Docker Overview

thinhducbui94@gmail.com

Module Target

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

- Nắm được kiến trúc container nói chung và Docker nói riêng, vài 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 images và Docker Container

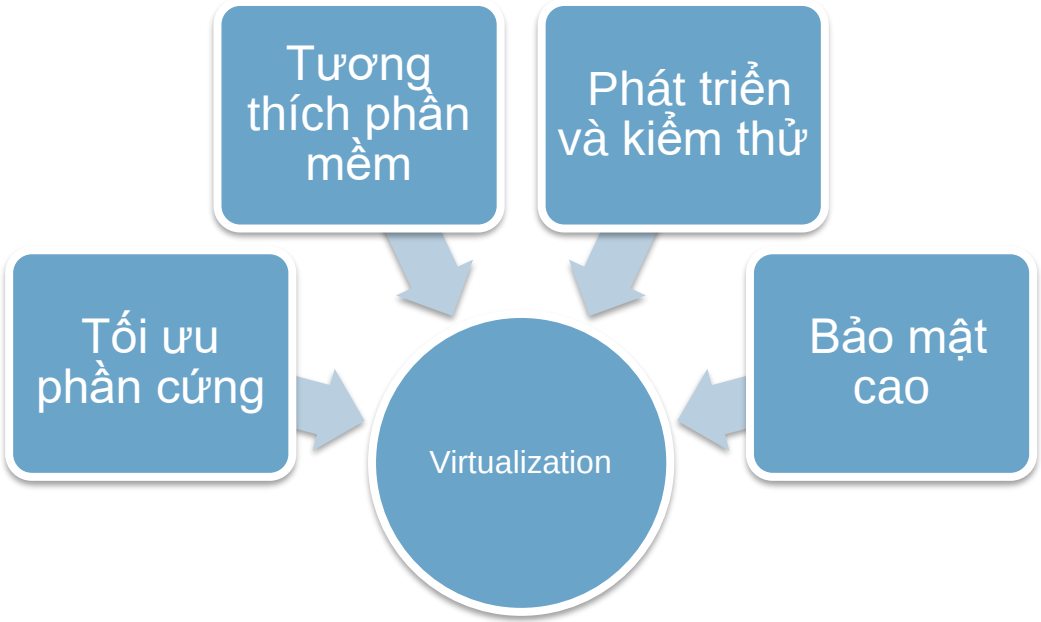
- Tương tác cơ bản với docker images
- Tương tác cơ bản với docker containers

Section 1:

Tổng quan về Container và Docker

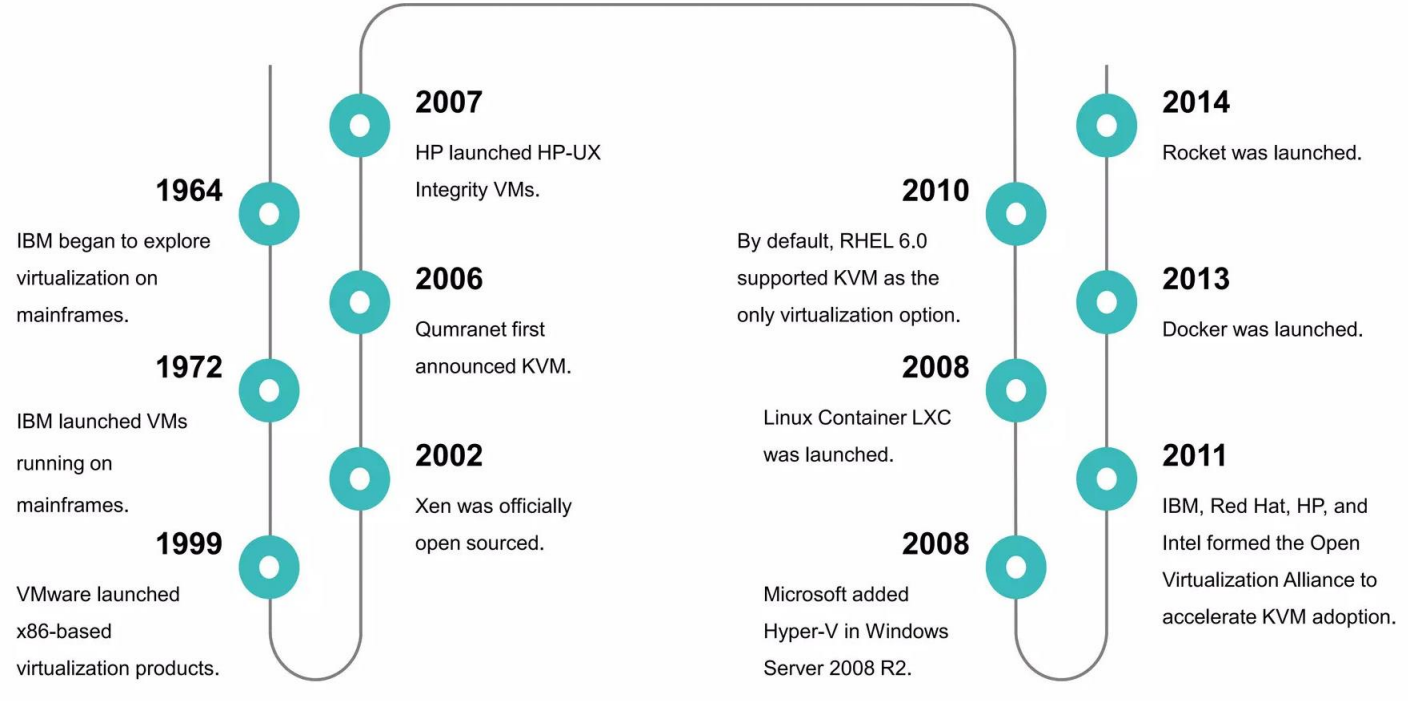
Lịch sử công nghệ ảo hóa

Tại sao cần ảo hóa (Virtualization)?



Lịch sử công nghệ ảo hóa

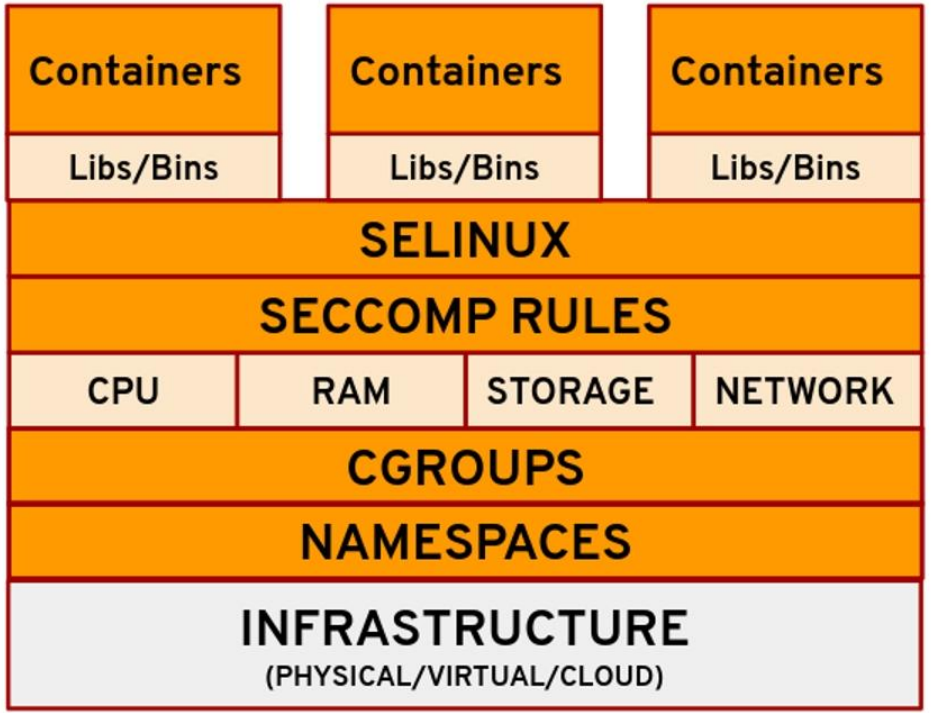
Timeline tóm tắt lịch sử công nghệ ảo hóa



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

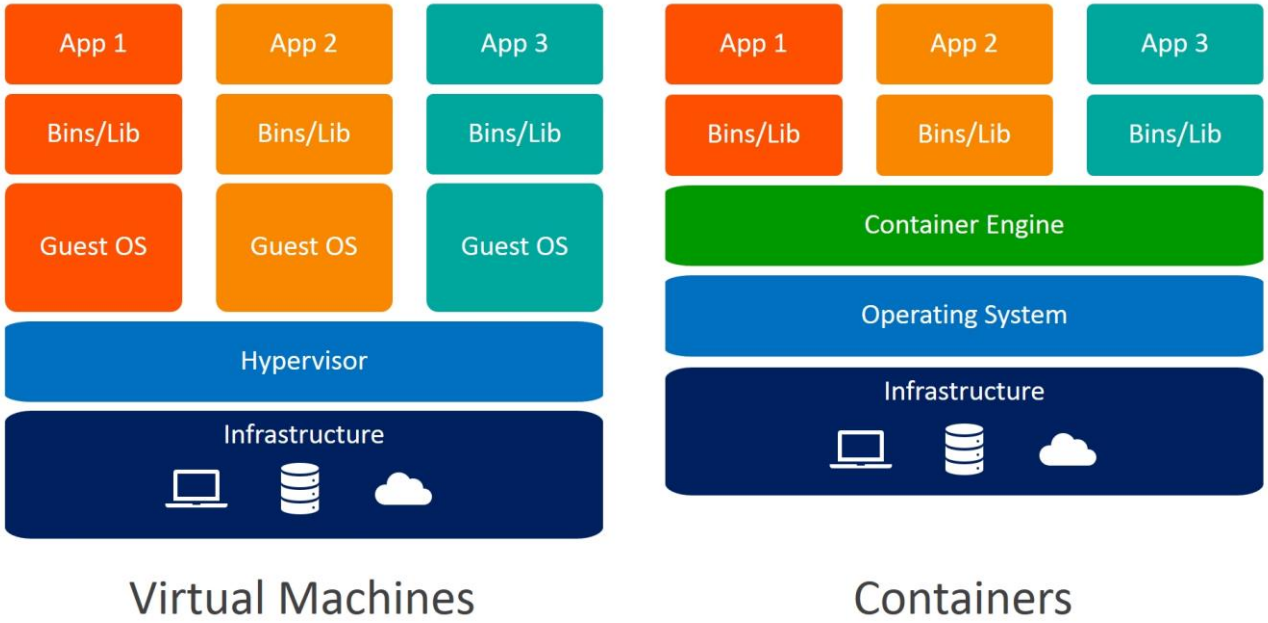
- Containers provide a consistent runtime environment, ensuring that applications run the same way across different platforms.

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



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?

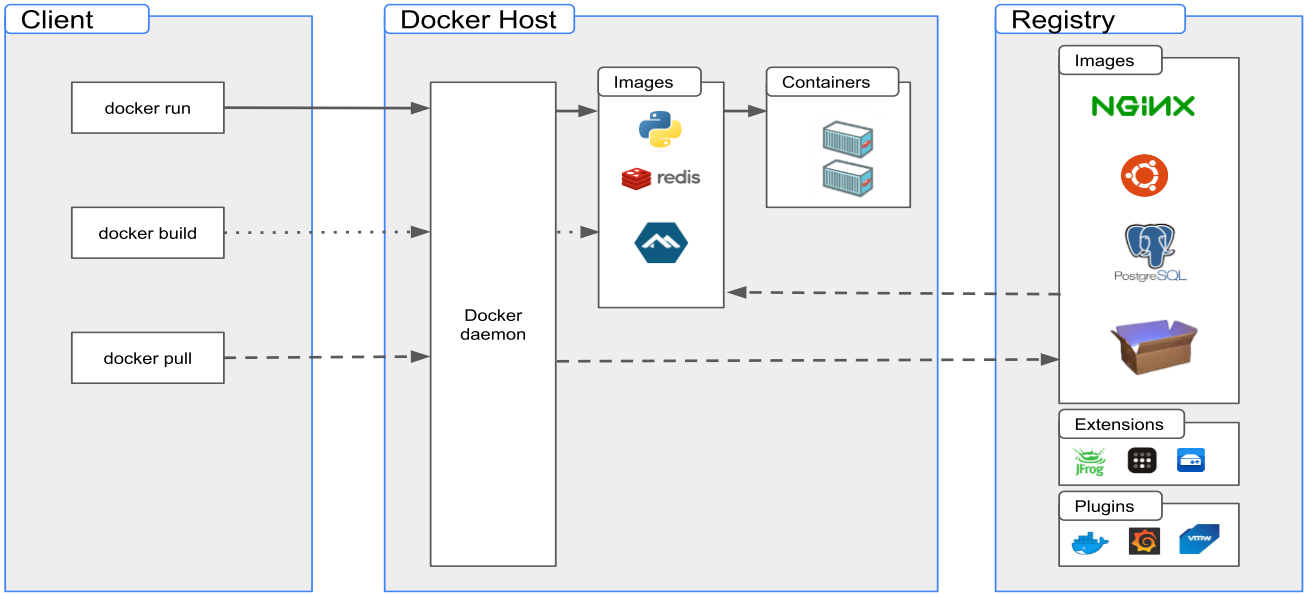
Docker là một nền tảng phần mềm để xây dựng, đóng gói và triển khai các ứng dụng trong các container. Các container này được phân tách và cô lập khỏi hệ thống máy chủ và nhau và cho phép chạy các ứng dụng một cách độc lập với môi trường máy chủ. Docker cũng cung cấp các công cụ và API để quản lý và tự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

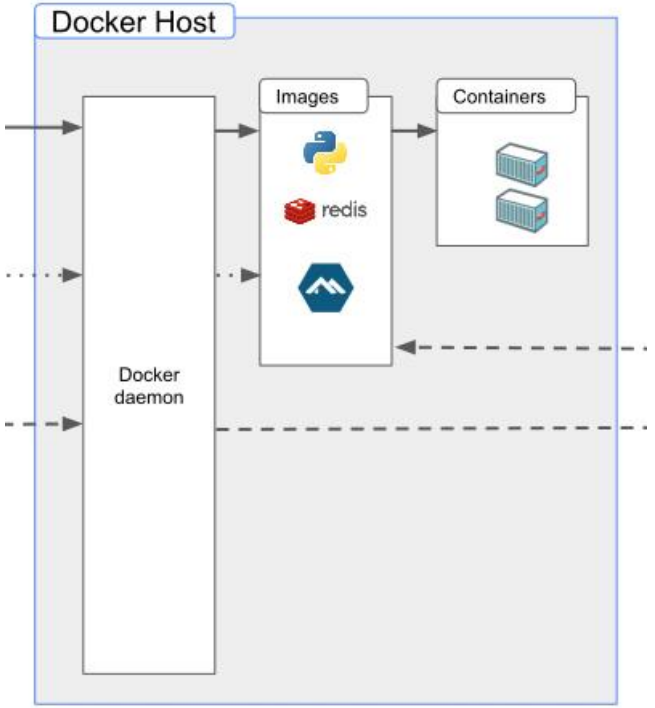


Docker sử dụng kiến trúc client-server

Tổng quan kiến trúc Docker

Docker daemon

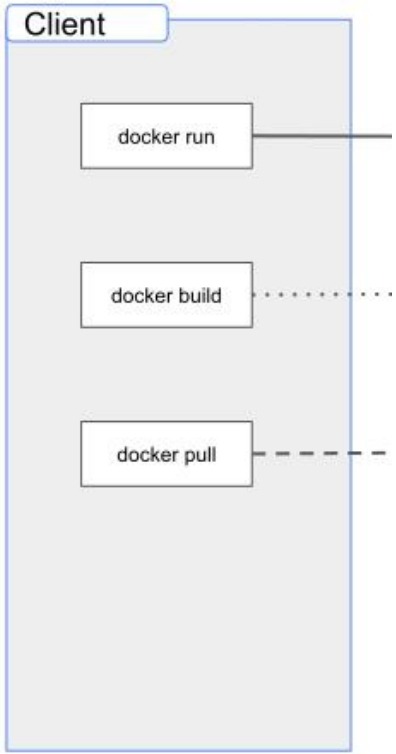
- Là một tiến trình trên hệ thống host, có trách nhiệm quản lý các container, images, networks và volumes. Docker daemon cũng cung cấp API cho các ứng dụng khác để tương tác với Docker.



Tổng quan kiến trúc Docker

Docker client

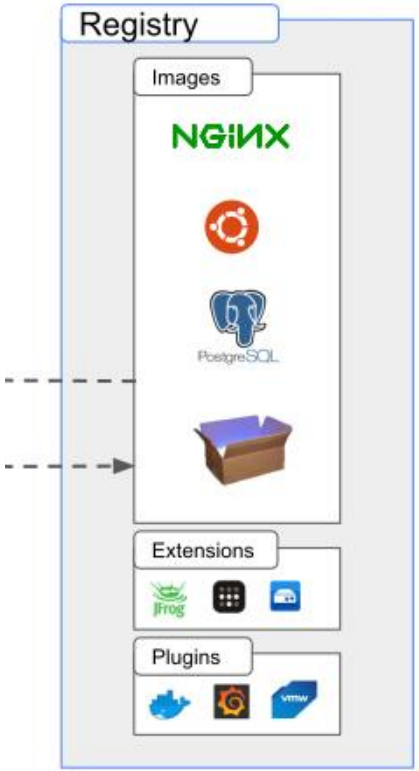
- Là một CLI (command-line interface) cho phép người dùng tương tác với Docker daemon. Docker client cũng cung cấp API cho các ứng dụng khác để tương tác với Docker.



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

Tương tác với Docker Images

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

Tương tác với Docker Containers

Manage images

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

Tương tác với Docker Containers

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

Tương tác với Docker Containers

Command	Description
docker container restart	Restart one or more containers
docker container rm	Remove one or more containers
docker container run	Create and run a new container from an image
docker container start	Start one or more stopped containers
docker container stats	Display a live stream of container(s) resource usage statistics
docker container stop	Stop one or more running containers
docker container top	Display the running processes of a container
docker container unpause	Unpause all processes within one or more containers
docker container wait	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 *todolist-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 !