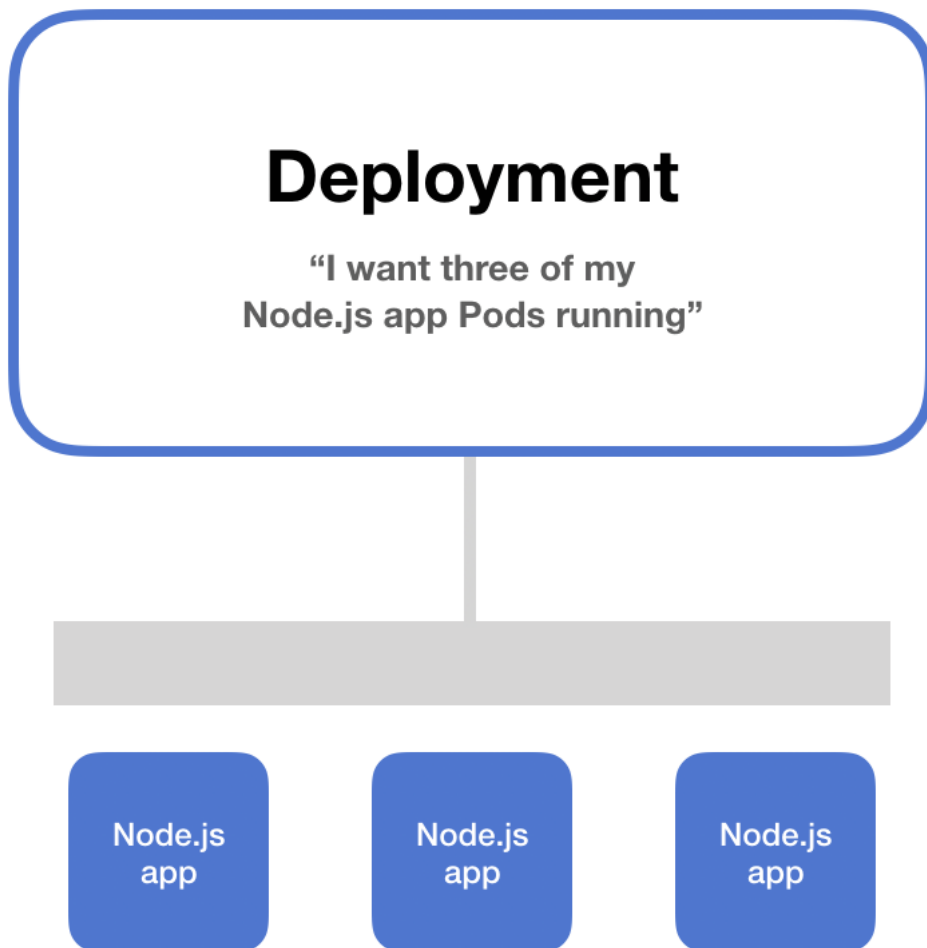


Kubernetes Deployment - Triển khai ứng dụng bằng Deployment

Deployment là một supervisor (kịch bản thực hiện) cho các Pod, cung cấp cho bạn quyền kiểm soát chi tiết về cách thức và thời điểm một phiên bản Pod mới được triển khai cũng như quay trở lại trạng thái trước đó.



1. Tạo Kubernetes Deployment

- File: `nginx-deployment.yaml`

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 80
```

- Giải thích: Trong ví dụ trên:

- Deployment được tạo với tên `nginx-deployment` đã được tạo, được chỉ định bởi trường `metadata.name`
- Deployment được tạo với 3 replicated Pod, được chỉ định bởi trường `spec.replicas`
- Trường `spec.selector` định nghĩa cách Deployment tìm thấy Pod cần quản lý. Trong trường hợp này, chúng ta sử dụng label của Pod (`app: nginx`). Tuy nhiên, có thể thực hiện các quy tắc lựa chọn phức tạp hơn, miễn là bản thân mẫu Pod thỏa mãn quy tắc.
- Trường `template` chứa các thông tin sau:
 - `template.metadata`: Label (nhãn) của Pod `app: nginx` được chỉ định bởi trường `metadata.labels`
 - `template.spec`: Đặc điểm kỹ thuật của Deployment
 - Trong ví dụ này, Pod có một container, chạy nginx image version 1.14.2 (từ docker hub)
 - `name`: Tên container
 - `image`: Docker image
 - `ports`: Các port của container tương ứng với port của image sau khi chạy

- Thực hiện:

```
kubectl apply -f nginx-deployment.yaml
```

- Kiểm tra:

```
kubectl get deployments
```

NAME	READY	UP- TO- DATE	AVAILABLE	AGE
nginx-deployment	0/3	0	0	1s

Khi bạn kiểm tra Deployment, các trường sau được hiển thị:

- NAME: Liệt kê tên của các Triển khai trong không gian tên.
- READY: Hiển thị số lượng bản sao của ứng dụng có sẵn cho người dùng của bạn. Nó theo mẫu đã sẵn sàng / mong muốn.
- UP-TO-DATE: Hiển thị số lượng bản sao đã được cập nhật để đạt được trạng thái mong muốn.
- AVAILABLE: Hiển thị số lượng bản sao của ứng dụng có sẵn cho người dùng của bạn.
- AGE: Hiển thị lượng thời gian ứng dụng đã chạy.

- Để kiểm tra trạng thái thực hiện của Deployment, thực hiện `kubectl rollout status deployment/nginx-deployment`

```
Waiting for rollout to finish: 2 out of 3 new replicas have been updated...  
deployment "nginx-deployment" successfully rolled out
```

- Kiểm tra số lượng ReplicaSet được tạo bởi Deployment, thực hiện:

```
kubectl get rs
```

NAME	DESIRED	CURRENT	READY	AGE
nginx-deployment-75675f5897	3	3	3	18s

ReplicaSet hiển thị các trường sau:

- NAME: Liệt kê tên của ReplicaSets trong không gian tên.
- DESIRED: Hiển thị số lượng bản sao mong muốn của ứng dụng mà bạn xác định khi tạo Triển khai. Đây là trạng thái mong muốn.
- CURRENT: Hiển thị số lượng bản sao hiện đang chạy.
- READY: Hiển thị số lượng bản sao của ứng dụng có sẵn cho người dùng của bạn.
- AGE: Hiển thị lượng thời gian ứng dụng đã chạy.

- Để kiểm tra Pod được tạo, thực hiện:

```
kubectl get pods --show-labels
```

NAME	READY	STATUS	RESTARTS	AGE	LABELS
nginx-deployment-75675f5897-7ci7o	1/1	Running	0	18s	app=nginx, pod-template-hash=3123191453
nginx-deployment-75675f5897-kzszj	1/1	Running	0	18s	app=nginx, pod-template-hash=3123191453
nginx-deployment-75675f5897-qqcnn	1/1	Running	0	18s	app=nginx, pod-template-hash=3123191453

2. Cập nhật Deployment

- Thực hiện cập nhật nginx Pod sử dụng image version `nginx: 1.16.1` từ version `nginx: 1.14.2`

```
kubectl set image deployment.v1.apps/nginx-deployment nginx=nginx:1.16.1
```

hoặc sử dụng:

```
kubectl set image deployment/nginx-deployment nginx=nginx:1.16.1
```

Kết quả:

```
deployment.apps/nginx-deployment image updated
```

Tuy nhiên, chúng ta có thể edit Deployment và thay đổi `.spec.template.spec.containers[0].image` từ `nginx: 1.14.2` thành `nginx: 1.16.1`:

```
kubectl edit deployment/nginx-deployment
```

Kết quả:

```
deployment.apps/nginx-deployment edited
```

- Kiểm tra trạng thái Deployment rollout, ReplicaSet, Pod theo các command ở trên để kiểm tra chi tiết.

3. Kiểm tra chi tiết Deployment

- Kiểm tra chi tiết Deployment, thực hiện:

```
kubectl describe deployments
```

- Kết quả:

Name: nginx-deployment
Namespace: default
CreationTimestamp: Thu, 30 Nov 2017 10:56:25 +0000
Labels: app=nginx
Annotations: deployment.kubernetes.io/revision=2
Selector: app=nginx
Replicas: 3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge

Pod Template:

Labels: app=nginx

Containers:

nginx:

Image: nginx:1.16.1

Port: 80/TCP

Environment: <none>

Mounts: <none>

Volumes: <none>

Conditions:

Type	Status	Reason
Available	True	MinimumReplicasAvailable
Progressing	True	NewReplicaSetAvailable

OldReplicaSets: <none>

NewReplicaSet: nginx-deployment-1564180365 (3/3 replicas created)

Events:

Type	Reason	Age	From	Message
Normal	ScalingReplicaSet	2m	deployment-controller	Scaled up replica set nginx-deployment-2035384211 to 3
Normal	ScalingReplicaSet	24s	deployment-controller	Scaled up replica set nginx-deployment-1564180365 to 1
Normal	ScalingReplicaSet	22s	deployment-controller	Scaled down replica set nginx-deployment-2035384211 to 2
Normal	ScalingReplicaSet	22s	deployment-controller	Scaled up replica set nginx-deployment-1564180365 to 2
Normal	ScalingReplicaSet	19s	deployment-controller	Scaled down replica set nginx-deployment-2035384211 to 1
Normal	ScalingReplicaSet	19s	deployment-controller	Scaled up replica set nginx-

```
deployment-1564180365 to 3
```

```
Normal ScalingReplicaSet 14s deployment-controller Scaled down replica set nginx-
```

```
deployment-2035384211 to 0
```

4. Deployment Rollover

Mỗi khi Deployment controller quan sát thấy một kịch bản triển khai Deployment mới (cập nhật), một ReplicaSet mới được tạo để quản lý các Pod. Nếu Deployment được update, ReplicaSet đang tồn tại có label trùng `.spec.selector` nhưng không trùng `template.spec.selector` sẽ bị scaled down. Cuối cùng, một ReplicaSet mới sẽ được scaled tới `.spec.replicas` và tất cả các ReplicaSet cũ sẽ bị scaled về 0

Nếu chúng ta cập nhật Deployment trong khi Deployment rollout hiện tại đang trong quá trình thực thi, Deployment sẽ tạo một ReplicaSet mới theo bản cập nhật và bắt đầu scaling up, và ReplicaSet hiện tại sẽ được thêm vào danh sách ReplicaSet cũ để thực hiện scaling down

Ví dụ: Trong trường hợp chúng ta tạo Deployment để tạo 5 replicas của `nginx: 1.14.2`, nhưng sau đó thực hiện cập nhật Deployment để tạo 5 replicas của `nginx: 1.16.1`, khi mới chỉ 3 replicas của `nginx: 1.14.2` đã được tạo. Trong trường hợp này, Deployment sẽ ngay lập tức killing 3 Pod `nginx: 1.14.2` (mà không chờ đến lúc tạo đủ 5 pod `nginx: 1.14.2`) đã được tạo này, sau đó thực hiện tạo pod `nginx: 1.16.1` theo kịch bản Deployment mới.

5. Rolling Back a Deployment

continue..

Revision #6

Created 5 March 2022 03:06:04 by Laptrinh.vn

Updated 6 May 2022 17:04:18 by Laptrinh.vn