1. docker --version  
    ➤ Kiểm tra phiên bản Docker đang cài đặt trên hệ thống.
2. docker run hello-world  
    ➤ Chạy container kiểm tra xem Docker đã được cài đặt và hoạt động đúng chưa.
3. docker pull nginx  
    ➤ Tải image nginx mới nhất từ Docker Hub về máy.
4. docker images  
    ➤ Liệt kê tất cả các Docker image đang có trên máy.
5. docker run -d nginx  
    ➤ Chạy image nginx trong chế độ nền (detached).
6. docker ps  
    ➤ Hiển thị danh sách các container đang **chạy**.
7. docker ps -a  
    ➤ Hiển thị tất cả container (kể cả container đã dừng).
8. docker logs <container\_id>  
    ➤ Hiển thị log output của container (thay <container\_id> bằng ID cụ thể).
9. docker exec -it <container\_id> /bin/sh  
    ➤ Truy cập vào terminal của container (giống như SSH).
10. docker stop <container\_id>  
     ➤ Dừng container.
11. docker restart <container\_id>  
     ➤ Khởi động lại container.
12. docker rm <container\_id>  
     ➤ Xóa container (sau khi đã stop).
13. docker container prune  
     ➤ Xóa tất cả các container đã dừng.
14. docker rmi <image\_id>  
     ➤ Xóa một image cụ thể.
15. docker image prune -a  
     ➤ Xóa tất cả image không còn được sử dụng (gồm cả image chưa dùng lần nào).
16. docker run -d -p 8080:80 nginx  
     ➤ Chạy container nginx và ánh xạ cổng 8080 của máy host sang cổng 80 trong container.
17. docker inspect <container\_id>  
     ➤ Xem thông tin chi tiết về container (mạng, volume, env, v.v.).
18. docker run -d -v mydata:/data nginx  
     ➤ Gắn volume tên mydata vào thư mục /data trong container nginx.
19. docker volume ls  
     ➤ Liệt kê tất cả các volume trên hệ thống.
20. docker volume prune  
     ➤ Xóa tất cả volume không được gắn với bất kỳ container nào.
21. docker run -d --name my\_nginx nginx  
     ➤ Tạo container nginx và đặt tên là my\_nginx.
22. docker stats  
     ➤ Hiển thị thống kê sử dụng tài nguyên của các container đang chạy.
23. docker network ls  
     ➤ Hiển thị tất cả các network của Docker.
24. docker network create my\_network  
     ➤ Tạo một Docker network mới có tên my\_network.
25. docker run -d --network my\_network --name my\_container nginx  
     ➤ Chạy container nginx gắn vào network my\_network với tên my\_container.
26. docker network connect my\_network my\_nginx  
     ➤ Kết nối container my\_nginx vào network my\_network.
27. docker run -d -e MY\_ENV=hello\_world nginx  
     ➤ Chạy container nginx với biến môi trường MY\_ENV=hello\_world.
28. docker logs -f my\_nginx  
     ➤ Theo dõi log real-time của container my\_nginx.

dockerfile

# Sử dụng image Node.js chính thức

FROM node:18-alpine

# Đặt thư mục làm việc trong container

WORKDIR /app

# Sao chép file package.json và package-lock.json (nếu có)

COPY package\*.json ./

# Cài đặt các dependencies

RUN npm install

# Sao chép toàn bộ mã nguồn vào container

COPY . .

# Mở cổng 3000

EXPOSE 3000

# Lệnh chạy ứng dụng

CMD ["node", "server.js"]

1. docker build -t my-node-app.  
    ➤ Build Docker image từ Dockerfile hiện tại, gán tên là my-node-app.
2. docker run -d -p 3000:3000 my-node-app  
    ➤ Chạy image my-node-app, ánh xạ port 3000 của máy với port 3000 trong container.





























































