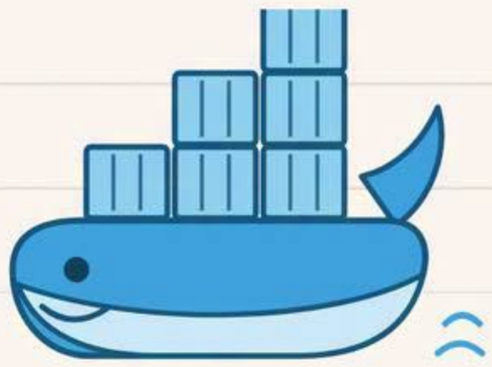
DOCKER



Docker là gì?

Docker là một nền tảng mã nguồn mở giúp chúng ta build, đóng gói và chạy ứng dụng bên trong các container.

Vì sao Docker ra đời?

Để giải quyết vấn đề "máy tôi chạy được mà". Docker đảm bảo ứng dụng chạy giống nhau ở mọi môi trường.

Docker hữu ích ở điểm nào?

- Portability : Chạy được ở mọi nơi (dev, test, prod)
- Lightweight : Container dùng ít tài nguyên hơn
- Fast Deployment : Build, test và deploy nhanh chóng
- Isolation : Mỗi container chạy trong môi trường riêng
- Consistency : Môi trường như nhau ở mọi nơi

CÁC KHÁI NIỆM QUAN TRỌNG

1. **Dockerfile** — Tệp văn bản chứa các chỉ dẫn để build một Docker image.
2. **Image** — Khuôn mẫu chỉ đọc được dùng để tạo ra container.
3. **Layers** — Một image được tạo nên từ nhiều lớp (layer) chỉ đọc.
4. **Registry** — Nơi lưu trữ và chia sẻ các Docker image (vd: Docker Hub).
5. **Engine** — Thành phần lõi của Docker, lo việc build, chạy và quản lý container.
6. **Container** — Một phiên bản đang chạy của Docker image.

Ghi nhớ : Image là bản thiết kế, Container là ứng dụng đang chạy.

DOCKERFILE

Dockerfile là một tệp văn bản chứa tập hợp các chỉ dẫn để build ra Docker image một cách tự động.

Dockerfile hoạt động thế nào?

Viết chỉ dẫn trong Dockerfile

Docker build ra image

Image dùng để chạy container

Ứng dụng chạy bên trong container

Các chỉ dẫn thường dùng trong Dockerfile

FROM	Đặt image nền (base image).
WORKDIR	Đặt thư mục làm việc bên trong container.
COPY	Sao chép tệp hoặc thư mục từ host vào container.
RUN	Thực thi câu lệnh trong lúc build image.
EXPOSE	Báo cho Docker biết container lắng nghe ở cổng nào.
CMD	Chỉ định lệnh mặc định chạy khi container khởi động.

Ví dụ Dockerfile

```
1 FROM node:20                  # Image nền
2 WORKDIR /app                  # Thư mục làm việc
3 COPY package*.json ./         # Copy file deps
4 RUN npm install               # Cài dependency
5 COPY . .                      # Copy toàn bộ file
6 EXPOSE 3000                   # Mở cổng
7 CMD ["npm","start"]           # Khởi chạy app
```

Giải thích

- Dùng image chính thức Node.js 20 làm nền.
- Đặt /app làm thư mục làm việc.
- Copy package.json để cài dependency trước (tận dụng cache tốt hơn).
- Cài toàn bộ dependency khai báo trong package.json.
- Copy phần code còn lại của ứng dụng.
- Container sẽ lắng nghe ở cổng 3000.
- Chạy `npm start` khi container khởi động.

Build và Run

- Build Image

```
$ docker build -t myapp .
```

. (dấu chấm) là thư mục hiện tại

- Chạy Container

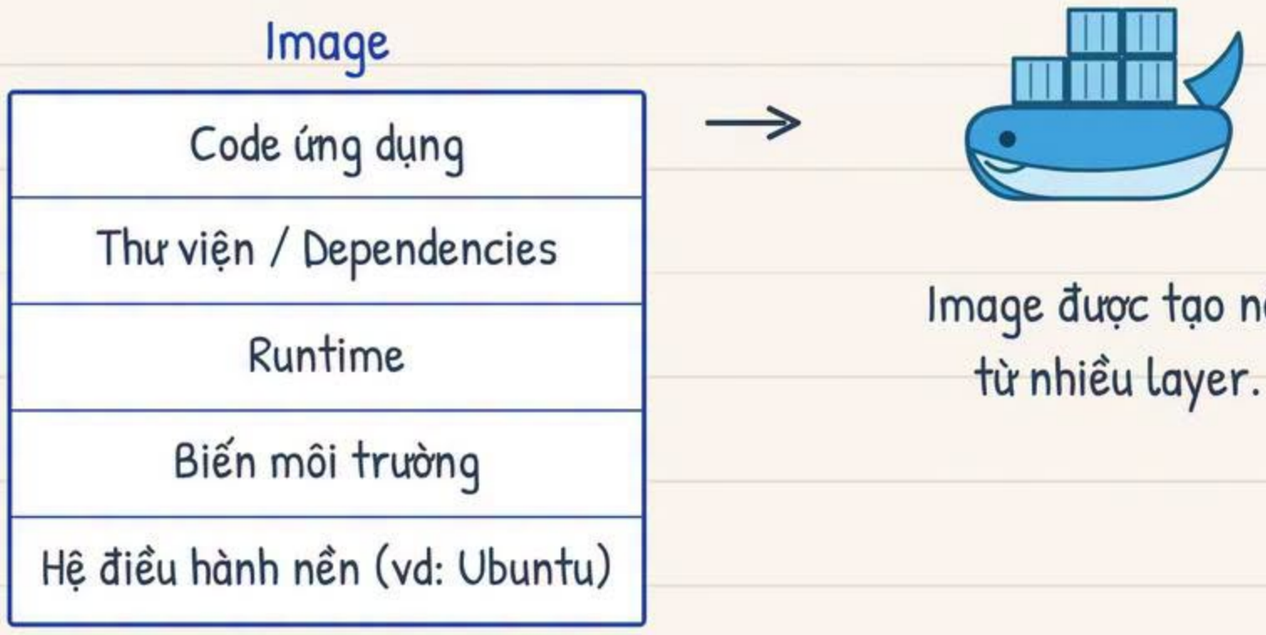
```
$ docker run -p 3000:3000 myapp
```

Cổng host — Cổng container

Dockerfile giúp việc tạo image lặp lại được, dễ mang đi và đơn giản.

★ Docker Image là gì?

- Image là khuôn mẫu chỉ đọc chứa các chỉ dẫn để tạo ra container.
- Nó chứa code ứng dụng, runtime, thư viện, biến môi trường và tệp tin.
- Image là bất biến (immutable).



★ Docker Container là gì?

- Container là phiên bản đang chạy của một image.
- Nó có filesystem, network, process space riêng nhưng dùng chung kernel của hệ điều hành host.
- Container có vòng đời ngắn (dễ dàng tạo, khởi động, dừng và xóa).

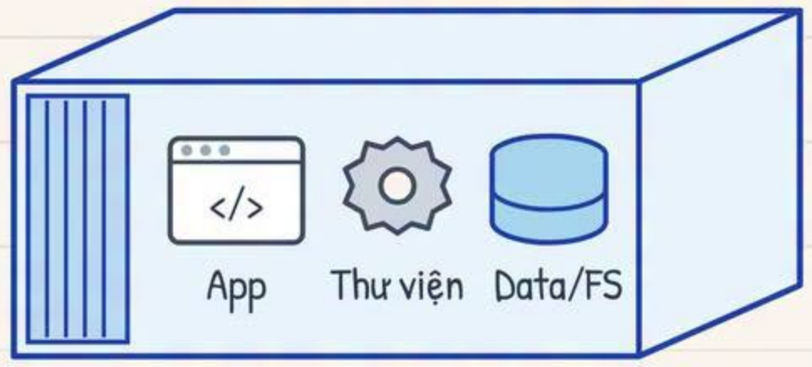


Image so với Container

Docker Image	Docker Container
• Khuôn mẫu chỉ đọc	• Phiên bản đang chạy của image
• Được tạo bằng Dockerfile	• Được tạo ra từ một image
• Bất biến	• Thay đổi được (chỉ tạm thời)
• Lưu ở Docker Hub hoặc máy local	• Tồn tại trên Docker engine (local)
• Không có trạng thái	• Có trạng thái (running/stopped)
• Có thể chia sẻ và tái sử dụng	• Không thể chia sẻ

Vòng đời của Container

★ Điểm cốt lõi

✓ Image = Bản thiết kế

✓ Container = Tiến trình đang chạy

✓ Image chỉ build một lần, container có thể tạo nhiều lần.

✓ Container nhanh, nhẹ và dễ mang đi.

DOCKER COMPOSE

Docker Compose là gì?

Docker Compose là công cụ để định nghĩa và chạy các ứng dụng Docker gồm nhiều container. Ta khai báo services, networks và volumes của ứng dụng trong tệp docker-compose.yml rồi điều phối mọi thứ bằng vài câu lệnh đơn giản.

Vì sao nên dùng Docker Compose?

- Quản lý ứng dụng nhiều container dễ dàng
- Chỉ một tệp cấu hình duy nhất
- Khởi động, dừng và quản lý bằng một lệnh
- Môi trường nhất quán cho cả nhóm

Cơ chế hoạt động

Ví dụ : docker-compose.yml

```yaml
version: '3.8'
services:
  web:
    image: nginx:latest
    ports:
      - "80:80"
    depends_on:
      - db
  db:
    image: mysql:8
    environment:
      MYSQL_ROOT_PASSWORD: root
    volumes:
      - db_data:/var/lib/mysql
volumes:
  db_data:
```

version : Phiên bản định dạng tệp Compose

services : Danh sách các container (service) trong ứng dụng

web / db : Định nghĩa của từng service

- image — Docker image sẽ dùng
- ports — Mở cổng của container
- depends_on — Khởi động db trước web
- environment — Đặt biến môi trường
- volumes — Lưu dữ liệu bền vững

Các lệnh thường dùng

- docker compose up ⟶ Khởi động tất cả service
- docker compose up -d ⟶ Chạy ở chế độ detached (nền)
- docker compose down ⟶ Dừng và xóa container, network
- docker compose ps ⟶ Liệt kê service đang chạy
- docker compose logs ⟶ Xem log của các service
- docker compose build ⟶ Build hoặc build lại service

Chỉ một tệp docker-compose.yml có thể quản lý cả một application stack!

Ghi nhớ : Compose rất hợp cho development, testing và triển khai quy mô nhỏ.
Với production quy mô lớn, hãy dùng công cụ orchestration như Docker Swarm hoặc Kubernetes.

5 LỆNH DOCKER QUAN TRỌNG

Chỉ 5 câu lệnh này là đủ để bạn bắt đầu với Docker!

1 docker build

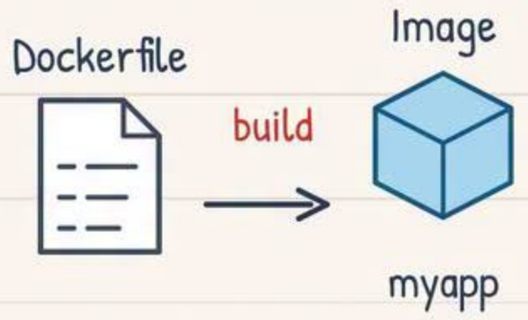
Build ra một image từ Dockerfile và gắn tên (tag) cho nó.

Cú pháp :

```
docker build -t <image-name> <path>
```

Ví dụ :

```
docker build -t myapp .
```



2 docker run

Tạo và khởi động một container từ image.

Cú pháp :

```
docker run -d -p <host-port>:<container-port> <image-name>
```

Ví dụ :

```
docker run -d -p 8080:80 nginx
```

3 docker ps

Liệt kê tất cả container đang chạy.

Cú pháp :

```
docker ps
```

Ví dụ :

```
docker ps
```

Container đang chạy

CONTAINER ID	IMAGE	STATUS	PORTS
a1b2c3d4e5f6	nginx	Up 5 min	0.0.0.0:8080 → 80/tcp
----	--	--	---

4 docker stop

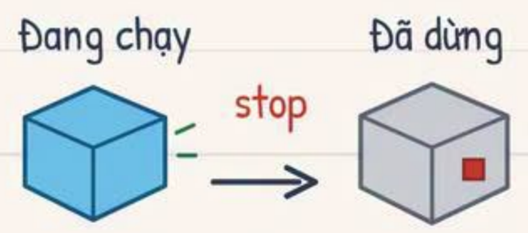
Dừng một container đang chạy.

Cú pháp :

```
docker stop <container-id / name>
```

Ví dụ :

```
docker stop myapp
```



5 docker rm

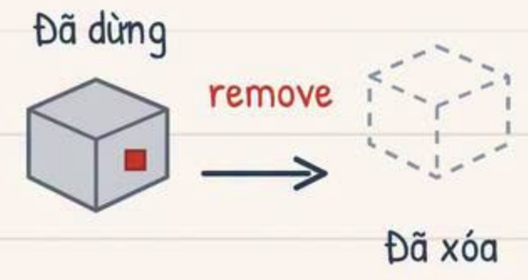
Xóa một container đã dừng khỏi hệ thống.

Cú pháp :

```
docker rm <container-id / name>
```

Ví dụ :

```
docker rm myapp
```



Ghi nhớ : • Dùng -d để chạy container ở chế độ detached (chạy nền).
• Dùng -p để publish (ánh xạ) cổng.