

Docker Overview

thinhducbui94@gmail.com



Module Target

Kết thúc bài học, học viên cần đạt được các kỹ năng sau:

- Nắm được kiến trúc container nói chung và Docker nói riêng, vai trò, tầm quan trọng của Docker.
- Biết cách tạo Docker image, quản lý images và containers.
- Hiểu và sử dụng Docker-compose để chạy các ứng dụng multi-containers
- Nắm được tổng quan về docker orchestration

Outline

1 Tổng quan về Container & Docker

- Lịch sử công nghệ ảo hóa
- Linux Container là gì? Tại sao dùng Container?
- So sánh Virtual Machines và Container
- Docker là gì? Hệ sinh thái Docker?

2 Kiến trúc Docker & Hướng dẫn cài đặt Docker

- Tổng quan kiến trúc của docker
- Cài đặt Docker.

3 Tương tác với Docker Image và Docker Container

- Tương tác cơ bản với docker images
- Tương tác cơ bản với docker containers

Section 1:

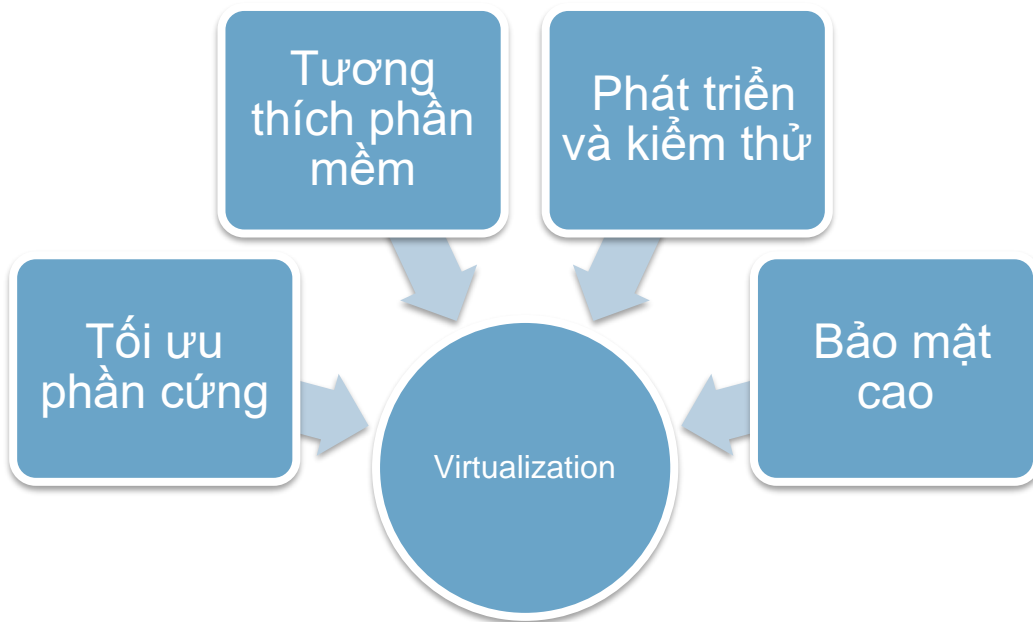
Tổng quan về Container và Docker



Lịch sử công nghệ ảo hóa



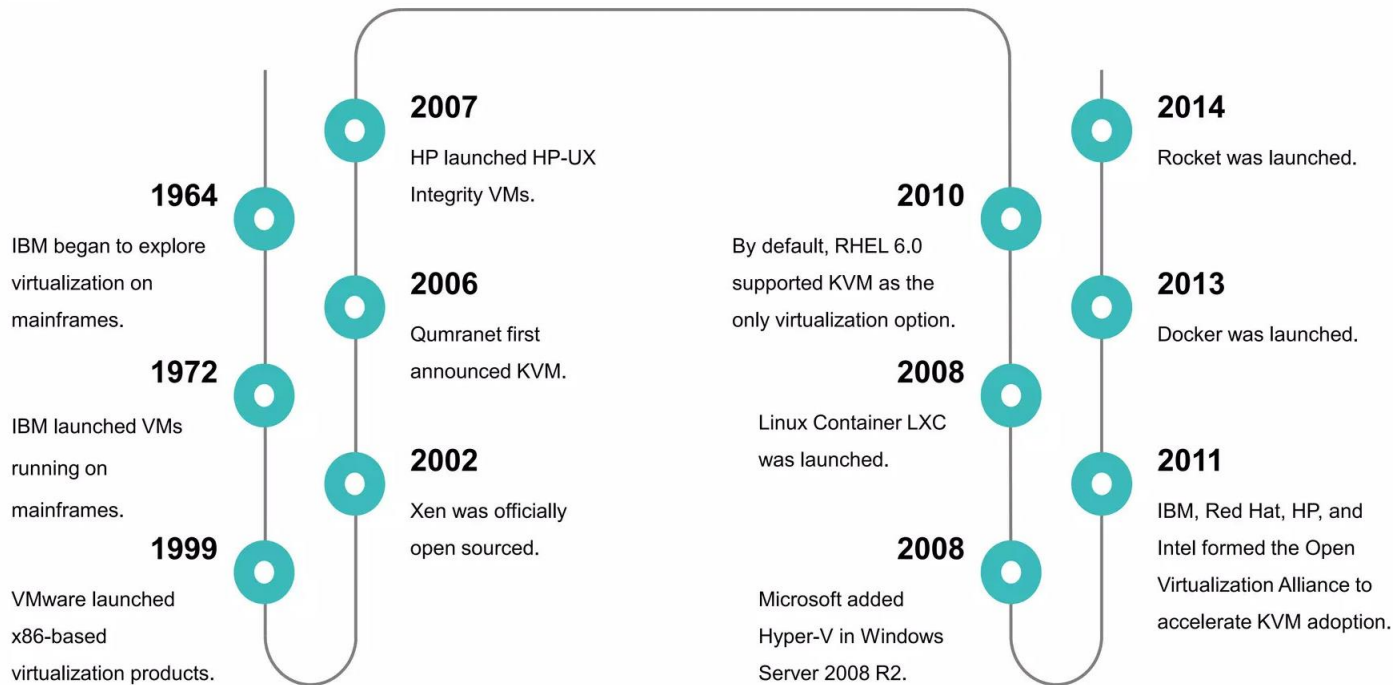
Tại sao cần ảo hóa (Virtualization)?



Lịch sử công nghệ ảo hóa



Timeline tóm tắt lịch sử công nghệ ảo hóa

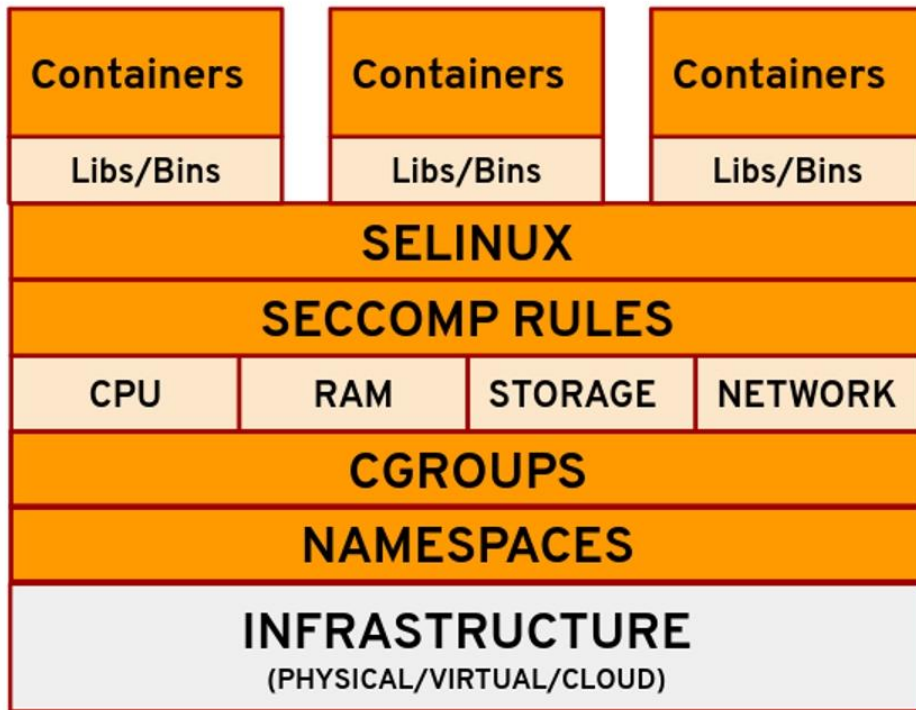


Linux Container là gì?



4 công nghệ nền tảng tạo nên Linux Container

1. Namespaces
2. Control Groups (cgroups)
3. Seccomp
4. SELinux



Tại sao sử dụng Container?

Lightweight

- Containers share the host operating system kernel, so they are much smaller and use fewer resources than traditional virtual machines.

Fast startup time

- Containers start up much faster than traditional virtual machines, which can take several minutes to boot up.

High portability

- Containers can run on any system that supports the containerization technology, as long as it has the necessary kernel features and resources.

High scalability

- Containers are highly scalable, as they can be easily cloned and deployed across multiple hosts.

Consistency

- Containers provide a consistent runtime environment, ensuring that applications run the same way across different platforms.

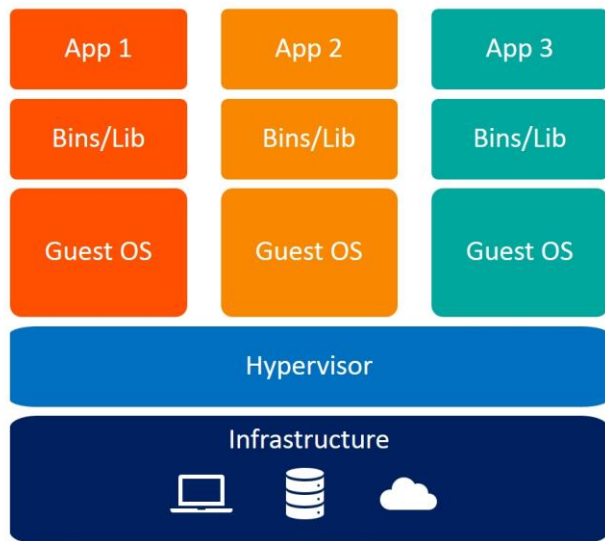
Isolation

- Containers provide a level of isolation between applications, allowing them to run independently without interfering with each other.

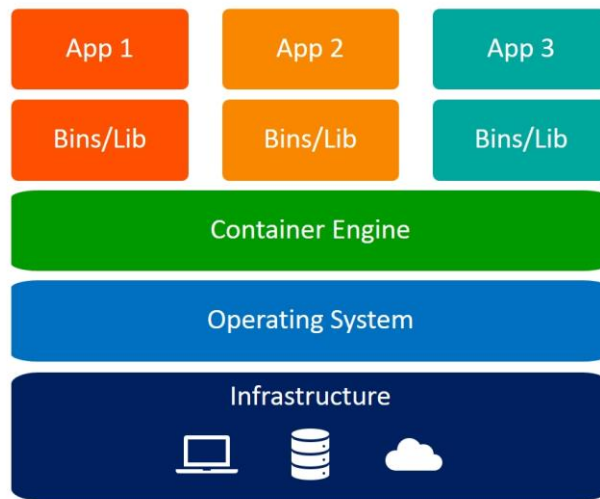
So sánh VM và Container



So sánh công nghệ



Virtual Machines



Containers

So sánh VM và Container



So sánh công nghệ

Máy ảo	Docker container
Kích thước (dung lượng) lớn.	Kích thước (dung lượng) nhỏ.
Hiệu suất hạn chế.	Hiệu suất gốc (native).
Mỗi máy ảo sẽ có một hệ điều hành riêng.	Container sẽ sử dụng hệ điều hành của host.
Ảo hóa về mặt phần cứng	Ảo hóa về mặt hệ điều hành
Thời gian khởi động tính theo phút	Thời gian khởi động tính theo mili giây
Phân bổ bộ nhớ theo nhu cầu cần thiết	Yêu cầu ít dung lượng bộ nhớ hơn
Hoàn toàn bị cô lập và an toàn hơn	Cô lập ở mức tiến trình, có thể kém an toàn hơn

Docker là gì?

Docker là gì?

Docker là một nền tảng phần mềm để xây dựng, đóng gói và triển khai các ứng dụng trong các container. Các container này được phân tách và cô lập khỏi hệ thống máy chủ và nhau và cho phép chạy các ứng dụng một cách độc lập với môi trường máy chủ. Docker cũng cung cấp các công cụ và API để quản lý và tự động hóa việc triển khai các ứng dụng trong container. Nó được sử dụng rộng rãi trong các môi trường phát triển, kiểm thử và triển khai ứng dụng.



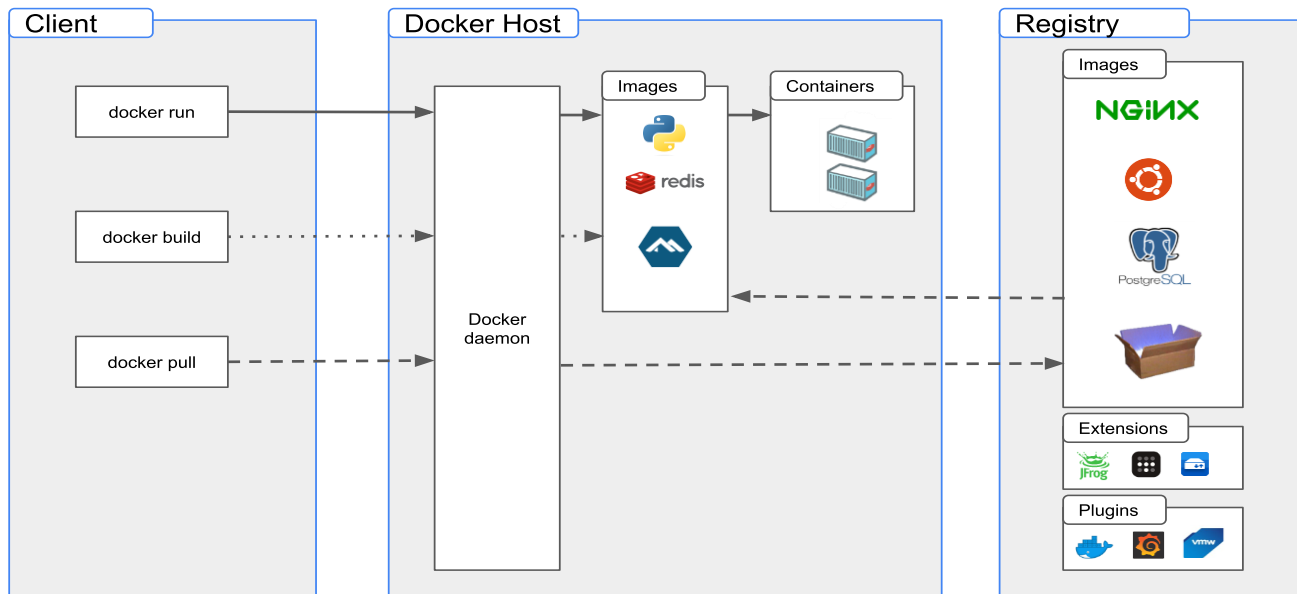
Section 2:

Kiến trúc Docker & Hướng dẫn cài đặt



Tổng quan kiến trúc Docker

Docker architecture diagram

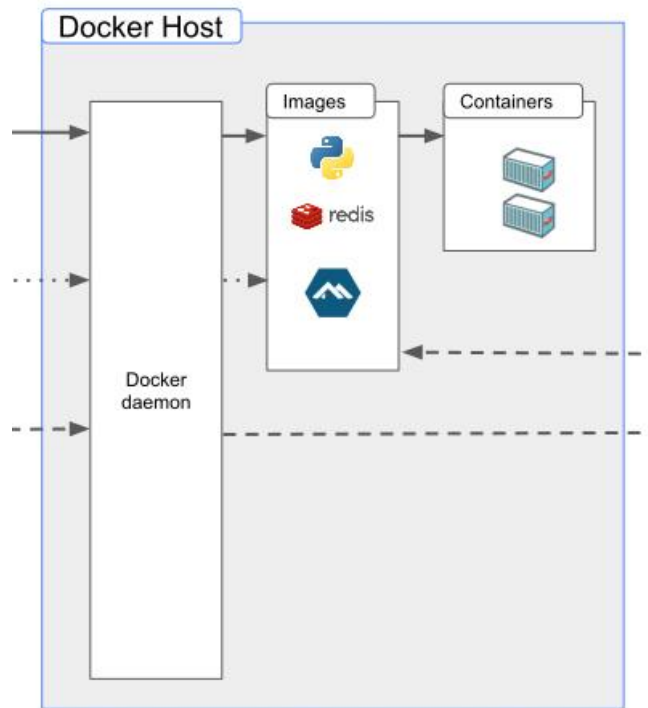


Docker sử dụng kiến trúc client-server

Tổng quan kiến trúc Docker

Docker daemon

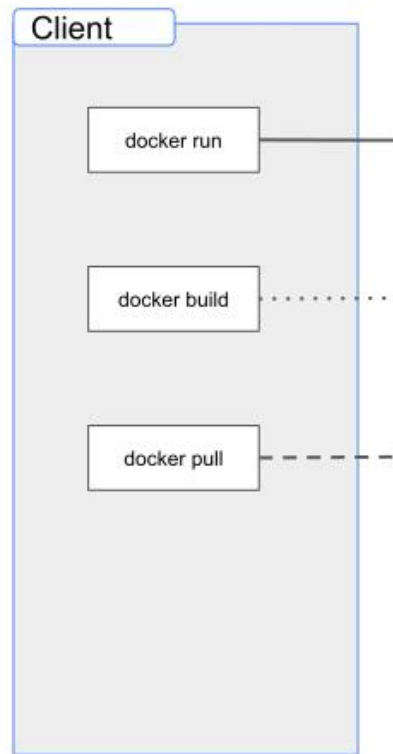
- Là một tiến trình trên hệ thống host, có trách nhiệm quản lý các container, images, networks và volumes. Docker daemon cũng cung cấp API cho các ứng dụng khác để tương tác với Docker.



Tổng quan kiến trúc Docker

Docker client

- Là một CLI (command-line interface) cho phép người dùng tương tác với Docker daemon. Docker client cũng cung cấp API cho các ứng dụng khác để tương tác với Docker.



Tổng quan kiến trúc Docker

Docker registry

- Docker registry chứa các docker images.
- Docker Hub là 1 public registry bất kì ai cũng có thể sử dụng, mặc định Docker sẽ tìm kiếm các images trên Docker Hub đầu tiên



Hướng dẫn cài đặt



Docker Desktop for Mac

A native application using the macOS sandbox security model which delivers all Docker tools to your Mac.

[Link cài đặt cho MacOS](#)



Docker Desktop for Windows

A native Windows application which delivers all Docker tools to your Windows computer.

[Link cài đặt cho Windows](#)



Docker Desktop for Linux

A native Linux application which delivers all Docker tools to your Linux computer.

[Link cài đặt cho Linux](#)

Section 3:

Tương tác với Image & Container



Tương tác với Docker Images



Manage images

Command	Description
<u>docker image build</u>	Tạo image từ Dockerfile
<u>docker image history</u>	Xem lịch sử image
<u>docker image import</u>	Import nội dung từ <i>tarball</i> để tạo ra filesystem của image.
<u>docker image inspect</u>	Hiển thị thông tin chi tiết của một hoặc nhiều images
<u>docker image load</u>	Nạp các images từ file *.tar hoặc STDIN
<u>docker image ls</u>	Hiển thị danh sách các images
<u>docker image prune</u>	Xóa các images không sử dụng

Tương tác với Docker Images

Command	Description
<u>docker image pull</u>	Tải xuống image từ 1 registry
<u>docker image push</u>	Tải image lên registry
<u>docker image rm</u>	Xóa 1 hoặc nhiều images
<u>docker image save</u>	Lưu một hoặc nhiều image ra file *.tar
<u>docker image tag</u>	Gắn tag cho TARGET_IMAGE tương ứng với SOURCE_IMAGE.

Tương tác với Docker Containers



Manage images

Command	Description
<u>docker container attach</u>	Attach local standard input, output, and error streams to a running container
<u>docker container commit</u>	Create a new image from a container's changes
<u>docker container cp</u>	Copy files/folders between a container and the local filesystem
<u>docker container create</u>	Create a new container
<u>docker container diff</u>	Inspect changes to files or directories on a container's filesystem
<u>docker container export</u>	Export a container's filesystem as a tar archive
<u>docker container exec</u>	Execute a command in a running container

Tương tác với Docker Containers

Command	Description
<u>docker container inspect</u>	Display detailed information on one or more containers
<u>docker container kill</u>	Kill one or more running containers
<u>docker container logs</u>	Fetch the logs of a container
<u>docker container ls</u>	List containers
<u>docker container pause</u>	Pause all processes within one or more containers
<u>docker container port</u>	List port mappings or a specific mapping for the container
<u>docker container prune</u>	Remove all stopped containers
<u>docker container rename</u>	Rename a container
<u>docker container update</u>	Update configuration of one or more containers

Tương tác với Docker Containers

Command	Description
<u>docker container restart</u>	Restart one or more containers
<u>docker container rm</u>	Remove one or more containers
<u>docker container run</u>	Create and run a new container from an image
<u>docker container start</u>	Start one or more stopped containers
<u>docker container stats</u>	Display a live stream of container(s) resource usage statistics
<u>docker container stop</u>	Stop one or more running containers
<u>docker container top</u>	Display the running processes of a container
<u>docker container unpause</u>	Unpause all processes within one or more containers
<u>docker container wait</u>	Block until one or more containers stop, then print their exit codes

Tương tác với Docker image & container Lab

Lab 1 Thực hành tải các docker image sau:

1. Tải image *ubuntu* phiên bản 20.04 trên docker hub
2. Tải image *todo1ist-sample* phiên bản mới nhất trên docker hub
3. Tải image *nginx* phiên bản bất kì trên docker hub

Tương tác với Docker image & container Lab

Lab 2 Thực hành chạy image ubuntu

1. Sử dụng lệnh `docker run` để khởi chạy 1 container từ image ubuntu đã tải về theo cú pháp sau:

```
docker run -it -name sample_name image:tag
```

2. Kiểm tra xem container vừa tạo đã chạy chưa (`ps`)
3. Stop container vừa khởi chạy trên
4. Start lại container vừa dừng
5. Attach vào container đang chạy
6. Xóa container
7. Kiểm tra thông tin chi tiết của container bằng lệnh `inspect`

Tương tác với Docker image & container Lab

Lab 3 Chạy image todolist

1. Sử dụng lệnh `docker run` để khởi chạy 1 container từ image todolist đã tải về theo cú pháp sau:

```
docker run -it -name sample_name image:tag
```

2. Kiểm tra các file đang có trong container
3. Tạo đường dẫn mới như sau trong image `/var/www/testing_container/`
4. Tạo 1 file tên `hello_world` trong đường dẫn vừa tạo có nội dung `hello from the inside`

Thank you !

