

Lesson 7

Introduction to Kubernetes (k8s)

1. Tìm hiểu Kubernetes (K8s) ?

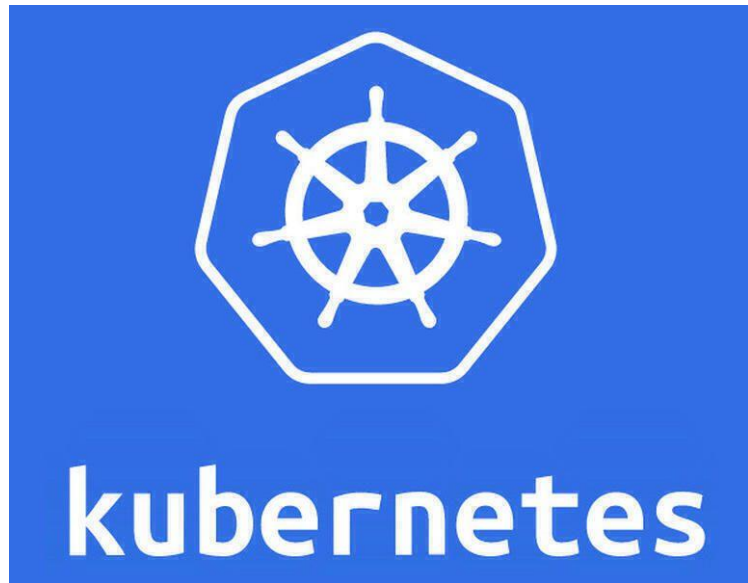
2. Mục đích sử dụng Kubernetes

3. Tìm hiểu về kiến trúc và các thành phần của Kubernetes

4. Tìm hiểu về Minikube và Setup Worker Node

1. Tìm hiểu Kubernetes (K8s)

Kubernetes là một dự án Open Source được phát triển bởi Google. Kubernetes giúp quá trình phát triển, vận hành và Scale Application được tự động hoá, bỏ qua được các quá trình quản lý manual như trước đây gây mất nhiều thời gian trong việc quản lý và khó để có thể maintain



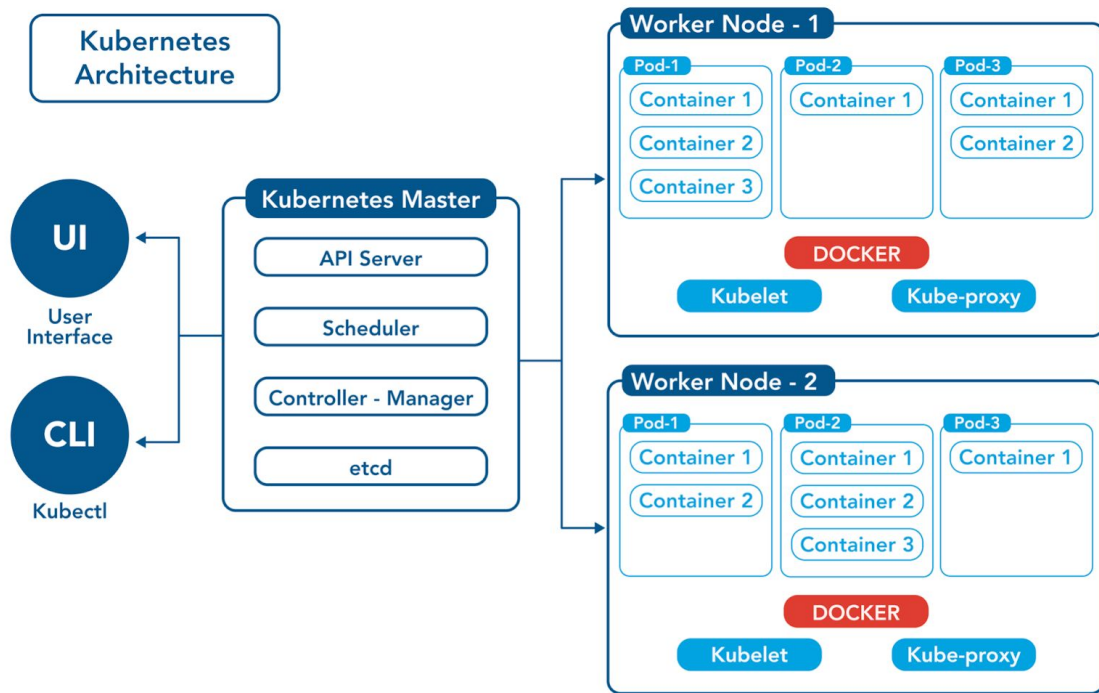
2. Mục đích sử dụng Kubernetes

- Quản lý nhiều môi trường như Dev, Staging, Production trên K8s.
- Kubernetes có thể Scale, ví dụ như mở rộng thêm nhiều môi trường khác để phục vụ những mục đích khác nhau.
- Kubernetes cũng hỗ trợ các công cụ để hỗ trợ trong việc Monitoring. Ví dụ như kiểm tra Resource như RAM, CPU, Network.
- Kubernetes hỗ trợ nhiều kiểu Deployment Application. Phục vụ cho nhiều trường hợp khác nhau.

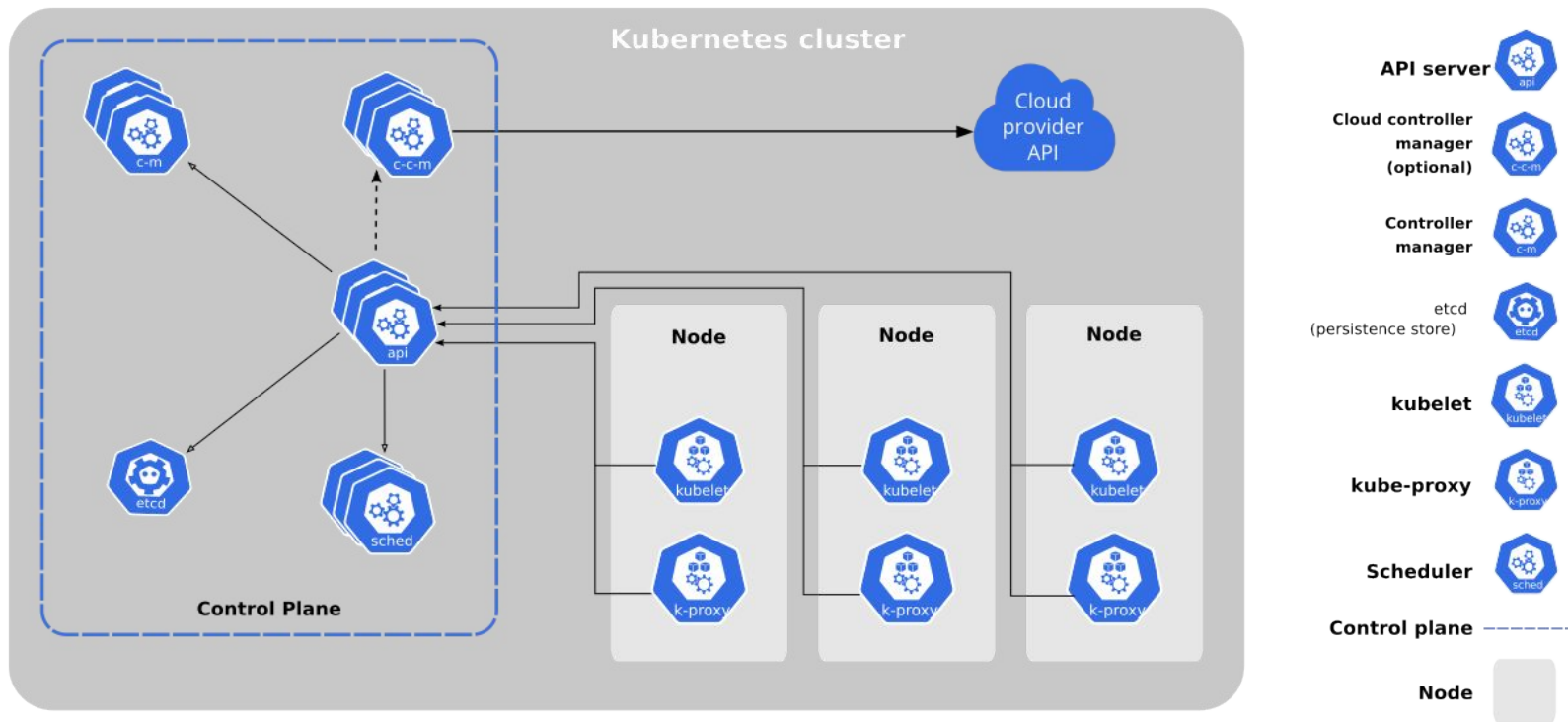
2. Mục đích sử dụng Kubernetes

- Rolling Update (Giúp ứng dụng có thể an toàn hơn trong quá trình Deploy).
- Service Discovery.
- Network (Load Balancer, ClusterIP,...).
- IaC (Infrastructure as Code).
- Quản lý Life Cycle của Pod

3. Tìm hiểu về kiến trúc và các thành phần của Kubernetes



3. Tìm hiểu về kiến trúc và các thành phần của Kubernetes



3. Tìm hiểu về kiến trúc và các thành phần của Kubernetes

Các thành phần trong Kubernetes:

- **Node:** là một máy tính vật lý hoặc máy ảo được sử dụng để chạy các container.
- **Cluster:** Cluster là một cụm các Node để quản lý Container Application như Tạo, Xoá, Run. Trong một Cluster sẽ có 2 thành phần chính là Master, Worker, 2 thành phần này cơ bản cũng chính là Node, nhưng được phân chia để xử lý các công việc khác nhau. Một Cluster có thể bao gồm nhiều Master Node và Worker Node

3. Tìm hiểu về kiến trúc và các thành phần của Kubernetes

- **Master Node:** master node sẽ có trách nhiệm quản lý và điều khiển các Worker Node. Trong Master Node phải có ít nhất một Worker Node và trong Master Node sẽ gồm 4 thành phần, 4 thành phần này trong K8s thường được gọi là được gọi là **Control Plane**
- **Worker Node:** Worker Node là nơi để Tạo và Run Container Application. Mỗi Worker Node sẽ có một hoặc nhiều Pod. Và mỗi Worker Node sẽ có 3 thành phần chính Container Runtime, Kubelet, Kube-proxy

3. Tìm hiểu về kiến trúc và các thành phần của Kubernetes

Master Node:

- **API Server:** API Server được xem như một Entry Point của hệ thống Kubernetes, mọi yêu cầu từ User sẽ đi qua thành phần này. Yêu cầu ở đây có thể là Tạo Pod, Xoá Pod, Tạo Service hay Xoá Service
- **Scheduler:** Scheduler là thành phần quan trọng trong K8s dùng để ra quyết định Pod sẽ được tạo ở Node nào, Scheduler sẽ dựa vào các điều kiện ví dụ như Resource của các Node, để xem Node nào đang sử dụng ít Resource thì sẽ tạo Pod ở Node đó, tránh việc tạo quá nhiều Pod vào một Node, gây mất cân bằng trong hệ thống.

3. Tìm hiểu về kiến trúc và các thành phần của Kubernetes

- **Controller Manager:** Controller Manager sẽ có nhiệm vụ là quản lý State của Cluster, phát hiện Pod xảy ra vấn đề và ra quyết định kịp thời để xử lý Pod đó. Ví dụ nếu một Pod xảy ra lỗi và bị Stop, thì Controller Manager sẽ kiểm tra số lượng Pod mong muốn mà Developer đã define trước đó, để ra quyết định tạo Pod mới thay thế cho Pod đã bị Stop trước đó.
- **Etcd:** Etcd là một nơi lưu trữ State (trạng thái) của Cluster, có thể xem như là bộ não của Cluster. Kiểu dữ liệu của Etcd sẽ được lưu theo Key/Value.

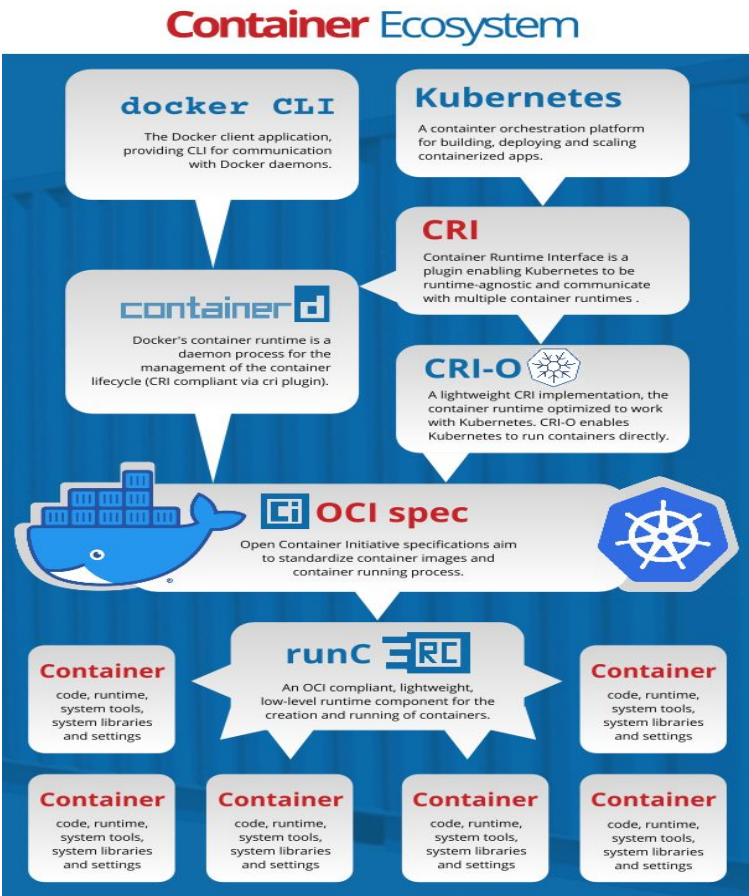
3. Tìm hiểu về kiến trúc và các thành phần của Kubernetes

Worker Node:

- **Container Runtime:** Container Runtime là một phần mềm chịu trách nhiệm cho việc Run Application. Ở K8s có hỗ trợ nhiều loại Container Runtime như: Docker, containerd và CRI-O.
 - **CRI-O:** Là một phiên bản nhẹ hơn được tạo bởi K8s và được sử dụng để hỗ trợ cho K8s thay thế cho việc sử dụng Docker là môi trường Runtime

3. Tìm hiểu về kiến trúc và các thành phần của Kubernetes

- **Kubelet:** Kubelet là một Process hoạt động ở trong Node, nó sẽ có nhiệm vụ tương tác với Node và Pod. Tương tác ở đây có nghĩa là nó có thể tạo được Pod.
- **Kube-proxy:** Kube-proxy dùng để quản lý và thiết lập Rule Network. Nó sẽ quyết định Network ra vào của một Node hoặc những. Hoặc những Pod giao tiếp với nhau cũng sẽ thông qua thành phần này.



4. Tìm hiểu về Minikube và Setup Worker Node

- **Minikube:**

- Cài đặt Minikube
- Start Minikube
- Sử dụng Dashboard Minikube

- **Setup Worker Node:**

- Thiết lập Node (Server)
- Thiết lập Kubeadmin
- Kết nối Node vào Cluster