



Nhập môn Kubernetes

By Minh Monmen

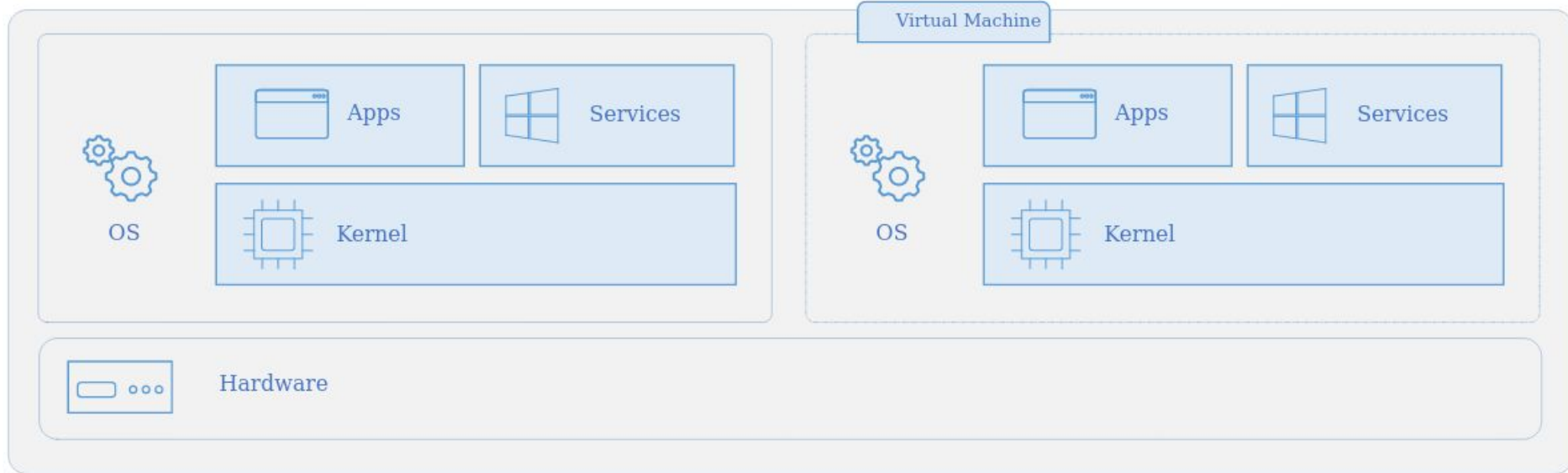


Nội dung

- Docker, Container, Container Orchestration
- Kubernetes vs Docker swarm
- Kiến trúc Kubernetes
- Cài đặt Kubernetes
- Thực hành 1: Cài đặt K8S với rancher
- Namespace và pod
- Thực hành 2: Khởi chạy 1 container với kubectl

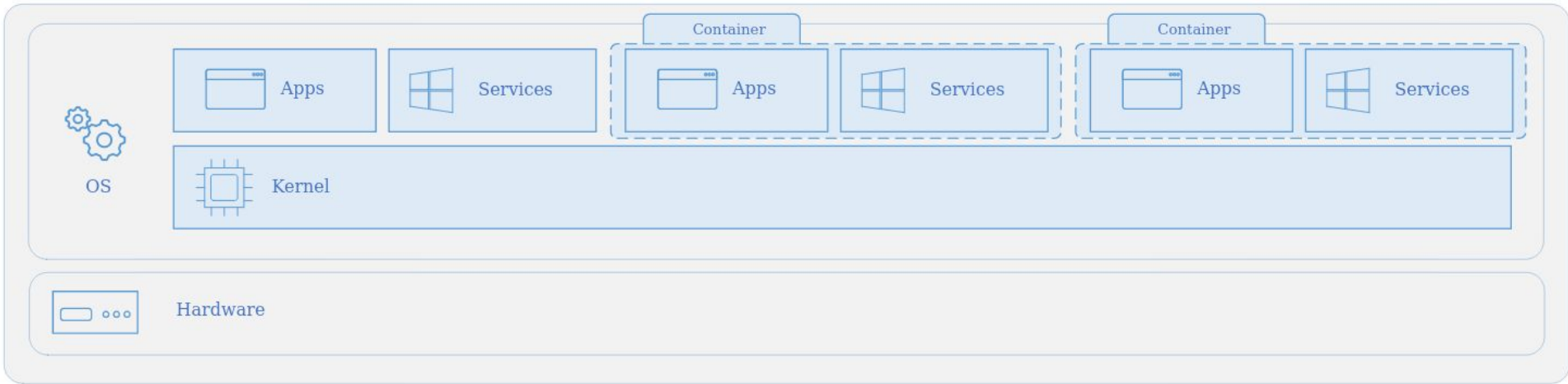


Virtual machine





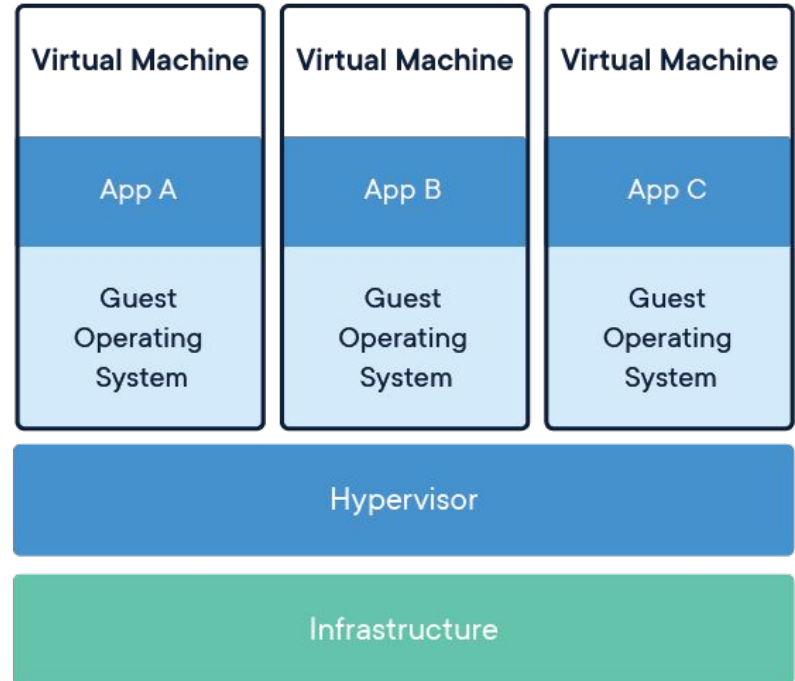
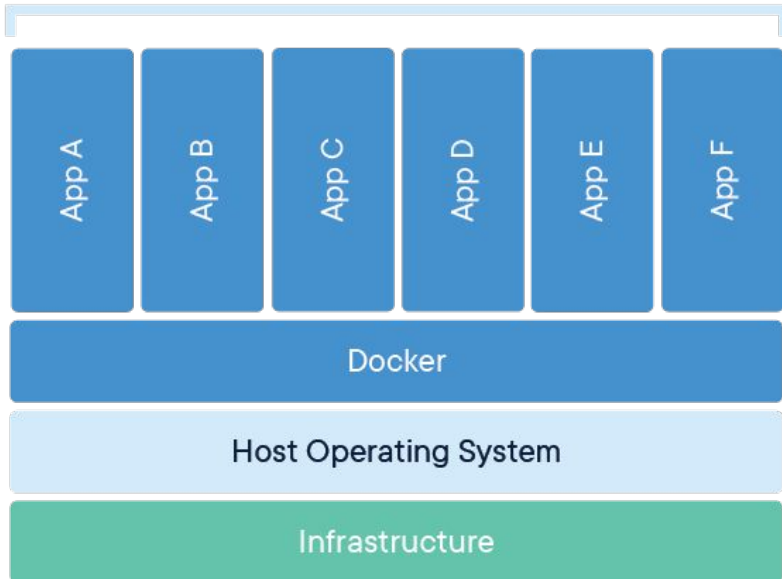
Container



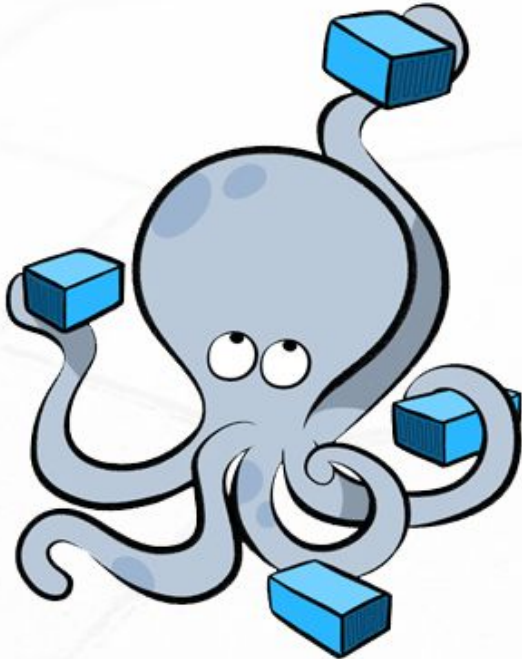


Docker

Containerized Applications



Docker compose

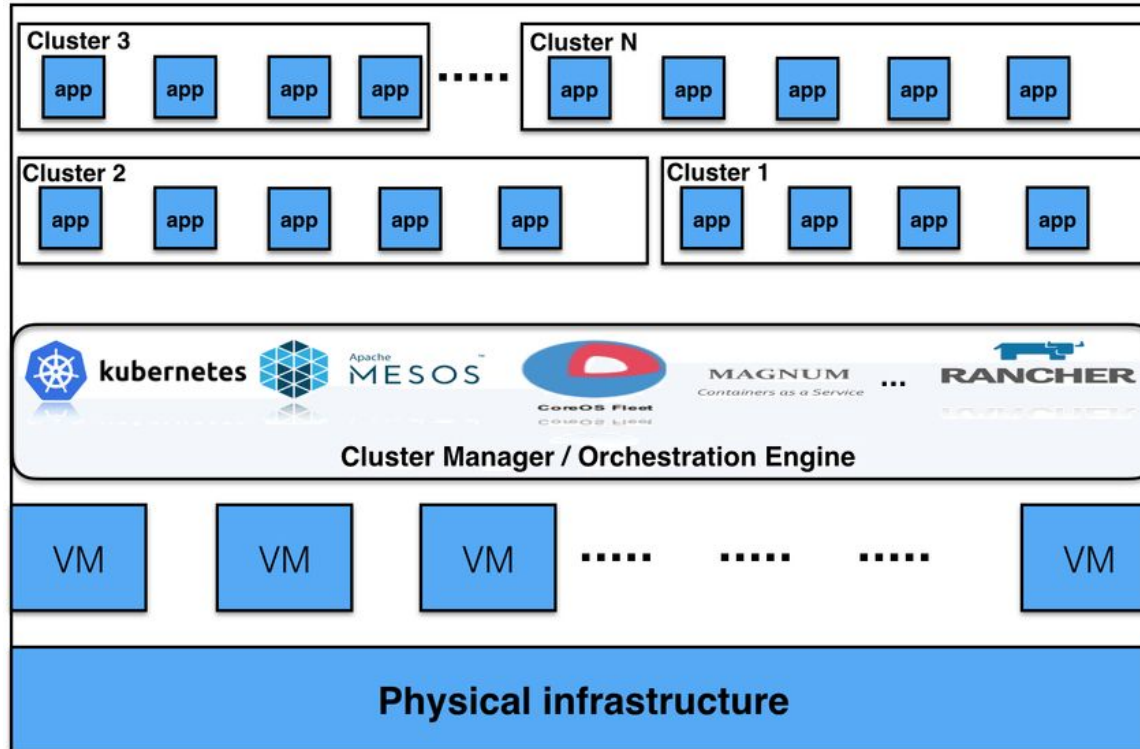


```
php:
  build: php
  ports:
    - "80:80"
    - "443:443"
  volumes:
    - ./php/www:/var/www/html
  links:
    - db
```

\$ docker-compose up



Container Orchestration



Kubernetes (K8S) là gì?

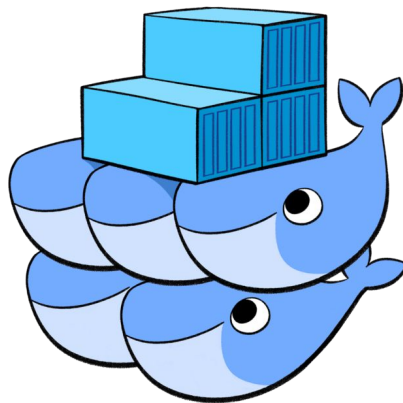
- Tiếng Hy Lạp: Người lái tàu
- Nền tảng Container Orchestration
- Auto Deployment, Scaling, Management
- Opensource từ Google
- Dành cho các hệ thống triển khai container lớn, phức tạp



kubernetes

Kubernetes vs Docker swarm

- Điều là nền tảng Container Orchestration
 - Schedule
 - Scale
 - Overlay network
 - Management
- Master - worker
- High availability



vs





Kubernetes

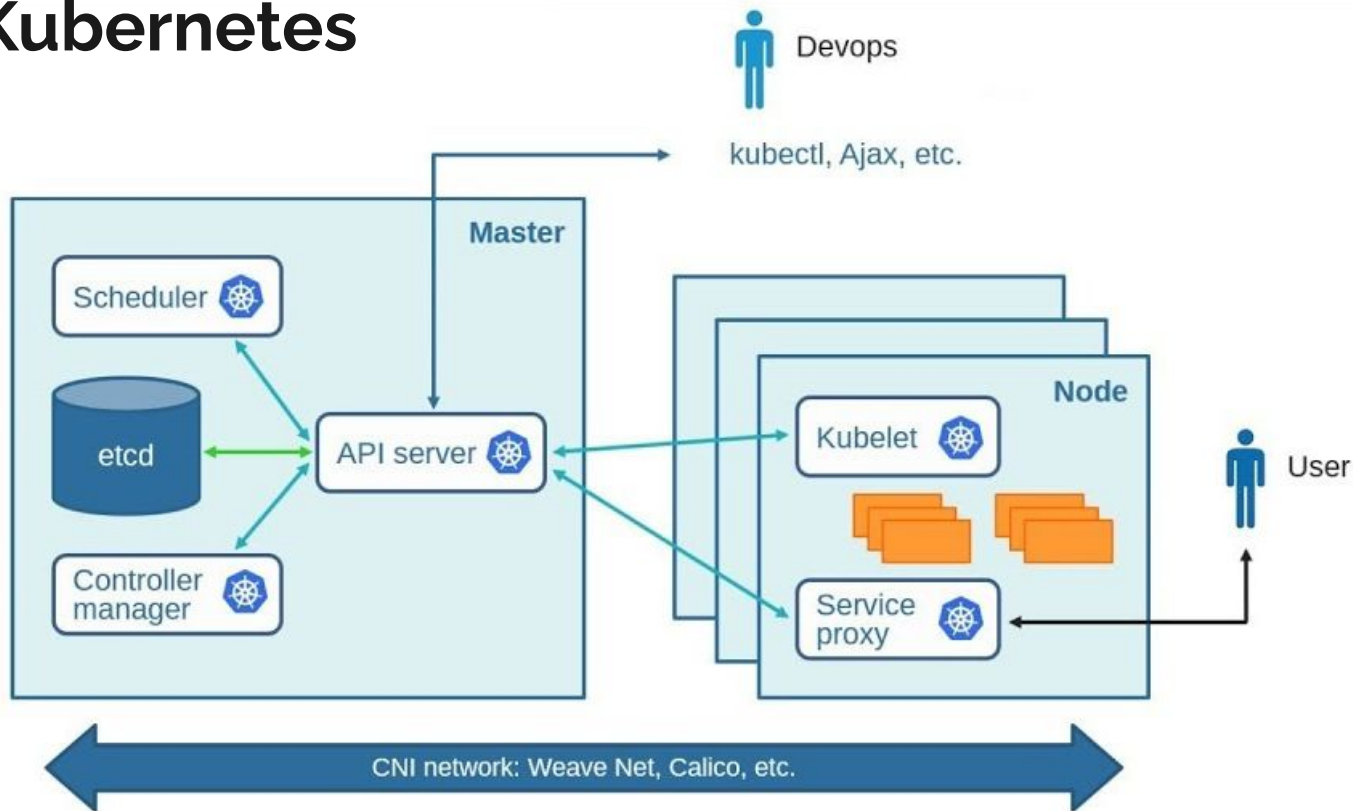
vs

Docker swarm

- Phức tạp
- Đảm bảo cluster state
- Auto scaling
- Pod
- Manual TLS
- Built-in GUI
- Built-in logging, monitoring

- Đơn giản hơn
- Deploy container nhanh
- Manual scaling
- Task
- Auto TLS
- 3rd GUI
- 3rd logging, monitoring

Kiến trúc Kubernetes





Cài đặt Kubernetes

- Có rất nhiều cách để cài K8S
- Chính chủ: **kubeadm**
- Local: **minikube**, **Docker desktop**
- Lite: **k3s**, **microk8s**
- Cloud / VM: **rancher**, **kops**, **kubespray**, **kubesphere**
- Managed: **GKE**, **EKS**, **AKS**, **DOKS**,...
- More: [Installers - Awesome-kubernetes](#)



Thực hành 1: Cài đặt K8S bằng rancher

- Tại sao lại dùng Rancher?
- RKE là gì?



Khởi chạy rancher

- Hướng dẫn ở <https://rancher.com/quick-start/>
- Chuẩn bị VM linux (có thể chạy 1 hoặc nhiều VM)
- Chạy rancher bằng docker:

```
sudo docker run -d --restart=unless-stopped -p 8080:80 -p 8443:443  
rancher/rancher:v2.4.6
```

- Truy cập `https://<host-ip>:8443`

Tạo cluster với rancher



Add Cluster - Select Cluster Type



With RKE and new nodes in an infrastructure provider



With a hosted Kubernetes provider



Cancel



Tạo cluster với rancher

Rancher Docs: Cluster Configuration Reference

Add Cluster - Custom

Cluster Name *

[Add a Description](#)

demo

▶ Member Roles


Control who has access to the cluster and what permission they have to change it.

▶ Labels & Annotations

Configure labels and annotations for the cluster.

None

Cluster Options

[Edit as YAML](#) 

[Expand All](#)

▶ Kubernetes Options

Customize the kubernetes cluster options

▶ Private Registry

Configure a default private registry for this cluster. When enabled, all images required for cluster provisioning and system add-ons startup will be pulled from this registry.

▶ Advanced Options

Customize advanced cluster options

▶ Authorized Endpoint

Enabling the authorized cluster endpoint allows direct communication with the cluster, bypassing the API proxy. Authorized endpoints can be retrieved by generating a kubeconfig for the cluster.

[Next](#)

[Cancel](#)

Tạo cluster node

Add Cluster - Custom

Cluster Options

▼ Customize Node Run Command
Editing node options will update the command you will run on your existing machines

1

Node Options

Choose what roles the node will have in the cluster

Node Role

☒ etcd

☒ Control Plane

☒ Worker

[Show advanced options](#)

2

Run this command on one or more existing machines already running a supported version of Docker.

```
sudo docker run -d --privileged --restart=unless-stopped --net=host -v /etc/kubernetes:/etc/kubernetes -v /var/run:/var/run rancher/rancher-agent:v2.4.7 --server https://10.0.0.9:8443 --token rsczsg6xhdmfhrkrwn48nbv4xm4rxndlbztngbl6spr4twhl8jxfn --ca-checksum df450ab2b363375b8d8a42fdbb97cfff009d20024296fa3fc9f98f97d9c4ef22 --etcd --controlplane --worker
```



Done

Cluster sau khi khởi tạo

Dashboard: demo

Launch kubectl Kubeconfig File

Provider: Custom

Kubernetes Version: v1.18.6

Created: Yesterday at 8:37 PM

Enable Monitoring to see live metrics

18%

CPU

0.7 of 4 Reserved

1%

Memory

0.1 of 15.3 GiB Reserved

5%

Pods

10 of 220 Used



Etcd



Controller Manager



Scheduler



Nodes



Events

Events of current Cluster



Tương tác với K8S

- Kubectl là CLI dùng để quản lý, tương tác với cluster K8S
- [Cài đặt và cấu hình kubectl](#)
- Kubeconfig file chứa thông tin để kubectl authen với K8S
- Set kubeconfig path bằng lệnh:
 - `export KUBECONFIG=/absolute/path/to/kubeconfig`
- Thử view all các resource đang có
 - `kubectl get all --all-namespaces`



Tương tác với K8S

- Cài đặt K8S dashboard:
 - `kubectl apply -f`
<https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.4/aio/deploy/recommended.yaml>
 - `kubectl -n kubernetes-dashboard port-forward --address 0.0.0.0 svc/kubernetes-dashboard 8444:443`
- Tại sao không dùng Rancher GUI?



Đăng nhập K8S Dashboard

Kubernetes Dashboard

☒ Token

Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the [Authentication](#) section.

☐ Kubeconfig

Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the [Configure Access to Multiple Clusters](#) section.

Enter token *

Sign in



Đăng nhập K8S dashboard

- Tạo tài khoản admin-user (chỉ cần tạo 1 lần)
 - `kubectl apply -f`
<https://gist.githubusercontent.com/minhpg331/6450e2fe9722f6aca4d964312ddc0a060/raw/admin-user.yaml>
- Get token từ tài khoản admin-user
 - `kubectl -n kube-system describe secret $(kubectl -n kube-system get secret | grep admin-user | awk '{print $1}') | grep "token:" | awk '{print $2}'`

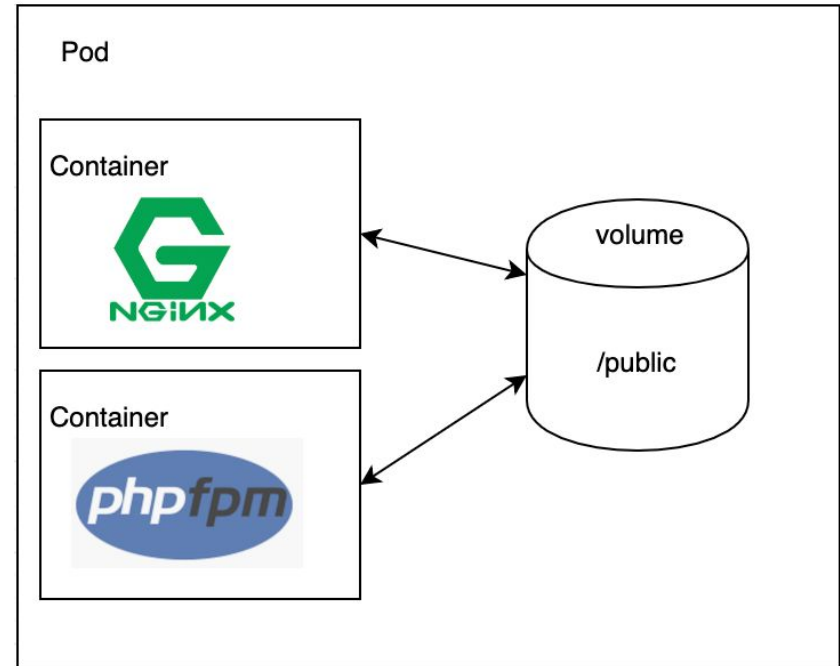


Namespace là gì?

- Virtual cluster
- Scoped name
- DNS
- Tương tác với namespace:
 - `kubectl create namespace <name>`
 - `kubectl get namespace`
- Tương tác với các object trong namespace
 - `kubectl -n <namespace-name> <command>`

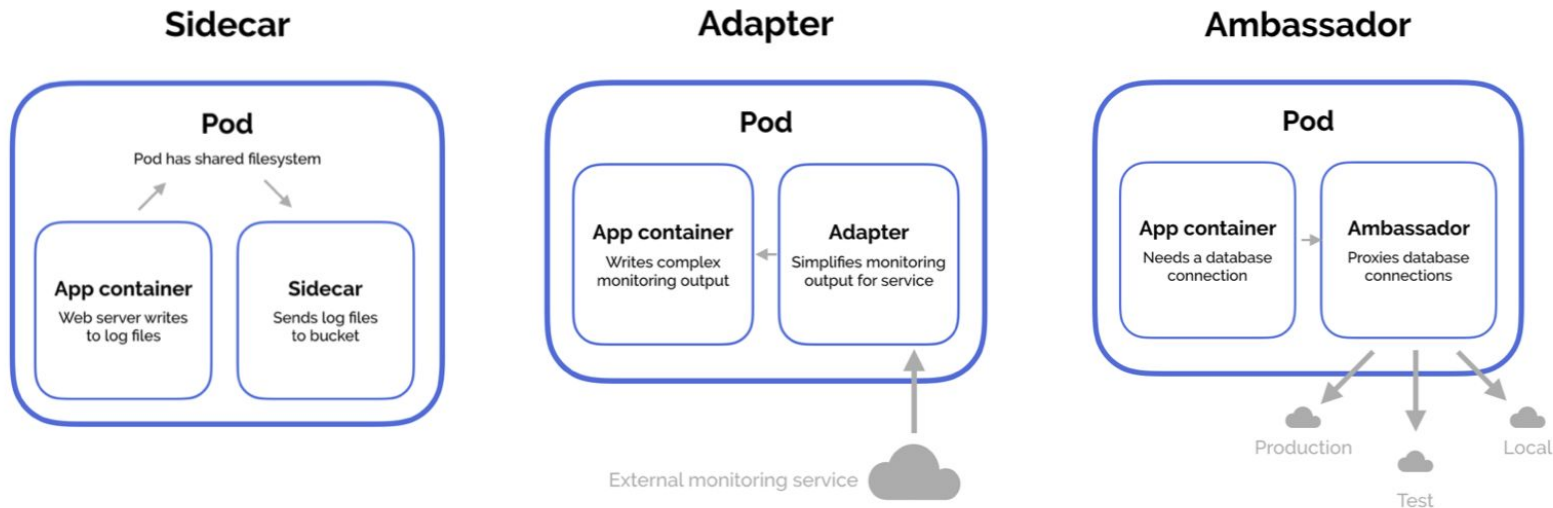
Pod là gì?

- Deployment unit của K8S
- Chứa 1 hoặc nhiều container
- Share network, storage
- Call 127.0.0.1



Thiết kế multi-container pod

- [Multi-Container Pod Design Patterns - CKAD Course](#)





Thực hành 2: Khởi chạy 1 pod với kubectl

- Đăng nhập vào máy chủ để sử dụng kubectl
- Tạo namespace mang tên mình
- Run 1 container bằng kubectl
 - `kubectl -n <ns> run test --port 8000 --image jwilder/whoami`
- Kiểm tra pod đã chạy trên dashboard hoặc bằng
 - `kubectl -n <ns> get pod test`
 - `kubectl -n <ns> describe pod test`



Thực hành 2: Khởi chạy 1 pod với kubectl

- Forward port của pod ra host bằng kubectl port-forward
 - `kubectl -n <ns> port-forward --address 0.0.0.0 pod/test 8000:8000`
- Truy cập `http://<host chạy kubectl>:8000`
- Delete pod sau khi sử dụng
 - `kubectl -n <ns> delete pod test`
- Usecase: Phục vụ mục đích debug cluster
 - `kubectl -n <ns> run -i --tty --rm test --image nicolaka/netshoot -- /bin/bash`

Định nghĩa pod với yaml

```
---
apiVersion: v1
kind: Pod
metadata:
  name: demo-sidecar          # Pod name
  labels:
    something: something-else # Pod label
spec:
  containers:                # Array of containers
  - name: web                 # Container name
    image: some-image         # Container image
    ports:
      - containerPort: 8080   # Expose container port
```



Thực hành 3: Khởi chạy pod với yaml

- <https://github.com/minhpg331/demo-sidecar>
- Tạo pod chạy 2 container trong namespace của mình
- `kubectl -n <namespace> apply -f pod.yaml`
- Expose nginx qua port 80
- Dùng port-forward để test xem app đã chạy chưa



Tài liệu liên quan

- <https://ramitsurana.gitbook.io/awesome-kubernetes/docs>
- <https://kubernetes.io/docs/concepts/overview/components/>
- <https://kubernetes.io/docs/reference/kubectl/cheatsheet/>
- <https://www.katacoda.com/courses/kubernetes>
- <https://rancher.com/docs/rancher/v2.x/en/cluster-provisioning/rke-clusters/custom-nodes/>
- <https://matthewpalmer.net/kubernetes-app-developer/articles/multi-container-pod-design-patterns.html>



Thanks for watching!