

1. By default, whenever we create a deployment, a replica set it is rolled out.
2. Updating the deployment – when every we want to change our application version ,update our application version , we can do the updating of deployment

3 .Easily rolling back a Development to our respective older deployment, whatever we want, because it is maintaining the versions. Or if you have updated from V1 to V2, V2 to V3,and then V3 to V4.And now again, if you want to go back to V2, yes,you will be able to go because it maintains the versions, and t

1. Scaling a Development : replicas where we can add the number of replicas

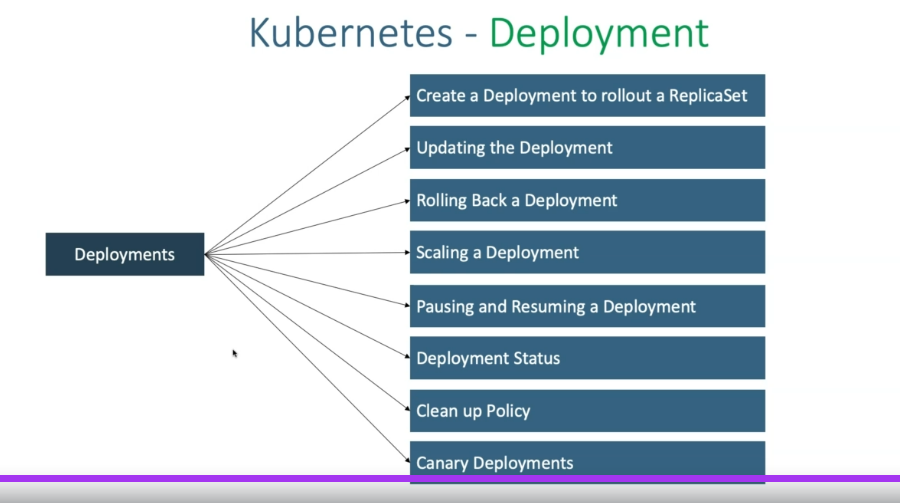
Or we can use it kubectl scale to scale our development

1. Pausing and resuming a development
2. Clean up Policy : it maintains the rollout version history and by default ,it maintained last 10 version of application

We will able to add more version using yaml in spce

1. Canary Development : if you want to add a new version of our application in the live production traffic and the traffic should be distribute with old version and the new version

Still we will able to do it using canary development



1. Create a development

#kubectl create deployment my-first-deployment - -image :stacksimplify/kubenginx:1.0.0

Then

#kubectl get deployments

#kubectl get deploy

#kubectl describe deployments name of deply

#kubectl get pods

#kubectl get svc

To Scale our deplovemnt

#kubectl scale –replicas =20 deployment my-deployment-name

#kubectl get deploy

#kubectl get pods

To scale down

#kubectl scale –replicas:10 deployment name\_of\_deployment

#kubectl get pods

EXPOSE DEPLOYMENTS AS SERVICE :

#kubectl expose deployment name\_of\_deplyment --type: NodePort –port: 80 --target-port: 80 –name: my-first-deployment-service

#kubect get svc

#kubect get nodes –o wide

UPDATE THE DEPLOYMENT:

We can update the deployment using two options:

1. Set Image

#kubectl set image deployment/Name\_of\_deplyment kubenginx=stacksimplify/kubenginx:2.0.0 –record=true

Kubebginx is container name

* -record=true – enable version

How do you get the container name?

Kubectl get deployment my-first-deployment –o yaml

\*++Then verify roll out

#kubectl rollout status deployment/my-first-deployment (name of Deployment)

#kubectl get pods

#kubectl get deploy

Describe DEVELOPEMT:

Verify the events and understand that kubernate by default do “Rolling Update” for new application release

Kubernates does the rolling update what happen is like the downtime of our application is going to be zero

So one after other it is going to update our pods

One more strategy in this update deployment there will be something called recreate

So if it recreates, which means like it will delete all the pods and then recreate at that time

So at the time you can expect the downtime for your application

By default it is going to be rolling update

Sometimes, my sql, need to re-create the data in this case you use recreate strategy

But in general application what every you use 99.99%will be using rolling updates

**Whenever you done the update, it will create the new replica set, old replicas set make 0**

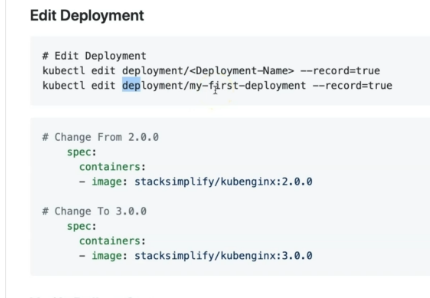
Verify your roll out history of development:

#kubectl rollout history deployment/my-first-development

1. **Edit deployment**

Updating the application from V2 to V3 using ËDIT Deployment”Option

#kubectl edit deployment/<deployment-name> --record=true



Roll out history

#kubectl rollout status deployment/my-first-deployment

#kubectl get pods

Check replicaset

#kubectl get rs

Create the new replicas set and make 0 for old rs

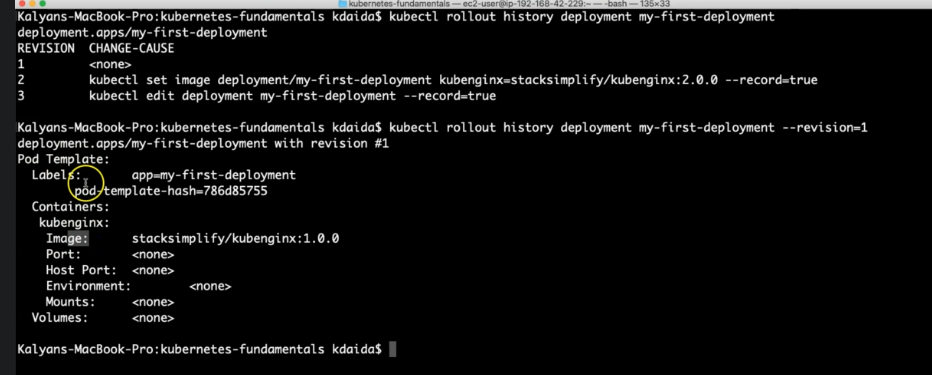
**RollBack Development of Application**

Check the roll out history of a deployment

#kubectl rollout history development development\_name

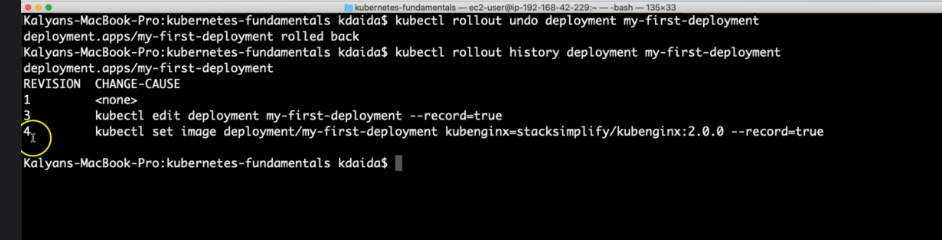
If you want to see the specific version what had changed

#kubectl rollout history developemt my-first-development –revision=1



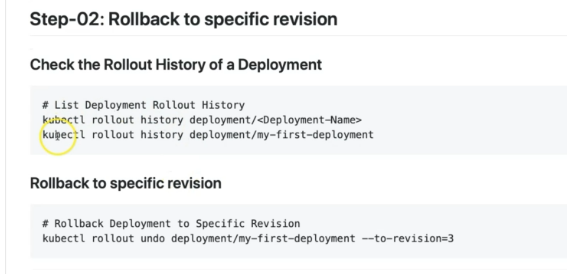
Roll back to previous version :

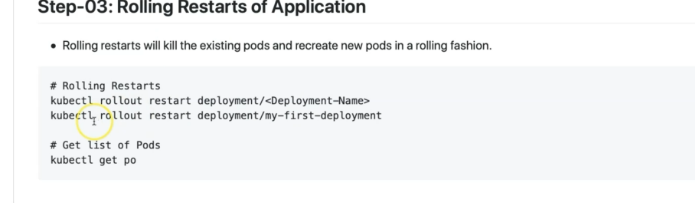
#kubectl rollout undo deploy my-first-deployment





Roll back to Specific Version:



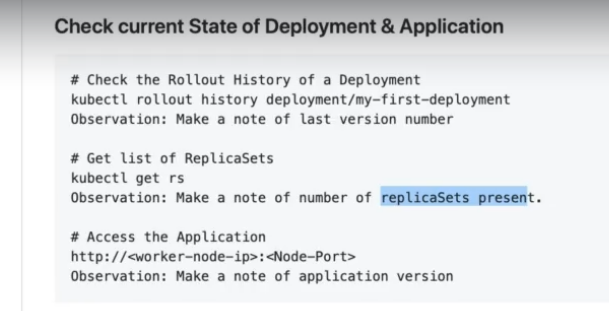


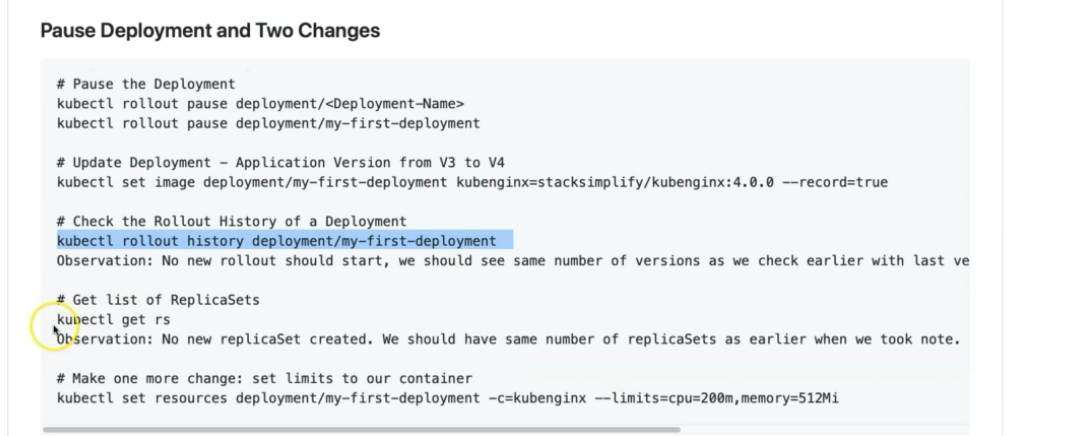
Kubernetes deployment –Pause and Resume Deployment

So why do we need pausing and then resuming deployments?

So if we want to make multiple changes to our deployments,

we can pause the deployment,make all changes to our deployment,and then resume it.So if you don't do so,as soon as you keep applying the changes to your deployment, live deployment, then automatically the changes gets deployed and then parts get terminated and then recreated, okay?





Once pause the development and will do some change on development but still doesn’t update because still development is pause

Once resume development then will update



