

## Lab Exercise 10– Creating Deployment in Kubernetes

Below is a lab exercise that demonstrates how to create and manage a Deployment in Kubernetes.

### Step 1: Create a Deployment Configuration File

Create a file named deployment.yaml with the following configuration:

Link of file: (Copy following code from my GitHub repo)

<https://github.com/hkshitesh/ACO-LAB-2021-25/blob/main/scripts/deployment.yaml>

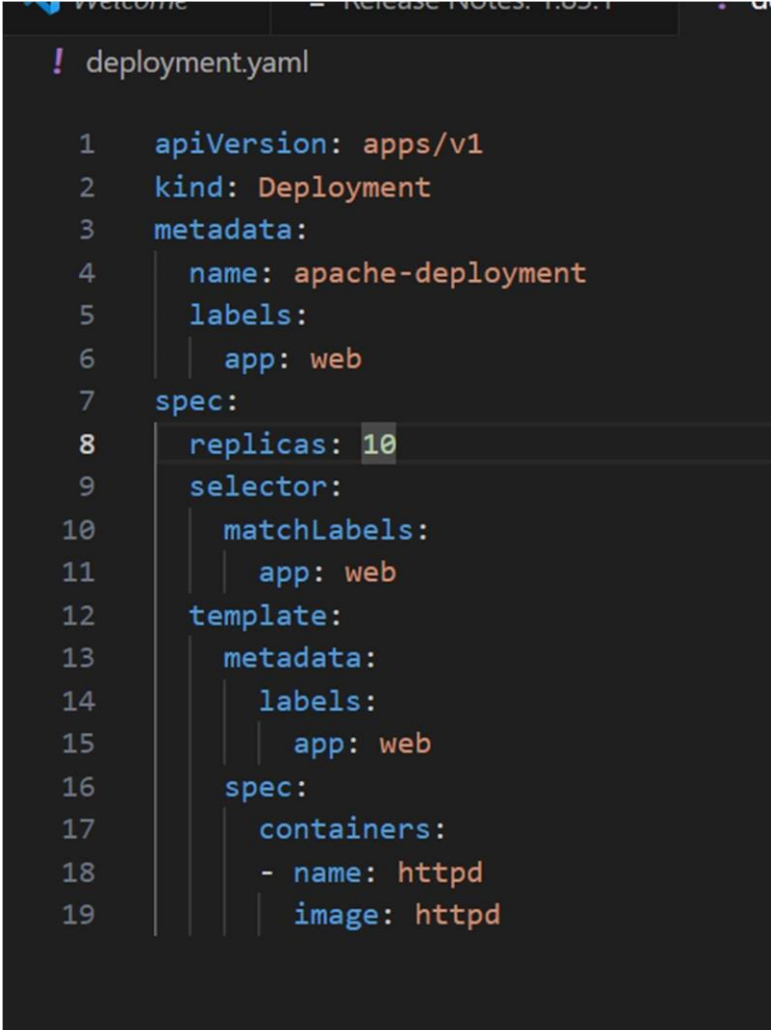
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: apache-deployment
  labels:
    app: web
spec:
  replicas: 10
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
```

```
spec:

  containers:

  - name: httpd

    image: httpd
```

A screenshot of a code editor with a dark theme. The file is named 'deployment.yaml' and is shown with line numbers from 1 to 19. The code is a Kubernetes Deployment manifest. It specifies the API version as 'apps/v1', the kind as 'Deployment', and the metadata including the name 'apache-deployment' and a label 'app: web'. The spec section defines 10 replicas, a selector matching the 'app: web' label, and a template with a container named 'httpd' using the 'httpd' image.

```
! deployment.yaml

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: apache-deployment
5    labels:
6      app: web
7  spec:
8    replicas: 10
9    selector:
10     matchLabels:
11       app: web
12   template:
13     metadata:
14       labels:
15         app: web
16     spec:
17       containers:
18       - name: httpd
19         image: httpd
```

## Step 2: Apply the Deployment Configuration

Apply the configuration to create the Deployment:

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

```
PS C:\Users\Maan\Desktop\Exp_10> kubectl apply -f deployment.yaml
deployment.apps/apache-deployment created
```

### Step 3: View the Deployment and Pods

View the created Deployment and the associated Pods:

```
kubectl get deployments
```

```
kubectl get pods
```

```
PS C:\Users\Maan\Desktop\Exp_10> kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
apache-deployment   0/10    10           0           34s
PS C:\Users\Maan\Desktop\Exp_10> kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
apache-deployment-5646dd6f6c-5ftmb  0/1     ContainerCreating   0          76s
apache-deployment-5646dd6f6c-7vx5f  0/1     ContainerCreating   0          76s
apache-deployment-5646dd6f6c-8rbph  1/1     Running             0          76s
apache-deployment-5646dd6f6c-cxw4m  1/1     Running             0          76s
apache-deployment-5646dd6f6c-h8bxq  1/1     Running             0          76s
apache-deployment-5646dd6f6c-nfn16  0/1     ContainerCreating   0          76s
apache-deployment-5646dd6f6c-ng6nw  0/1     ContainerCreating   0          76s
apache-deployment-5646dd6f6c-ph855  0/1     ContainerCreating   0          76s
```

### Step 4: Delete the Deployment

Delete the Deployment:

```
kubectl delete deployment my-deployment
```

```
Error from server (NotFound): deployments.apps "my-deployment" not found
PS C:\Users\Maan\Desktop\Exp_10> kubectl delete deployment apache-deployment
deployment.apps "apache-deployment" deleted
PS C:\Users\Maan\Desktop\Exp_10> █
```

## **Conclusion**

This exercise demonstrated how to create, manage, and update a Deployment in Kubernetes. You learned how to scale the Deployment, update the image, and perform a rolling update to the Deployment. Experiment further with different configurations and update strategies to deepen your understanding of managing Deployments in Kubernetes.