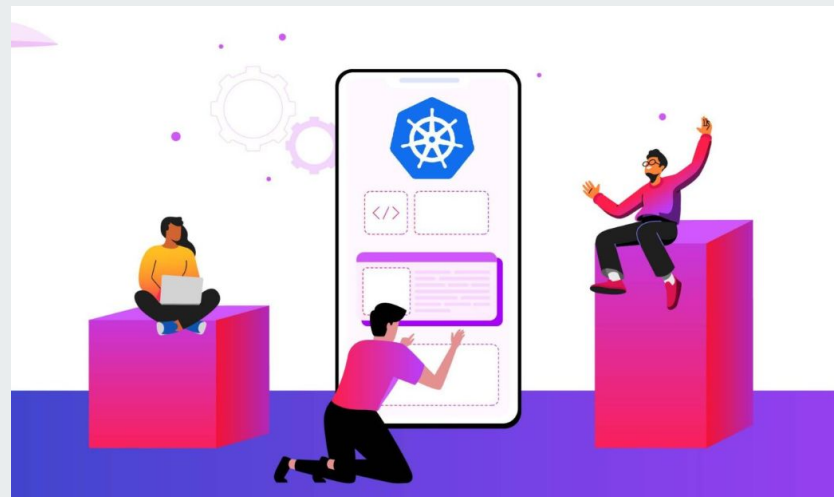


Kubernetes 1

BASIC, INSTALLATION, CONFIGURATION AND ARCHITECTURE

caodangsao@gmail.com



Learning Paths



Kubernetes Basics

1

Kubernetes overview

- Học về các thành phần cơ bản của Kubernetes
- Cài đặt và cấu hình cụm Kubernetes
- Kiến trúc Kubernetes



2

Pods

- Pods và Containers trong Kubernetes
- Phân phối Pods trong Kubernetes



3

Deployments

- Scaling ứng dụng trong Kubernetes
- Replicaset trong Kubernetes
- Deployments trong Kubernetes



4

Networking and Services

- K8s networking overview
- CNI Plugins, Kubernetes DNS và Kubernetes Network Policies
- Services trong Kubernetes



5

Storage

- K8s Storage overview
- Sử dụng K8s Volume
- Sử dụng K8s Persistent Volumes và Persistent Volumes Claim



1-5

Ôn tập

- Triển khai ứng dụng lên cụm Kubernetes



Kubernetes Advanced

6

Security

- Bảo mật cụm bằng TLS
- Role based access control
- Sử dụng Security context và network policies



7

Helm

- K8s Storage overview
- Sử dụng K8s Volume
- Sử dụng K8s Persistent Volumes và Persistent Volumes Claim



Kubernetes Learning Objectives

Học xong phần này, học viên sẽ nắm được những điều sau:

- Lịch sử và sự phát triển của Kubernetes.
- Kiến trúc và thành phần tổng quan của nó.
- API, các tài nguyên quan trọng nhất tạo nên API và cách sử dụng chúng.
- Cách triển khai và quản lý một ứng dụng sử dụng Kubernetes.
- Có khả năng quản lý cụm Kubernetes.
- Sử dụng HELM để đóng gói và triển khai ứng dụng.

Module Target

Kết thúc bài học, học viên cần đạt được các kỹ năng sau:

- Có hiểu biết về nguồn gốc, tầm quan trọng của Kubernetes
- Nắm được các khái niệm về Kubernetes, các thành phần quan trọng của nó
- Có khả năng cài đặt, cấu hình được cụm Kubernetes phục vụ mục đích học tập hay nhu cầu sử dụng thực tế
- Khả năng tương tác ở mức độ cơ bản với cụm Kubernetes

Nội dung

1 Getting Started

- Tại sao sử dụng Kubernetes
- Tổng quan kiến trúc Kubernetes
- Cài đặt cụm Kubernetes sử dụng minikube

2 Kubernetes cluster management

- Tổng quan quản lý cụm Kubernetes
- Tính sẵn sàng của cụm trong K8s
- Các công cụ quản lý K8s

3 Summary

- Tóm tắt nội dung buổi học

Lab 1: Minikube

Lab 2: Tương tác Kubernetes

Lab 3: Cụm Kubernetes HA

Lab 4: Triển khai ứng dụng đơn giản



Section 1:

Getting Started

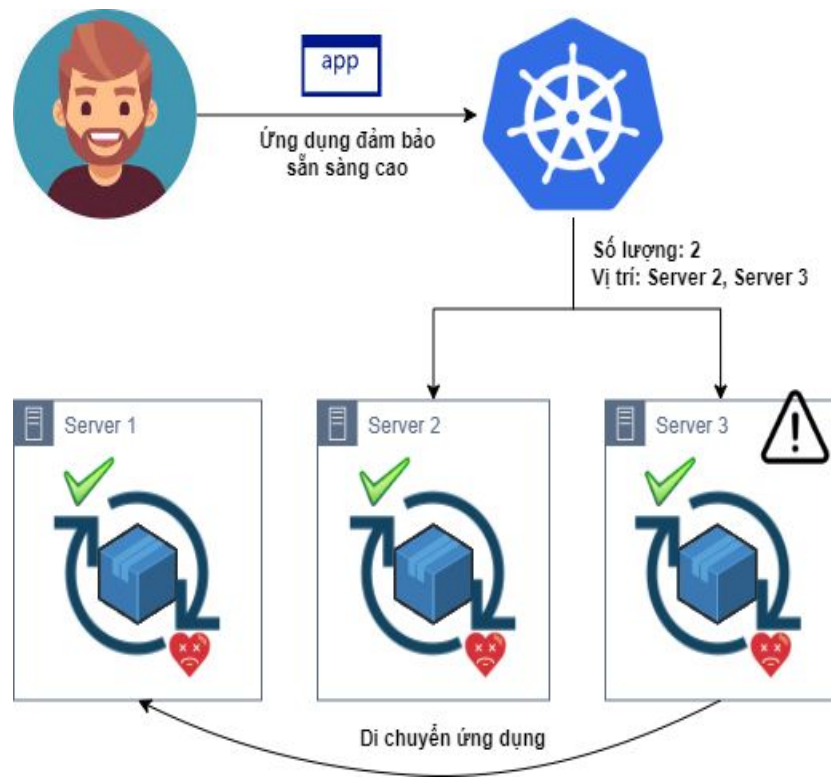
Kubernetes Introduction

- Kubernetes là hệ thống **điều phối containers**, nó là **Open Source**.
- Kubernetes được tạo ra bởi **Google**.
- Kubernetes là một nền tảng loại bỏ các quy trình thủ công tham gia vào việc **triển khai ứng dụng trong container**.

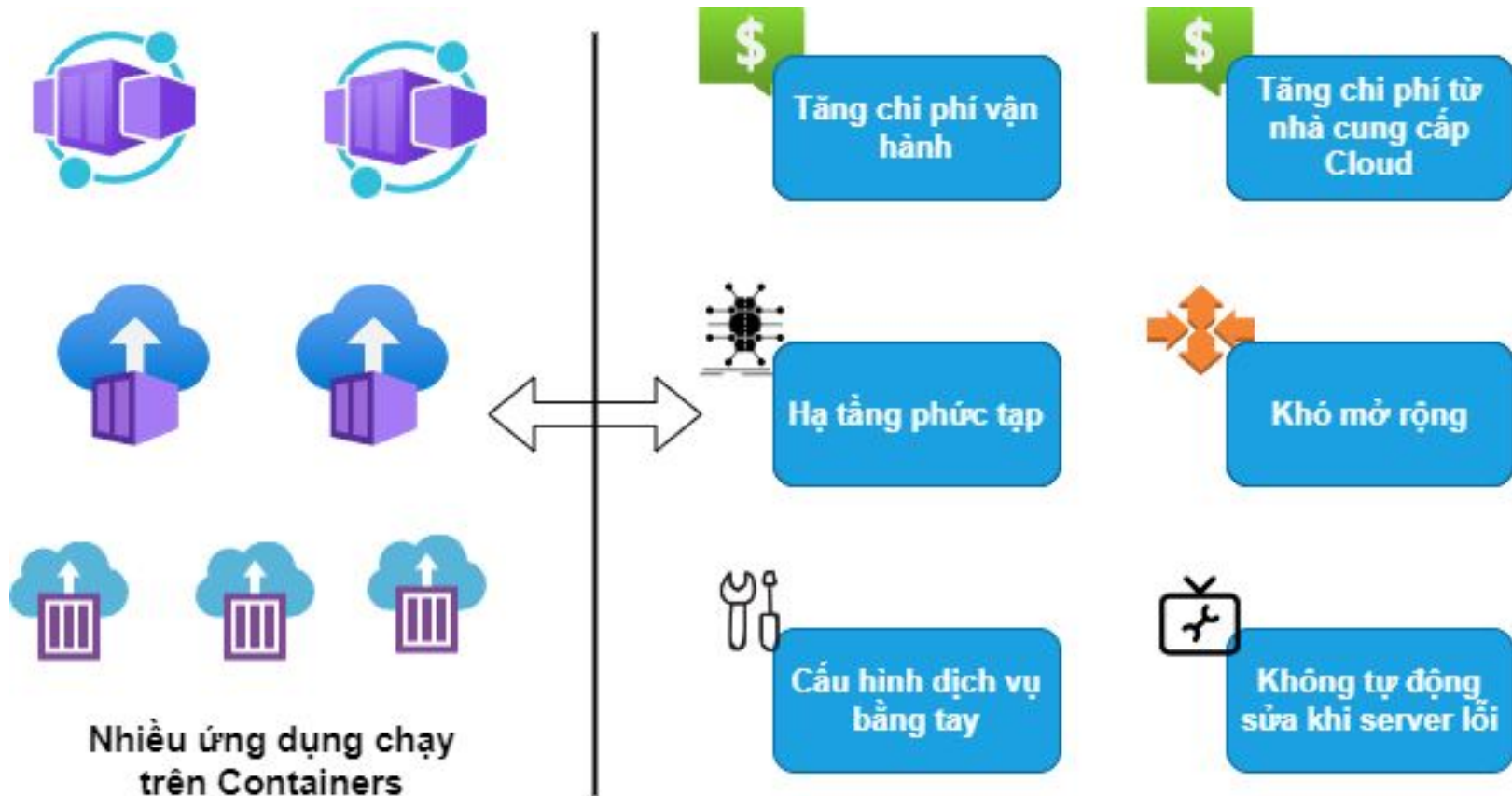


Kubernetes Introduction

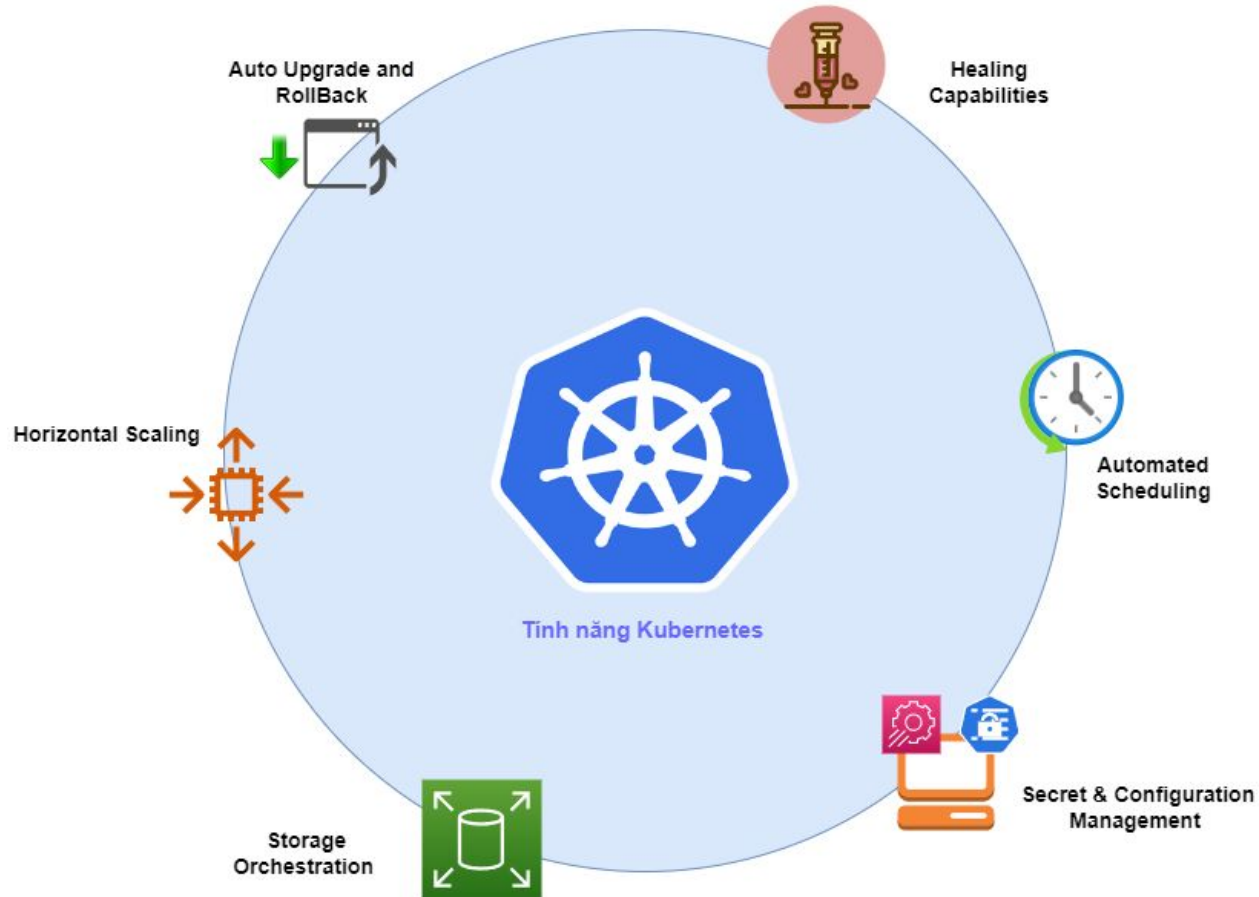
- ➔ Kubernetes sử dụng để quản lý **Trạng thái của Containers**:
 - ◆ Khởi tạo Containers trên server cụ thể.
 - ◆ Khởi động lại Containers khi nó bị chết.
 - ◆ Di chuyển containers giữa các server
- ➔ Kubernetes được gọi tắt là **K8s**.



Why Kubernetes



Why Kubernetes



Why Kubernetes

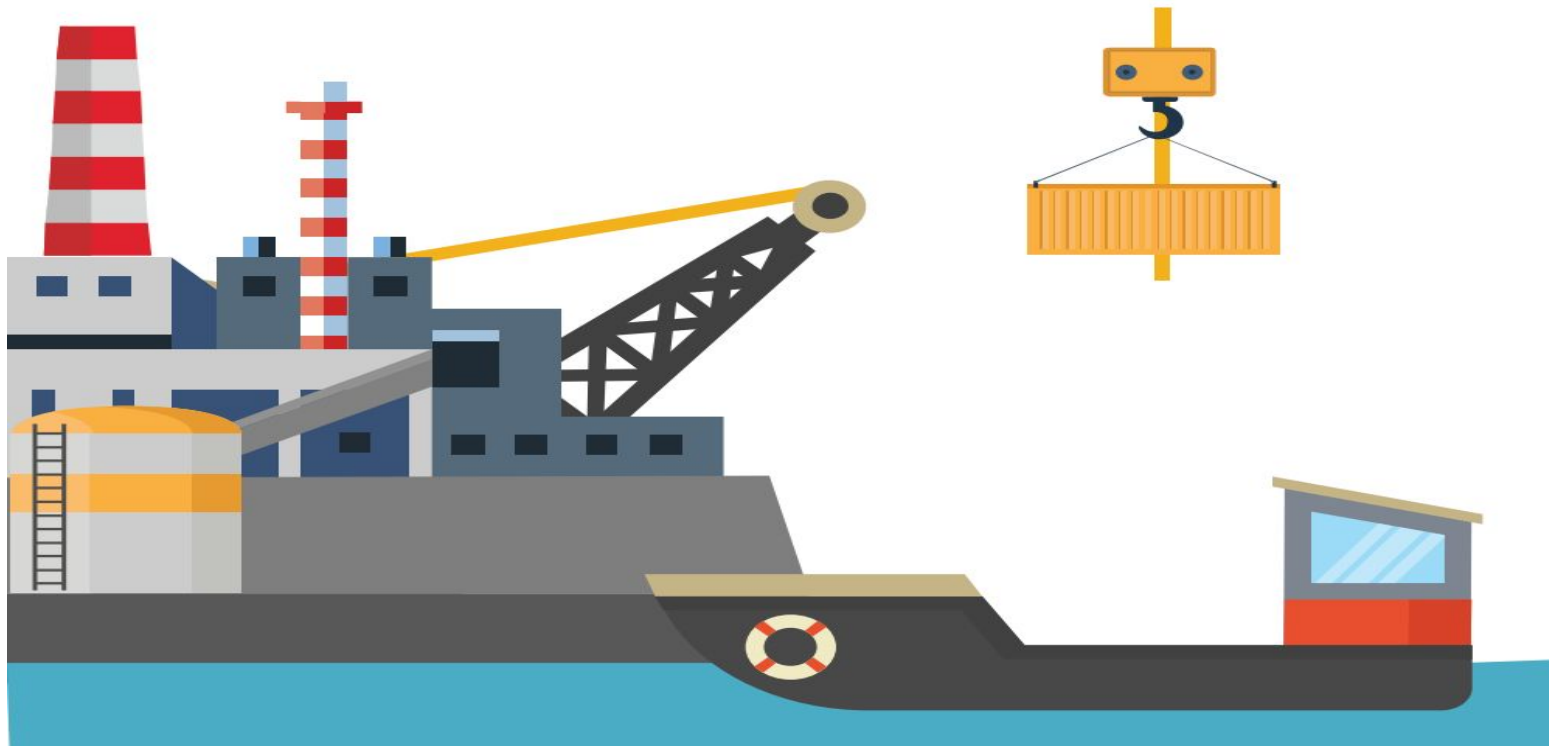


Được sử dụng trong:

- Public cloud
- Private Cloud (Datacenter)
- Hybrid

Kubernetes Architecture Intro

→ Giả sử ta đang ở bến cảng, việc điều phối quản lý và vận chuyển hàng hóa do những đối tượng nào đảm nhiệm ?



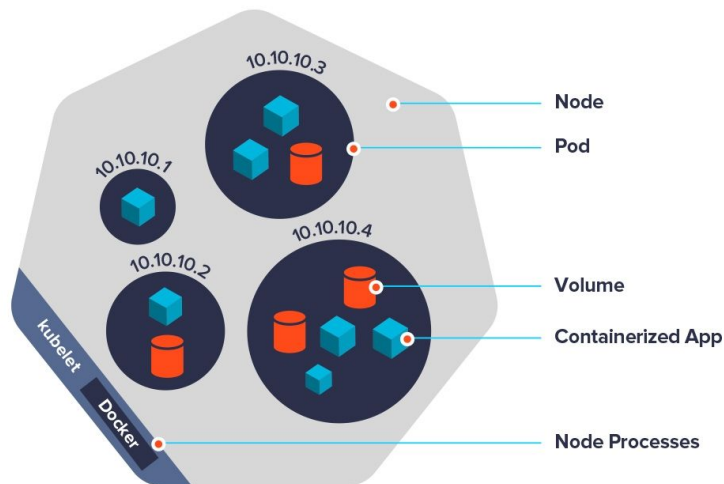
Kubernetes Architecture - Node types



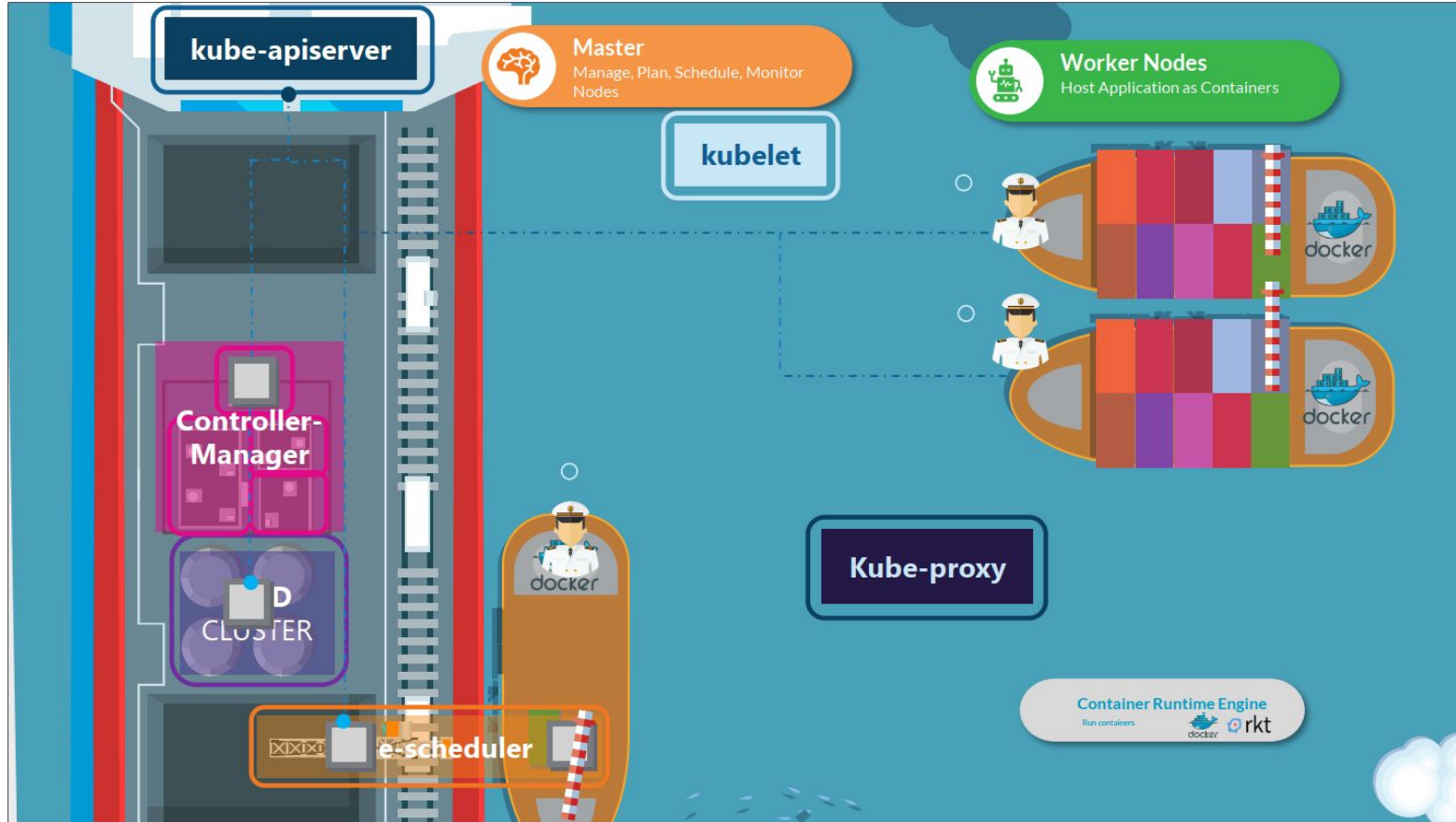
- **Master** hay **Control Planes** nodes: Quản lý, lên kế hoạch, lập lịch, giám sát nodes.
- **Worker** Nodes: Nơi chạy các ứng dụng Containers

Kubernetes Architecture - Pod

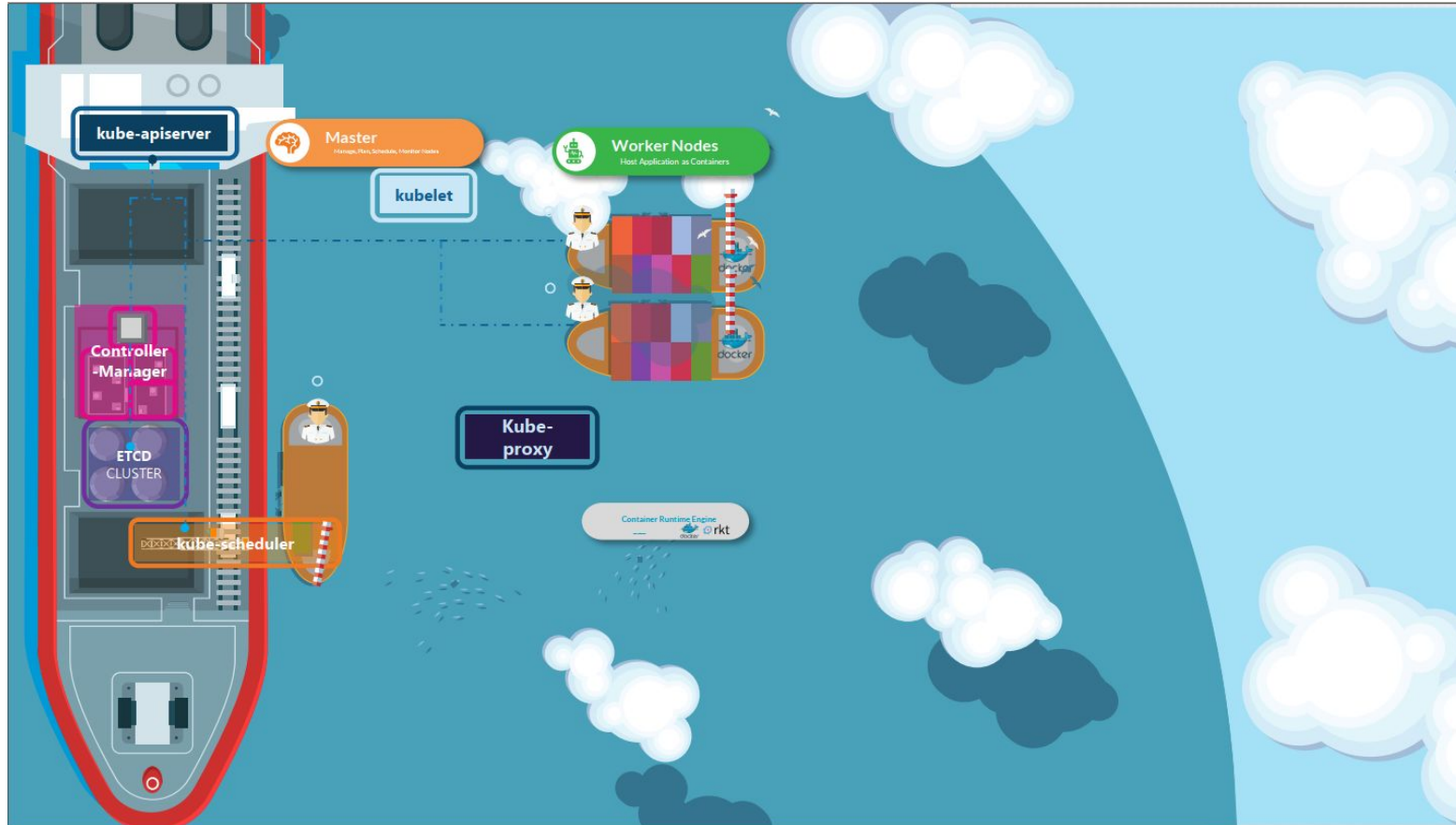
- **Pod** là một hay nhiều **containers sử dụng chung (storage/network)**, và mô tả chi tiết cách chạy các containers này.
- Containers trong Pod **dùng chung storage, địa chỉ IP.**
- Mỗi Pod có thể chạy trên nhiều nodes và mỗi node có thể chạy nhiều pods.



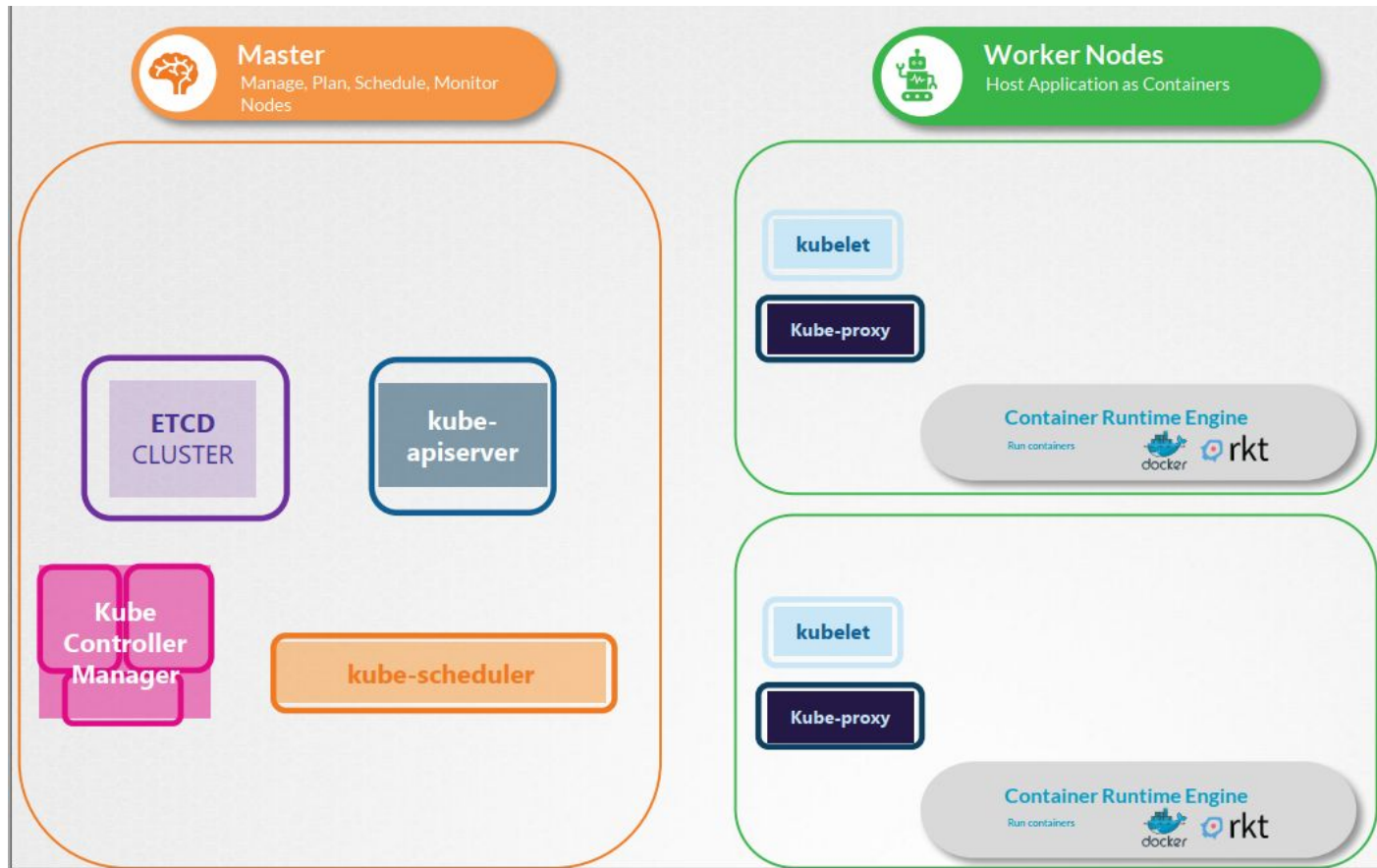
Kubernetes Architecture Overview



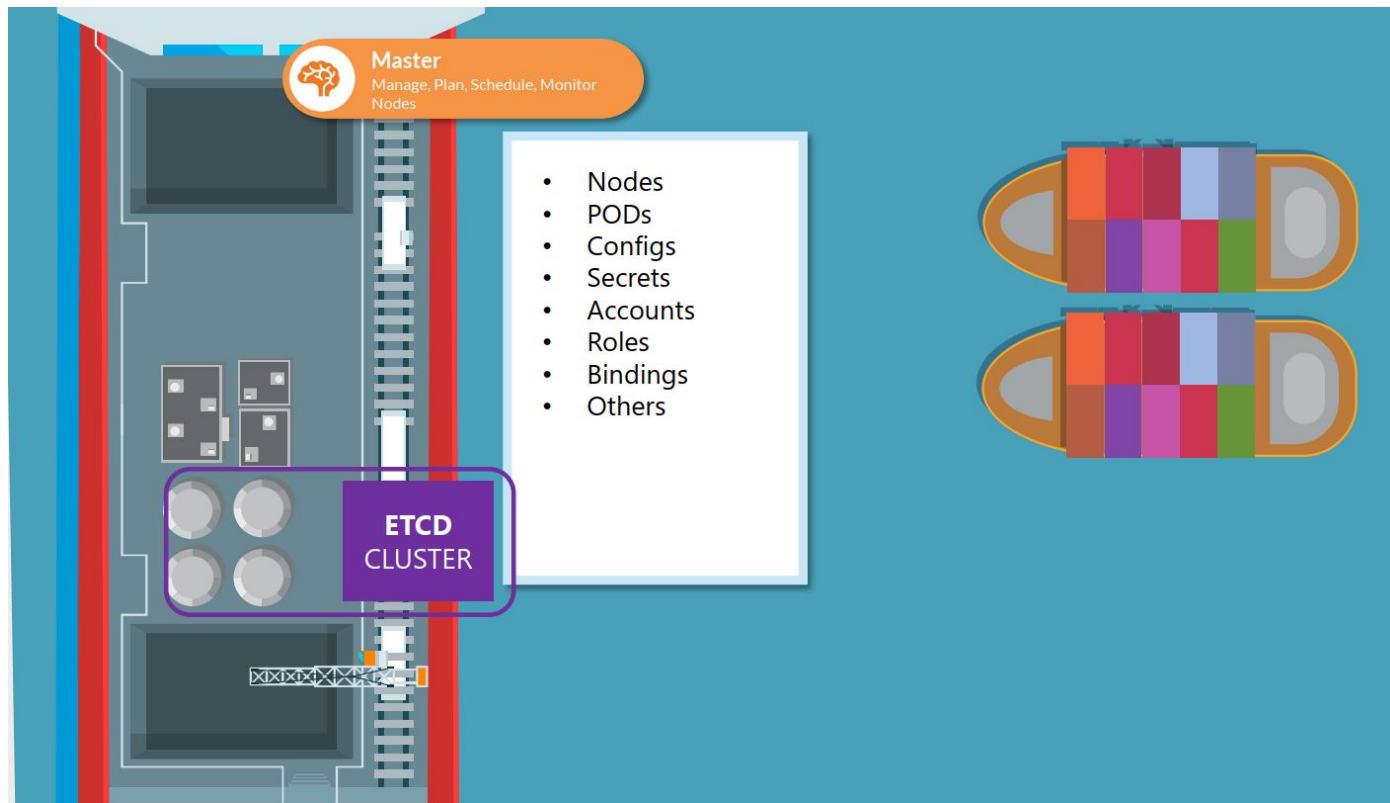
Kubernetes Architecture Overview



Kubernetes Architecture

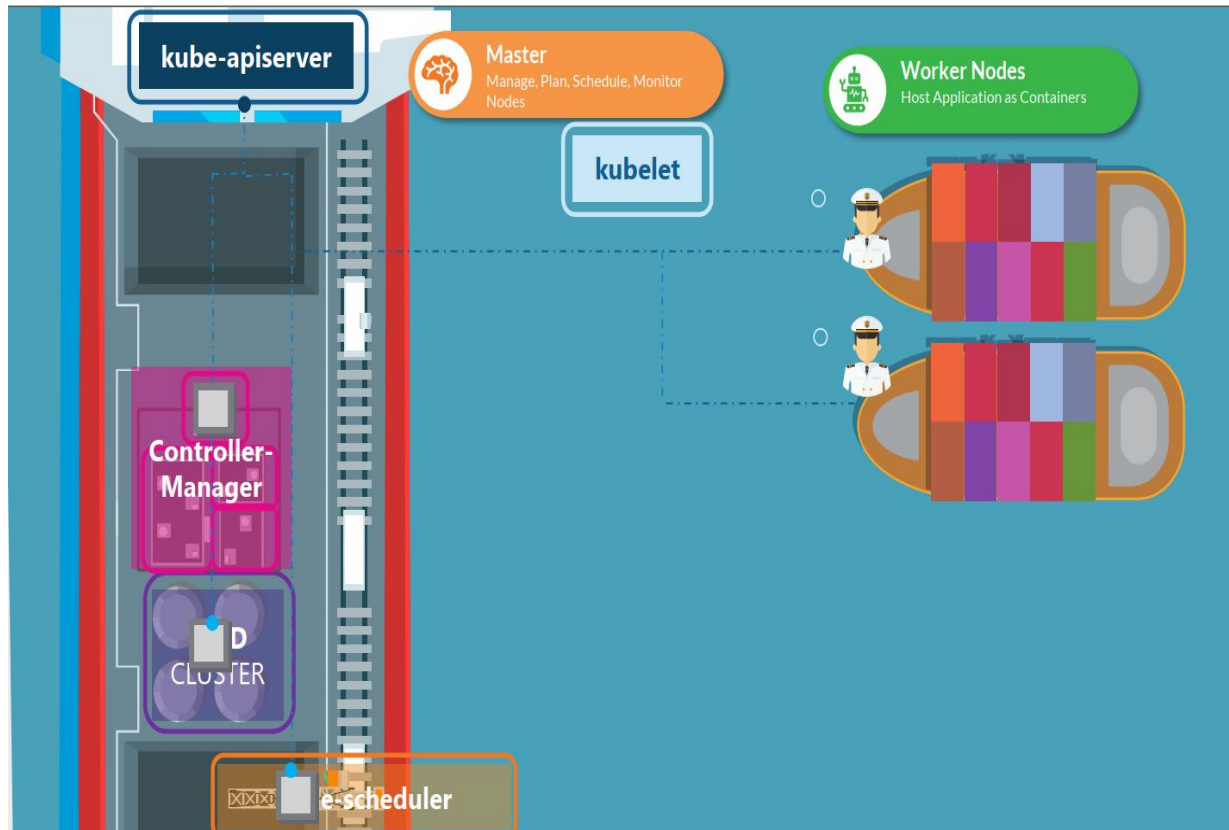


ETCD In Kubernetes



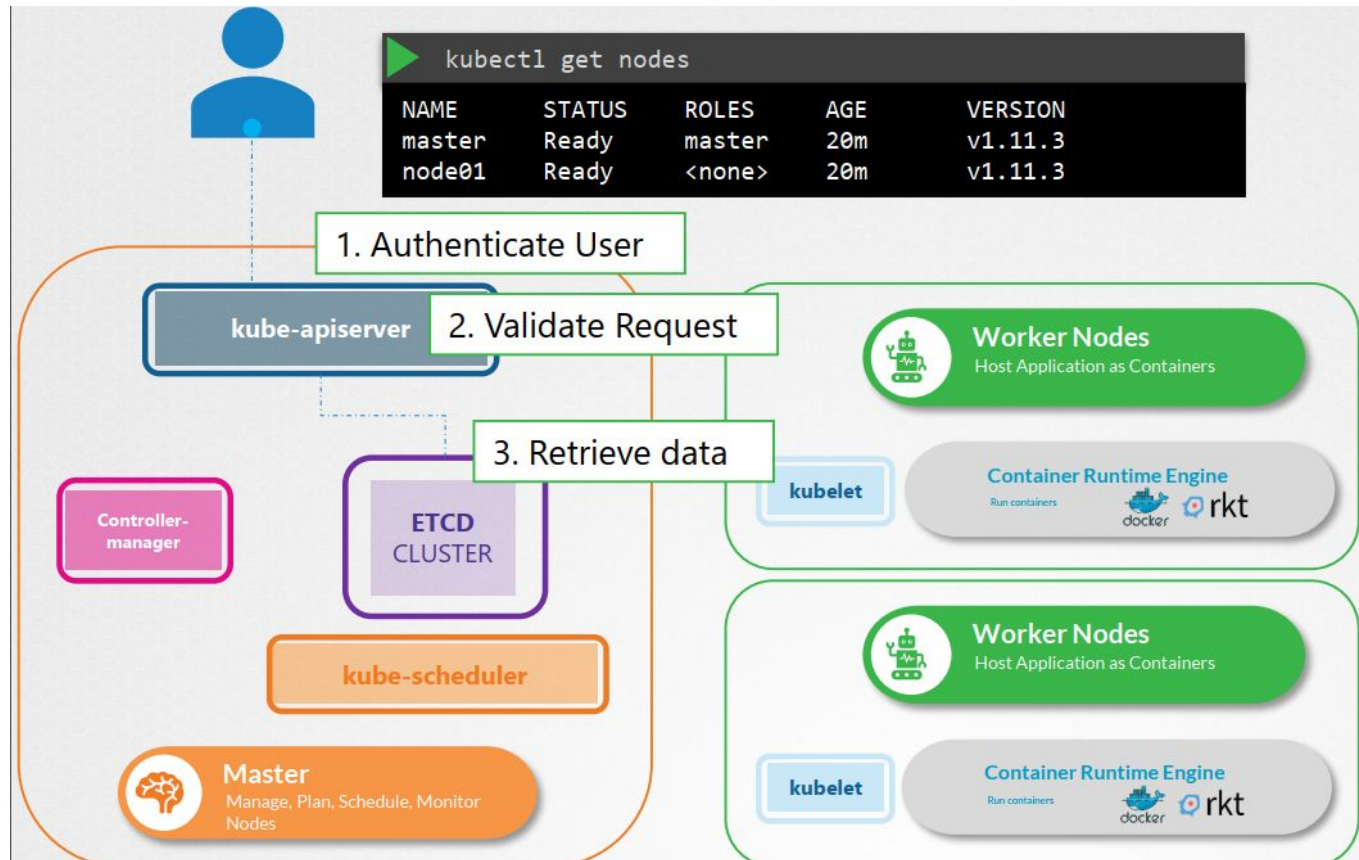
- ★ Lưu trữ **trạng thái của cụm** trong cơ sở dữ liệu phân tán dưới dạng key-value.
- ★ **Back-end** của k8s.
- ★ Cung cấp **tính sẵn sàng cao** của dữ liệu trạng thái của cụm.
- ★ ETCD đáp ứng **đơn giản, bảo mật và nhanh**.

Kube-apiserver



- **Kube-apiserver** là **entry point** cho tất cả các lệnh REST được sử dụng để điều khiển cụm.
- **Điểm tương tác** với Kubernetes.

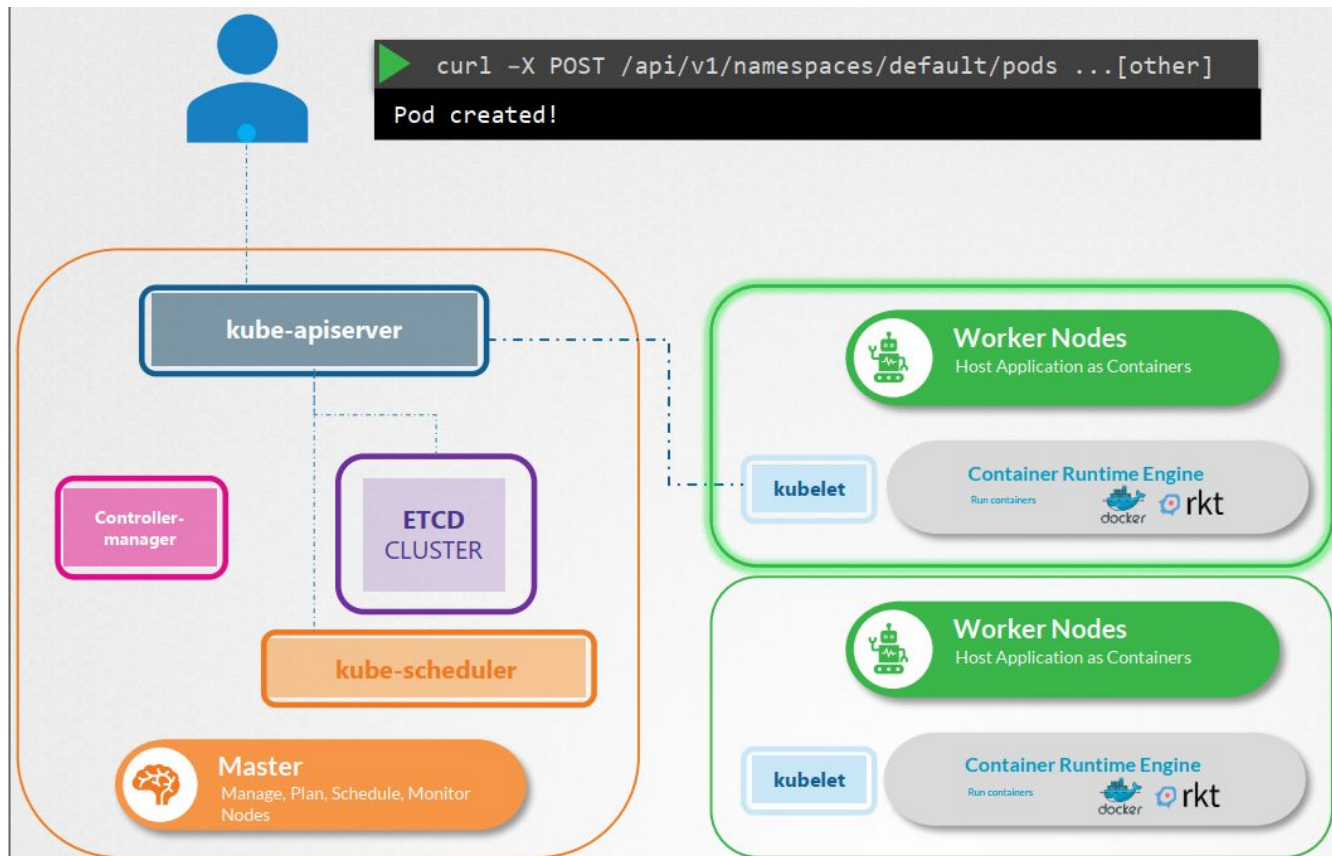
Kube-apiserver



Nhiệm vụ của API server:

1. **Xác thực người dùng**
2. **Xác thực yêu cầu**
3. **Lấy dữ liệu**

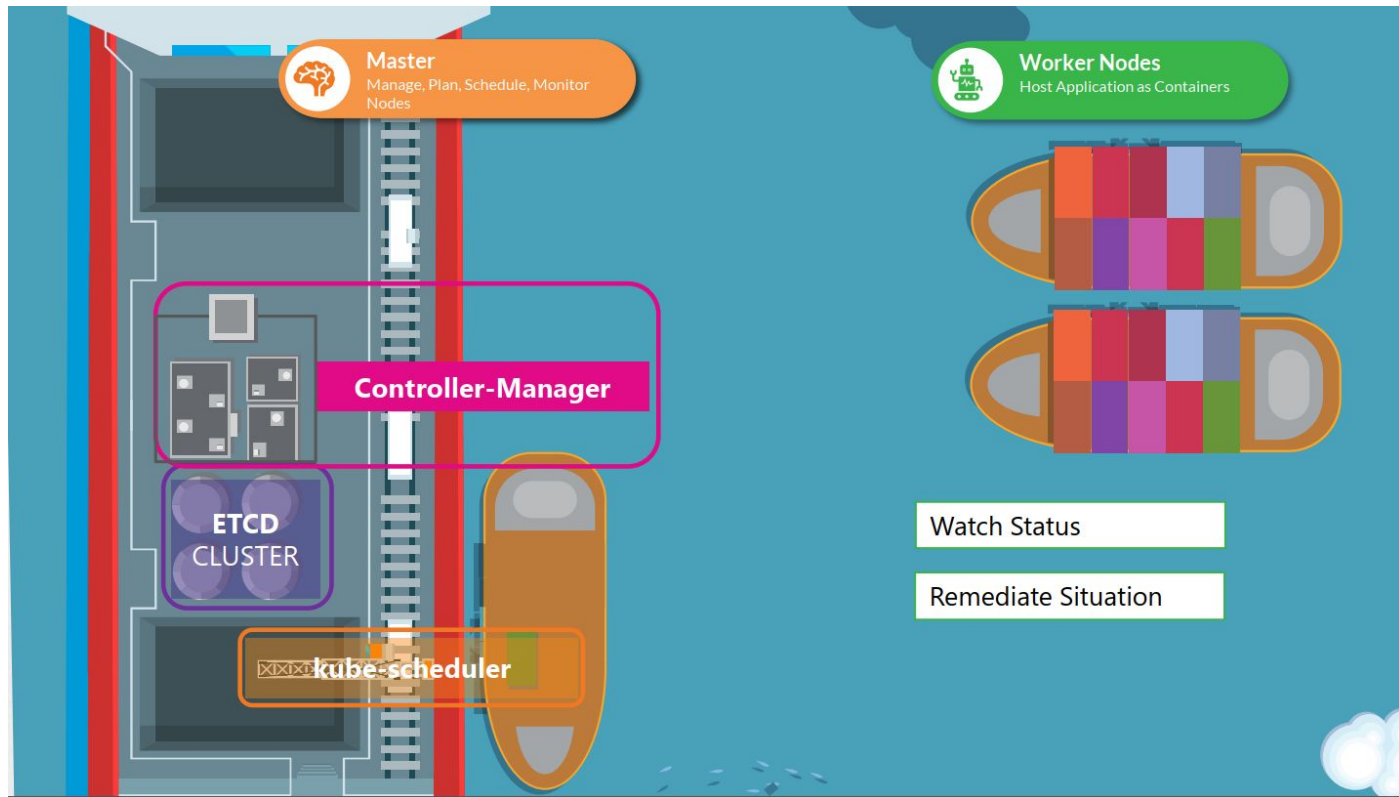
Kube-apiserver



Nhiệm vụ của API server:

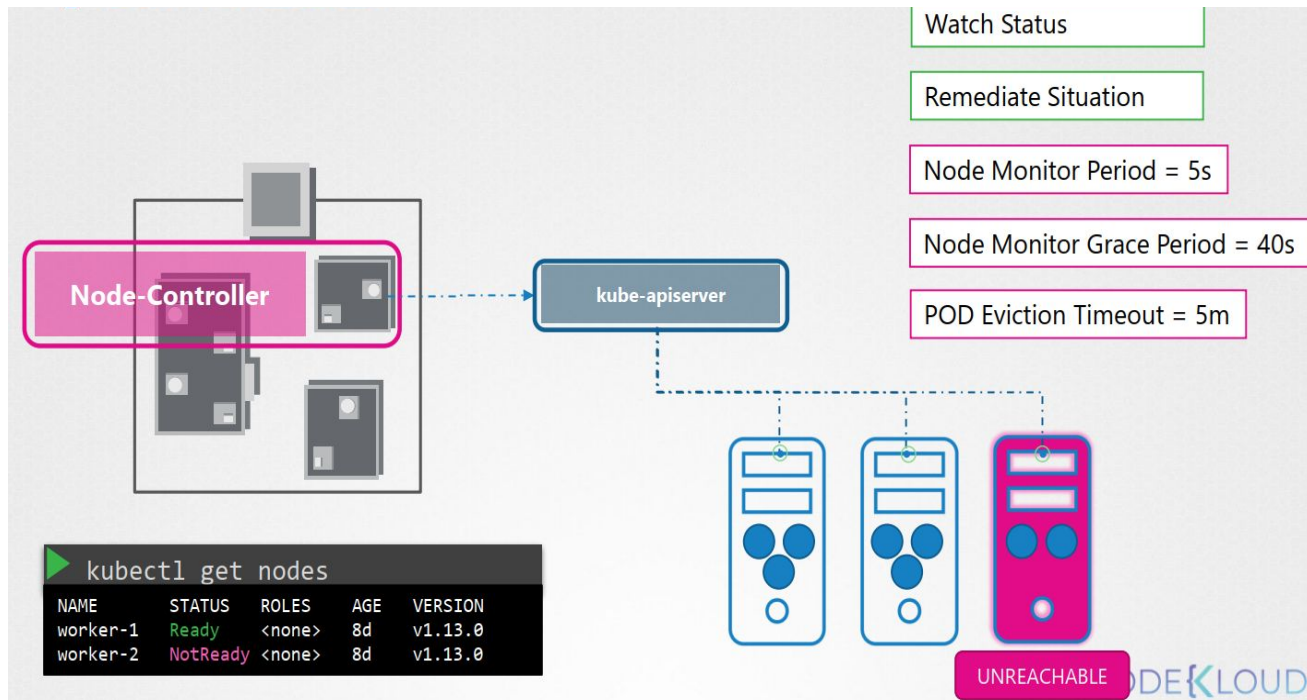
1. **Xác thực người dùng**
2. **Xác thực yêu cầu**
3. **Lấy dữ liệu**
4. **Cập nhật ETCD**
5. **Tương tác Kuber scheduler**
6. **Tương tác kubelet**

Kube Controller Manager



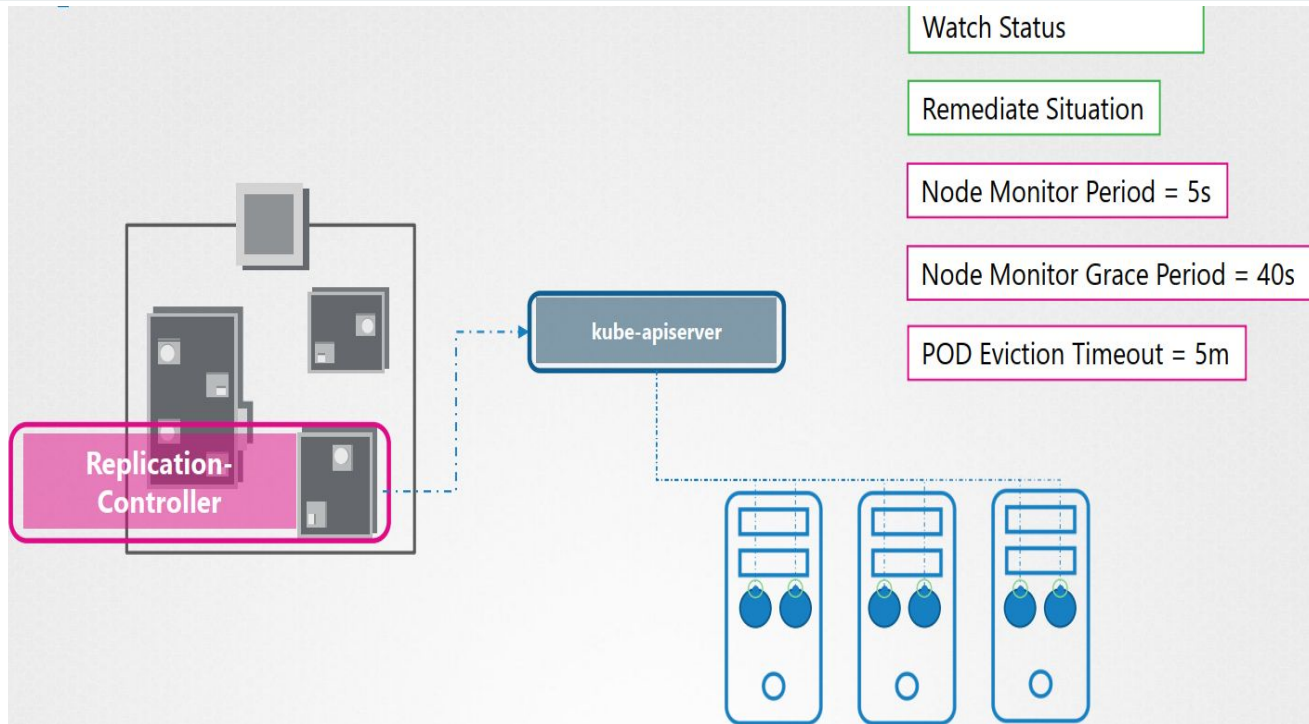
- **Controller** : Chạy nhiều loại controller trong một tiến trình duy nhất.
- Thực hiện các **tác vụ tự động** trong cụm K8s
- Nhiệm vụ chính:
 - ◆ **Theo dõi tình trạng**
 - ◆ **Xử lý sự cố**

Kube Controller Manager



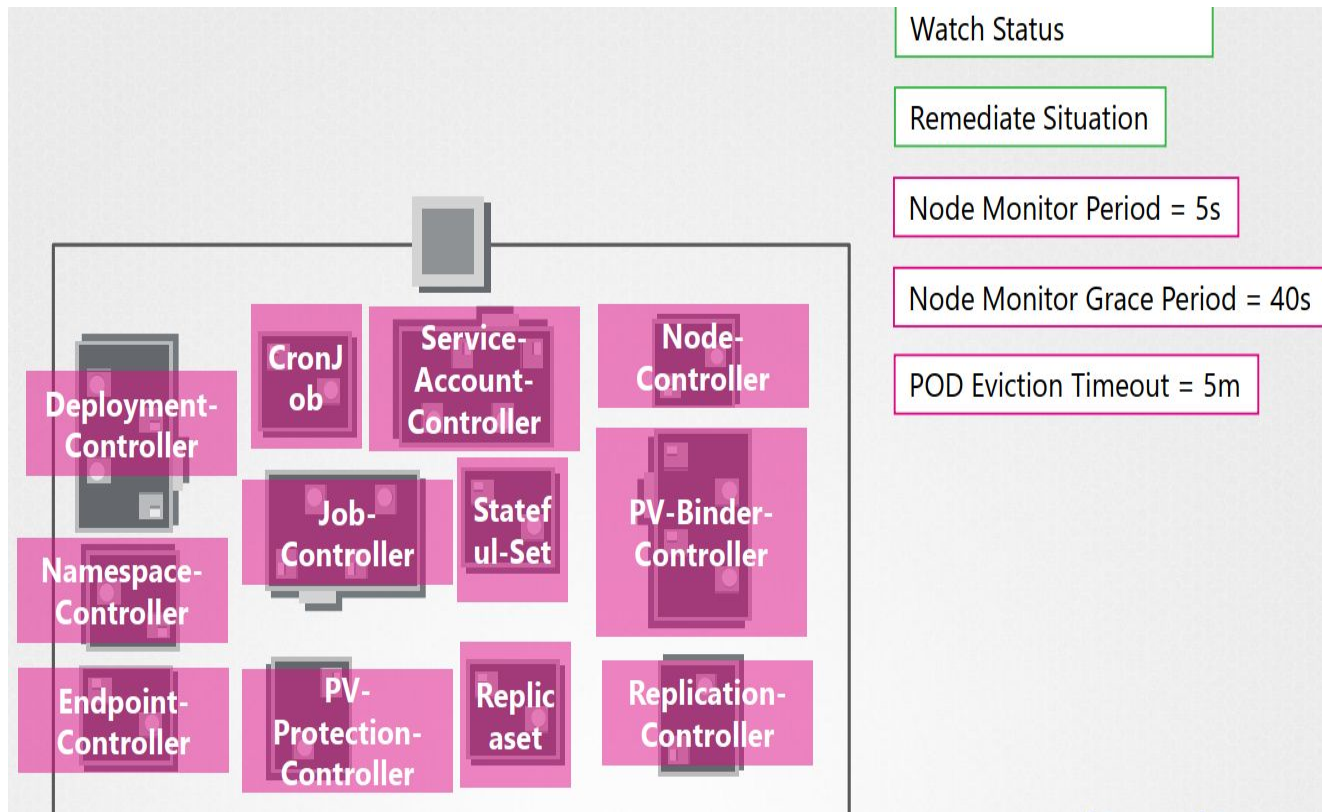
- **Node-Controller** : làm nhiệm vụ điều khiển và giám sát node:
- ◆ Theo dõi tình trạng
 - ◆ Xử lý sự cố

Kube Controller Manager



- **Replication-Controller** :
- làm nhiệm vụ điều khiển và giám sát số lượng Pod:
- ◆ Theo dõi tình trạng
 - ◆ Xử lý sự cố

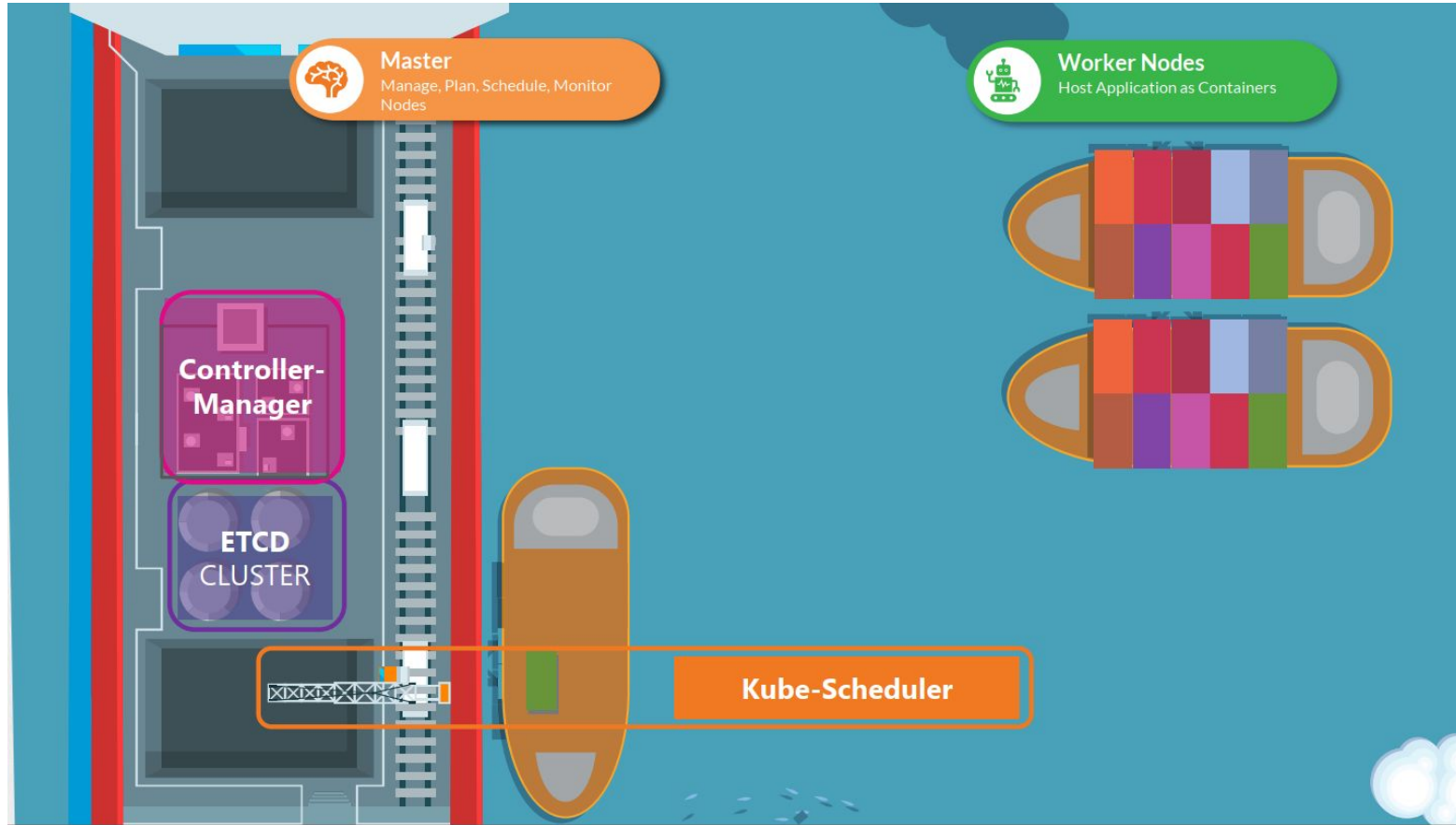
Kube Controller Manager



→ **Nhiều loại controller** làm nhiệm vụ khác nhau:

- ◆ Deployment-controller
- ◆ Job-controller
- ◆ Replicaset
- ◆ ...

Kube Scheduler

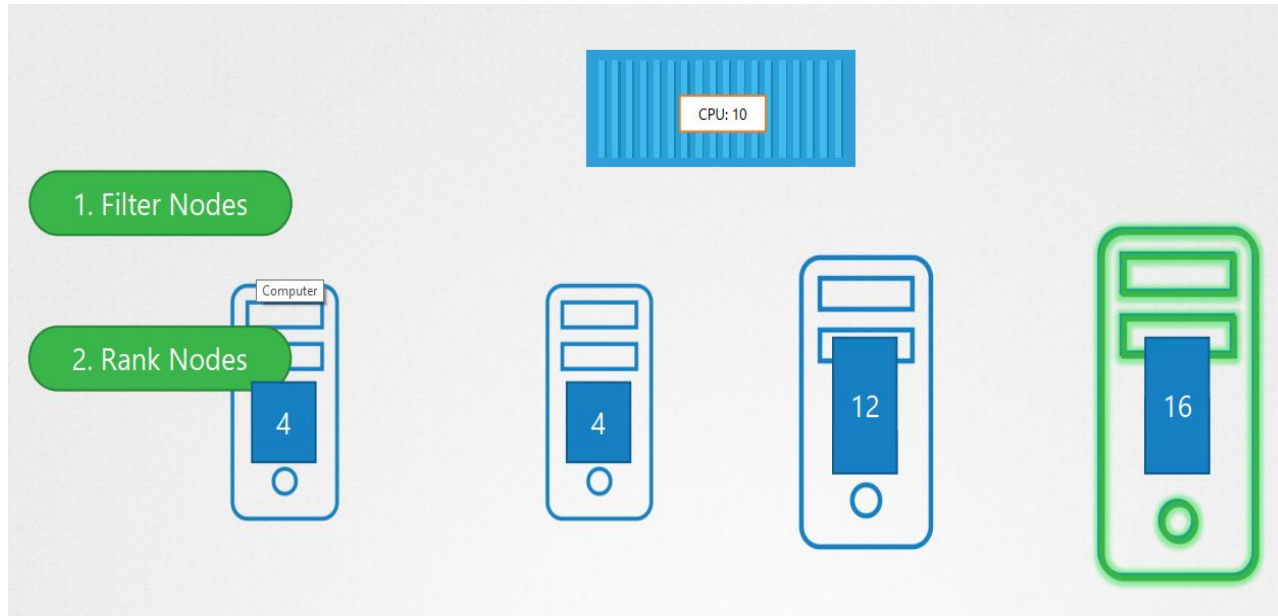


Kube Scheduler

- Đối với mỗi Pod mà Scheduler phát hiện, Scheduler sẽ chịu trách nhiệm tìm Node tốt nhất cho Pod đó để chạy.

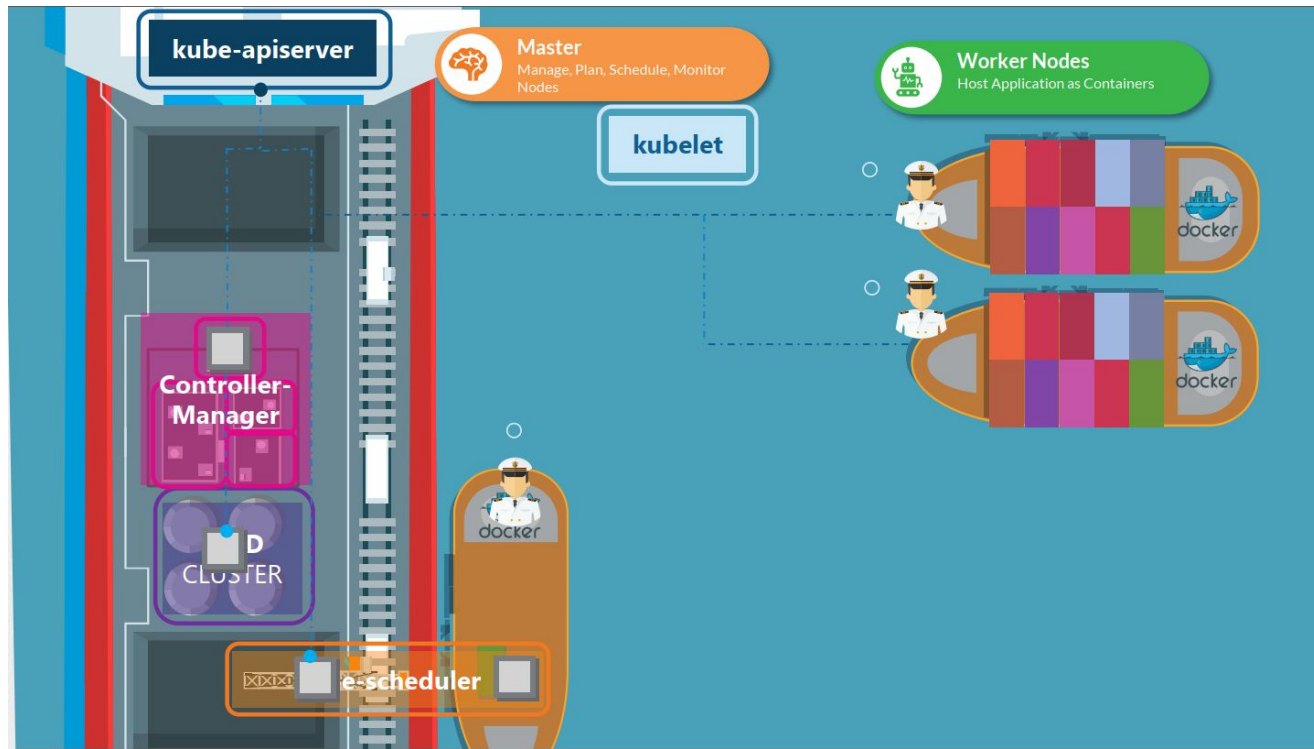


Kube Scheduler



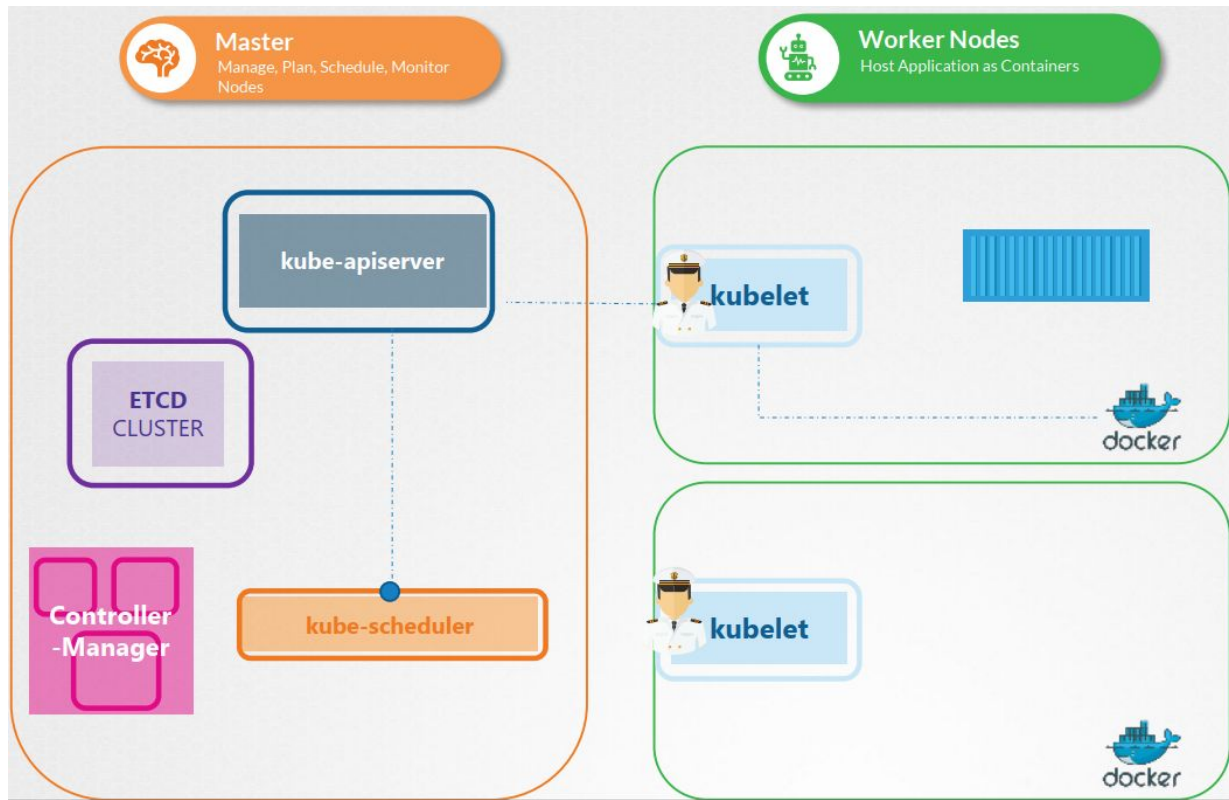
- Kube Scheduler lần lượt thực hiện các bước sau:
1. **Lọc các nodes phù hợp.**
 2. **Xếp hạng nodes**
 3. **Lựa chọn node.**

Kubelet



- **Kubelet: K8s agent** chạy trên node làm nhiệm vụ **thực thi các lệnh** từ api server và giám sát.
- Kubelet lấy cấu hình của Pod từ API server và đảm bảo rằng các containers được mô tả đang hoạt động.

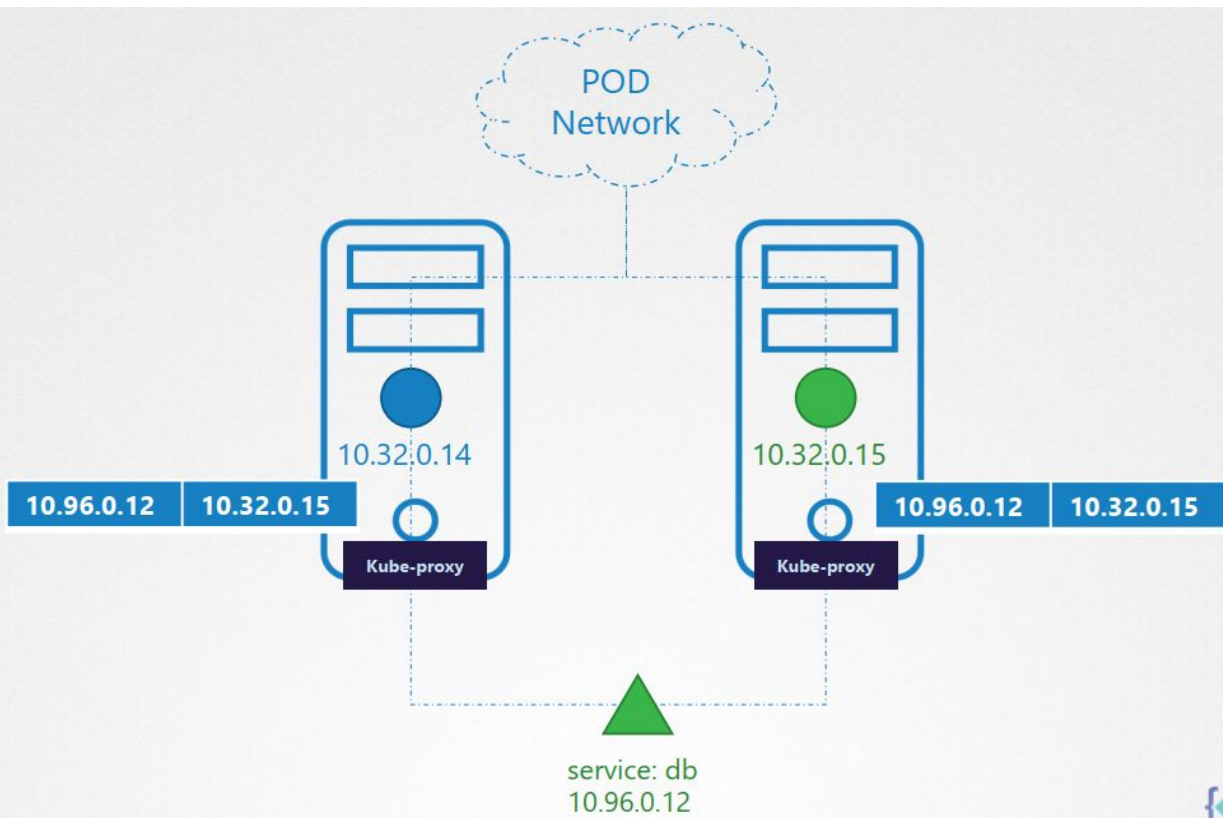
Kubelet



→ Kubelet làm nhiệm vụ:

1. **Đăng kí node.**
2. **Tạo Pods.**
3. **Giám sát node và Pods.**

Kube-proxy



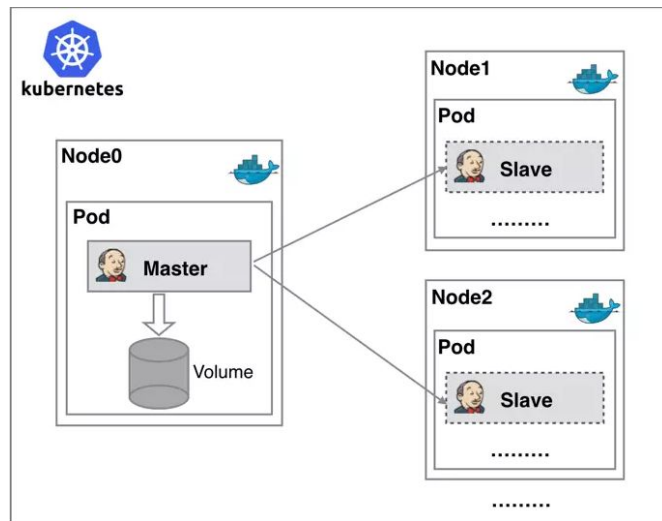
- **Kube-Proxy** : Kube-proxy chạy trên mỗi node để phân chia network (**sub-netting**) và đảm bảo rằng các **dịch vụ sẵn sàng** để kết nối.
- Kube-proxy hoạt động như một **network proxy** và **bộ cân bằng tải** cho dịch vụ trên một worker node duy nhất.

Kubernetes Installation

- Kubernetes có thể được cài đặt bằng 2 cách:
- Triển khai một node (**Minikube K8s Cluster**) - phù hợp cho phát triển và tập luyện.
- Triển khai đảm bảo tính sẵn sàng cao (**1-Master | 2-Worker**) - phù hợp cho môi trường thực tế.



minikube



Labs



Lab 1: Cài đặt Minikube

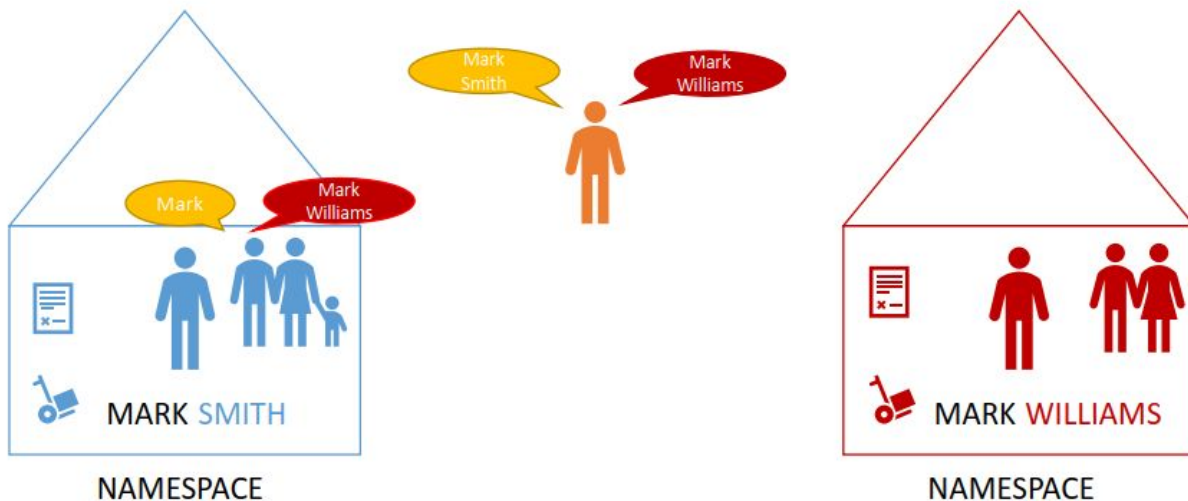
Labs



Lab 2: Tương tác cụm Kubernetes

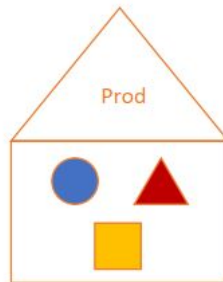
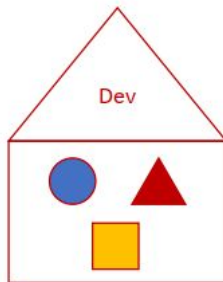
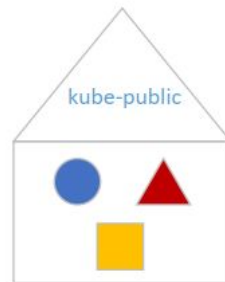
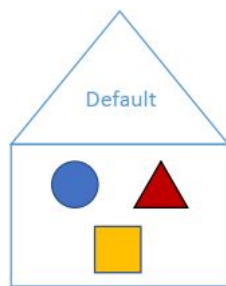
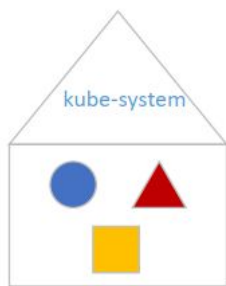
Namespace trong K8s

- Namespace được sử dụng để phân vùng các cụm lớn thành các nhóm nhỏ hơn, dễ quản lý và nhận dạng.



Namespace trong K8s

→ Tách biệt và tổ chức các Object trong K8s



Namespace trong K8s

- Liệt kê các Namespace trong cụm:

```
$kubectl get namespaces
```

- Mọi cụm đều có namespace **default**.
- Chỉ định namespace bằng cách cung cấp tham số sau với kubectl

```
$kubectl get pods --namespace <namespace>
```

- Lưu ý: Nếu không chỉ định namespace thì câu lệnh sẽ lấy ở trong namespace default.
- Tạo namespace như sau:

```
$kubectl create namespace <namespace>
```

Section 2:

Kubernetes cluster management

K8s Management Overview

- ➔ Cụm Kubernetes đảm bảo tính sẵn sàng cao
- ➔ Các công cụ quản lý cụm K8s
- ➔ Cài đặt cụm K8s đảm bảo tính sẵn sàng cao

HA in K8s

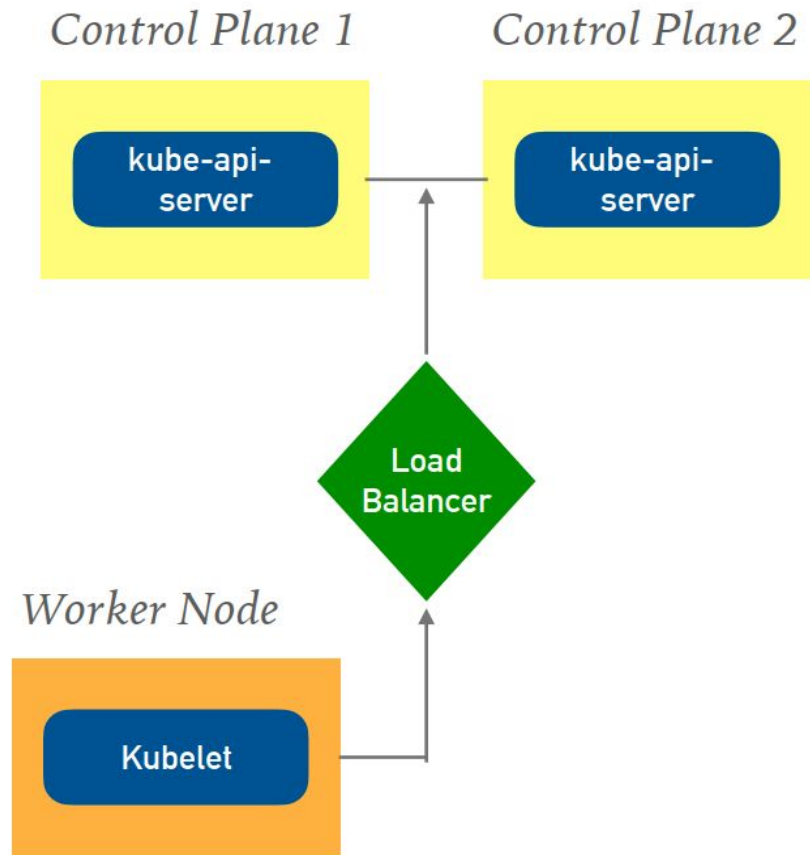
- High Availability (HA) hay Tính sẵn sàng cao là gì ?
- Đảm bảo tính HA các node Master hay Control Plane
- Quản lý Etcd

HA in K8s - Requirements



- K8s hỗ trợ ứng dụng HA.
- Đảm bảo HA hạ tầng.
- Cụm cũng cần đảm bảo HA để hỗ trợ các ứng dụng HA.

HA in K8s - HA Cluster

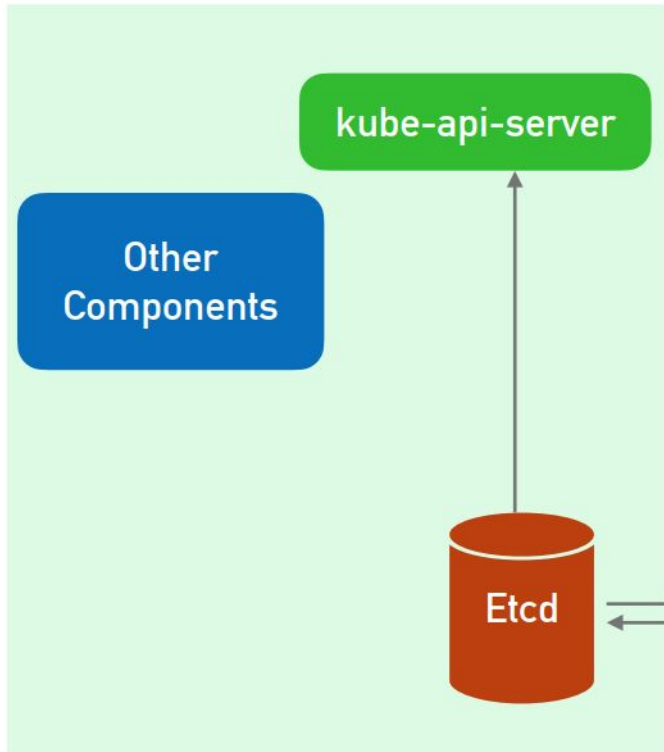


- Để đảm bảo HA cụm, ta cần nhiều node Control Plane.
- Người dùng cần Load Balancer để giao tiếp với nhiều control planes.

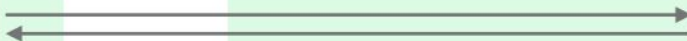
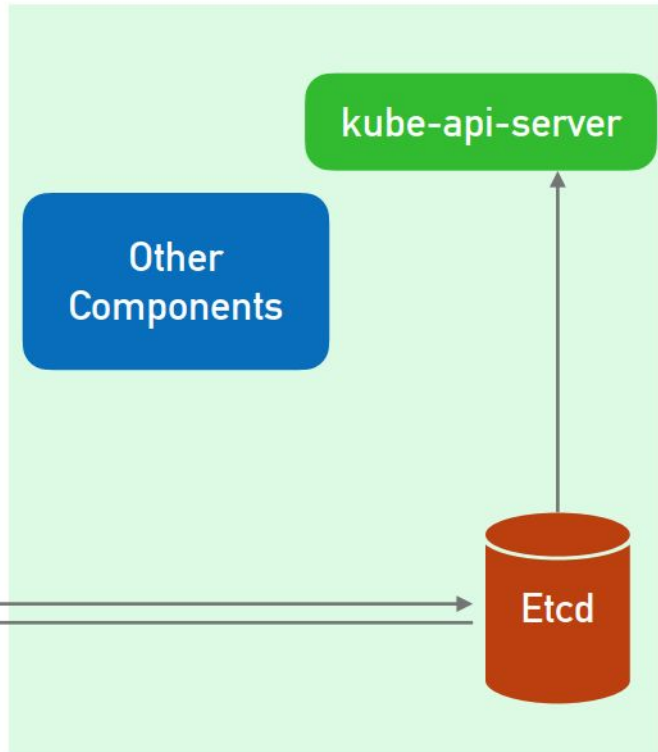
HA in K8s - HA Etcd

Stacked Etcd

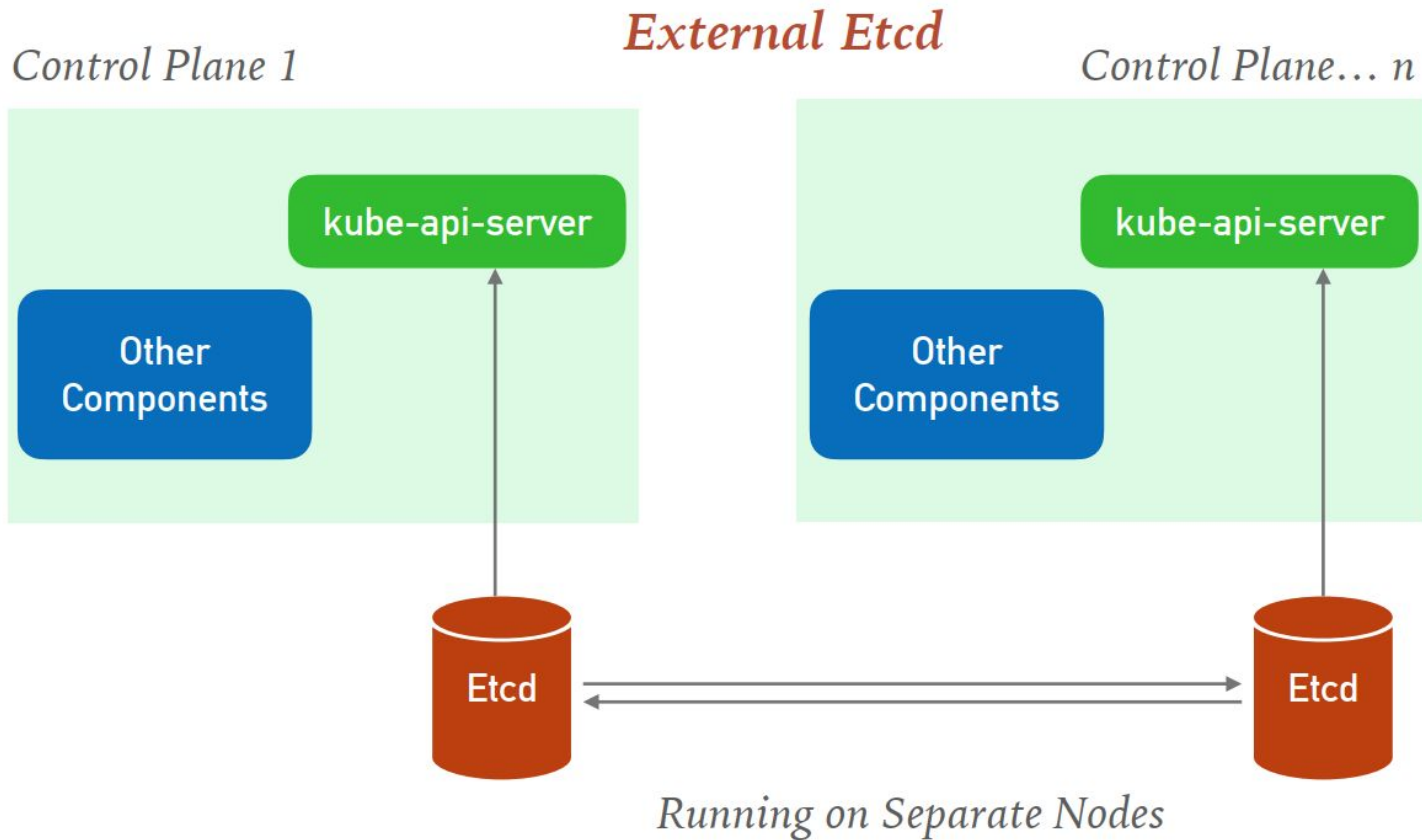
Control Plane 1



Control Plane... n



HA in K8s - HA Etcd



Công cụ quản lý K8s



- Các công cụ K8s cung cấp nhiều chức năng hơn.
- Thực sự cần thiết khi làm quen với các công cụ này.

Công cụ quản lý K8s

→ Các công cụ K8s bao gồm:

→ Kubectl

→ Kubeadm

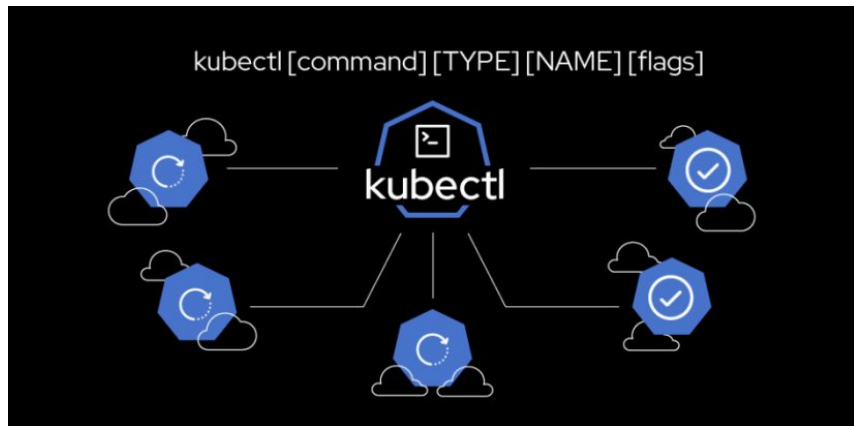
→ Minikube

→ Helm

→ Kompose

→ Kustomize

Công cụ quản lý K8s



- **Kubectl** là CLI chính thức cho K8s.
- Chúng ta sẽ sử dụng công cụ này xuyên suốt học phần K8s này.

Công cụ quản lý K8s



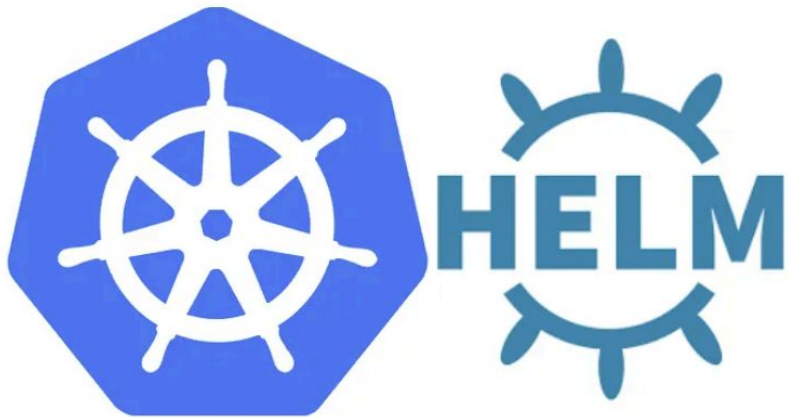
- **Kubeadm** là công cụ giúp đơn giản hóa việc tạo cụm K8s.
- Giúp người dùng cài đặt và thực hiện chức năng của cụm K8s.

Công cụ quản lý K8s



- **Minikube** là công cụ giúp cài đặt nhanh chóng cụm K8s.
- Cài đặt cụm k8s với 1 node.

Công cụ quản lý K8s



- **Helm** là công cụ mạnh mẽ để quản lý Template và Package K8s.
- Có khả năng chuyển các đối tượng K8s sang template có thể tái sử dụng.

Công cụ quản lý K8s



- **Kompose** là công cụ chuyển từ tập Docker Compose sang đối tượng K8s.
- Có khả năng chuyển các containers từ Compose sang K8s.

Công cụ quản lý K8s



- **Kustomize** là công cụ quản lý cấu hình cho cấu hình đối tượng K8s.
- Tương tự như Helm và có khả năng tạo ra templates tái sử dụng được cho K8s.

Labs



Lab 3: Cài đặt cụm HA K8s

Labs



Lab 4: Triển khai ứng dụng đơn giản



Section 3:

Summary

Mục tiêu

Kết thúc bài học, học viên cần đạt được các kỹ năng sau:

- Có hiểu biết về nguồn gốc, tầm quan trọng của Kubernetes
- Nắm được các khái niệm về Kubernetes, các thành phần quan trọng của nó
- Có khả năng cài đặt, cấu hình được cụm Kubernetes phục vụ mục đích học tập hay nhu cầu sử dụng thực tế
- Khả năng tương tác ở mức độ cơ bản với cụm Kubernetes

Tài liệu tham khảo

Thank you