



Deployment cơ bản

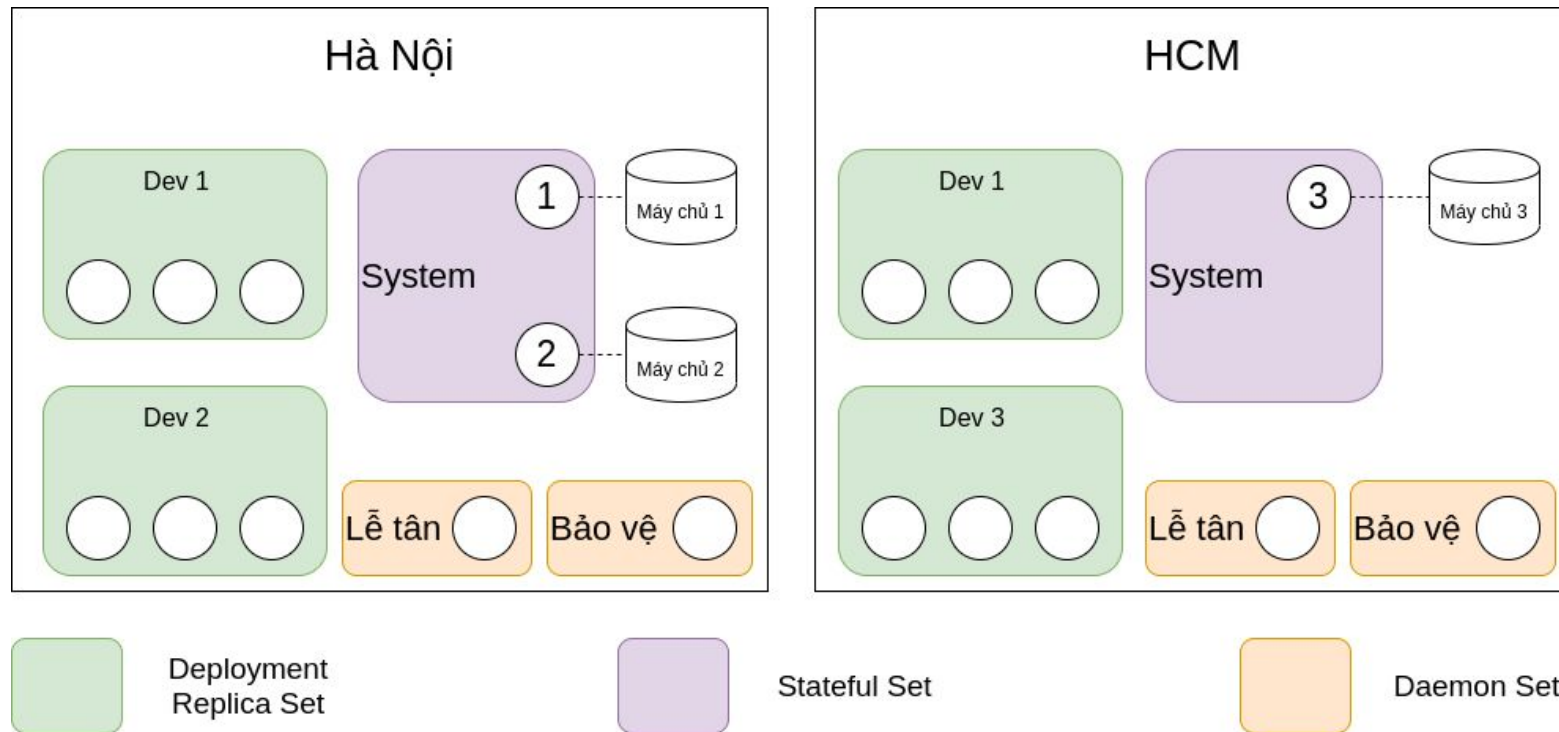
By Minh Monmen



Nội dung

- Cách triển khai ứng dụng trên K8S
- Replica Set, Deployment, Stateful Set, Daemon Set
- Production System Design
- Thực hành 1: Triển khai ứng dụng NodeJS trên K8S
- ConfigMap, Secret
- Thực hành 2: Mounting config

3 cách triển khai ứng dụng

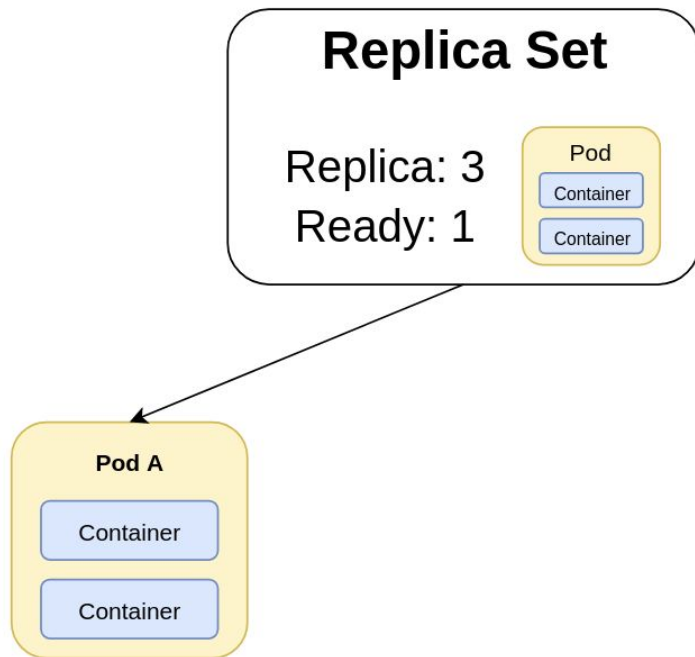


Replica Set

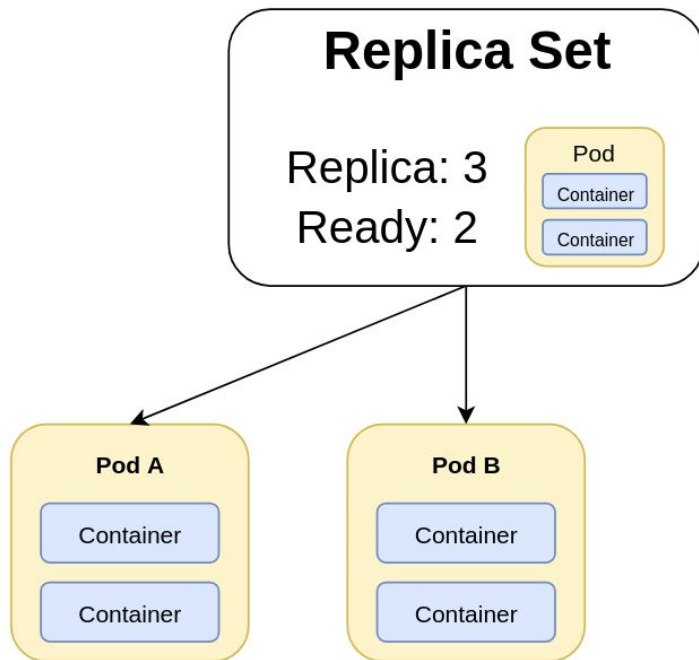
- Định nghĩa template chạy 1 pod
- Quản lý số lượng pod của 1 app



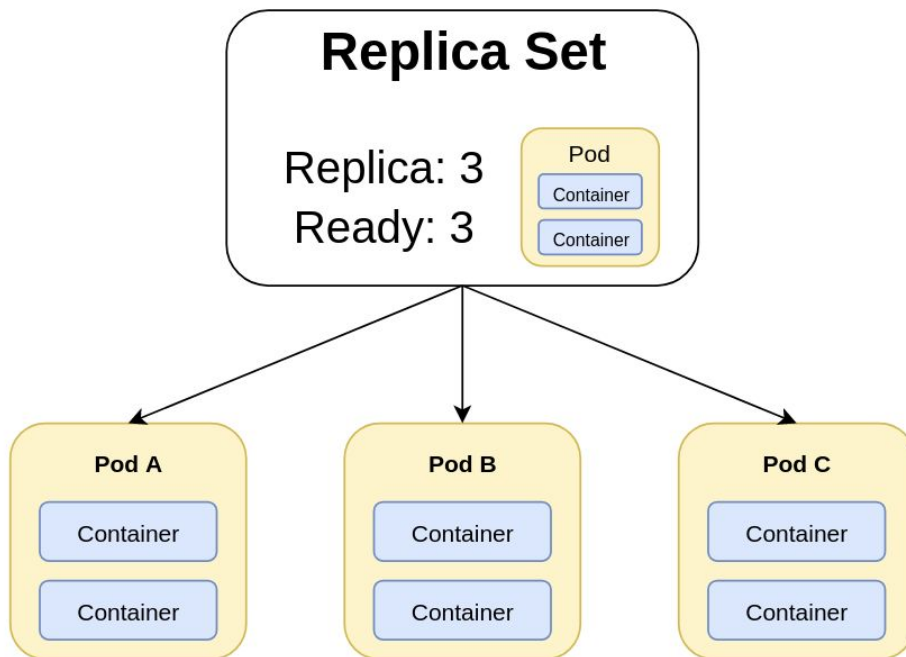
Replica Set



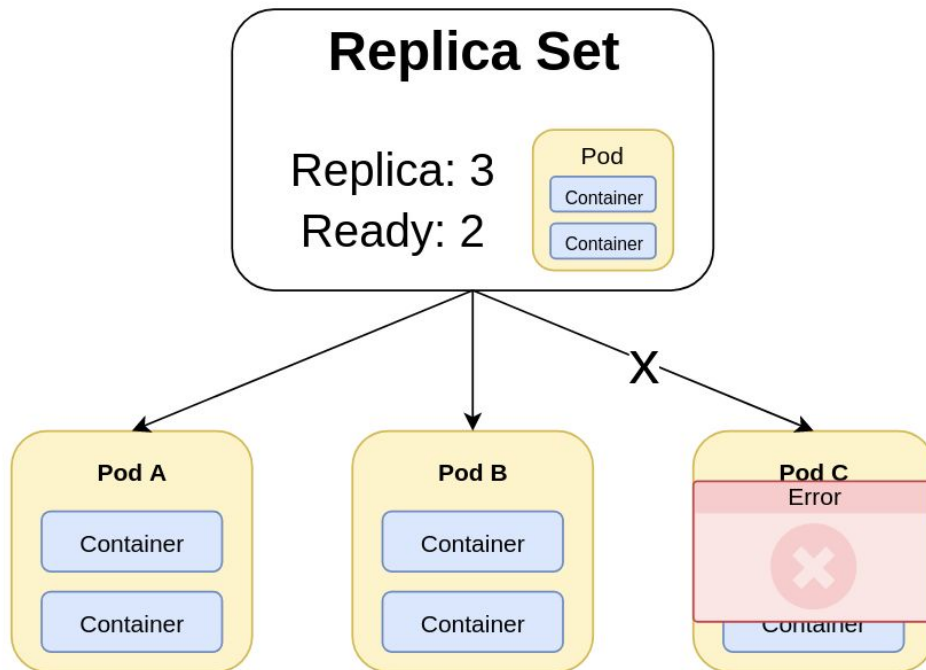
Replica Set



Replica Set

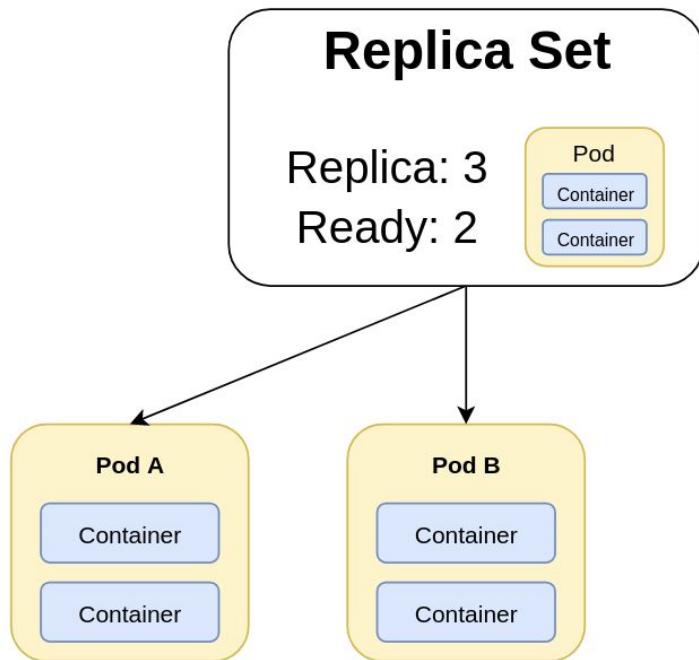


Replica Set

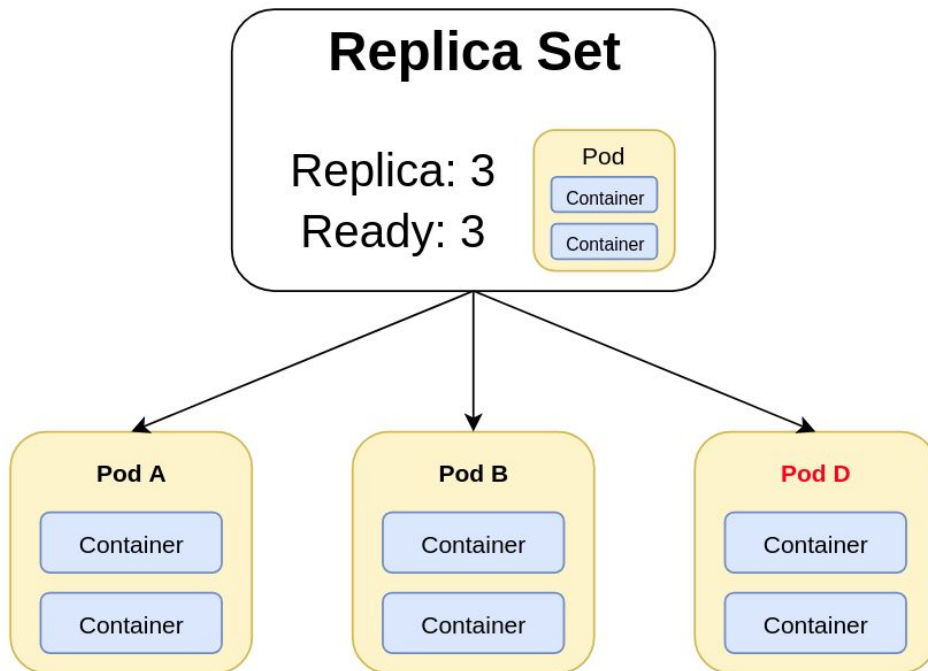




Replica Set



Replica Set





Replica Set

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: demo-replicaset
  labels:
    something: day-la-label-cua-replicaset
spec:

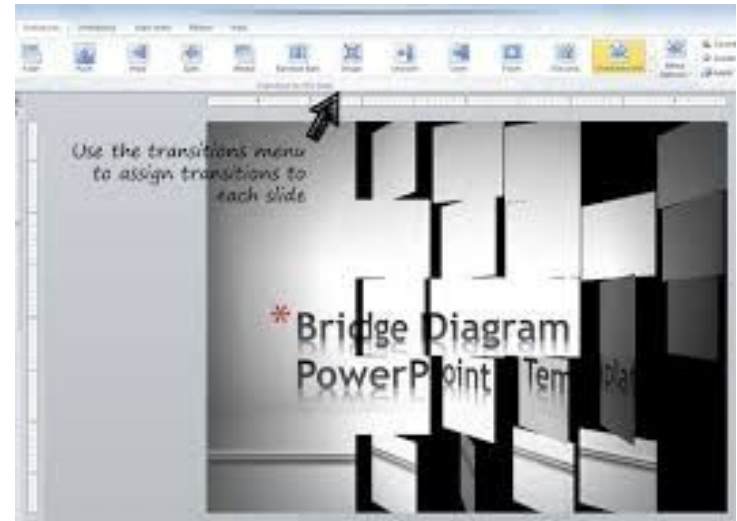
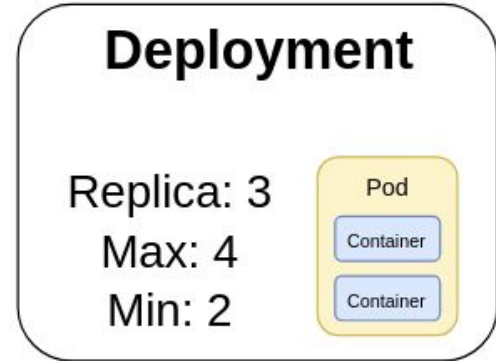
  replicas: 3                # Number of pods

  selector:                  #
    matchLabels:             # This selector MUST MATCH below labels
      app: sidecar-v1        #
  template:

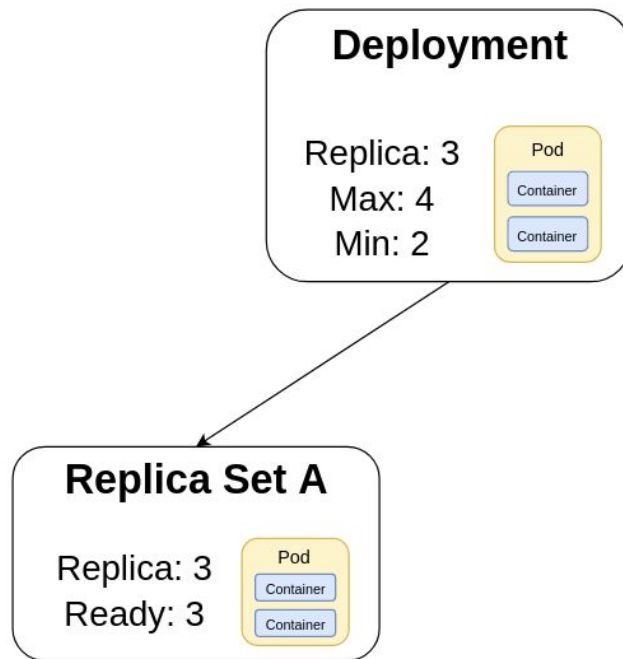
    # Pod template
    metadata:
      name: demo-pod
      labels:
        app: sidecar-v1      # Pod labels
    spec:
      containers:
        - name: web
          image: some-image
          ports:
            - containerPort: 8080
```

Deployment

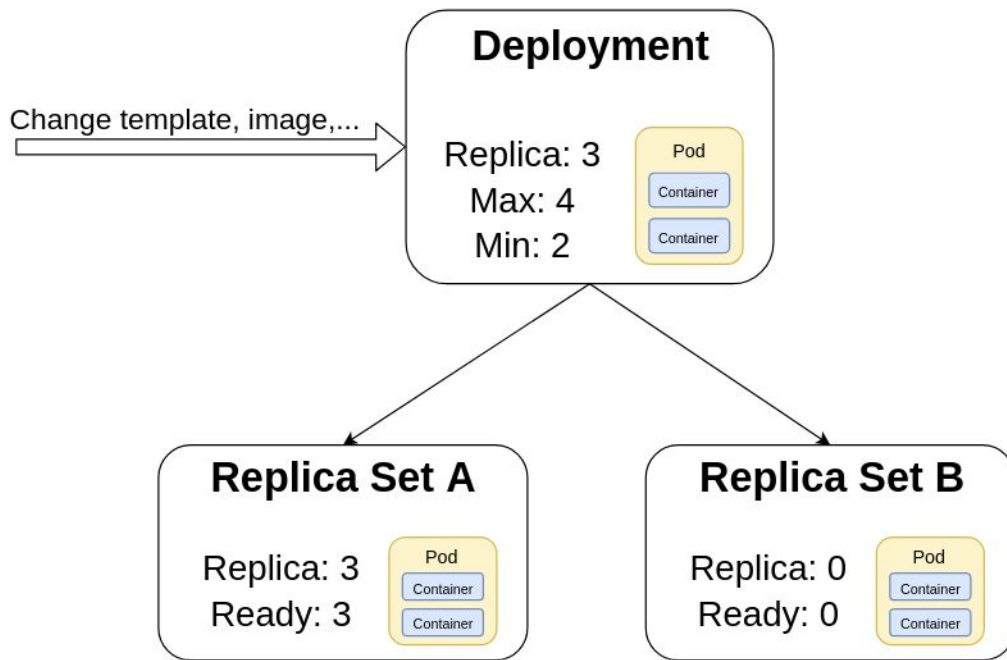
- Quản lý Replica Set (tự động tạo, xóa)
- Quản lý việc chuyển giao giữa các Replica Set (chuyển cảnh)
- Phù hợp với ứng dụng stateless
- Tên pod được đặt random



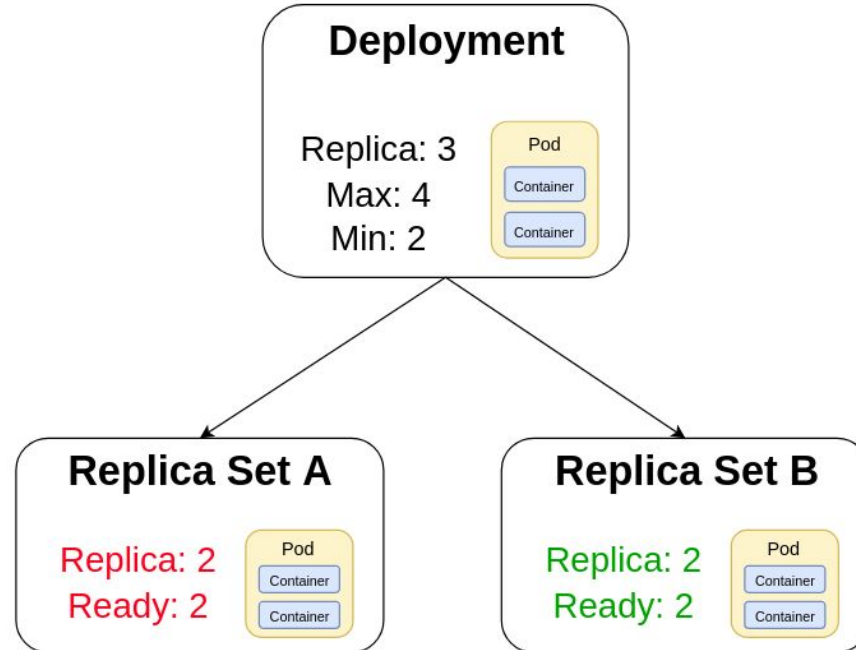
Deployment



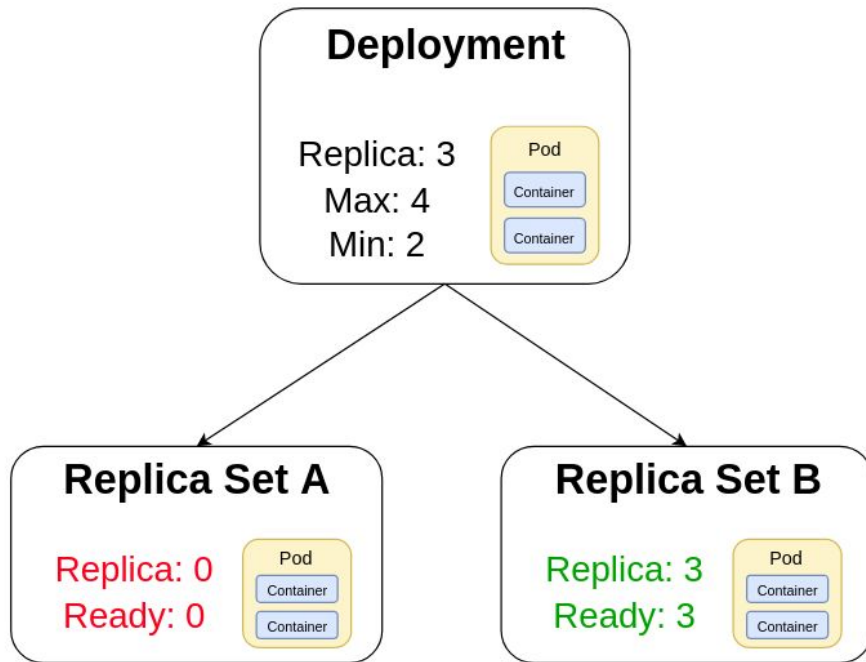
Deployment



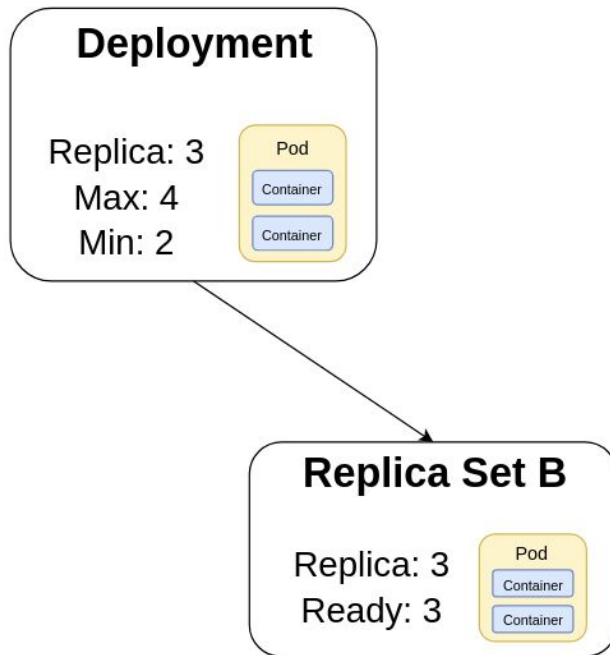
Deployment



Deployment



Deployment





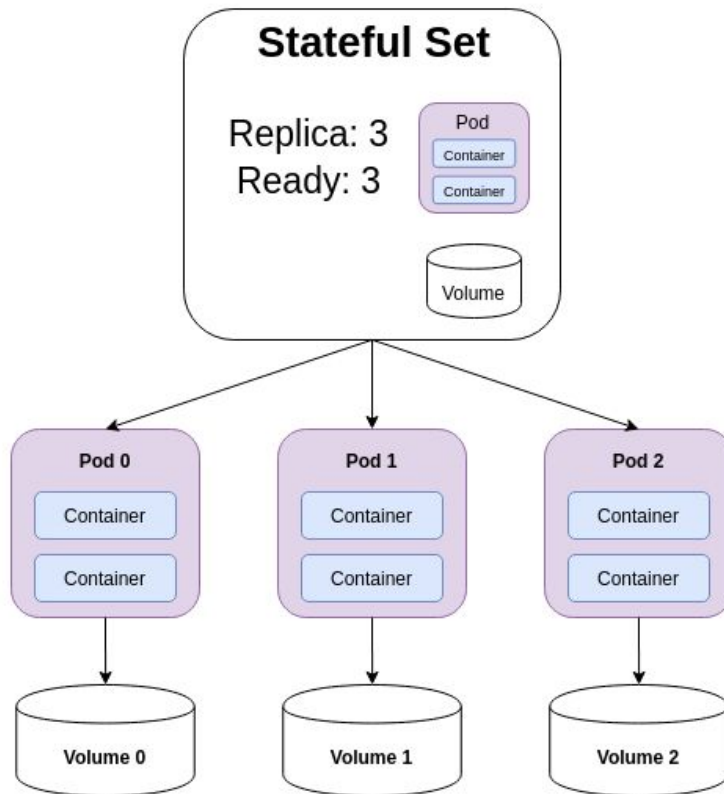
Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: demo-deployment
  labels:
    something: day-la-label-cua-deployment
spec:
  strategy:
    type: rollingUpdate      # Strategy name
    rollingUpdate:
      maxSurge: 1           # Number of over-scheduled pod
      maxUnavailable: 0     # Number of unavailable pod

  replicas: 3               # Number of pods
  selector:
    matchLabels:
      app: sidecar-v1      # This selector MUST MATCH below labels
                          #

  # Pod template
  template:
    metadata:
      name: demo-pod
      labels:
        app: sidecar-v1    # Pod labels
    spec:
      containers:
        - name: web
          image: some-image
          ports:
            - containerPort: 8080
```

Stateful Set

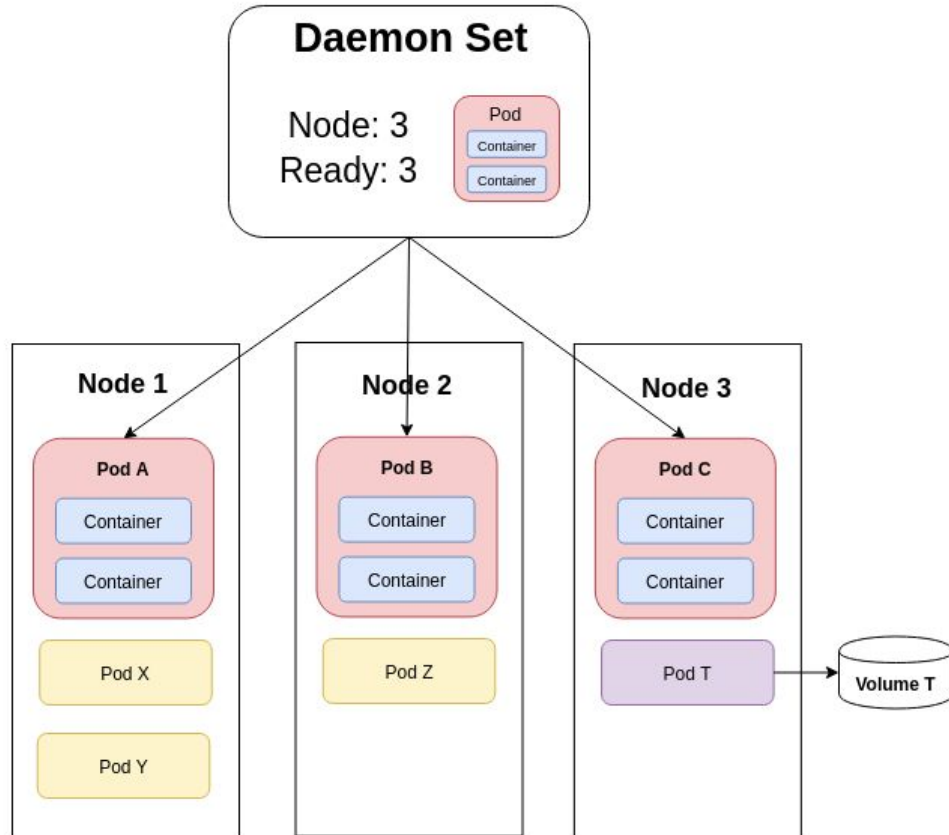




Stateful Set

- Pod được dựng lên CÓ THỨ TỰ và được đánh số 0, 1, 2
- Triển khai ứng dụng có ghi file như Session, DB
- Rolling update dễ gây downtime
- Best practice: NÊN TRÁNH nếu có thể
- Thông thường chỉ sử dụng cho mục đích không quá quan trọng như lưu data của Monitoring service chạy trong cluster

Daemon Set

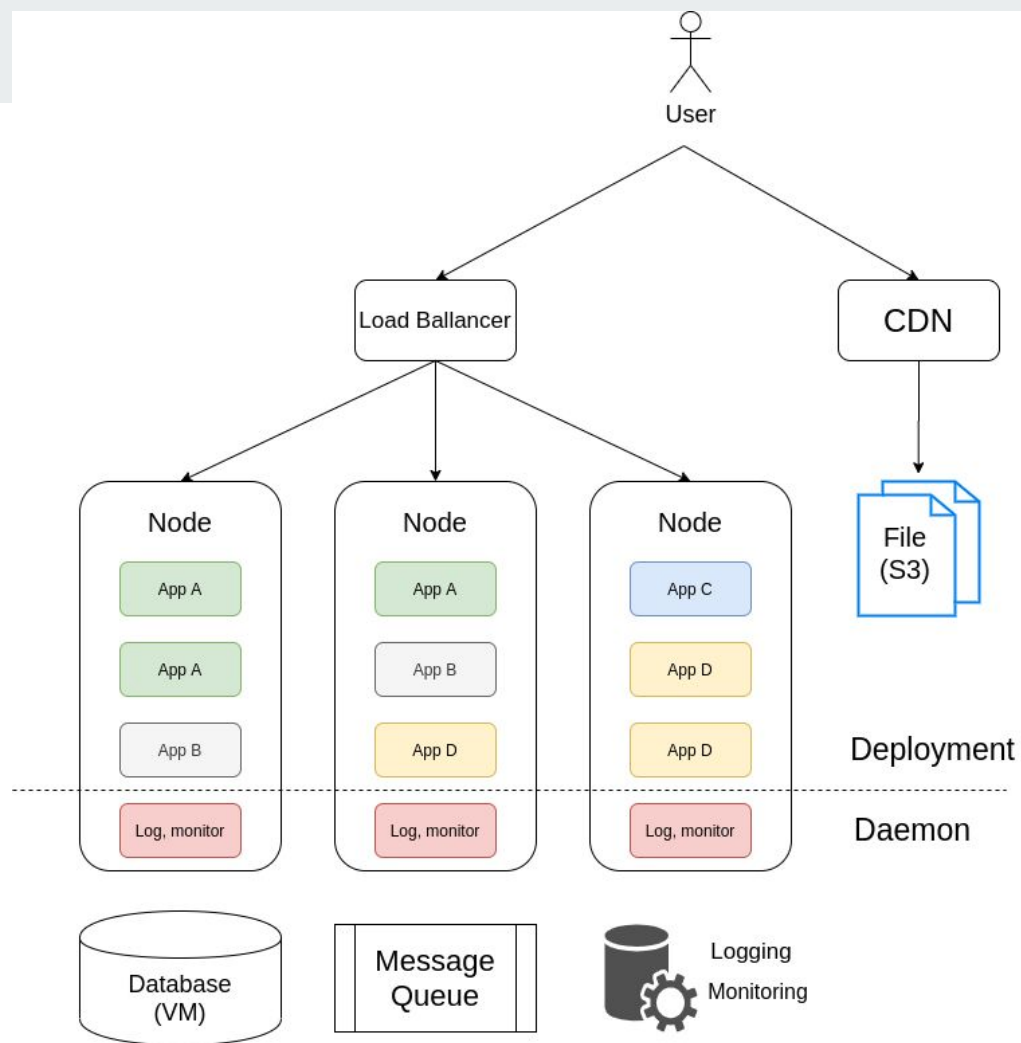




Daemon Set

- Đảm bảo
 - Mỗi host chỉ có 1 pod duy nhất
 - Số pod bằng số host (tất cả hoặc được chỉ định)
- Được khởi chạy trước tiên khi node tham gia cluster
- Phù hợp với các tác vụ của Cluster như logging, monitoring

Production System





Thực hành 1: Triển khai NodeJS API

- <https://github.com/minhpg331/demo-deployment>
- Tạo namespace mang tên mình
- Deploy app v1.0 bằng file deployment.yaml đính kèm
 - `kubectl -n <ns> apply -f deployment.yaml`
- Kiểm tra log pod
 - `kubectl -n <ns> logs <pod-name>`
- Kiểm tra pod đã chạy chưa bằng port-forward tới pod

ConfigMap và Secret

- Chứa config của ứng dụng dạng key-value
- Key là string
- Value là 1 string (support multi-line) hoặc binary
- Dùng như biến môi trường hoặc mount file

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

ConfigMap

- Chiếc tủ kính ai cũng có thể thấy
- Full HD không che



Secret

- Chứa thông tin nhạy cảm
- Thông tin được lưu trữ encode base64
- "Full HD nhưng CÓ CHE"
- KHÔNG PHẢI MÃ HÓA





ConfigMap

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: demo-configmap
data:
  something: "Single value"
  somethingelse.json: |
    Multiline
    Version
    Config
```

Secret

```
apiVersion: v1
kind: Secret
type: Opaque
metadata:
  name: demo-secret
data:
  encoded_secret: YWJjCg==
stringData:
  something: "Single value without encode"
  somethingelse.json: |
    Multiline
    Version
```



Sử dụng configmap

```
containers:
  - name: web
    image: some-image
    imagePullPolicy: Always
    ports:
      - containerPort: 8080
    env:
      - name: SOME_ENV
        value: "hello"
      - name: CONFIG_ENV
        valueFrom:
          configMapKeyRef:
            name: demo-configmap
            key: something
```

```
containers:
  - name: web
    image: some-image
    imagePullPolicy: Always
    ports:
      - containerPort: 8080
    volumeMounts:
      - name: config-volume
        mountPath: /app/config.json
        subPath: somethingelse.json
    volumes:
      - name: config-volume
        configMap:
          name: demo-configmap
```



Thực hành 2: Mounting config

- <https://github.com/minhpg331/demo-deployment>
- Sử dụng configmap và secret như yêu cầu trong readme
- Sửa deployment.yaml và expose pod bằng port-forward
- Check 4 endpoint
 - /config-env
 - /config-file
 - /secret-env
 - /secret-file



1 số kinh nghiệm xương máu

- Nhớ set giá trị **revisionHistoryLimit** để xóa old template
- Lựa chọn **strategy** phù hợp (sẽ nói kỹ hơn trong bài rolling update)
- Nhớ set giá trị **imagePullPolicy**
- Tag docker image bằng git SHA để lưu version (ko dùng latest)
- Luôn hỏi dev về khả năng chạy nhiều pod trước khi deploy !!!
- Luôn dùng nháy kép với giá trị biến môi trường (số sẽ lỗi)



Tham khảo

- <https://kubernetes.io/docs/concepts/configuration/configmap/#using-configmaps>
- <https://kubernetes.io/docs/concepts/configuration/secret/#using-secrets>



Thanks for watching!