## **🚀 1. Kiến thức nền tảng cần có trước khi học Kubernetes**

### **✅ Linux cơ bản:**

* Câu lệnh terminal: ls, cd, cat, grep, curl, top, ...
* Quản lý tiến trình, network, file permission

### **✅ Docker / Container:**

* Hiểu cách tạo, chạy, build image Docker
* Dockerfile, docker-compose
* Các khái niệm: image, container, volume, network

### **✅ Networking cơ bản:**

* IP, DNS, Port, NAT, Load Balancer
* Khái niệm service discovery

## **🧠 2. Kiến thức Kubernetes cốt lõi**

### **🔹 Kiến trúc Kubernetes:**

* **Pod**, **Node**, **Cluster**, **Control Plane**
* **Master vs Worker node**
* **Kubelet, kube-proxy, API server, etcd**

### **🔹 Các thành phần quan trọng:**

* **Pod**: Đơn vị nhỏ nhất trong K8s
* **ReplicaSet**: Đảm bảo có X pod đang chạy
* **Deployment**: Quản lý việc cập nhật phiên bản ứng dụng
* **Service**: Cung cấp địa chỉ truy cập vào Pod (ClusterIP, NodePort, LoadBalancer)
* **ConfigMap & Secret**: Quản lý cấu hình & thông tin nhạy cảm
* **Volume & PersistentVolume**: Lưu trữ dữ liệu bền vững
* **Namespace**: Chia vùng quản lý tài nguyên

## **🧰 3. Kỹ năng thao tác thực tế**

### **💡 Biết dùng công cụ:**

* kubectl: CLI để tương tác với cluster
* Viết YAML file để khai báo resource
* Helm: Dùng để quản lý template và cài ứng dụng nhanh

### **☁️ Biết triển khai môi trường:**

* Local: Minikube, Kind, k3s
* Cloud: GKE (Google), EKS (AWS), AKS (Azure)

### **🛠 Monitoring & Debugging:**

* kubectl logs, describe, exec
* Quan sát pod, service hoạt động
* Dùng Prometheus, Grafana, hoặc K9s

## **🔐 4. Các khái niệm nâng cao hơn (sau này học thêm)**

* Horizontal Pod Autoscaler (HPA)
* NetworkPolicy
* RBAC (phân quyền truy cập)
* Ingress Controller
* StatefulSet
* CI/CD với Kubernetes (ArgoCD, Jenkins, GitLab CI)
* Service Mesh (Istio, Linkerd)

### **✅ Bước 1: Tạo Namespace mới**

Namespace là cách nhóm các tài nguyên trong Kubernetes lại với nhau. Bạn có thể tạo namespace mới cho riêng mình bằng cách dùng lệnh sau:

bash

Sao chépChỉnh sửa

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" create namespace <tên\_namespace\_của\_bạn>

Ví dụ nếu bạn muốn tạo một namespace tên là my-namespace:

bash

Sao chépChỉnh sửa

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" create namespace my-namespace

👉 Namespace này sẽ giúp bạn phân tách các ứng dụng, tài nguyên để dễ quản lý hơn.

### **✅ Bước 2: Tạo Service Account (Tài khoản người dùng)**

Bây giờ, bạn muốn tạo một **Service Account** riêng để quản lý quyền truy cập. Bạn có thể tạo tài khoản này trong Kubernetes, rồi sử dụng nó để thao tác thay cho tài khoản admin.

Tạo một **ServiceAccount** trong namespace bạn vừa tạo (ví dụ my-namespace):

bash

Sao chépChỉnh sửa

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" -n my-namespace create serviceaccount <tên\_account\_của\_bạn>

Ví dụ tạo một ServiceAccount tên là my-user:

bash

Sao chépChỉnh sửa

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" -n my-namespace create serviceaccount my-user

### **✅ Bước 3: Gán Role cho Service Account**

Service Account cần quyền (role) để có thể thao tác trong Kubernetes. Mình sẽ tạo một **RoleBinding** để gán quyền truy cập cho my-user (hoặc tên tài khoản bạn đã tạo).

Ví dụ nếu bạn muốn cho tài khoản này quyền quản lý pod trong namespace my-namespace, bạn có thể tạo RoleBinding như sau:

bash

Sao chépChỉnh sửa

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" -n my-namespace create rolebinding my-user-rolebinding \

--role=edit --serviceaccount=my-namespace:my-user

* --role=edit là quyền edit (quản lý tài nguyên như Pods, Deployments, v.v. trong namespace đó).
* Bạn có thể thay --role=edit thành --role=admin nếu muốn quyền cao hơn (admin của namespace).

### **✅ Bước 4: Lấy kubeconfig cho Service Account**

Bây giờ bạn cần lấy kubeconfig cho tài khoản của mình để sử dụng thay cho tài khoản admin.

Đầu tiên, tạo token cho my-user:

bash

Sao chépChỉnh sửa

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" -n my-namespace create token my-user

Lệnh này sẽ trả về một token mà bạn cần để đăng nhập với tài khoản này.

### **✅ Bước 5: Cấu hình lại kubeconfig của bạn để sử dụng tài khoản mới**

Bạn cần cập nhật file kubeconfig của mình để sử dụng tài khoản my-user. Cấu hình sẽ trông như thế này:

yaml

Sao chépChỉnh sửa

apiVersion: v1

clusters:

- cluster:

server: https://<IP\_public>:<port> # Đặt địa chỉ IP public của cụm K8s ở đây

name: my-cluster

contexts:

- context:

cluster: my-cluster

user: my-user

namespace: my-namespace

name: my-user-context

current-context: my-user-context

kind: Config

preferences: {}

users:

- name: my-user

user:

token: <token\_của\_bạn> # Token bạn nhận được từ lệnh trước

👉 Lưu ý: Thay <IP\_public>:<port> bằng IP public của cụm Kubernetes và port API server.

## Dự án của tôi:

Bạn có 1 hệ thống web:

* Frontend (React)
* Backend (Node.js)
* Database (PostgreSQL)
* Redis (cache)
* Worker (Xử lý hàng đợi)

👉 Mỗi cái này bạn build thành container riêng.

✅ Dùng **K8s để**:

* Deploy từng thằng vào Pod
* Cấu hình service để chúng kết nối được
* Quản lý log, scaling, volume, v.v
* Rollback khi backend bị lỗi
* Phân quyền: dev chỉ thấy namespace của họ

—----------------------------------------------------------------------------------------------------------------------------------

Áp dụng các file YAML vào cluster:

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" apply -f D:\thuctap\k8s-manifests\secrets.yaml

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" apply -f D:\thuctap\k8s-manifests\flask-db-deployment.yaml

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" apply -f D:\thuctap\k8s-manifests\redis-deployment.yaml

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" apply -f D:\thuctap\k8s-manifests\backend-deployment.yaml

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" apply -f D:\thuctap\k8s-manifests\celery-worker-deployment.yaml

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" apply -f D:\thuctap\k8s-manifests\frontend-deployment.yaml

Kiểm tra xem các pod có đang chạy với image mới không:

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" get pods -n dangdangtest

—nếu có pod lôiz thì kiểm tra log:

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" logs <pod-name> -n dangdangtest

Sử dụng port-forward để truy cập từ máy local:

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" port-forward svc/frontend-service 3001:80 -n dangdangtest

sau khi sửa thì áp dụng lại chúng để Kubernetes sử dụng image mới.

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" apply -f D:\thuctap\k8s-manifests\frontend-deployment.yaml

lệnh xóa kube:

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" delete pod -l app=frontend -n dangdangtest

lệnh truy cập vào psql từ back end:  
kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" exec -it backend-fcbb5ff46-newid -n dangdangtest -- psql -h flask-db-service -U postgres -d postgres

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" exec -it -n dangdangtest frontend-79d9b96856-vh9rn -- /bin/bash

kubectl --kubeconfig="D:\thuctap\bocu7deq-kubeconfig" exec -it -n dangdangtest backend-fcbb5ff46-l7ztx -- /bin/bash