

Giới thiệu về docker

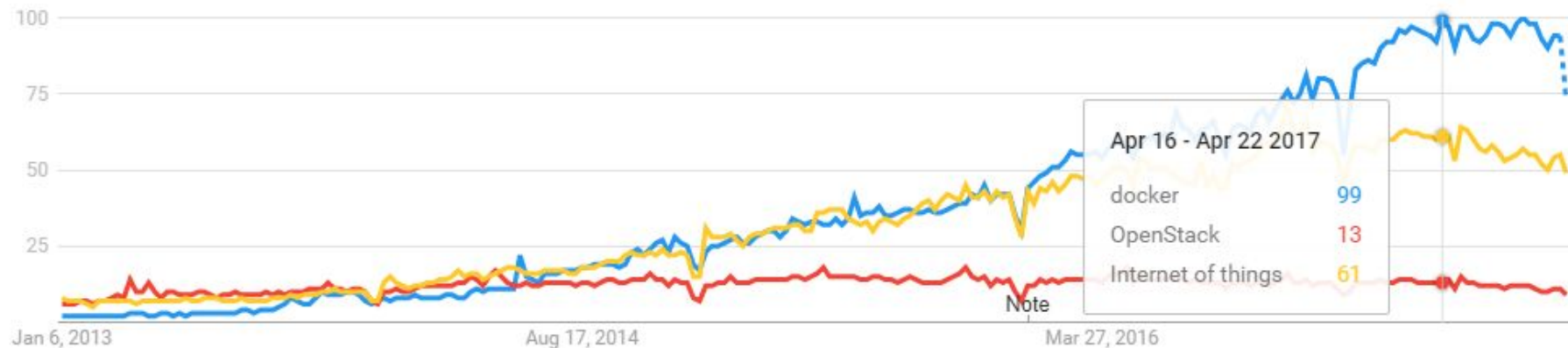
(Docker nằm trong hệ sinh thái về Container)

Trình bày: Team Cloud - TTSI - VNPT DATA

1. Container sinh ra để làm gì
2. Hệ sinh thái của container
3. Giới thiệu về docker
4. Thành phần / mô hình / kiến trúc của docker
5. Thực hành / trao đổi / thảo luận

1. Container sinh ra để làm gì (1)

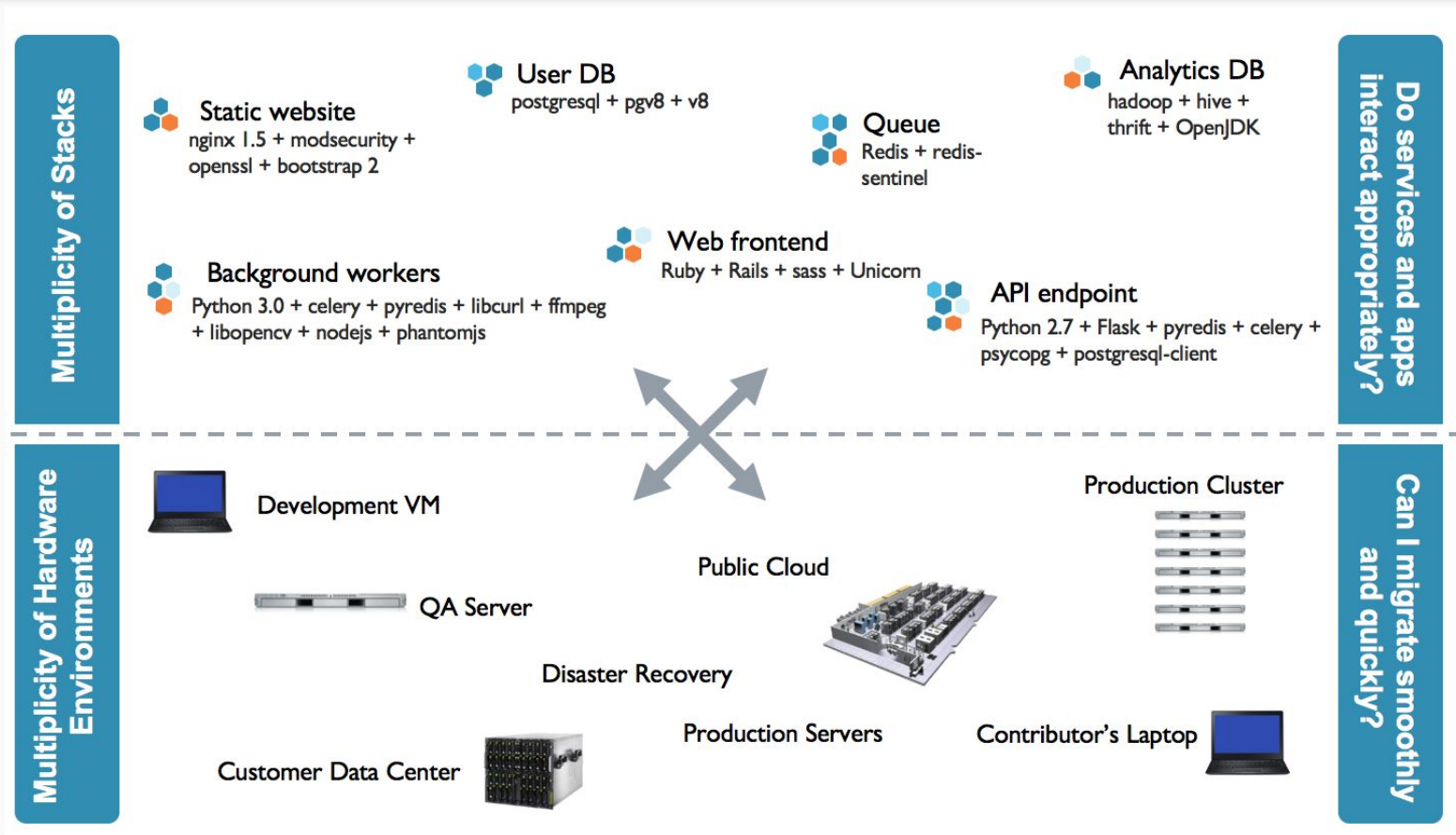
Xu hướng của container









Nếu một công nghệ đang có xu hướng gia tăng thì cần nắm bắt càng sớm càng tốt, vì:

1. Học về một công nghệ mới sẽ tốn về thời gian và công sức.
2. Công nghệ mới thì đồng nghĩa với nhu cầu mới và thị trường mới.








1. Container sinh ra để làm gì (2)



1. Container sinh ra để làm gì (3)

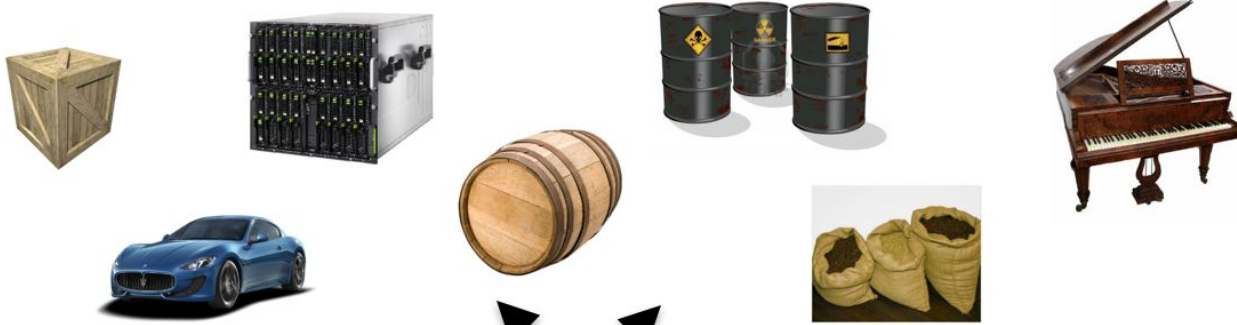


Static website	?	?	?	?	?	?	?
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?
	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers



1. Container sinh ra để làm gì (3)

Multiplicity of
Goods










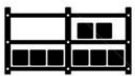





Do I worry about how
goods interact (e.g.
coffee beans next to
spices)

Multiplicity of
methods for
transporting/storin
g

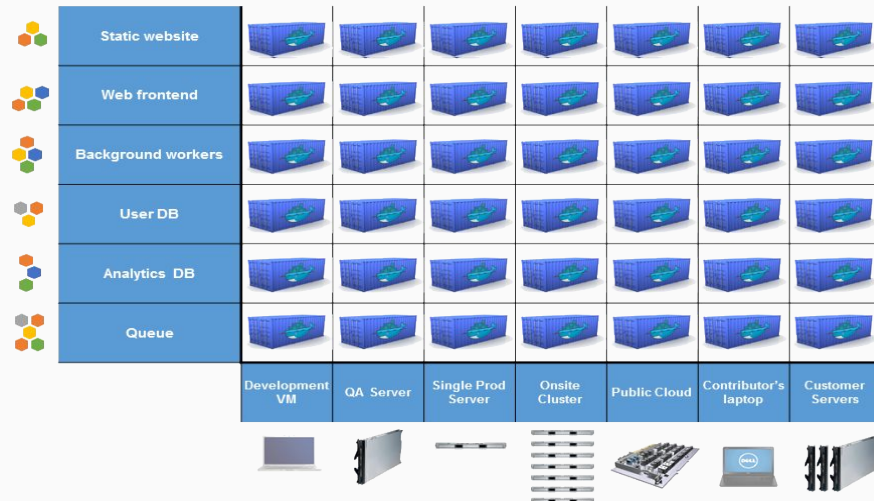
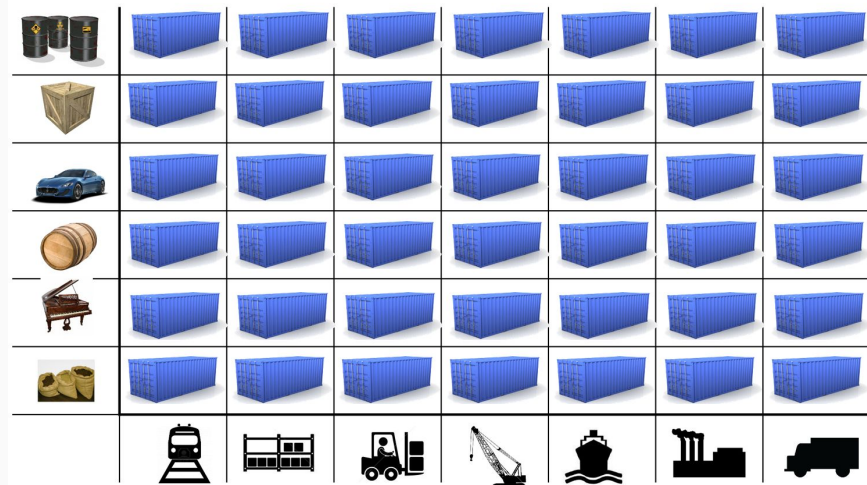


Can I transport quickly
and smoothly
(e.g. from boat to train
to truck)

1. Container sinh ra để làm gì (4)

	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
							

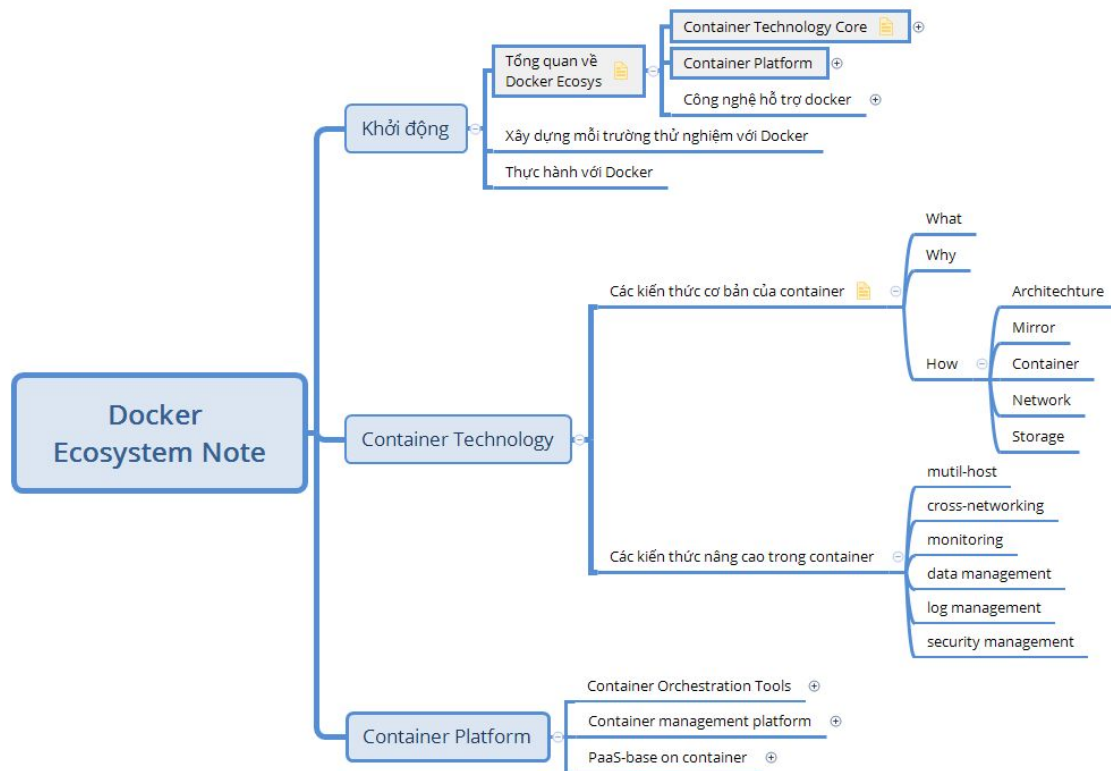
1. Container sinh ra để làm gì (5)



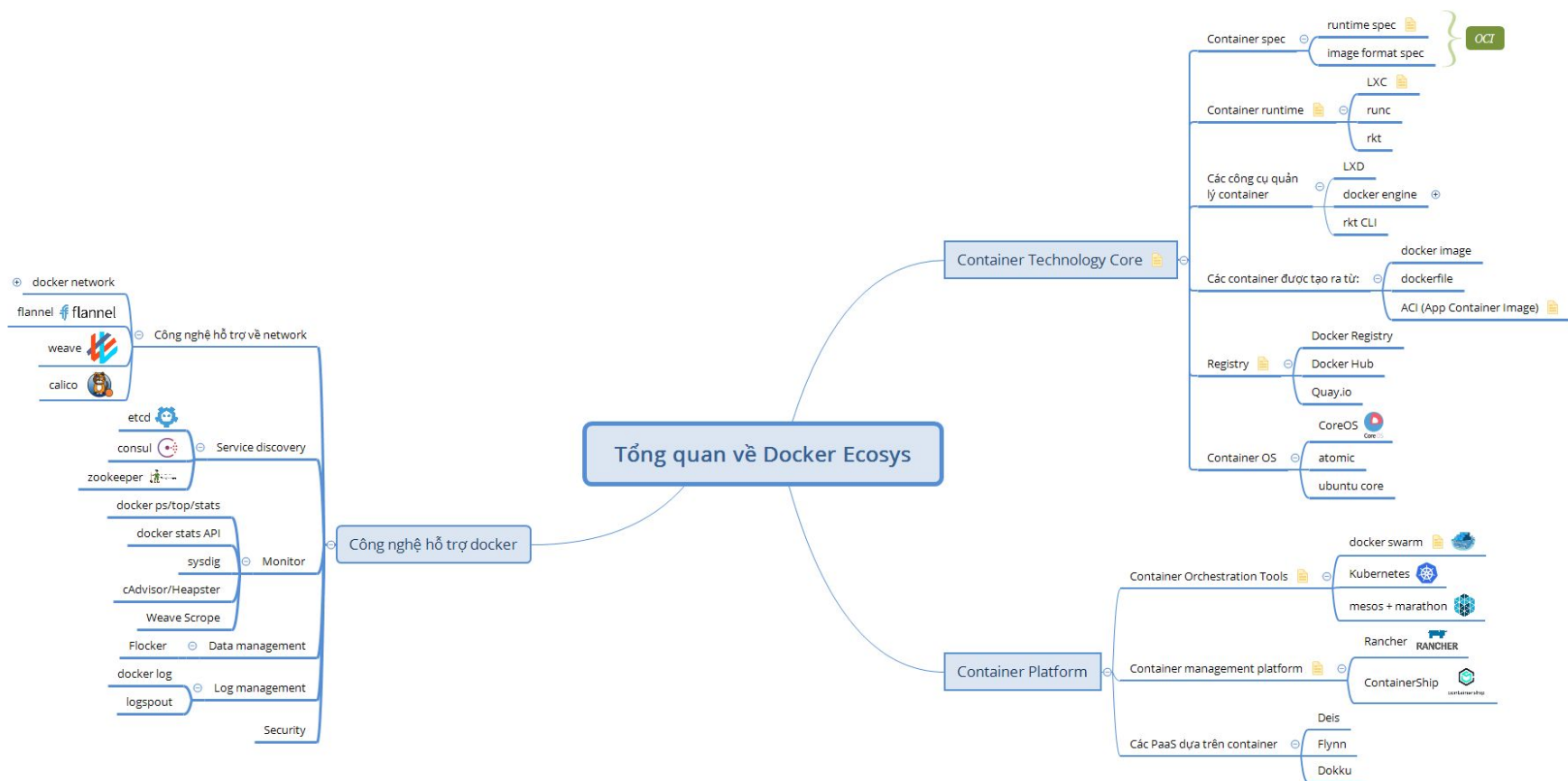
Đối với Developer: Build Once, Run Anywhere

Đối với SysAd: Configure Once, Run Anything

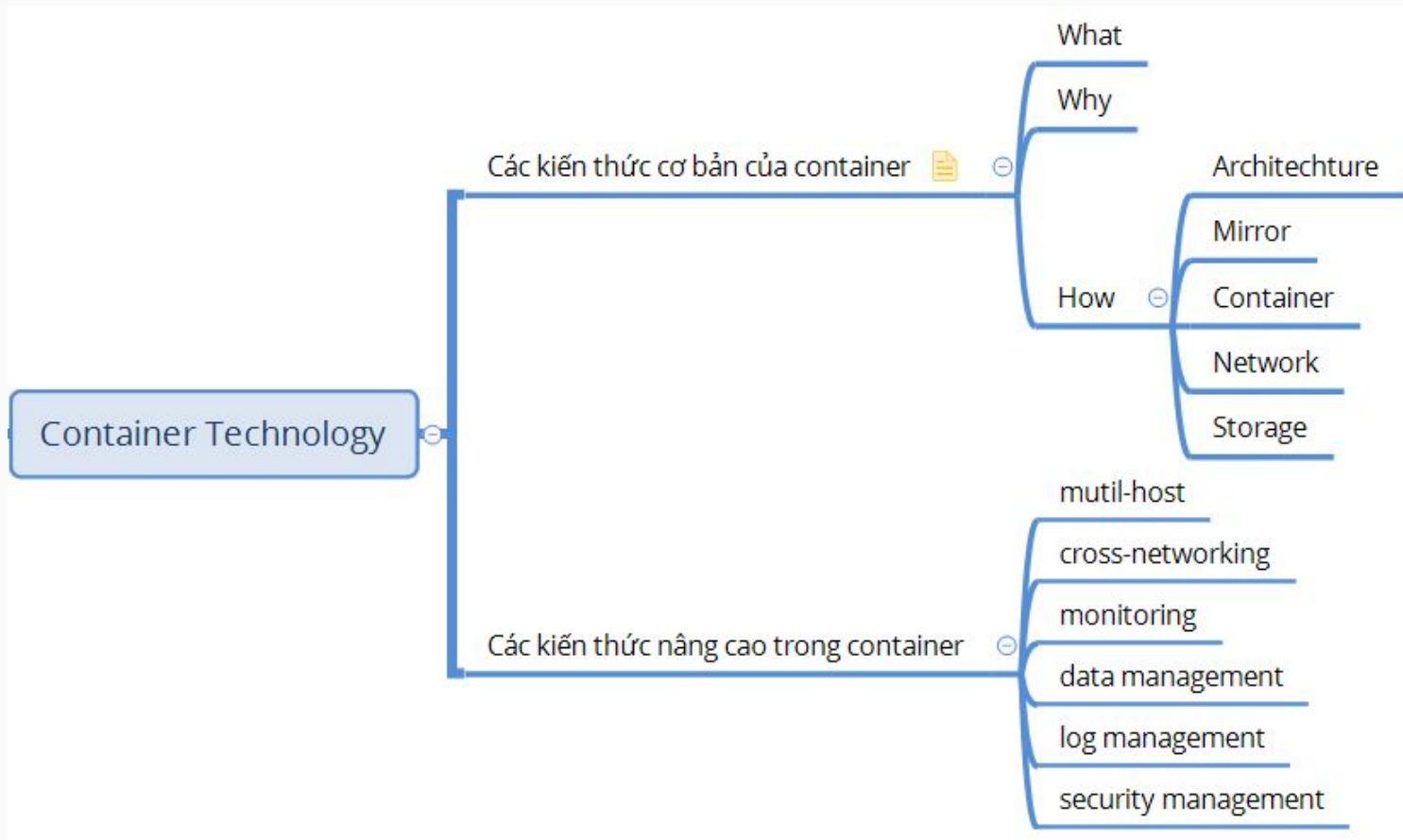
1. Hệ sinh thái của container (2)



1. Hệ sinh thái của container (2)



1. Hệ sinh thái của container (3)



3. Giới thiệu về docker

02 định nghĩa về docker:

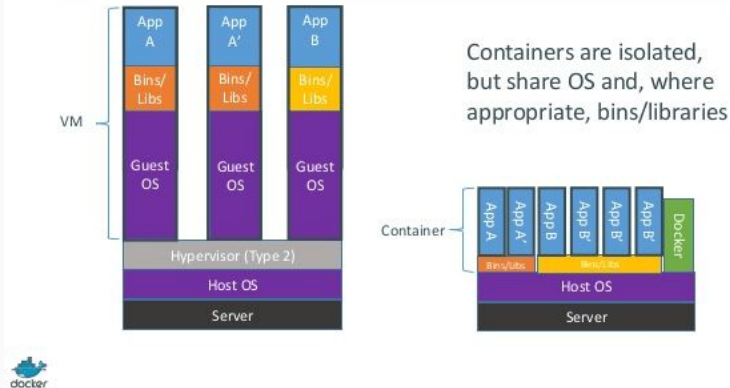
Docker là một nền tảng mở cho phép phát triển, vận chuyển và chạy các ứng dụng dựa vào công nghệ ảo hóa container.

(Theo [Docker](#))

Docker là một dự án mã nguồn mở cho phép tự động hóa việc triển khai các ứng dụng bên trong các phần mềm container bằng cách cung cấp thêm một lớp trừu tượng và tự động hóa việc “ảo hóa cấp độ hệ điều hành” trên Linux, Mac OS và Windows.

(Theo [wikipedia](#))

Containers vs. VMs



3. Giới thiệu về docker(2)

Lịch sử hình thành:

- Được tạo ra bởi Solomon Hykes (@solomonstre)
- Trong một dự án nội bộ của dotCloud (bây giờ là Docker Inc)
- Phát hành lần đầu tiên vào tháng 3-2013
- Theo giấy phép Apache 2.0
- Được đánh giá 30k sao trên Github
- 260k public repositories trên hub.docker.com
- Docker Inc đã mua lại (sản phẩm, nhân sự)
- Docker tham gia “Open Container Initiative” và tháng 6-2015

3. Giới thiệu về docker(3)

Lợi ích của Docker:

- Nhanh (triển khai, di chuyển, khởi động)
 - Bảo mật
 - Lightweight (tiết kiệm disk & CPU)
 - Mã nguồn mở
 - Portable software
 - Microservices và integrations (APIs)
 - DevOps đơn giản
 - Có khả năng điều khiển phiên bản
- Demo cài đặt wordpress/drupal
 - Demo cài đặt zabbix
 - Demo cài đặt rabbitmq



3. Giới thiệu về docker(4)

Ngữ cảnh sử dụng Docker:

- Đóng gói môi trường
- Liên tục tích hợp và triển khai
- Mở rộng các ứng dụng
- Cộng tác phát triển
- Infrastructure configuration
- Phát triển cục bộ
- Các ứng dụng multi-tier
- PaaS, SaaS

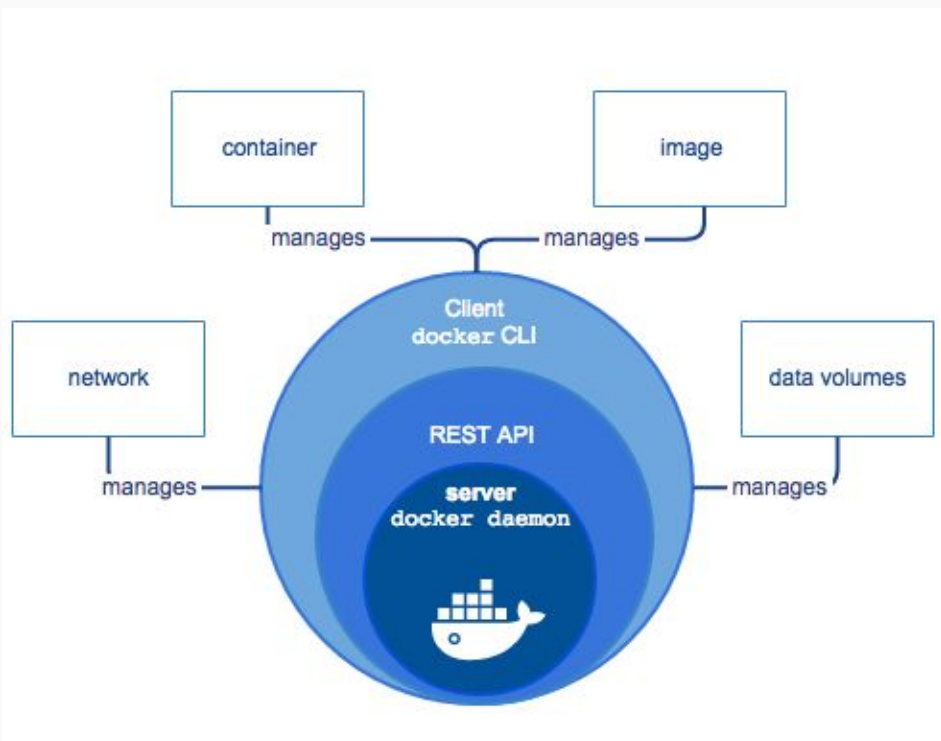
Công nghệ sử dụng:

- Linux x86-64
- Ngôn ngữ Go
- Kiến trúc Client - Server (daemon)
- File file (AUFS, btrfs, vfs,...)
- Namespace (pid, net, ipc, mnt, uts)
- Control Groups (cgroups)
- Container format (libcontainer)

Kết quả [khảo sát 2016](#)

Tham khảo thêm [Understanding docker](#)

4. Các thành phần/kiến trúc trong Docker (1)



CÁC BÀI LAB

- Cài đặt Docker
- Các thao tác với images
- Các thao tác với container
- Nat port cho container
- Xem thông tin container
- Gắn volume vào container
- Tạo một Dockerfile
- Khởi tạo image từ Dockerfile
- Thực hành với network
- Thực hành docker hub

Cài đặt docker engine

Cài đặt docker engine trên CentOS 7.x 64 bit, Ubuntu 14.04 64bit, Ubuntu 16.04



LAB1

```
curl -sSL https://get.docker.com/ | sudo sh  
sudo usermod -aG docker `whoami`
```

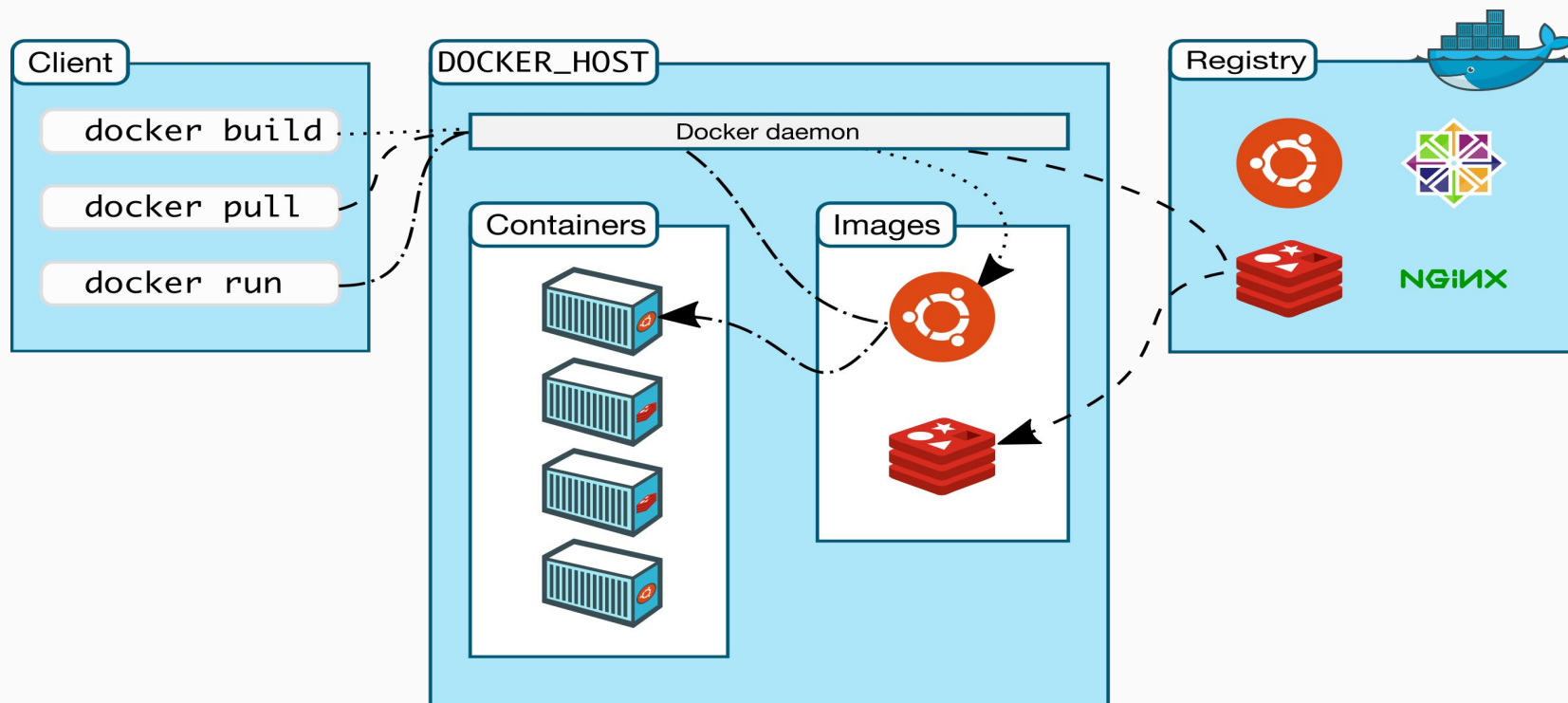
```
systemctl start docker.service  
systemctl enable docker.service
```

```
docker version
```

Tham khảo: <https://github.com/congto/ghichep-docker/blob/master/docs/docker-thuchanh-caidat.md>

4. Các thành phần/kiến trúc trong Docker (2)

Các thành phần chính:



4. Các thành phần/kiến trúc trong Docker (3)

Docker client:

- Giao diện cơ bản cho người dùng sử dụng Docker. Nó nhận lệnh từ người dùng và truyền thông qua lại với Docker daemon
- Client và daemon có thể chạy trên cùng hoặc khác host

Docker daemon:

- Chạy trên một host machine
- Người dùng không tương tác trực tiếp với daemon, mà thông qua Docker client với RESTful api hoặc sockets.

LAB2: Thực hiện tải image có tên là hello-word và tạo container đầu tiên

```
docker pull hello-world
```

```
docker run hello-world
```

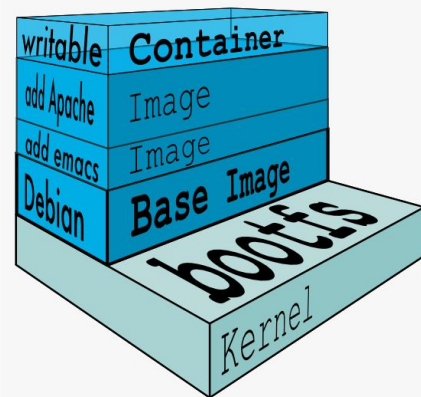
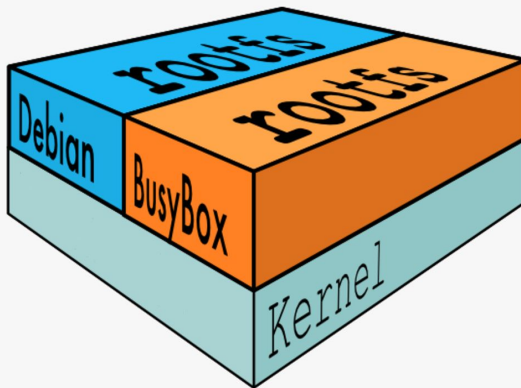
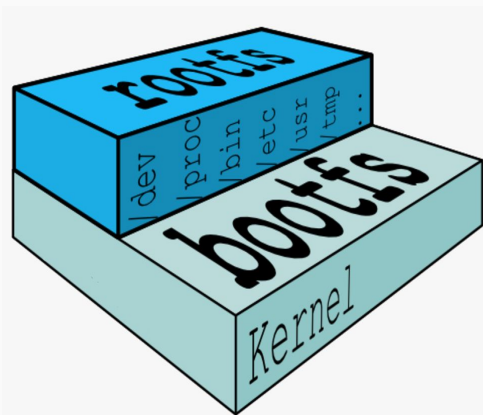


LAB2

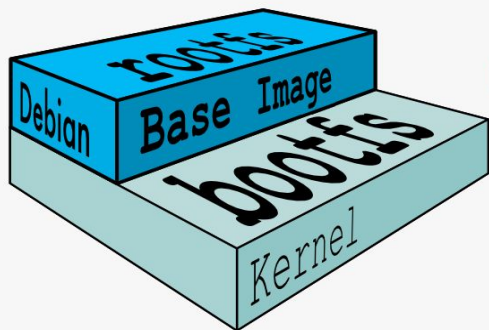
4. Các thành phần/kiến trúc trong Docker (3)

Docker image:

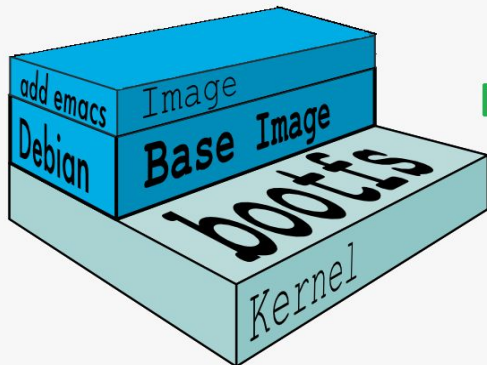
- Docker image hay còn gọi là docker mirror
- Là read-only template dùng để tạo ra các container
- Có một vài cách để tạo ra các bản mirror
 - Tạo ra một mirror từ đầu
 - Tạo ra một mirror bằng cách download của người khác
 - Tạo ra từ một mirror đã tồn tại trước đó.



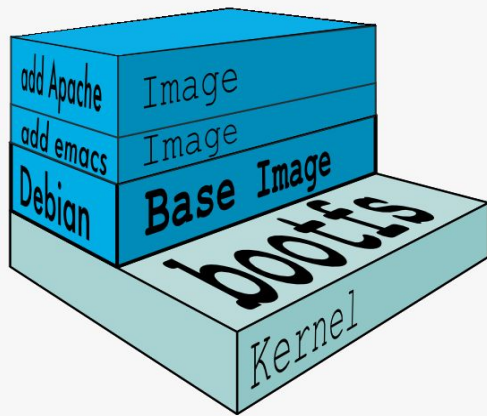
FROM debian ①



RUN apt-get install emacs ②



RUN apt-get install apache2 ③



LAB3: Tìm images, tải image, liệt kê, xóa images

```
docker search ubuntu
```

#Tìm image có tên là ubuntu

```
docker pull httpd
```

#tải image có tên là httpd

```
docker images
```

#Liệt kê các images đã tải

```
docker rmi ubuntu
```

#Xóa image có tên là ubuntu



LAB3

4. Các thành phần/kiến trúc trong Docker (4)

Docker container:

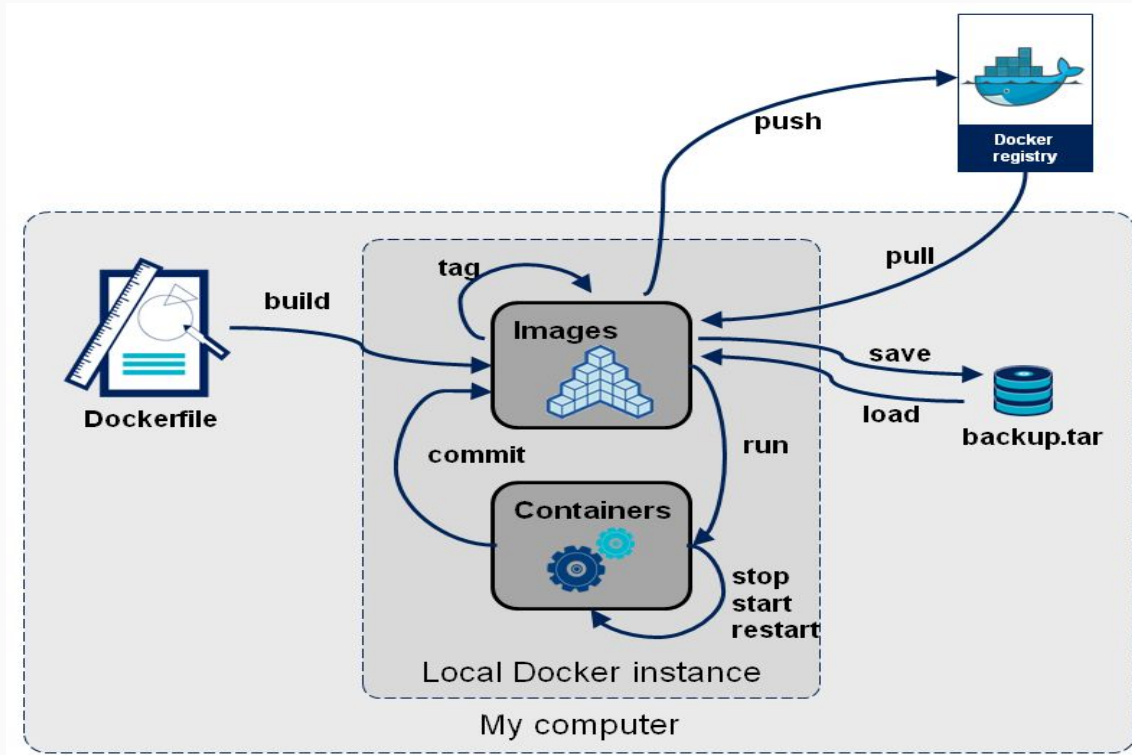
- Docker container chính là một thể hiện của docker image
- Chúng ta có thể có các thao tác cơ bản: start, stop, move, delete với container thông qua CLI hoặc API

Docker registry:

- Registry là nơi lưu trữ các image
- Registry có thể là private hoặc public
- Docker Hub Repositories bên trong Registry

4. Các thành phần/kiến trúc trong Docker (5)

Workflow cơ bản:



4. Các thành phần/kiến trúc trong Docker (6)

Các thành phần liên quan khác:

- Dockerfile
- Docker volume
- Docker network
- Docker Hub (một lựa chọn trong thành phần registry)

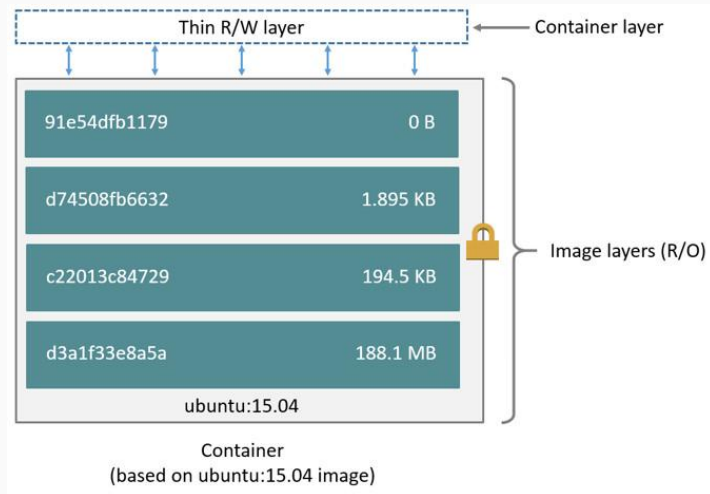
4. Các thành phần/kiến trúc trong Docker (7)

Dockerfile: LAB4

- Là một tài liệu chứa tất cả các lệnh của người dùng để tạo ra một image
- Dockerfile có thể sử dụng comment để giải thích

Docker volume: LAB5

- Sử dụng để chia sẻ dữ liệu giữa host - container, container - container
- Có 2 kiểu docker volume: *bind mount* và *data managed volume*

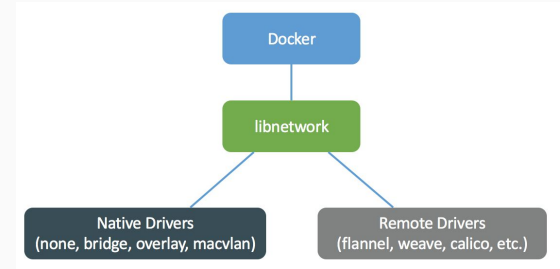
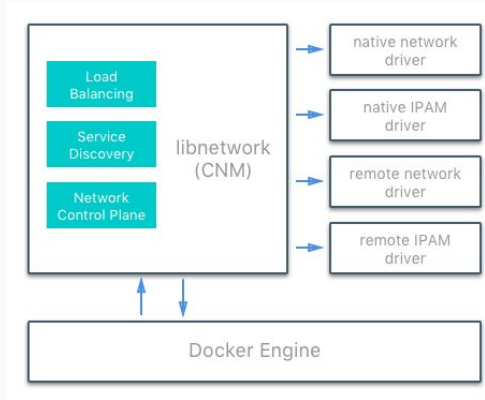
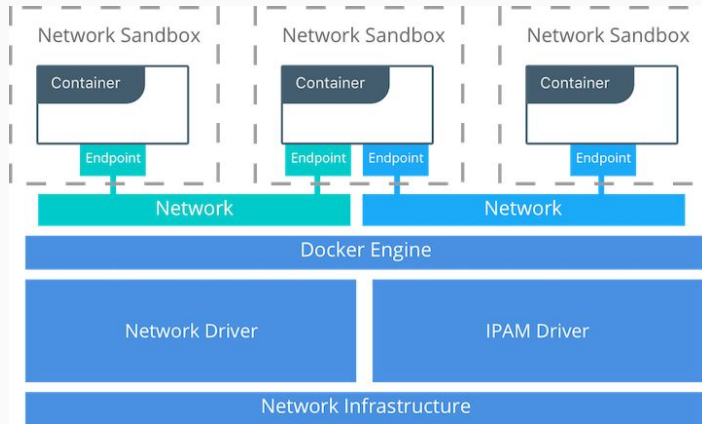


4. Các thành phần/kiến trúc trong Docker (8)

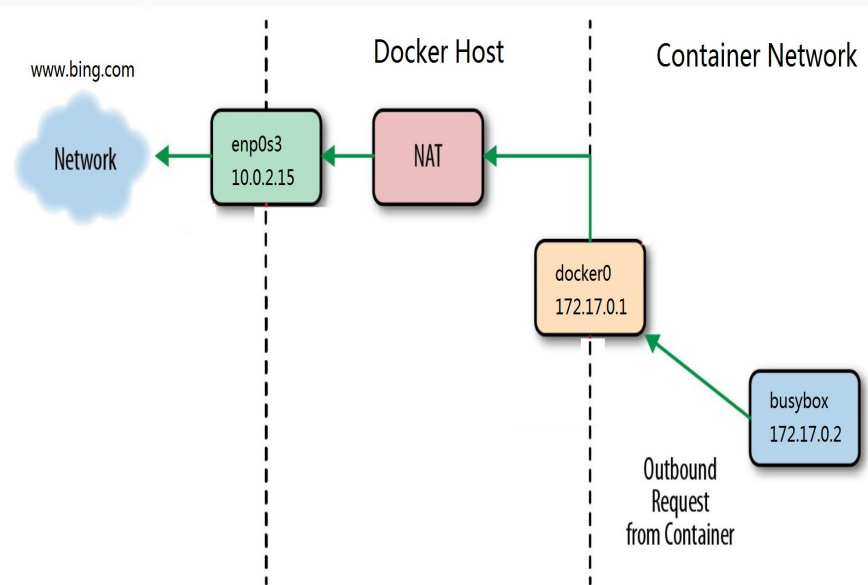
Docker network:



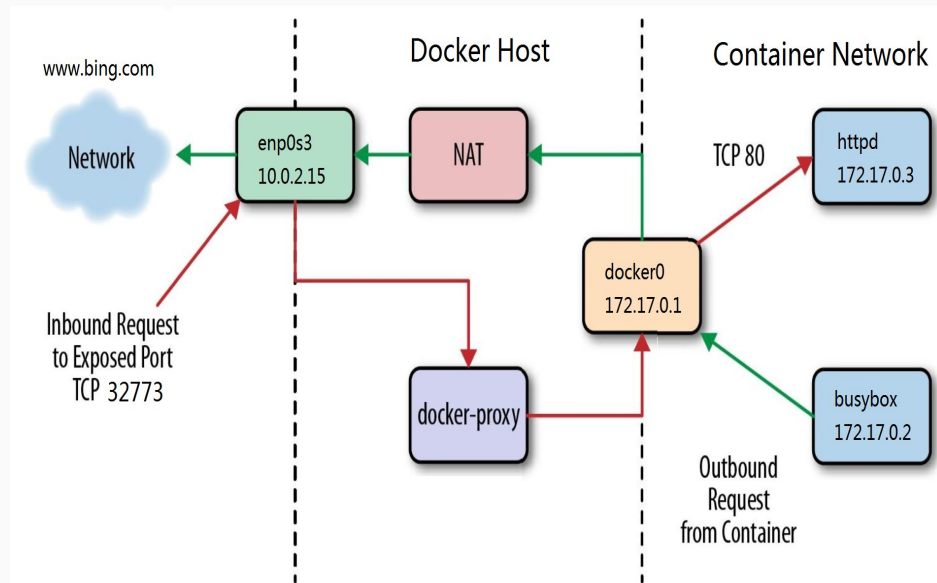
LAB6



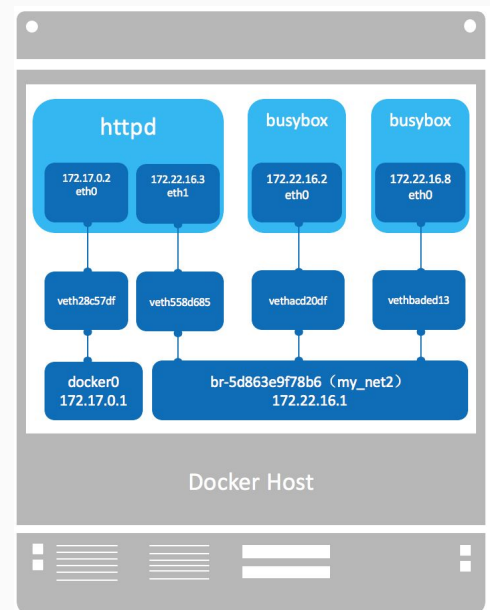
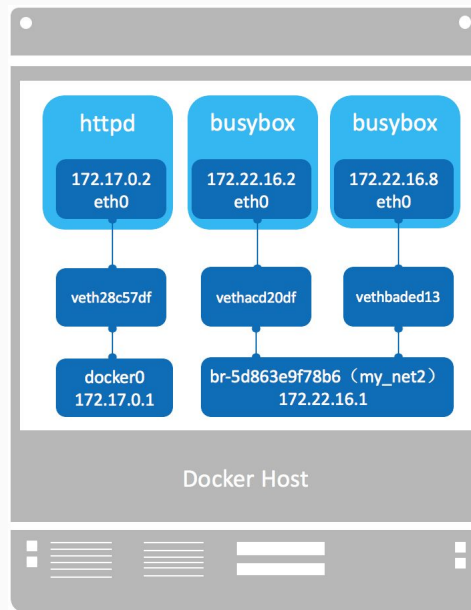
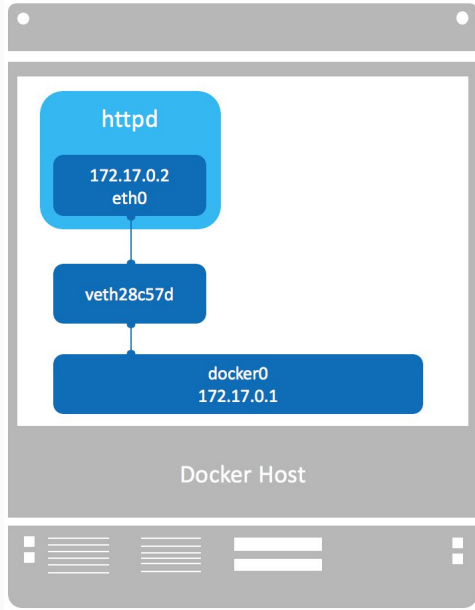
Outbound Request



Inbound Request



Minh họa về kết nối của container



Tham khảo

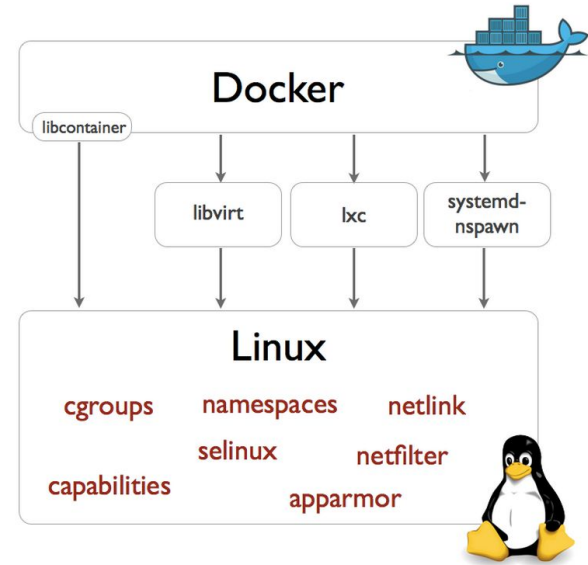
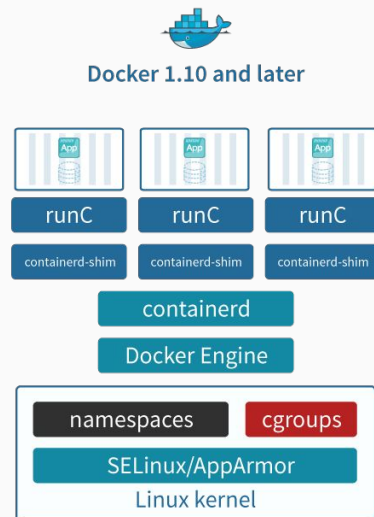
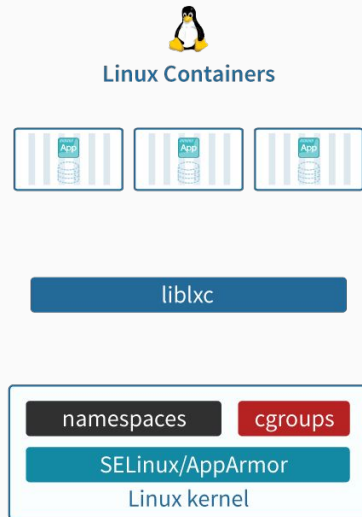
1. <http://www.cnblogs.com/CloudMan6/tag/Docker/>
2. <https://github.com/hocchudong/ghichep-docker>
3. <https://trends.google.com/trends/explore?date=2013-01-01%202017-09-07&q=docker,OpenStack,%2Fm%2F02vnd10>
4. <https://www.docker.com/survey-2016>
5. <https://docs.docker.com/engine/docker-overview/#docker-engine>

Cám ơn & Thảo luận

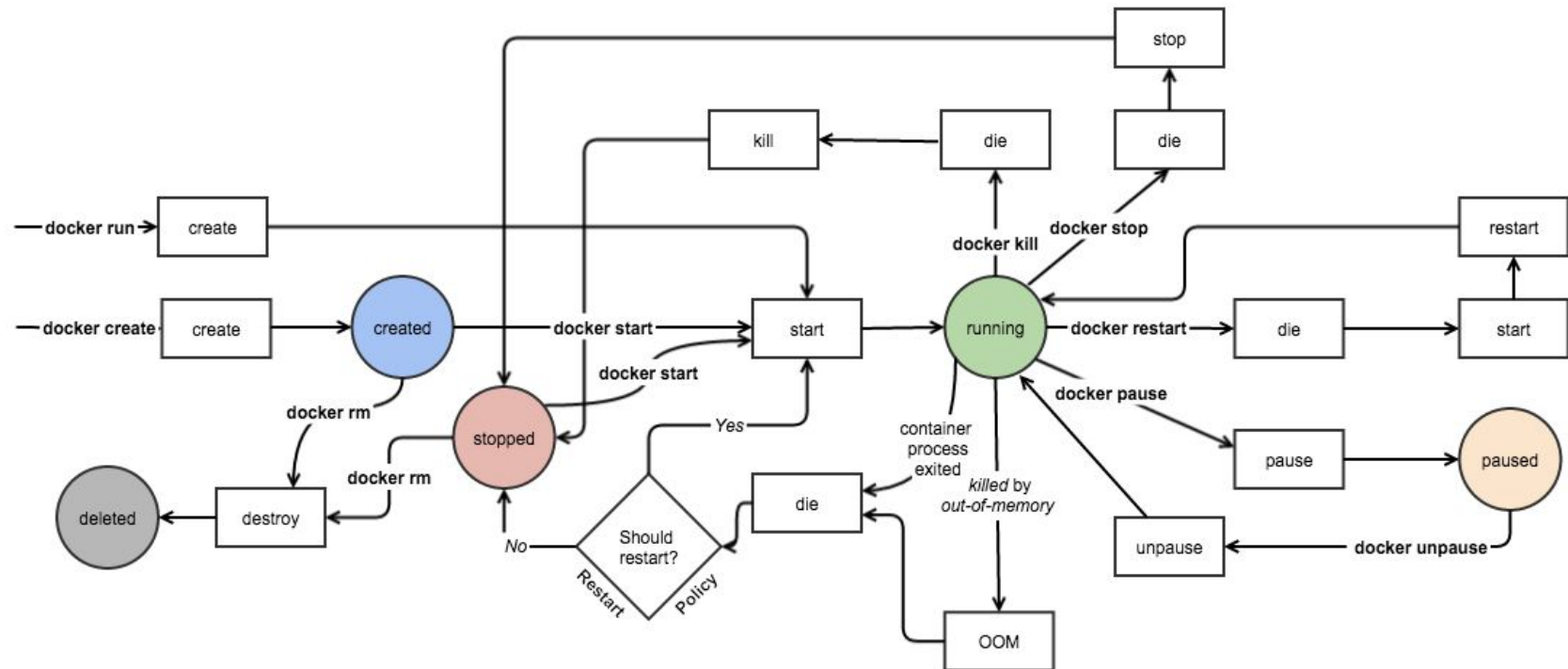
BONOUS

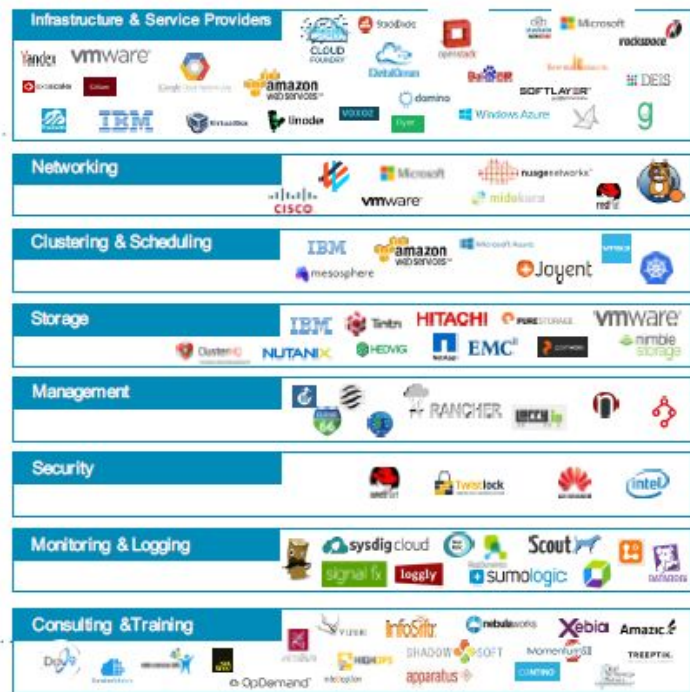
Các slide bổ trợ khác

So sánh kiến trúc low level của Docker với LXC



Các state của container

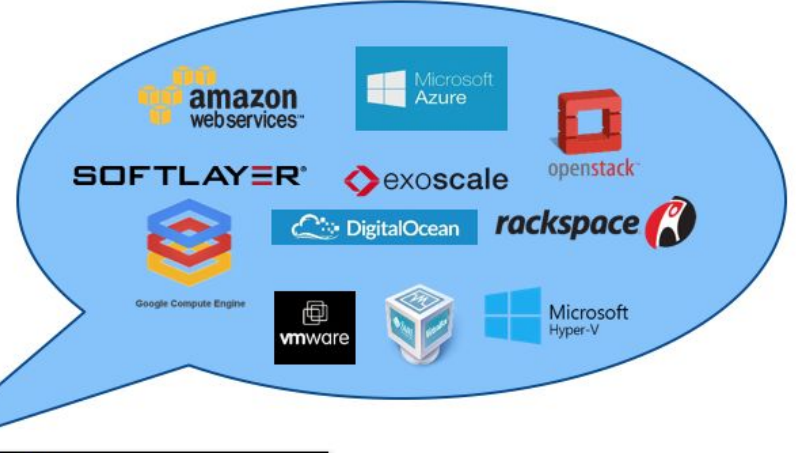


[illegible]

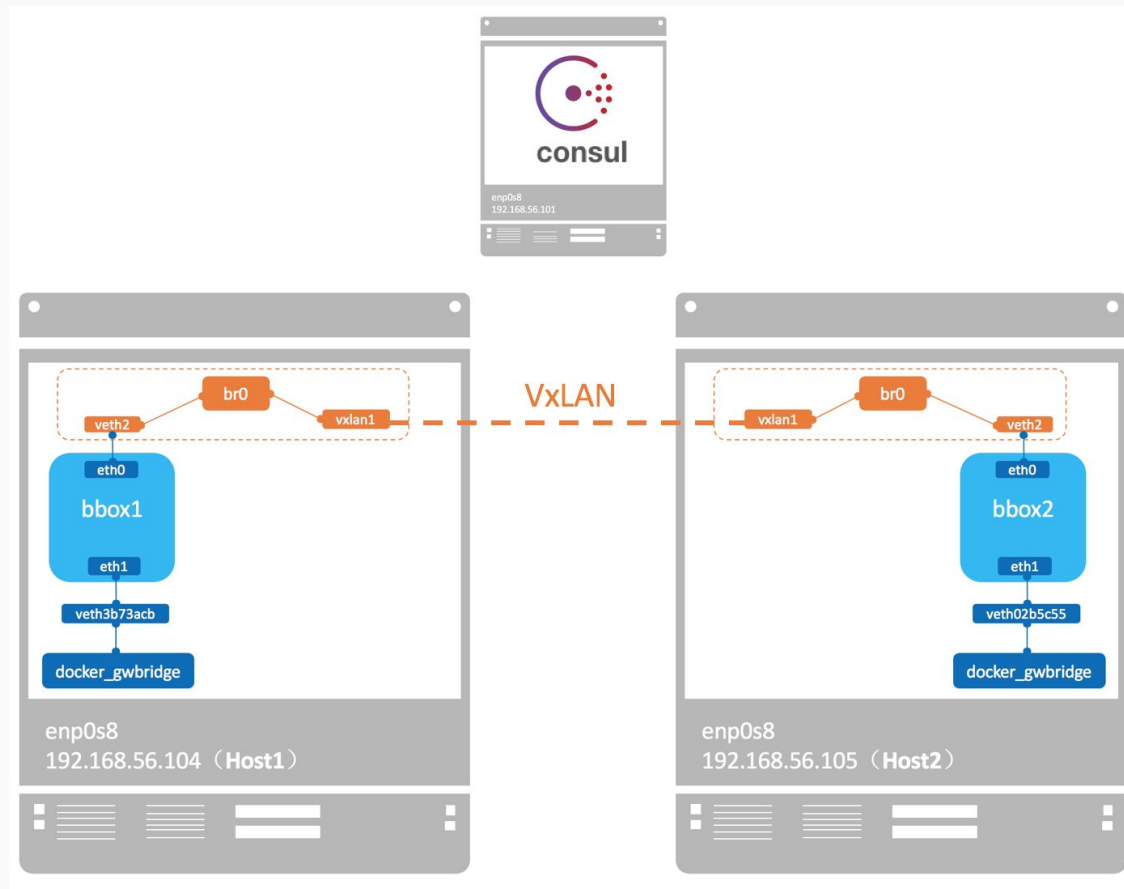
Docker machine



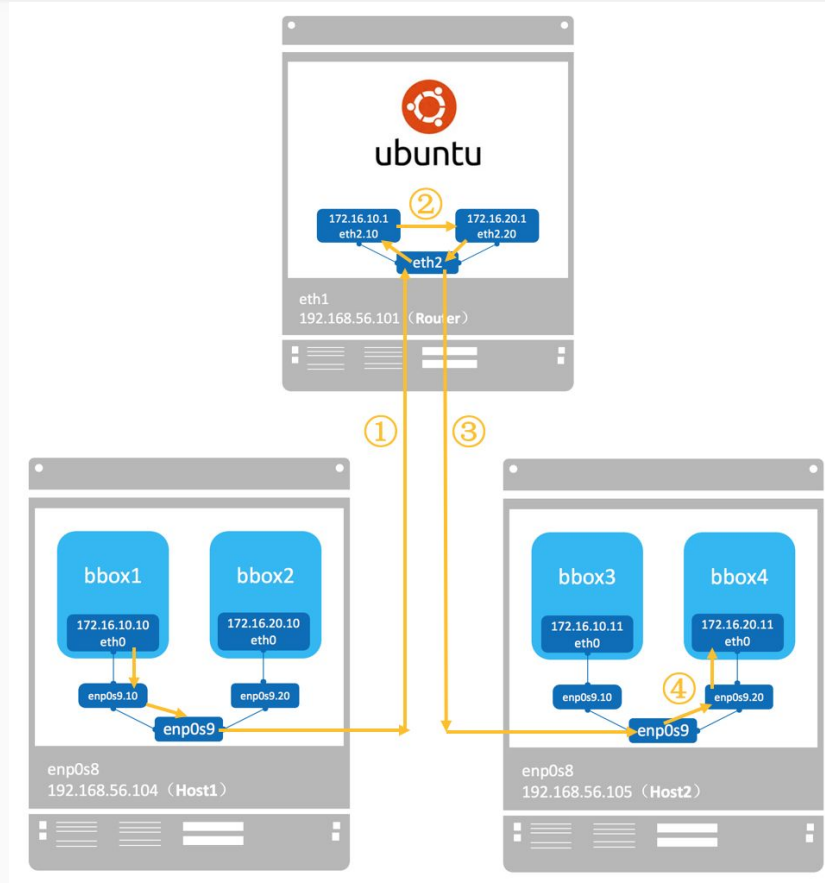
```
$ docker-machine create --driver [Provider] hostx
```



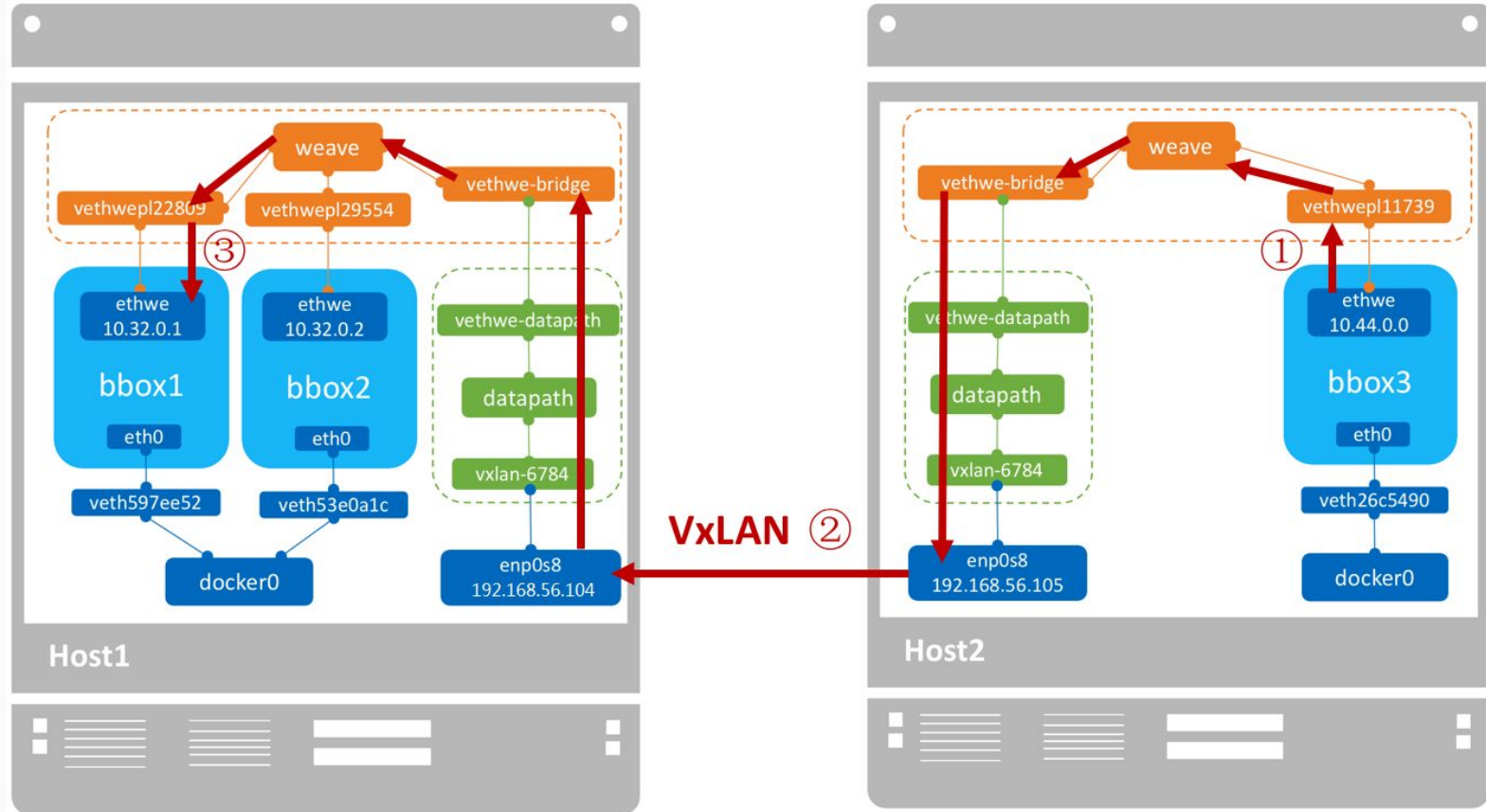
Overlay network



macvlan



Weave network



Flannet

