**Phần 8: Docker Image**

**A.1 Docker Image là gì?**

[**Docker image**](https://www.geeksforgeeks.org/what-is-docker-image/) được xây dựng bằng [**Dockerfile**](https://www.geeksforgeeks.org/what-is-dockerfile/) bao gồm một tập hợp các hướng dẫn cần thiết để chứa một ứng dụng. **Docker image** bao gồm các mục sau để chạy một phần mềm. **Docker image** là một hình ảnh độc lập với nền tảng có thể được xây dựng trong môi trường Windows và có thể được đẩy lên docker [hub](https://www.geeksforgeeks.org/what-is-docker-hub/) và được kéo bởi những người khác có môi trường hệ điều hành khác nhau như Linux.

**A.2 Các thành phần của Docker Image**

**Layers**: Các lớp hệ thống tập tin không thay đổi được xếp chồng lên nhau để tạo thành một hình ảnh hoàn chỉnh.

**Base Image**: Lớp nền tảng, thường là hệ điều hành tối thiểu hoặc môi trường thời gian chạy.

**Dockerfile** : Tệp văn bản chứa hướng dẫn để xây dựng hình ảnh Docker.

**Image ID**: Mã định danh duy nhất cho mỗi hình ảnh Docker.

**Tags** : Nhãn được sử dụng để quản lý và phiên bản hình ảnh Docker.

**A.2 SubCommands của Docker Image**

| **Yêu cầu** | **Sự miêu tả** |
| --- | --- |
| **docker image build** | Lệnh này được sử dụng để xây dựng một hình ảnh từ Dockerfile |
| **docker image history** | Nó được sử dụng để biết lịch sử của hình ảnh docker |
| **docker image inspect** | Nó được sử dụng để hiển thị thông tin chi tiết trên một hoặc nhiều hình ảnh |
| **docker image prune** | Nó được sử dụng để xóa các hình ảnh không sử dụng không liên quan đến bất kỳ vùng chứa nào |
| **docker image save** | Lệnh này giúp lưu hình ảnh docker vào các tệp được lưu trữ tar |
| **docker image tag** | Nó giúp tạo thẻ cho hình ảnh mục tiêu để tham chiếu đến hình ảnh nguồn. |

**A.3 Docker Image Prune**

Docker image prune là lệnh được sử dụng trong máy chủ Docker để xóa những hình ảnh không được sử dụng hoặc lệnh Docker image prune được sử dụng để xóa những hình ảnh Docker không sử dụng.

**docker image prune**

**A.4 Docker Image Build**

Lệnh được sử dụng để xây dựng hình ảnh docker.

**docker build -t your\_image\_name:tag -f path/to/Dockerfile .**

Docker build: Khởi tạo quá trình xây dựng.

-t your\_image\_name:tag: Đặt tên cho hình ảnh bạn đang tạo và nếu muốn, đặt thẻ .

path/to/Dockerfile . : Cung cấp vị trí của Dockerfile. Cung cấp đường dẫn đúng nếu nó không nằm trong thư mục hiện tại. “(.) DOT” biểu thị wordir hiện tại.

**A.4 Thẻ hình ảnh Docker**

Thẻ Docker là nhãn cho hình ảnh container, được sử dụng để phân biệt các phiên bản và biến thể của hình ảnh trong quá trình phát triển và triển khai. Thẻ Docker sẽ giúp bạn xác định các phiên bản khác nhau của hình ảnh Docker và giúp phân biệt giữa chúng. Hình ảnh Docker sẽ giúp chúng ta xây dựng triển khai liên tục rất nhanh chóng

**A.5 Sau đây là những công dụng của Docker Images:**

* Chúng ta có thể chạy container một cách dễ dàng và hiệu quả với sự hỗ trợ của hình ảnh docker.
* Tất cả mã, cài đặt cấu hình, biến môi trường, thư viện và thời gian chạy đều có trong hình ảnh Docker.
* Hình ảnh Docker không phụ thuộc vào nền tảng.
* Lớp là các khối xây dựng nên hình ảnh.
* Khi sử dụng lệnh build, người dùng có tùy chọn bắt đầu hoàn toàn từ đầu hoặc sử dụng hình ảnh hiện có cho lớp đầu tiên.

**A.6 Sau đây là sự khác biệt giữa Docker Image và Docker Container :**

| **Hình ảnh Docker** | **thùng chứa Docker** |
| --- | --- |
| Hình ảnh Docker là mã nguồn của vùng chứa Docker. | Bộ chứa Docker là phiên bản của hình ảnh Docker. |
| Dockerfile là điều kiện tiên quyết đối với Docker Image. | Docker Image là điều kiện tiên quyết để có Docker Container. |
| Hình ảnh Docker có thể được chia sẻ giữa người dùng với sự trợ giúp của Docker Registry. | Không thể chia sẻ các container Docker giữa những người dùng. |
| Để thực hiện thay đổi trong docker image, chúng ta cần thực hiện thay đổi trong Dockerfile. | Chúng ta có thể tương tác trực tiếp với container và có thể thực hiện những thay đổi cần thiết. |

**A.7 Cấu trúc của Docker Image**

* **Ảnh cơ sở:** Ảnh cơ sở sẽ là điểm khởi đầu cho phần lớn các Dockerfile và có thể được tạo từ đầu.
* **Parent Image:** Parent Image là image mà image của chúng ta dựa trên. Chúng ta có thể tham chiếu đến parent image trong Dockerfile bằng lệnh FROM và mỗi khai báo sau đó sẽ ảnh hưởng đến parent image.
* **Layers:** Docker image có nhiều layer. Để tạo một chuỗi image trung gian, mỗi layer được tạo trên layer trước nó.
* **Docker Registry:** Tham khảo trang này về Docker Registry để biết thêm thông tin.

**A.8 Thực hiện theo các bước dưới đây để tạo Docker Image và chạy Container:**

**Bước 1:**[Tạo Dockerfile.](https://www.geeksforgeeks.org/docker-concept-of-dockerfile/)

**Bước 2:** Chạy lệnh sau trong terminal và nó sẽ tạo một ảnh docker của ứng dụng và tải xuống tất cả các phụ thuộc cần thiết để ứng dụng chạy thành công.

**docker build -t <tên>:<thẻ>**

Thao tác này sẽ bắt đầu xây dựng hình ảnh.

**Bước 3:** Chúng ta đã tạo thành công [Dockerfile](https://www.geeksforgeeks.org/what-is-dockerfile/) và Docker image tương ứng.

**Bước 4:** Chạy lệnh sau trong terminal và nó sẽ tạo một container đang chạy với tất cả các phụ thuộc cần thiết và khởi động ứng dụng.

**docker run -p <port\_run>:80 <tên-hình-ảnh>:<thẻ>**

**port\_run** là cổng mà chúng ta muốn truy cập vào ứng dụng. 80 là cổng mà container đang mở để máy chủ có thể truy cập.

**A.9** **Lệnh Docker Image**

**Liệt kê hình ảnh Docker: docker images**

**Pull an Docker Image From a Registry: docker image pull <image-name>**

Docker Images Filter: **docker image ls -f "reference**= **image-name"**

Remove an Image from Docker : **docker rmi <id-of-image>**

Searching for a specific image on Docker Hub : **docker search image-name**