

Nội dung

3.3.1. Pod

3.3.2. Replicaset

3.3.3. Service

3.3.4. Deployment

Cách tạo pod trong k8s

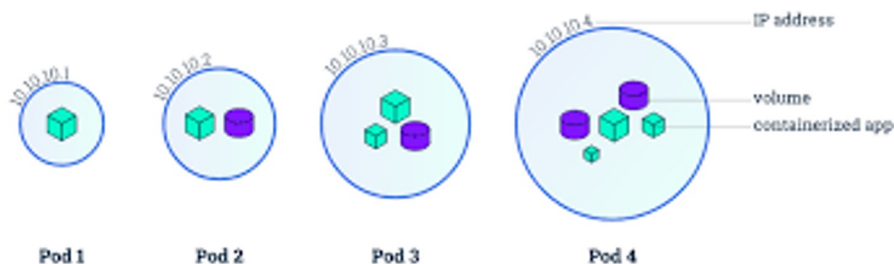
- Tạo 1 file yaml với nội dung như hình, có thể lấy luôn image vừa tạo phần docker
- Chạy lệnh

kubectl apply -f <filename>.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: <POD NAME>
spec:
  containers:
  - name: <container name>
    image: <image>
    ports:
    - containerPort: 80
```

Giới thiệu về k8s pod

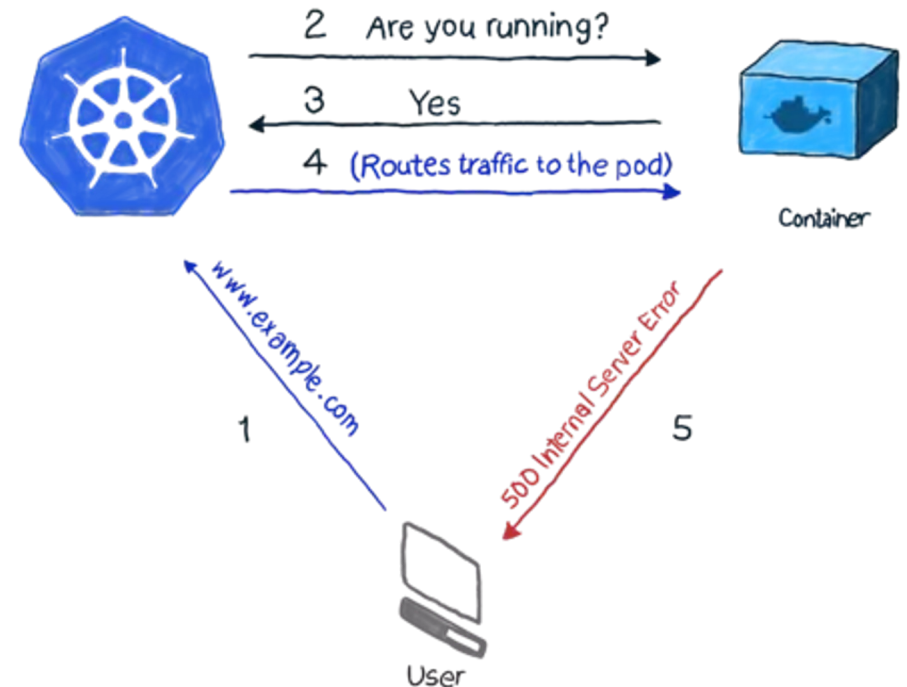
- Pod là thành phần cơ bản nhất để deploy và chạy một ứng dụng, được tạo và quản lý bởi kubernetes
- Pod được dùng để nhóm (group) và chạy một hoặc nhiều container lại với nhau trên cùng một worker node, những container trong một pod sẽ chia sẻ chung tài nguyên với nhau



K8s pod

Tại sao không chạy container trực tiếp mà phải qua pod ?

-> Pod giúp quản lý tài nguyên và trạng thái container tốt hơn



Cách tạo pod trong k8s

- Tạo 1 file yaml với nội dung như hình, có thể lấy luôn image vừa tạo phần docker
- Chạy lệnh

kubectl apply -f <filename>.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: <POD NAME>
spec:
  containers:
  - name: <container name>
    image: <image>
    ports:
    - containerPort: 80
```

Nội dung

3.3.1. Pod

3.3.2. Replicaset

3.3.3. Service

3.3.4. Deployment

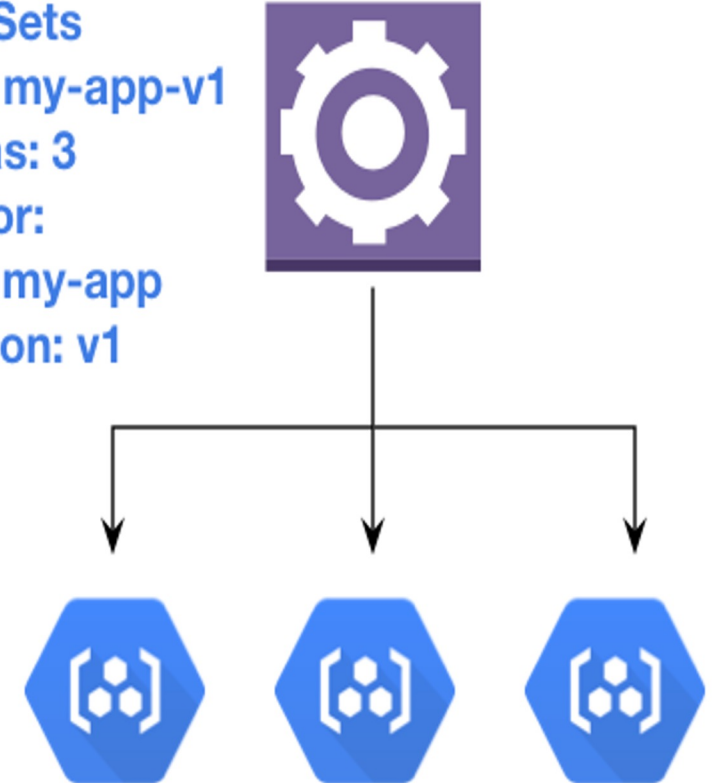
Replica set

ReplicationSet là một resource mà sẽ tạo và quản lý pod, và chắc chắn là số lượng pod nó quản lý không thay đổi và tiếp tục chạy.

ReplicationControllers sẽ tạo số lượng pod bằng với số ta chỉ định ở thuộc tính replicas và quản lý pod thông qua labels của pod

ReplicaSets

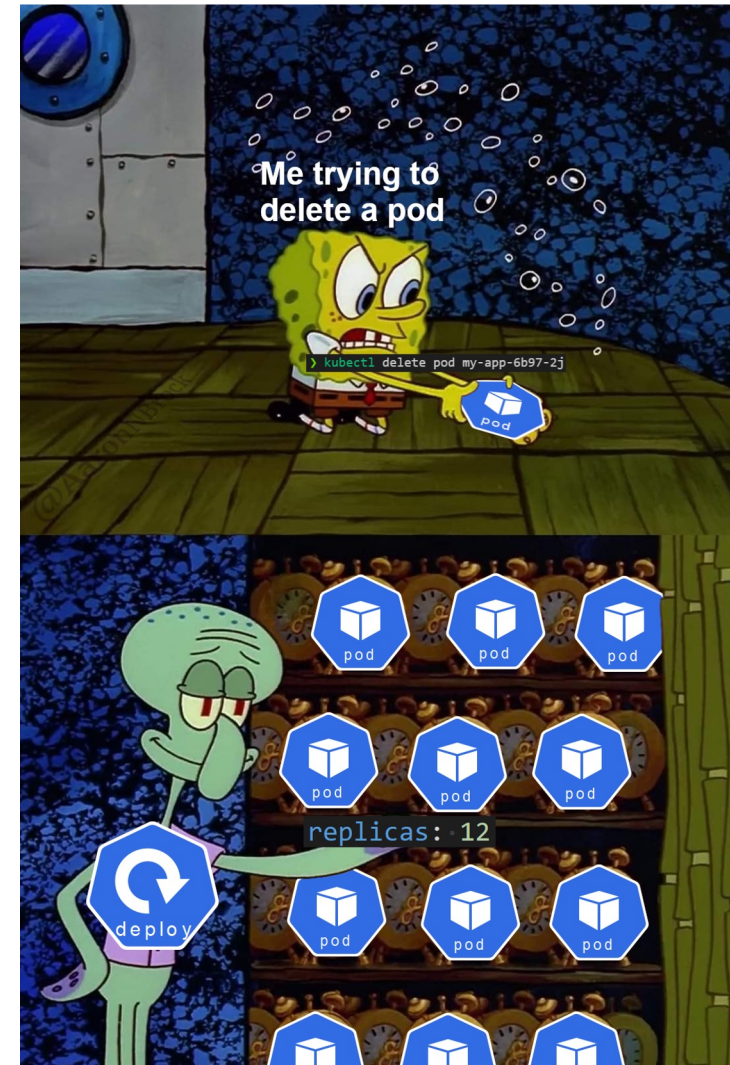
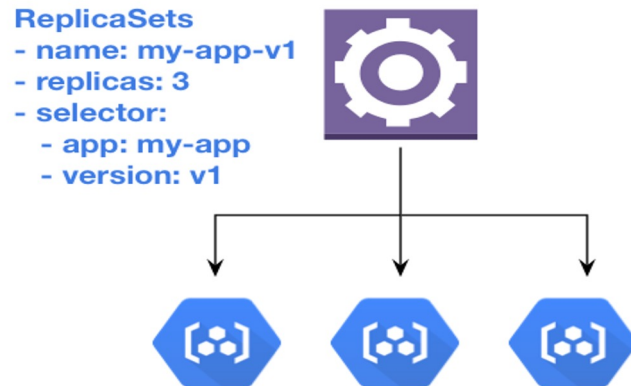
- name: my-app-v1
- replicas: 3
- selector:
 - app: my-app
 - version: v1



Replica set

ReplicaSet là một resource mà sẽ tạo và quản lý pod, và chắc chắn là số lượng pod nó quản lý không thay đổi và tiếp tục chạy.

ReplicationControllers sẽ tạo số lượng pod bằng với số ta chỉ định ở thuộc tính replicas và quản lý pod thông qua labels của pod

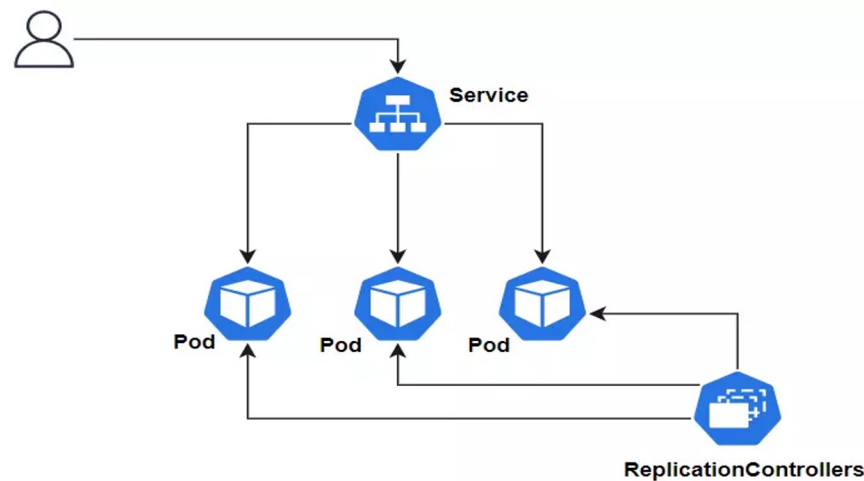


Tại sao dùng Replica set

- Đảm bảo tính khả dụng cho ứng dụng:

-> Vì RC sẽ chắc chắn rằng số lượng pod mà nó tạo ra không thay đổi, nên ví dụ khi ta tạo một thằng RC với số lượng replicas = 1, RC sẽ tạo 1 pod và giám sát nó, khi một thằng worker node die, nếu pod của thằng RC quản lý có nằm trong worker node đó, thì lúc này thằng RC sẽ phát hiện ra là số lượng pod của nó bằng 0

- Tăng tốc độ của ứng dụng:



Tạo replicaset

- Chạy lệnh

kubectl apply -f <filename>.yaml

- Kiểm tra:

kubectl get rs

- Xem mô tả:

kubectl describe <rs name>

```
apiVersion: apps/v1 # change version API
kind: ReplicaSet # change resource name
metadata:
  name: <replica name>
spec:
  replicas: 2
  selector:
    matchLabels: # change here
      app: <pod labels>
  template:
    metadata:
      labels:
        app: <pod labels>
    spec:
      containers:
        - image: <image name>
          name: <container name>
          ports:
            - containerPort: 3000
```

Nội dung

3.3.1. Pod

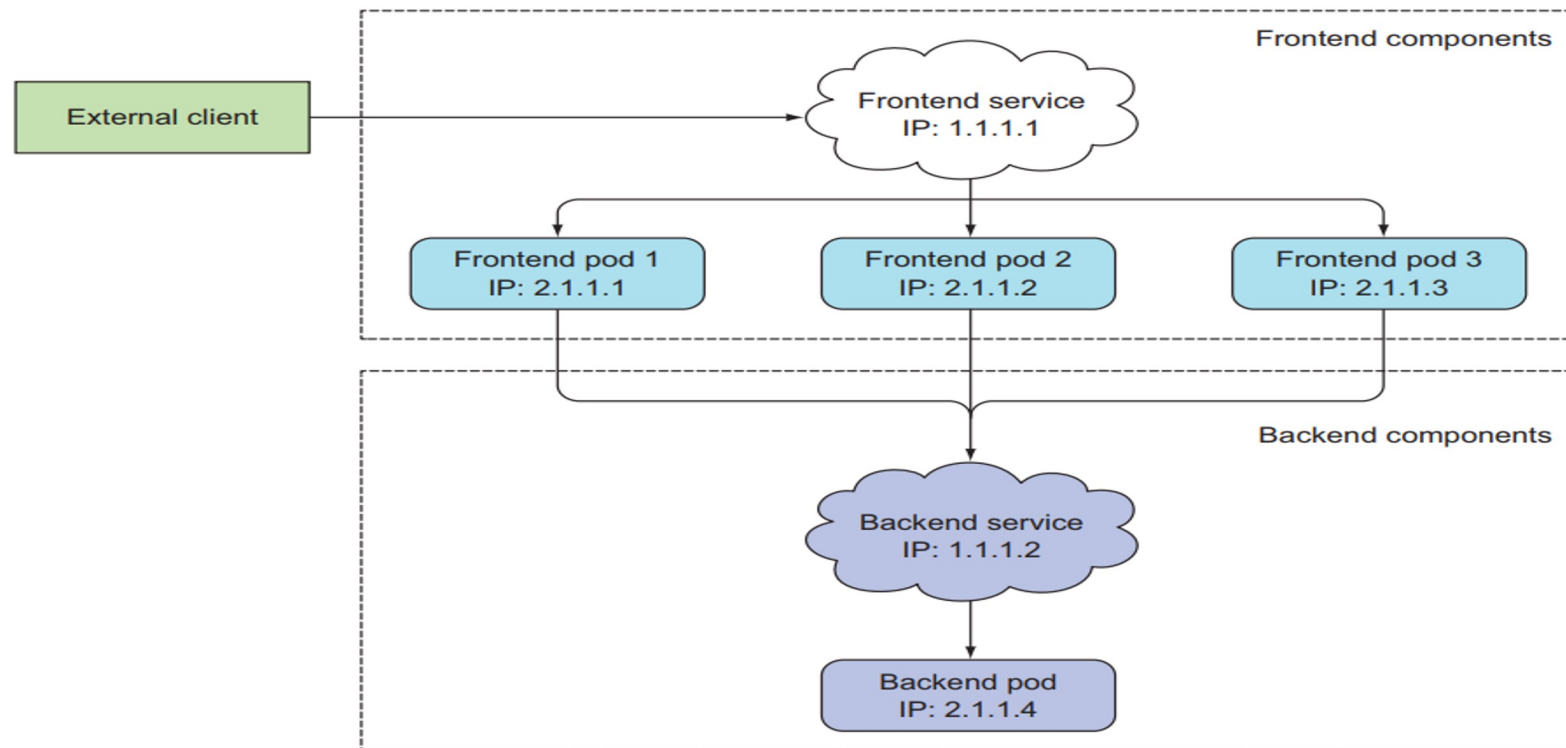
3.3.2. Replicaset

3.3.3. Service

3.3.4. Deployment

Service

Là một resource sẽ tạo ra một single, constant point của một nhóm Pod phía sau nó. Mỗi service sẽ có một địa chỉ IP và port không đổi, trừ khi ta xóa nó đi và tạo lại. Client sẽ mở connection tới service, và connection đó sẽ được dẫn tới một trong những Pod ở phía sau.



Service giải quyết vấn đề gì

- Mỗi pod có địa chỉ ip riêng tại sao không gọi luôn ip của pod
-> pod có thể bị thay đổi (kill, die,...)
- Ứng dụng chạy nhiều pod thì biết gọi vào pod nào ?
-> Service sẽ tạo ra một endpoint không đổi cho các Pod phía sau, client chỉ cần tương tác với endpoint này.

Các service

- ClusterIP: Đây là loại service sẽ tạo một IP và local DNS mà sẽ có thể truy cập ở bên trong cluster, không thể truy cập từ bên ngoài, được dùng chủ yếu cho các Pod ở bên trong cluster để dàng giao tiếp với nhau.
- Nodeport: Đây là cách đầu tiên để expose Pod cho client bên ngoài có thể truy cập vào được Pod bên trong cluster.
- LoadBalancer: Khi bạn chạy kubernetes trên cloud, nó sẽ hỗ trợ LoadBalancer Service, nếu bạn chạy trên môi trường không có hỗ trợ LoadBalancer thì bạn không thể tạo được loại Service này
- ExternalName

Nội dung

3.3.1. Pod

3.3.2. Replicaset

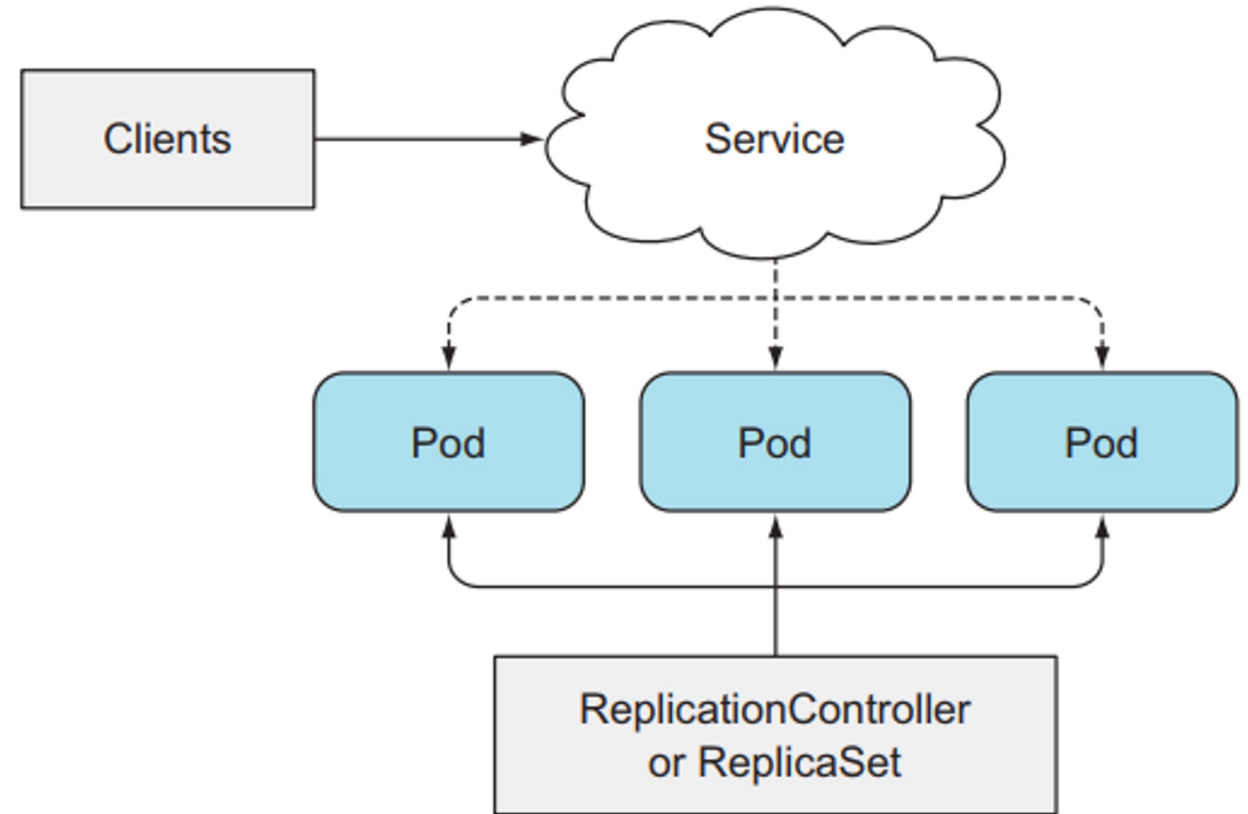
3.3.3. Service

3.3.4. Deployment

Deployment

Bây giờ các dev trong team đã viết xong tính năng mới, ta build lại image với code mới, và ta muốn update lại các Pod đang chạy này với image mới. Ta sẽ làm như thế nào?

- Rolling Update
- Recreate



Recreate

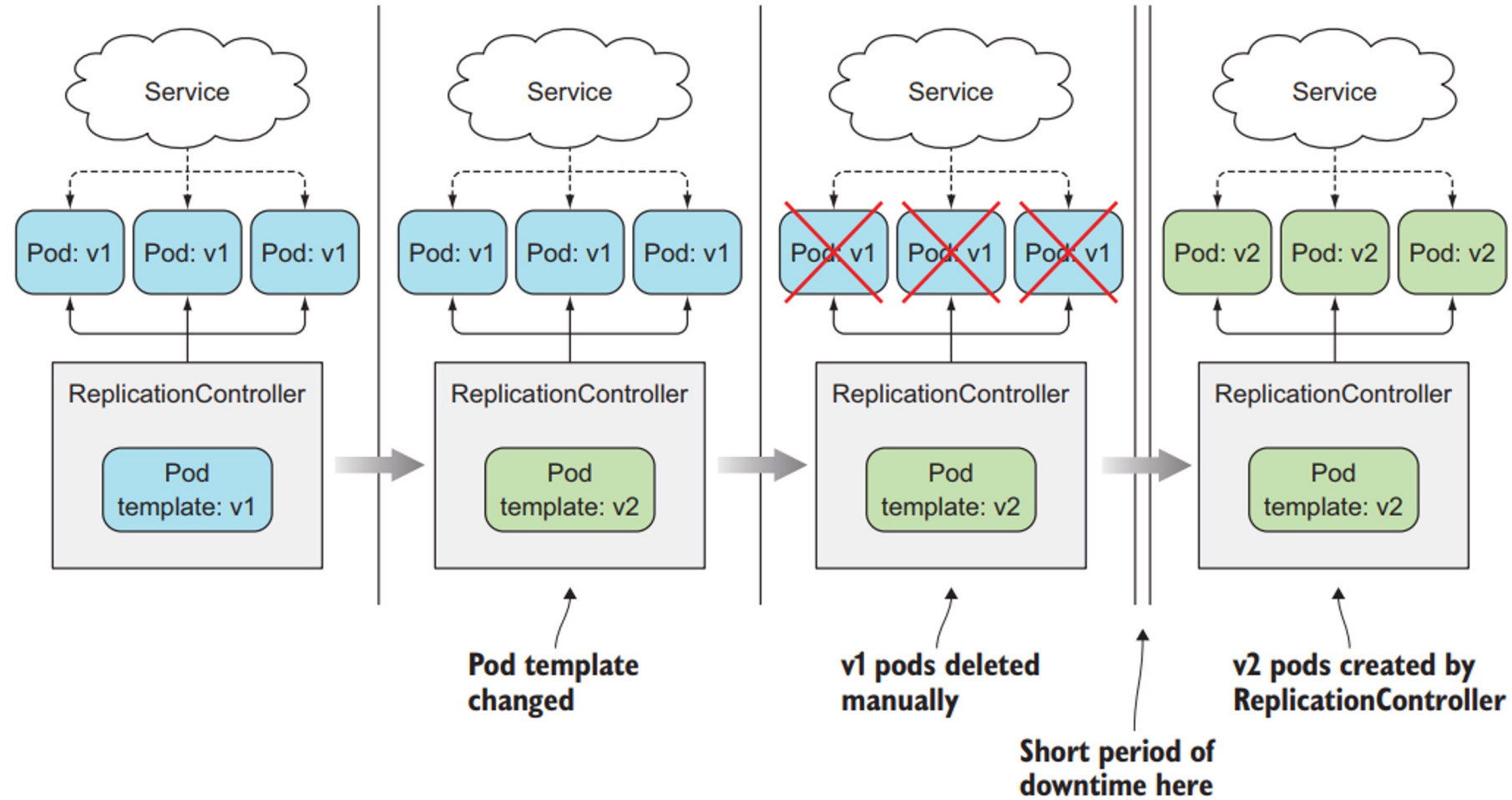


Figure 9.2 Updating pods by changing a ReplicationController's pod template and deleting old Pods

Rolling update

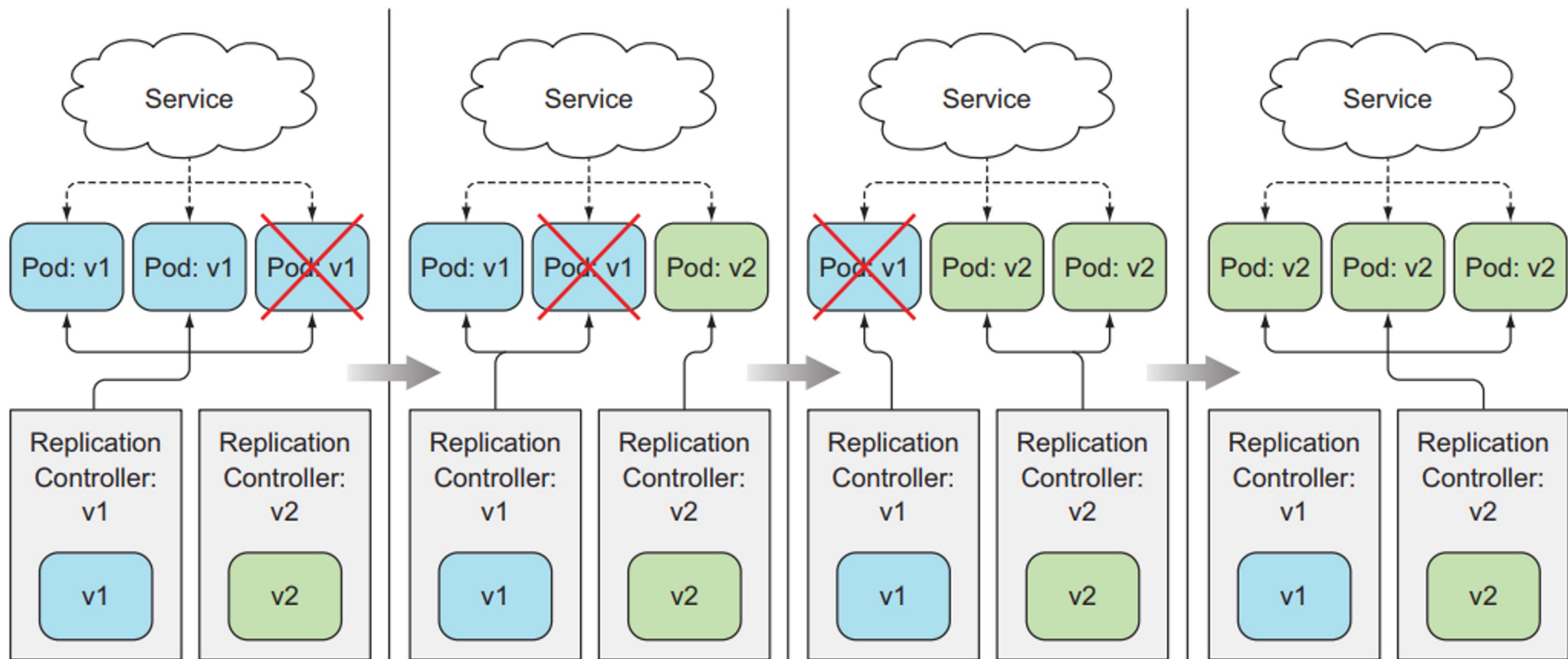


Figure 9.4 A rolling update of pods using two ReplicationControllers

Deployment giải quyết vấn đề gì

- Deployment là một resource của kubernetes giúp ta trong việc cập nhật một version mới của ứng dụng một cách dễ dàng, nó cung cấp sẵn 2 strategy để deploy là Recreate và RollingUpdate, tất cả đều được thực hiện tự động bên dưới, và các version được deploy sẽ có một history ở đằng sau, ta có thể rollback and rollout giữa các phiên bản bất cứ lúc nào mà không cần chạy lại CI/CD.

