

|  |
| --- |
| **BỘ MÔN: ĐIỆN TOÁN ĐÁM MÂY**    **----------**  BÙI NGỌC LÂM : 18110140  LƯƠNG VĂN HOÀI : 18110094 |
| **BÁO CÁO CUỐI KỲ**  **\*\*\***  **TÌM HIỂU VỀ KUBERNETES VÀ ỨNG DỤNG DEMO**  **GVHD: HUỲNH XUÂN PHỤNG**  **LỚP: THỨ 5 TIẾT 12-15**  **TP.HCM, ngày 30 tháng 11 năm 2021** |

**DANH SÁCH THÀNH VIÊN**

*HỌC KỲ I NĂM HỌC 2021-2022*

**Nhóm 1: ( Lớp thứ 5 – Tiết 12-15)**

**Tên đề tài:** ***Tìm hiểu về Kubernetes và ứng dụng demo***

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **HỌ VÀ TÊN SINH VIÊN** | **MÃ SỐ SINH VIÊN** | **TỈ LỆ %**  **HOÀN THÀNH** |
| 1 | Lương Văn Hoài | 18110094 | 100% |
| 2 | Bùi Ngọc Lâm | 18110140 | 100% |

*Ngày 30 tháng 11 năm 2021*

**MỤC LỤC**



[MỞ ĐẦU](#_Toc89976937)

[1. Lí do chọn đề tài](#_Toc89976938)

[2. Mục đích nghiên cứu](#_Toc89976939)

[3. Phương pháp nghiên cứu](#_Toc89976940)

[NỘI DUNG](#_Toc89976942)

1.Kubernetes là gì?

**2. Những khái niệm cơ bản trong Kubernetes**

**3.Ai cần kubernetes ?**

**4.Các sản phẩm dựa trên Kubernetes**

**5.Các thành phần trong Kubernetes**

**6.Cách cài đặt Kubernetes**

**7.Kubernetes Dashboard**

**8.Deployment trong Kubernetes**

[KẾT LUẬN](#_Toc89976955)

[TÀI LIỆU THAM KHẢO](#_Toc89976957)

**MỞ ĐẦU**

**1.Lí do chọn đề tài**

Kubernetes là nền tảng có mã nguồn mở rất thuận tiện cho người sử dụng , nó có lợi ích vô cùng lớn đối với người dùng.Những lợi ích đó là:

Quy mô toàn cầu: Được thiết kế tương đồng với các nguyên tắc mà chính các nguyên tắc này cho phép Google chạy hàng tỷ container mỗi tuần, Kubernetes có thể mở rộng quy mô mà không cần tăng sự hoạt động của công việc của bạn.

Không bao giờ cạn : Cho dù thử nghiệm tại địa phương hay điều hành tại một doanh nghiệp toàn cầu, tính linh hoạt của hệ thóng phần mềm này sẽ phát triển tùy vào yêu cầu của bạn để cung cấp các ứng dụng của bạn một cách nhất quán và dễ dàng cho dù nhu cầu của bạn phức tạp đến mức nào.

Vận hành mọi nơi : với hệ thống mã nguồn mở cho phép bạn tự do tận dụng cơ sở hạ tầng tại chỗ, lai hoặc đám mây công cộng, cho phép bạn dễ dàng di chuyển khối lượng công việc đến nơi quan trọng với bạn.

**2.Mục đích nghiên cứu**

Tìm hiểu những khái niệm về Kubernetes

Cài đặt kubernetes cluster

Deployment Nginx web server

Cài đặt Kubernetes Dashboard

**NỘI DUNG**

1. **Kubernetes là gì**

Kubernetes (còn gọi là k8s) là một hệ thống để chạy, quản lý, điều phối các ứng dụng được container hóa trên một cụm máy (1 hay nhiều) gọi là cluster. Với Kubernetes bạn có thể cấu hình để chạy các ứng dụng, dịch vụ sao cho phù hợp nhất khi chúng tương tác với nhau cũng như với bên ngoài. Bạn có thể điều chỉnh tăng giảm tài nguyên, bản chạy phục vụ cho dịch vụ (scale), bạn có thể cập nhật (update), thu hồi update khi có vấn đề ... Kubernetes là một công cụ mạnh mẽ, mềm dẻo, dễ mở rộng khi so sánh nó với công cụ tương tự là Docker Swarm!

Kubernetes được thiết kế bởi Google, về ngữ nghĩa thì tên này là tiếng Hy Lạp có nghĩa là tài công

1. **Những khái niệm cơ bản trong Kubernetes**

**Master node:** Là server điều khiển các máy Worker chạy ứng dụng. Master node bao gồm 4 thành phần chính:

* **Kubernetes API Server:** là thành phần giúp các thành phần khác liên lạc nói chuyện với nhau. Lập trình viên khi triển khai ứng dụng sẽ gọi API Kubernetes API Server này.
* **Scheduler:** Thành phần này lập lịch triển khai cho các ứng dụng, ưng dụng được đặt vào Worker nào để chạy
* **Controler Manager:** Thành phần đảm nhiệm phần quản lý các Worker, kiểm tra các Worker sống hay chết, đảm nhận việc nhân bản ứng dụng…
* **Etcd:** Đây là cơ sở dữ liệu của Kubernetes, tất cả các thông tin của Kubernetes được lưu trữ cố định vào đây.

**Worker node** : Là server chạy ứng dụng trên đó. Bao gồm 3 thành phần chính:

* **Container runtime:** Là thành phần giúp chạy các ứng dụng dưới dạng Container. Thông thường người ta sử dụng Docker.
* **Kubelet:** đây là thành phần giao tiếp với Kubernetes API Server, và cũng quản lý các container
* **Kubernetes Service Proxy:** Thành phần này đảm nhận việc phân tải giữa các ứng dụng

**Kubectl :** Tool quản trị Kubernetes, được cài đặt trên các máy trạm, cho phép các lập trình viên đẩy các ứng dụng mô tả triển khai vào cụm Kubernetes, cũng như là cho phép các quản trị viên có thể quản trị được cụm Kubernetes.

**Pod :** là khái niệm cơ bản và quan trọng nhất trên Kubernetes. Bản thân Pod có thể chứa 1 hoặc nhiều hơn 1 container. Pod chính là nơi ứng dụng được chạy trong đó. Pod là các tiến trình nằm trên các Worker Node. Bản thân Pod có tài nguyên riêng về file system, cpu, ram, volumes, địa chỉ network…

**Image :** Là phần mềm chạy ứng dụng đã được gói lại thành một chương trình để có thể chạy dưới dạng container. Các Pod sẽ sử dụng các Image để chạy.

**Deployment :** Là cách thức để giúp triển khai, cập nhật, quản trị Pod.

**Replicas Controller :** Là thành phần quản trị bản sao của Pod, giúp nhân bản hoặc giảm số lượng Pod.

**Service :** Là phần mạng (network) của Kubernetes giúp cho các Pod gọi nhau ổn định hơn, hoặc để Load Balancing giữa nhiều bản sao của Pod, và có thể dùng để dẫn traffic từ người dùng vào ứng dụng (Pod), giúp người dùng có thể sử dụng được ứng dụng.

**Label :** ra đời để phân loại và quản lý Pod,. Ví dụ chúng ta có thể đánh nhãn các Pod chạy ở theo chức năng frontend, backend, chạy ở môi trường dev, qc, uat, production…

1. **Ai cần Kubernetes?**

Các doanh nghiệp lớn, có nhu cầu thực sự phải scaling hệ thống nhanh chóng, và đã sử dụng container (Docker).

Các dự án cần chạy >= 5 container CÙNG LOẠI cho 1 dịch vụ. (Ví dụ dùng >=5 máy cùng để chạy code website XYZ). Còn nhỏ hơn thì tốt nhất không dùng - đừng mang dao mổ trâu đi giết gà.

Các startup hiện đại, chịu đầu tư vào công nghệ, để nhỡ về sau có to ra, thì to rất dễ

Các sysadmin/DevOps muốn tăng lương, nhảy việc, vọc công nghệ mới.

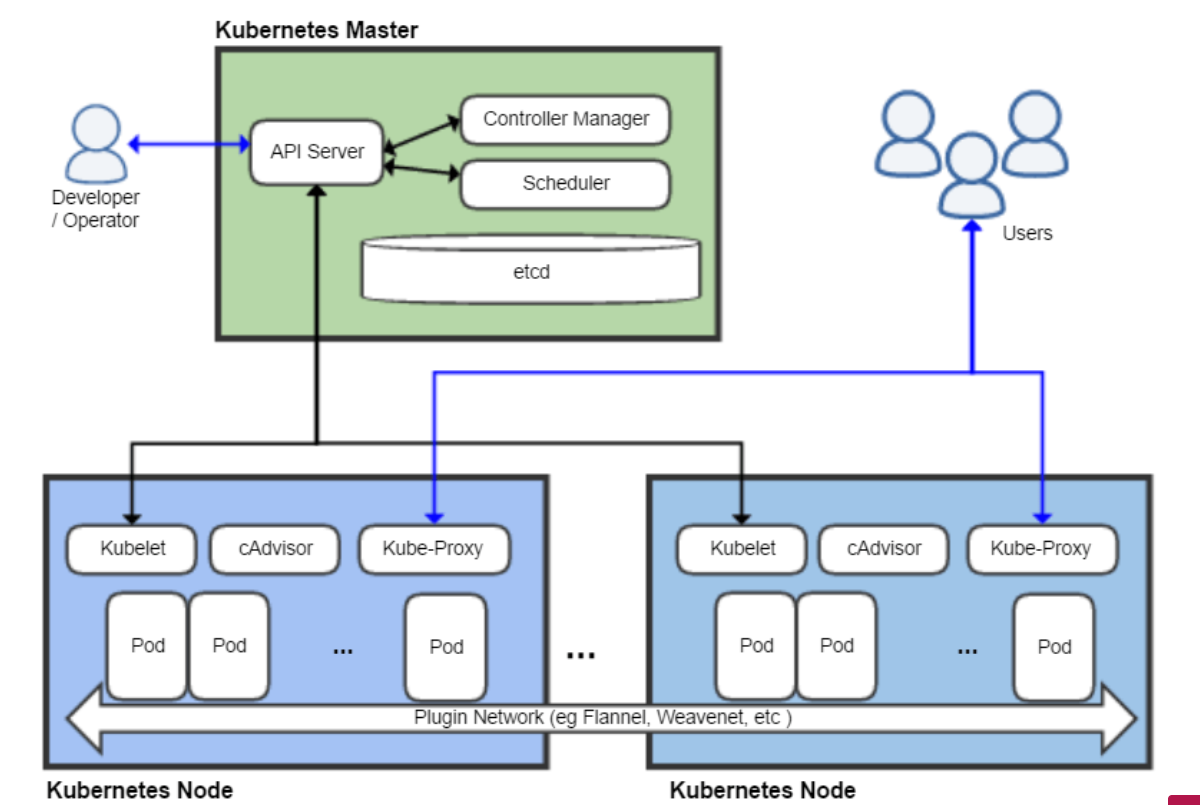
1. **Các sản phẩm dựa trên Kubernetes**

Cũng là nơi ta có thể nhìn rõ hơn K8s làm được gì, hay tìm tài liệu nếu trang chủ là không đủ:

* RedHat Openshift 3: <https://www.openshift.org/>
* Deis Workflow - opensource platform as a service (PaaS): <https://deis.com/paas/>
* Rancher - container management platform: <http://rancher.com/rancher/>

1. **Các thành phần của Kubernetes**

Hệ thống Kubernetes thật sự phức tạp, các thành phần của sẽ được tìm hiểu dần qua từng bài viết, nhưng ở đây có một sơ đồ và một vài khái quát ban đầu về kiến trúc / thành phần của nó



Master Server là máy chính của cluster, tại đây điều khiển cả cụm máy.

etct là thành phần cơ bản cần thiết cho Kubernetes, nó lưu trữ các cấu hình chung cho cả cụm máy, etct chạy tại máy master. etct là một dự án nguồn mở (xem tại etcd) nó cung cấp dịch vụ lưu dữ liệu theo cặp key/value

kube-apiserver chạy tại máy master, cung cấp các API Restful để các client (như kubectl) tương tác với Kubernetes

kube-scheduler chạy tại master, thành phần này giúp lựa chọn Node nào để chạy các ứng dụng căn cứ vào tài nguyên và các thành phần khác sao cho hệ thống ổn định.

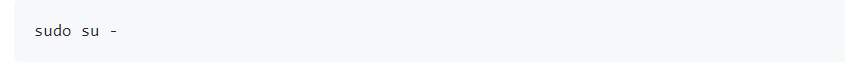
kube-controller chạy tại master, nó điều khiển trạng thái cluster, tương tác để thực hiện các tác vụ tạo, xóa, cập nhật ... các tài nguyên

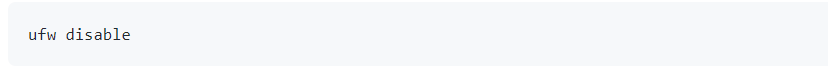
Kubelet dịch vụ vụ chạy trên tất cả các máy (Node), nó đảm đương giám sát chạy, dừng, duy trì các ứng dụng chạy trên node của nó.

Kube-proxy: cung cấp mạng proxy để các ứng dụng nhận được traffic từ ngoài mạng vào cluster.

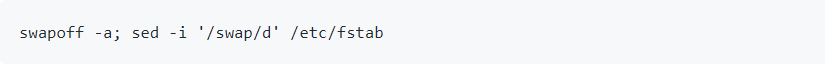
1. **Cách cài đặt Kubernetes**

Đăng nhập với tư cách rootngười dùng

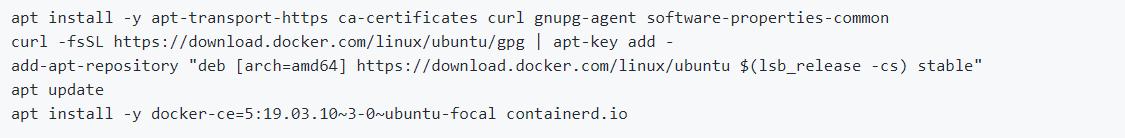
****Tắt tường lửa



Vô hiệu hóa hoán đổi

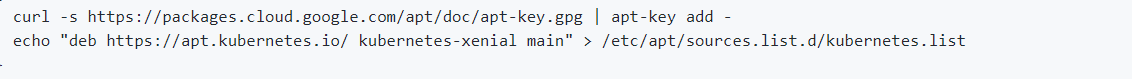


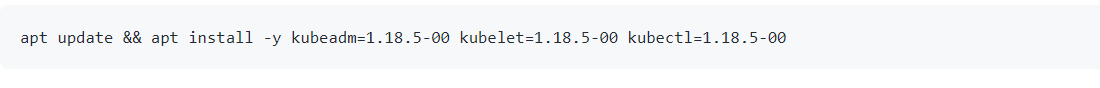
Cài đặt công cụ docker

****

Thiết lập Kubernetes

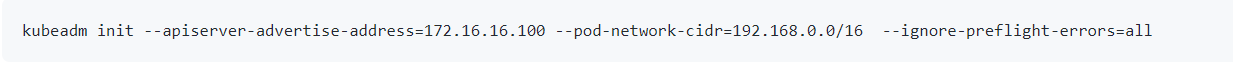
Thêm kho lưu trữ Apt

****Cài đặt các thành phần Kubernetes

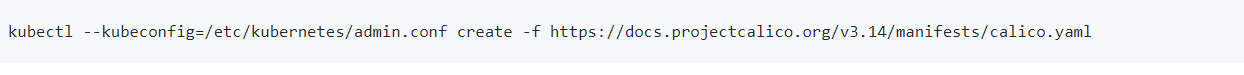


Trên kmaster

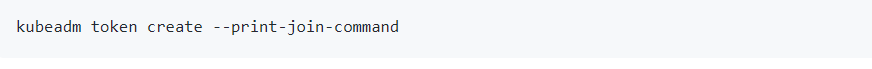
Khởi tạo Kubernetes Cluster



Triển khai mạng lưới Calico

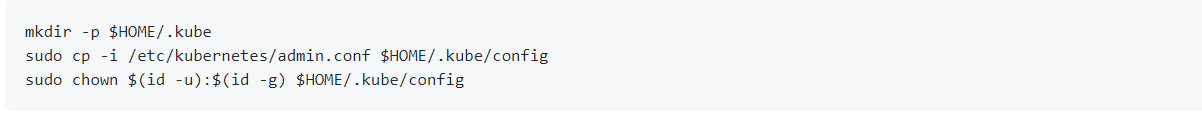


Lệnh tham gia cụm

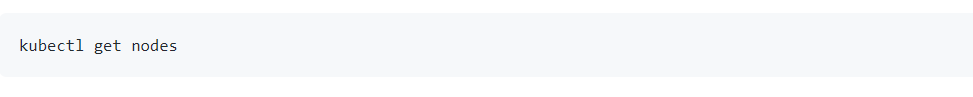
****

Để có thể chạy các lệnh kubectl với tư cách người dùng không phải root

Nếu bạn muốn có thể chạy các lệnh kubectl với tư cách là người dùng không phải root, thì với tư cách là người dùng không phải root, hãy thực hiện các lệnh này

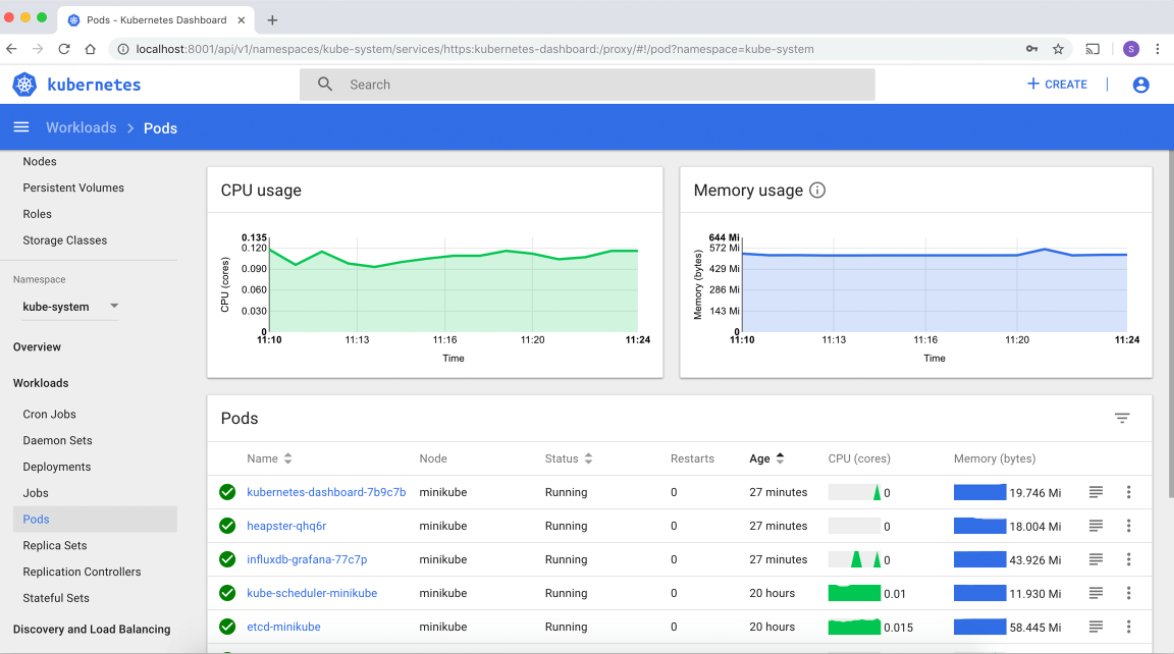
****Trên Kworker

Xác minh cụm (Trên kmaster)

****

1. **Kubernetes Dashboard**

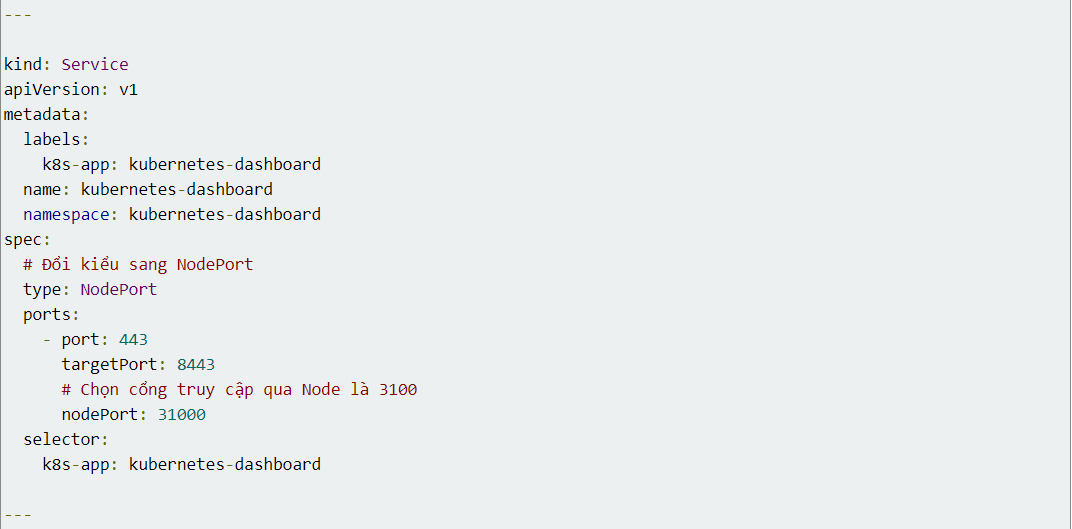
Kubernetes Dashboard là công cụ trên nền Web, kết nối đến Kubernetes để thực hiện việc quản lý (giống kubectl), ưu điểm của nó là trực quan dễ sử dụng. Thông tin chính về nó tại dashboard, các phiên bản Dashboard Release



**Cài đặt Kubernetes Dashboard trên Cluster:**

Để triển khai Kubernetes trước tiên lấy file cấu hình bản phù hợp về tại Dashboard Release, giả sử chọn v2.0.0-beta6 thì file cấu hình triển khai là:https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0-beta6/aio/deploy/recommended.yaml

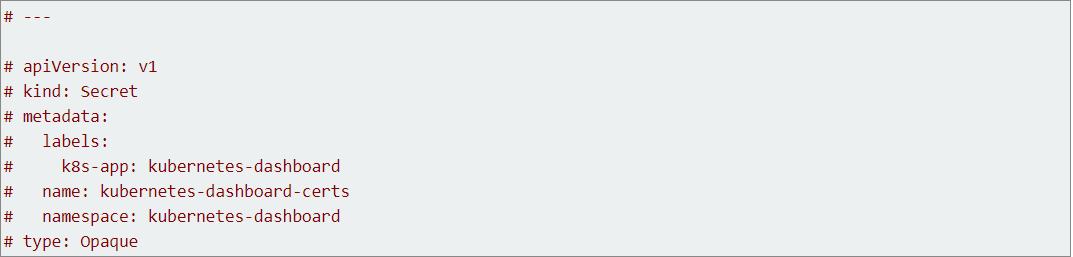
Mặc định dịch vụ này không truy cập trực tiếp qua IP máy mà phải qua proxy, mở file dashboard-v2-beta6.yaml, tìm đến phần kind: Service bên trong có tên name: kubernetes-dashboard sửa thành:



Do cấu hình mặc định của Kubernetes Cluster, cổng được mở ra ngoài phải chọn trong khoảng 30000-32767 sau này bạn có thể sửa cấu hình này với tham số chạy Cluster --service-node-portrange

**Xóa Secret có tên kubernetes-dashboard-certs**

Đoạn cấu hình này để khởi tạo Secret cấu hình xác thực SSL khi truy cập Dashboard, ta xóa nó đi bằng cách comment như sau:



Triển khai các thành phần định nghĩa, cấu hình trong dashboard-v2-beta6.yaml, bạn thực hiện lệnh: kubectl apply -f dashboard-v2-beta6.yaml

Tạo kubernetes-dashboard-certs, xác thực SSL

Ta sẽ dùng OpenSSL để sinh certificates tự xác thực SSL (ngoài ra khi triển khai thực tế có thể lấy các certificate do mua hoặc miễn phí từ https://letsencrypt.org/ theo Domain), chạy các lệnh sau:

*sudo mkdir certs*

*sudo chmod 777 -R certs*

*openssl req -nodes -newkey rsa:2048 -keyout certs/dashboard.key -out certs/dashboard.csr -subj "/C=/ST=/L=/O=/OU=/CN=kubernetes-dashboard"*

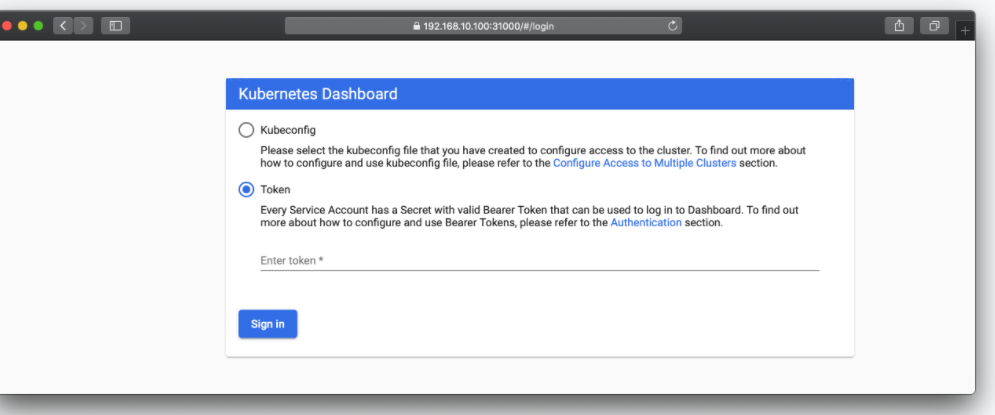
*openssl x509 -req -sha256 -days 365 -in certs/dashboard.csr -signkey certs/dashboard.key -out certs/dashboard.crt*

*sudo chmod -R 777 certs*

Sau khi hoàn thành thì các thông tin certificate lưu ở thư mục certs, chạy lệnh sau để tạo Secret

*kubectl create secret generic kubernetes-dashboard-certs --from-file=certs -n kubernetes-dashboard*

Màn hình khi đăng nhập



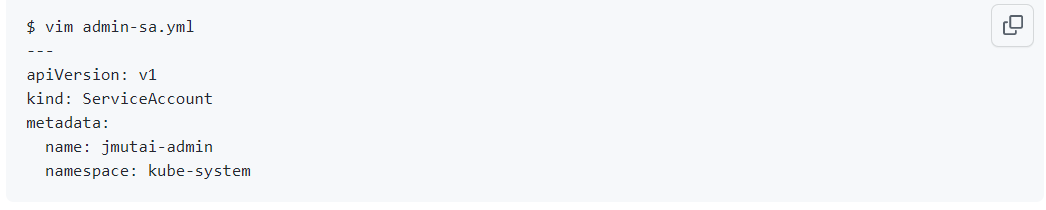
Lấy Token đăng nhập Dashboard

Create Admin User để truy cập Kubernetes Dashboard

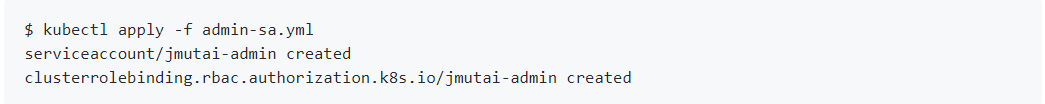
Phần này sẽ thảo luận về cách tạo người dùng admin để có quyền truy cập vào tất cả các tài nguyên Kubernetes. Người dùng admin có thể sửa đổi tất cả các đối tượng trong tất cả các namespace cũng như quản lý bất kỳ thành phần nào khác trong một cụm cluster.

Bước 1: Tạo tài khoản Admin service

Hãy bắt đầu bằng cách tạo file mainfest tài khoản dịch vụ. Và sẽ đặt tên cho tài khoản dịch vụ là jmutai-admin.

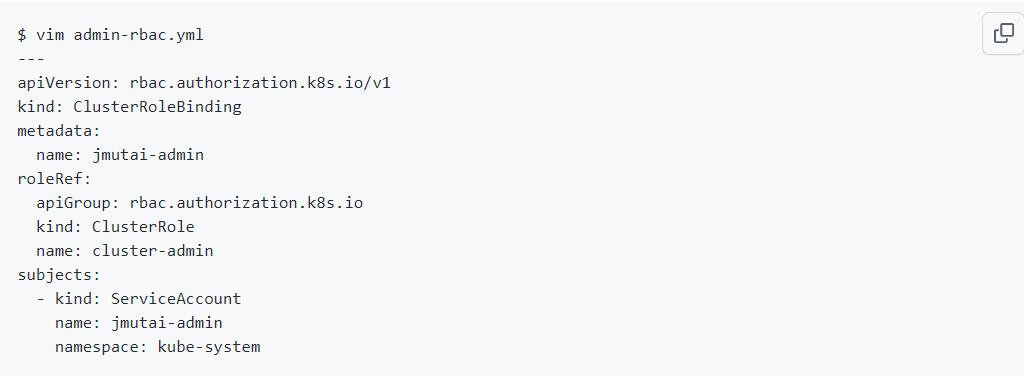


Trong đó jmutai-admin là tên của tài khoản dịch vụ được tạo. Sau khi tạo tệp, hãy áp dụng mainfesr để tạo đối tượng trong kubernetes cluster.



Bước 2: Tạo ràng buộc role cluster

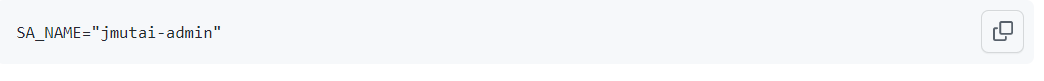
Tiếp theo là gán tài khoản service đã tạo ràng buộc role cluster của cluster-admin.



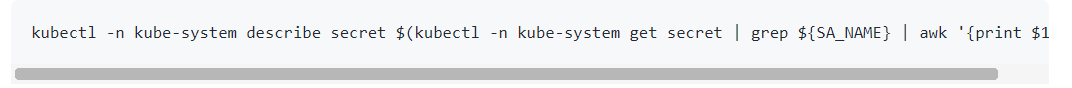
Bước 3: Lấy Token người dùng admin

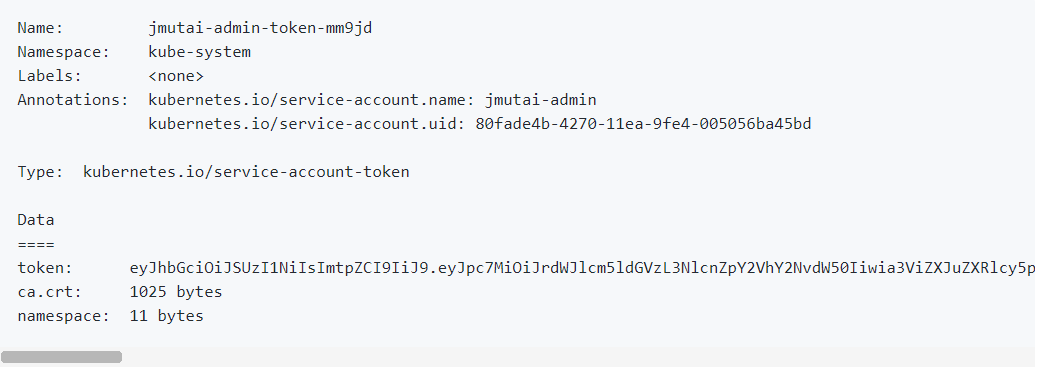
Bạn có thể in mã token đã tạo cho tài khoản service bằng cách sử dụng lệnh kubectl.

Đặt một biến để lưu trữ tên của tài khoản service.



Sau đó chạy lệnh bên dưới để in mã Token cho người dùng admin đã tạo.





Sao chép nội dung trong khóa mã Token vừa lấy được và truy cập kubernetes dashboard

1. **Deployment trong Kubernetes**

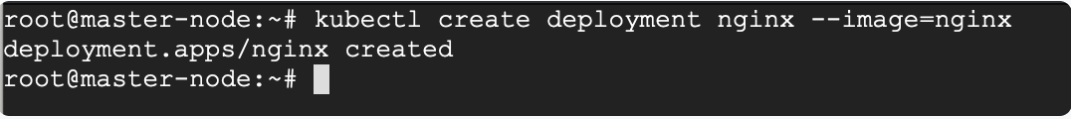
Deployment quản lý một nhóm các Pod - các Pod được nhân bản, nó tự động thay thế các Pod bị lỗi, không phản hồi bằng pod mới nó tạo ra. Như vậy, deployment đảm bảo ứng dụng của bạn có một (hay nhiều) Pod để phục vụ các yêu cầu.

Deployment sử dụng mẫu Pod (Pod template - chứa định nghĩa / thiết lập về Pod) để tạo các Pod (các nhân bản replica), khi template này thay đổi, các Pod mới sẽ được tạo để thay thế Pod cũ ngay lập tức.

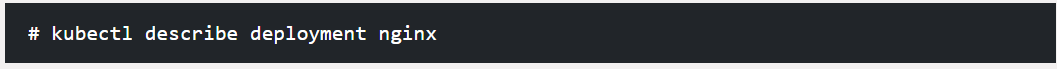
**Cách deploy Nginx trên kubernetes**

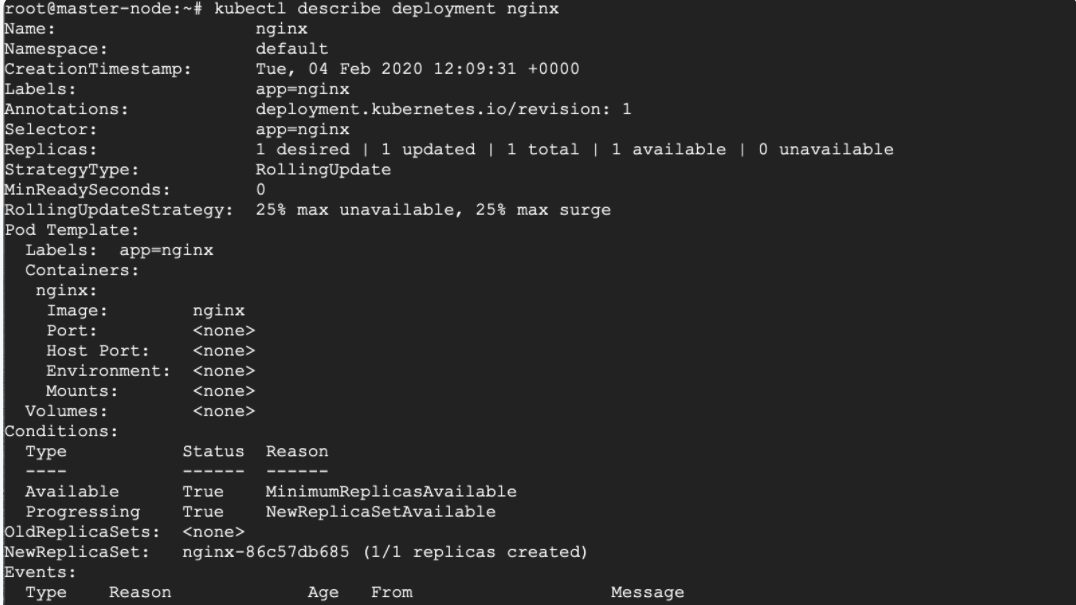
Tạo một deployment của Nginx sử dụng Nginx image





Xem chi tiết deployment vừa tạo

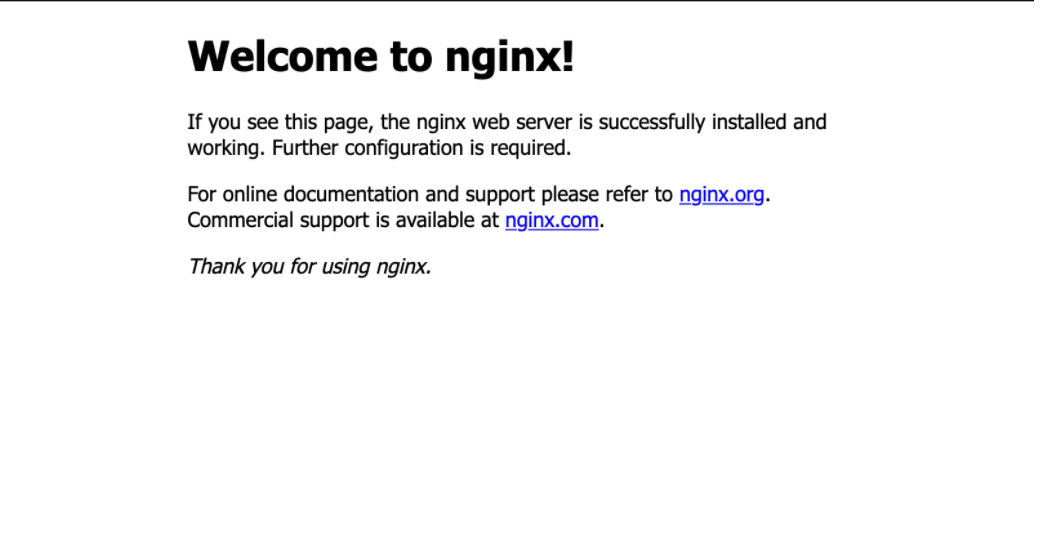




Sử dụng NortPort service cho deployment

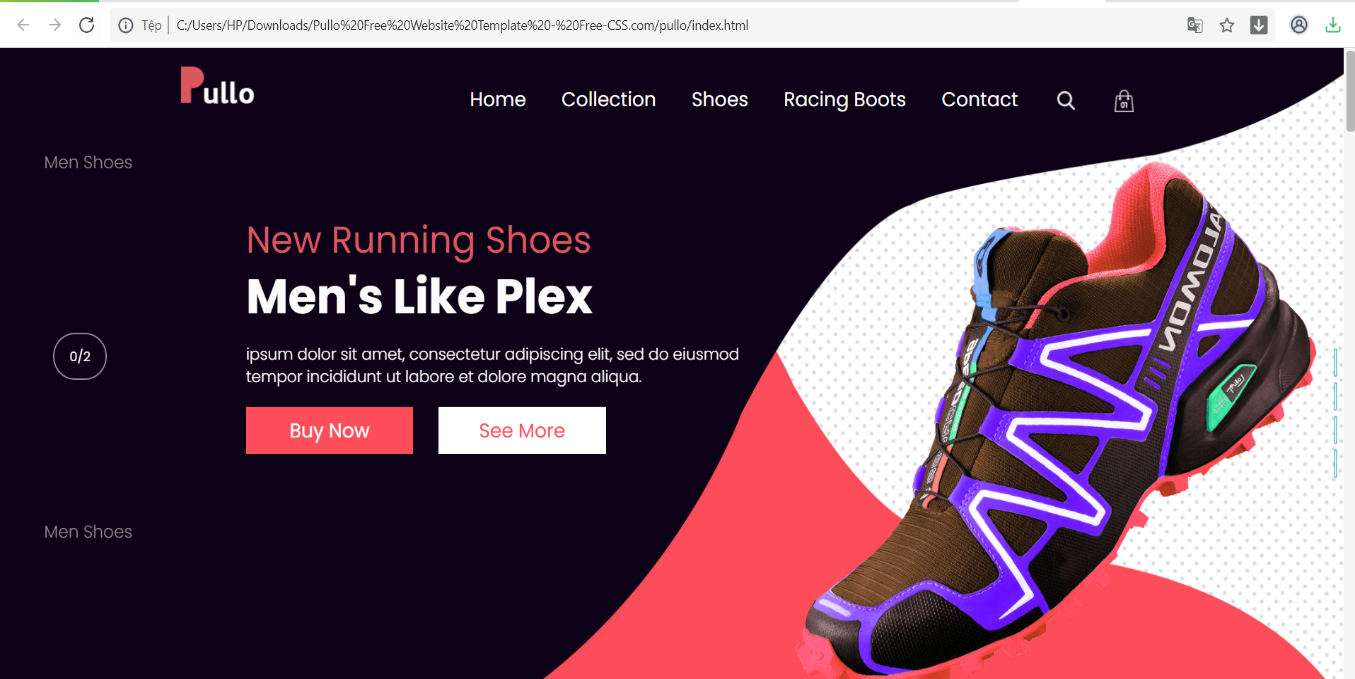


Chạy dịch vụ trên browers



**Cách deploy image trên kubernetes**

Chuẩn bị source code một website đơn giản

****

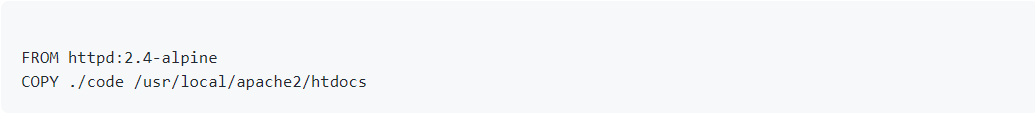
Thực hiện lệnh để trỏ vào thư mục source code

*cd webdemodtdm/*

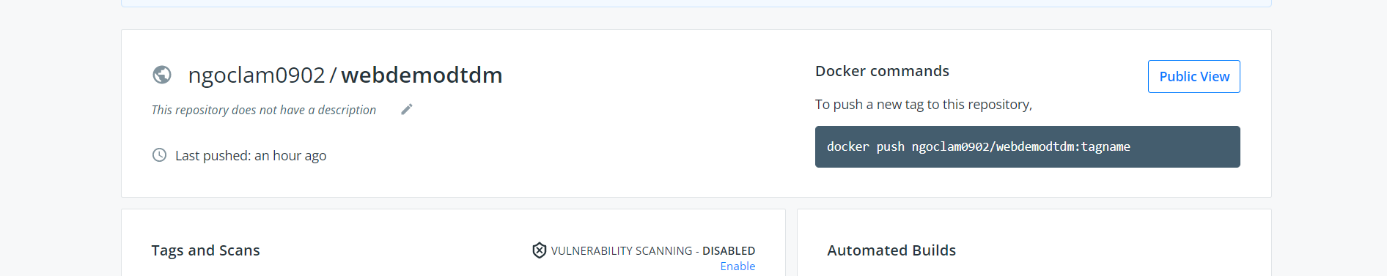
Tạo file Dockerfile

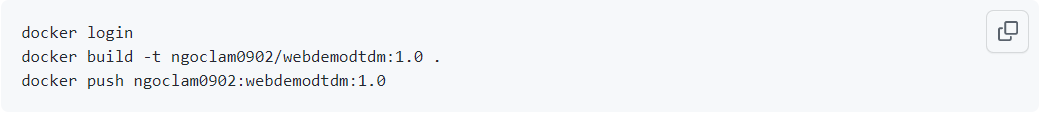
*vim Dockerfile*

Gõ 2 dòng lệnh sau sau đó lưu file

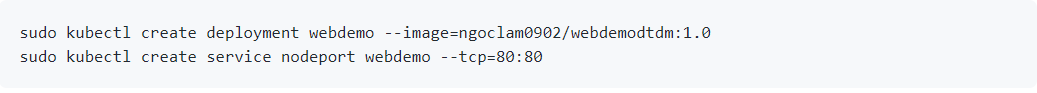


Thực hiện build và push lên dockerhub (ta đã có sẵn 1 repositories)

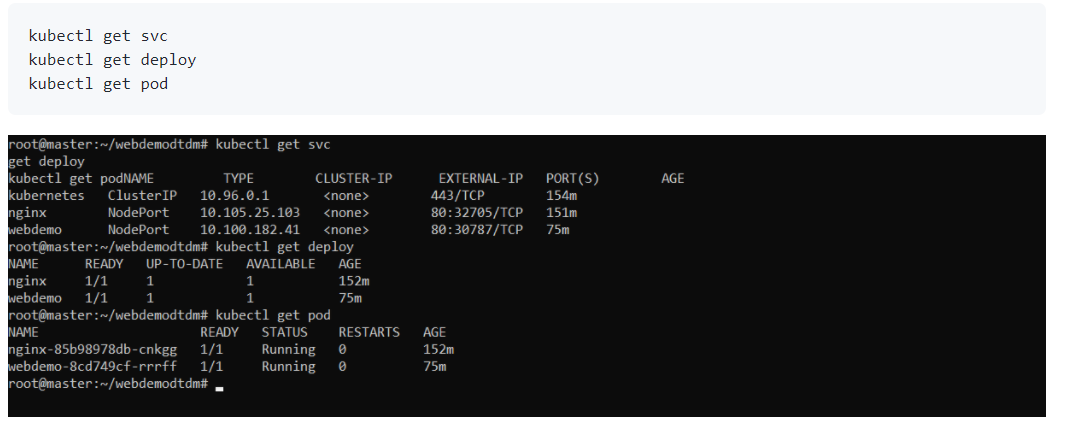




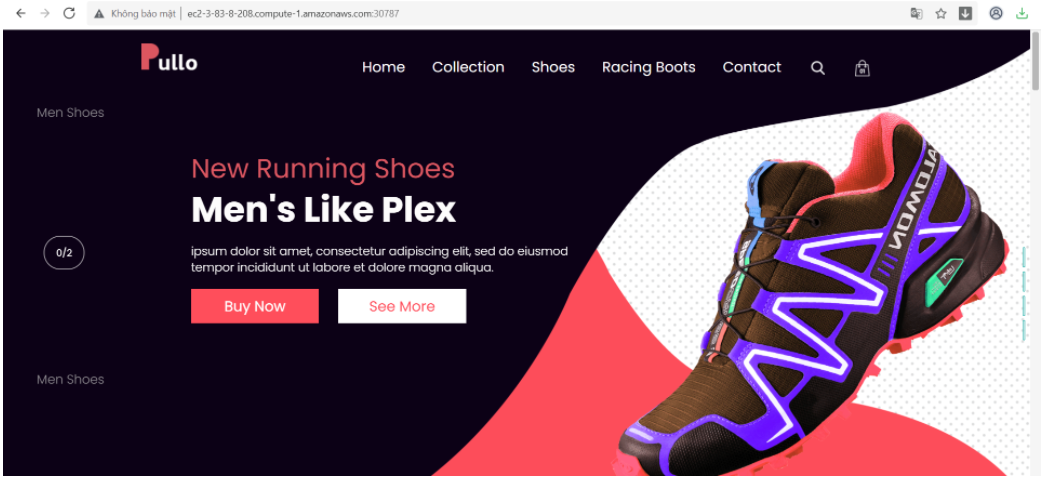
Thực hiện tạo service và deployment



Kiểm tra deployment, pod và service



Chạy web trên trình duyêt



**KẾT LUẬN**

Cài đặt được Kubernetes cluster , hiểu được về deployment và cách deploy Nginx web server

Cài đặt được Kubernetes Dashboard

Kubernetes là một nền tảng nguồn mở, khả chuyển, 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 hoá 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.

Với những tính năng ưu việt và vượt trội, tích hợp Kubernetes Engine có thể mang lại những tác động tích cực đến các Dev team/DevOps cũng như toàn bộ doanh nghiệp.

Kubernetes không phải là một hệ thống PaaS (Nền tảng như một dịch vụ) truyền thống, bao gồm tất cả. Vì Kubernetes hoạt động ở cấp vùng chứa chứ không phải ở cấp phần cứng, nó cung cấp một số tính năng áp dụng chung phổ biến cho các dịch vụ PaaS, chẳng hạn như triển khai, mở rộng quy mô, cân bằng tải và cho phép người dùng tích hợp các giải pháp ghi nhật ký, giám sát và cảnh báo

**TÀI LIỆU THAM KHẢO**

<https://topdev.vn/blog/kubernetes-la-gi/#9684>

<https://viblo.asia/p/phan-1-gioi-thieu-ve-kubernetes-924lJO6m5PM>

<https://xuanthulab.net/gioi-thieu-va-cai-dat-kubernetes-cluster.html>

<https://xuanthulab.net/cai-dat-va-su-dung-kubernetes-dashboard.html>

<https://xuanthulab.net/deployment-trong-kubernetes-trien-khai-cap-nhat-va-scale.html>

<https://github.com/justmeandopensource/kubernetes/blob/master/docs/install-cluster-ubuntu-20.md>