

BÁO CÁO TỔNG HỢP KIẾN THỨC DOCKER

Giải thích Docker là gì

Docker là một công nghệ giúp đóng gói ứng dụng và toàn bộ thư viện, công cụ, phiên bản framework... vào một gói gọn gọi là container. Khi dùng container thì người phát triển, người test, người triển khai đều chạy đúng một môi trường giống nhau, nên tránh được việc chạy trên máy mình thì được nhưng trên máy người khác thì không chạy.

Docker nhẹ hơn nhiều so với máy ảo vì không cần guest OS riêng cho mỗi ứng dụng. Các container chia sẻ kernel nhưng vẫn độc lập, không ảnh hưởng nhau. Việc chạy, tắt, xoá container rất nhanh.

Docker hoạt động theo mô hình client – server. Docker daemon xử lý các yêu cầu qua REST API, còn Docker client gửi lệnh. Image là mẫu có sẵn, container là phiên bản đang chạy của image. Image có thể chia sẻ qua Docker Hub hoặc registry riêng.

Khi một dự án có nhiều ứng dụng khác nhau thì việc cài thủ công từng thứ trên từng máy mất thời gian. Docker giải quyết vấn đề này vì chỉ cần gửi image cho nhau, không phải cài lại các phần mềm như trước. Container chạy độc lập, nhẹ, khởi động nhanh, dễ mở rộng.

Docker Container

Container giống như gói tất cả trong một: ứng dụng, dependencies, thư viện, file cấu hình. Khi chạy thì container tách biệt hoàn toàn với máy chủ.

Ví dụ có thể có container chạy Java + Tomcat, container chạy Python + MySQL, container chạy .NET + SQL Server. Tất cả đều chạy chung trên Docker engine mà không đụng nhau.

Container giải quyết việc mỗi người dùng một máy khác nhau: dev cài JBoss, tester cài JBoss nhưng khác phiên bản, system admin cài khác nữa nên khó đồng nhất. Dùng container thì mọi người dùng cùng một gói.

Container chạy nhanh, ít tốn tài nguyên, dễ chia sẻ, và có thể bật/tắt tức thì. Nhiều container chạy được cùng lúc trên một máy.

Dockerfile và cách tạo image

Dockerfile là file văn bản chứa danh sách các lệnh dùng để tạo ra Docker image. Khi chạy docker build, Docker sẽ đọc từng dòng trong Dockerfile và tạo image theo từng lớp (layer).

Dockerfile gồm các lệnh như:

FROM chọn image nền

RUN chạy lệnh trong môi trường build

COPY hoặc ADD để đưa file vào trong image

WORKDIR đặt thư mục làm việc

CMD hoặc ENTRYPOINT để chỉ định lệnh khi container chạy

ENV đặt biến môi trường

Image được tạo ra là dạng read-only. Khi chạy container mới có lớp read-write phía trên.

Hiểu được Dockerfile thì có thể tạo môi trường tùy chỉnh, ví dụ ứng dụng Python, Node.js, Java... Việc viết Dockerfile giúp tự động hóa quá trình dựng môi trường, có thể đưa vào CI/CD.

Một số bước thực hành

1. Tạo container

Chạy docker run

Kiểm tra container bằng *docker ps*

Đổi tên container bằng *docker rename*

Dừng container bằng *docker stop*

Xoá container bằng *docker rm*

2. Tạo image bằng Dockerfile (Python)

Tạo file Dockerfile với FROM python

COPY mã nguồn vào container

RUN pip install

Build image bằng *docker build -t python-app .*

Kiểm tra bằng *docker images*

Docker Compose

Khi có nhiều container liên quan nhau (ví dụ web + database), việc chạy thủ công từng cái bằng docker run khá mất thời gian. Docker Compose dùng file docker-compose.yml để định nghĩa nhiều service cùng lúc. Chỉ cần một lệnh là chạy tất cả.

File YAML có cấu trúc:

version: "3.5"

services:

web:

 image: nginx

db:

 image: redis

Lệnh:

docker-compose up -d để chạy

docker-compose down để tắt

docker-compose config để kiểm tra lỗi file cấu hình

Compose rất phù hợp cho môi trường dev/test vì có thể khởi động cả hệ thống bằng 1 lệnh. YAML dễ đọc và dễ sửa. Compose hoạt động trên 1 máy (single host).

Docker Swarm

Swarm được dùng khi cần chạy Docker trên nhiều máy khác nhau như một cụm (cluster). Trong Swarm có:

- Manager node: điều khiển toàn bộ cụm
- Worker node: thực thi container theo lệnh từ manager

Swarm có những khả năng sau:

- tự cân bằng tải
- phục hồi container khi lỗi
- tạo môi trường phân tán
- giao tiếp bảo mật cao
- có thể rollback

Service là mô tả trạng thái mong muốn (ví dụ chạy 5 bản sao web). Task là container thật đang chạy. Có 2 kiểu service:

- replicated: chạy số lượng bản sao tùy chọn
- global: chạy đúng 1 container trên mỗi node

Swarm phù hợp khi cần vận hành thật, nhiều server hoạt động cùng lúc.

Điều muôn tìm hiểu thêm

1. Cách triển khai CI/CD kết hợp Docker, Compose và Swarm.
2. Cách tối ưu Dockerfile để image nhỏ nhất và chạy nhanh nhất.