🐳 SỔ TAY DOCKER

**1. Docker là gì?**

**Docker** là một nền tảng mã nguồn mở giúp các nhà phát triển xây dựng, triển khai, chạy, cập nhật và quản lý các ứng dụng container hóa. Docker cho phép đóng gói ứng dụng và tất cả các phụ thuộc của nó vào một đơn vị tiêu chuẩn gọi là container, giúp ứng dụng có thể chạy một cách nhất quán trên nhiều môi trường khác nhau.

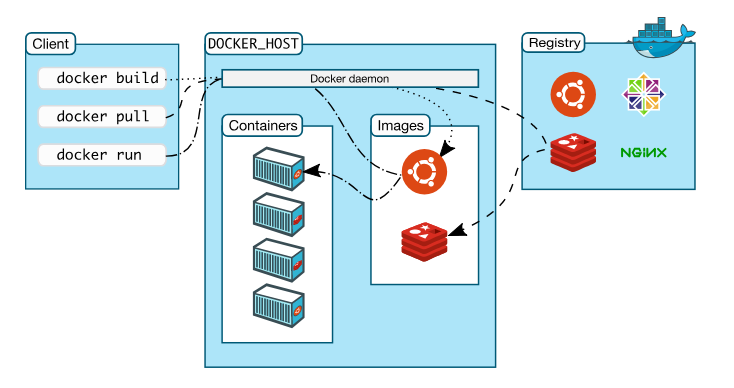
**2. Kiến trúc của Docker**

Docker sử dụng mô hình kiến trúc **client-server**, bao gồm các thành phần chính sau:

1. **Docker Client**: là giao diện dòng lệnh (CLI) cho phép người dùng tương tác với Docker daemon. Người dùng có thể sử dụng Docker client để gửi các lệnh như docker build, docker pull, docker run đến Docker daemon.
2. **Docker Daemon**: chạy trên hệ điều hành máy chủ và chịu trách nhiệm quản lý các container Docker. Docker daemon nhận các lệnh từ Docker client và thực hiện các tác vụ như xây dựng, chạy và phân phối các container.
3. **Docker Registry**: là nơi lưu trữ các Docker image. Docker Hub là một registry công cộng phổ biến, nhưng người dùng cũng có thể thiết lập registry và đẩy (push) các image lên registry.
4. **Docker Objects**: bao gồm Docker images, containers, networks và volumes. **Docker images** là các mẫu chỉ đọc dùng để tạo container. **Docker containers** là các instance đang chạy của Docker images. **Docker networks** cho phép các container giao tiếp với nhau. **Docker volumes** cung cấp khả năng lưu trữ dữ liệu bền vững cho các container.

- **Docker Image**: là một mẫu chỉ đọc chứa tất cả các thành phần cần thiết để chạy một ứng dụng, bao gồm mã nguồn, runtime, thư viện, và các công cụ hệ thống. Docker image được xây dựng từ một file cấu hình gọi là **Dockerfile**. Dockerfile chứa các lệnh để xây dựng image, chẳng hạn như cài đặt gói phần mềm và sao chép mã nguồn vào image.

- **Docker Container**: là một instance đang chạy của Docker image. Container cung cấp một môi trường cách ly để chạy ứng dụng, giúp đảm bảo rằng ứng dụng sẽ hoạt động nhất quán trên nhiều môi trường khác nhau. Container chia sẻ kernel của hệ điều hành chủ, nhưng có hệ thống file, mạng và không gian tiến trình riêng.



**3. Cài đặt Docker trên Windows**

- Tải Docker Desktop: truy cập trang web chính thức của Docker và tải Docker Desktop cho Windows: <https://www.docker.com/get-started>.

- Chạy File cài đặt: sau khi tải về, chạy file Docker Desktop Installer.exe. Trong quá trình cài đặt, nếu Windows chưa bật Hyper-V, sẽ có một checkbox hỏi xem có muốn bật Hyper-V luôn không, hãy chọn checkbox này để bật.

- Khởi động lại máy.

- Kiểm tra cài đặt: sau khi khởi động lại, mở Docker Desktop và kiểm tra xem Docker đã được cài đặt thành công chưa bằng cách chạy lệnh sau trong Command Prompt hoặc PowerShell: docker --version. Nếu Docker được cài đặt thành công, bạn sẽ thấy phiên bản Docker hiện tại.

- Chạy Container đầu tiên: để kiểm tra Docker hoạt động bình thường, chạy lệnh sau để kéo và chạy container hello-world: docker run hello-world. Nếu Docker hoạt động bình thường, bạn sẽ thấy thông báo “Hello from Docker!”.

**4. Các lệnh Docker cơ bản**

Dưới đây là một số lệnh Docker cơ bản mà bạn sẽ sử dụng thường xuyên khi làm việc với Docker:

1. **docker --version**:hiển thị phiên bản Docker hiện tại.

docker --version

1. **docker pull**: kéo một Docker image từ Docker Hub.

docker pull <image\_name>

1. **docker run**: tạo và chạy một container từ một Docker image.

docker run <image\_name>

1. **docker ps**: liệt kê các container đang chạy.

docker ps

1. **docker ps -a**: liệt kê tất cả các container, bao gồm cả các container đã dừng.

docker ps -a

1. **docker stop**: dừng một container đang chạy.

docker stop <container\_id>

1. **docker start**: khởi động lại một container đã dừng.

docker start <container\_id>

1. **docker rm**: xoá một container đã dừng.

docker rm <container\_id>

1. **docker rmi**: xoá một Docker image.

docker rmi <image\_name>

**5. Ví dụ tạo và chạy một ứng dụng web đơn giản với Docker**

Trong ví dụ này, chúng ta sẽ tạo và chạy một ứng dụng web đơn giản sử dụng Docker.

1. **Tạo Dockerfile**:

Tạo một file có tên là Dockerfile với nội dung sau:

A screenshot of a computer program

AI-generated content may be incorrect.

1. **Tạo file package.json**:

Tạo một file có tên là package.json với nội dung sau:

A screen shot of a computer code

AI-generated content may be incorrect.

1. **Tạo file app.js**:

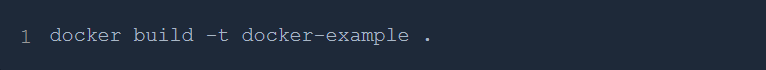
Tạo một file có tên là app.js với nội dung sau:

A screen shot of a computer code

AI-generated content may be incorrect.

1. **Xây dựng Docker image**:

Sử dụng lệnh docker build để xây dựng Docker image từ Dockerfile:



1. **Chạy Docker container**:

Sử dụng lệnh docker run để tạo và chạy một container từ Docker image vừa xây dựng:

A blue background with white text

AI-generated content may be incorrect.

1. **Truy cập ứng dụng**:

Mở trình duyệt web và truy cập <http://localhost:8080>. Bạn sẽ thấy thông báo “Hello, Docker!”.

**6. Lợi ích của Docker**

Docker mang lại nhiều lợi ích cho quá trình phát triển và triển khai ứng dụng:

- **Tính nhất quán**: Docker đảm bảo rằng ứng dụng sẽ hoạt động nhất quán trên nhiều môi trường khác nhau, từ máy phát triển đến máy chủ sản xuất.

- **Tính di động**: Docker container có thể chạy trên nhiều nền tảng khác nhau, bao gồm máy tính cá nhân, máy chủ vật lý, máy ảo, và các dịch vụ đám mây.

- **Hiệu quả tài nguyên**: Docker container nhẹ hơn so với máy ảo vì chúng chia sẻ kernel của hệ điều hành chủ và chỉ bao gồm các thành phần cần thiết, để chạy ứng dụng.

- **Tính linh hoạt**: Docker cho phép dễ dàng mở rộng và thu nhỏ ứng dụng bằng cách thêm hoặc xoá các container.

- **Tích hợp IC/CD**: Docker tích hợp tốt với các công cụ CI/CD, giúp tự động hoá quá trình xây dựng, kiểm thử và triển khai ứng dụng.