

|  |
| --- |
| **BỘ MÔN: ĐIỆN TOÁN ĐÁM MÂY**  **----------** |
| **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*

**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.

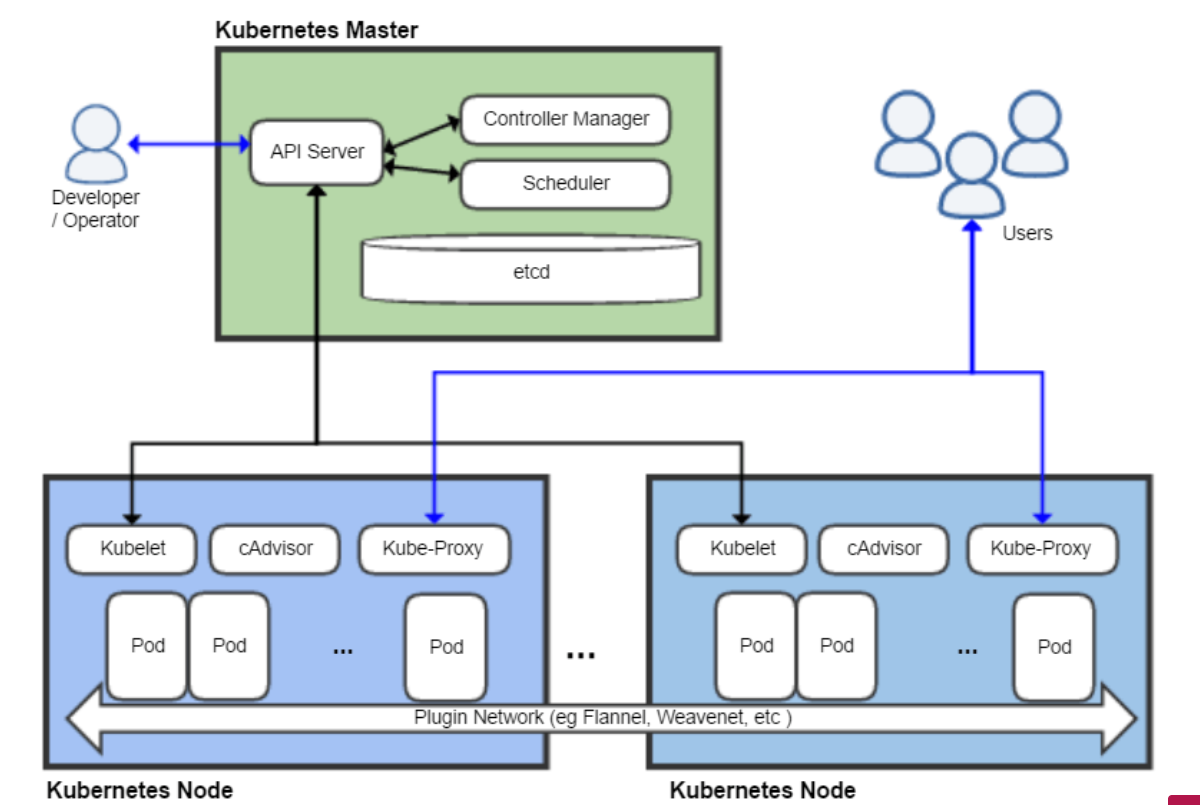
1. **Các sản phẩm dựa trên Kubernetes**

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.

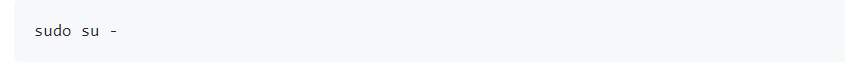
kube-controller chạy tại master, nó điều khiển trạng thái cluster, tương tác để thực hiện các tác vụ tạo, xóa, cập nhật ... các tài nguyên

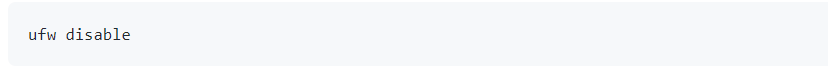
Kubelet dịch vụ vụ chạy trên tất cả các máy (Node), nó đảm đương giám sát chạy, dừng, duy trì các ứng dụng chạy trên node của nó.

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

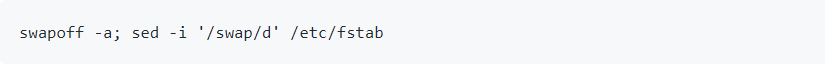
1. **Cách cài đặt Kubernetes**

Đăng nhập với tư cách rootngười dùng

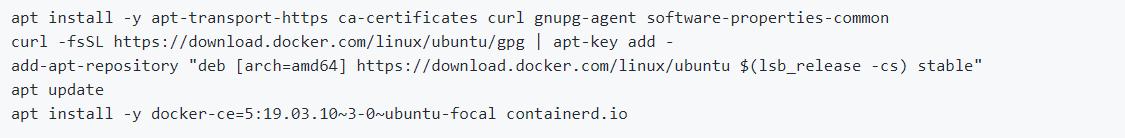
****Tắt tường lửa



Vô hiệu hóa hoán đổi

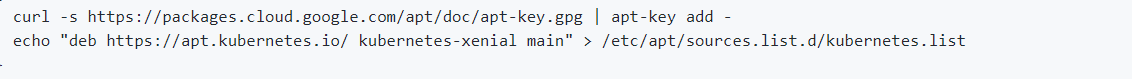


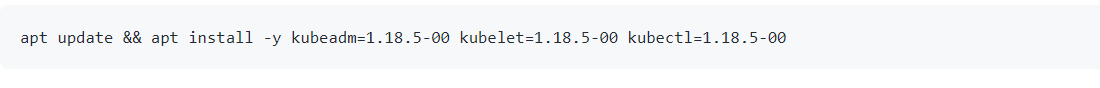
Cài đặt công cụ docker

****

Thiết lập Kubernetes

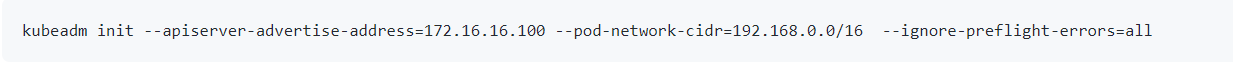
Thêm kho lưu trữ Apt

****Cài đặt các thành phần Kubernetes

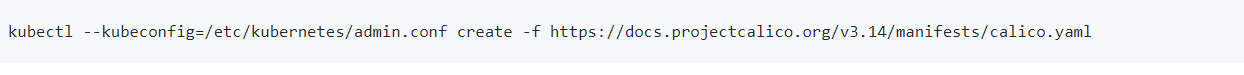


Trên kmaster

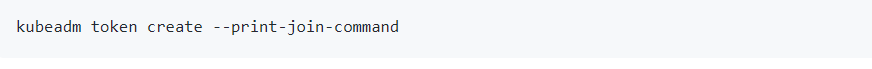
Khởi tạo Kubernetes Cluster



Triển khai mạng lưới Calico

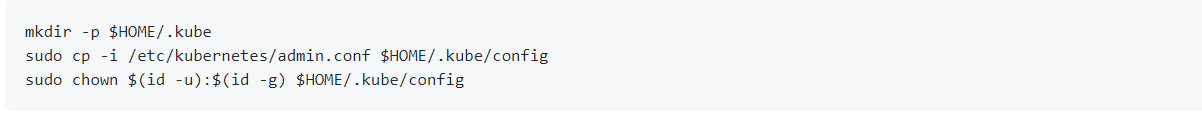


Lệnh tham gia cụm

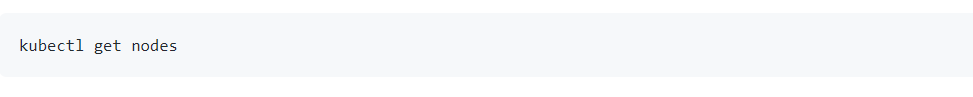
****

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

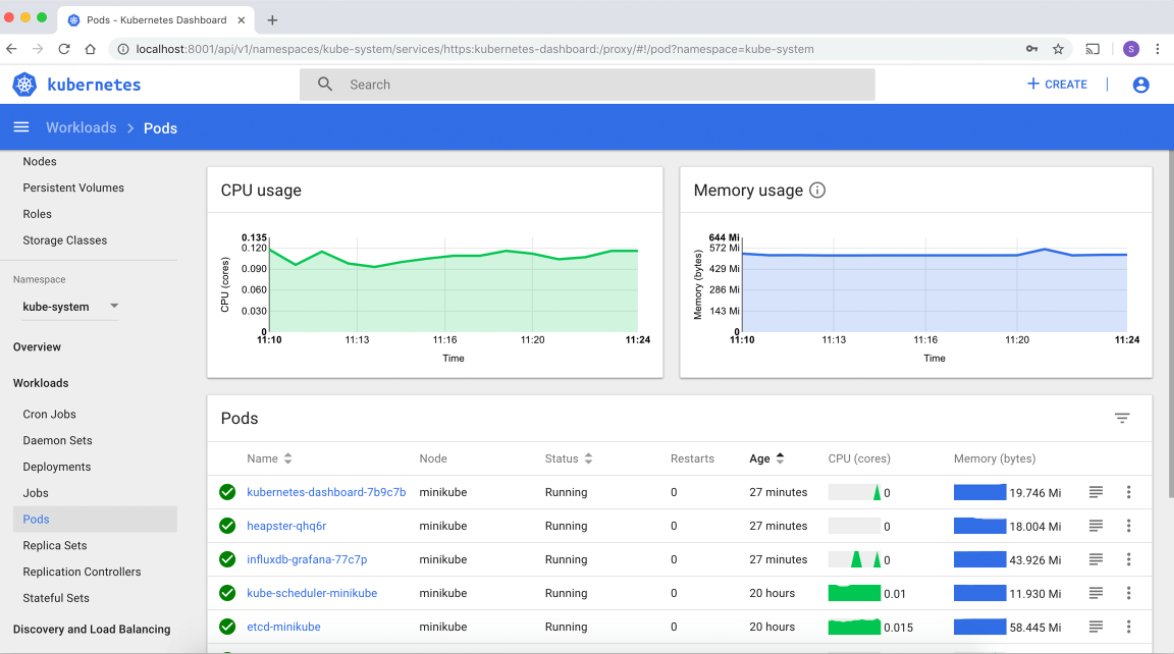
****Trên Kworker

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



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.

1. **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>