VIỆN CÔNG NGHỆ THÔNG TIN T3H

Đào tạo chuyên sâu - Trải nghiệm thực tế

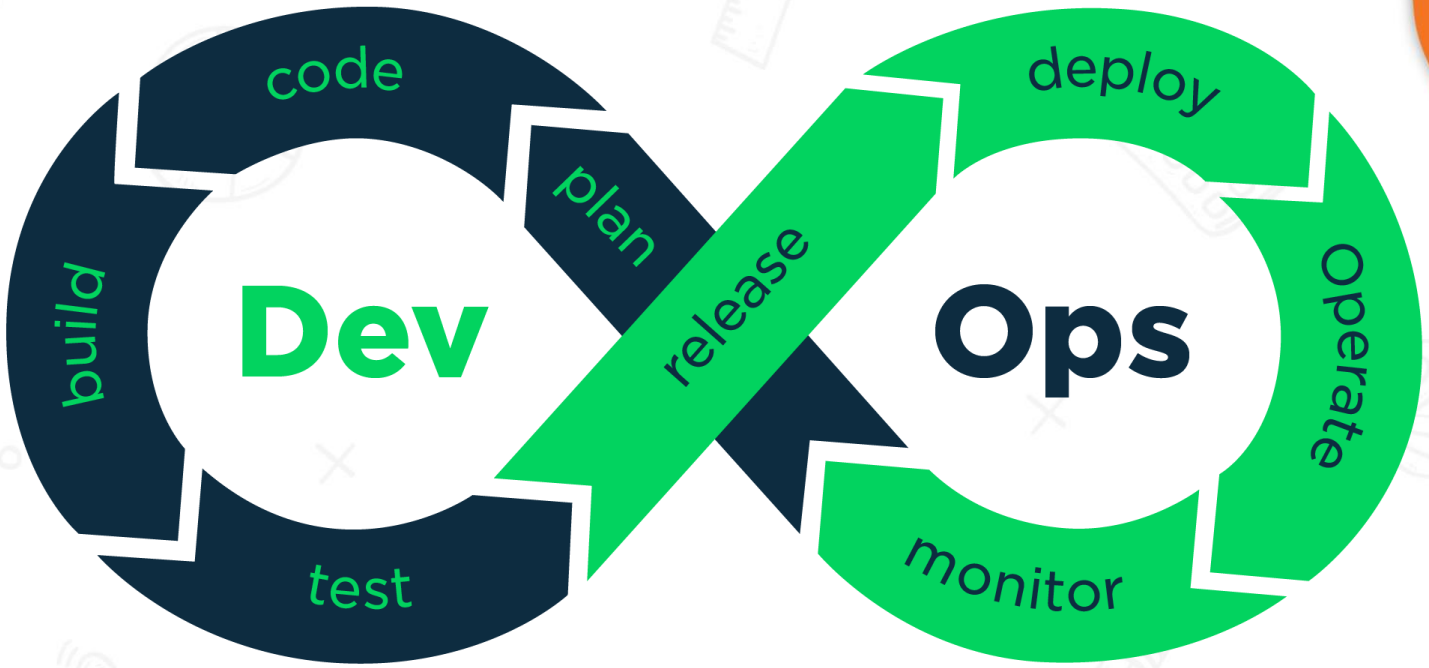
Môn học

DevOps

Giảng viên

Tan Do

0868880797



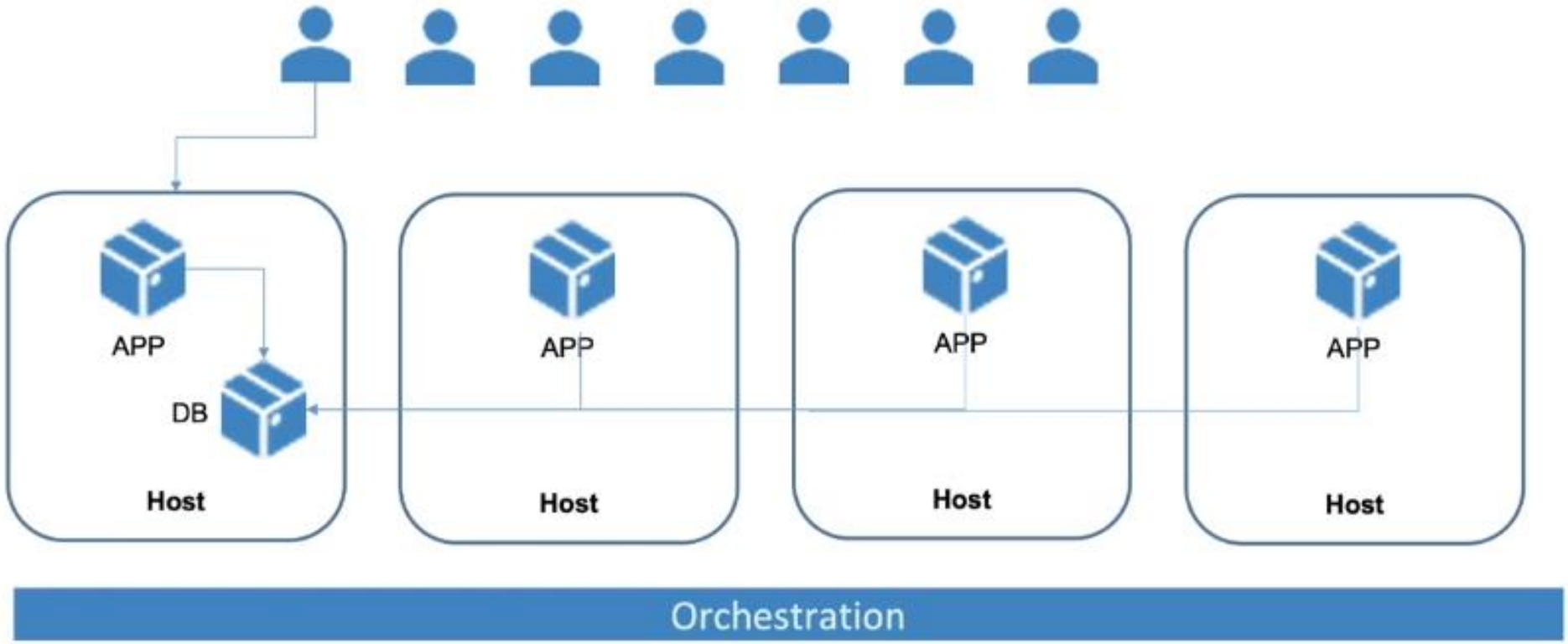
Nội dung buổi 12

- Điều phối Container là gì? (container orchestration)
- Giới thiệu về Kubenetes
- Kiến trúc Kubernetes
- Các thành phần & khái niệm cơ bản k8s
- Cài đặt và thực hành cơ bản
- Review Project cuối khóa

Điều phối Container là gì?

- Container Orchestration
- Tự động hóa việc triển khai, quản lý, nhân rộng và kết nối mạng của container
- Các doanh nghiệp cần triển khai và quản lý hàng trăm hoặc hàng nghìn container
- Hệ thống điều phối container có thể được sử dụng trong bất kỳ môi trường nào sử dụng container
- Giúp triển khai cùng một ứng dụng trên các môi trường khác nhau mà không cần thiết kế lại
- Microservices trong các container giúp việc sắp xếp các dịch vụ dễ dàng hơn, bao gồm lưu trữ, kết nối mạng và bảo mật

Điều phối Container là gì?



Điều phối Container sử dụng làm gì?

Sử dụng hệ thống điều phối container để tự động hóa và quản lý các tác vụ như:

- Cung cấp và triển khai
- Cấu hình và lập lịch
- Phân bổ nguồn lực
- Tạo tính sẵn sàng cho container
- Thu nhỏ hoặc loại bỏ các container dựa trên việc cân bằng khối lượng công việc trên cơ sở hạ tầng của bạn
- Cân bằng tải và định tuyến giao thông
- Theo dõi sức khỏe container
- Định cấu hình các ứng dụng dựa trên vùng chứa mà chúng sẽ chạy
- Giữ tương tác giữa các container an toàn

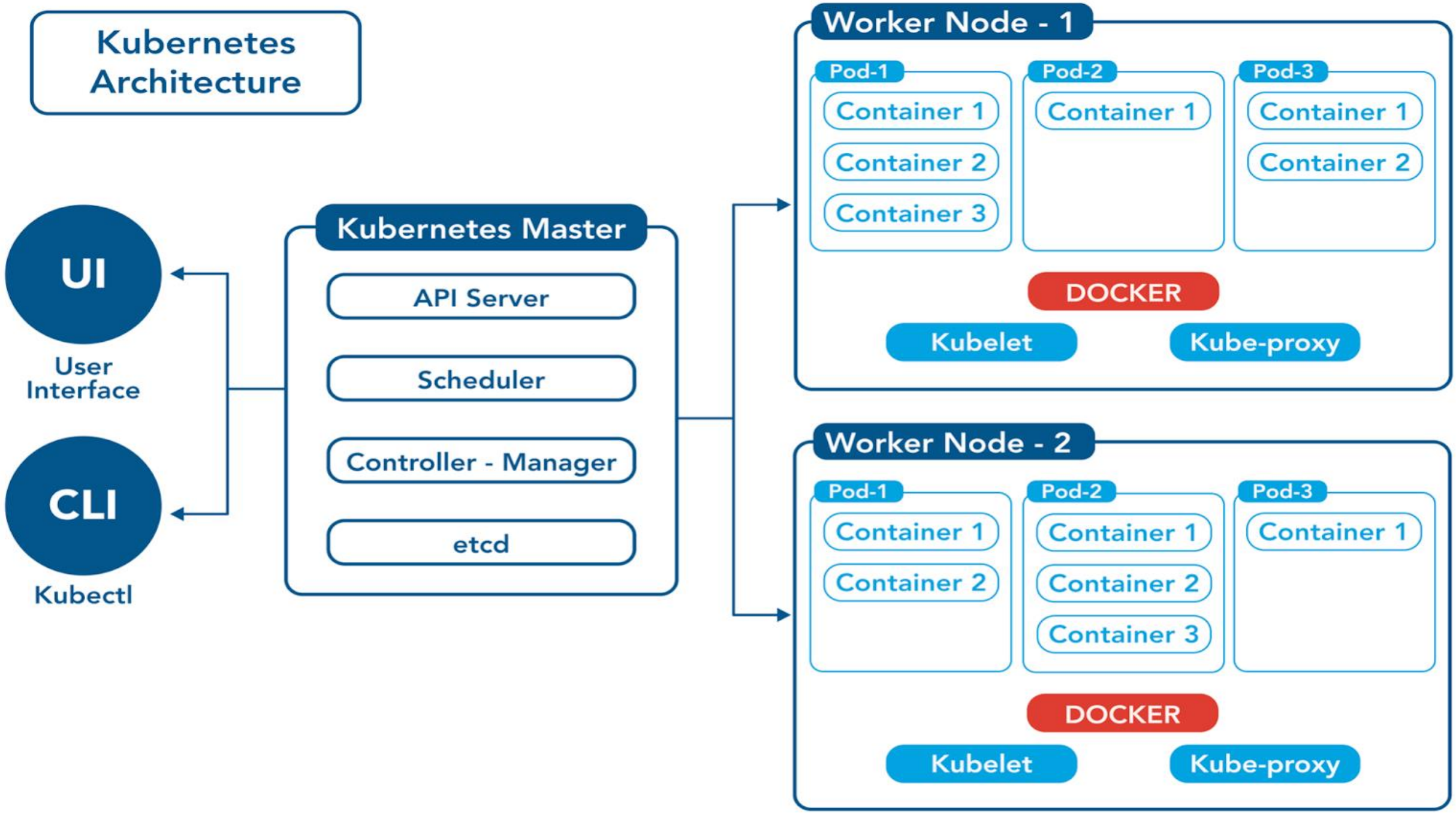
Container Orchestrator

Amazon Elastic
Container Service

kubernetes

Kubernetes là gì?

- Là một mã nguồn mở được dùng để tự động triển khai hệ thống, scaling, quản lý các container
- Là một hệ thống mạnh mẽ, được phát triển bởi Google
- Google sử dụng Kubernetes để quản lý hàng tỉ docker container mà họ đang quản lý
- Viết tắt của K**ubernete**s là k8s (K-**8 chữ cái**-s)


Kubernetes Architecture
UI
User Interface
CLI
Kubectl
Kubernetes Master
API Server
Scheduler
Controller - Manager
etcd
Worker Node - 1
Pod-1
Container 1
Container 2
Container 3
Pod-2
Container 1
Pod-3
Container 1
Container 2
DOCKER
Kubelet
Kube-proxy
Worker Node - 2
Pod-1
Container 1
Container 2
Pod-2
Container 1
Container 2
Container 3
Pod-3
Container 1
DOCKER
Kubelet
Kube-proxy

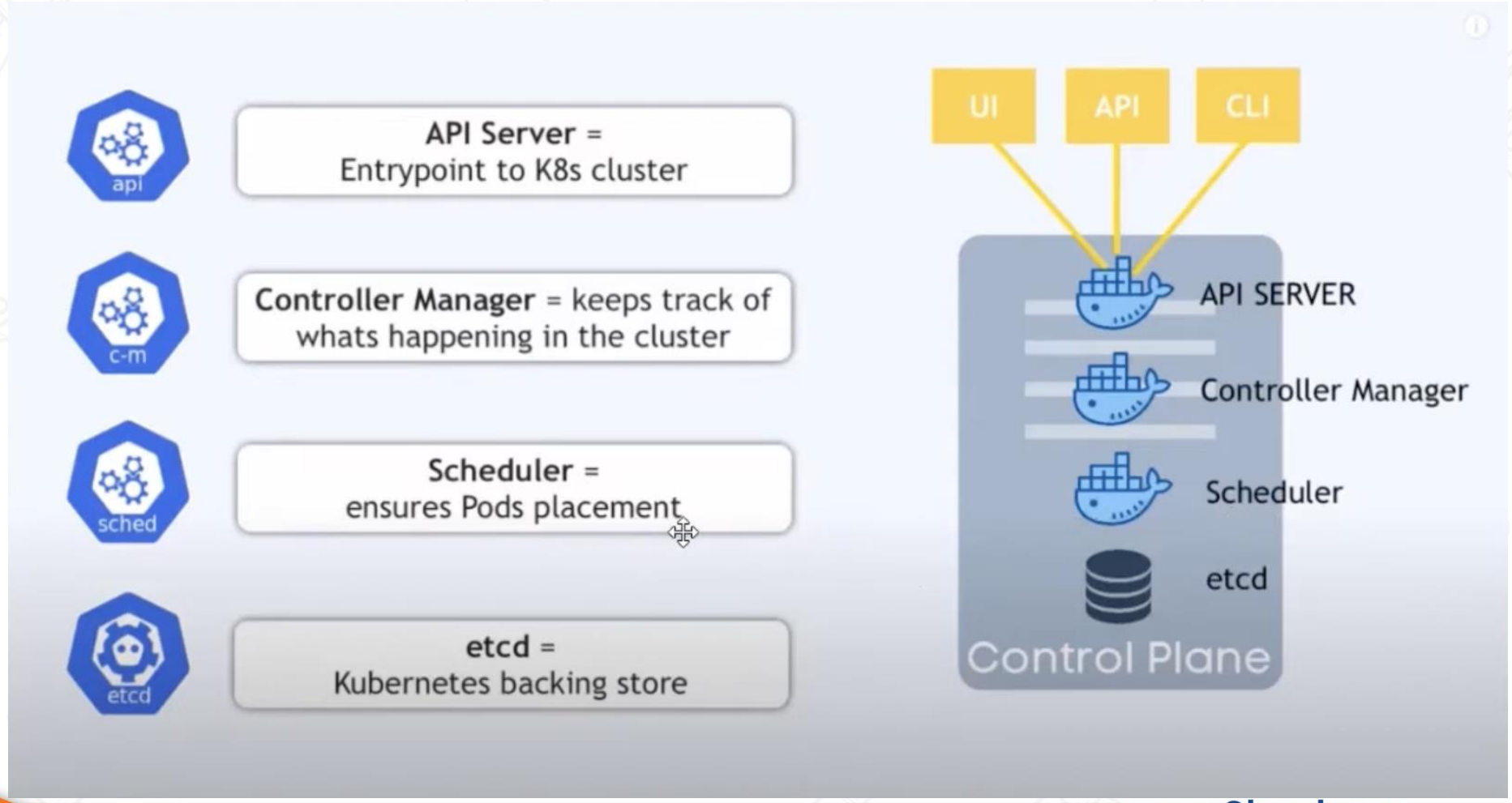
Kiến trúc K8s

- K8s cluster bao gồm nhiều node, trên mỗi node sẽ cần chạy một "kubelet", đây là chương trình để chạy k8s
- Cần một máy để làm "chủ" cluster, trên đó sẽ cài API server, scheduler ...
- Các máy còn lại sẽ chạy kubelet để sinh ra các container

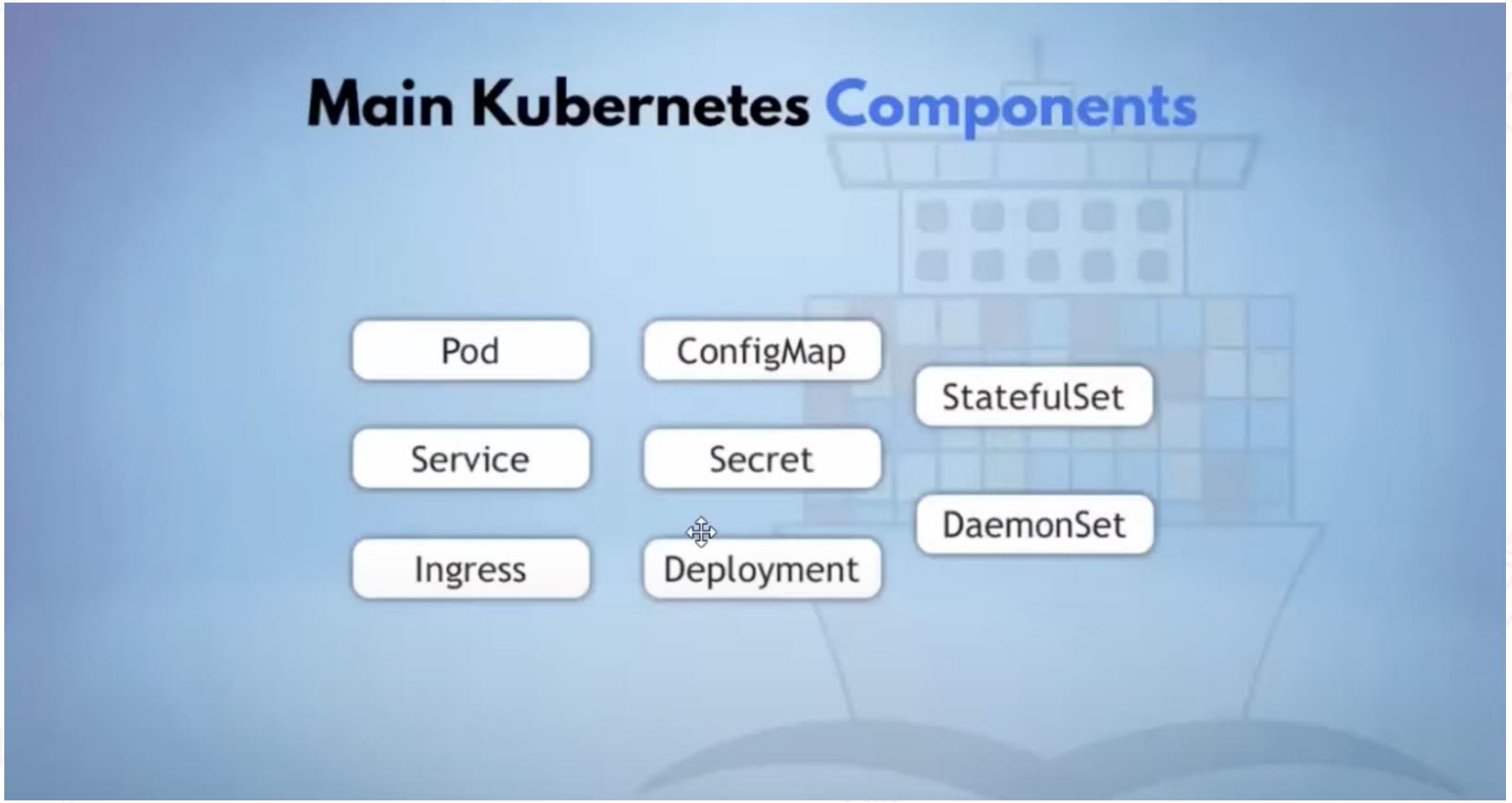
Các thành phần K8s - MASTER NODE

- API Server
  - Là server cung cấp Kubernetes API
  - Có nhiệm vụ đặt Pod vào Node
  - Đồng bộ hoá thông tin của Pod bằng REST API tiếp nhận cài đặt của pod/service/replicationController
- Controller Manager Service
  - Giống như “kube-controller manager”,
  - Quản lý tất cả các bộ điều khiển xử lý các tác vụ thông thường trong cluster
  - Bao gồm Node Controller, Replication Controller, Endpoints Controller, and Service Account and Token Controllers. Chi tiết của các hoạt động này được ghi vào etcd, nơi controller manager theo dõi sự thay đổi thông qua API Server
- Scheduler Service
  - Có trách nhiệm giám sát việc sử dụng tài nguyên trên mỗi máy chủ để đảm bảo rằng hệ thống không bị quá tải
  - Scheduler Service phải biết tổng số tài nguyên có sẵn trên mỗi máy chủ, cũng như các tài nguyên được phân bổ cho các khối lượng công việc hiện có được gán trên mỗi máy chủ
- etcd
  - Có thể liên kết cài đặt với từng node thông qua etcd

Các thành phần K8s - MASTER NODE



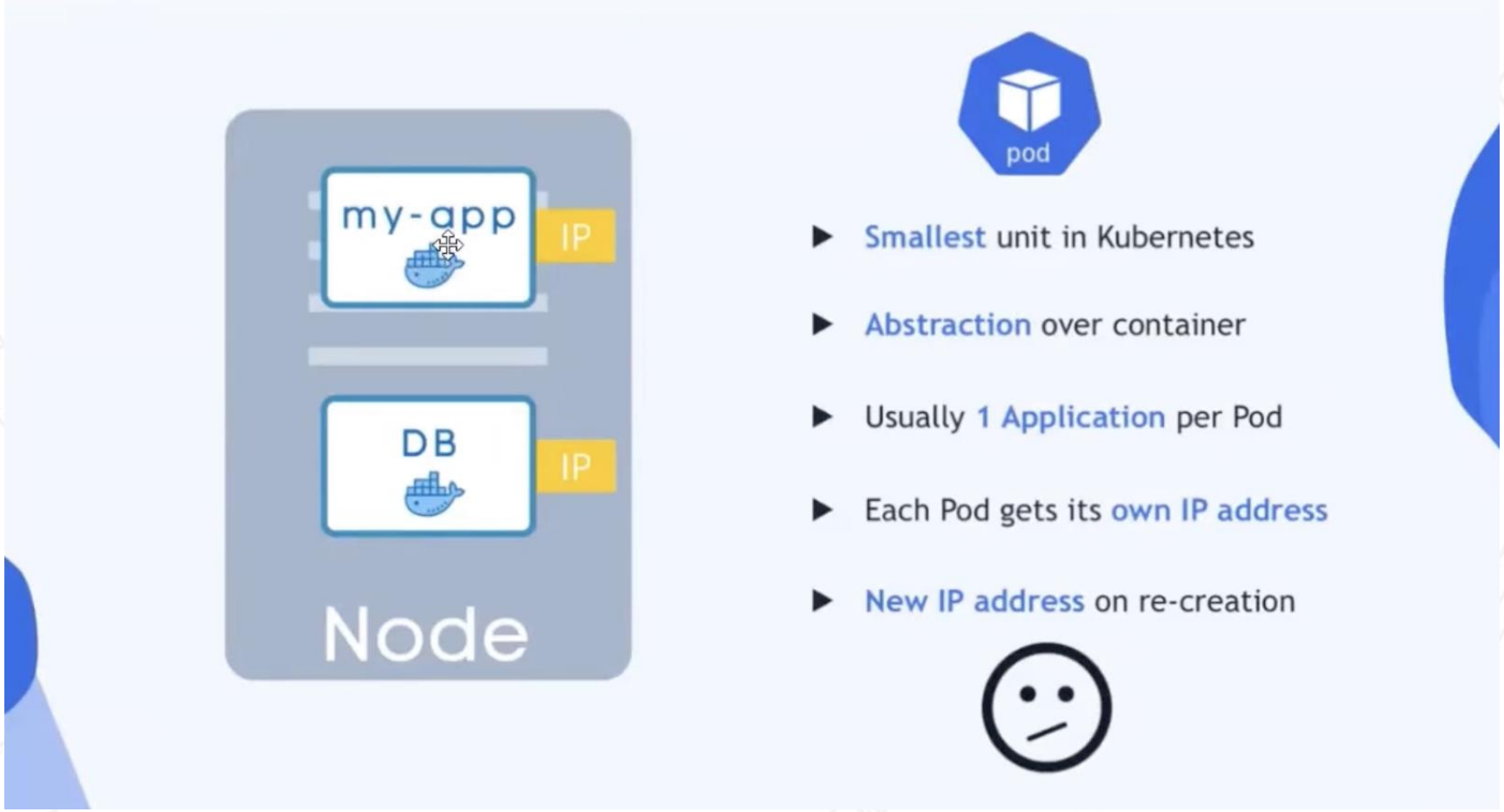
Kiến trúc K8s



Các thành phần K8s – WORKER NODE

- Pod
  - Pod là 1 nhóm (1 trở lên) các container thực hiện một mục đích nào đó, như là chạy software nào đó
  - Nhóm này chia sẻ không gian lưu trữ, địa chỉ IP với nhau
  - Pod được tạo ra hoặc xóa tùy thuộc vào yêu cầu của dự án
- Nếu k8s chỉ có mỗi khái niệm pod, thì dùng k8s giống như dùng docker bình thường - tức muốn thêm tính năng gì thì ta phải tự kiến trúc/ thiết kế/ thực hiện

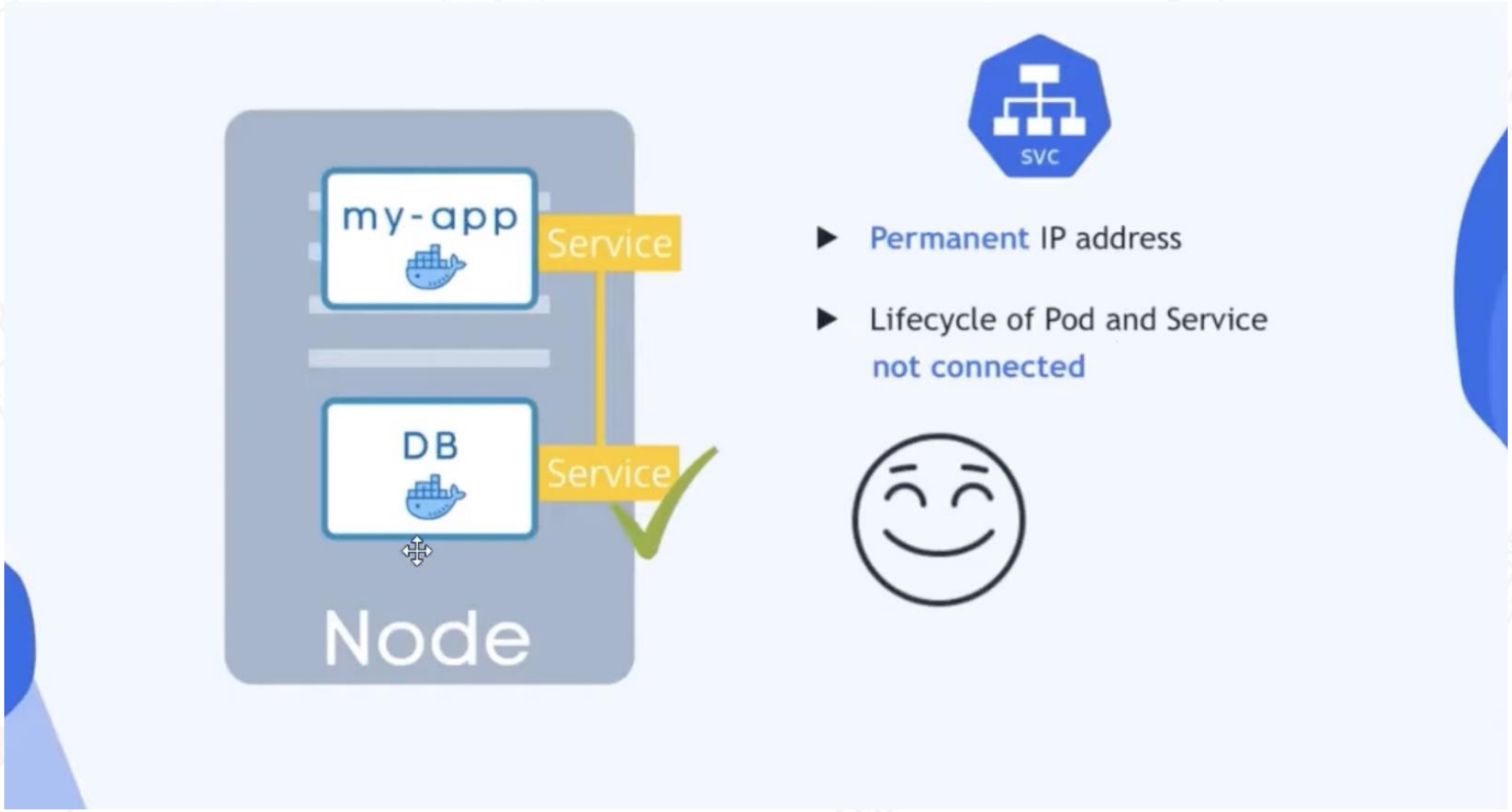
Các thành phần K8s – WORKER NODE



Các thành phần K8s – WORKER NODE

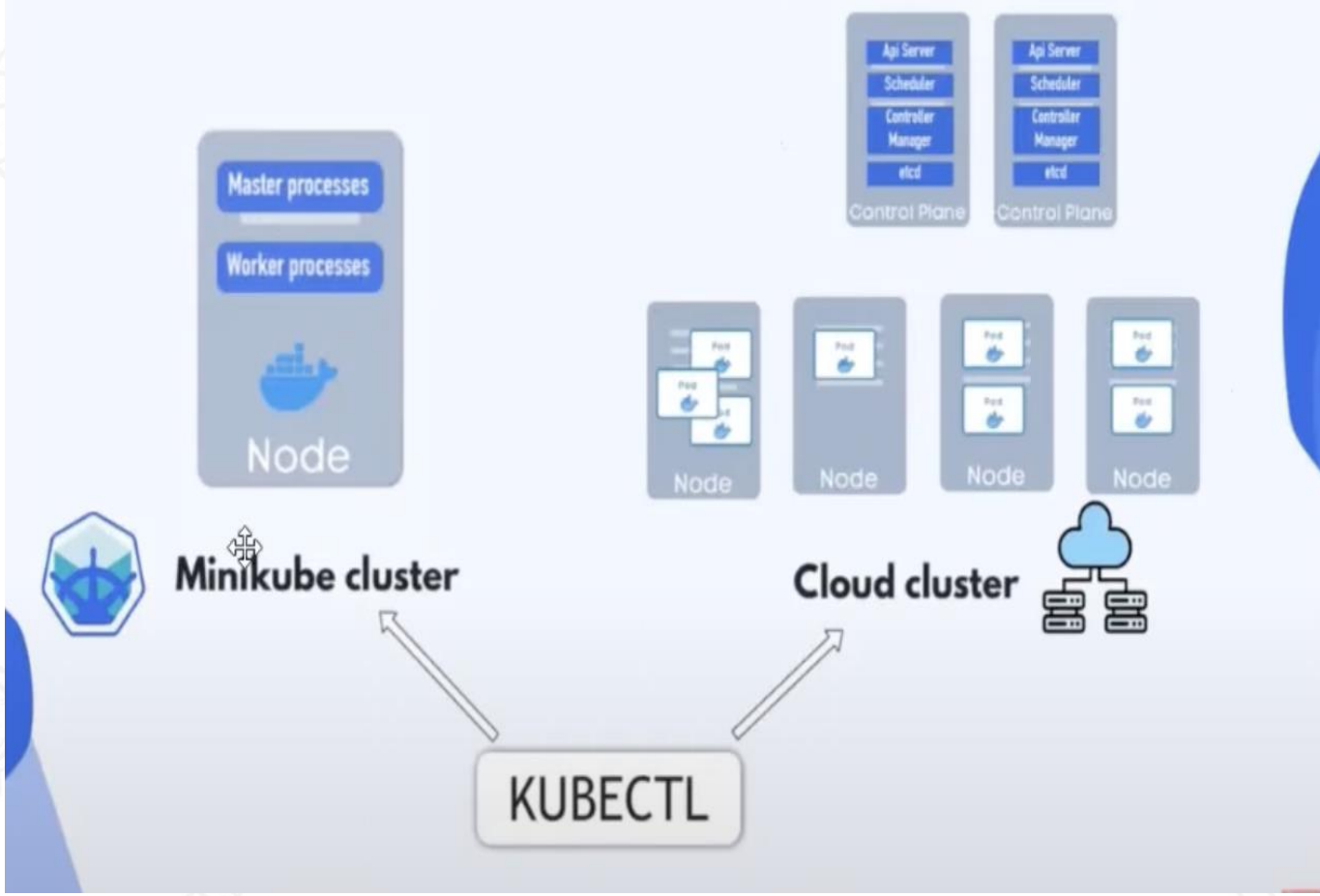
- Service (svc)
  - Vì các Pod có tuổi thọ ngắn, do vậy nó không đảm bảo về địa chỉ ip luôn cố định, khiến cho việc giao tiếp giữa các microservices trở nên khó khăn
  - Do đó, Service là một lớp nằm trên một số nhóm Pod
  - Được được gắn địa chỉ IP tĩnh và có thể trỏ domain vào dịch vụ này, tại đây có thể thực hiện cân bằng tải
  - Mỗi service sẽ được gán 1 domain do người dùng lựa chọn, khi ứng dụng cần kết nối đến service, chỉ cần dùng domain là xong
  - Domain được quản lý bởi hệ thống name server SkyDNS nội bộ của k8s - một thành phần sẽ được cài khi cài k8s
- Nếu chỉ có 1 máy chạy 1 dịch vụ, thì service không có ý nghĩa gì. Vậy nên khi dùng k8s, hãy nhớ rằng nó được thiết kế và đưa vào các khái niệm để phục vụ cho hàng trăm, ngàn service/container, chứ không phải 1 cái

Các thành phần K8s – WORKER NODE



Các thành phần K8s – WORKER NODE

- Kubectl
  - Tương tác với Master Node sử dụng kubectl, kubectl là command line interface(CLI) cho Kubernetes
  - Kubectl có config file được gọi là kubeconfig. File này có nhưng thông tin như: server information, authentication infomation để access API Server



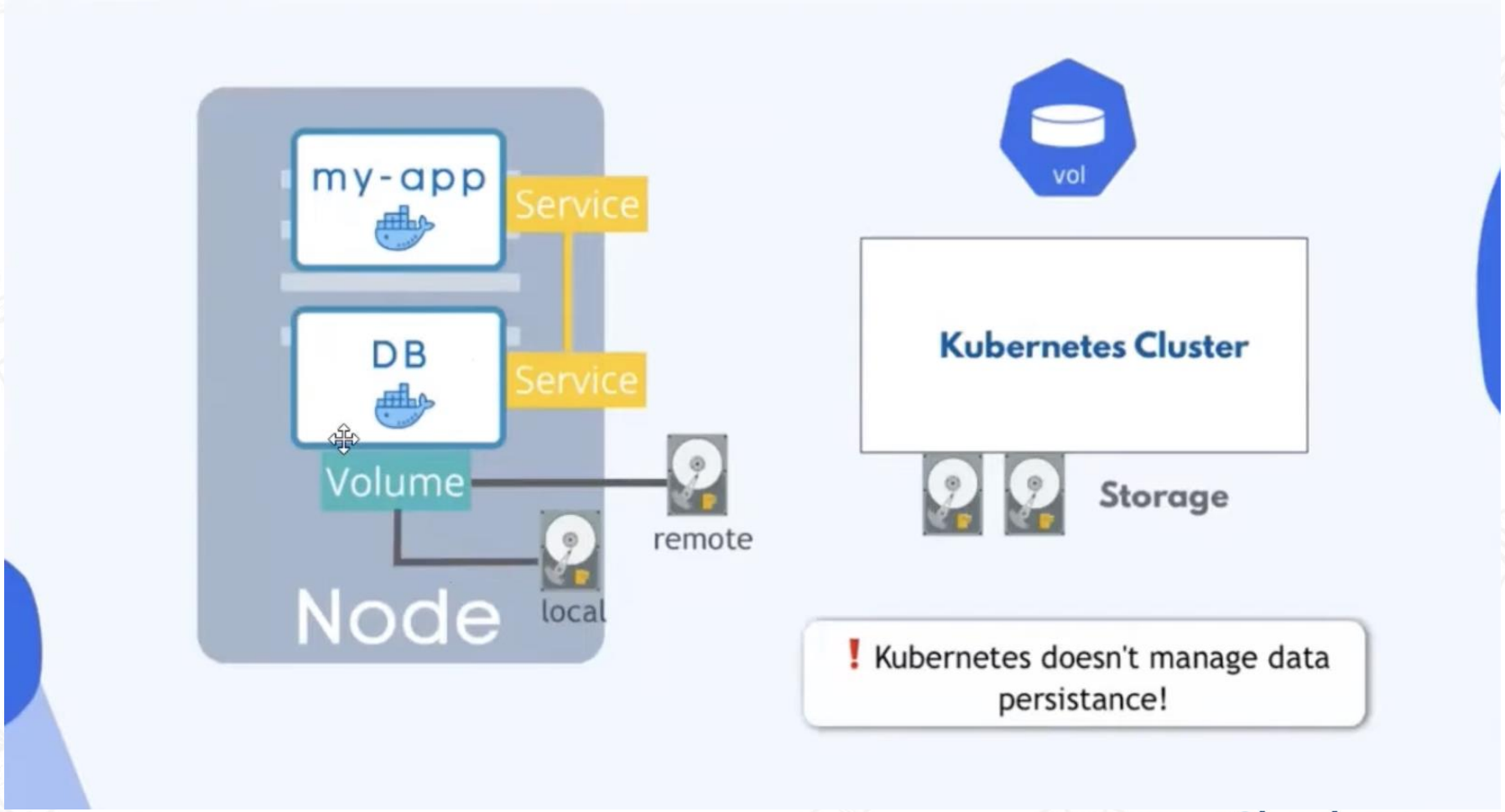
Các thành phần K8s – WORKER NODE

- Worker node
  - Worker node là node nơi mà ứng dụng (application) được triển khai
  - Worker node giao tiếp với master node
  - Giao tiếp với worker node được xử lý bởi Kubelet. Kubelet là một agent giao tiếp với API Server để xem liệu Pods đã được cho vào Node hay chưa
    - kubelet chạy các Pod containers thông qua container engine
    - Nó mount và runs Pod volume, secrets. Nó xem trạng thái của Pod của Node và phản hồi về Master Node
- Kube Proxy
  - kube-proxy là Network Proxy và load balancer cho dịch vụ, trên một single worker node. Nó đảm nhiệm network routing cho TCP và UDP packets, và thực hiện connection forwarding
  - 1 hoặc nhiều containers sẽ được gói trong Pod. Pod là unit nhỏ nhat có thể được scheduled như 1 deployment trong Kubernetes. Nhóm containers trong pod dùng chung storage, Linux name space, IP addresses, ...
  - Khi Pod được deployed và running, kubelet process giao tiếp với Pods để check state và health. kube-proxy sẽ route any packets tới Pods từ resources khác muốn communicate với chúng. Worker node có thể được expose tới internet thông qua load balancer. Traffic đi vào các node cũng sẽ được kube-proxy đảm nhiệm

Các thành phần & khái niệm K8s

- Persistent Volumes (PV)
  - Khi làm container cũng lưu ý rằng không lưu dữ liệu trên container mà phải lưu nó vào một chỗ nào đó, vì khi container restart / die thì dữ liệu cũng sẽ mất
  - K8s sử dụng là các hệ thống lưu trữ "network“ tức lưu vào một hệ thống storage khác như NFS, GlusterFS, Ceph ...
  - PV là khái niệm để đưa ra 1 dung lượng lưu trữ thực tế 1GB, 10GB ...
  - PVC là khái niệm ảo đưa ra 1 dung lượng cần thiết mà ứng dụng yêu cầu
  - Khi 1 PV thoả mãn yêu cầu của 1 PVC thì chúng "match" nhau, rồi "bound" (buộc / kết nối) lại với nhau

Các thành phần & khái niệm K8s

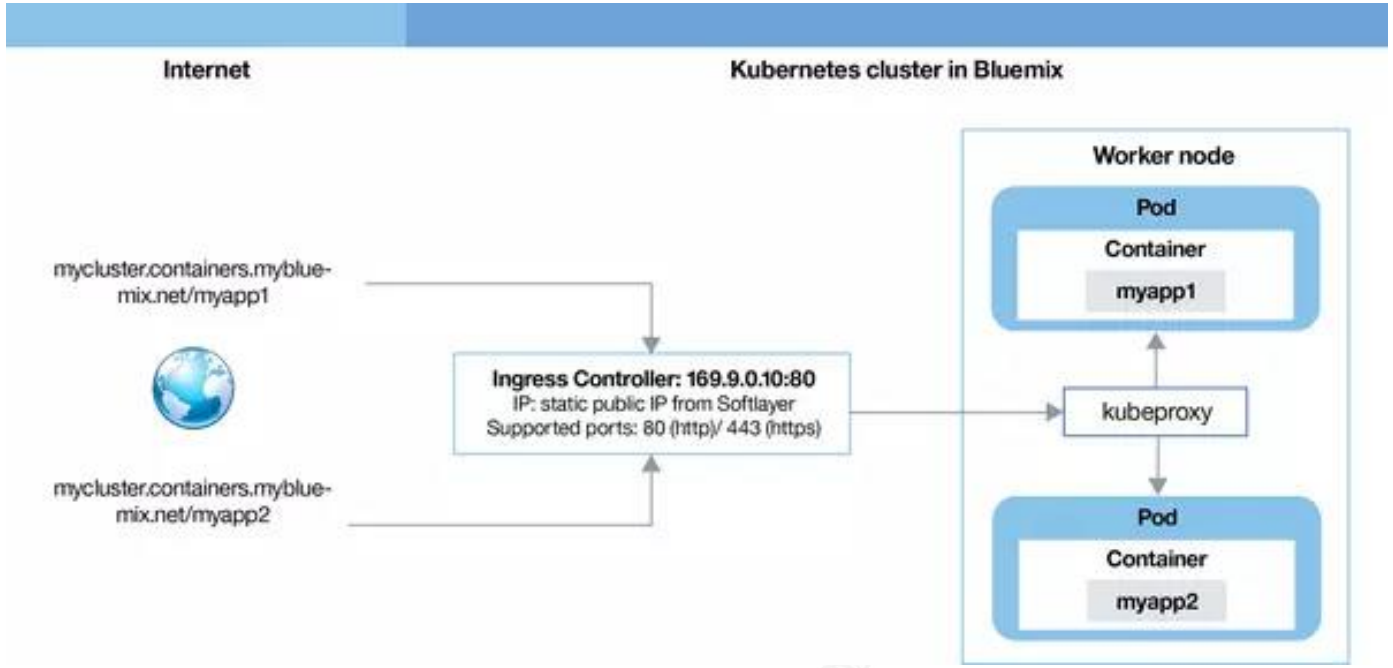


Các thành phần & khái niệm K8s

- Namespaces
  - Đây là một công cụ dùng để nhóm hoặc tách các nhóm đối tượng. Namespaces được sử dụng để kiểm soát truy cập, kiểm soát truy cập network, quản lý resource và quoting
  - Nếu đặt service này là "web" lúc chạy production, còn lúc dev thì chạy nó ở đâu? có phải đổi tên service? Namespace giải quyết vấn đề này
  - Mặc định các dịch vụ sẽ sử đụng namespace "default", nhưng có thể tạo namespace tuỳ ý
  - K8s sử dụng 1 namespace riêng: kube-system, vì vậy đừng quên namespace khi gọi câu lệnh

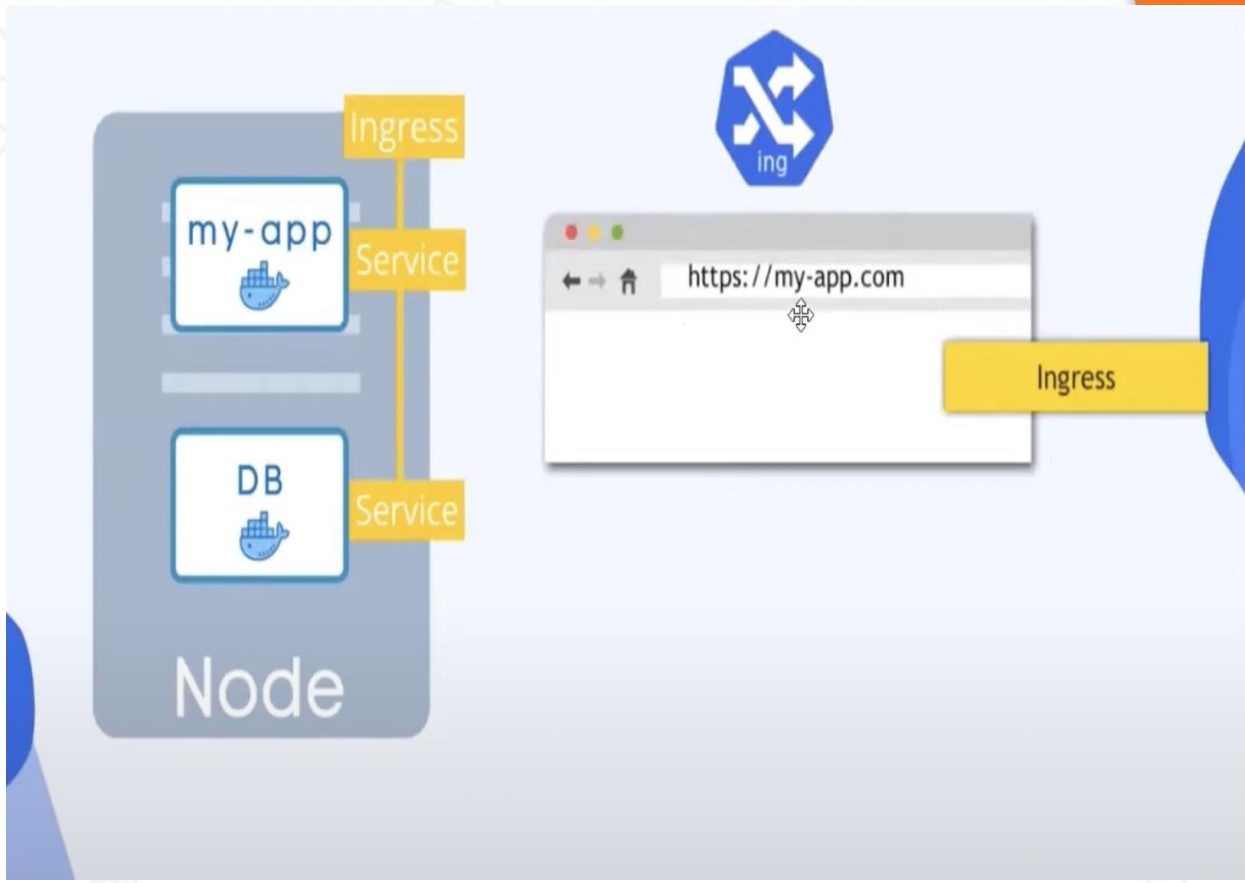
Các thành phần & khái niệm K8s

- Ingress rules
  - Dùng để quản lý network ra và vào các service và pod



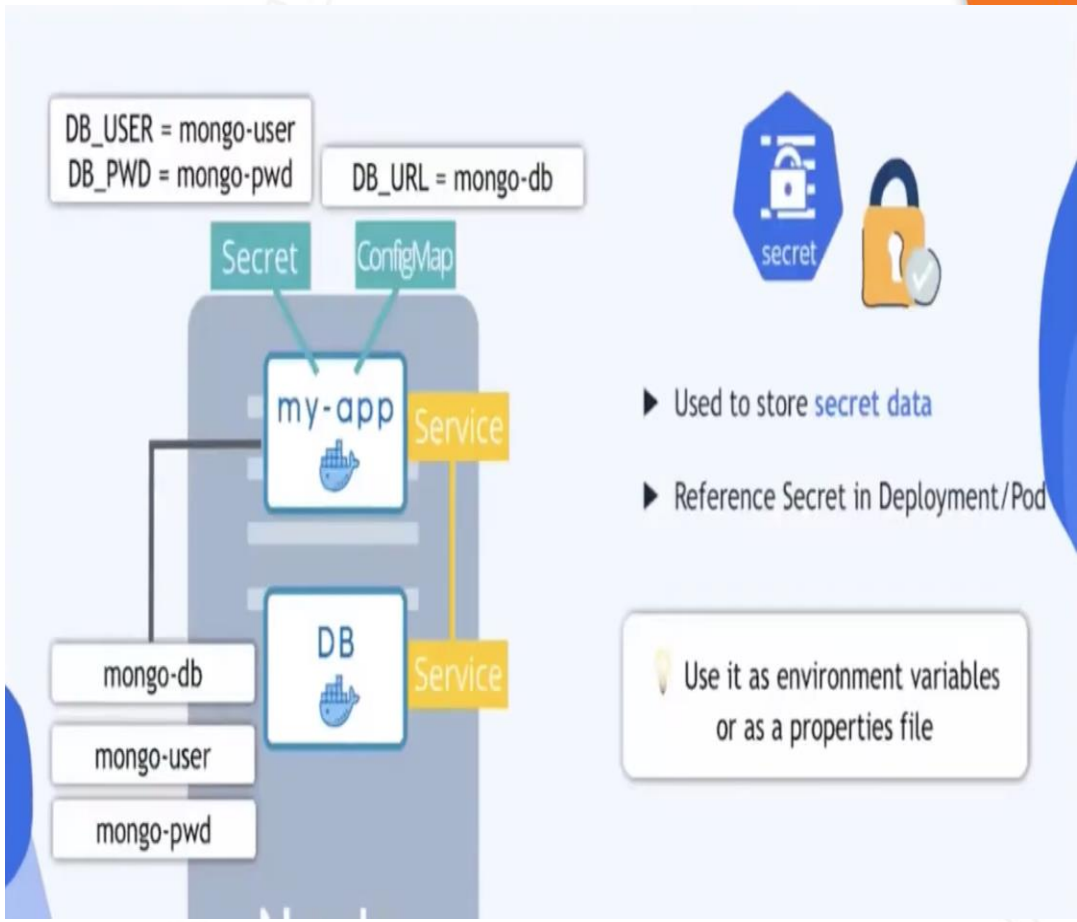
Các thành phần & khái niệm K8s

- Network policies
  - Định nghĩa các quy tắc truy cập mạng giữa các Pod bên trong Cluster
- Network
  - Có nhiều loại phần mềm để triển khai container network, như Flannel, Weaver ... nếu dùng Google Cloud, vấn đề này không cần quan tâm



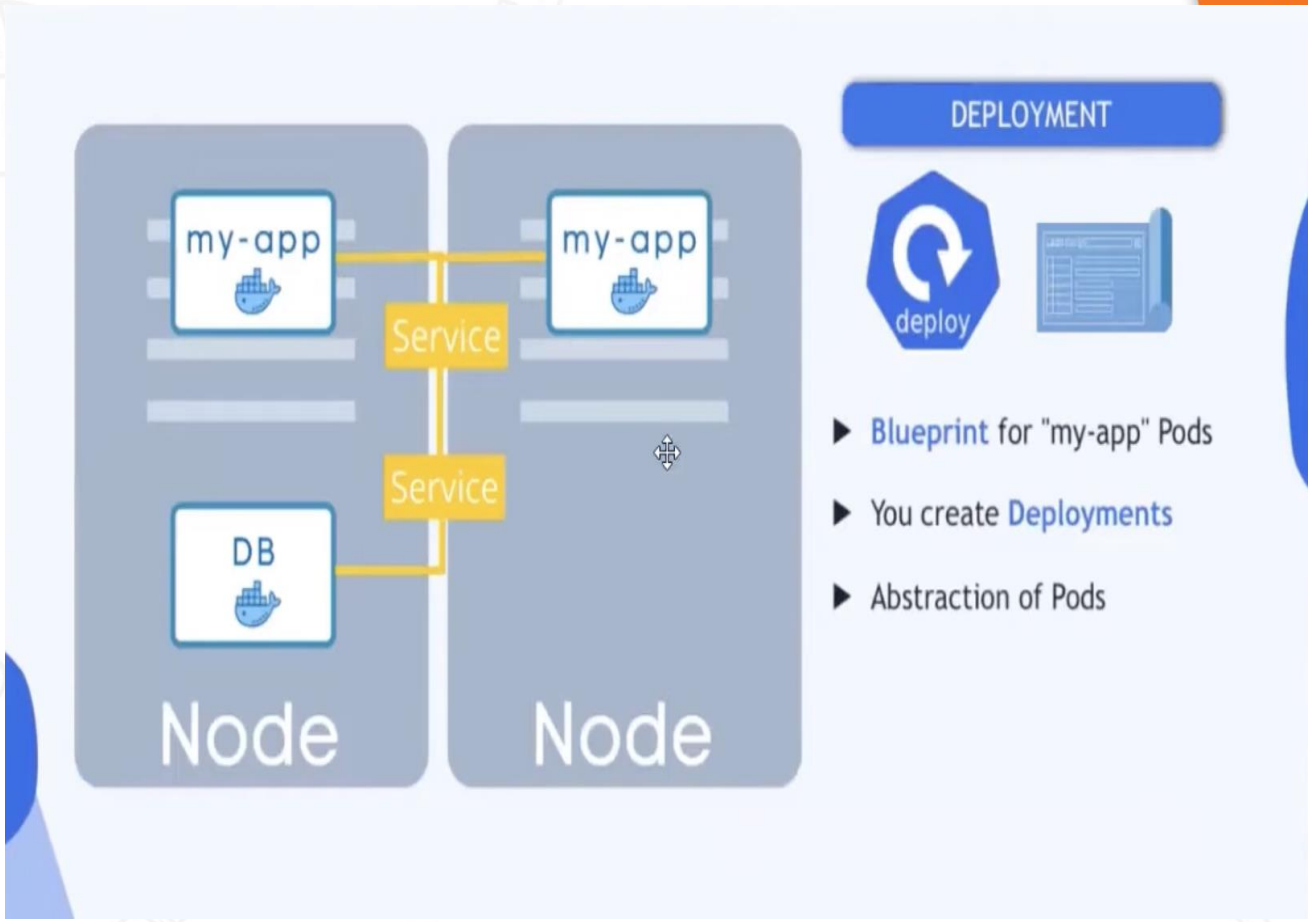
Các thành phần & khái niệm K8s

- ConfigMaps and Secrets
  - Một software ít khi chạy luôn mà không cần config
  - ConfigMap là giải pháp để tạo 1 file config / đặt các ENVironment var hay set các argument khi gọi câu lệnh
  - ConfigMap là một cục config, mà pod nào cần, thì chỉ định là nó cần - giúp dễ dàng chia sẻ file cấu hình
  - Ít khi đặt mật khẩu vào file cấu hình, vậy nên K8s có "secret", để lưu trữ các mật khẩu, token, ... hay những gì cần giữ bí mật



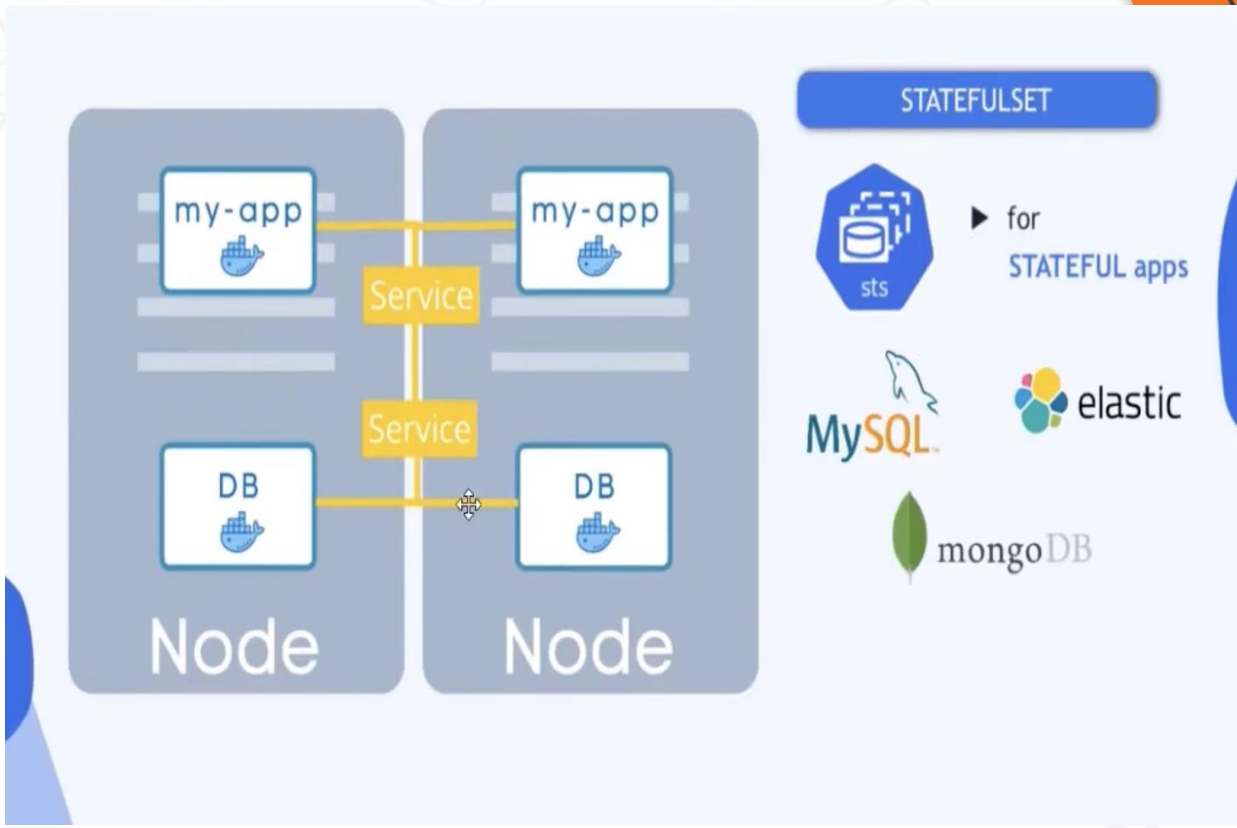
Các thành phần & khái niệm K8s

- Deployment:
  - Là loại chung nhất, khi muốn deploy một dịch vụ nào đó
  - Tạo ra pod bằng cách tạo ra một deployment (hoặc statefulSets, hoặc các khái niệm tương đương)
  - StatefulSets được dùng khi ta cần các service bật lên theo tứ tự nhất định



Các thành phần & khái niệm K8s

- DaemonSet:
  - Thường dành cho các dịch vụ cần chạy trên tất cả các node
- StatefulSet:
  - Là 1 file "manifest" đặt trong thư mục chỉ định bởi kubelet, các pod này sẽ được chạy khi kubelet chạy
  - Không thể điều khiển chúng bằng kubectl



Các thành phần & khái niệm K8s

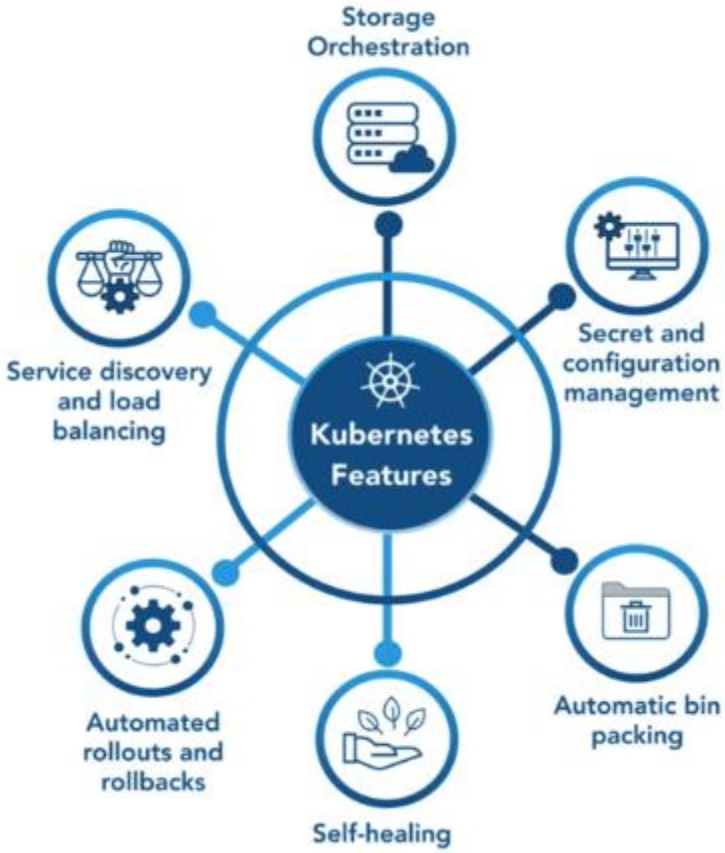
- Dashboard
  - Cho phép xem tổng quan về cluster k8s đang dùng, nó được cài vào k8s như một add-on

    https://github.com/kubernetes/dashboard

Các thành phần & khái niệm K8s

- Monitoring
  - Monitoring trên K8s rất dễ dàng, chỉ cần cài 1 phần mềm có khả năng tích hợp với k8s, nó sẽ hỏi K8s để lấy thông tin về tất cả các pod trong hệ thống

K8s làm được gì?



K8s làm được gì?

- Service discovery và cân bằng tải
  - Kubernetes có thể expose một container sử dụng DNS hoặc địa chỉ IP của riêng nó
  - Nếu lượng traffic truy cập đến một container cao, Kubernetes có thể cân bằng tải và phân phối lưu lượng mạng (network traffic) để việc triển khai được ổn định
- Điều phối bộ nhớ
  - Kubernetes cho phép tự động mount một hệ thống lưu trữ như local storages, public cloud providers, …

K8s làm được gì?

- Tự động rollouts và rollbacks
  - Có thể mô tả trạng thái mong muốn cho các container được triển khai dùng Kubernetes và nó có thể thay đổi trạng thái thực tế sang trạng thái mong muốn với tần suất được kiểm soát
  - Ví dụ, có thể tự động hoá Kubernetes để tạo mới các container cho việc triển khai, xoá các container hiện có và áp dụng tất cả các resource của chúng vào container mới

K8s làm được gì?

- Đóng gói tự động
  - Cung cấp cho Kubernetes một cluster gồm các node mà nó có thể sử dụng để chạy các tác vụ được đóng gói (containerized task)
  - Cho Kubernetes biết mỗi container cần bao nhiêu CPU và bộ nhớ (RAM) thì Kubernetes có thể điều phối các container đến các node để tận dụng tốt nhất các resource

K8s làm được gì?

- Tự phục hồi
  - Kubernetes khởi động lại các containers bị lỗi, thay thế các container, xoá các container không phản hồi lại cấu hình health check do người dùng xác định và không cho các client biết đến chúng cho đến khi chúng sẵn sàng hoạt động
- Quản lý cấu hình và bảo mật
  - Kubernetes cho phép lưu trữ và quản lý các thông tin nhạy cảm như: password, OAuth token và SSH key
  - Có thể triển khai và cập nhật lại secret và cấu hình ứng dụng mà không cần build lại các container image và không để lộ secret trong cấu hình stack

Cài đặt và thực hành cơ bản

- Cài đặt k8s
  - Sử dụng Minikube
  - Sự dụng Docker Desktop
- Thực hành cơ bản
  - kubectl version
  - kubectl cluster-info
  - kubectl get node # kubectl get nodes # node vs nodes đều ok
  - kubectl describe node/docker-desktop

T3H
IT-Institute

VIỆN CÔNG NGHỆ THÔNG TIN T3H

Đào tạo chuyên sâu - Trải nghiệm thực tế

REVIEW PROJECT

www.t3h.edu.vn

T3H
IT-Institute

VIỆN CÔNG NGHỆ THÔNG TIN T3H

Đào tạo chuyên sâu - Trải nghiệm thực tế

THANK YOU

www.t3h.edu.vn