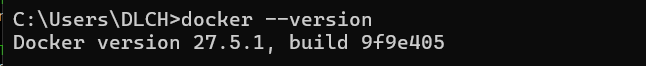
# Phần 1:

1. docker –version



1. docker run hello-world

A screenshot of a computer

AI-generated content may be incorrect.

1. docker pull nginx

A computer screen with white text

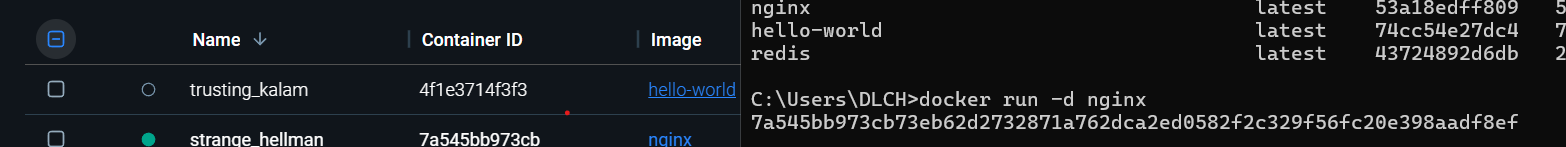
AI-generated content may be incorrect.

1. docker images

A screenshot of a computer screen

AI-generated content may be incorrect.

1. docker run -d nginx



1. docker ps

A screenshot of a computer screen

AI-generated content may be incorrect.

1. docker ps -a

A black screen with white text

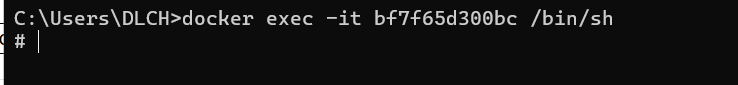
AI-generated content may be incorrect.

1. docker logs <container\_id>

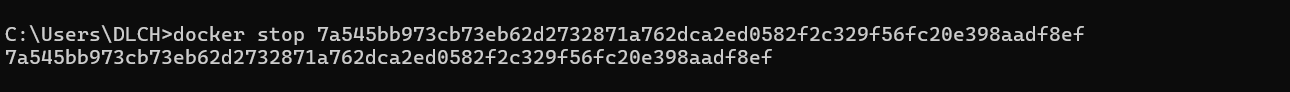
A black screen with white text

AI-generated content may be incorrect.

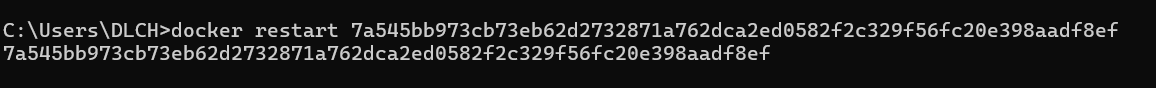
1. docker exec -it <container\_id> /bin/sh



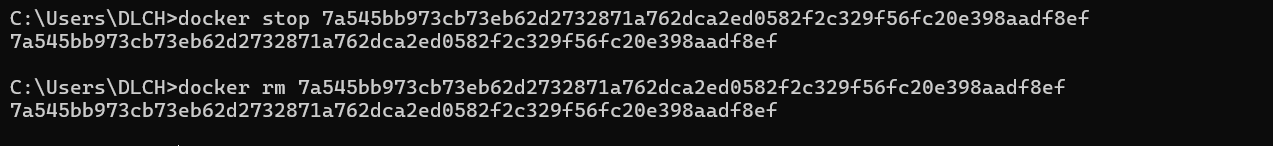
1. docker stop <container\_id>



1. docker restart <container\_id>



1. docker rm <container\_id>

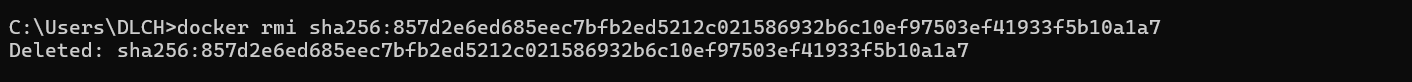


1. docker container prune

A black screen with white text

AI-generated content may be incorrect.

1. docker rmi <image\_id>



1. docker image prune -a

A screenshot of a computer screen

AI-generated content may be incorrect.

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

A screenshot of a computer

AI-generated content may be incorrect.

1. docker inspect <container\_id>

A black and white screen

AI-generated content may be incorrect.

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

A black screen with white text

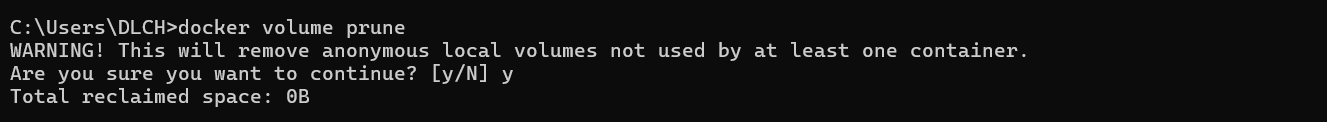
AI-generated content may be incorrect.

1. docker volume ls

A screen shot of a computer

AI-generated content may be incorrect.

1. docker volume prune



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



1. docker stats

A screenshot of a computer screen

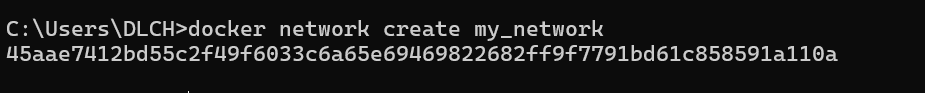
AI-generated content may be incorrect.

1. docker network ls

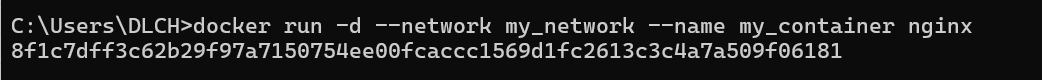
A screen shot of a computer

AI-generated content may be incorrect.

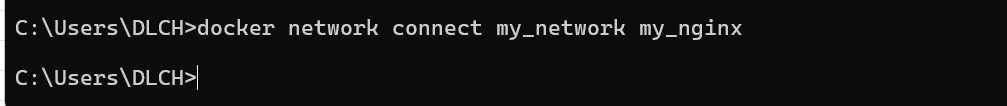
1. docker network create my\_network



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



1. docker network connect my\_network my\_nginx



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

A black background with white text

AI-generated content may be incorrect.

1. docker logs -f my\_nginx

A screenshot of a computer

AI-generated content may be incorrect.

1. FROM nginx

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

A screenshot of a computer

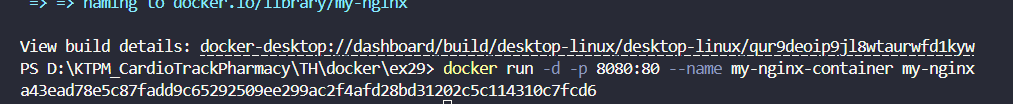
AI-generated content may be incorrect.

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

A screenshot of a computer program

AI-generated content may be incorrect.

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



A white screen with red text

AI-generated content may be incorrect.

# Phần 2 bài tập:

## 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.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A white rectangular object with a black border

AI-generated content may be incorrect.

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

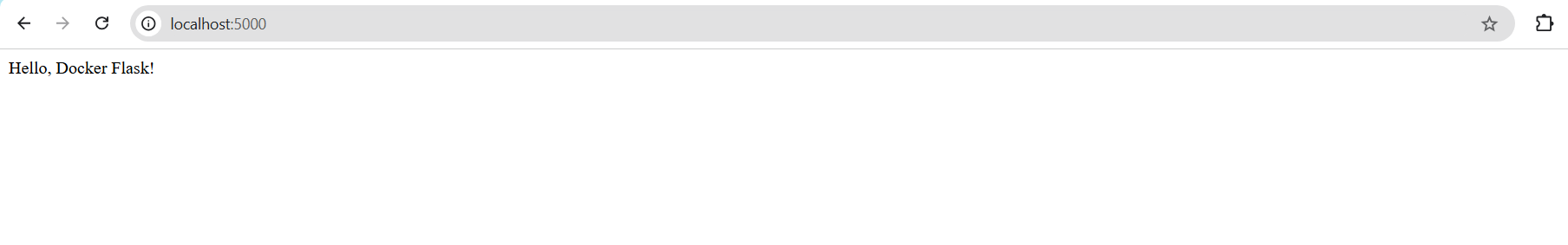
Yêu cầu:

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

Sử dụng python:3.9 làm base image.

A screenshot of a computer program

AI-generated content may be incorrect.

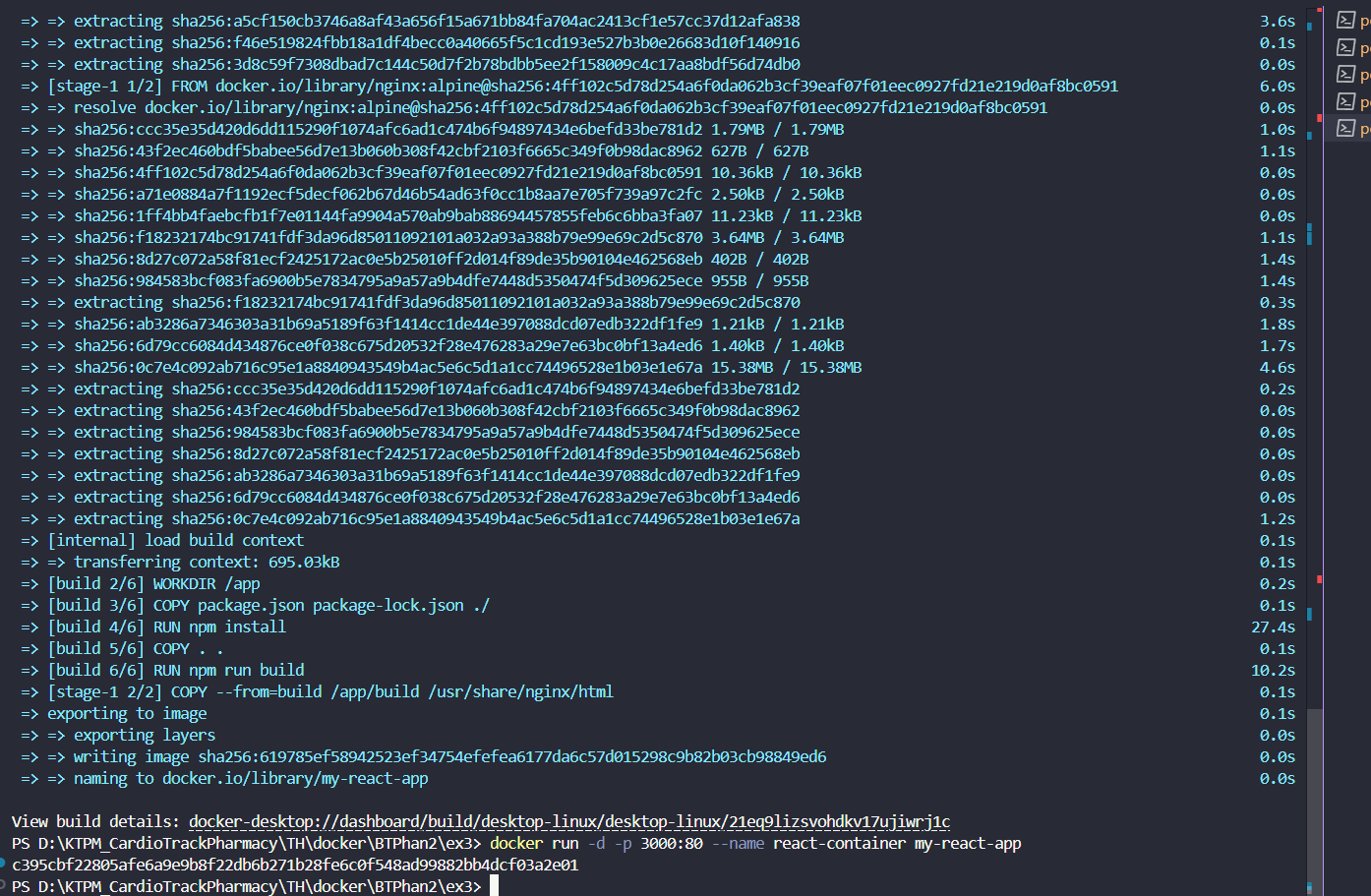


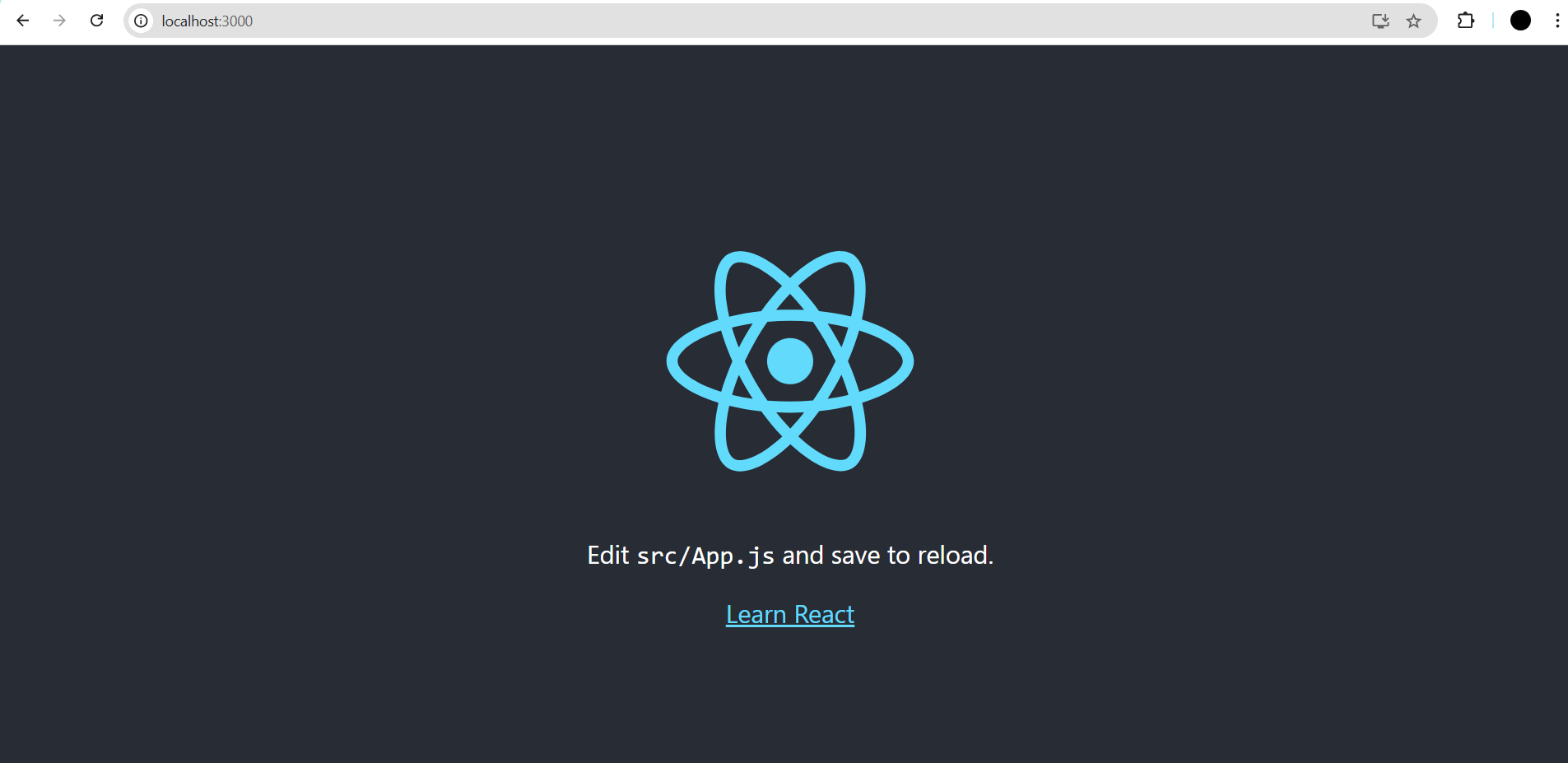
## Bài 3: Tạo Dockerfile chạy một ứng dụng React

Yêu cầu:

Viết Dockerfile để build và chạy một ứng dụng React.

Sử dụng node:18-alpine làm base image.





## Bài 4: Tạo Dockerfile chạy một trang web tĩnh bằng Nginx

Yêu cầu:

Tạo một file index.html đơn giản và sử dụng nginx:latest để phục vụ trang web.

## Bài 5: Tạo Dockerfile cho ứng dụng Go

Yêu cầu:

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

## 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.

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

Yêu cầu:

Viết Dockerfile cho ứng dụng Python đọc biến môi trường APP\_ENV và in ra màn hình.

Sử dụng ENV APP\_ENV=development trong Dockerfile.

## Bài 8: Tạo Dockerfile cho PostgreSQL tùy chỉnh

Yêu cầu:

Viết Dockerfile để chạy PostgreSQL (postgres:15).

Thêm file SQL để tự động tạo database khi container chạy lần đầu tiên.

## Bài 9: Tạo Dockerfile chạy Redis với cấu hình tùy chỉnh

Yêu cầu:

Viết Dockerfile sử dụng redis:latest.

Thêm file redis.conf vào container.

## Bài 10: Chạy ứng dụng PHP với Apache

Yêu cầu:

Viết Dockerfile để chạy một ứng dụng PHP đơn giản (php:8.2-apache).

Mount mã nguồn từ máy host vào container.