

Lesson 4

Introduction to Docker Compose

1. Sử dụng Dockerfile và Image như nào hiệu quả ?

2. Tìm hiểu về Docker Compose

3. Host Docker Registry

1. Sử dụng Dockerfile và Image như nào hiệu quả ?

Docker Image:

- Sử dụng những Image có độ tin cậy cao, và được nhiều lượt đánh giá.
- Sử dụng đúng **version** của Image, mà bạn đang muốn dùng. Hạn chế sử dụng version **latest**.
- Sử dụng những Image đã được tối ưu với kích thước nhỏ.
- Nên sử dụng **docker scan** để kiểm tra **mức độ an toàn** của Image

1. Sử dụng Dockerfile và Image như nào hiệu quả ?

Dockerfile:

- Sắp xếp các Layer Instruction theo đúng thứ tự và logic.
- Sử dụng Cache Layer một cách hiệu quả
- Cố gắng tối ưu cho mỗi Layer xử lý không quá lớn.
- Khi sử dụng những lệnh như COPY không nên COPY những file không cần thiết.
- Nếu trong Dockerfile có sử dụng Package Manager như NPM, PIP... , nên Install những package có sử dụng.

1. Sử dụng Dockerfile và Image như nào hiệu quả ?

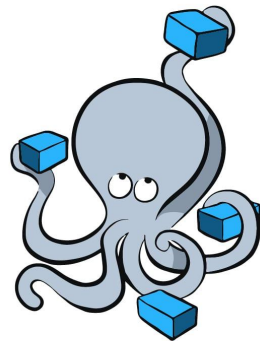
Dockerfile:

- Không nên sử dụng quá nhiều Layer Instruction trong Dockerfile.
- Có thể chia thành Multi-Stage để dễ dàng quản lý.
- Sử dụng Image phù hợp với Application đang Build.
- Sử dụng .dockerignore để tránh thêm những thư mục không cần thiết.

2. Tìm hiểu Docker Compose

Docker Compose:

- Công cụ hỗ trợ để run một hoặc nhiều container cùng một thời điểm bằng cách sử dụng file yaml.
- Bên trong một Docker Compose sẽ define các cấu hình cần thiết để run Container.



docker
Compose

Tại sao phải sử dụng Docker Compose:

- Giúp giảm thời gian Run Multi Container cùng một thời điểm
- Chỉ tạo lại Container khi có sự thay đổi
- Cài đặt nhanh chóng và dễ dàng
- Khả năng kết nối các Container tốt hơn



Sự khác nhau giữa Dockerfile và Docker Compose:

Dockerfile sẽ build một Image mỗi lần, còn với Docker Compose có thể group các Image và build tất cả Image bằng một Command



Cấu trúc của một Docker Compose YAML:

- version
- services
 - define service
- networks
- volume

```
version: '3.9'

services:
  mysqldb:
    platform: linux/amd64
    image: mysql:5.7
    restart: unless-stopped
    environment:
      - MYSQL_ROOT_PASSWORD=root
      - MYSQL_DATABASE=root
    ports:
      - 3306:3306
    volumes:
      - db:/var/lib/mysql
  app:
    depends_on:
      - mysqldb
    build: ../Docker/
    restart: unless-stopped
    ports:
      - 8080:8080
    environment:
      - DB_HOST=mysqldb
      - DB_USER=root
      - DB_PASSWORD=root
      - DB_NAME=root
      - DB_PORT=3306

volumes:
  db:
```

version: chọn phiên bản Docker Compose mà bạn muốn sử dụng

services: định nghĩa những thông tin về Container, Docker Compose sẽ sử dụng những thông tin này để Build và Run

networks: định nghĩa những network nào sẽ sử dụng khi Docker Compose build và run

volumes: định nghĩa những volume nào sẽ sử dụng khi Docker Compose build và run

service:

- **image:** Image sử dụng để tạo Container
- **build:** để Build Image và Run Container, nếu không muốn sử dụng Image có sẵn.

Thay thế cho **thuộc tính image**

- **context:** đường dẫn tới thư mục chứa Dockerfile
- **dockerfile:** chỉ định tên Dockerfile, nếu trong thư mục có nhiều Dockerfile
- **args:** định nghĩa Arguments để sử dụng trong quá trình Build
- **depend_on:** khai báo những Container phụ thuộc, khi những Container phụ thuộc Run hoàn thành, thì Container này sẽ được Run
- **ports:** khai báo những Port được sử dụng trong Container

service:

- **env_file:** đường dẫn tới file env, sử dụng trong quá trình Run Container
- **environments:** khai báo cụ thể các biến môi trường sử dụng trong quá trình Run Container
- **networks:** khai báo những network mà Container sẽ sử dụng trong quá trình Run
- **command:** giống như Dockerfile, khai báo command mà Container sẽ thực thi khi Run

service:**- restart:**

- **no**: khi Container xảy ra lỗi hoặc stop, sẽ không khởi động lại
- **always**: Container sẽ luôn khởi động lại khi Container bị stop. Container sẽ tự động Run, nếu Docker Daemon Restart
- **on-failure**: Container sẽ chỉ khởi động lại khi xảy ra lỗi. Container sẽ không tự động Run, nếu Docker Daemon Restart.
- **unless-stopped**: Container luôn khởi động lại khi xảy ra lỗi, ngoại trừ container bị Stop. Container sẽ không tự động Run, nếu Container bị manual stop

Health check trong Docker Compose:

Trong quá trình Container Run, đôi khi xảy ra vấn đề khiến Container bị dừng đột ngột, vì vậy cần có một cơ chế để kiểm tra trạng thái của Container định kỳ.

healthcheck:

- **test:** chứa câu lệnh thực thi để kiểm tra trạng thái của Container
- **interval:** khai báo thời gian mà healthcheck sẽ chạy để kiểm tra
- **timeout:** nếu Container timeout quá thời gian khai báo thì Container Unhealthy
- **retries:** số lần healthcheck sẽ kiểm tra nếu vượt qua số này thì Container Unhealthy

Những câu lệnh sử dụng trong Docker Compose:

- docker compose build: sử dụng để Build Image
- docker compose run: sử dụng để run Container, từ những Image đã Build trước đó
- docker compose up: Build Image và Run Container
- docker compose down: Stop Container, sau đó là xoá Container và Image

3.Host Docker Image

- Sử dụng Image Registry Docker có sẵn để run Container
- Sử dụng Image đã build từ Dockerfile để Push lên Host Registry
- Sau đó Pull Image từ Host Registry và Run