

Kubernetes-Rolling Updates

Rolling Updates in Kubernetes Deployments

A rolling update is a strategy used to deploy new versions of an application in a Kubernetes cluster while minimizing downtime and ensuring high availability. It involves creating multiple replicas of the old version, updating one replica at a time, and then scaling down the old version once the update is complete.

Example Use Case:

Suppose we have a web server deployment that serves a website (e.g., `example.com`). We want to deploy a new version of the application with some bug fixes and minor UI changes. Our current deployment has 3 replicas running on nodes A, B, and C.

Step-by-Step Rolling Update:

- 1. Create a new deployment:** We create a new deployment (`v2`) for the updated version of our web server application.
- 2. Update rollout strategy:** We update the rollout strategy for the `example.com` deployment to use a rolling update with 3 replicas. This will ensure that we have multiple replicas available during the update process.

Here's an example YAML configuration:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: example-com
spec:
  selector:
    matchLabels:
      app: example-com
  template:
    metadata:
      labels:
        app: example-com
    spec:
      containers:
        - name: web-server
          image: example-com:v2
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 0
```

In this configuration, we're allowing up to 1 additional replica (i.e., `maxSurge`) during the update process, and we want all replicas to be updated at once (`maxUnavailable` is set to 0).

3. Scale down old deployment: Once the rollout is complete and all replicas are running with the new version, we can scale down the original deployment.

Benefits:

- Minimized downtime
- High availability ensured
- Easy rollback in case of issues

By following this rolling update strategy, you can ensure a smooth deployment experience for your Kubernetes applications.

Curated by Brajesh Kumar