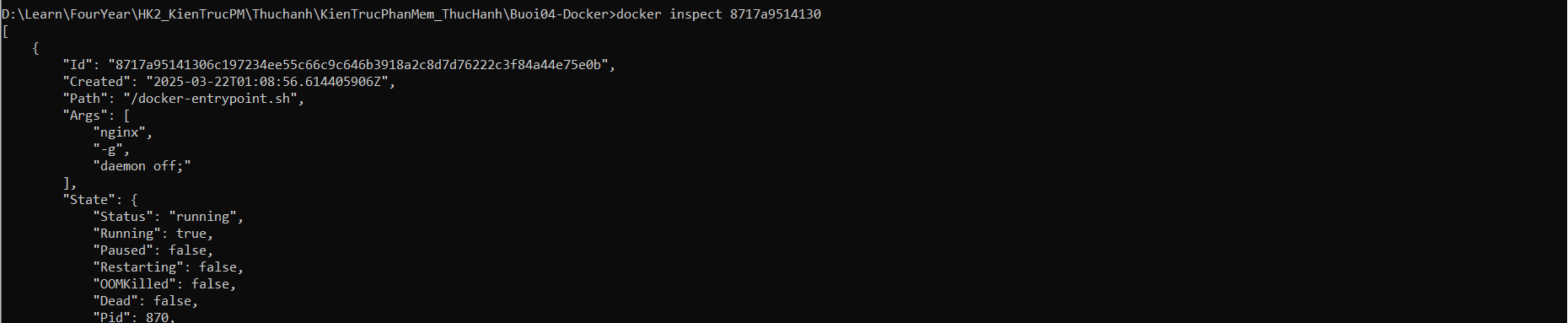
1. docker run -d -p 8080:80 nginx: chạy container và -d (chế độ nền) và -p (port 8080)



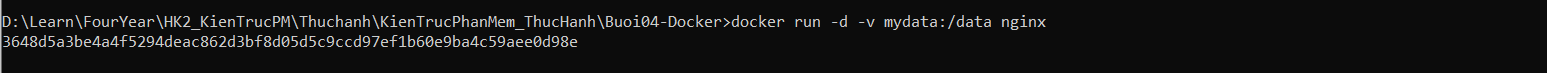
Ket qua



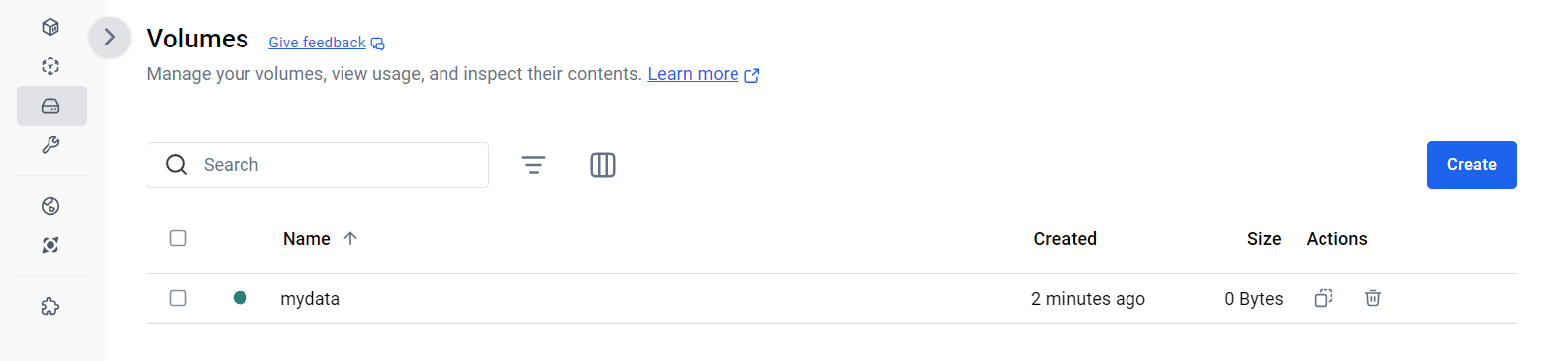
1. docker inspect 8717a9514130:xem trạng thái, cấu hình của container



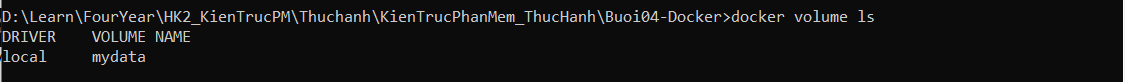
1. docker run -d -v mydata:/data nginx: lưu trữ dữ liệu bên ngoài container. Khi container bị xóa thì dữ liệu vẫn còn /data



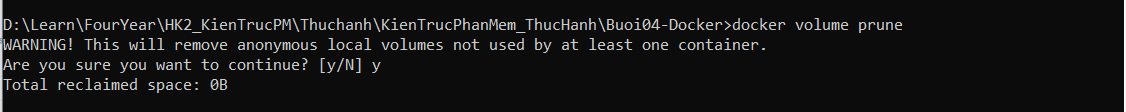
Kiểm tra volumes đã được tạo /mydata



1. docker volume ls: kiểm tra danh sách volume hiện có.



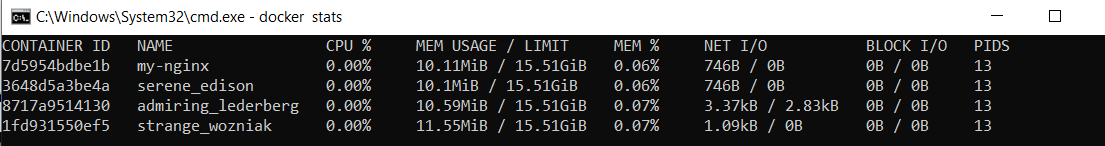
1. docker volume prune: Xóa các volume đang không được sử dụng



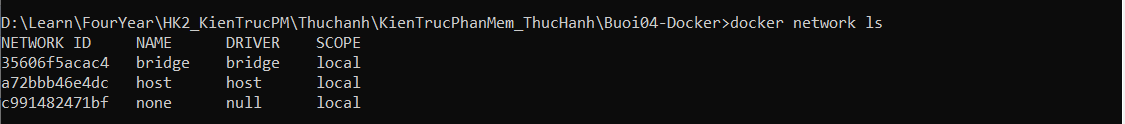
1. docker run -d --name my-nginx nginx: chạy 1 container mới có tên my-nginx bằng image nginx



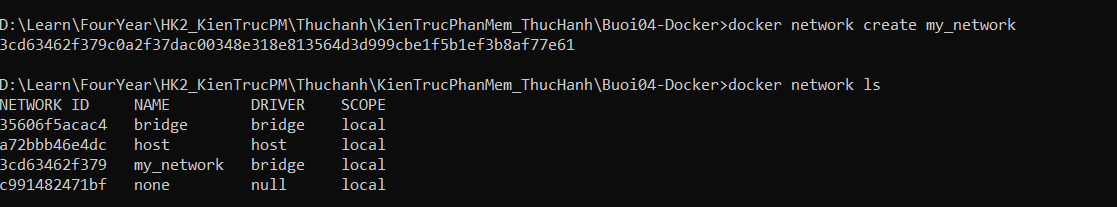
1. docker stats: kiểm tra hệ thống (CPU, I/O) theo thời gian thực



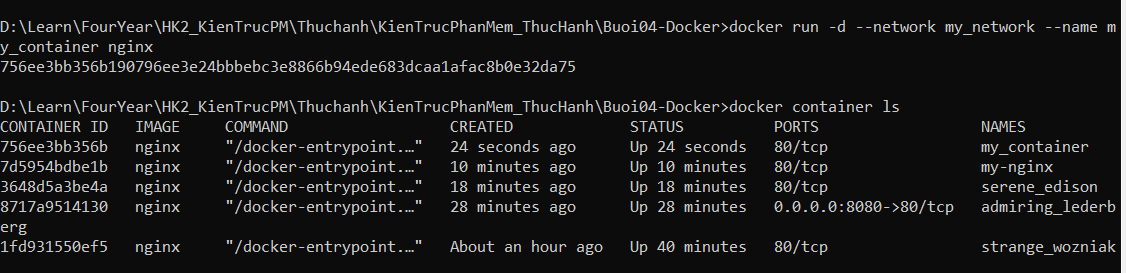
1. docker network ls: kiểm tra mạng trong docker



1. docker network create my\_network: tạo mới 1 network



1. docker run -d --network my\_network --name my\_container nginx: tạo mới 1 container và setup network của nó là my\_network



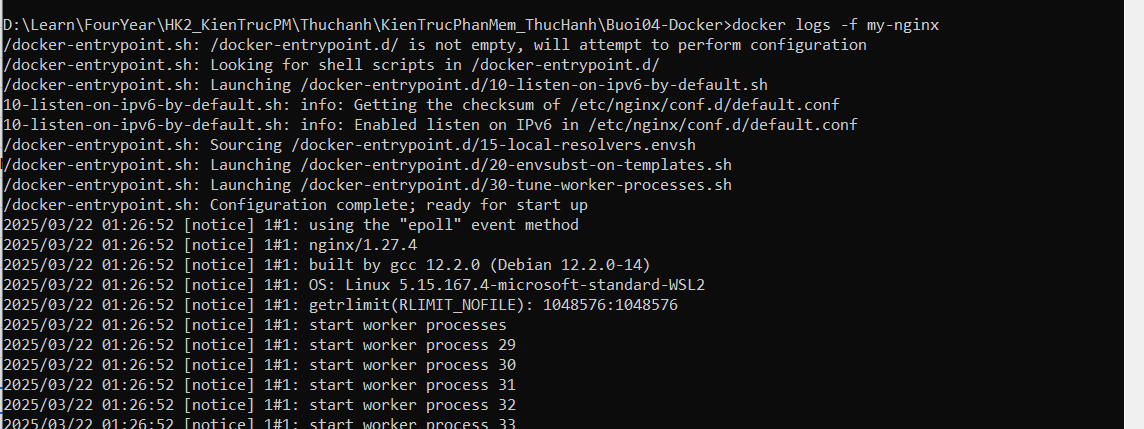
1. docker network connect my\_network my-nginx:

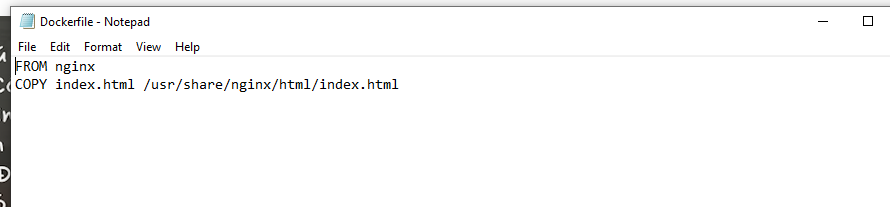
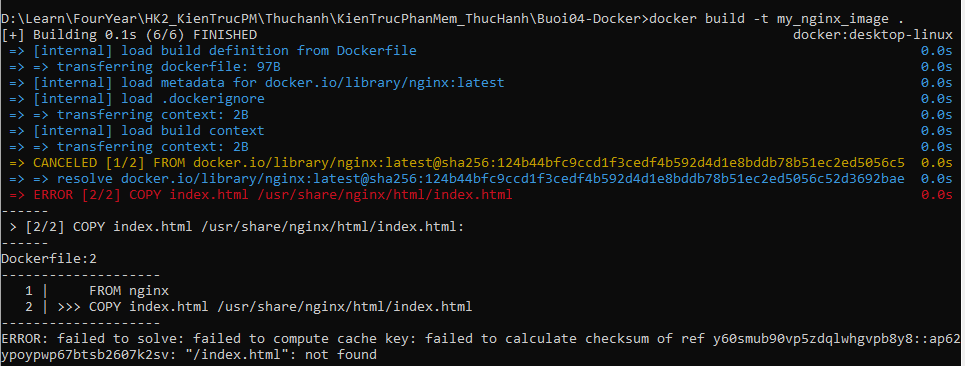
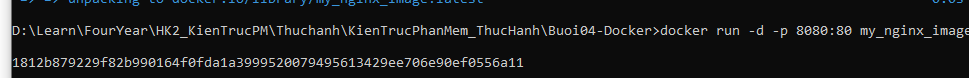
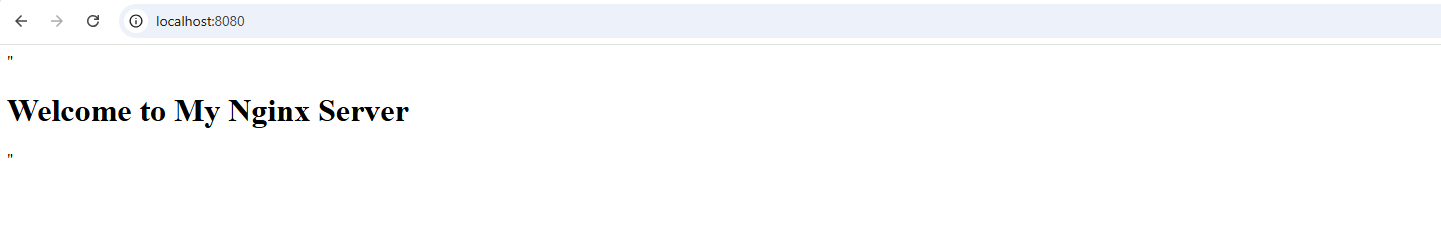


1. docker run -d -e MY\_ENV=hello\_world nginx: Thiết lập biến môi trường MY\_ENV với giá trị "hello\_world".

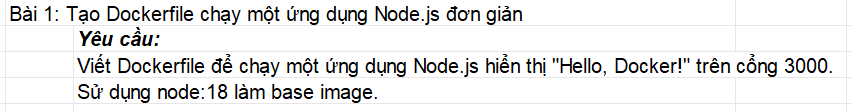


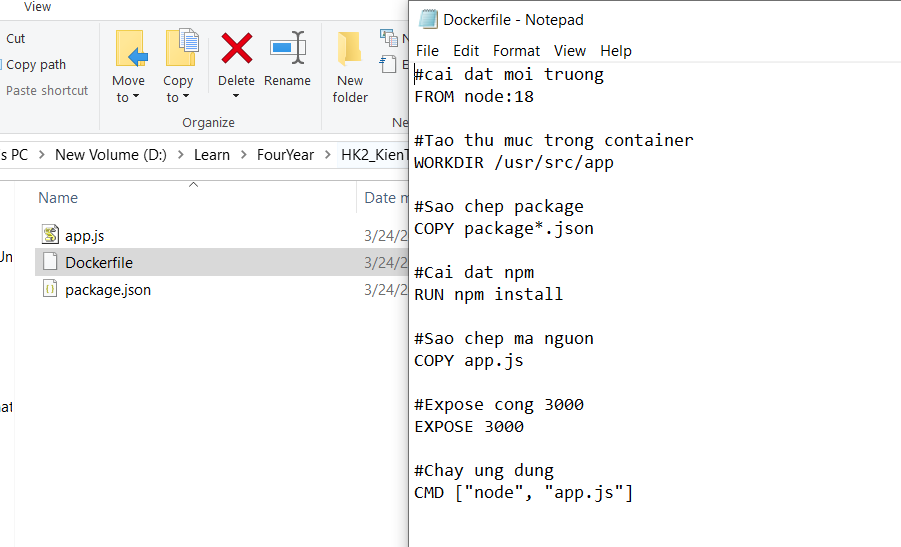
1. docker logs -f my-nginx: theo dõi log. Có thêm -f : theo dõi liên tục, hiển thị ngay khi có log mới



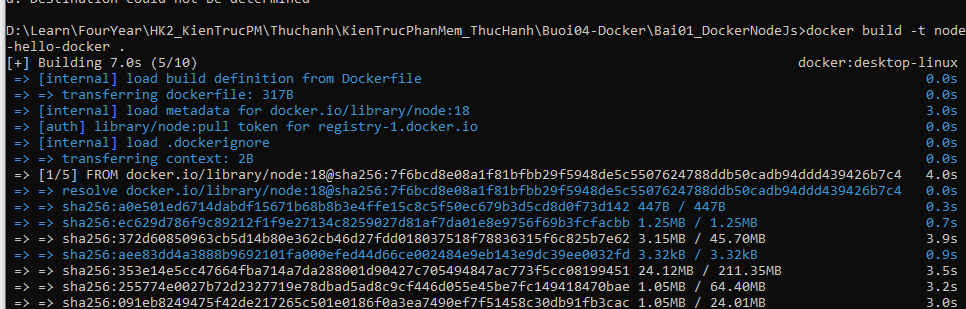
1. Viết Dockerfile  
   
2. docker build -t my\_nginx\_image .: buil a image from dockerfile  
   
3. docker run -d -p 8080:80 my\_nginx\_image  
     
   

**THAO TAC VOI DOCKERFILE**

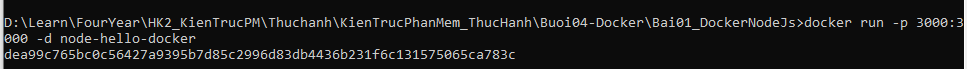




Build image

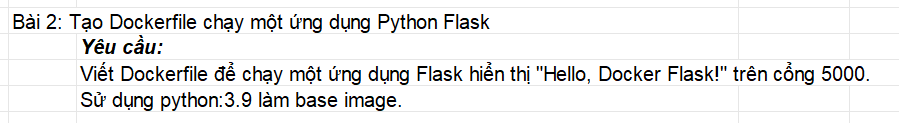


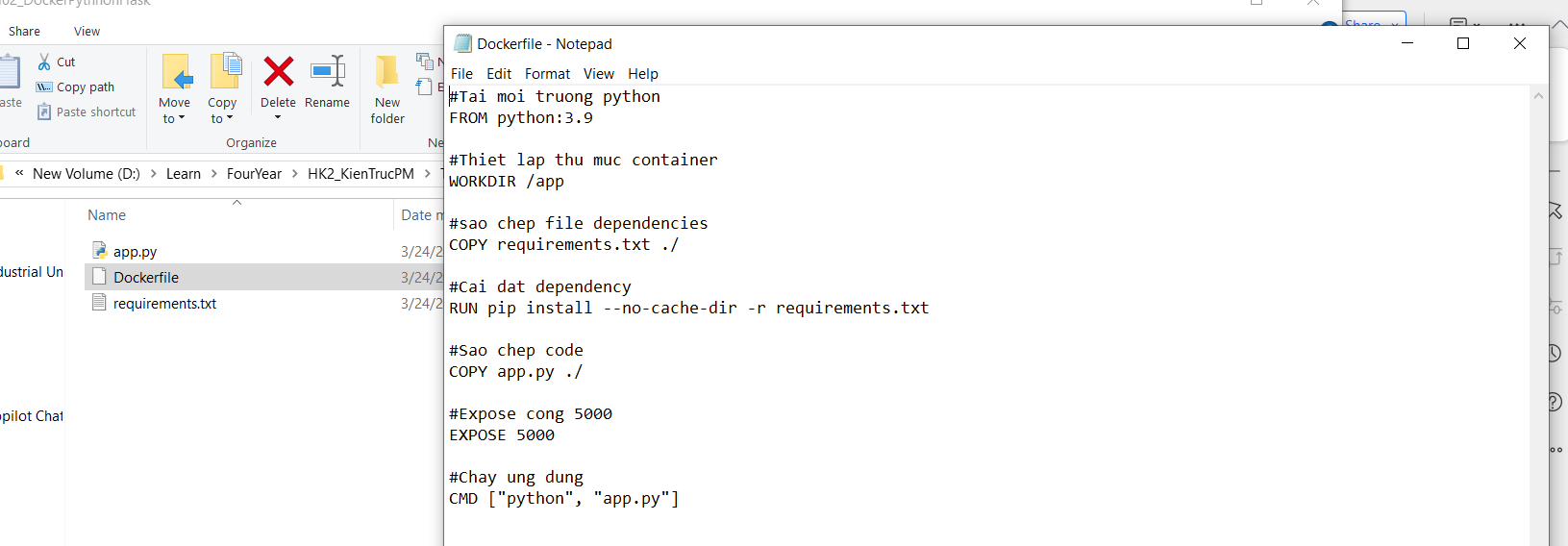
RUN



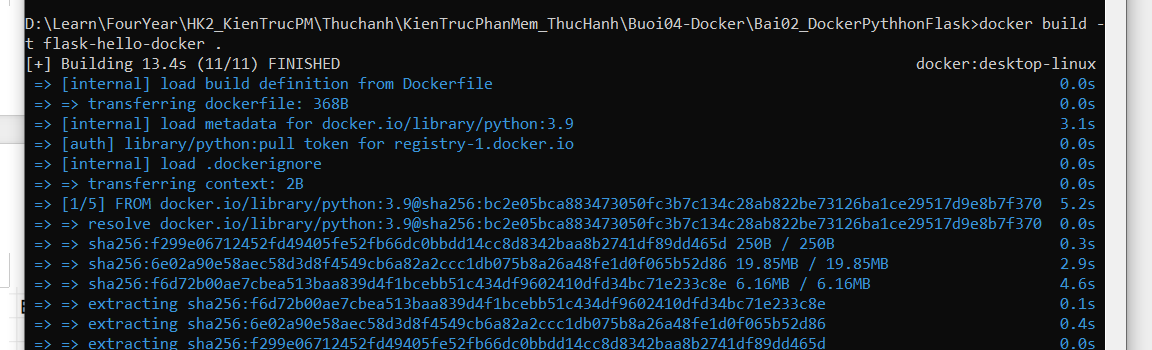
Ketqua



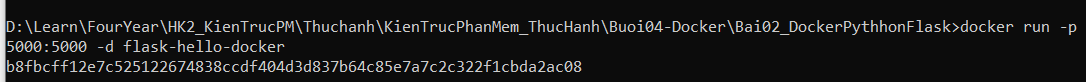




Build: docker build -t flask-hello-docker .

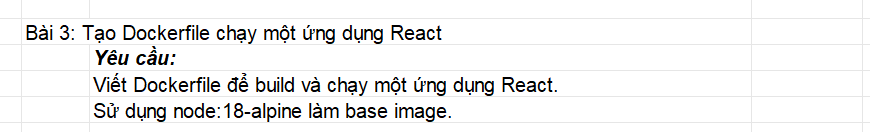


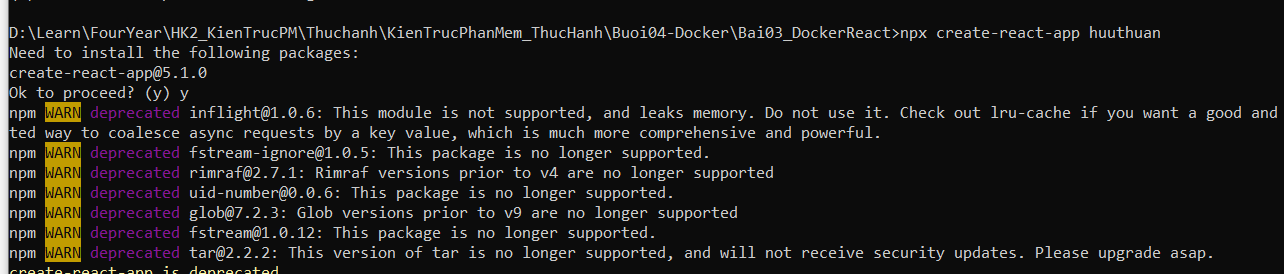
RUN: docker run -p 5000:5000 -d flask-hello-docker

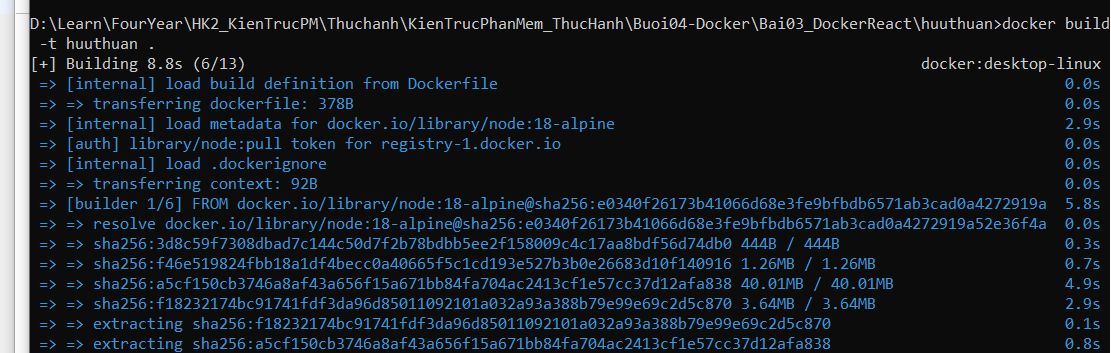


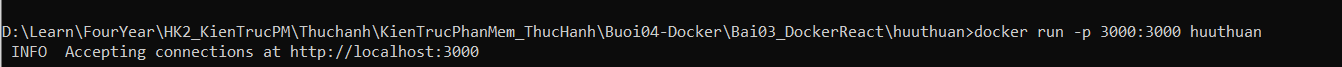
Ket qua: curl http://localhost:5000



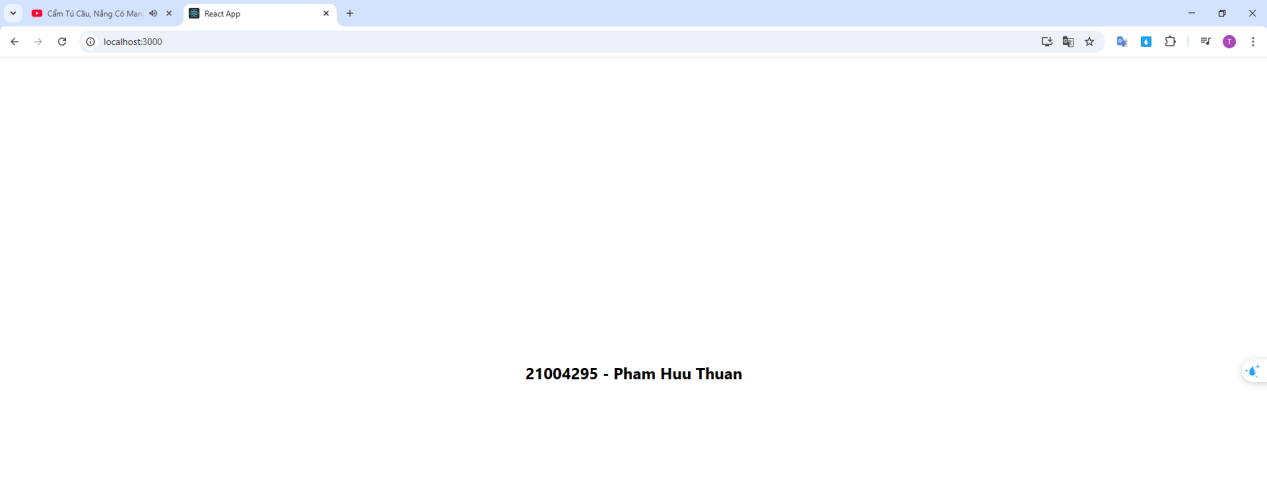


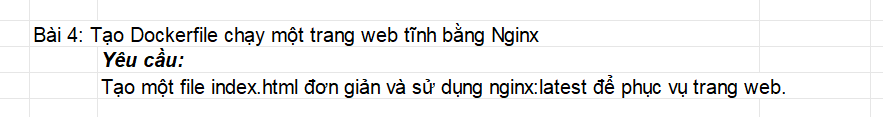
Tao Project  


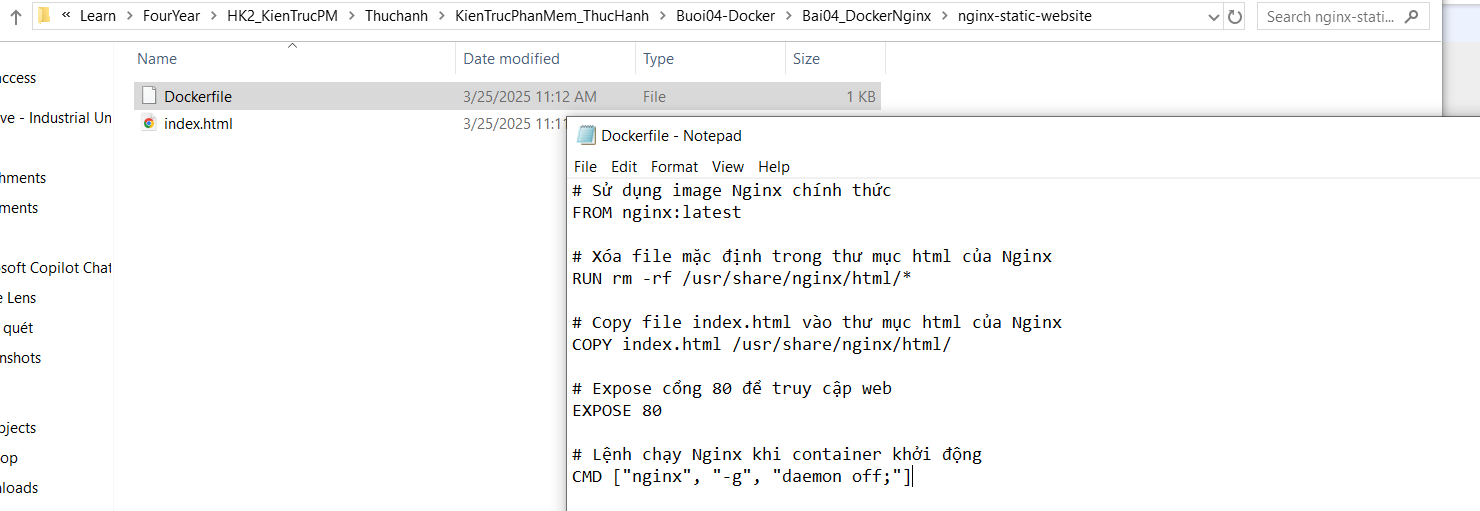




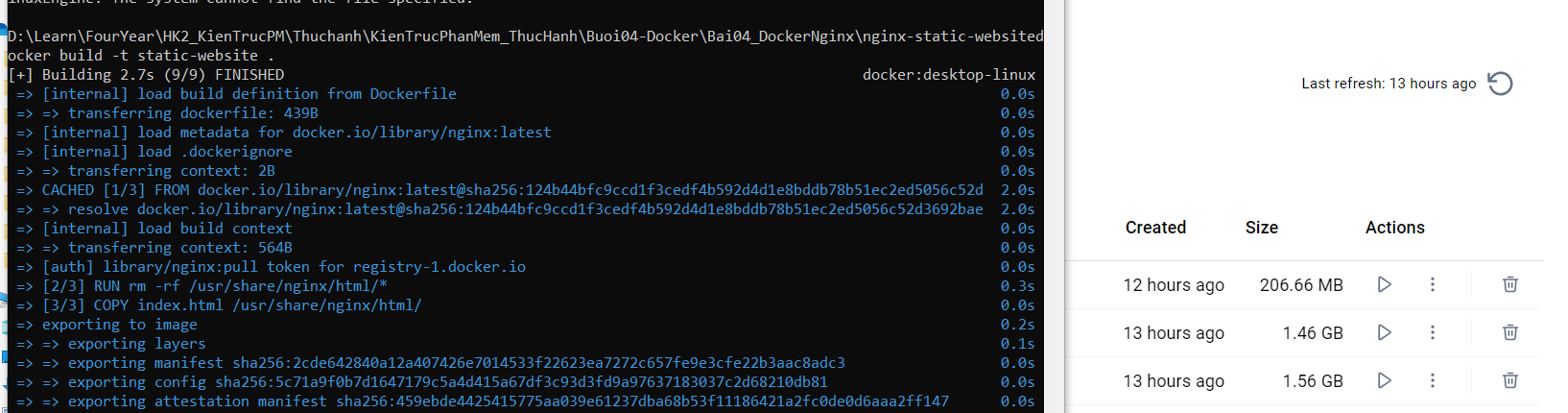
Ket qua



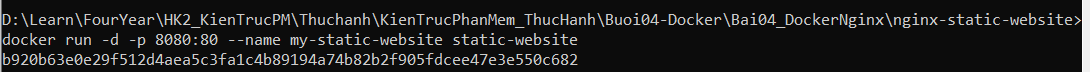


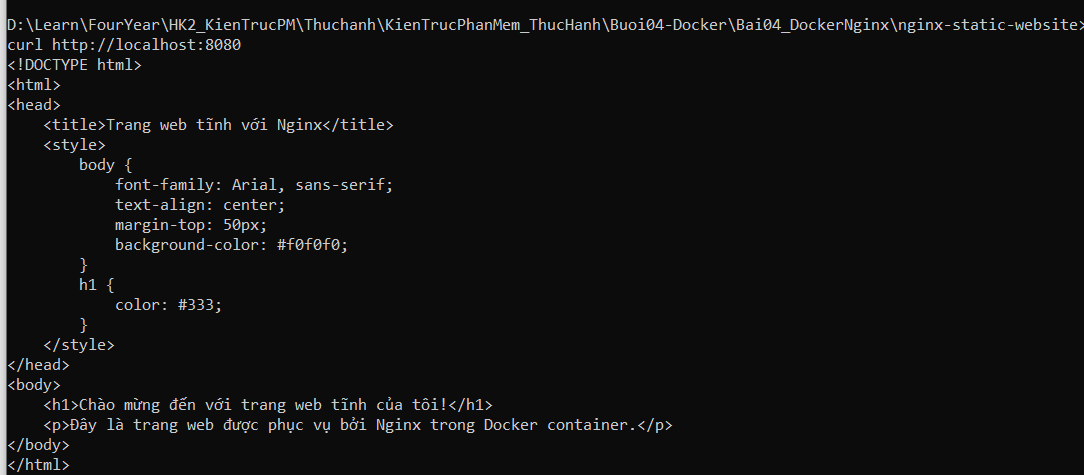


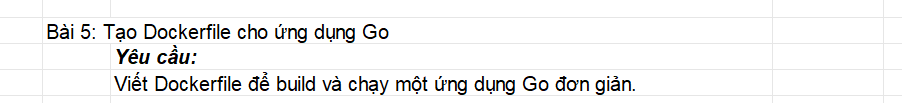
Build

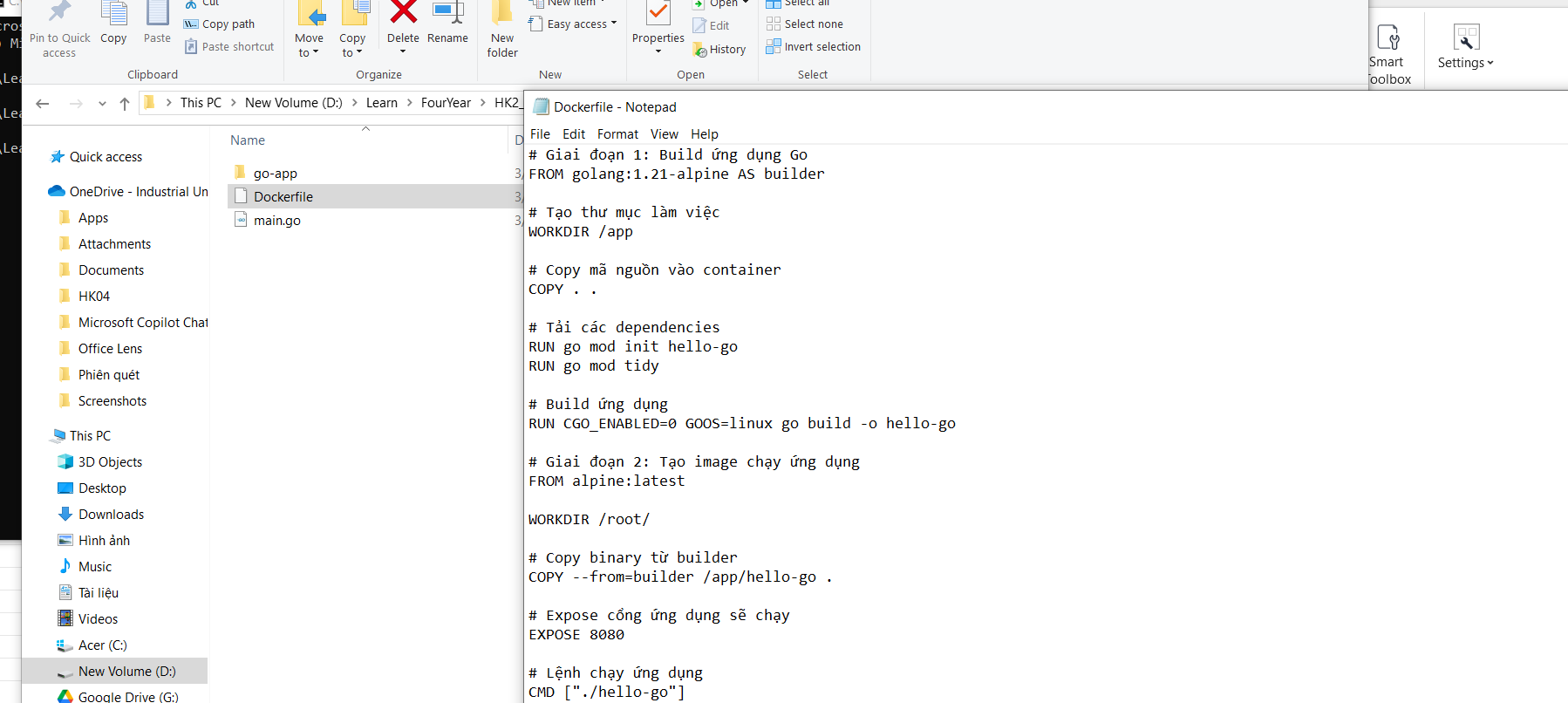


Run

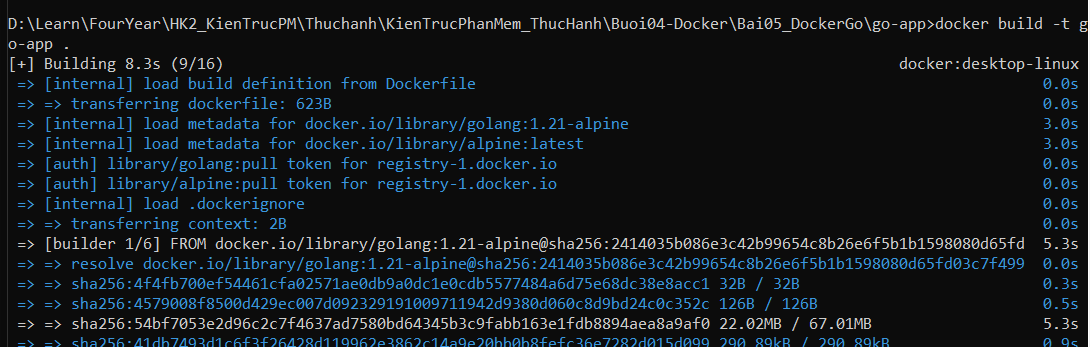




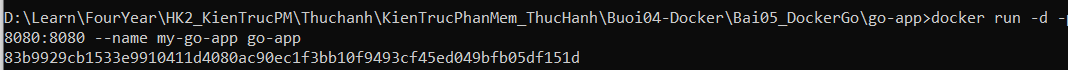




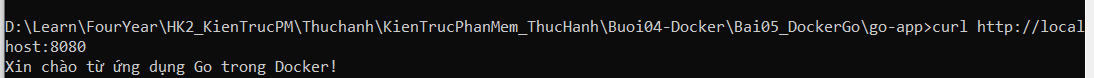
Build

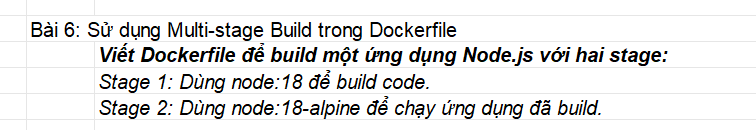


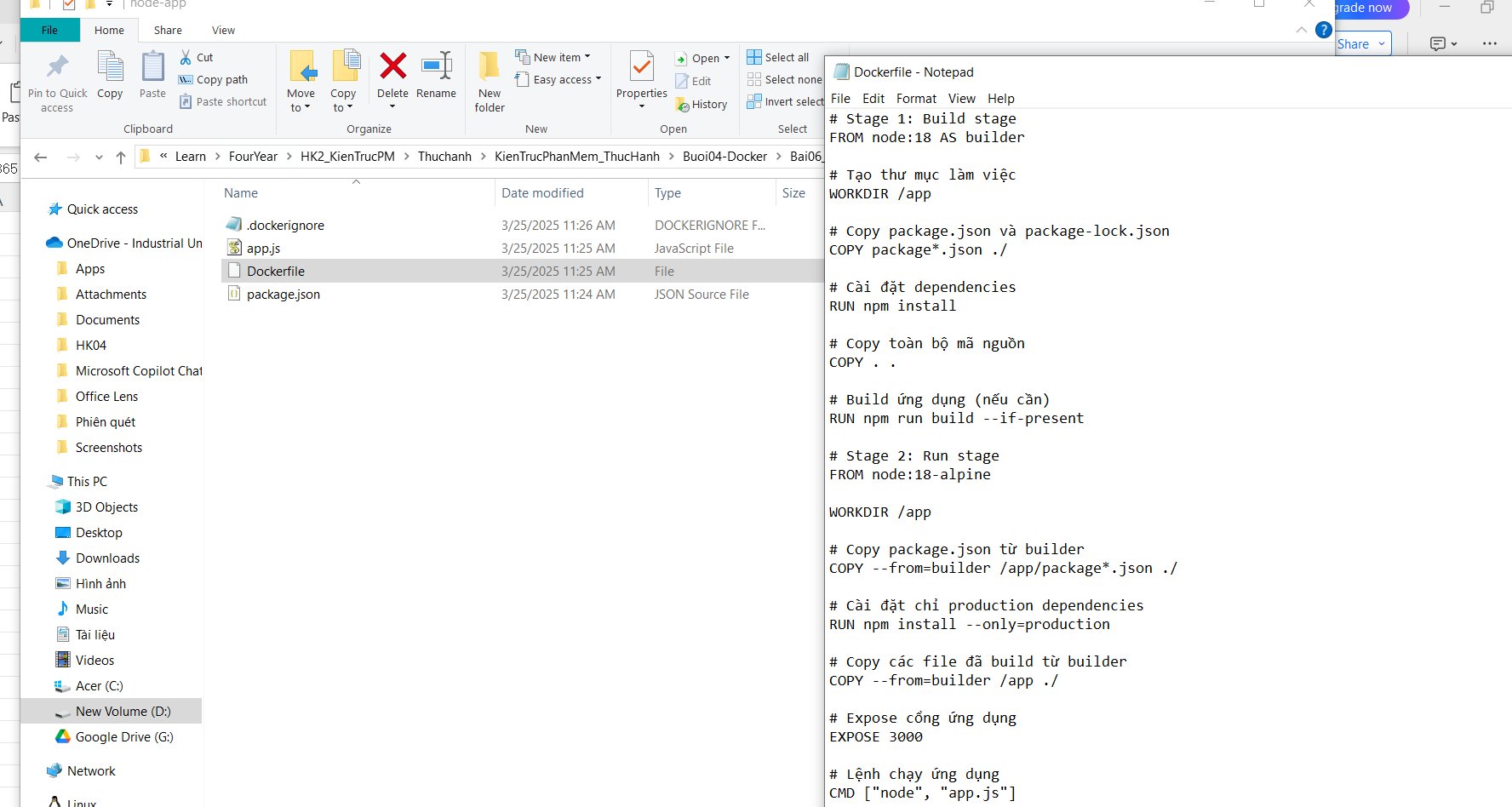
Run



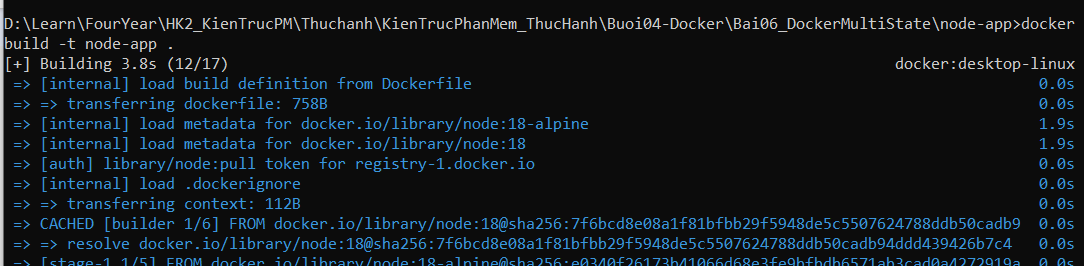
Ket qua







Build



Run

