

	<b>VIETTEL AI RACE</b>	Public 192
	<b>GIỚI THIỆU VỀ KUBERNETES. HƯỚNG DẪN CÀI ĐẶT CÔNG CỤ KUBECTL TRÊN WINDOWS</b>	Lần ban hành: 1

## 1. CƠ SỞ LÝ THUYẾT

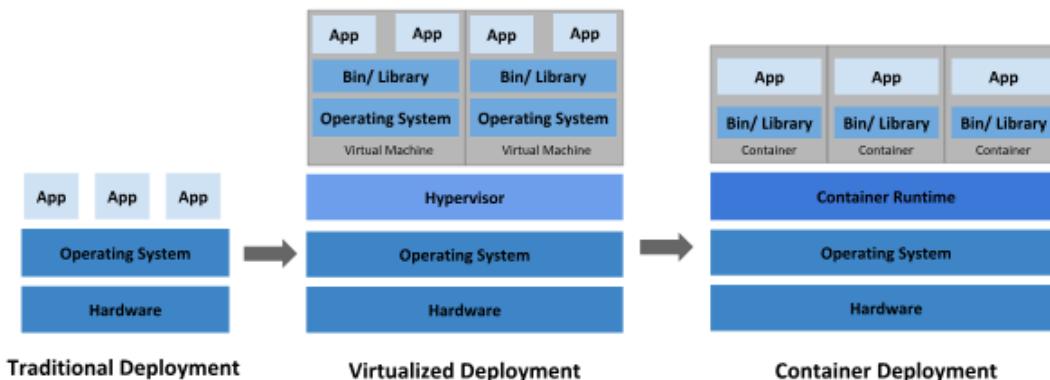
### 1.1 Tổng quan Kubernetes

Kubernetes là một nền tảng nguồn mở, khả chuyen, có thể mở rộng để quản lý các ứng dụng được đóng gói và các service, giúp thuận lợi trong việc cấu hình và tự động hóa việc triển khai ứng dụng. Kubernetes là một hệ sinh thái lớn và phát triển nhanh chóng. Các dịch vụ, sự hỗ trợ và công cụ có sẵn rộng rãi.

Tên gọi Kubernetes có nguồn gốc từ tiếng Hy Lạp, có ý nghĩa là người lái tàu hoặc hoa tiêu. Google mở mã nguồn Kubernetes từ năm 2014. Kubernetes xây dựng dựa trên một thập kỷ rưỡi kinh nghiệm mà Google có được với việc vận hành một khối lượng lớn workload trong thực tế, kết hợp với các ý tưởng và thực tiễn tốt nhất từ cộng đồng.

### 1.2 Lý do ra đời của Kubernetes

Chúng ta hãy xem tại sao Kubernetes rất hữu ích bằng cách quay ngược thời gian.



**Thời đại triển khai theo cách truyền thống:** Ban đầu, các ứng dụng được chạy trên các máy chủ vật lý. Không có cách nào để xác định ranh giới tài nguyên cho các ứng dụng trong máy chủ vật lý và điều này gây ra sự cố phân bổ tài nguyên. Ví dụ, nếu nhiều ứng dụng cùng chạy trên một máy chủ vật lý, có thể có những trường hợp một ứng dụng sẽ chiếm phần lớn tài nguyên hơn và kết quả là các ứng dụng khác sẽ hoạt động kém đi. Một giải pháp cho điều này sẽ là chạy từng ứng dụng trên một máy chủ vật lý khác nhau. Nhưng giải pháp này không tối ưu vì tài nguyên không được sử dụng đúng mức và rất tốn kém cho các tổ chức để có thể duy trì nhiều máy chủ vật lý như vậy.

**Thời đại triển khai ảo hóa:** Như một giải pháp, ảo hóa đã được giới thiệu. Nó cho phép bạn chạy nhiều Máy ảo (VM) trên CPU của một máy chủ vật lý. Ảo hóa cho phép các ứng dụng được cài đặt giữa các VM và cung cấp mức độ bảo mật vì thông tin của một ứng dụng không thể được truy cập tự do bởi một ứng dụng khác.

	<b>VIETTEL AI RACE</b> <b>GIỚI THIỆU VỀ KUBERNETES.</b> <b>HƯỚNG DẪN CÀI ĐẶT CÔNG CỤ</b> <b>KUBECTL TRÊN WINDOWS</b>	Public 192  Lần ban hành: 1
--	---	-----------------------------------

Ảo hóa cho phép sử dụng tốt hơn các tài nguyên trong một máy chủ vật lý và cho phép khả năng mở rộng tốt hơn vì một ứng dụng có thể được thêm hoặc cập nhật dễ dàng, giảm chi phí phần cứng và hơn thế nữa. Với ảo hóa, bạn có thể có một tập hợp các tài nguyên vật lý dưới dạng một cụm các máy ảo sẵn dùng.

Mỗi VM là một máy tính chạy tất cả các thành phần, bao gồm cả hệ điều hành riêng của nó, bên trên phần cứng được ảo hóa.

**Thời đại triển khai Container:** Các container tương tự như VM, nhưng chúng có tính cô lập để chia sẻ Hệ điều hành (HĐH) giữa các ứng dụng. Do đó, container được coi là nhẹ (lightweight). Tương tự như VM, một container có hệ thống tệp (filesystem), CPU, bộ nhớ, process space, v.v. Khi chúng được tách rời khỏi cơ sở hạ tầng bên dưới, chúng có thể khả chuyển (portable) trên cloud hoặc các bản phân phối Hệ điều hành.

Các container đã trở nên phổ biến vì chúng có thêm nhiều lợi ích, chẳng hạn như:

- Tạo mới và triển khai ứng dụng Agile: gia tăng tính dễ dàng và hiệu quả của việc tạo các container image so với việc sử dụng VM image.
- Phát triển, tích hợp và triển khai liên tục: cung cấp khả năng build và triển khai container image thường xuyên và đáng tin cậy với việc rollbacks dễ dàng, nhanh chóng.
- Phân biệt giữa Dev và Ops: tạo các images của các application container tại thời điểm build/release thay vì thời gian triển khai, do đó phân tách các ứng dụng khỏi hạ tầng.
- Khả năng quan sát không chỉ hiển thị thông tin và các metric ở mức Hệ điều hành, mà còn cả application health và các tín hiệu khác.
- Tính nhất quán về môi trường trong suốt quá trình phát triển, testing và trong production: Chạy tương tự trên laptop như trên cloud.
- Tính khả chuyển trên cloud và các bản phân phối HĐH: Chạy trên Ubuntu, RHEL, CoreOS, on-premises, Google Kubernetes Engine và bất kì nơi nào khác.
- Quản lý tập trung ứng dụng: Tăng mức độ trùu tượng từ việc chạy một Hệ điều hành trên phần cứng ảo hóa sang chạy một ứng dụng trên một HĐH bằng logical resources.
- Các micro-services phân tán, elastic: ứng dụng được phân tách thành các phần nhỏ hơn, độc lập và thể được triển khai và quản lý một cách linh hoạt - chứ không phải một app nguyên khối (monolithic).
- Cô lập các tài nguyên: dự đoán hiệu năng ứng dụng
- Sử dụng tài nguyên: hiệu quả

### 1.3 Tại sao lại cần đến Kubernetes

	<b>VIETTEL AI RACE</b> <b>GIỚI THIỆU VỀ KUBERNETES.</b> <b>HƯỚNG DẪN CÀI ĐẶT CÔNG CỤ</b> <b>KUBECTL TRÊN WINDOWS</b>	Public 192 Lần ban hành: 1
--	---	-------------------------------

Các container là một cách tốt để đóng gói và chạy các ứng dụng của bạn. Trong môi trường production, bạn cần quản lý các container chạy các ứng dụng và đảm bảo rằng không có khoảng thời gian downtime. Ví dụ, nếu một container bị tắt đi, một container khác cần phải khởi động lên. Điều này sẽ dễ dàng hơn nếu được xử lý bởi một hệ thống.

Đó là cách Kubernetes đến với chúng ta. Kubernetes cung cấp cho bạn một framework để chạy các hệ phân tán một cách mạnh mẽ. Nó đảm nhiệm việc nhân rộng và chuyển đổi dự phòng cho ứng dụng của bạn, cung cấp các mẫu deployment và hơn thế nữa. Ví dụ, Kubernetes có thể dễ dàng quản lý một triển khai canary cho hệ thống của bạn.

Kubernetes cung cấp cho bạn:

- **Service discovery và cân bằng tải**

Kubernetes có thể expose một container sử dụng DNS hoặc địa chỉ IP của riêng nó. Nếu lượng traffic truy cập đến một container cao, Kubernetes có thể cân bằng tải và phân phối lưu lượng mạng (network traffic) để việc triển khai được ổn định.

- **Điều phối bộ nhớ**

Kubernetes cho phép bạn tự động mount một hệ thống lưu trữ mà bạn chọn, như local storages, public cloud providers, v.v.

- **Tự động rollouts và rollbacks**

Bạn có thể mô tả trạng thái mong muốn cho các container được triển khai dùng Kubernetes và nó có thể thay đổi trạng thái thực tế sang trạng thái mong muốn với tần suất được kiểm soát. Ví dụ, bạn có thể tự động hóa Kubernetes để tạo mới các container cho việc triển khai của bạn, xoá các container hiện có và áp dụng tất cả các resource của chúng vào container mới.

- **Đóng gói tự động**

Bạn cung cấp cho Kubernetes một cluster gồm các node mà nó có thể sử dụng để chạy các tác vụ được đóng gói (containerized task). Bạn cho Kubernetes biết mỗi container cần bao nhiêu CPU và bộ nhớ (RAM). Kubernetes có thể điều phối các container đến các node để tận dụng tốt nhất các resource của bạn.

- **Tự phục hồi**

Kubernetes khởi động lại các containers bị lỗi, thay thế các container, xoá các container không phản hồi lại cấu hình health check do người dùng xác định và không cho các client biết đến chúng cho đến khi chúng sẵn sàng hoạt động.

- **Quản lý cấu hình và bảo mật**

Kubernetes cho phép bạn lưu trữ và quản lý các thông tin nhạy cảm như: password, OAuth token và SSH key. Bạn có thể triển khai và cập nhật lại

	<b>VIETTEL AI RACE</b> <b>GIỚI THIỆU VỀ KUBERNETES.</b> <b>HƯỚNG DẪN CÀI ĐẶT CÔNG CỤ</b> <b>KUBECTL TRÊN WINDOWS</b>	Public 192 Lần ban hành: 1
--	---	-------------------------------

secret và cấu hình ứng dụng mà không cần build lại các container image và không để lộ secret trong cấu hình stack của bạn

## 2. CÀI ĐẶT CÔNG CỤ KUBECTL TRÊN WINDOWS

Trước khi bắt đầu:

Bạn phải sử dụng phiên bản Kubectl có sự khác biệt ít so với cụm của bạn. Ví dụ: máy khách v1.34 có thể giao tiếp với các control planes v1.33, v1.34 và v1.35. Sử dụng phiên bản Kubectl tương thích mới nhất sẽ giúp tránh các sự cố không lường trước được.

### 2.1 Cài đặt Kubectl binary trên Windows ( thông qua tải xuống trực tiếp hoặc curl)

Bạn có hai tùy chọn để cài đặt kubectl trên thiết bị Windows của mình.

#### 2.1.1 Tải xuống trực tiếp:

Tải xuống bản phát hành bản vá 1.34 mới nhất trực tiếp cho kiến trúc cụ thể của bạn bằng cách truy cập trang phát hành Kubernetes. Hãy đảm bảo chọn đúng bản nhị phân cho kiến trúc của bạn (ví dụ: amd64, arm64, v.v.).

#### 2.1.2 Sử dụng curl:

Nếu bạn đã cài đặt curl, hãy sử dụng lệnh này:

curl.exe -LO <https://dl.k8s.io/release/v1.34.0/bin/windows/amd64/kubectl.exe>

### 2.2 Xác thực tệp nhị phân (tùy chọn)

Tải xuống tệp tổng kiểm tra kubectl:

curl.exe -LO <https://dl.k8s.io/v1.34.0/bin/windows/amd64/kubectl.exe.sha256>

Xác thực tệp nhị phân kubectl với tệp tổng kiểm tra:

Sử dụng Dấu nhắc Lệnh để so sánh thủ công đầu ra của CertUtil với tệp tổng kiểm tra đã tải xuống:

```
CertUtil -hashfile kubectl.exe SHA256
type kubectl.exe.sha256
```

Sử dụng PowerShell để tự động xác minh bằng toán tử -eq để có kết quả  
Đúng hoặc Sai:

```
$(Get-FileHash -Algorithm SHA256 .\kubectl.exe).Hash -eq $(Get-Content .\kubectl.exe.sha256)
```



	VIETTEL AI RACE	Public 192
	GIỚI THIỆU VỀ KUBERNETES. HƯỚNG DẪN CÀI ĐẶT CÔNG CỤ KUBECTL TRÊN WINDOWS	Lần ban hành: 1

### 2.2.1 Thêm hoặc thêm thư mục nhị phân kubectl vào biến môi trường PATH của bạn

### 2.2.2 Kiểm tra để đảm bảo phiên bản kubectl giống với phiên bản đã tải xuống

```
kubectl version --client
```

Hoặc sử dụng lệnh này để xem chi tiết phiên bản:

```
kubectl version --client --output=yaml
```