**Notes by R V K ROHINI PRIYA**

**Deployment in Kubernetes:**

Deployment is a combination of Pod and a Replicaset.

A pod is a thing on this container with our application runs.

A replica set is one that makes alternate pod available when current pod destroys for some reason.

A Deployment technique has different number of pods running for the same application.

**Advantages of Deployment:**

1. High availability of application.

2. Load is distributes between different pods.

This is called as Round Robin Fashion Technique.

**Example:**

Let us consider an application ABC with version one (V1).

By using deployment we make it run on different pods.

Now if I want to make any changes in the application and change the version from V1 to V2.

By deploying directly there is a chance of occurence of downtime, which is not a good thing from both from customer and developer point of view.

Also, there is chance of ecxisting of bugs in new version. Before deploying the V2 we need to verify that everything is working smooth.

So by using deployment technique we only deploy new version (V2) in one for few pods. If it works fine then we deploy on other pods too.

There are two ways of deployment technique.

1. Rolling update.

2. Recreate.

**Rolling update:**

As we discussed, the technique in which one by one pod is updated with new version is called as Rolling update.

In this we use disruption.

**Recreate:**

In this what happens is all the pods running similar application will at a time change the application with updated version.

Problem with is downtime occurs for few seconds.

And if any bugs are there in application, then application doesnt work properly and creates problem to user who are using, since all pods have version with bugs.

**Rollout a replicaset:**

With this we can run the same application any number of times on any numbers of pods at a time.

It simply means replica ( a copy)

We declare it as "replica:number of replica pods needed"

Like "replica:4"

**Rollback:**

There is a concept of Rollback. With this we can rollback one by one version back to the

previous version again.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Types of Deployment:**

1. Blue green Deployment (most popular)
2. Canary Deployment
3. A/B Deployment

**Patching:**

Means there is a bug in our code and patch it (remove or update or fix).

So, we create a small code and add or patch with old code to remove bug.

(updating means totally changing the code with new code)

Example: windows update.

Everytime we dont install everything. Just install that small changes of code.

So we can say thing every update of application is called patching.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Hands-On:**

1. Create an unbuntu Instance

2. Connect to the instance from our system by using

ssh -i ubuntu@<Ip of instance>

3. Become a root user

sudo -i

4. To see the available nodes

kubectl get nodes

5. Now write a yaml code for deployment and replicasets and use here

nano deployment.yaml

6. Create deployment pods by using

kubectl apply -f deployment.yaml

7. To see the Deployment

kubectl get deployment

or

kubectl get deploy

8. To see the Replicaset

kubectl get rs

9. To see the available number of pods

kubectl get pod

10. To know complete details of deployment

kubectl describe deploy

11. To know complete details of pods

kubectl describe pod <pod id>

kubectl describe pod myapp-deployment-84fcb7989f-zwhk7

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Note:**

1. Deployment will have a unique Id

Example: myapp-deployment

1. Replicaset will have an Id

Example: myapp-deployment-84fcb7989f

1. Pod is a combination of Three

Its id had “Deployment-Id + Replica Set-Id + its own unique hash Id”

Example: myapp-deployment-84fcb7989f-6wtkj