**Phần 15 Docker – Liên kết Container**

Docker là một tập hợp các sản phẩm nền tảng dưới dạng dịch vụ (PaaS) sử dụng trực quan hóa cấp độ Hệ điều hành để phân phối phần mềm trong các gói được gọi là container. Có những lúc trong quá trình phát triển ứng dụng của chúng ta, chúng ta cần hai container để có thể giao tiếp với nhau. Có thể các dịch vụ của cả hai container phụ thuộc vào nhau. Điều này có thể được thực hiện với sự trợ giúp của Liên kết container.

**Kết nối với Hệ thống liên kết**

Có hai cách để liên kết các container

* Cách mặc định.
* Cách do người dùng xác định.

Để hiểu được sự hình thành của mạng tùy chỉnh giữa hai container, chúng ta cần hiểu cách docker tự động chỉ định mạng.

**A.1 Cách Mặc Định**

Sau khi [cài đặt docker](https://www.geeksforgeeks.org/how-to-install-and-configure-docker-in-ubuntu/) và tạo một container, một mạng bắc cầu mặc định được gán cho docker, theo tên Docker0.

**Lệnh: ifconfig**

**Bước 1:**Tạo các container mới bằng lệnh: docker run.

**Bước 2:** Kiểm tra IP của container mới bằng lệnh: docker network inspect bridge.

**Bước 3:** Vào bên trong 1 container [docker container exec -it ….] và thử ping [ping “dia\_chi\_ip”] đến container khác, nếu bạn nhận được phản hồi thì điều này có nghĩa là kết nối mặc định đã được thiết lập.

**A.2 Cách do người dùng xác định**

**Bước 1:** Tạo [mạng cầu nối tùy chỉnh.](https://www.geeksforgeeks.org/computer-network-tutorials/)

**docker network create <bridge\_name>**

(Điều này sẽ tạo một cầu nối với mạng con và cổng tùy chỉnh)

[Chúng ta cũng có thể cung cấp mạng con](https://www.geeksforgeeks.org/introduction-to-subnetting/) và cổng riêng của mình [.](https://www.geeksforgeeks.org/introduction-of-gateways/)

**docker network create --subnet <mạng\_con\_của\_bạn>**

**--gateway <Cổng\_của\_bạn> bridgename**

**Bước 2:** Kiểm tra xem mạng của bạn đã được tạo hay chưa.

**docker network ls**

**Bước 3:** Liên kết hai container trên mạng mà bạn vừa tạo bằng cách sử dụng cờ “–net”.

**docker run --name <tên\_container> --net=<custom\_net> -d <tên\_hình\_ảnh>**

**Bước 4:** Vào bên trong container (IP1) và ping container (IP2).

**Tầm quan trọng của việc đặt tên**

Docker chủ yếu phụ thuộc vào tên của container mà chúng ta có thể thấy trong ví dụ trên. Bất cứ khi nào bạn tạo một container mới, tên sẽ được tạo tự động. Chúng ta cũng có thể đặt tên cho container của mình theo hai cách khác nhau.

* Bằng cách đặt tên cho container, chúng ta có thể theo dõi loại chương trình được thực thi bên trong container, chẳng hạn như ứng dụng web hoặc cơ sở dữ liệu.
* Ví dụ, nếu một ứng dụng web muốn giao tiếp với máy chủ DB, nó có thể hoạt động như một rào cản giữa hai vùng chứa như một liên kết kết nối.

Chúng ta có thể đặt tên cho container của mình bằng lệnh được hiển thị bên dưới

**(--name) docker run -d -P --name <tên/tên hình ảnh/thẻ>**

**Biến môi trường**

Giả sử nhà phát triển đã đề cập đến một số –env [(biến môi trường)](https://www.geeksforgeeks.org/environment-variables-in-linux-unix/) trong mã nguồn mà chúng ta có thể sử dụng để kết nối với máy chủ cơ sở dữ liệu, ví dụ: Tên người dùng và mật khẩu thì khi tạo vùng chứa, chúng ta sẽ đặt tên người dùng và mật khẩu như được hiển thị trong lệnh bên dưới.

**docker run -d --name <tên> -e TÊN NGƯỜI DÙNG=<\*\*\*> -e MẬT KHẨU=<\*\*\*> --network <\*\*\*>**

Chúng ta có thể thiết lập các biến env được đề cập ở trên vào vùng chứa cơ sở dữ liệu bằng cách sử dụng lệnh sau.

**docker run -d -p <cổng> --tên <tên> -e TÊN MÁY CHỦ=<\*\*\*> -e TÊN NGƯỜI DÙNG=<\*\*\*> -e MẬT KHẨU=<\*\*\*> --mạng <\*\*\*>**

**Cập nhật tệp /etc/hosts**

Docker thêm mục lưu trữ cho vùng chứa nguồn vào /etc/hosts ngoài các biến môi trường mà chúng tôi cung cấp lệnh để liên kết hai vùng chứa được đề cập bên dưới.

**docker run -t -i --rm --link <Đề cập mục nhập>**

Để kiểm tra danh sách các mục đã được đề cập trong tệp /etc/hosts, chúng ta có thể sử dụng lệnh bên dưới.

**cat /etc/hosts**