Docker Compose là gì? Các lệnh cơ bản trong Docker Compose

24 Feb, 2023

DevOps

Pum
Author

Docker Compose là một công cụ hỗ trợ xác định và chạy các ứng dụng nhiều container. Chúng có thể xử lý đồng thời multi-container trong sản xuất

Mục Lục

Phát triển một ứng dụng sẽ gồm rất nhiều container chạy trên các services khác nhau, việc khởi động và quản lý các container sẽ trở nên khó khăn. Vì vậy, Docker đã tạo ra một công cụ hữu ích giúp tăng tốc quá trình này - **Docker Compose**.

Bạn có thể tham khảo bài viết này để hiểu rõ hơn về Docker nhé!

Docker là gì? Khi nào nên dùng Docker?

Docker Compose là gì?

Docker Compose là gì?

Docker Compose là một công cụ hỗ trợ xác định và chạy các ứng dụng multi-container . Docker Compose có thể xử lý đồng thời multi-container trong sản xuất, staging, phát triển, thử nghiệm và CI.

Docker Compose hoạt động bằng cách áp dụng các quy tắc được xác định trong tệp **docker-compose.yaml.**

Giả sử bạn đang phát triển một ứng dụng web có cấu trúc như sau:

- Một container chứa ứng dụng web Node.js, chạy trên cổng 3000.
- Một container chứa cơ sở dữ liệu MySQL, chạy trên cổng 3306.

Để triển khai ứng dụng này bằng Docker Compose, bạn cần tạo một tệp docker-compose.yml với các thông tin sau:

Yaml

```yaml
version: "3.9"
services:
  web:
    build: .

```

```
    depends_on:
      - db
  db:
    image: mysql:latest
    environment:
      MYSQL_ROOT_PASSWORD: password
    ports:
      - "3306:3306"
```

Trong đó:

- `version`: phiên bản của Docker Compose
- `services`: danh sách các dịch vụ cần triển khai
- `web`: định nghĩa dịch vụ web, bao gồm việc build image từ Dockerfile và ánh xạ cổng 3000 của container với cổng 3000 của máy host.
- `depends_on` cho biết dịch vụ này cần phụ thuộc vào dịch vụ `db`.
- `db`: định nghĩa dịch vụ cơ sở dữ liệu MySQL, bao gồm việc sử dụng image MySQL, đặt mật khẩu cho root user và ánh xạ cổng 3306 của container với cổng 3306 của máy host.

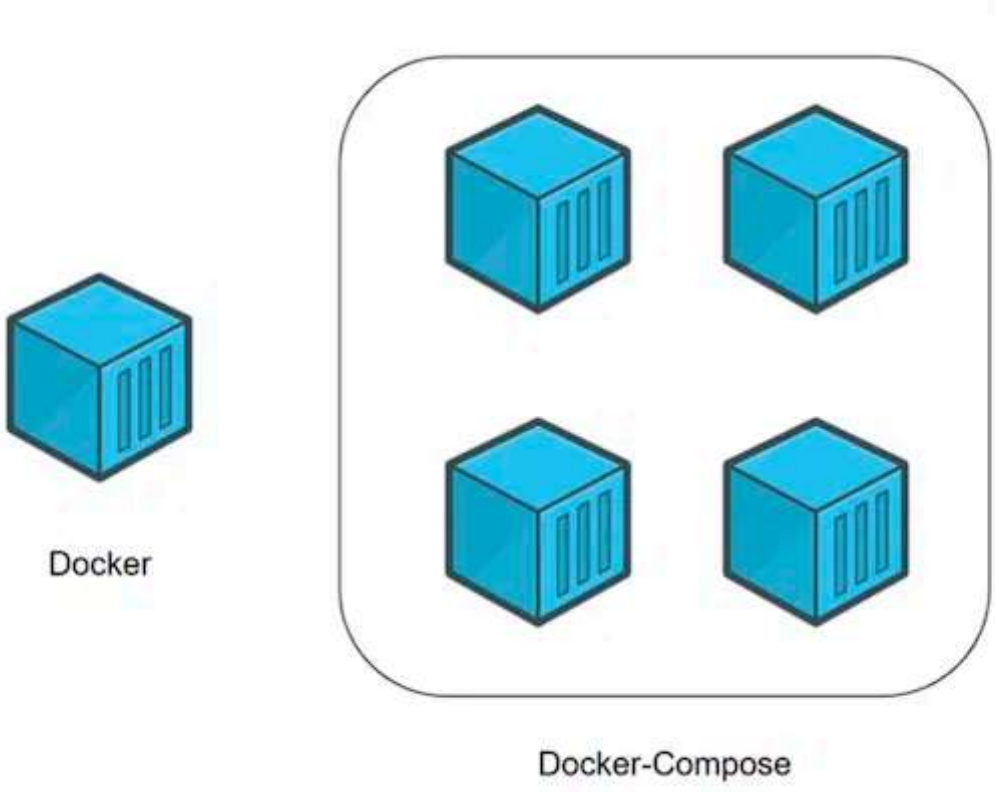
Các lệnh cơ bản trong Docker Compose

- **docker-compose up**: Khởi động các container
- **docker-compose down**: Dừng và xóa các container
- **docker-compose ps**: Hiển thị trạng thái của các container
- **docker-compose build**: Tạo image từ Dockerfile trong mỗi dịch vụ
- **docker-compose restart**: Khởi động lại các container
- **docker-compose stop**: Dừng các container

- **docker-compose config**: Hiển thị các cấu hình của Docker Compose
- **docker-compose exec**: Thực thi một lệnh trên một container
- **docker-compose port**: Hiển thị các port của các container
- **docker-compose top**: Hiển thị các process đang chạy trong các container

Lưu ý: Các lệnh trên phải được thực hiện trong thư mục chứa file docker-compose.yml.

Cài đặt Docker Compose



Cài đặt Docker Compose

Theo từng hệ điều hành sẽ có bước cài đặt Docker Compose khác nhau. Bạn nên tải phiên bản mới nhất của Docker Compose từ trang web chính thức của Docker Compose.

Cài đặt Docker Compose trên macOS

Cài đặt Docker Compose trên Windows

Để sử dụng Docker Compose trên Windows, bạn chỉ cần cài đặt Docker Desktop cho Windows và không cần cài đặt riêng Docker Compose.

Cài đặt Docker Compose trên Linux

- Hãy cấp quyền thực thi cho tệp tin Docker Compose binary bằng lệnh sau:

Bash

```
sudo chmod +x /path/to/docker-compose
```

Trong đó, "/path/to/docker-compose" là đường dẫn tới tệp tin Docker Compose binary.

- Di chuyển tệp tin Docker Compose binary vào một trong các đường dẫn $PATH trên hệ thống của bạn để có thể truy cập nó từ bất kỳ thư mục nào.

Ví dụ, để di chuyển tệp tin Docker Compose binary vào thư mục /usr/local/bin trên Linux, bạn có thể sử dụng lệnh sau:

Bash

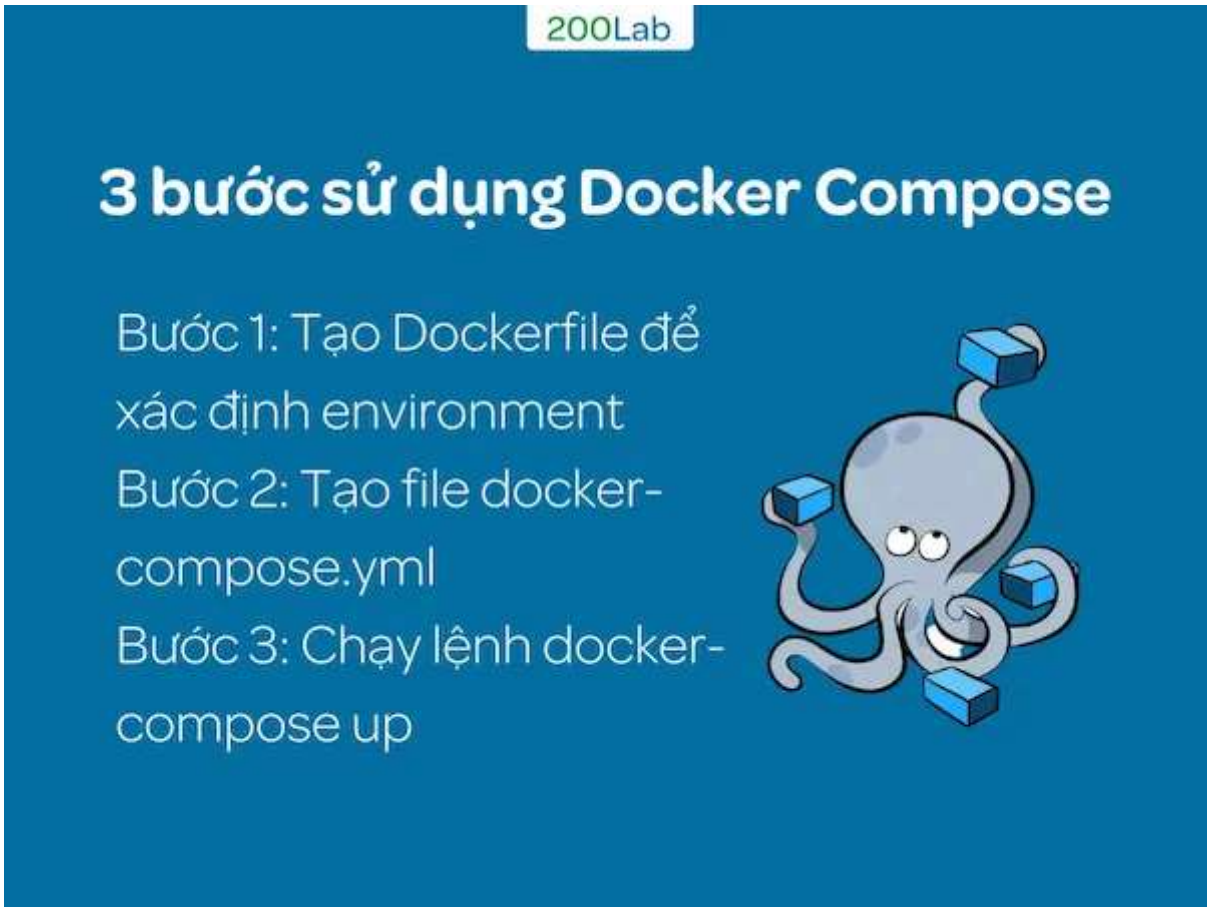
```
sudo mv /path/to/docker-compose /usr/local/bin/docker-compose
```

- Kiểm tra xem Docker Compose đã được cài đặt thành công bằng cách chạy lệnh sau:

CSS

```
docker-compose --version
```

Sử dụng Docker Compose



Sử dụng Docker Compose

Để sử dụng Docker Compose, bạn cần thực hiện các bước sau:

Bước 1: Cài đặt Docker Compose

Bạn cần cài đặt Docker Compose trên máy tính của mình. Bạn có thể tải từ trang chủ của Docker hoặc có thể cài đặt thông qua các gói phần mềm của hệ điều hành.

Tiếp đến, bạn sẽ tạo Dockerfile để xác định environment. Mỗi dịch vụ sẽ chạy trên một container riêng và sử dụng image tương ứng.

Bạn cần tạo một file docker-compose.yml để định nghĩa các container và cấu hình của chúng. Ví dụ, tệp sẽ bao gồm các nội dung sau:

Yaml

```yaml
version: '3'

services:
  web:
    image: nginx:latest
    ports:
      - "80:80"
    volumes:
      - ./web:/usr/share/nginx/html
    networks:
      - webnet
    depends_on:
      - db
  db:
    image: postgres:latest
    environment:
      POSTGRES_USER: example
      POSTGRES_PASSWORD: example
      POSTGRES_DB: example
    volumes:
      - dbdata:/var/lib/postgresql/data
    networks:
      - webnet

networks:
  webnet:


```

Ý nghĩa của các giá trị trong file docker-compose:

- **version**: phiên bản của Docker Compose file. Ở đây, chúng ta sử dụng phiên bản 3.
- **services**: là khu vực khai báo các services cần thiết cho ứng dụng.
- **web**: dịch vụ web, sử dụng image nginx, chia sẻ volume và network với dịch vụ db. Cổng 80 được định nghĩa để web có thể truy cập được từ bên ngoài.
- **db**: dịch vụ db, sử dụng image postgresql, chia sẻ volume và network với dịch vụ web. Các biến môi trường cài đặt cho dịch vụ postgresql được định nghĩa ở phần environment.
- **networks**: danh sách các networks được sử dụng cho container.
- **webnet**: mạng webnet để chia sẻ giữa dịch vụ web và db.
- **volumes**: là option nên config, volumes cho phép mount data từ container ra máy local. Khi config option này thì mỗi lần stop container data của container đó sẽ không bị mất đi.
- **dbdata**: là container chứa thông tin về database.

Trong ví dụ này, Docker Compose sẽ khởi tạo 2 container, một container sử dụng image nginx và một container sử dụng image postgresql. Container sử dụng image nginx sẽ được kết nối với container sử dụng image postgresql thông qua mạng webnet. Sau đó, chúng sẽ chia sẻ volume dbdata để lưu trữ dữ liệu của postgresql.

Bước 3: Chạy lệnh docker-compose up

Sử dụng lệnh `docker-compose up` để khởi động các container được định nghĩa trong file docker-compose.yml. Nếu các image không được tải xuống trước đó, Docker Compose sẽ tự động tải chúng xuống và khởi động các container.

Bạn có thể quản lý các container bằng các lệnh Docker Compose như `docker compose stop` để dừng các container, `docker compose start` khởi động sau khi dừng các container, `docker compose restart` để khởi động lại các container đã dừng, `docker compose -f docker.yaml down` để xóa các container đã dừng, `docker-compose ps` để hiển thị trạng thái của các container, ...

Tùy chỉnh cấu hình

Nếu bạn muốn thay đổi cấu hình của các container, bạn chỉ cần chỉnh sửa **file docker-compose.yml** và chạy lại lệnh `docker-compose up`.

Lưu ý: các lệnh Docker Compose phải được thực hiện trong thư mục chứa file docker-compose.yml.

Lời kết

200Lab hy vọng bạn đã hiểu rõ **Docker Compose** là gì thông qua những ví dụ minh hoạ ở trên. Bạn nhớ theo dõi 200Lab để không bỏ lỡ những chủ đề thú vị khác liên quan đến Docker, hẹn gặp lại bạn trong những bài viết sau nhé!

Nếu bạn có định hướng trở thành **DevOps chuyên nghiệp** thì bạn có thể tham khảo khoá học **DevOps for Backend Developer** tại đây.

Bài viết liên quan:

- Docker: Deploy Nginx, Let's Encrypt web service có SSL đơn giản nhất
- VirtualBox là gì? Hướng dẫn tạo máy ảo Ubuntu trên VirtualBox
- Microservices: Những sai lầm và chiến lược chuyển đổi từ Monolith
- Tất cả những điều cần biết về Microservices

Backend

MinIO: Giải pháp Object Storage Mã nguồn mở thay thế S3

Backend DevOps

DevOps

MLOps là gì? Những Kỹ r

DevOps Data

Đăng ký nhận thông báo

Nhập email của bạn

Đăng kí

200Lab

Công ty TNHH công nghệ 200LAB
MST: 0317035178
Văn phòng: 9A Phạm Văn Hai, P1, Tân Bình, TP.HCM
CN1: 70 Huỳnh Văn Bánh, P15, Phú Nhuận, TP.HCM
Hotline: 0777098082