Deployments are upgraded and higher version of replication controller. They manage the deployment of replica sets which is also an upgraded version of the replication controller. They have the capability to update the replica set and are also capable of rolling back to the previous version.

They provide many updated features of **matchLabels** and **selectors**. We have got a new controller in the Kubernetes master called the deployment controller which makes it happen. It has the capability to change the deployment midway.

Changing the Deployment

Updating − The user can update the ongoing deployment before it is completed. In this, the existing deployment will be settled and new deployment will be created.

Deleting − The user can pause/cancel the deployment by deleting it before it is completed. Recreating the same deployment will resume it.

Rollback − We can roll back the deployment or the deployment in progress. The user can create or update the deployment by using **DeploymentSpec.PodTemplateSpec = oldRC.PodTemplateSpec.**

Deployment Strategies

Deployment strategies help in defining how the new RC should replace the existing RC.

Recreate − This feature will kill all the existing RC and then bring up the new ones. This results in quick deployment however it will result in downtime when the old pods are down and the new pods have not come up.

Rolling Update − This feature gradually brings down the old RC and brings up the new one. This results in slow deployment, however there is no deployment. At all times, few old pods and few new pods are available in this process.

The configuration file of Deployment looks like this.

```
apiVersion: extensions/v1beta1 --------------------->1
kind: Deployment --------------------------> 2
metadata:
   name: Tomcat-ReplicaSet
spec:
   replicas: 3
   template:
      metadata:
         lables:
            app: Tomcat-ReplicaSet
            tier: Backend
   spec:
      containers:
         - name: Tomcatimage:
            tomcat: 8.0
            ports:
               - containerPort: 7474
```

In the above code, the only thing which is different from the replica set is we have defined the kind as deployment.

Create Deployment

```
$ kubectl create –f Deployment.yaml --record
deployment "Deployment" created Successfully.
```

Fetch the Deployment

```
$ kubectl get deployments
NAME           DESIRED     CURRENT     UP-TO-DATE      AVILABLE   AGE
Deployment        3           3           3              3        20s
```

Check the Status of Deployment

```
$ kubectl rollout status deployment/Deployment
```

Updating the Deployment

```
$ kubectl set image deployment/Deployment tomcat=tomcat:6.0
```

Rolling Back to Previous Deployment

```
$ kubectl rollout undo deployment/Deployment –to-revision=2
```