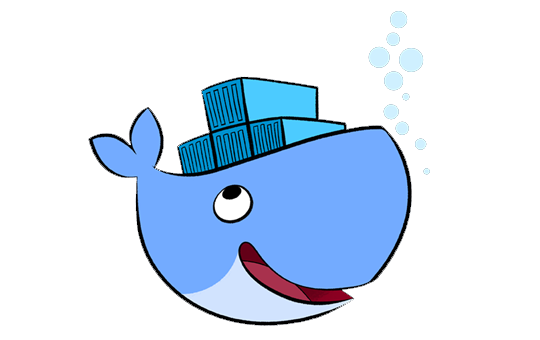
Docker là gì? Cơ bản về Docker

Trong thời gian gần đây có rất nhiều bài báo, hay blog, và cả trên Twitter có rất nhiều người đề cập đến Docker. Vậy Docker là gì và tại sao nó hữu ích.

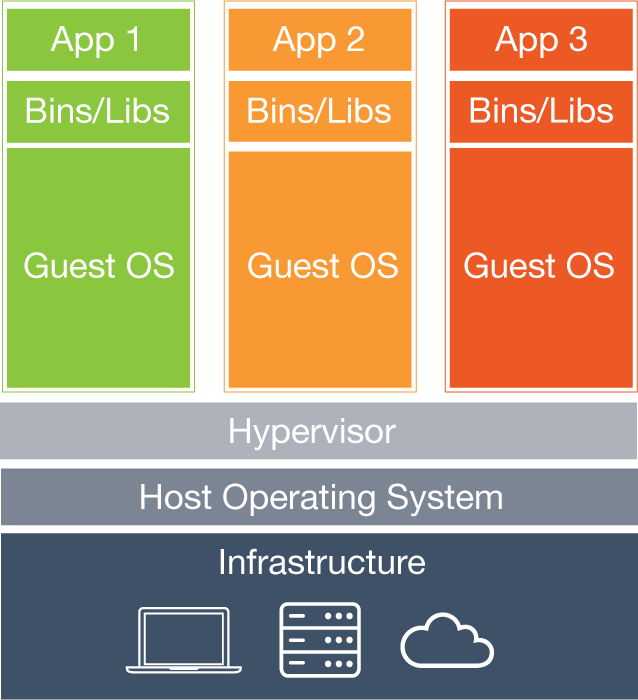
[](https://blog.duyetdev.com/2015/12/docker-la-gi-co-ban-ve-docker.html)

[#](http://blog.duyet.net/2015/12/docker-la-gi-co-ban-ve-docker.html" \l "Dockerlg) Docker là gì?

Docker - đây là một công cụ tạo môi trường được "đóng gói" (còn gọi là Container) trên máy tính mà không làm tác động tới môi trường hiện tại của máy, môi trường trong Docker sẽ chạy độc lập.

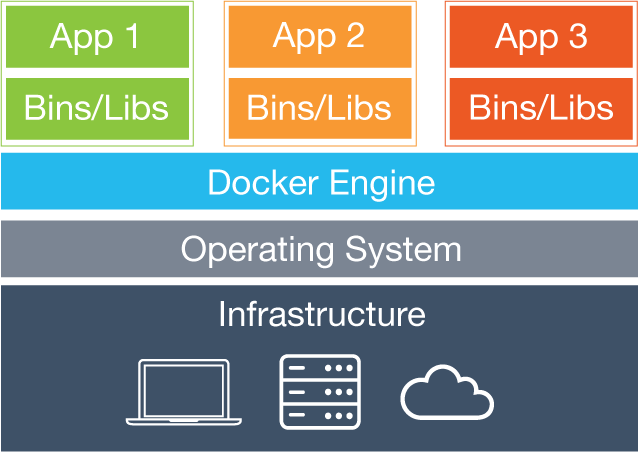
Một số developer thường tạo sẵn các môi trường này, và upload lên mạng để mọi người lấy về dùng, và mấy cái này gọi là các Images.

[#](http://blog.duyet.net/2015/12/docker-la-gi-co-ban-ve-docker.html" \l "Dockervmyo) Docker và máy ảo?

Docker là công cụ tạo môi trường đóng gói, nó còn đóng gói cả hệ điều hành trong đó, vậy Docker khác máy ảo chỗ nào?  


### Máy ảo

Mỗi máy ảo chứa ứng dụng, kèm theo các thư viện và hệ điều hành, mỗi ứng dụng như vậy chiếm hàng chục GBs.



### Docker

Mỗi containers chứa ứng dụng gồm chứa các thư viện riêng, nhưng kernel được chia sẻ với các containers khác. Mỗi kernel được chạy trong các môi trường độc lập với nhau. Docker infrastructure chạy được trên mọi máy tính, đám mây hay nền tảng nào. Vì vậy bạn có thể vận chuyển ứng dụng đến bất cứ đâu và không cần phải quan tâm đến môi trường phát triển, thiếu thư viện, ...

Tóm lại, docker:

* Docker rất tốt tại việc xây dựng và chia sẻ Disk Image qua hệ thống Docker Index
* Docker là một phần mềm quản lý cơ sở hạ tầng.
* Docker làm việc tuyệt vời với các công cụ quản lý file config (vd: Chef, Puppet)
* Docker sử dụng btrfs để giảm sát các file hệ thống và có thể được chia sẻ với user khác. (Như cách hoạt động của Git)
* Docker có một bộ kho trung tâm của các Disk Images (có thể được public hoặc private), điều này cho phép bạn dễ dàng chạy trên nhiều hệ điều hành khác nhau (Ubuntu, Centos, Fedora, Gentoo).

[#](http://blog.duyet.net/2015/12/docker-la-gi-co-ban-ve-docker.html" \l "KhinothsdngDocker) Khi nào thì sử dụng Docker

* Docker là một công cụ đơn giản, như Git hay Java, mà cho phép bạn kết hợp chặt chẽ tới công việc phát triển hay điều hành hàng ngày của bạn.
* Sử dụng Docker như là một phần mềm quản lý phiên bản (version control system) cho toàn hệ điều hành của bạn.
* Sử dụng Docker khi bạn muốn đóng góp hay hợp tác hệ điều hành của bạn với một nhóm nào đó.
* Sử dụng Docker để chạy những dòng code trên laptop của bạn trong môi trường giống hệt như trên server của bạn.
* Sử dụng Docker khi app của bạn cần trải qua nhiều giai đoạn khác nhau của quá trình phát triển.

Hãy thử dùng và trải nghiệm docker. Mình sẽ liệt kê một số lệnh hay dùng trong Docker

[#](http://blog.duyet.net/2015/12/docker-la-gi-co-ban-ve-docker.html" \l "Cclnhcbn) Các lệnh cơ bản

Pull một image từ Docker Hub

docker pull <image name> 

Tạo một container từ image có sẵn

docker run -v <thư mục trên máy tính>:<thư mục trong container> -it <image name> /bin/bash

Lệnh trên tạo container, liên kết một thư mục trên máy tính vào bên trong container, và mở bash trong máy đó.  
  
Khi cần phải map cổng đó từ container ra máy tính ngoài, khi đó chúng ta dùng thêm tham số -p như sau:

docker run -v /abc:/abc -p 8080:8080 -it ubuntu /bin/bash

Lệnh trên map cổng **8080** của container ra cổng **8080** của máy tính hiện tại.

Liệt kê các images hiện có

docker images

Trong kết quả trả về của lệnh này, chúng ta lưu ý các thông số: 

* TAG: là tên của image, ví dụ duyetdev/docker-spark
* IMAGE ID: là ID của image lưu trong hệ thống, ví dụ 91e54dfb1179

Liệt kê các container đang chạy

docker ps

docker ps -a # liệt kê các container đã tắt

* **CONTAINER ID**: Là ID của container đó, ví dụ 4cc671941ee3
* **NAME**: Là tên riêng của container, được tạo ra một cách ngẫu nhiên hoặc có thể tự đặt, ví dụ stupefied\_blackwell

Khởi động và truy cập lại vào một container đã tắt

Nếu một container đã tắt (không xuất hiện khi dùng lệnh docker ps nữa, chúng ta có thể chạy lệnh docker ps -a để lấy ID hoặc NAME của nó, sau đó dùng lệnh sau để khởi động và truy cập lại vào đó)

docker start <ID hoặc NAME>

docker **exec** -it <ID hoặc NAME> /bin/bash

Xoá một container

Nếu một container đã hết giá trị lợi dụng, dù nó đã tắt nhưng nó vẫn chiếm một phần dung lượng trên máy tính, để xoá nó đi, chúng ta dùng lệnh

docker rm <ID hoặc NAME>

Nếu container đang chạy, bạn cũng có thể xoá nhưng phải thêm tham số **-f** vào sau rm để force remove:

docker rm -f <ID hoặc NAME>

Xoá một image

Cũng như container, nếu bạn đã ko còn nhu cầu sử dụng một image nào đó nữa, thì nên xoá nó đi. Dùng lệnh rmi

docker rmi <ID hoặc NAME>

# or

docker rmi -f <ID hoặc NAME>

[#](http://blog.duyet.net/2015/12/docker-la-gi-co-ban-ve-docker.html" \l "Kt) Kết

Sự thật là Docker đang dần thay đổi nhiều lập trình viên và đặc biệt là các admin cách làm việc của họ.

Và cộng đồng Docker đang phát triển rất mạnh. Vậy hãy dành chút thời gian và thử dùng Docker và bạn sẽ cảm thấy sự tuyệt vời của Docker qua chính trải nghiệm của mình..

Có thời gian mình sẽ chia sẻ một số Image docker hay sử dụng.  
Đây là một bài viết về cách deploy Apache Spark, deploy số máy vài chục hoặc vài trăm node chỉ với vài bước đơn giản:[**http://blog.duyetdev.com/2015/12/apache-spark-on-docker.html#.VnmHCBV97IU**](http://liink.pw/uyWS)  
  
Tham khảo:

* Docker - [http://docker.io](http://liink.pw/oHH0G3X)
* Docker Solutions and Use Cases - <http://liink.pw/K2ZcK6>