

# Lab Exercise 11- Deployments with Rolling Update and Recreate Strategies

Name – Ayush Bhardwaj  
500124917  
B2 DevOps

Understand how to use the rolling update and recreate strategies for deploying applications using Kubernetes Deployments.

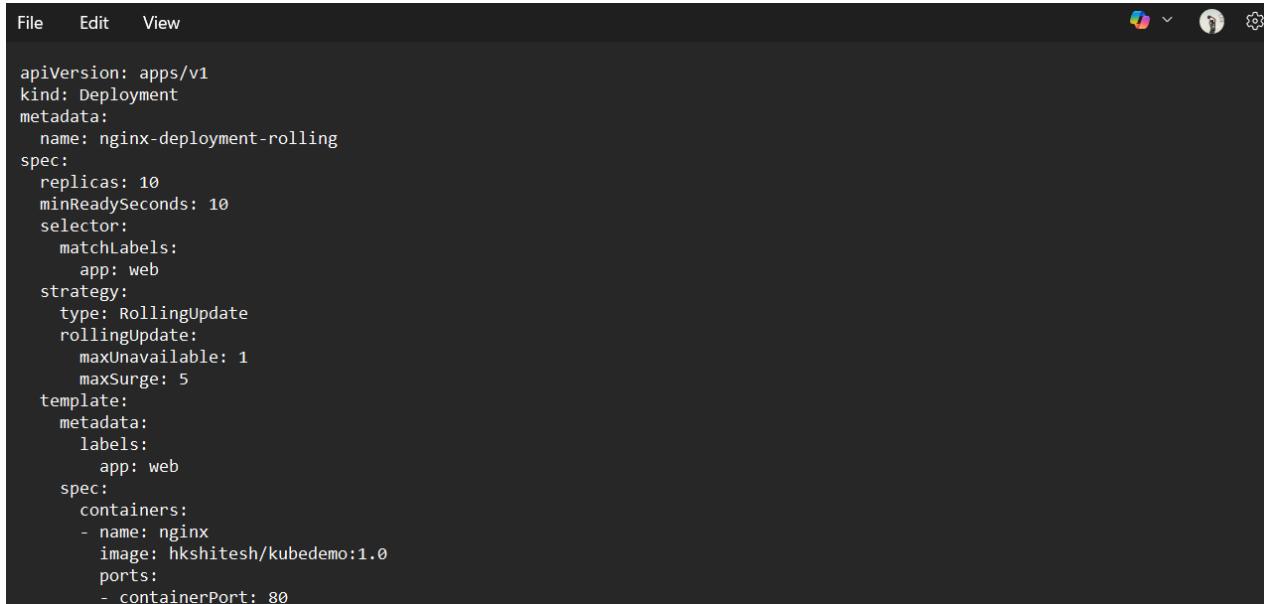
## Step 1: Create a Deployment with Rolling Update Strategy

Create a YAML file for the deployment:

Create a file named **nginx-deployment-rolling.yaml** with the following content:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment-rolling
spec:
  replicas: 10
  minReadySeconds: 10
  selector:
    matchLabels:
      app: web
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 5
```

```
template:
  metadata:
    labels:
      app: web
  spec:
    containers:
      - name: nginx
        image: hkshitesh/kubedemo:1.0
    ports:
      - containerPort: 80
```



```
File Edit View
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment-rolling
spec:
  replicas: 10
  minReadySeconds: 10
  selector:
    matchLabels:
      app: web
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 1
      maxSurge: 5
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
        - name: nginx
          image: hkshitesh/kubedemo:1.0
        ports:
          - containerPort: 80
```

Apply the deployment:

```
kubectl apply -f nginx-deployment-rolling.yaml ; watch "kubectl get rs -o wide"
```

```
PS C:\Users\ASUS\Desktop> kubectl apply -f nginx-deployment-rolling.yaml
deployment.apps/nginx-deployment-rolling created
PS C:\Users\ASUS\Desktop> |
```

Verify the deployment:

```
kubectl get deployments  
kubectl get pods -l app=web
```

```
PS C:\Users\ASUS\Desktop> kubectl get deployments  
NAME READY UP-TO-DATE AVAILABLE AGE  
nginx-deployment-rolling 10/10 10 10 74s  
PS C:\Users\ASUS\Desktop> kubectl get pods -l app=web  
NAME READY STATUS RESTARTS AGE  
nginx-deployment-rolling-85cf8767cc-4zfvb 1/1 Running 0 91s  
nginx-deployment-rolling-85cf8767cc-5fw4w 1/1 Running 0 91s  
nginx-deployment-rolling-85cf8767cc-5xn5n 1/1 Running 0 91s  
nginx-deployment-rolling-85cf8767cc-7b2km 1/1 Running 0 91s  
nginx-deployment-rolling-85cf8767cc-7xczx 1/1 Running 0 91s  
nginx-deployment-rolling-85cf8767cc-bxscx 1/1 Running 0 91s  
nginx-deployment-rolling-85cf8767cc-cshh2 1/1 Running 0 91s  
nginx-deployment-rolling-85cf8767cc-gccdc 1/1 Running 0 91s  
nginx-deployment-rolling-85cf8767cc-n4fb8 1/1 Running 0 91s  
nginx-deployment-rolling-85cf8767cc-pw7wl 1/1 Running 0 91s  
PS C:\Users\ASUS\Desktop> |
```

Update the deployment to a new image:

```
kubectl set image deployment/nginx-deployment-rolling nginx= hkshitesh/kubedemo:2.0
```

```
PS C:\Users\ASUS\Desktop> kubectl set image deployment/nginx-deployment-rolling nginx=hkshitesh/kubedemo:2.0  
deployment.apps/nginx-deployment-rolling image updated  
PS C:\Users\ASUS\Desktop>
```

Monitor the rolling update:

```
kubectl rollout status deployment nginx-deployment-rolling
```

```
PS C:\Users\ASUS\Desktop> kubectl rollout status deployment nginx-deployment-rolling  
deployment "nginx-deployment-rolling" successfully rolled out  
PS C:\Users\ASUS\Desktop> |
```

Verify the updated pods:

```
kubectl get pods -l app=web -o wide
```

```
PS C:\Users\ASUS\Desktop> kubectl get pods -l app=web -o wide
NAME                               READY   STATUS    RESTARTS   AGE
nginx-deployment-rolling-8587c95656-2qzt2   1/1     Running   0          56s
  10.244.0.36   minikube   <none>           <none>
nginx-deployment-rolling-8587c95656-6mwb4   1/1     Running   0          76s
  10.244.0.31   minikube   <none>           <none>
nginx-deployment-rolling-8587c95656-7pwft   1/1     Running   0          76s
  10.244.0.30   minikube   <none>           <none>
nginx-deployment-rolling-8587c95656-89czj   1/1     Running   0          53s
  10.244.0.37   minikube   <none>           <none>
nginx-deployment-rolling-8587c95656-dbghz   1/1     Running   0          76s
  10.244.0.32   minikube   <none>           <none>
nginx-deployment-rolling-8587c95656-gqx85   1/1     Running   0          76s
  10.244.0.33   minikube   <none>           <none>
nginx-deployment-rolling-8587c95656-kdswg   1/1     Running   0          76s
  10.244.0.29   minikube   <none>           <none>
nginx-deployment-rolling-8587c95656-n4ftg   1/1     Running   0          58s
  10.244.0.35   minikube   <none>           <none>
nginx-deployment-rolling-8587c95656-rrlvg   1/1     Running   0          51s
  10.244.0.38   minikube   <none>           <none>
nginx-deployment-rolling-8587c95656-zdt29   1/1     Running   0          76s
  10.244.0.34   minikube   <none>           <none>
```

```
PS C:\Users\ASUS\Desktop> kubectl delete deployment nginx-deployment-rolling
deployment.apps "nginx-deployment-rolling" deleted from default namespace
PS C:\Users\ASUS\Desktop> |
```

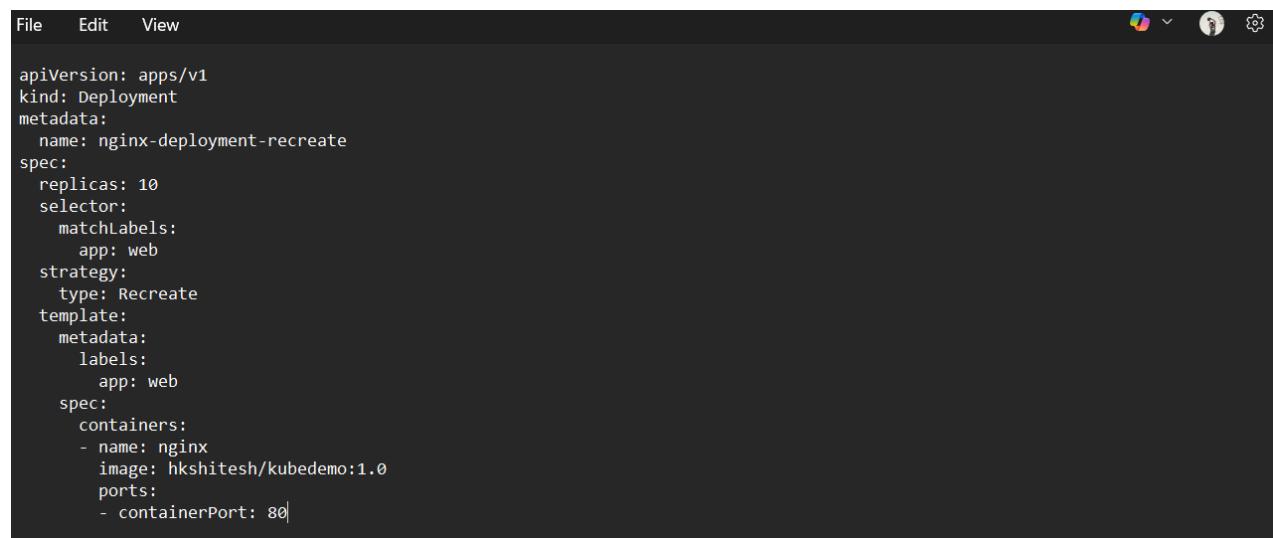
## Step 2: Create a Deployment with Recreate Strategy

Create a YAML file for the deployment:

Create a file named **nginx-deployment-recreate.yaml** with the following content:

```
apiVersion: apps/v1
kind: Deployment
metadata:
```

```
name: nginx-deployment-recreate
spec:
  replicas: 10
  selector:
    matchLabels:
      app: web
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
        - name: nginx
          image: nginx: hkshitesh/kubedemo:1.0
      ports:
        - containerPort: 80
```



The screenshot shows a dark-themed code editor window with a menu bar at the top. The menu bar includes 'File', 'Edit', 'View', and several icons for file operations and settings. The main area of the editor displays the same YAML configuration as the previous code block. The code defines a Deployment named 'nginx-deployment-recreate' with 10 replicas, a 'web' selector, a 'Recreate' strategy, and a template containing an NGINX container with port 80.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment-recreate
spec:
  replicas: 10
  selector:
    matchLabels:
      app: web
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
        - name: nginx
          image: hkshitesh/kubedemo:1.0
      ports:
        - containerPort: 80
```

Apply the deployment:

```
kubectl apply -f nginx-deployment-recreate.yaml ; watch "kubectl get rs -o wide"
```

```
PS C:\Users\ASUS\Desktop> notepad nginx-deployment-recreate.yaml
PS C:\Users\ASUS\Desktop> kubectl apply -f nginx-deployment-recreate.yaml
deployment.apps/nginx-deployment-recreate created
PS C:\Users\ASUS\Desktop> |
```

Verify the deployment:

```
kubectl get deployments
kubectl get pods -l app=nginx-recreate
```

```
PS C:\Users\ASUS\Desktop> kubectl get deployments
NAME                   READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment-recreate   10/10    10          10         21s
PS C:\Users\ASUS\Desktop> kubectl get pods -l app=web
NAME                           READY   STATUS    RESTARTS   AGE
nginx-deployment-recreate-85cf8767cc-768rp   1/1     Running   0          28s
nginx-deployment-recreate-85cf8767cc-7dmd4   1/1     Running   0          28s
nginx-deployment-recreate-85cf8767cc-lcjzqz   1/1     Running   0          28s
nginx-deployment-recreate-85cf8767cc-nxlk9    1/1     Running   0          28s
nginx-deployment-recreate-85cf8767cc-pb926    1/1     Running   0          28s
nginx-deployment-recreate-85cf8767cc-ps88c    1/1     Running   0          28s
nginx-deployment-recreate-85cf8767cc-q8rl7    1/1     Running   0          28s
nginx-deployment-recreate-85cf8767cc-rgjnc    1/1     Running   0          28s
nginx-deployment-recreate-85cf8767cc-v99xx    1/1     Running   0          28s
nginx-deployment-recreate-85cf8767cc-wfxmw    1/1     Running   0          28s
PS C:\Users\ASUS\Desktop> |
```

Update the deployment to a new image:

```
kubectl set image deployment/nginx-deployment-recreate nginx=nginx:1.21.1
```

```
PS C:\Users\ASUS\Desktop> kubectl set image deployment/nginx-deployment-recreate nginx=nginx:1.21.1
deployment.apps/nginx-deployment-recreate image updated
PS C:\Users\ASUS\Desktop> kubectl rollout status deployment nginx-deployment-recreate
Waiting for deployment "nginx-deployment-recreate" rollout to finish: 1 of 10
updated replicas are available...
Waiting for deployment "nginx-deployment-recreate" rollout to finish: 2 of 10
updated replicas are available...
Waiting for deployment "nginx-deployment-recreate" rollout to finish: 3 of 10
updated replicas are available...
Waiting for deployment "nginx-deployment-recreate" rollout to finish: 4 of 10
updated replicas are available...
Waiting for deployment "nginx-deployment-recreate" rollout to finish: 5 of 10
updated replicas are available...
Waiting for deployment "nginx-deployment-recreate" rollout to finish: 6 of 10
updated replicas are available...
Waiting for deployment "nginx-deployment-recreate" rollout to finish: 7 of 10
updated replicas are available...
Waiting for deployment "nginx-deployment-recreate" rollout to finish: 8 of 10
updated replicas are available...
Waiting for deployment "nginx-deployment-recreate" rollout to finish: 9 of 10
updated replicas are available...
deployment "nginx-deployment-recreate" successfully rolled out
PS C:\Users\ASUS\Desktop> |
```

Monitor the update:

```
kubectl rollout status deployment nginx-deployment-recreate
```

Verify the updated pods:

```
kubectl get pods -l app=nginx-recreate -o wide
```

### Step 3: Clean Up

Delete the deployments:

```
kubectl delete deployment nginx-deployment-rolling
```

```
kubectl delete deployment nginx-deployment-recreate
```

```
PS C:\Users\ASUS\Desktop> kubectl delete deployment nginx-deployment-recreate
deployment.apps "nginx-deployment-recreate" deleted from default namespace
```

Verify that all resources are cleaned up:

```
kubectl get deployments
kubectl get pods -l app=nginx
kubectl get pods -l app=nginx-recreate
```

```
PS C:\Users\ASUS\Desktop> kubectl get deployments
No resources found in default namespace.
PS C:\Users\ASUS\Desktop> kubectl get pods
No resources found in default namespace.
PS C:\Users\ASUS\Desktop> |
```