**15. Kubernetes Deployment - Update Deployment using Set Image Option**

--- Reference - <https://github.com/stacksimplify/kubernetes-fundamentals/tree/master/04-Deployments-with-kubectl/04-02-Update-Deployment>

**--- We can update deployments using two options**

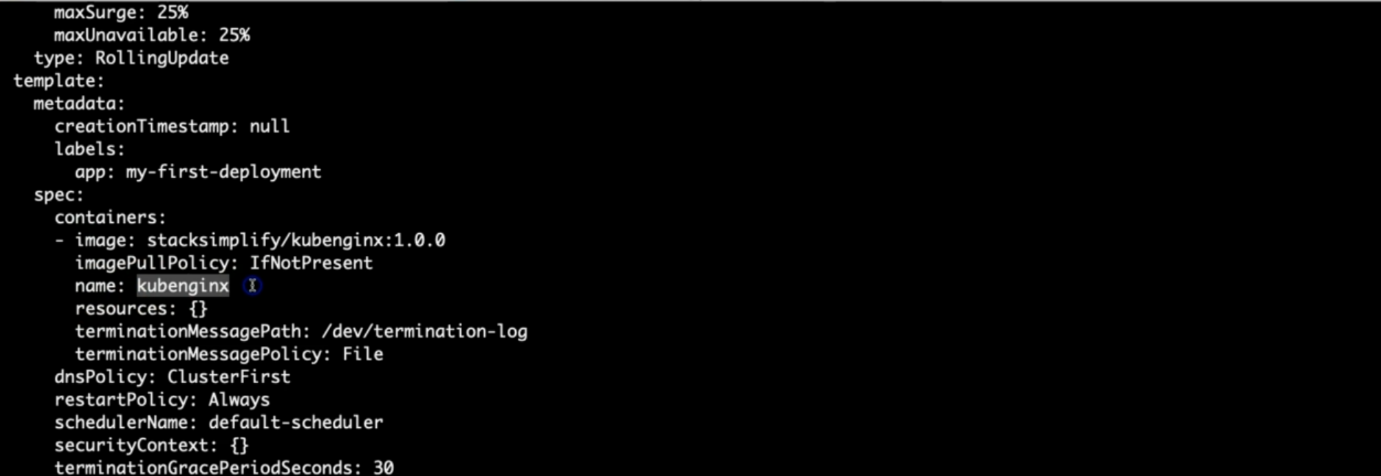
1. **Set Image**
2. **Edit Deployment**

**Updating Application version V1 to V2 using "Set Image" Option**

--- **Observation:** Please Check the container name in spec.container.name yaml output and make a note of it and replace in kubectl set image command

**# Get Container Name from current deployment**

--- **kubectl get deployment my-first-deployment -o yaml**



--- **note** – use command to get container name.

**# Update Deployment - SHOULD WORK NOW**

--- **kubectl set image deployment/<Deployment-Name> <Container-Name>=<Container-Image> --record=true**

--- **kubectl set image deployment/my-first-deployment kubenginx=stacksimplify/kubenginx:2.0.0 --record=true**

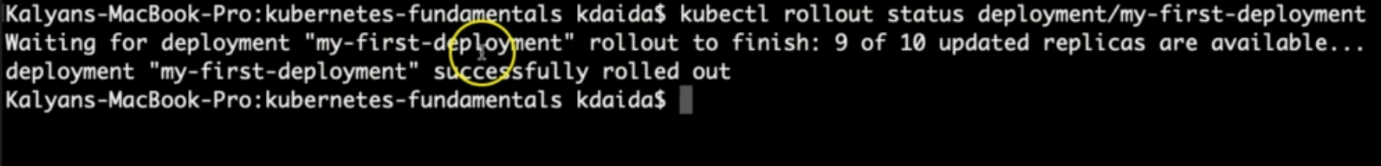
--- **--record=true** – it will enable versioning. If you give this parameter then kubernetes will record the deployment.

**Verify Rollout Status (Deployment Status)**

--- **Observation:** By default, rollout happens in a rolling update model, so no downtime.

**# Verify Rollout Status**

--- **kubectl rollout status deployment/my-first-deployment**



**# Verify Deployment**

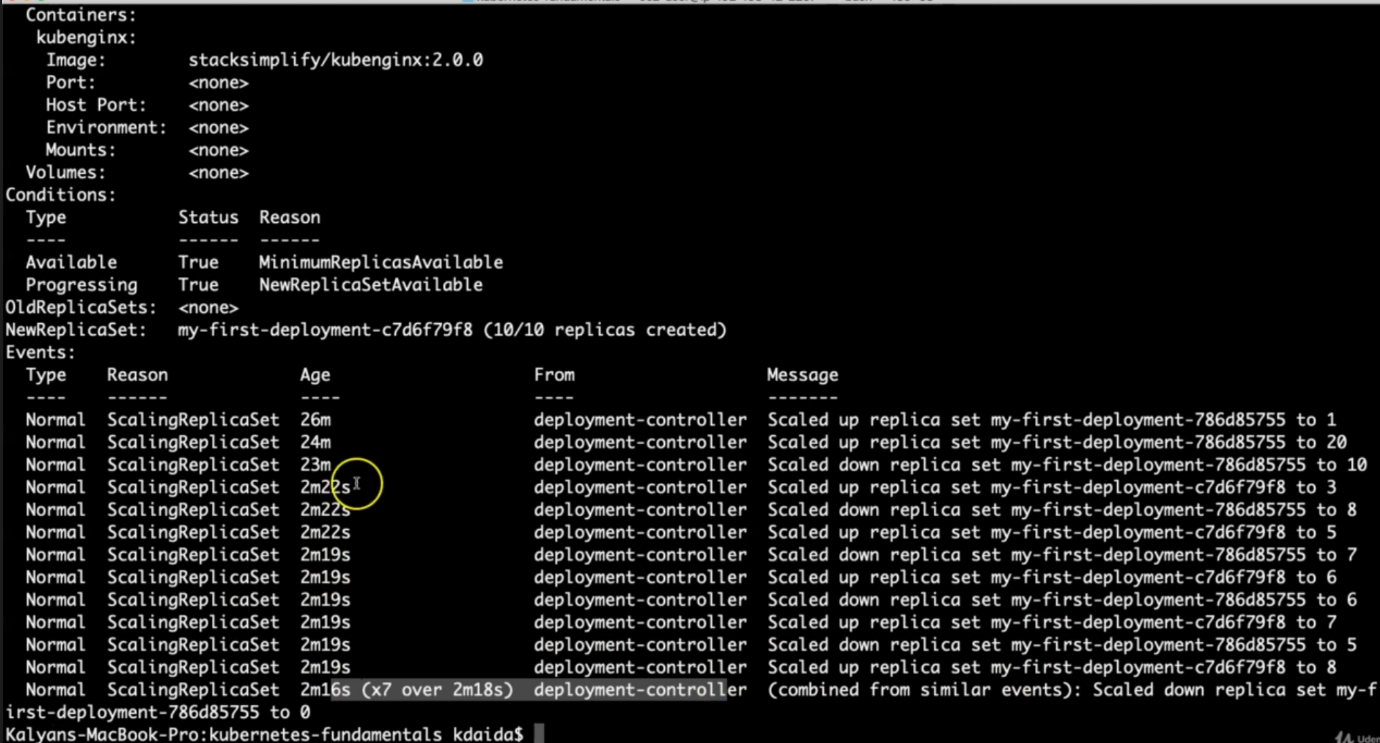
--- **kubectl get deploy**

**Describe Deployment**

--- **Observation** - Verify the Events and understand that Kubernetes by default do "Rolling Update" for new application releases. With that said, we will not have downtime for our application.

**# Descibe Deployment**

--- **kubectl describe deployment my-first-deployment**



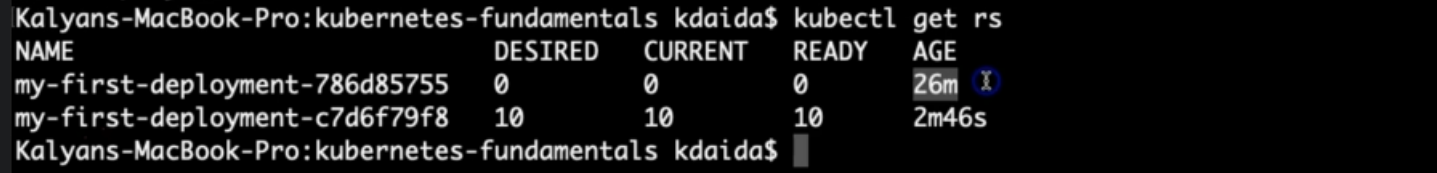
--- **note** – kubernetes scale up our application one by one.

**Verify ReplicaSet**

--- **Observation**: New ReplicaSet will be created for new version

**# Verify ReplicaSet**

--- **kubectl get rs**



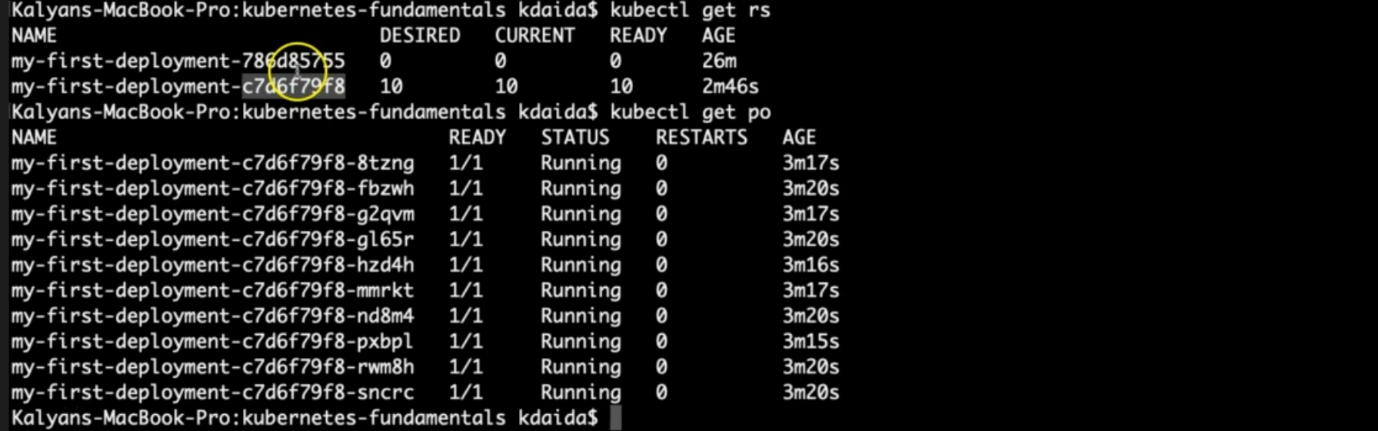
--- **note** – a new replicaset will be created for new version. Here you can see that the 1st replicaset age is 26m, this is old replicaset. 2nd replicaset age is 2m46s and this one is new replicaset.

**Verify Pods**

--- **Observation:** Pod template hash label of new replicaset should be present for PODs letting us know these pods belong to new ReplicaSet.

**# List Pods**

--- kubectl get po



--- note – our pods label**(c7d6f79f8)** is pointing towards second replicaset.

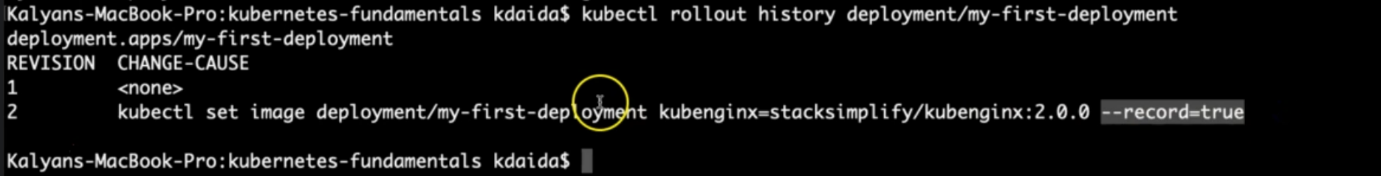
**Verify Rollout History of a Deployment**

--- **Observation:** We have the rollout history, so we can switch back to older revisions using revision history available to us.

**# Check the Rollout History of a Deployment**

--- **kubectl rollout history deployment/<Deployment-Name>**

--- **kubectl rollout history deployment/my-first-deployment**



--- **note** – it recorded our latest deployment.

**Access the Application using Public IP**

--- **note** - We should see Application Version:V2 whenever we access the application in browser

**# Get NodePort**

--- kubectl get svc

--- **Observation:** Make a note of port which starts with 3 (Example: 80:3xxxx/TCP). Capture the port 3xxxx and use it in application URL below.

**# Get Public IP of Worker Nodes**

--- kubectl get nodes -o wide

--- **Observation:** Make a note of "EXTERNAL-IP" if your Kubernetes cluster is setup on AWS EKS.

**# Application URL**

--- **http://<worker-node-public-ip>:<Node-Port>**