

## Upgrade Strategy

### Introduction:

In Kubernetes, we can simply upgrade our deployment using the set image command or editing the Yaml file. Kubernetes supports **RollingUpdate** and **Recreate** method upgrading methods.

### Objectives:

1. Upgrade the deployment using RollingUpdate method
2. Upgrade the deployment using Recreate method

#### 1. Upgrade the deployment using RollingUpdate method:

RollingUpdate is the default upgrade behaviour in a deployment. Let's see how we can upgrade our deployment.

#### Step1: Create a deployment

Use the below command to create a deployment.

```
kubectl create deploy prod --image nginx --replicas=3
```

The above command will create a Deployment , one Replicaset and 3 pods with nginx image.

```
root@master:~#  
root@master:~# kubectl get deploy  
NAME      READY   UP-TO-DATE   AVAILABLE   AGE  
prod      3/3     3            3           22s  
root@master:~#  
root@master:~# kubectl get rs  
NAME                DESIRED   CURRENT   READY   AGE  
prod-5c57d87787      3         3         3       28s  
root@master:~#  
root@master:~# kubectl get pods  
NAME                                READY   STATUS    RESTARTS   AGE  
prod-5c57d87787-ckq6c              1/1     Running   0          34s  
prod-5c57d87787-fqgmj              1/1     Running   0          34s  
prod-5c57d87787-x9fk5              1/1     Running   0          34s  
root@master:~#
```

As you can see above that we have only one replica set which is **prod-5c57d87787** which is responsible for pods with nginx image.

```
root@master:~# kubectl describe deploy prod
Name:                prod
Namespace:           default
CreationTimestamp:   Fri, 13 Jan 2023 09:37:27 +0000
Labels:              app=prod
Annotations:         deployment.kubernetes.io/revision: 1
Selector:            app=prod
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=prod
  Containers:
    nginx:
      Image:        nginx
      Port:         <none>
      Host Port:    <none>
      Environment:  <none>
      Mounts:       <none>
      Volumes:      <none>
  Conditions:
    Type           Status  Reason
    ----           -
    Available      True    MinimumReplicasAvailable
    Progressing    True    NewReplicaSetAvailable
  OldReplicaSets:  <none>
  NewReplicaSet:   prod-5c57d87787 (3/3 replicas created)
Events:
  Type           Reason             Age   From                  Message
  ----           -
  Normal        ScalingReplicaSet   4m37s  deployment-controller  Scaled up replica set prod-5c57d87787 to 3
```

In above output, we can see that the **StrategyType** is **RollingUpdate** and **RollingUpdateStrategy** is **25% max unavailable, 25% max surge**.

Under the events section, we can see the replica set is having 3 pods.

## **Step2: Upgrade the deployment**

Now its time to upgrade the application to a specific version. Use the below command to upgrade our deployment.

```
kubectl set image deploy prod nginx=nginx:1.23.3
```

This will set this image for us and pods will be terminated and created again but under a new replica set. The upgrade will not delete all the pods in one go in fact it will use the **RollingUpdateStrategy** to delete and create the pods.

Let's see the output below:

```

root@master:~# kubectl get pods
NAME                                READY    STATUS             RESTARTS   AGE
prod-5c57d87787-fqgmg              0/1     Terminating       0           14m
prod-79cf67cb5d-8vw6f              1/1     Running             0           15s
prod-79cf67cb5d-dzvrl              1/1     Running             0           11s
prod-79cf67cb5d-l4p2x              1/1     Running             0           13s
root@master:~#
root@master:~# kubectl get rs
NAME                                DESIRED   CURRENT   READY   AGE
prod-5c57d87787                    0         0         0       14m
prod-79cf67cb5d                    3         3         3       33s

```

```

root@master:~# kubectl describe deploy prod
Name:                               prod
Namespace:                           default
CreationTimestamp:                   Fri, 13 Jan 2023 09:37:27 +0000
Labels:                               app=prod
Annotations:                         deployment.kubernetes.io/revision: 2
Selector:                             app=prod
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=prod
  Containers:
    nginx:
      Image:                           nginx:1.23.3
      Port:                             <none>
      Host Port:                       <none>
      Environment:                     <none>
      Mounts:                           <none>
      Volumes:                         <none>
  Conditions:
    Type                               Status  Reason
    ---                               -
    Available                          True    MinimumReplicasAvailable
    Progressing                        True    NewReplicaSetAvailable
  OldReplicaSets:                     <none>
  NewReplicaSet:                      prod-79cf67cb5d (3/3 replicas created)
Events:
  Type                               Reason                               Age    From                               Message
  ---                               -
  Normal                             ScalingReplicaSet                   14m    deployment-controller             Scaled up replica set prod-5c57d87787 to 3
  Normal                             ScalingReplicaSet                   46s    deployment-controller             Scaled up replica set prod-79cf67cb5d to 1
  Normal                             ScalingReplicaSet                   44s    deployment-controller             Scaled down replica set prod-5c57d87787 to 2
  Normal                             ScalingReplicaSet                   44s    deployment-controller             Scaled up replica set prod-79cf67cb5d to 2
  Normal                             ScalingReplicaSet                   42s    deployment-controller             Scaled down replica set prod-5c57d87787 to 1
  Normal                             ScalingReplicaSet                   42s    deployment-controller             Scaled up replica set prod-79cf67cb5d to 3
  Normal                             ScalingReplicaSet                   40s    deployment-controller             Scaled down replica set prod-5c57d87787 to 0

```

In above output, we can see under the Events field, a new replica set(prod-79cf67cb5d) is created with a new pod followed by a pod will be terminated in old replica set(prod-5c57d87787).

```

Normal ScalingReplicaSet 46s deployment-controller Scaled up replica set prod-79cf67cb5d to 1
Normal ScalingReplicaSet 44s deployment-controller Scaled down replica set prod-5c57d87787 to 2
Normal ScalingReplicaSet 44s deployment-controller Scaled up replica set prod-79cf67cb5d to 2
Normal ScalingReplicaSet 42s deployment-controller Scaled down replica set prod-5c57d87787 to 1
Normal ScalingReplicaSet 42s deployment-controller Scaled up replica set prod-79cf67cb5d to 3
Normal ScalingReplicaSet 40s deployment-controller Scaled down replica set prod-5c57d87787 to 0

```

We can also check the status of rollout using below command.

```
kubectl rollout status deploy prod
```

```
root@master:~#  
root@master:~# kubectl rollout status deploy prod  
deployment "prod" successfully rolled out  
root@master:~#
```

### Step3: Rollback to older version

If we think that the latest version is not the stable one, we can rollback to the older version using the below command.

```
kubectl rollout undo deploy prod
```

The above command will rollback to earlier version and new pods will be created with old version whereas the pods with latest image will be terminated.

```
root@master:~# kubectl rollout undo deploy prod  
deployment.apps/prod rolled back  
root@master:~#  
root@master:~#  
root@master:~# kubectl get pods  
NAME                                READY   STATUS             RESTARTS   AGE  
prod-5c57d87787-lfps8               1/1     Running            0           4s  
prod-5c57d87787-rjlbz               0/1     ContainerCreating  0           2s  
prod-79cf67cb5d-8vw6f               1/1     Running            0           20m  
prod-79cf67cb5d-dzvrl               0/1     Terminating      0           20m  
prod-79cf67cb5d-l4p2x               1/1     Running            0           20m  
root@master:~#  
root@master:~#  
root@master:~# kubectl get rs  
NAME                                DESIRED   CURRENT   READY   AGE  
prod-5c57d87787                     3         3         3       34m  
prod-79cf67cb5d                     0         0         0       20m  
root@master:~#
```

Let's check the events using the describe command.

```
kubectl describe deploy prod
```

In the below output, we can see that the pods rollback has happened and all the pods are now a part of old replica set which is **prod-5c57d87787**.

```

Name: prod
Namespace: default
CreationTimestamp: Fri, 13 Jan 2023 09:37:27 +0000
Labels: app=prod
Annotations: deployment.kubernetes.io/revision: 3
Selector: app=prod
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=prod
  Containers:
    nginx:
      Image: nginx
      Port: <none>
      Host Port: <none>
      Environment: <none>
      Mounts: <none>
      Volumes: <none>
Conditions:
  Type           Status  Reason
  ----           -
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets: <none>
NewReplicaSet: prod-5c57d87787 (3/3 replicas created)
Events:
  Type           Reason             Age           From              Message
  ----           -
  Normal         ScalingReplicaSet   20m           deployment-controller Scaled up replica set prod-79cf67cb5d to 1
  Normal         ScalingReplicaSet   20m           deployment-controller Scaled down replica set prod-5c57d87787 to 2
  Normal         ScalingReplicaSet   20m           deployment-controller Scaled up replica set prod-79cf67cb5d to 2
  Normal         ScalingReplicaSet   20m           deployment-controller Scaled down replica set prod-5c57d87787 to 1
  Normal         ScalingReplicaSet   20m           deployment-controller Scaled up replica set prod-79cf67cb5d to 3
  Normal         ScalingReplicaSet   20m           deployment-controller Scaled down replica set prod-5c57d87787 to 0
  Normal         ScalingReplicaSet   20s           deployment-controller Scaled up replica set prod-5c57d87787 to 1
  Normal         ScalingReplicaSet   18s           deployment-controller Scaled down replica set prod-79cf67cb5d to 2
  Normal         ScalingReplicaSet   18s           deployment-controller Scaled up replica set prod-5c57d87787 to 2
  Normal         ScalingReplicaSet   15s (x2 over 35m) deployment-controller Scaled up replica set prod-5c57d87787 to 3
  Normal         ScalingReplicaSet   15s           deployment-controller Scaled down replica set prod-79cf67cb5d to 1
  Normal         ScalingReplicaSet   13s           deployment-controller Scaled down replica set prod-79cf67cb5d to 0

```

## 2. Upgrade the deployment using Recreate method:

In Recreate Strategy, all the pods are terminated first then new pods will be created. So, we will witness a downtime until the new pods are created. This method is not recommended and we have to explicitly ask the deployment to follow this strategy.

### Step1: Create a deployment:

Use the below Yaml file to create a deployment with Recreate method.

Create a Yaml file.

```
vi dev.yaml
```

Copy and paste the below code into the file and apply the file.

Here we are having deployment named **devops** with 3 replicas and image **nginx**.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: devops
  name: devops
spec:
  replicas: 3
  selector:
    matchLabels:
      app: devops
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: devops
    spec:
      containers:
        - image: nginx
          name: nginx
```

Now apply the above code using below command.

```
kubectl apply -f dev.yaml
```

```

root@master:~# kubectl apply -f dev.yaml
deployment.apps/devops created
root@master:~#
root@master:~# kubectl get deploy
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
devops    3/3     3            3           23s
root@master:~#
root@master:~#
root@master:~# kubectl get rs
NAME                               DESIRED   CURRENT   READY   AGE
devops-76bd54c875                 3         3         3       28s
root@master:~#
root@master:~#
root@master:~# kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
devops-76bd54c875-8z5b7           1/1     Running   0          36s
devops-76bd54c875-cq8p2           1/1     Running   0          36s
devops-76bd54c875-nzb7p           1/1     Running   0          36s
root@master:~#

```

```

root@master:~#
root@master:~# kubectl describe deploy devops
Name:          devops
Namespace:     default
CreationTimestamp: Fri, 13 Jan 2023 10:59:15 +0000
Labels:        app=devops
Annotations:    deployment.kubernetes.io/revision: 1
Selector:      app=devops
Replicas:      3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:  Recreate
MinReadySeconds: 0
Pod Template:
  Labels:  app=devops
  Containers:
    nginx:
      Image:      nginx
      Port:       <none>
      Host Port:  <none>
      Environment: <none>
      Mounts:      <none>
      Volumes:      <none>
  Conditions:
    Type           Status  Reason
    ----           -
    Available       True    MinimumReplicasAvailable
    Progressing     True    NewReplicaSetAvailable
OldReplicaSets: <none>
NewReplicaSet:  devops-76bd54c875 (3/3 replicas created)
Events:
  Type           Reason             Age   From               Message
  ----           -
  Normal         ScalingReplicaSet   3m9s  deployment-controller  Scaled up replica set devops-76bd54c875 to 3

```

In above output, we have described the deployment and we can see the strategy type is **Recreate**.

## Step2: Upgrade the deployment

Now let's update the deployment to the latest image using the **set image** command.

```
kubectl set image deploy devops nginx=nginx:1.23.3
```

Above command will create a new replica set (**devops-6f54478cf8**) and old pods will be terminated. Check the below output.

```
root@master:~#
root@master:~# kubectl get pods
NAME                                READY   STATUS            RESTARTS   AGE
devops-76bd54c875-8z5b7             0/1     Terminating      0           5m7s
devops-76bd54c875-nzb7p             0/1     Terminating      0           5m7s
root@master:~#
root@master:~#
root@master:~# kubectl get pods
NAME                                READY   STATUS            RESTARTS   AGE
devops-76bd54c875-8z5b7             0/1     Terminating      0           5m11s
devops-76bd54c875-nzb7p             0/1     Terminating      0           5m11s
root@master:~#
root@master:~#
root@master:~# kubectl get pods
NAME                                READY   STATUS            RESTARTS   AGE
devops-6f54478cf8-5mkqb             0/1     ContainerCreating  0           1s
devops-6f54478cf8-mv25g             0/1     ContainerCreating  0           1s
devops-6f54478cf8-wkhnb             0/1     ContainerCreating  0           1s
root@master:~#
root@master:~#
root@master:~# kubectl get pods
NAME                                READY   STATUS            RESTARTS   AGE
devops-6f54478cf8-5mkqb             1/1     Running            0           4s
devops-6f54478cf8-mv25g             1/1     Running            0           4s
devops-6f54478cf8-wkhnb             1/1     Running            0           4s
root@master:~#
root@master:~# kubectl get rs
NAME                                DESIRED   CURRENT   READY   AGE
devops-6f54478cf8                   3         3         3       17s
devops-76bd54c875                   0         0         0       5m31s
```

Let's see the events using the describe command.

Under events field, we can clearly see that all the pods gets terminated first then new pods are created under new replica set.

```
Normal ScalingReplicaSet 5m42s deployment-controller Scaled up replica set devops-76bd54c875 to 3
Normal ScalingReplicaSet 41s deployment-controller Scaled down replica set devops-76bd54c875 to 0
Normal ScalingReplicaSet 28s deployment-controller Scaled up replica set devops-6f54478cf8 to 3
```



```

root@master:~# kubectl describe deploy devops
Name: devops
Namespace: default
CreationTimestamp: Fri, 13 Jan 2023 10:59:15 +0000
Labels: app=devops
Annotations: deployment.kubernetes.io/revision: 2
Selector: app=devops
Replicas: 3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType: Recreate
MinReadySeconds: 0
Pod Template:
  Labels: app=devops
  Containers:
    nginx:
      Image: nginx:1.23.3
      Port: <none>
      Host Port: <none>
      Environment: <none>
      Mounts: <none>
      Volumes: <none>
  Conditions:
    Type           Status    Reason
    ----           -
    Available       True      MinimumReplicasAvailable
    Progressing     True      NewReplicaSetAvailable
OldReplicaSets: <none>
NewReplicaSet: devops-6f54478cf8 (3/3 replicas created)
Events:
  Type           Reason             Age   From                  Message
  ----           -
  Normal         ScalingReplicaSet   5m42s deployment-controller Scaled up replica set devops-76bd54c875 to 3
  Normal         ScalingReplicaSet   41s   deployment-controller Scaled down replica set devops-76bd54c875 to 0
  Normal         ScalingReplicaSet   28s   deployment-controller Scaled up replica set devops-6f54478cf8 to 3

```

### Step3: Rollback to earlier version:

If we feel the latest version is not stable then we can simply go back to earlier version using **rollout undo** command.

Use the below command to get back to older version.

```
kubectl rollout undo deploy devops
```

Now the deployment goes back to the older version.

```

root@master:~# kubectl rollout undo deploy devops
deployment.apps/devops rolled back
root@master:~#
root@master:~# kubectl get rs
NAME                                DESIRED    CURRENT    READY    AGE
devops-6f54478cf8                   0          0          0        9m7s
devops-76bd54c875                   3          3          3        14m
root@master:~#
root@master:~#
root@master:~#
root@master:~# kubectl get pods
NAME                                READY      STATUS    RESTARTS   AGE
devops-76bd54c875-bmxbbs            1/1       Running   0           41s
devops-76bd54c875-fbbxq             1/1       Running   0           41s
devops-76bd54c875-srdpb             1/1       Running   0           41s
root@master:~#

```

```
root@master:~# kubectl describe deploy devops
Name: devops
Namespace: default
CreationTimestamp: Fri, 13 Jan 2023 10:59:15 +0000
Labels: app=devops
Annotations: deployment.kubernetes.io/revision: 3
Selector: app=devops
Replicas: 3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType: Recreate
MinReadySeconds: 0
Pod Template:
  Labels: app=devops
  Containers:
    nginx:
      Image: nginx
      Port: <none>
      Host Port: <none>
      Environment: <none>
      Mounts: <none>
      Volumes: <none>
  Conditions:
    Type          Status    Reason
    ----          -
    Progressing   True      NewReplicaSetAvailable
    Available     True      MinimumReplicasAvailable
OldReplicaSets: <none>
NewReplicaSet: devops-76bd54c875 (3/3 replicas created)
Events:
  Type          Reason              Age             From              Message
  ----          -
  Normal        ScalingReplicaSet    9m44s           deployment-controller Scaled down replica set devops-76bd54c875 to 0
  Normal        ScalingReplicaSet    9m31s           deployment-controller Scaled up replica set devops-6f54478cf8 to 3
  Normal        ScalingReplicaSet    73s             deployment-controller Scaled down replica set devops-6f54478cf8 to 0
  Normal        ScalingReplicaSet    60s (x2 over 14m) deployment-controller Scaled up replica set devops-76bd54c875 to 3
root@master:~#
```

From above output, we can clearly see the new replica set becomes zero and old replica set becomes 3.