

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM



BÁO CÁO ĐỒ ÁN 1

**Đề tài: Tìm hiểu về Kubernetes và xây
dựng ứng dụng**

GVHD: Thầy Trịnh Công Hoan

Sinh viên thực hiện

Trần Đức Tâm - 19522166

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Contents

I.	Giới thiệu	5
A.	Các ứng dụng Monolithic	5
B.	Nguồn gốc của Kuberetes	6
C.	Tổng quan về Kubernetes	7
D.	Kiến trúc của Kubernetes cluster	8
1.	The control plane - master node	8
2.	Worker node	9
E.	Chạy một ứng dụng trên Kubernetes	9
1.	Kubernetes giữ cho các container luôn vận hành	10
2.	Scaling	11
F.	Lợi ích của việc sử dụng Kubernetes.....	11
1.	Đơn giản hóa quá trình deploy	11
2.	Tối ưu phần cứng	12
3.	Kiểm tra và tự hồi phục	12
II.	Lịch sử phát triển của Kubernetes.....	13
III.	Cộng đồng sử dụng và các công cụ hỗ trợ	25
A.	Các công cụ hỗ trợ	25
B.	Cộng đồng	26
1.	Cộng đồng đóng góp xây dựng hệ thống	29
IV.	Các thành phần cơ bản	30
A.	Pod	30
1.	Giới thiệu	30
2.	Tạo một pod từ mô tả YAML	31
3.	Xem các pod trong hệ thống	32

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

4.	Tạm dừng và xóa một pod.....	32
B.	ReplicaSet.....	32
1.	Giới thiệu	32
2.	Tạo một ReplicaSet từ mô tả YAML.....	34
3.	Xem các ReplicaSet trong hệ thống.....	35
4.	Xem các Pod trong ReplicaSet đó.....	35
C.	Services	36
1.	Giới thiệu	36
2.	Tạo một Service từ mô tả YAML.....	39
3.	Các loại Services.....	40
D.	Secrets.....	43
1.	Giới thiệu	43
2.	Opaque secrets	45
E.	Deployment.....	46
1.	Giới thiệu	46
2.	Tạo một Deployment.....	47
3.	Cập nhật một Deployment	47
V.	Hướng dẫn chạy một ứng dụng bằng Kubernetes	48
A.	Yêu cầu	48
B.	Tạo và chạy một container image.....	49
1.	Tạo một ứng dụng bằng Node.js	49
2.	Tạo file Dockerfile cho image	49
3.	Build Docker image	50
4.	Push image lên DockerHub.....	51
5.	Setup cluster Kubernetes	51

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

6. Chạy ứng dụng đầu tiên trên Kubernetes	54
VI. Xây dựng ứng dụng minh hoạ	55
A. Kiến trúc ứng dụng.....	55
1. Chi tiết về một vài services	57
2. Các message được giao tiếp giữa các services	58
B. Routing	58
C. Các màn hình ứng dụng	59
D. Đường dẫn đến ứng dụng.....	65

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

I. Giới thiệu

A. Các ứng dụng Monolithic

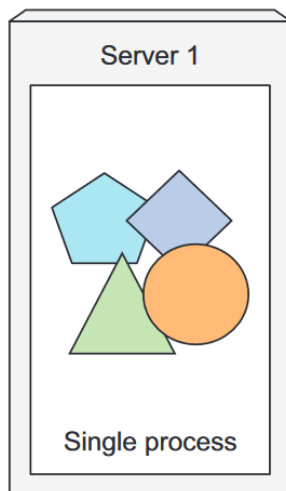
Các ứng dụng monolithic bao gồm các components được kết nối chặt chẽ với nhau, và chúng được phát triển, vận hành và quản lý như một khối thống nhất. Điều này dẫn đến các vấn đề sau:

- Cần sửa chữa ứng dụng, chúng ta cần redeploy lại cả hệ thống
- Yêu cầu các server ứng dụng mạnh mẽ
- Khó có khả năng mở rộng

Vì lẽ đó, chúng ta đã đi đến giải pháp tách các ứng dụng monolithic thành các components có thể hoạt động độc lập gọi là microservices. Mỗi microservices chạy trên một tiến trình độc lập, và chúng giao tiếp với nhau thông qua interface. Đồng thời, mỗi microservice có thể được phát triển, deploy và vận hành một cách độc lập.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Monolithic application



Microservices-based application

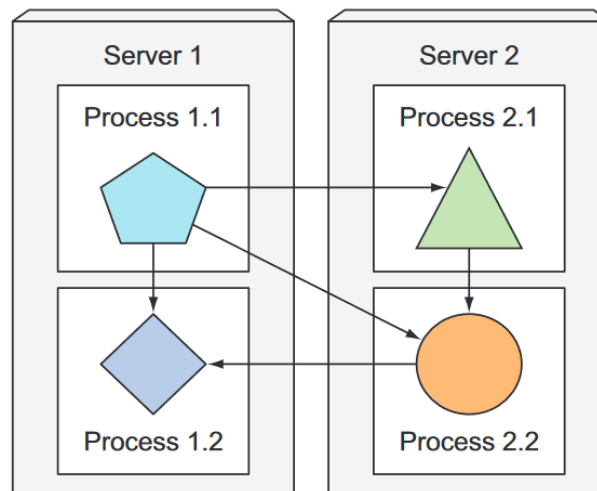


Figure 1.1 Components inside a monolithic application vs. standalone microservices

B. Nguồn gốc của Kubernetes

Dù cho các lợi ích đáng kể đối với việc xây dựng theo kiến trúc Microservices, chúng ta dễ dàng thấy được sự khó khăn trong việc quản lý các component (deployable application components) và trong việc mở rộng hệ thống. Google đã nhận thấy các công ty cần một cách tốt hơn để quản lý và deploy các components tốt hơn.

Qua nhiều năm, Google đã phát triển một hệ thống gọi là Borg (sau này là Omega) giúp cho các nhà phát triển (dev) và admin hệ thống quản lý các hàng ngàn ứng dụng và service. Ngoài việc đơn giản hóa quá trình deploy và phát triển, nó còn giúp tối ưu hạ tầng cơ sở (infrastructure). Năm 2014, Google đã giới thiệu Kubernetes dựa trên kinh nghiệm phát triển Borg, Omega.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

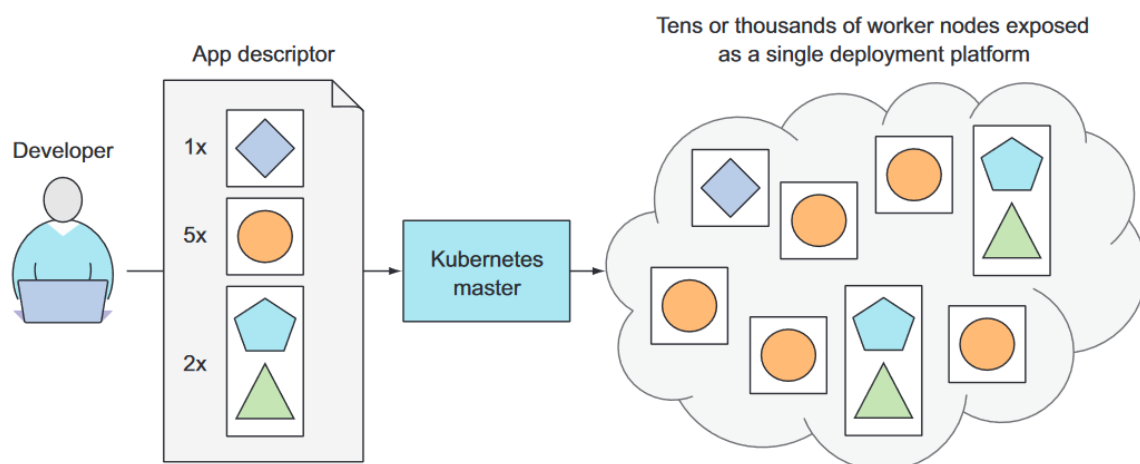
C. Tổng quan về Kubernetes

Kubernetes là một hệ thống phần mềm giúp chúng ta dễ dàng deploy và quản lý các **ứng dụng chạy trong container** (containerized application).

Kubernetes có thể chạy các ứng dụng trong container mà không cần phải biết bất kỳ thông tin gì về chúng, đồng thời ta không phải deploy thủ công những ứng dụng này vào các host. Bởi vì các ứng dụng này được chạy trong các container khác nhau, chúng hoạt động độc lập nên ta có thể chạy nhiều ứng dụng khác nhau trên 1 phần cứng.

Kubernetes cho phép bạn chạy ứng dụng trên hàng ngàn máy tính như đó chỉ là một máy tính duy nhất. Nó abstract đi hệ thống cơ sở hạ tầng bên dưới (infrastructure) giúp đơn giản hóa quá trình phát triển, deploy và maintain.

Deploy một ứng dụng trên Kubernetes luôn giống nhau cho dù cluster đó có 1-2 node hoặc 1000 node.



Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Đây là góc nhìn đơn giản nhất về Kubernetes. Hệ thống bao gồm 1 master node và nhiều work node. Khi dev deploy hệ thống, Kubernetes sẽ deploy chúng vào 1 cluster gồm các worker node. Node nào sẽ chạy components thì chúng ta (và cả admin hệ thống) không cần phải quan tâm.

D. Kiến trúc của Kubernetes cluster

Ở mức độ phần cứng, Kubernetes cluster được cấu thành từ nhiều **node**, chúng ta có thể phân loại thành 2 loại node:

- master node, có Kubernetes Control Plane để quản lý cả hệ thống Kubernetes
- worker node, để chạy ứng dụng được deploy.

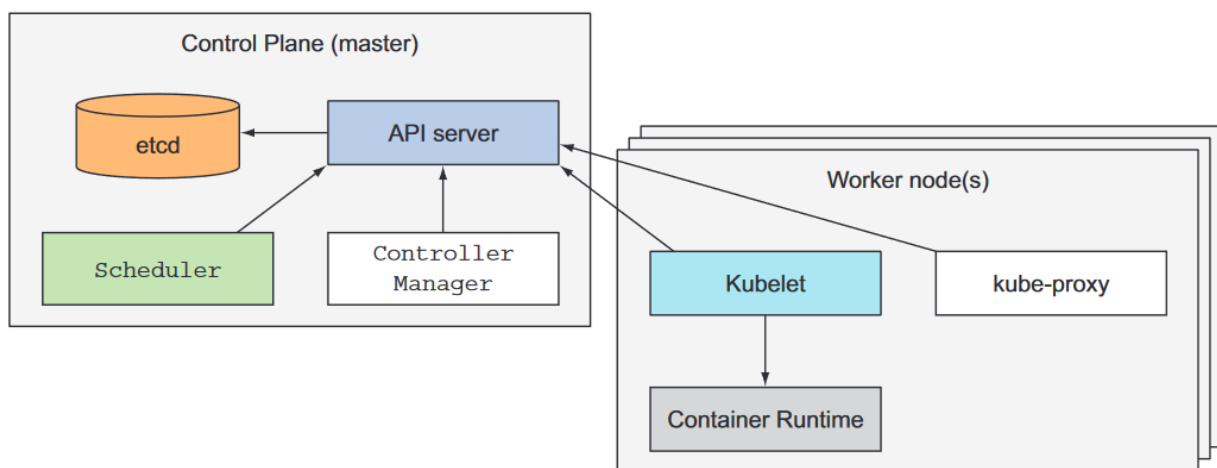


Figure 1 Các thành phần cơ bản

1. The control plane - master node

Giữ nhiệm vụ quản lý cluster mà giữ cho cluster hoạt động. Nó bao gồm các component:

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

- **Kubernetes API Server:** giúp ta giao tiếp với các component của control plane khác
- **Scheduler:** lập lịch cho ứng dụng (giao component cho cho một worker node)
- **Controller Manager:** Xử lí các chức năng ở mức cluster (cluster-level function) như quản lý các worker node, tái tạo các component, xử lý các node lỗi,...
- **etcd:** dữ liệu về các cài đặt cluster và trạng thái (state) của cluster

The control plane nắm giữ và quản lý các trạng thái của cluster, nhưng chúng không chạy các ứng dụng. Điều đó được thực hiện bởi worker node.

2. Worker node

Worker node là cỗ máy (machine) chạy các ứng dụng trong container. Các tác vụ vận hành, quản lý và cung cấp các dịch vụ cho ứng dụng được thực thi bởi các component sau:

- Docker, rkt hoặc là bất kì container runtime khác
- kubelet, giao tiếp với cluster và quản lý các container trên node
- Kubernetes Service Proxy (kube-proxy), cân bằng tải giữa các components

E. Chạy một ứng dụng trên Kubernetes

Muốn chạy một ứng dụng trên Kubernetes, ta phải đóng gói nó lại thành một image container, push image đó lên image registry, và đăng miêu tả (description) về image đó lên Kubernetes API server.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

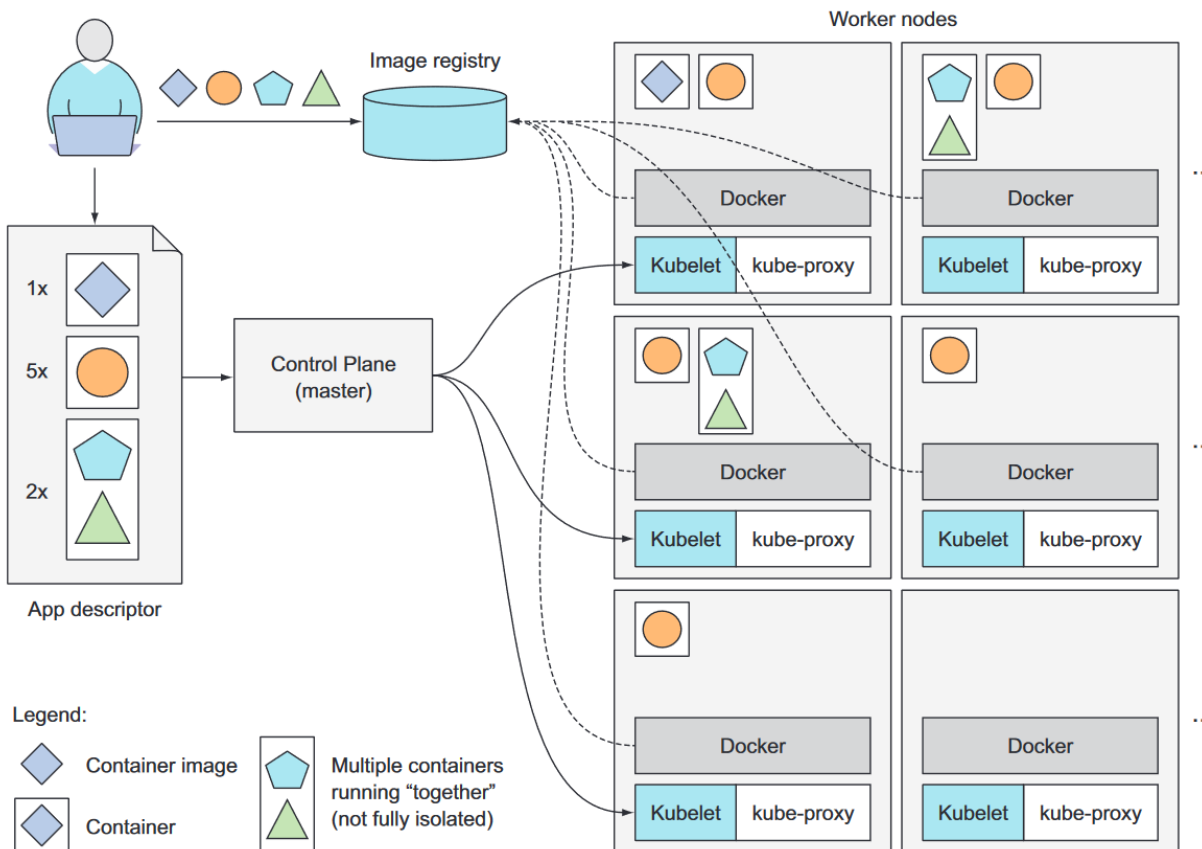


Figure 1.10 A basic overview of the Kubernetes architecture and an application running on top of it

Khi API server xử lý miêu tả về ứng dụng của bạn, Scheduler sẽ lập lịch tạo ra các container trên các work node dựa trên nguồn tài nguyên hiện có của bạn. ****Kubelet**** trên các node sẽ giao tiếp với với Kubernetes API Server để tải các image và chạy các image đó.

1. Kubernetes giữ cho các container luôn vận hành

Khi ứng dụng được đưa vào hoạt động (deployed), Kubernetes sẽ liên tục đảm bảo rằng trạng thái của ứng dụng đúng với miêu tả mà bạn cung cấp. Ví dụ như chúng ta luôn

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

muốn có 5 instance của web server hoạt động, Kubernetes sẽ luôn đảm bảo 5 instance. Nếu 1 instance ngừng hoạt động hay không phản hồi, Kubernetes sẽ tự động restart nó

Tương tự, nếu 1 node chết đi, Kubernetes sẽ chọn 1 node khác và chạy lại tất cả container của nó.

2. Scaling

Khi ứng dụng đang chạy, bạn có thể quyết định tăng giảm số lượng copies tùy thích , Kubernetes sẽ cập nhật theo yêu cầu. Thậm chí, chúng ta có thể để Kubernetes điều này, dựa vào các thông số thực tế (real-times metrics) như CPU load, tiêu thụ bộ nhớ, lượng truy vấn từng giây,...

F. Lợi ích của việc sử dụng Kubernetes

Nếu bạn deploy Kubernetes trên tất cả server, đội ngũ vận hành (Ops team) sẽ không cần xử lý việc deploy các ứng dụng. Bởi vì các ứng dụng chạy trong container (containerized application) đã được bổ sung các tất cả những gì nó cần để chạy, admin hệ thống không cần phải cài đặt thứ gì để deploy và chạy ứng dụng.

1. Đơn giản hóa quá trình deploy

Bởi vì Kubernetes cung cấp tất cả worker node như là một **deployment platform**, developer có thể bắt đầu deploy ứng dụng mà không cần biết gì về các server tạo nên cluster.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Một vài trường hợp đặc biệt, như khi ta muốn ứng dụng của mình được chạy trên một node sử dụng SSD thay vì HDD.

- Nếu không sử dụng k8s → admin hệ thống sẽ phải chọn 1 node cụ thể và deploy ứng dụng ở đó
- Khi sử dụng k8s → ta có thể nói cho k8s chọn 1 node sử dụng SSD và k8s sẽ tự động vận hành theo yêu cầu.

2. Tối ưu phần cứng

Bằng việc set up k8s trên server, chúng ta sẽ decouple ứng dụng khỏi kiến trúc. Khi ta yêu cầu k8s chạy ứng dụng, k8s sẽ chọn node phù hợp nhất dựa trên **mô tả tài nguyên yêu cầu** (description of the application's resource requirements) và **nguồn tài nguyên hiện có** trên các node.

Bằng việc sử dụng các container và không cố định các node trên cluster, ta cho phép ứng dụng tự do vận chuyển trên cluster. các component khác nhau sẽ được sắp xếp sao cho tối ưu phần cứng nhất trên các node.

3. Kiểm tra và tự hồi phục

Ứng dụng tự do vận chuyển trên cluster là một yếu tố đáng giá trong trường hợp các server của chúng ta bị hỏng (server failure). Nếu quy mô của cluster chúng ta tăng lên → các component sẽ vận hành thất bại thường xuyên hơn.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Kubernetes sẽ quản lý các node và các components đang chạy trên đó, và sẽ tự động lập lịch các components đó trên các node khác nếu server bị lỗi. Điều này cho phép đội vận hành tập trung vào việc xử lý vấn đề node lỗi thay vì phải tìm cách tích hợp các components cũ vào ứng dụng.

II. Lịch sử phát triển của Kubernetes

2003-2004: Sự ra đời của Hệ thống Borg	<ul style="list-style-type: none"> Google đã giới thiệu Hệ thống Borg vào khoảng năm 2003-2004. Nó bắt đầu như một dự án quy mô nhỏ, với khoảng 3-4 người ban đầu cộng tác với một phiên bản mới của công cụ tìm kiếm mới của Google. Borg là một hệ thống quản lý cụm nội bộ quy mô lớn, chạy hàng trăm nghìn công việc, từ nhiều nghìn ứng dụng khác nhau, trên nhiều cụm, mỗi cụm có tới hàng chục nghìn máy.
2013: Từ Borg đến Omega	<ul style="list-style-type: none"> Sau Borg, Google đã giới thiệu hệ thống quản lý cụm Omega, một công cụ lập lịch linh hoạt, có thể mở rộng cho các cụm máy tính lớn.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

2014: Google giới thiệu Kubernetes	<ul style="list-style-type: none"> ▪ Giữa năm 2014: Google giới thiệu Kubernetes như một phiên bản mã nguồn mở của Borg ▪ Ngày 7 tháng 6: Bản phát hành đầu tiên - cam kết github đầu tiên cho Kubernetes ▪ Ngày 10 tháng 7: Microsoft, RedHat, IBM, Docker tham gia cộng đồng Kubernetes.
2015: Năm của Kube v1.0 & CNCF	<ul style="list-style-type: none"> ▪ Ngày 21 tháng 7: Kubernetes v1.0 được phát hành. Cùng với việc phát hành, Google đã hợp tác với Linux Foundation để thành lập Cloud Native Computing Foundation (CNCF). CNCF nhằm mục đích xây dựng hệ sinh thái bền vững và thúc đẩy cộng đồng xung quanh một loạt các dự án chất lượng cao sắp xếp các thùng chứa như một phần của kiến trúc microservices. ▪ Ngày 3 tháng 11: Hệ sinh thái Kubernetes tiếp tục phát triển ! Các công ty đã tham gia: Deis, OpenShift, Huawei và Gindor. ▪ Ngày 9 tháng 11: Kubernetes 1.1 mang đến những nâng cấp lớn về hiệu suất, cải tiến công cụ và các tính năng mới giúp ứng dụng dễ dàng hơn trong việc xây dựng và triển khai.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

	<ul style="list-style-type: none"> ▪ Ngày 9-11 tháng 11: KubeCon 2015 là hội nghị Kubernetes cộng đồng khai mạc đầu tiên ở San Francisco. Mục tiêu của nó là cung cấp các cuộc nói chuyện chuyên môn kỹ thuật được thiết kế để khơi dậy sự sáng tạo và thúc đẩy giáo dục Kubernetes.
2016: Năm Kubernetes trở nên chính thống!	<ul style="list-style-type: none"> ▪ Ngày 23 tháng 2: Bản phát hành đầu tiên của Helm, trình quản lý gói của Kubernetes. ▪ Ngày 24 tháng 2: KubeCon EU 2016 là hội nghị Kubernetes châu Âu khai mạc với gần 500 người tham dự, sau sự ra mắt của Mỹ vào tháng 11 năm 2015. ▪ Ngày 16 tháng 3: Kubernetes 1.2 được phát hành - Các cải tiến bao gồm mở rộng quy mô, triển khai ứng dụng đơn giản hóa và quản lý cụm tự động. ▪ Ngày 1 tháng 7: Kubernetes 1.3: Kết nối khối lượng công việc dành cho doanh nghiệp và nền tảng đám mây. v1.3 giới thiệu Rktnetes 1.0 và đối tượng alpha 'PetSet' mới, đồng thời giúp bạn có thể khám phá các dịch vụ đang chạy trong nhiều cụm.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

	<ul style="list-style-type: none"> ▪ Ngày 11 tháng 7: Phát hành chính thức Minikube: một công cụ giúp dễ dàng chạy Kubernetes cục bộ. ▪ Ngày 8 tháng 9: Giới thiệu Kops, một dự án Kubernetes chính thức để quản lý các cụm Kubernetes cấp sản xuất. ▪ Ngày 19 tháng 9: Monzo phát hành một nghiên cứu điển hình về cách họ sử dụng Kubernetes để xây dựng hệ thống ngân hàng từ đầu. ▪ 26 tháng 9: Kubernetes 1.4 giới thiệu một công cụ mới kubeadm, giúp cải thiện khả năng cài đặt của Kubernetes. Bản phát hành này cung cấp thiết lập dễ dàng hơn, hỗ trợ ứng dụng trạng thái với Helm tích hợp và các tính năng liên kết giữa các cụm mới. ▪ Ngày 29 tháng 9: Pokemon GO! Nghiên cứu điển hình Kubernetes được phát hành! Pokémon GO là lần triển khai Kubernetes lớn nhất trên Google Container Engine từ trước đến nay. May mắn thay, những người sáng tạo đã phát hành một nghiên cứu điển hình về cách họ đã làm điều đó.
--	---

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

	<ul style="list-style-type: none"> ▪ Ngày 8-9 tháng 11: CloudNativeCon + KubeCon 2016 Bắc Mỹ, tại Seattle. Hơn 1.000 người dùng cuối, những người đóng góp và nhà phát triển hàng đầu trên khắp thế giới đã cùng nhau trao đổi kiến thức về Fluentd, Kubernetes, Prometheus, OpenTracing và các công nghệ gốc đám mây khác. ▪ Ngày 7 tháng 12: Khám phá tính năng nút cho Kubernetes Arrives - Gói này cho phép khám phá tính năng nút cho Kubernetes. Nó phát hiện các tính năng phần cứng có sẵn trên mỗi nút trong một cụm Kubernetes và quảng cáo các tính năng đó bằng cách sử dụng nhãn nút. ▪ Ngày 21 tháng 12: Kubernetes 1.5 - Hỗ trợ Windows Server đến với Kubernetes. Các tính năng mới bao gồm các ứng dụng đa dạng được chứa trong container, hỗ trợ container máy chủ Windows và container hyper-V, hệ sinh thái mở rộng của các ứng dụng, phạm vi bảo hiểm cho các trung tâm dữ liệu không đồng nhất, v.v. ▪ Ngày 23 tháng 12: Kubernetes hỗ trợ OpenAPI, cho phép các nhà cung cấp API xác định các hoạt động và mô hình của họ, đồng
--	--

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

	thời cho phép các nhà phát triển tự động hóa các công cụ của họ.
2017: Năm áp dụng và hỗ trợ doanh nghiệp	<p>Ngày 28 tháng 3: Kubernetes 1.6 là bản phát hành ổn định. Cập nhật cụ thể: vdv3 được bật theo mặc định, xóa phụ thuộc trực tiếp vào container runtime duy nhất, RBAC trong phiên bản beta, cung cấp tự động các đối tượng StorageClass.</p> <p>Ngày 29-30 tháng 3: CloudNativeCon + KubeCon Europe, Berlin. 1.500 người dùng cuối, những người đóng góp và nhà phát triển hàng đầu từ khắp nơi trên thế giới đã tham dự để trao đổi kiến thức về Cloud Native.</p> <p>Ngày 24 tháng 5: Google và IBM công bố Istio, một công nghệ mở cung cấp cách kết nối liền mạch, quản lý và bảo mật mạng của các dịch vụ vi mô khác nhau - bất kể nền tảng, nguồn hoặc nhà cung cấp.</p> <p>Ngày 30 tháng 6: Kubernetes 1.7: Tiêu chuẩn điều phối container thêm lưu trữ cục bộ, mã hóa bí mật và khả năng mở rộng, tức là: tổng hợp API, tài nguyên bên thứ ba, giao diện container runtime, v.v.</p>

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

	<p>Ngày 16 tháng 8: Github chạy trên Kubernetes: tất cả các yêu cầu web và API được phục vụ bởi các container chạy trong các cụm Kubernetes được triển khai trên metal cloud.</p> <p>Ngày 31 tháng 8: Kelsey Hightower phát hành Kubernetes the Hard Way. Kubernetes Con đường cứng được tối ưu hóa cho việc học, có nghĩa là thực hiện một lộ trình dài để đảm bảo bạn hiểu từng nhiệm vụ cần thiết để khởi động một cụm Kubernetes.</p> <p>Ngày 11 tháng 9: Cloud Native Computing Foundation công bố các nhà cung cấp dịch vụ được chứng nhận Kubernetes đầu tiên, nhóm sáng lập của hơn 22 nhà cung cấp dịch vụ được chứng nhận Kubernetes (KCSP), các tổ chức đủ điều kiện có kinh nghiệm sâu sắc giúp các doanh nghiệp áp dụng thành công Kubernetes.</p> <p>Ngày 13 tháng 9: Oracle gia nhập Cloud Native Computing Foundation với tư cách là thành viên bạch kim. Oracle mở nguồn trình cài đặt Kubernetes</p>
--	--

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

	<p>cho Cơ sở hạ tầng đám mây Oracle và phát hành Kubernetes trên Oracle Linux.</p> <p>Ngày 29 tháng 9: Kubernetes 1.8: Bản phát hành này là một cột mốc quan trọng đối với trình cấp quyền kiểm soát truy cập dựa trên vai trò (RBAC), một cơ chế để kiểm soát quyền truy cập vào API Kubernetes, sẽ là khối xây dựng cơ bản để bảo mật các cụm Kubernetes.</p> <p>Tháng 10: Docker hoàn toàn nắm giữ Kubernetes - các nhà phát triển và nhà điều hành có thể xây dựng ứng dụng với Docker và kiểm tra và triển khai chúng một cách liền mạch bằng cả Docker Swarm và Kubernetes.</p> <p>Ngày 17 tháng 10: Nền tảng Docker và Moby Projekt thêm Kubernetes: khách hàng và nhà phát triển có tùy chọn sử dụng cả Kubernetes và Swarm để sắp xếp khối lượng container workload.</p> <p>Ngày 24 tháng 10: Microsoft giới thiệu bản xem trước của AKS - AKS có mặt phẳng điều khiển do Azure lưu trữ, nâng cấp tự động, tự phục hồi, mở rộng dễ dàng và trải nghiệm người dùng đơn giản</p>
--	---

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

	<p>cho các nhà phát triển và vận hành cụm. Khách hàng có được Kubernetes mã nguồn mở mà không cần chi phí vận hành.</p> <p>Ngày 29 tháng 11: Amazon công bố dịch vụ Elastic Container Service cho Kubernetes. Triển khai, quản lý và mở rộng quy mô các ứng dụng được container hóa bằng Kubernetes trên AWS.</p> <p>Ngày 6-8 tháng 12: KubeCon + CloudNativeCon Bắc Mỹ, Austin - tập hợp hơn 4.100 người dùng cuối, nhà cung cấp, cộng tác viên và nhà phát triển hàng đầu từ khắp nơi trên thế giới.</p> <p>Ngày 15 tháng 12: Kubernetes 1.9 phát hành: Ứng dụng workload GA và Hệ sinh thái mở rộng. Các tính năng mới: tính khả dụng chung của ứng dụng/v1 Workload API, hỗ trợ windows (beta), cải tiến bộ nhớ và hơn thế nữa.</p> <p>Ngày 21 tháng 12: Giới thiệu Kubeflow - Machine Learning Stack có thể thay đổi, di động, có thể mở rộng được xây dựng cho Kubernetes.</p>
--	--

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

2018:

- Ngày 2 tháng 3: Phiên bản Beta đầu tiên của Kubernetes 1.10 được công bố. Người dùng có thể kiểm tra các phiên bản sẵn sàng sản xuất của Kubelet TLS Bootstrapping, tổng hợp API và các chỉ số lưu trữ chi tiết hơn.
- Ngày 1 tháng 5: Google ra mắt Kubernetes Podcast, được lưu trữ bởi Craig Box.
- Ngày 2-4 tháng 5: KubeCon + CloudNativeCon Europe 2018, được tổ chức tại Copenhagen. Hơn 4.300 nhà phát triển đã tập hợp, bao gồm cả Risingstack tại gian hàng của Node Foundation
- Ngày 2 tháng 5: DigitalOcean đi sâu vào Kubernetes, công bố một sản phẩm Kubernetes được lưu trữ mới. DigitalOcean Kubernetes sẽ cung cấp nền tảng điều phối và quản lý container dưới dạng dịch vụ miễn phí trên các tùy chọn lưu trữ và tính toán đám mây hiện có của nó.
- Ngày 4 tháng 5: Kubeflow 0.1 được công bố, cung cấp một bộ gói tối thiểu để bắt đầu phát triển, đào tạo và triển khai ML.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

	<ul style="list-style-type: none"> • Ngày 21 tháng 5: Google Kubernetes Engine 1.10 nói chung đã có sẵn và sẵn sàng cho doanh nghiệp, có Đám mây riêng ảo dùng chung, Đĩa liên tục theo khu vực & cụm khu vực, Node Auto-Repair GA và Custom Horizontal Pod Autoscaler để tự động hóa nhiều hơn. • Ngày 24 tháng 5: Tích hợp Kubernetes Containerd đi đến GA. Containerd 1.1 hoạt động với Kubernetes 1.10 trở lên và hỗ trợ tất cả các tính năng của Kubernetes. Phạm vi thử nghiệm của tích hợp containerd trên Google Cloud Platform trong cơ sở hạ tầng thử nghiệm Kubernetes hiện tương đương với tích hợp Docker. • Ngày 5 tháng 6: Amazon EKS nói chung trở nên khả dụng. Amazon EKS đơn giản hóa quá trình xây dựng, bảo mật, vận hành và duy trì các cụm Kubernetes, đồng thời mang lại lợi ích của điện toán dựa trên container cho các tổ chức muốn tập trung vào việc xây dựng ứng dụng thay vì thiết lập một cụm Kubernetes từ đầu.
--	---

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

	<ul style="list-style-type: none"> Ngày 13 tháng 6: Dịch vụ Azure Kubernetes (AKS) thường khả dụng. Với AKS, người dùng có thể triển khai và quản lý các ứng dụng Kubernetes sản xuất của họ với sự tự tin rằng các kỹ sư của Azure đang cung cấp khả năng giám sát, hoạt động và hỗ trợ liên tục cho các cụm Kubernetes được quản lý đầy đủ của khách hàng của chúng tôi. Ngày 27 tháng 6: Kubernetes 1.11 đã công bố: Cân bằng tải trong cụm và Plugin CoreDNS nâng cấp lên mức khả dụng chung. Phiên bản mới nhất này hoàn thiện các tính năng chính trong mạng, mở ra hai tính năng chính từ SIG-API Machinery và SIG-Node để thử nghiệm beta và tiếp tục nâng cao các tính năng lưu trữ vốn là tiêu điểm của hai bản phát hành trước đây.
--	--

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

III. Cộng đồng sử dụng và các công cụ hỗ trợ

A. Các công cụ hỗ trợ

Kubernetes Dashboard	<ul style="list-style-type: none"> • Công cụ giúp quản lý, deploy, giám sát các ứng dụng và tài nguyên trong Kubernetes trong web UI. • Cho phép bạn mô hình hóa Kubernetes một cách trực quan hơn. • Link: https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/
Kube-monkey	<ul style="list-style-type: none"> • Công cụ giúp phát triển hệ thống ổn định hơn bằng việc ngẫu nhiên tạo ra các sự cố trong hệ thống (Ví dụ như xóa một pod ngẫu nhiên) • Link: https://github.com/asobti/kube-monkey
Kube-hunter	<ul style="list-style-type: none"> • Công cụ giúp kiểm tra tính bảo mật của ứng dụng và môi trường hoạt động qua việc pen-testing cluster và các node. • Link: https://github.com/aquasecurity/kube-hunter

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Kube-bench	<ul style="list-style-type: none"> • Công cụ giúp bạn stress-test hệ thống. • Link: https://github.com/cloud-bulldozer/kube-burner
Prometheus	<ul style="list-style-type: none"> • Là một công cụ đơn giản và hiệu quả giúp cho bạn quản lý hệ thống trên Kubernetes, đồng thời việc thiết lập cài đặt tương đối đơn giản • Link: https://prometheus.io/

B. Cộng đồng

Có hơn 24 ngàn doanh nghiệp sử dụng Kubernetes và càng có nhiều doanh nghiệp đang dần dịch chuyển sang công nghệ này. Hơn thế, những người sử dụng Kubernetes đều nhận thấy được ích lợi rõ rệt của nó đối với các quá trình xây dựng, phát triển và quản lý hệ thống.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

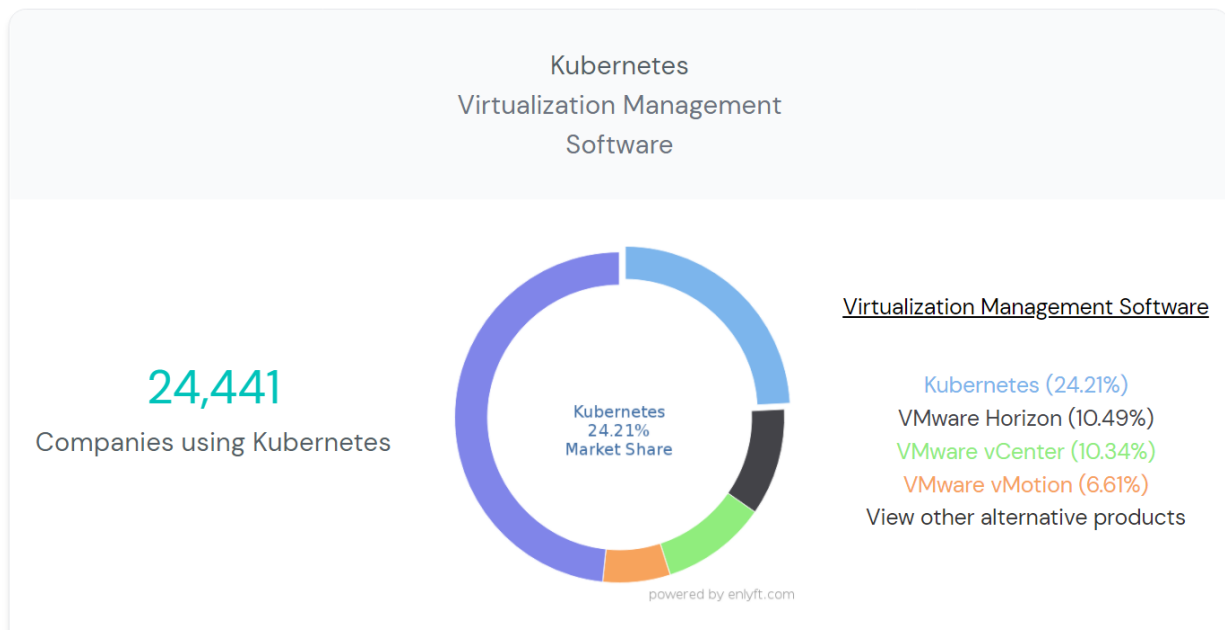


Figure 2 : Số lượng doanh nghiệp sử dụng và marketshare

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

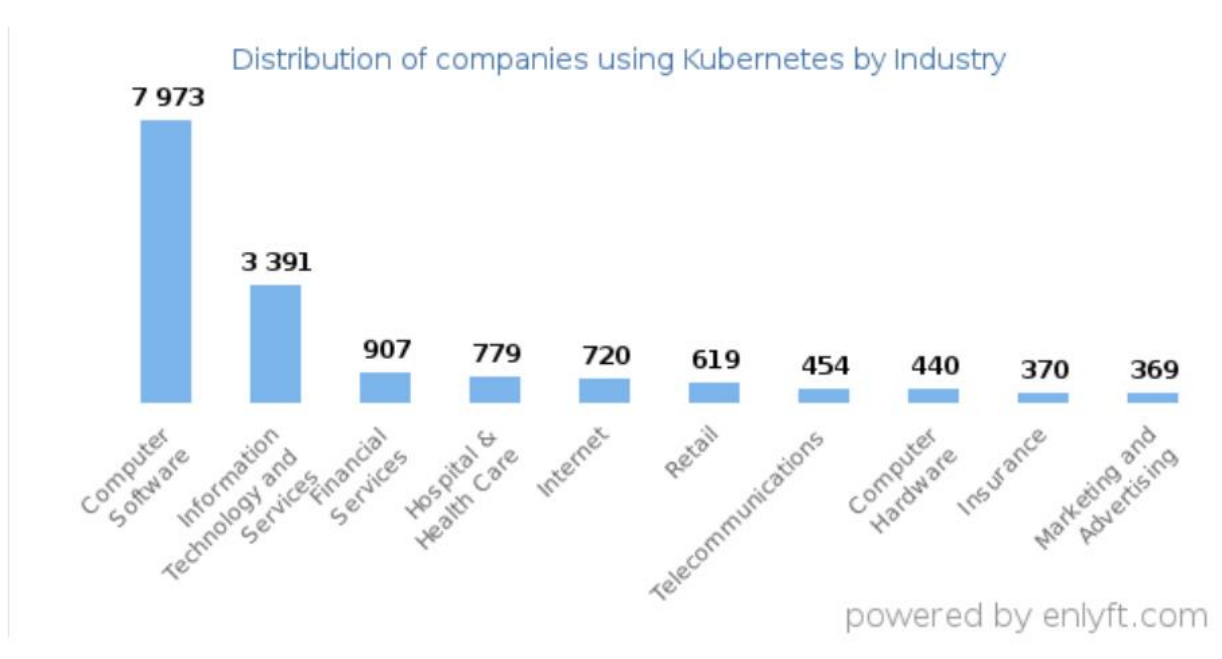


Figure 3: Phân phối doanh nghiệp sử dụng theo chuyên môn

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

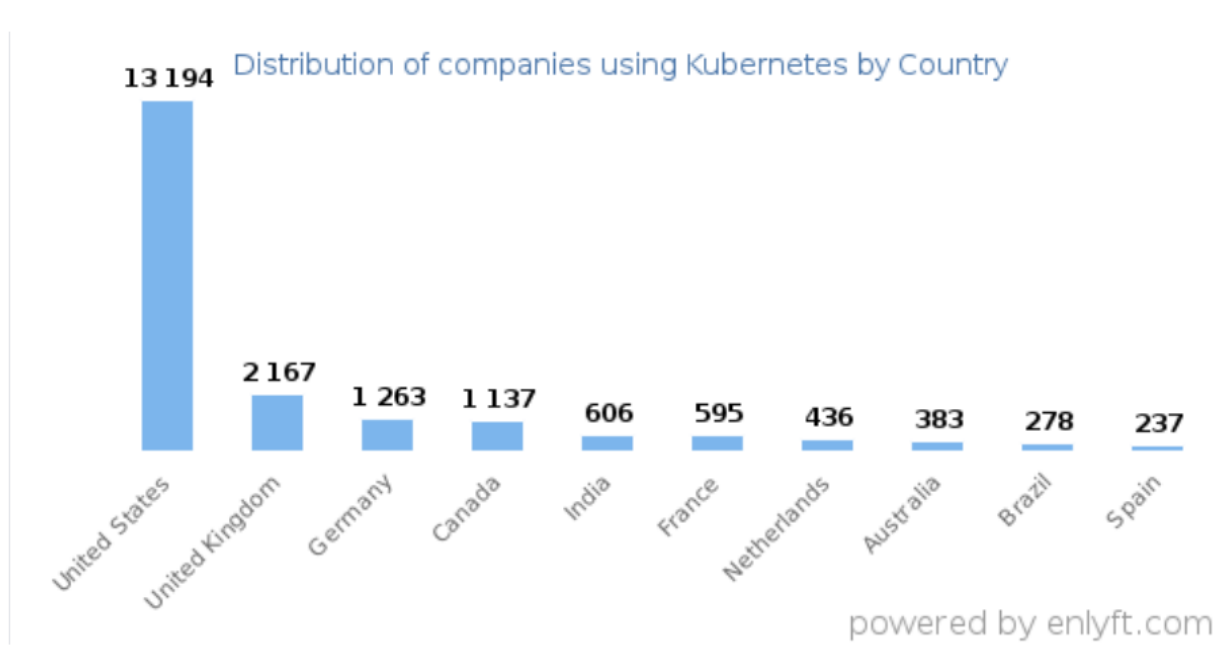


Figure 4: Phân phối doanh nghiệp sử dụng theo đất nước

1. Cộng đồng đóng góp xây dựng hệ thống

- Hơn 52 ngàn người đóng góp cho đến thời điểm hiện tại
- 2.8 triệu: Là tổng số lượng các contribution cho Kubernetes. Trong đó Google đóng góp hơn 912.058, theo sau là Red Hat (339,904 contributions) và VMware (199,848 contributions)
- 4,162: Tổng số công ty đã có ít nhất 1 contribution cho Kubernetes.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

IV. Các thành phần cơ bản

A. Pod

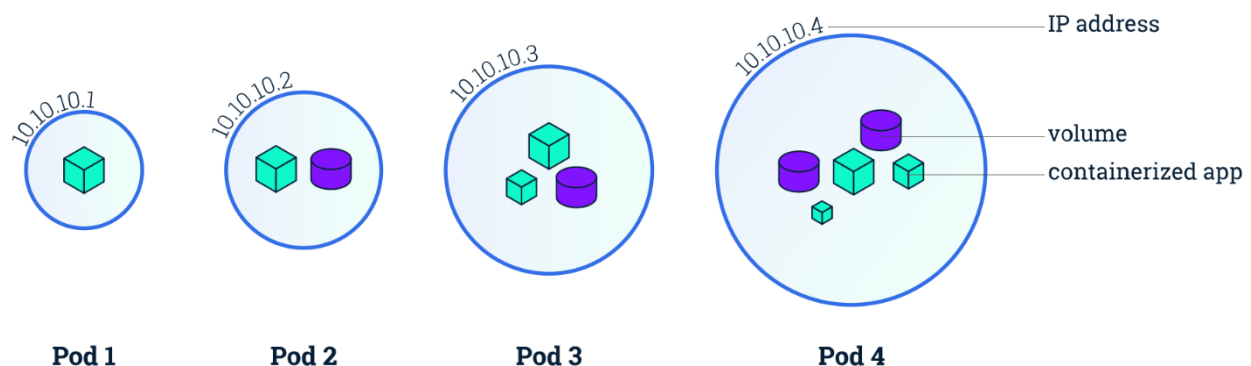
1. Giới thiệu

Một Pod là một khái niệm trừu tượng của Kubernetes, đại diện cho một nhóm gồm một hoặc nhiều ứng dụng containers (ví dụ như Docker hoặc rkt) và một số tài nguyên được chia sẻ cho các containers đó. Những tài nguyên đó bao gồm:

Dữ liệu được chia sẻ

Networking

Thông tin về cách tạo ra container đó, port,...



Pod là thành phần đơn vị (nhỏ nhất) để Kubernetes thực hiện việc nhân bản (replication), có nghĩa là khi cần thiết thì Kubernetes có thể cấu hình để triển khai nhân bản ra nhiều pod có chức năng giống nhau để tránh quá tải, thậm chí nó vẫn tạo ra

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

nhiều bản copy của pod khi không quá tải nhằm phòng lỗi. Pod cũng là đơn vị nhỏ nhất trong Kubernetes.

2. Tạo một pod từ mô tả YAML

Pod thường được khởi tạo bằng việc sử dụng các file YAML hoặc JSON, tuy nhiên, ta cũng có thể sử dụng những cách đơn giản hơn như sử dụng lên **kubectl run**, tuy nhiên điều đó không đem lại nhiều khả năng tùy chỉnh. Đồng thời, sử dụng file để khởi tạo cho phép chúng ta lưu trữ chúng ở những hệ thống quản lý phiên bản (version control system).

```
apiVersion: v1
kind: Pod
metadata:
  name: demo
  labels:
    name: demo
spec:
  containers:
    - name: demo
      image: luksa/kubia
      ports:
        - containerPort: 8080
          protocol: TCP
```

Figure 5: demo.yaml

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

```
kubectl apply -f demo.yaml
```

3. Xem các pod trong hệ thống

```
$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
kubia-manual  1/1     Running   0           32s
kubia-zxzij   1/1     Running   0           1d
```

4. Tạm dừng và xóa một pod

Nếu xóa một pod, bạn sẽ kết thúc tất cả những container đã chạy trong pod. Kubernetes sẽ gửi tín hiệu SIGTERM đến process và đợi một khoảng thời gian nhất định (30 giây) để process tự kết thúc. Nếu thời gian mặc định bị vượt quá, process sẽ bị kill bằng SIGKILL

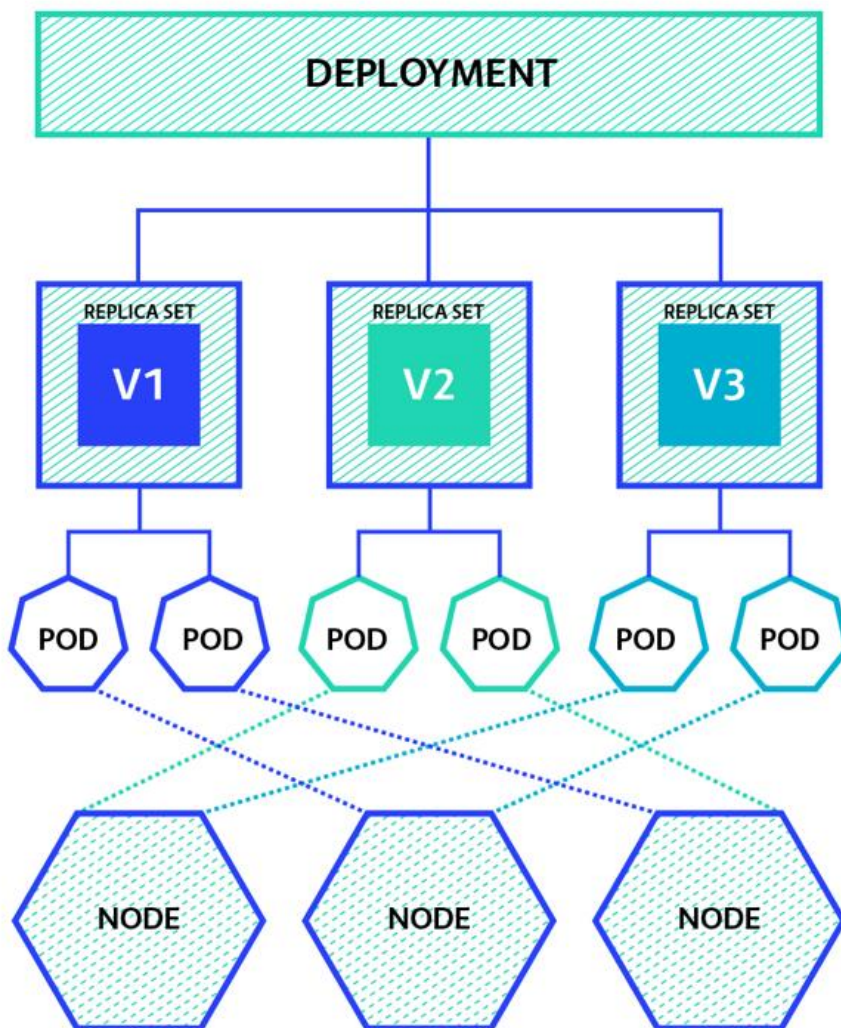
```
$ kubectl delete po kubia-gpu
pod "kubia-gpu" deleted
```

B. ReplicaSet

1. Giới thiệu

Pod được xem như một đơn vị deploy cơ bản trong Kubernetes. Chúng ta có thể tạo, tùy chỉnh và quản lý pod thủ công. Thế nhưng, trong thế giới thực, chúng ta muốn Kubernetes giải quyết và tự động xử lý các pod để đảm bảo tính ổn định cho hệ thống. Vì lẽ đó, trong thực tế, Pod không bao giờ được sử dụng trực tiếp mà thay vào đó chúng ta sử dụng các thành phần khác như ReplicaSet và Deployment.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021



ReplicaSet sẽ luôn đảm bảo một số lượng Pod cụ thể được chạy dưới mọi thời điểm. Như vậy, nó thường được dùng để đảm bảo tính sẵn sàng của hệ thống bằng việc xác định số lượng pod cụ thể.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

2. Tạo một ReplicaSet từ mô tả YAML

```

apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: frontend
  labels:
    app: guestbook
    tier: frontend
spec:
  # modify replicas according to your case
  replicas: 3
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
        - name: php-redis
          image: gcr.io/google_samples/gb-frontend:v3

```

Figure 6 frontend.yaml

```
kubectl apply -f frontend.yaml
```

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

3. Xem các ReplicaSet trong hệ thống

```
kubectl get rs
```

NAME	DESIRED	CURRENT	READY	AGE
frontend	3	3	3	6s

4. Xem các Pod trong ReplicaSet đó

```
kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
frontend-b2zdvd	1/1	Running	0	6m36s
frontend-vcmts	1/1	Running	0	6m36s
frontend-wtsmm	1/1	Running	0	6m36s

Như đã thấy ở đây, ReplicaSet này chứa 3 pod như chúng ta đã khai báo ở trường **replicas**.

Miêu tả ReplicaSet

```
kubectl describe rs/frontend
```

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

```

Name:          frontend
Namespace:     default
Selector:      tier=frontend
Labels:        app=guestbook
               tier=frontend
Annotations:   kubectl.kubernetes.io/last-applied-configuration:
               {"apiVersion":"apps/v1","kind":"ReplicaSet","metadata":{"annotations":{},"labels":{"app":"guestbook"},"spec":{"replicas":3,"selector":{"matchLabels":{"app":"guestbook","tier":"frontend"},"matchExpressions":[]},"template":{"metadata":{"labels":{"app":"guestbook","tier":"frontend"},"spec":{"containers":[{"name":"php-redis","image":"gcr.io/google_samples/gb-frontend:v3","ports":[{"containerPort":80}]}]}}}}
Replicas:      3 current / 3 desired
Pods Status:   3 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  tier=frontend
  Containers:
    php-redis:
      Image:          gcr.io/google_samples/gb-frontend:v3
      Port:           <none>
      Host Port:      <none>
      Environment:    <none>
      Mounts:         <none>
      Volumes:        <none>
Events:
  Type     Reason             Age   From                    Message
  ----     -
  Normal   SuccessfulCreate    117s  replicaset-controller   Created pod: frontend-wtsmm
  Normal   SuccessfulCreate    116s  replicaset-controller   Created pod: frontend-b2zdv
  Normal   SuccessfulCreate    116s  replicaset-controller   Created pod: frontend-vcmts

```

Câu lệnh này được sử dụng để kiểm tra các đặc tả của ReplicaSet được tạo.

C. Services

1. Giới thiệu

Chúng ta đã biết về Pod, cách để deploy chúng thông qua ReplicaSet để đảm bảo cho chúng hoạt động. Mặc dù có một vài Pods đặc thù có thể hoặc độc lập không phụ thuộc vào các tác nhân bên ngoài, đa phần các Pod sẽ có sự tương tác và giao tiếp qua lại lẫn nhau. Trong trường hợp chúng ta sử dụng kiến trúc **Microservices**, các Pod sẽ

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

phải phản hồi lại các yêu cầu từ bên ngoài (HTTP request) từ ngoài hệ thống hoặc có thể đến từ các Pod khác. Pod cần có phương pháp để giao tiếp với nhau để tạo nên một hệ thống hoàn chỉnh. Trong các nền tảng khác, Sysadmin sẽ cấu hình cho các thiết bị giao tiếp với nhau qua một địa chỉ IP cụ thể, tuy nhiên, điều này là không thể đối với Kubernetes bởi vì:

Pod không có tính vững bền, chúng có thể chết bất kì lúc nào, có thể bởi chúng được chúng được loại bỏ để dành tài nguyên cho các Pod khác hoặc đơn giản là Cluster gặp sự cố.

Kubernetes gán địa chỉ IP cho Pod sau khi nó được cài đặt vào Node và trước khi bắt đầu hoạt động – Chúng ta không thể biết trước địa chỉ IP của một Pod.

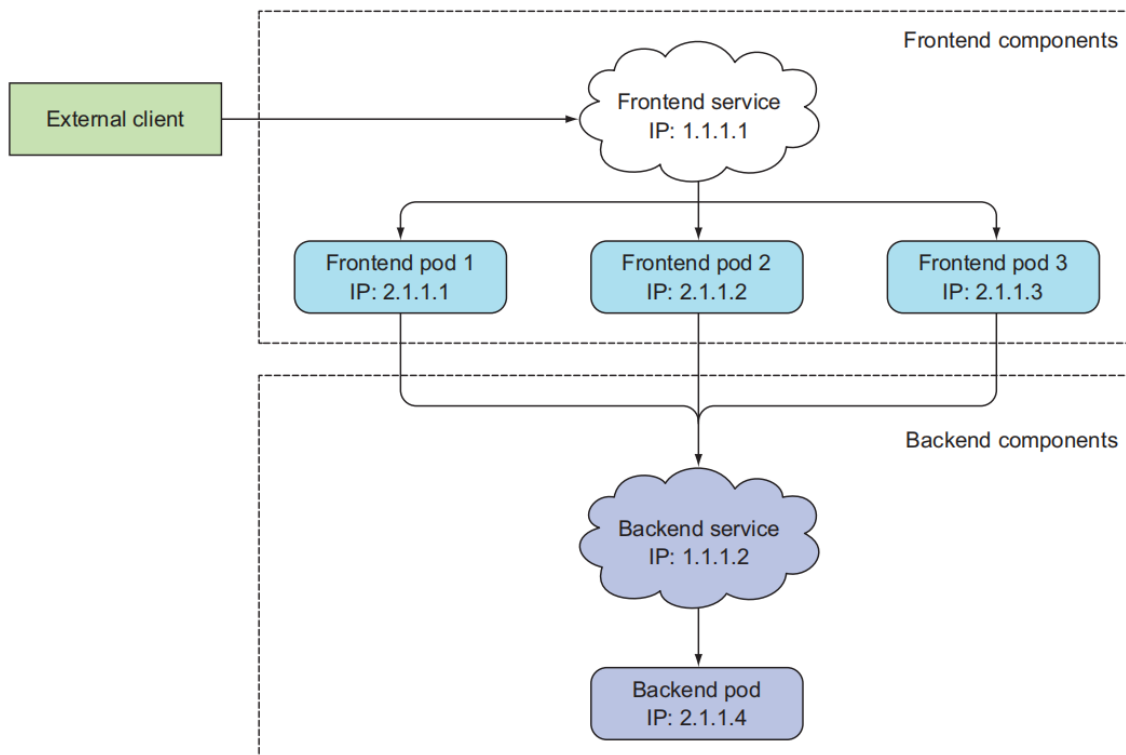
Vì lẽ đó, Kubernetes cung cấp cho chúng ta một công cụ đặc biệt – **Services**

Service là một thành phần được tạo nên để trở thành con đường liên kết đến nhóm khác Pod có chung chức năng. Mỗi service có một địa chỉ IP và port mà sẽ không thay đổi trong quá trình hoạt động. Chúng ta có thể tạo các kết nối đến địa chỉ IP và port đó và chúng sẽ dẫn đến các Pod.

Ví dụ: Ta có một trang web có frontend web server và backend web server , trong đó có nhiều Pod được sử dụng cho frontend và chỉ có một Pod cho backend. Bằng việc tạo một Service cho các Pod frontend và cấu hình cho phép hoạt động bên ngoài Cluster, chúng ta cung cấp duy nhất một địa chỉ IP để người dùng kết nối vào đến hệ thống. Đồng thời chúng ta cũng cung cấp một Service để kết nối giữa frontend và backend, tạo

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

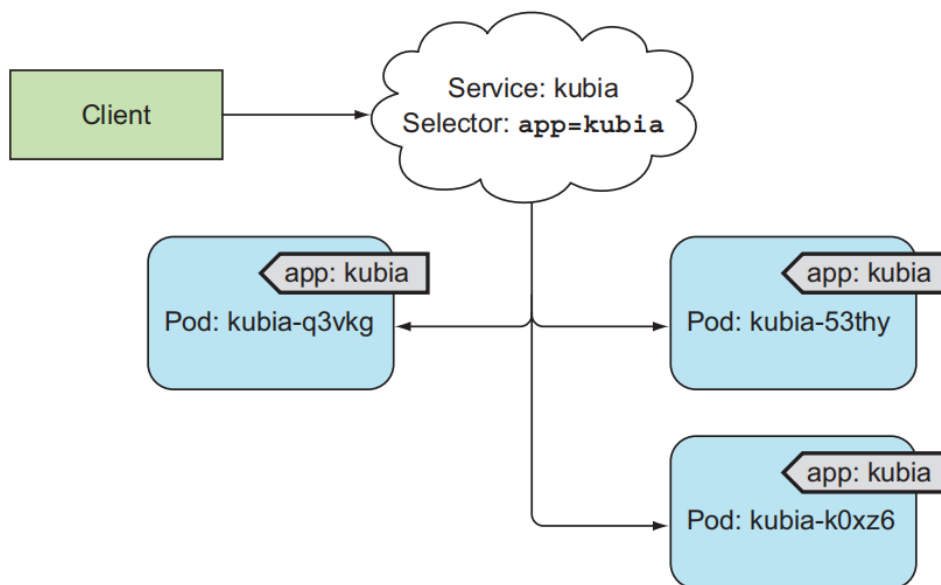
ra một kết nối ổn định giữa các Pod.



Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

2. Tạo một Service từ mô tả YAML

Các Label selector sẽ kết nối giữa Pod và Service



```

apiVersion: v1
kind: Service
metadata:
  name: kubia
spec:
  ports:
    - port: 80
      targetPort: 8080
  selector:
    app: kubia
  
```

Figure 7 Service đơn giản

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

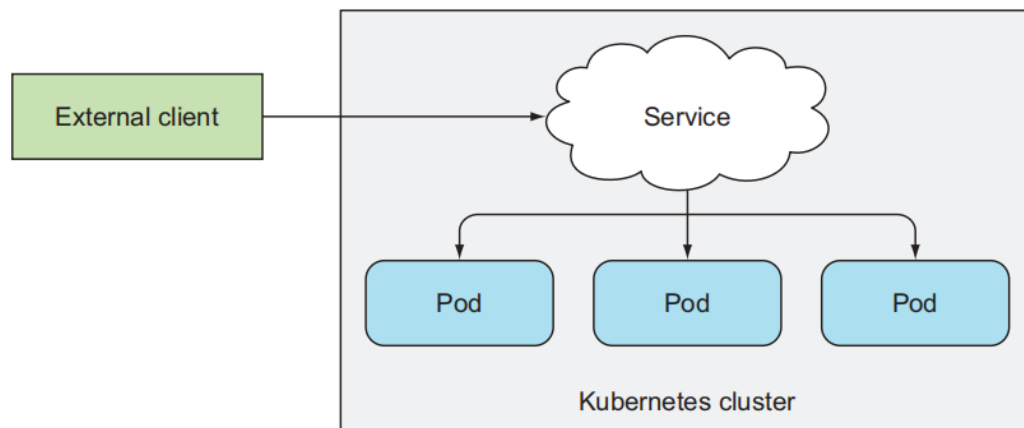
Chúng ta vừa khai báo một Service tên là kubia, Service này sẽ chấp nhận những kết nối ở port 80 và sẽ điều hướng những kết nối này đến với những Pods có label selector là app:kubia.

3. Các loại Services

ClusterIP: Service này cung cấp địa chỉ IP chỉ có thể được nhìn thấy trong phạm vi nội bộ của Cluster. Đây là Service mặc định.

NodePort: Cung cấp một Node IP ở một port tĩnh (NodePort). Chúng ta có thể kết nối với NodePort Service bên ngoài Cluster qua địa chỉ <NodeIP>:<NodePort>.

NodePort cho phép chúng ta truy cập từ ngoài Cluster.



Ví dụ

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

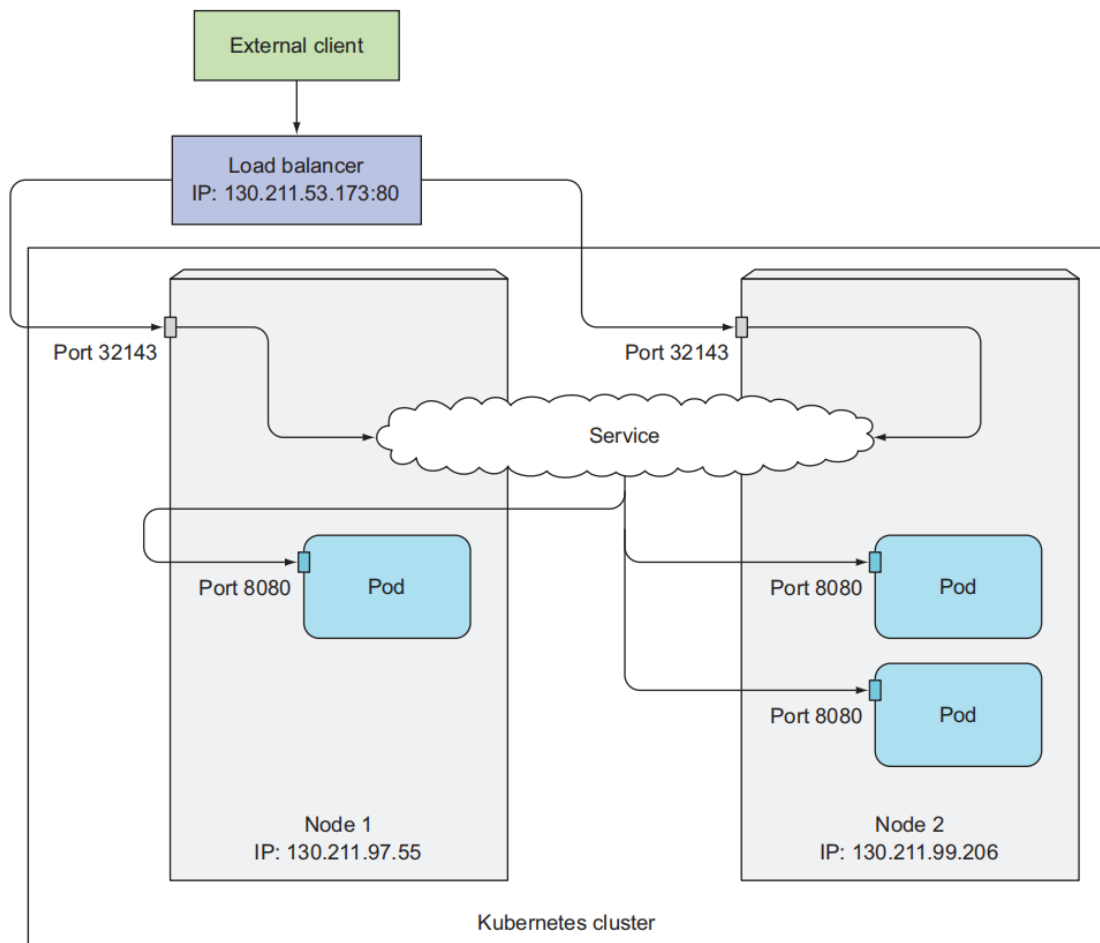
```

apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  type: NodePort
  selector:
    app: MyApp
  ports:
    # By default and for convenience, the `targetPort` is set to the same value as the `port` field
    - port: 80
      targetPort: 80
    # Optional field
    # By default and for convenience, the Kubernetes control plane will allocate a port from a random
    nodePort: 30007

```

LoadBalancer: Kubernetes chạy trên các dịch vụ có chức năng cân bằng tải cho hệ thống. Tất cả những gì chúng ta cần thiết lập là chuyển kiểu Service sang LoadBalancer thay vì NodePort. Service này sẽ có địa chỉ IP dành riêng của nó và sẽ chịu trách nhiệm điều hướng các kết nối các Pods. Quy tắc điều hướng của Service sẽ do máy chủ đám mây quyết định.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021



Ví dụ:

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

```

apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
    app: MyApp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
  clusterIP: 10.0.171.239
  type: LoadBalancer
status:
  loadBalancer:
    ingress:
      - ip: 192.0.2.127

```

D. Secrets

1. Giới thiệu

Secret là thành phần lưu trữ số lượng dữ liệu quan trọng và mật thiết như mật khẩu, token, hoặc là API KEY. Sử dụng Secret đồng nghĩa với việc bạn không cần bỏ những thông tin quan trọng đó vào ứng dụng.

Vì Secret có thể khởi tạo độc lập với Pod sử dụng chúng, điều đó bảo mật dữ liệu tốt hơn trong quá trình khởi tạo, quản lý và sử chữa các Pods.

Để sử dụng Secret, Pod cần phải tham chiếu đến Secret đó. Một Secret có thể được sử dụng trong Pod theo 3 cách:

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Như là một file được Volume

Như là một biến môi trường của containers

Bởi **kubelet** khi pull một image về cho Pod

Bạn có thể tùy chọn trường data và stringData khi khởi tạo file cài đặt cho Secret, đồng thời hai trường này là tùy chọn. Giá trị của tất cả các key trong trường data phải được mã hóa theo chuẩn base64. Nếu chúng ta không hài lòng với kiểu mã hóa này, chúng ta có thể dùng trường stringData thay thế - trường không có giới hạn về kiểu kí tự.

Các loại Secret

Chúng ta có thể chọn loại Secret qua trường type

Builtin Type	Usage
Opaque	arbitrary user-defined data
kubernetes.io/service-account-token	service account token
kubernetes.io/dockercfg	serialized ~/.dockercfg file
kubernetes.io/dockerconfigjson	serialized ~/.docker/config.json file
kubernetes.io/basic-auth	credentials for basic authentication
kubernetes.io/ssh-auth	credentials for SSH authentication
kubernetes.io/tls	data for a TLS client or server
bootstrap.kubernetes.io/token	bootstrap token data

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Chúng ta có thể tự khai báo và sử dụng type Secret mong muốn bằng việc gán chuỗi vào trường giá trị type. Một chuỗi rỗng sẽ được xem như kiểu Opaque.

2. Opaque secrets

```
kubectl create secret generic empty-secret
kubectl get secret empty-secret
```

NAME	TYPE	DATA	AGE
empty-secret	Opaque	0	2m6s

Service account token Secret

kubernetes.io/service-account-token là kiểu Secret dùng để lưu trữ token cho việc xác thực tài khoản. Ví dụ

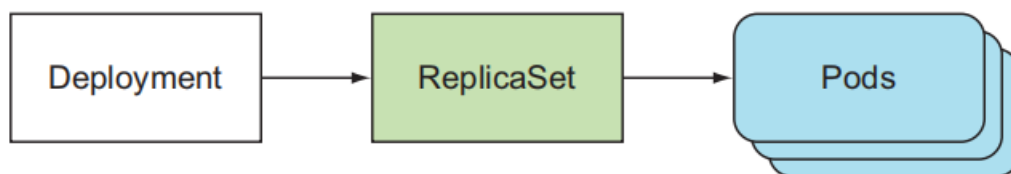
```
apiVersion: v1
kind: Secret
metadata:
  name: secret-sa-sample
  annotations:
    kubernetes.io/service-account.name: "sa-name"
type: kubernetes.io/service-account-token
data:
  # You can include additional key value pairs as you do with Opaque Secrets
  extra: YmFyCg==
```

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

E. Deployment

1. Giới thiệu

Là một trong các thành phần quan trọng nhất của Kubernetes được tạo ra để cập nhật cho Pods và ReplicaSet. Deployment là một đơn vị bậc cao hơn so với Pod và ReplicaSet, khi chúng ta tạo Deployment, ReplicaSet sẽ được tạo bên dưới và chịu trách nhiệm quản lý các Pods.



Sử dụng Deployment giúp chúng ta cập nhật hệ thống một cách đơn giản hơn, chúng ta chỉ cần cài đặt trạng thái mong muốn (desired state) và Kubernetes sẽ quản lý phần còn lại.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

2. Tạo một Deployment

```

apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: kubia
spec:
  replicas: 3
  template:
    metadata:
      name: kubia
      labels:
        app: kubia
    spec:
      containers:
        - image: luksa/kubia:v1
          name: nodejs

```




Figure 8 kubia.yaml

```
kubectl apply -f kubia.yaml
```

3. Cập nhật một Deployment

Set image mới cho Deployment demo-depl

```

$ kubectl set image deployment kubia nodejs=luksa/kubia:v3
deployment "kubia" image updated

```

Rollout để cập nhật Deployment đó.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

```
$ kubectl rollout status deployment kubia
Waiting for rollout to finish: 1 out of 3 new replicas have been updated...
Waiting for rollout to finish: 2 out of 3 new replicas have been updated...
Waiting for rollout to finish: 1 old replicas are pending termination...
deployment "kubia" successfully rolled out
```

Đồng thời, chúng ta có thể quan sát lịch sử rollout của Deployment và trở về một vị trí cụ thể nếu muốn

```
$ kubectl rollout history deployment kubia
deployments "kubia":
REVISION    CHANGE-CAUSE
2           kubectl set image deployment kubia nodejs=luksa/kubia:v2
3           kubectl set image deployment kubia nodejs=luksa/kubia:v3
```

V. Hướng dẫn chạy một ứng dụng bằng Kubernetes

A. Yêu cầu

Cài đặt Docker and Kubernetes

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

B. Tạo và chạy một container image

1. Tạo một ứng dụng bằng Node.js

```
import { createServer } from 'http';
import { hostname } from 'os';
console.log('DucTam server starting...');
var handler = function (request, response) {
  console.log('Received request from ' + request.connection.remoteAddress);
  response.writeHead(200);
  response.end('You've hit ' + hostname() + '\n');
};
var www = createServer(handler);
www.listen(8080);
```

app.js

2. Tạo file Dockerfile cho image

Để đóng gói ứng dụng thành một image, chúng ta cần tạo 1 file gọi là Dockerfile, bao gồm các câu lệnh mà Docker sẽ thực thi để xây dựng lên image.

```
FROM node:7
ADD app.js /app.js
ENTRYPOINT ["node", "app.js"]
```

Dockerfile

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

3. Build Docker image

Đảm bảo rằng bạn đang ở thư mục chứa Dockerfile.

```
docker build -t ductam2943/demo .
```

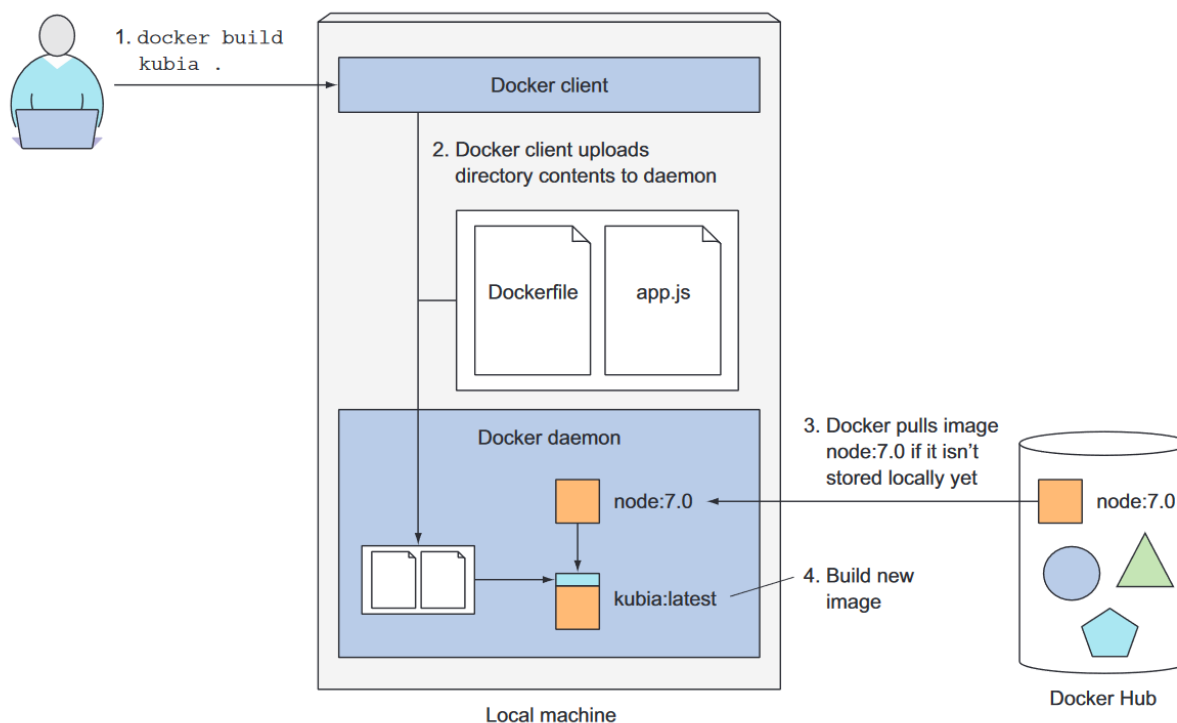


Figure 2.2 Building a new container image from a Dockerfile

Khi quá trình build hoàn thành, bạn sẽ có một image được lưu trữ cục bộ. Bạn có thể nhìn thấy bằng câu lệnh `docker image list`.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

```

Duc Tam@DESKTOP-IUD40SD MINGW64 ~/demo-app
$ docker image list
REPOSITORY                                TAG
ductam/demo                               latest

```

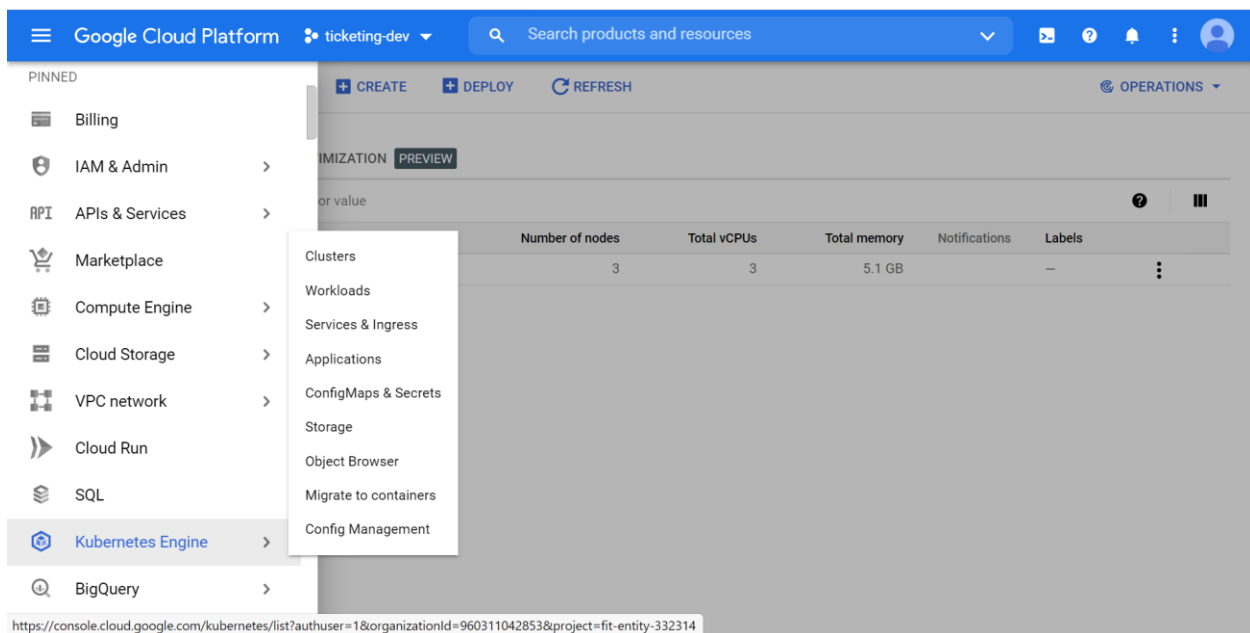
4. Push image lên DockerHub

```
docker push ductam2943/demo
```

Sau khi push image lên registry, bạn có thể sử dụng nó ở các thiết bị khác.

5. Setup cluster Kubernetes

Google Cluster Engine



Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Chọn phần Cluster, đến Create sau đó chọn cấu hình mong muốn.

The screenshot shows the Google Cloud Platform interface for Kubernetes clusters. The top navigation bar includes the Google Cloud Platform logo, the account 'ticketing-dev', and a search bar. The main header shows 'Kubernetes clusters' with buttons for 'CREATE', 'DEPLOY', and 'REFRESH', and a link to 'OPERATIONS'. The left sidebar contains various navigation icons. The main content area has tabs for 'OVERVIEW', 'COST OPTIMIZATION', and 'PREVIEW'. Under the 'OVERVIEW' tab, there is a filter bar and a table listing the clusters.

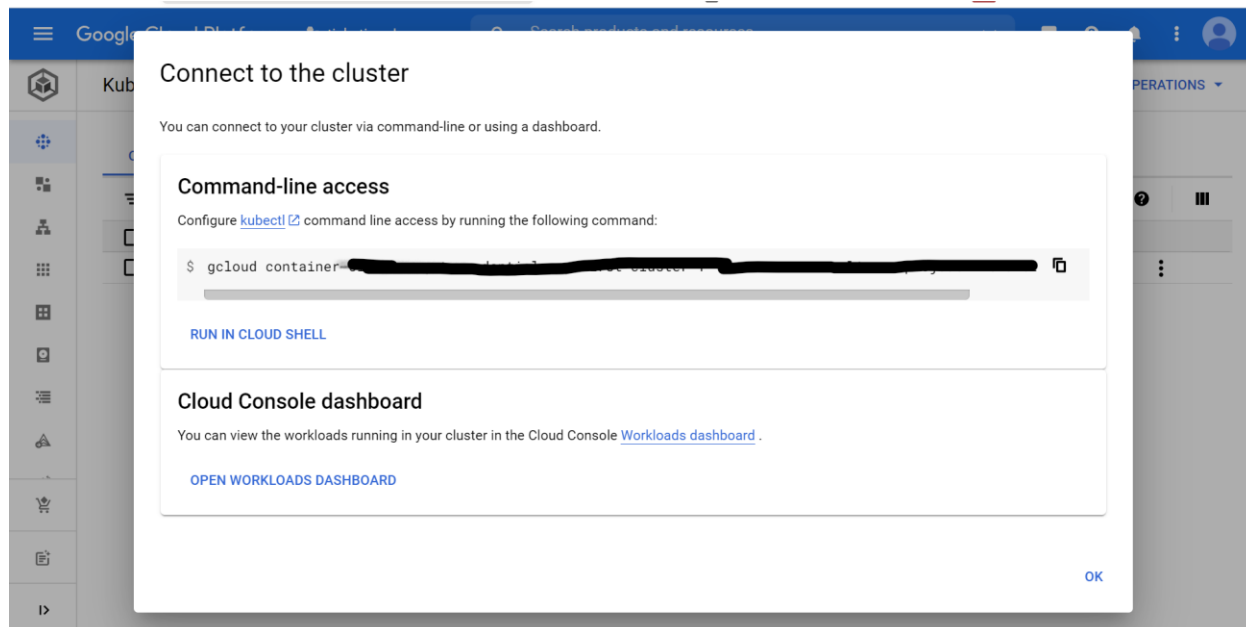
Status	Name	Location	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
<input checked="" type="checkbox"/>	my-first-cluster-1	us-central1-c	3	3	5.1 GB	—	—

Connect với Google Cluster Engine

This screenshot is similar to the previous one, but with a context menu open over the cluster 'my-first-cluster-1'. The menu contains three options: 'Edit', 'Connect', and 'Delete'.

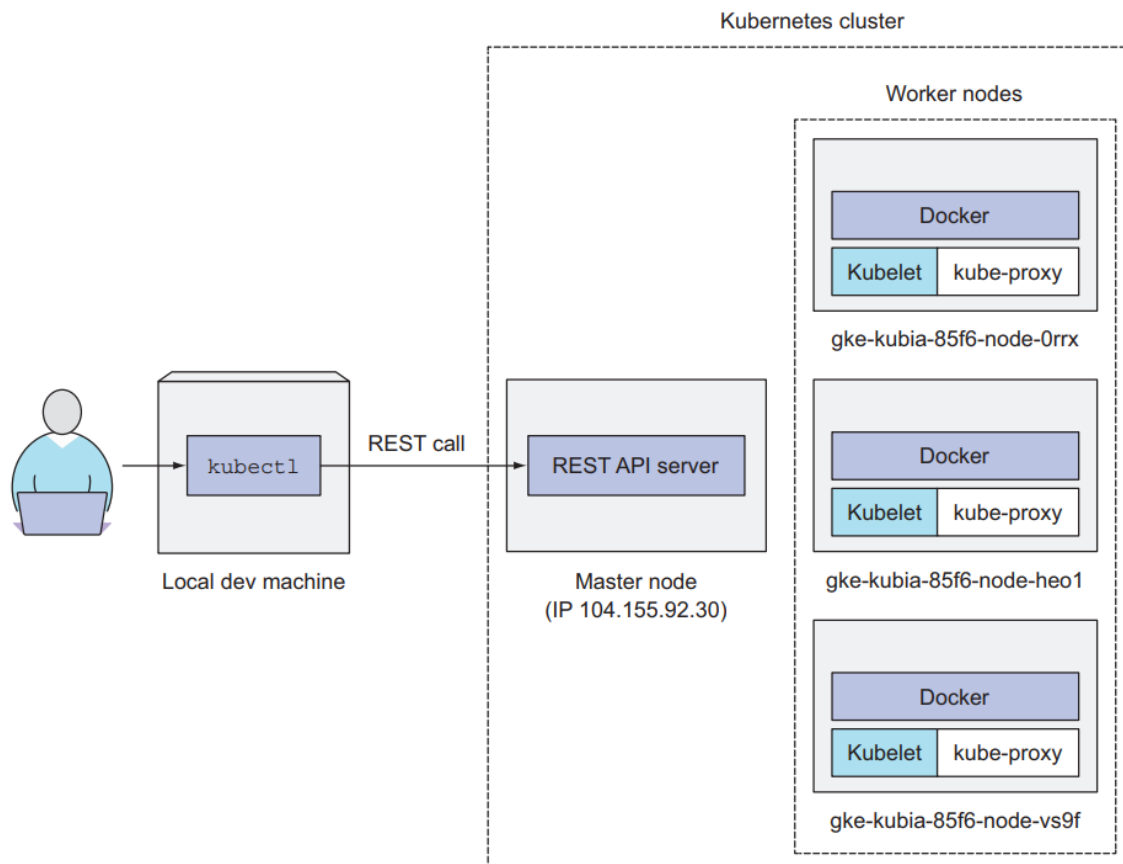
Status	Name	Location	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
<input checked="" type="checkbox"/>	my-first-cluster-1	us-central1-c	3	3	5.1 GB	—	—

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021



Copy lệnh và chạy trên terminal. Cách Google Cluster hoạt động.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021



6. Chạy ứng dụng đầu tiên trên Kubernetes

Cách đơn giản nhất là sử dụng lệnh `kubectl run` sẽ tạo ra tất cả components cần thiết mà không phải viết các file mô tả.

```

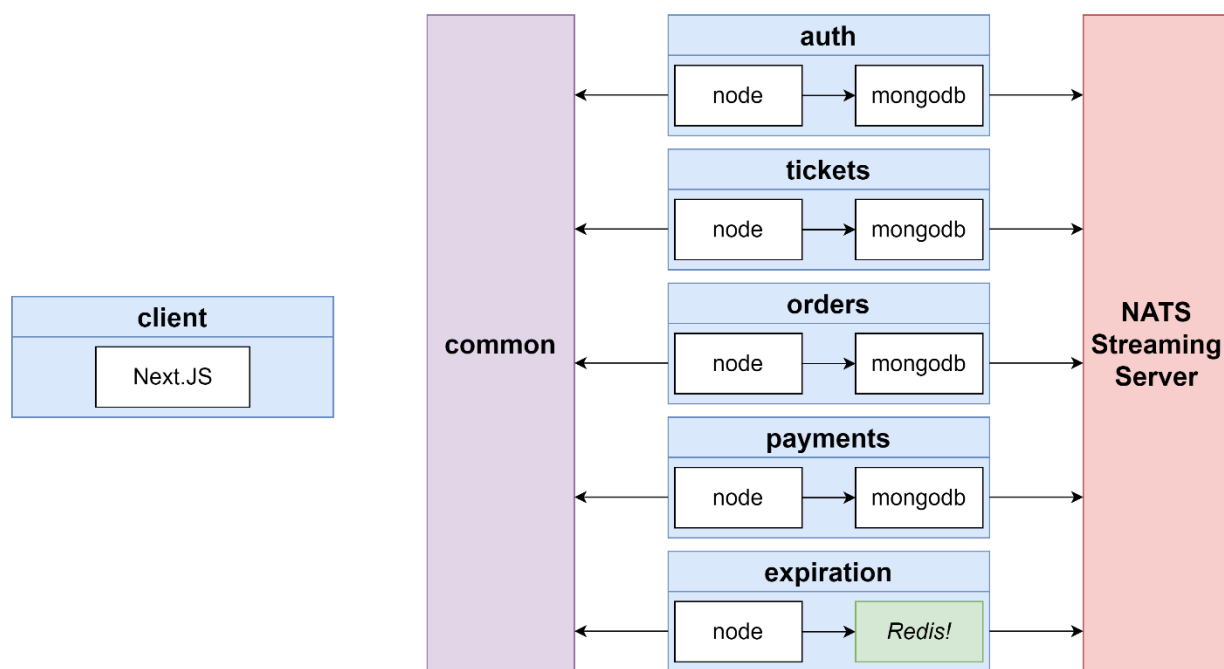
... Docker
kubectl run demo --image=ductam2943/demo --port=8080
Copy Caption

```

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

VI. Xây dựng ứng dụng minh họa

A. Kiến trúc ứng dụng



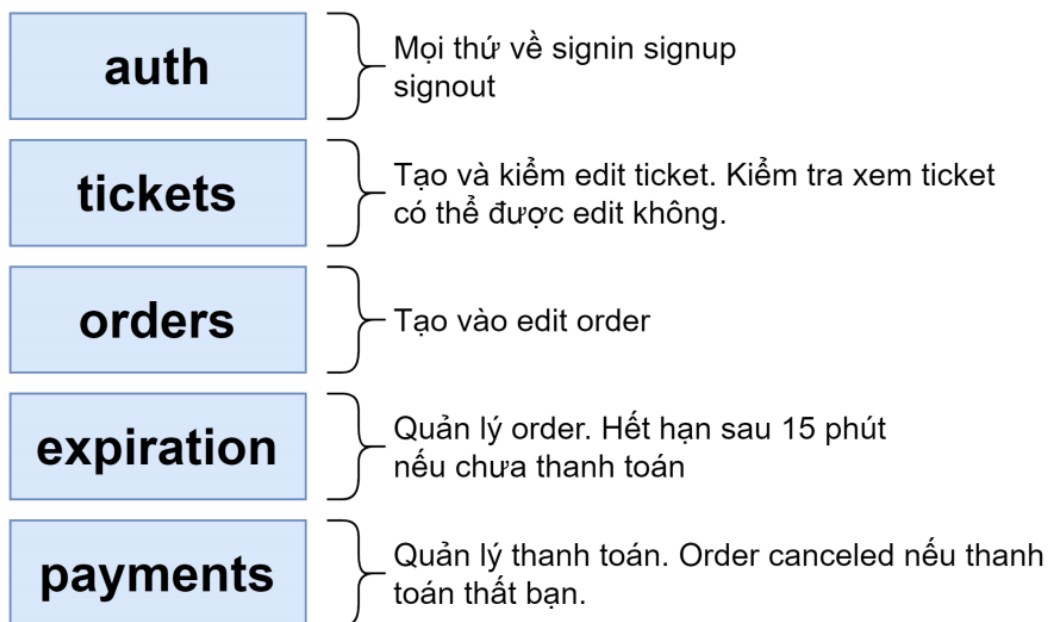
Ứng dụng sẽ được xây dựng theo kiến trúc Vi dịch vụ (Microservices) bao gồm các thành phần sau:

- Client service: là service cho UI ứng dụng, đảm nhận vai trò frontend, gọi đến các service khác để thực hiện nhiệm vụ.
- Auth service: là service quản lý xác thực người dùng, đảm nhận các chức năng đăng nhập, đăng xuất, xác thực người dùng.
- Ticket service: là service quản lý các tickets. Cho phép người dùng thực hiện thao tác mua và bán các tickets trên hệ thống.
- Order service: là service quản lý các order (đơn đặt hàng) ticket.

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

- Expiration service: là service quản lý việc hết hạn trong order (cho phép các ticket được trở lại thị trường sau một khoản thời gian nhất định).
- Payment service: là service quản lý việc thanh toán các order.
- NAT Streaming Server: là server quản lý việc truyền thông điệp giữa các service đồng thời giúp hệ thống khôi phục các thông điệp khi gặp sự cố.
- Common library: là thư viện chia sẻ giữa các services

Services



Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

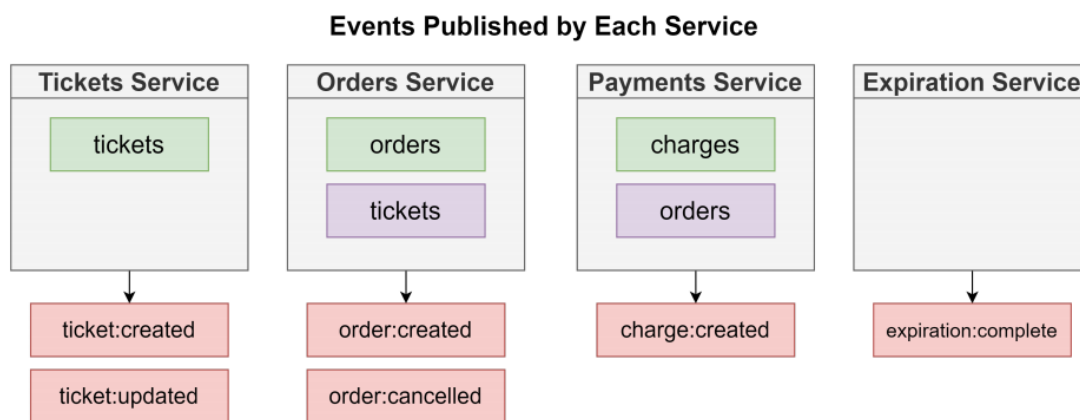
1. Chi tiết về một vài services

auth			
Route	Method	Body	Purpose
/api/users/signup	POST	{ email: string, password: string }	Đăng ký tài khoản
/api/users/signin	POST	{ email: string, password: string }	Đăng nhập
/api/users/signout	POST	{ }	Đăng xuất
/api/users/currentuser	GET	-	Lấy thông tin tài khoản

orders			
Route	Method	Body	Purpose
/api/orders	GET	-	Lấy tất cả order của người dùng
/api/orders/:id	GET	-	Lấy chi tiết về order
/api/orders	POST	{ ticketId: string }	Tạo một order cho một ticket
/api/orders/:id	DELETE	-	Hủy order

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

2. Các message được giao tiếp giữa các services



B. Routing

Các đường dẫn điều hướng cho frontend

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Route	File in Pages Dir	Goal
/auth/signin	/auth/signin.js	màn hình đăng ký
/auth/signup	/auth/signup.js	màn hình đăng nhập
/auth/signout	/auth/signout.js	đăng xuất
/	/index.js	danh sách các ticket
/tickets/new	/tickets/new.js	màn hình tạo ticket mới
/tickets/:ticketId	/tickets/[ticketId].js	màn hình chi tiết về ticket
/orders/:orderid	/orders/[orderId].js	Chi tiết về order và thanh toán

C. Các màn hình ứng dụng

Màn hình trước khi đăng nhập

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Ticketing

[Sign Up](#)

[Sign In](#)

Now

Title	Price	Link
Buổi hòa nhạc AAA	121	View
Buổi hòa nhạc BBB	1212	View
Buổi hòa nhạc CCC	121	View
Trận cầu giữa hai đội tuyển AA và BB	121	View
Trận đấu quyền anh giữa A và B	121	View

Màn hình mặc định sau khi đăng nhập

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Ticketing			Sell Tickets	My Orders	Stock	Sign Out
Now						
Title	Price	Link				
Buổi hòa nhạc AAA	121	View				
Buổi hòa nhạc BBB	1212	View				
Buổi hòa nhạc CCC	121	View				
Trận cầu giữa hai đội tuyển AA và BB	121	View				
Trận đấu quyền anh giữa A và B	121	View				

Màn hình tạo một ticket để bán

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Ticketing

Sell TicketsMy OrdersStockSign Out

Create a Ticket

Title

Price

Submit

Màn hình chi tiết ticket

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Ticketing

Sell TicketsMy OrdersStockSign Out

Buổi hòa nhạc AAA

Price: 121

Purchase

Màn hình thanh toán

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

TicketingSell TicketsMy OrdersStockSign Out

Time left to pay: 54 seconds [Pay With Card](#)

TicketingSell TicketsMy OrdersStockTEST MODE

Time left to pay: 25 seconds [Pay With Card](#)

tk2@gmail.com

2323 2323 323

03 / 2020

2132

[Pay \\$121.00](#)

Powered by **stripe**

Tài liệu báo cáo	SE121.M11.PMCL - VN(ĐA)
Tìm hiểu về Kubernetes và xây dựng ứng dụng	Date: 12/12/2021

Màn hình lịch sử mua ticket

Ticketing				Sell Tickets	My Orders	Stock	Sign Out
History							
Title	Price	Status	Link				
Buổi hòa nhạc AAA	121	cancelled	View				

D. Đường dẫn đến ứng dụng

Link github: <https://github.com/DucTam2411/KubernetesDemo.git>