

Lab Exercise 11- Deployments with Rolling Update and Recreate Strategies

NAME : JIYA TYAGI

SAP ID: 500119743

BATCH 2 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
```

```
nginx-deplo: | ▶ +
File Edit View
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment-rolling
spec:
  replicas: 5
  selector:
    matchLabels:
      app: web
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
      - name: nginx
        image: nginx:1.19
        ports:
        - containerPort: 80
Ln 21, Col 28 | 353 character | Plain text | 100% | Windows (CF | UTF-8
```

Apply the deployment:

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

```
PS C:\Users\dimpl\k8s-lab> kubectl apply -f nginx-deployment-rolling.yaml
deployment.apps/nginx-deployment-rolling created
```

Verify the deployment:

```
kubectl get deployments
```

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

```
deployment.apps/nginx-deployment-rolling created
PS C:\Users\dimpl\k8s-lab> kubectl get deployments
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
nginx-deployment-rolling            0/5      5             0            10s
PS C:\Users\dimpl\k8s-lab> kubectl get pods -l app=web
NAME                                READY    STATUS
my-pod                              1/1      Running
nginx-deployment-rolling-5dcb777f77-2vw26 0/1      ContainerCre
ating 0          19s
nginx-deployment-rolling-5dcb777f77-4vqtn 0/1      ContainerCre
ating 0          19s
nginx-deployment-rolling-5dcb777f77-57c9j 0/1      ContainerCre
ating 0          19s
nginx-deployment-rolling-5dcb777f77-9d46w 0/1      ContainerCre
ating 0          19s
nginx-deployment-rolling-5dcb777f77-xjrcx 0/1      ContainerCre
```

Update the deployment to a new image:

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

Monitor the rolling update:

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

Verify the updated pods:

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

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
```

Apply the deployment:

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

Verify the deployment:

```
kubectl get deployments
```

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

```
PS C:\Users\dimpl\k8s-lab> notepad nginx-deployment-rolling.yaml
```

```
PS C:\Users\dimpl\k8s-lab> kubectl apply -f nginx-deployment-rolling.yaml
```

```
deployment.apps/nginx-deployment-rolling created
```

```
PS C:\Users\dimpl\k8s-lab> kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
nginx-deployment-rolling	0/5	5	0	10s

```
PS C:\Users\dimpl\k8s-lab> kubectl get pods -l app=web
```

NAME	READY	STATUS
my-pod	1/1	Running
nginx-deployment-rolling-5dcb777f77-2vw26	0/1	ContainerCreating
nginx-deployment-rolling-5dcb777f77-4vqtn	0/1	ContainerCreating
nginx-deployment-rolling-5dcb777f77-57c9j	0/1	ContainerCreating
nginx-deployment-rolling-5dcb777f77-9d46w	0/1	ContainerCreating
nginx-deployment-rolling-5dcb777f77-xjrcx	0/1	ContainerCreating

Update the deployment to a new image:

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

```
PS C:\Users\dimpl\k8s-lab> kubectl set image deployment/nginx-deployment-rolling nginx=nginx:latest
deployment.apps/nginx-deployment-rolling image updated
```

Monitor the update:

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

```

PS C:\Users\dimpl\k8s-lab> kubectl rollout status deployment nginx-deployment-rolling
Waiting for deployment "nginx-deployment-rolling" rollout to finish: 1 old replicas are pending termination...
deployment "nginx-deployment-rolling" successfully rolled out

```

Verify the updated pods:

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

```

PS C:\Users\dimpl\k8s-lab> kubectl get pods -l app=nginx-recreate

```

NAME	READY	STATUS	RES
nginx-deployment-recreate-96466949-9c9b4	1/1	Running	0
nginx-deployment-recreate-96466949-c968g	1/1	Running	0
nginx-deployment-recreate-96466949-ghzm5	1/1	Running	0
nginx-deployment-recreate-96466949-hqbm5	1/1	Running	0
nginx-deployment-recreate-96466949-w6s8t	1/1	Running	0

Step 3: Clean Up

Delete the deployments:

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

```

PS C:\Users\dimpl\k8s-lab> kubectl delete deployment nginx-deployment-rolling
deployment.apps "nginx-deployment-rolling" deleted from default namespace
PS C:\Users\dimpl\k8s-lab> kubectl delete deployment nginx-deployment-recreate
deployment.apps "nginx-deployment-recreate" deleted from default namespace
PS C:\Users\dimpl\k8s-lab>

```

Verify that all resources are cleaned up:

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

```
namespace
PS C:\Users\dimpl\k8s-lab> kubectl get deployments
No resources found in default namespace.
PS C:\Users\dimpl\k8s-lab>
```