

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



# BÁO CÁO THUYẾT TRÌNH HỆ ĐIỀU HÀNH - CO2018

---

## VIRTUAL MACHINE VERSUS CONTAINER

---

GV hướng dẫn: Hoàng Lê Hải Thanh  
SV thực hiện: Nguyễn Đức An – 2010102  
Trần Phúc Anh – 2010133  
Nguyễn Quang Huy – 1916081  
Trần Hà Tuấn Kiệt – 2011493

Tp. Hồ Chí Minh, Tháng 6/2022

# Mục lục

<b>1</b>	<b>Giới thiệu về Virtual Machine &amp; Container</b>	<b>2</b>
1.1	Giới thiệu chung . . . . .	2
1.2	Virtualization & Hypervisor . . . . .	3
1.2.1	Virtualization . . . . .	3
1.2.2	Hypervisor . . . . .	3
1.3	Virtual Machine . . . . .	3
1.3.1	Khái niệm . . . . .	3
1.3.2	Đặc điểm . . . . .	4
1.4	Container . . . . .	5
1.4.1	Khái niệm . . . . .	5
1.4.2	Đặc điểm . . . . .	6
<b>2</b>	<b>So sánh một số đặc điểm giữa Virtual Machine &amp; Container</b>	<b>8</b>
<b>3</b>	<b>Một số ưu điểm, nhược điểm của Virtual Machine &amp; Container</b>	<b>9</b>
3.1	Ưu nhược điểm của Virtual Machine . . . . .	9
3.2	Ưu nhược điểm của Container . . . . .	10
<b>4</b>	<b>Ứng dụng của Virtual Machine &amp; Container</b>	<b>11</b>
4.1	Ứng dụng Virtual Machine . . . . .	11
4.2	Ứng dụng Container . . . . .	11
<b>5</b>	<b>Tài liệu tham khảo</b>	<b>12</b>

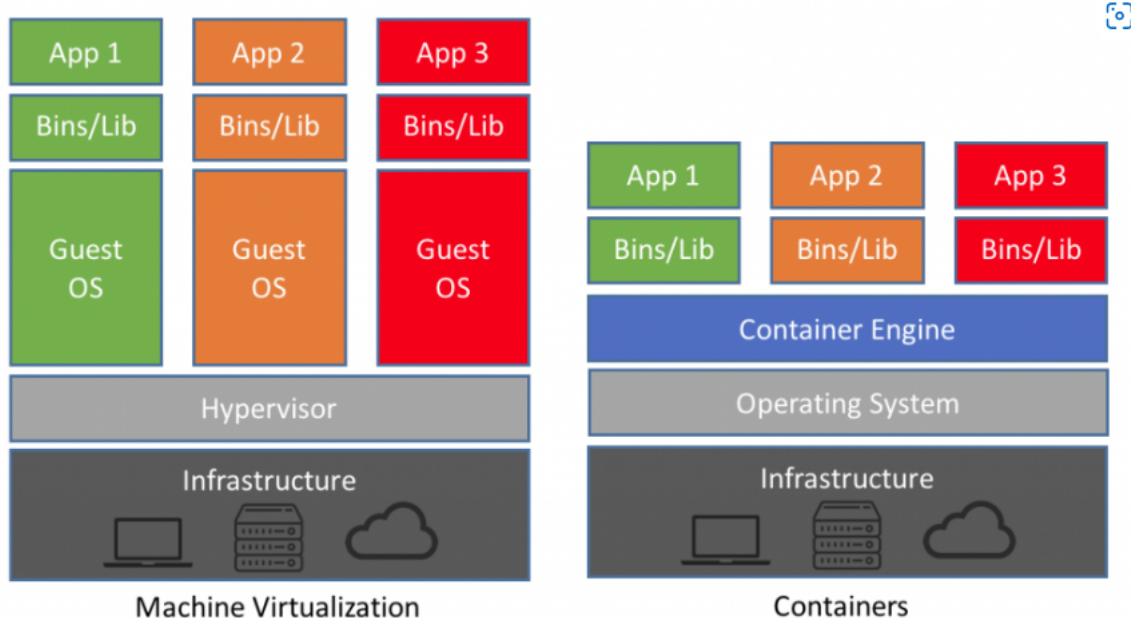
# 1 Giới thiệu về Virtual Machine & Container

## 1.1 Giới thiệu chung

Cả Virtual Machine và Container đều dùng công nghệ ảo hóa (virtualization) để tận dụng tối đa được tài nguyên của phần mềm và phần cứng.

Container chỉ mới phổ biến cách đây không lâu với sự có mặt của Docker nhưng những ứng dụng rộng rãi của nó trong những năm qua đã làm thay đổi đáng kể đến ngành công nghệ thông tin.

Trong khi đó, Virtual Machine đã phổ biến từ lâu và có mặt trong một số dịch vụ phổ biến chúng ta hay sử dụng như Virtual Box, VMware Workstation Player,...



**Hình 1:** *Virtual Machine và Container*

## 1.2 Virtualization & Hypervisor

### 1.2.1 Virtualization

- Công nghệ ảo hóa (Virtualization) là quá trình tạo ra những **computing environment**, môi trường này được trừu tượng hóa (abstract) so với physical computing hardware.
- Với công nghệ ảo hóa, ta có thể tạo ra nhiều máy ảo (instance) từ tài nguyên của máy tính vật lý. Các máy ảo này có đầy đủ các đặc điểm như một máy tính thông thường, tuy nhiên sử dụng tài nguyên được chia ra từ máy tính vật lý, nên lượng tài nguyên thường ít hơn.

### 1.2.2 Hypervisor

- Hypervisor là một phần mềm hỗ trợ virtualization. Nó là một lớp layer mỏng nằm giữa phần cứng và môi trường ảo.
- Hypervisor là một lớp trung gian phân phát tài nguyên máy tính cho các máy ảo.

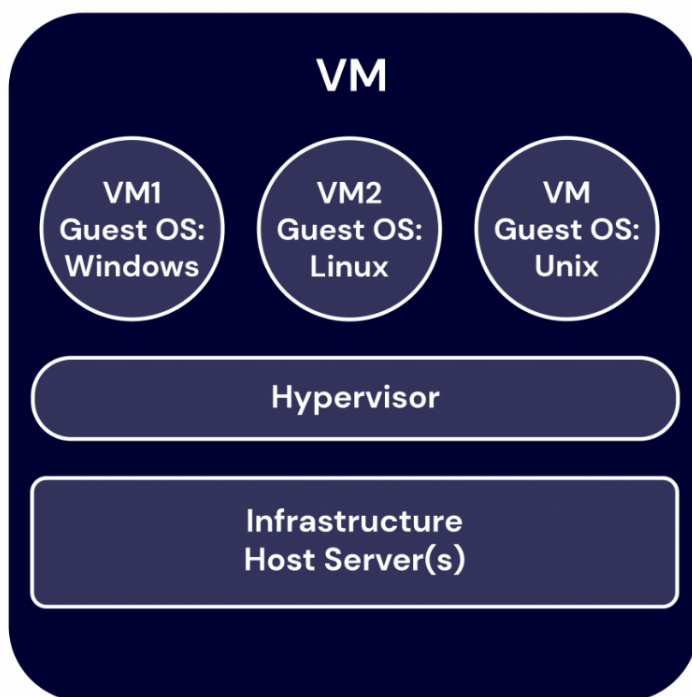
## 1.3 Virtual Machine

### 1.3.1 Khái niệm

Virtual Machine hiểu một cách đơn giản đó là những computer-generated computers được sinh ra từ công nghệ ảo hóa (virtualization) - mỗi máy ảo là một máy tính riêng biệt chạy trên phần cứng của một máy tính vật lý.

Mỗi Virtual Machine có một hệ điều hành (OS) riêng của nó. Các máy ảo (Virtual Machine) trong cùng máy tính cùng chia sẻ với nhau tài nguyên của máy tính vật lý chung (physical computer).

Kể từ khi sự ra đời của các công nghệ virtualization với giá thành rẻ và các dịch vụ điện toán đám mây, các bộ phận IT lớn và nhỏ đã sử dụng Virtual Machine như là một cách để giảm chi phí vì có thể sử dụng nhiều máy ảo khác nhau trên cùng một máy tính vật lý, thay vì phải đầu tư chi phí cho nhiều máy tính vật lý.



Hình 2: *Virtual Machine*

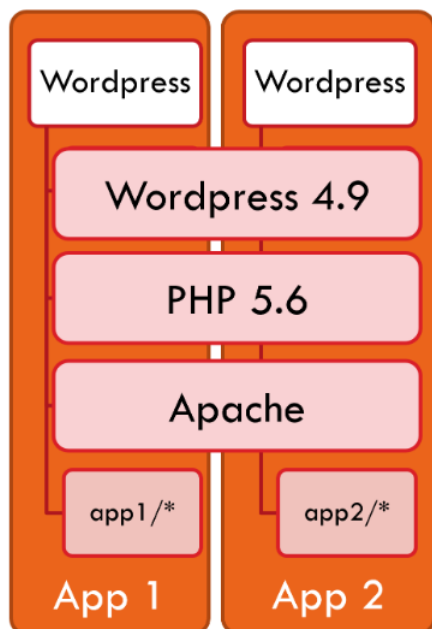
### 1.3.2 Đặc điểm

#### Một số đặc điểm phổ biến

- Các máy ảo sử dụng nguồn tài nguyên được phân phát từ tài nguyên máy tính vật lý, mỗi máy ảo có thể cài đặt hệ điều hành riêng của nó. Do đó mỗi máy ảo có thể được xem như một computer-generated computer.
- Máy ảo có thể cài hệ điều hành khác với hệ điều hành của máy chủ (Host OS).
- Các máy ảo dù cùng được chia tài nguyên từ cùng một máy chủ vật lý, nhưng hoạt động của mỗi máy ảo là hoàn toàn độc lập nhau. Do đó, ta có thể chạy nhiều máy ảo cùng lúc.
- Máy ảo thường gắn với **kiến trúc monolithic** - tất cả đều nằm trong cùng một khối và thường được dùng để thực thi các chương trình đơn hoặc các files dữ liệu lớn.
- Sử dụng máy ảo để tiết kiệm chi phí khi muốn làm việc trên nhiều hệ điều hành khác nhau, thay vì phải thực thi trên các máy tính vật lý khác nhau.

**Lưu ý:** Một vấn đề lớn nhất dẫn đến tại sao chúng ta phải nghiên cứu container thay cho virtual machine là xảy ra trường hợp **conflict**, thường là **software conflict**. Một số conflict có thể xảy ra liên quan đến: environments—software versions, permissions, database access,.. đối với các library được chia sẻ từ Host OS đến OS của các máy ảo, vì

dù như một phần mềm trên máy ảo là phiên bản mới, có một số tính năng mới thì không thể chạy được nếu thư viện do host OS cung cấp cho máy ảo là phiên bản cũ hơn.



Hình 3: Conflict when using Virtual Machine

## 1.4 Container

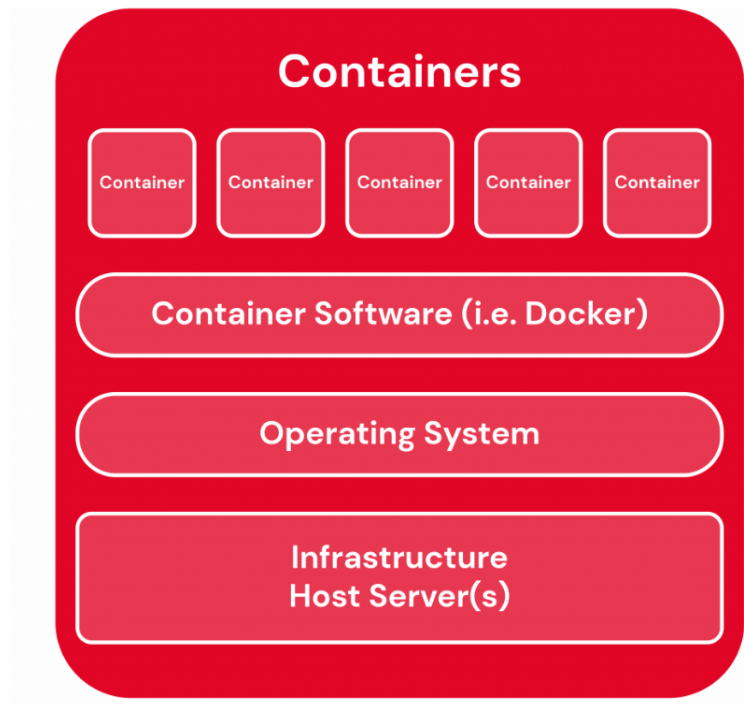
### 1.4.1 Khái niệm

Container là một gói (packages) các thành phần cần thiết để phần mềm có thể chạy trên những môi trường khác nhau. Khác với virtual machine, việc ảo hóa xảy ra trên toàn bộ máy ảo thì container việc ảo hóa chỉ xảy ra với hệ điều hành (OS). Do đó, container có thể được dùng để chạy trên nhiều nền tảng khác nhau, từ cloud cho đến personal computer.

Container đóng gói tất cả những thành phần cần để chạy chương trình (runtime environment) như: chương trình(application), cùng với các thành phần liên quan như dependencies, libraries, binaries file, configuration files cần thiết để build và chạy chương trình.

Do Container chỉ là đóng gói các file, phần mềm để chạy chương trình, do đó để chạy được chương trình trong container, ta cần phải cần có các **container engine**, một trong các engine phổ biến nhất hiện nay là **Docker**.

Các thành phần được chia sẻ này sẽ được thiết lập ở chế độ chỉ đọc (read-only), nghĩa là thành phần của một container người dùng không thể thay đổi (immutable).



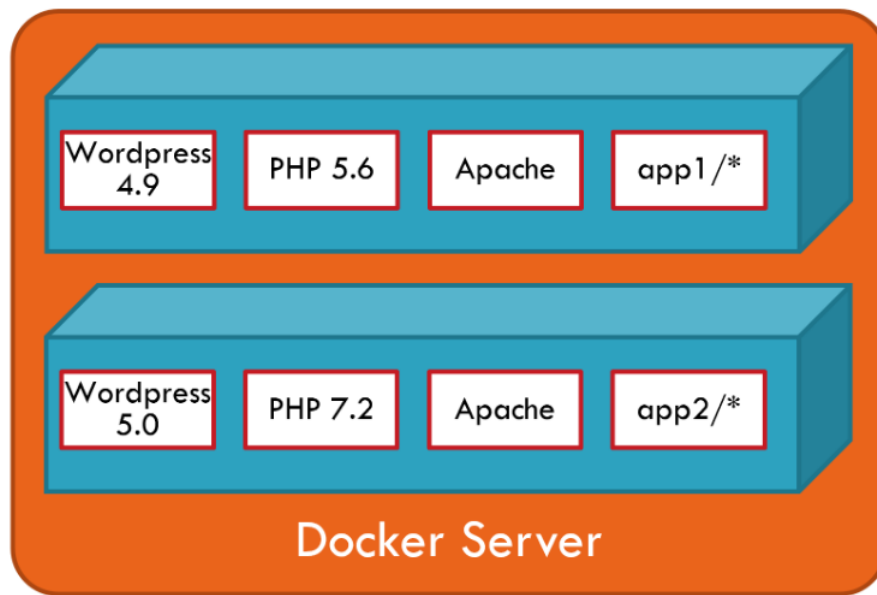
Hình 4: Container

#### 1.4.2 Đặc điểm

##### Một số đặc điểm phổ biến

- Container chỉ được chạy trên engine, ví dụ như Docker và container sử dụng hệ điều hành từ máy chủ (host OS).
- Do container chỉ là đóng gói các phần mềm, do đó mỗi container không bao gồm hệ điều hành (OS), cũng như tài nguyên phần cứng (RAM, SSD,...) như Virtual Machine, do đó Container có kích thước rất nhẹ, chỉ vài megabytes và chỉ mất khoảng vài giây để khởi động.
- Tại một thời điểm, chúng ta có thể sử dụng nhiều hơn gấp đôi hay gấp ba lần số lượng chương trình trên một máy chủ với Container so với Virtual Machine.
- Chúng ta có thể sử dụng kiến trúc microservices để phát triển ứng dụng đã phát triển ứng dụng với Container. Các ứng dụng có thể được chia ra thành các phần nhỏ hơn có thể được phát triển độc lập với nhau thay vì nằm trong cùng một khối monolithic. Ví dụ, chúng ta có một ứng dụng giúp người dùng có thể mua được bất kỳ món hàng nào trên thế giới. Chúng ta có thanh tìm kiếm, vỏ mua hàng, nút mua hàng, .... Mỗi "services" đó có thể tồn tại trong Container riêng của nó. Vì thế, nếu chẳng hạn thanh search bị lỗi thì cả hệ thống sẽ không bị break theo.

**Lưu ý:** Container đã giải quyết được vấn đề conflict trên, Ở mỗi container, chương trình và môi trường chạy (runtime environment) được đóng gói thành một khối hoàn chỉnh, các thư viện (libraries), dependencies có cùng phiên bản (version) với phiên bản của chương trình, do đó sẽ không xảy ra lỗi không tương thích hay software conflict khi chạy trên các nền tảng khác nhau cho dù đó là local, development hay product.



**Hình 5:** *Conflict Solve with Container*



## 2 So sánh một số đặc điểm giữa Virtual Machine & Container

Đặc điểm	Virtual Machine	Container
Cách sử dụng	Khi sử dụng VM, chúng ta cài đặt trực tiếp các phần mềm (software) lên máy ảo, và sử dụng tương tự như khi cài đặt trên máy thật	Container cho phép chúng ta sử dụng phần mềm bằng cách chạy trên engine như Docker, chứ không cần cài đặt trực tiếp vào máy
Hệ điều hành	Phần mềm chạy trên VM sử dụng hệ điều hành của máy ảo đó, có thể khác với hệ điều hành trên máy chủ (Host OS)	Phần mềm chạy trên Container sử dụng hệ điều hành của máy chủ (Host OS)
Virtualization	VM ảo hóa toàn bộ computer system, mỗi máy ảo như một computer-generated computers	Container chỉ ảo hóa phần hệ điều hành (Host OS)
Kích thước	VM có kích thước lớn, tính theo megabytes hoặc thậm chí gigabytes	Container có kích thước tương đối nhỏ, chỉ vài megabytes
Thời gian chạy	VM tốn vài phút để chạy chương trình	Container chỉ tốn vài giây để chạy chương trình
Khối lượng bộ nhớ	VM sử dụng nhiều memory	Container sử dụng ít memory hơn.
Tính bảo mật	VM có tính bảo mật cao hơn, những vấn đề xảy ra ở máy ảo sẽ không ảnh hưởng đến máy thật	Container có tính bảo mật yếu hơn, do sử dụng chung hệ điều hành với máy chủ.
Ứng dụng	Sử dụng khi có nhu cầu sử dụng nhiều hệ điều hành khác nhau	Sử dụng khi có nhu cầu chạy nhiều ứng dụng khác nhau.

## 3 Một số ưu điểm, nhược điểm của Virtual Machine & Container

### 3.1 Ưu nhược điểm của Virtual Machine

#### Ưu điểm:

- Máy ảo có tính bảo mật cao do các máy ảo hoạt động độc lập với nhau. Các tài nguyên vật lý khác được bảo vệ hoàn toàn vì các máy ảo có các thiết bị ảo riêng. Nhờ vậy, mình có thể yên tâm cài đặt những ứng dụng không đáng tin cậy để thử trong môi trường máy ảo, hoặc dùng để nghiên cứu virus máy tính mà không sợ bị ảnh hưởng vì nếu có sao cũng chỉ hỏng máy ảo.
- Máy ảo giúp ta dễ dàng phát triển hệ thống hơn mà không sợ làm ảnh hưởng tới toàn bộ hệ máy đang vận hành.
- Với máy ảo, ta có thể cài đặt các hệ điều hành khác nhau trên nó để phục vụ các mục đích như học tập, nghiên cứu...
- Máy ảo giúp ta tiết kiệm chi phí vì không cần phải mua thêm các tài nguyên vật lý mà vẫn có thể sử dụng môi trường ảo. Ngoài ra, máy ảo còn giúp ta giảm nhu cầu duy trì nhiều máy chủ, tiết kiệm chi phí bảo trì và lượng điện sử dụng.
- Máy ảo cho phép dễ dàng mở rộng ứng dụng của mình bằng cách thêm nhiều máy chủ vật lý hoặc ảo hơn để phân phối khối lượng công việc trên nhiều máy ảo. Do đó, làm tăng tính khả dụng và hiệu suất của một số ứng dụng.

#### Nhược điểm:

- Nếu máy tính có cấu hình phần cứng thấp nhưng cài quá nhiều chương trình máy ảo, máy sẽ chậm và ảnh hưởng đến các chương trình khác.
- Do tập trung vào một máy tính, nên nếu máy bị hư thì toàn bộ các máy tính ảo đã được thiết lập trên nó cũng bị ảnh hưởng theo.
- Ở góc độ bảo mật, máy ảo có nhược điểm là nếu hacker nắm quyền điều khiển máy tính chứa các máy ảo thì hacker có thể kiểm soát được tất cả các máy ảo trong nó.

## 3.2 Ưu nhược điểm của Container

### Ưu điểm:

- Container thực hiện quản lý, cấp phát và chia sẻ tài nguyên hiệu quả hơn, không bị giới hạn về tài nguyên "cứng" như trong máy ảo.
- Có khả năng tiết kiệm tài nguyên hệ thống nhiều hơn so với khi sử dụng máy ảo vì Container không phải tốn tài nguyên cho các Hypervisor và hệ điều hành khách.
- Đạt hiệu năng cao hơn so với máy ảo vì Container chạy trực tiếp trên cùng một nhân hệ điều hành, không phải thông qua Hypervisor và hệ điều hành khách như hình thức ảo hóa phần cứng.
- Vì có kích thước nhỏ nên các Container tốn rất ít thời gian để khởi động, rất thích hợp với các ứng dụng cần được đẩy lên và xuống liên tục.
- Các Container được thiết kế để có thể khởi động thường xuyên, nhờ vậy mà dễ dàng tiếp nhận các thay đổi.
- Container có khả năng làm tăng tính di động của ứng dụng, các ứng dụng chạy trong Container có thể dễ dàng được triển khai trên nhiều hệ điều hành và các nền tảng phần cứng khác nhau.

### Nhược điểm:

- Vì giữa các Container chia sẻ chung kernel của hệ điều hành nên Container sẽ có tính bảo mật kém hơn so với VM, chỉ cần kernel bị tấn công thì toàn bộ các Container sẽ bị ảnh hưởng theo.
- Hiệu năng hoạt động của Container sẽ không ổn định nếu các tài nguyên không được phân chia hợp lý. Vì tài nguyên hệ thống được chia sẻ giữa các Container chứ không được phân chia "cứng" như trên máy ảo, nên khi 1 container chiếm dụng quá nhiều tài nguyên sẽ gây ra ảnh hưởng đến các Container khác trong cùng hệ thống.
- Gặp bất lợi trong việc lưu dữ liệu hơn so với khi dùng máy ảo. Trong một số trường hợp, dữ liệu trong container sẽ bị mất vĩnh viễn trừ khi ta lưu nó ở chỗ khác trước khi tắt container.

## 4 Ứng dụng của Virtual Machine & Container

### 4.1 Ứng dụng Virtual Machine

VM giúp các tổ chức xây dựng và triển khai các ứng dụng trên nền tảng điện toán đám mây. Các VM có thể được triển khai để cung ứng lượng năng lượng tiêu thụ dành cho việc xử lý ở các mức độ khác nhau, để đáp ứng môi trường lý tưởng cho phần mềm hoạt động hoặc để kiểm tra các ứng dụng trong một môi trường thử nghiệm độc lập và an toàn.

Các VM có lịch sử lâu đời trong việc ảo hoá server, giúp quản lý tài nguyên tính toán và cải thiện hiệu suất vận hành. Thêm vào đó, các VM có thể thực hiện các tác vụ đặc biệt, thường được xem là quá mạo hiểm để thực hiện trên máy chủ, chẳng hạn truy cập dữ liệu bị nhiễm mã độc/virus hoặc thử nghiệm hệ điều hành mới. Việc ảo hoá còn giúp các ứng dụng tương thích với hệ điều hành đời cũ hoặc chỉ tương thích với hệ điều hành khác hoạt động theo ý muốn của dùng.

### 4.2 Ứng dụng Container

Container cung cấp cơ chế đóng gói cục bộ cho phép các ứng dụng tách biệt khỏi môi trường mà chúng thực sự chạy. Cơ chế này giúp các ứng dụng chạy trên nền container triển khai một cách dễ dàng và bền vững, bất kể môi trường phát triển là trung tâm dữ liệu cá nhân hoá, môi trường cloud công cộng, hay trên máy tính cá nhân của các nhà phát triển.

Sự nhanh gọn: Container cho phép các nhà phát triển chuyển đổi máy chủ một cách nhanh chóng mà không cần bận tâm dependency và môi trường. Việc này giúp tăng sự cộng tác giữa các nhóm phát triển, từ đó đẩy nhanh tiến độ ra mắt ứng dụng.

Sự linh động: Container cung cấp một định dạng chuẩn cho việc đóng gói và giữ các thành phần ở mức cần thiết cho việc chạy ứng dụng mong muốn. Điều này giúp tối đa hoá sự linh động giữa các nền tảng OS và giữa các nền tảng đám mây. Mỗi khi một container được triển khai đâu đó trên khắp nơi, container đó vẫn sẽ thực thi trong một môi trường nhất quán.

Khả năng mở rộng: Cơ chế của Container gọn nhẹ và cho phép sử dụng tài nguyên tính toán cần thiết. Các container có thể khởi chạy và tạm dừng nhanh hơn rất nhiều so với VM, mở ra khả năng mở rộng và thu hẹp nhanh chóng của hệ thống.

## 5 Tài liệu tham khảo

### Một số tài liệu

- [1] BackBlaze, What's the Diff: VMs vs. Containers, Link: <https://www.backblaze.com/blog/vm-vs-containers/>
- [2] G4G, Difference between Virtual Machines and Containers, Link: <https://www.geeksforgeeks.org/difference-between-virtual-machines-and-containers/>
- [3] Wiley, *Chapter 18 Virtual Machines, Operating System Concepts*, Tenth Edition.
- [4] Docker Website, *Docker, Image, Container, Dockerfile*, <https://www.docker.com/resources/what-container/>
- [5] Accelerate Rediscover In 4 Minutes, *Virtual Machine - Explained in 4 Minutes*, Youtube, Link: <https://youtu.be/C49SQBoKCP4>
- [6] Accelerate Rediscover In 4 Minutes, *Containers - Explained in 4 Minutes*, Youtube, Link: <https://youtu.be/pR-cGS6IGvI>
- [7] XuanThuLab, Docker Tutorial, Link: <https://youtu.be/r6JiWwh-08c>
- [8] IBM, Containers vs. Virtual Machines (VMs): What's the Difference?, Link: <https://www.ibm.com/cloud/blog/containers-vs-vm>
- [9] Microsoft Azure, What is a virtual machine (VM)?, Link: <https://azure.microsoft.com/en-us/overview/what-is-a-virtual-machine/#what-used-for>
- [10] Microsoft Azure, What is a container?, Link: <https://azure.microsoft.com/en-us/overview/what-is-a-container/#why-containers>