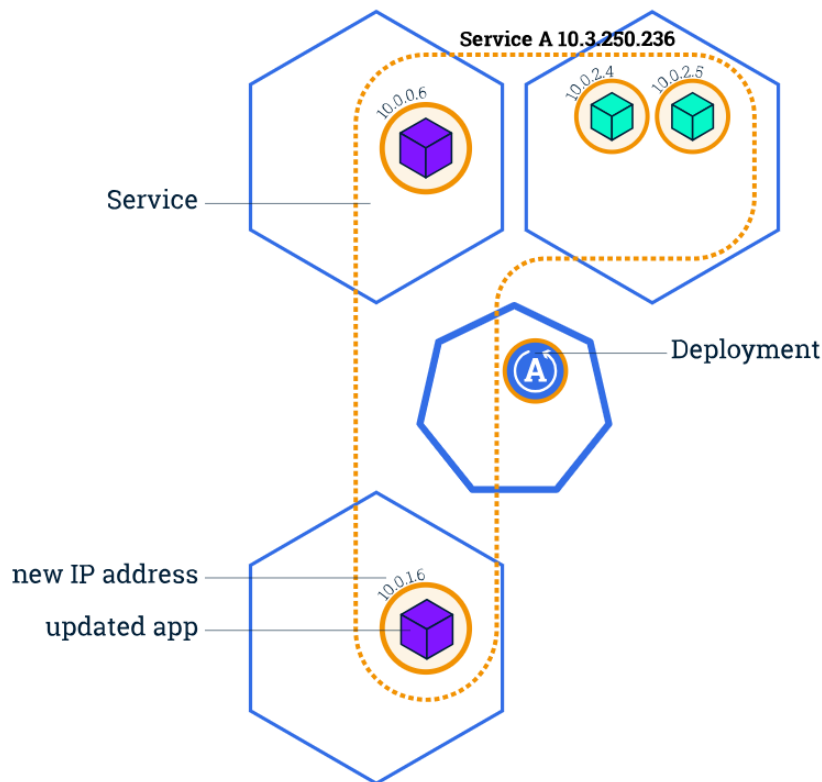




Kubernetes

Rolling Updates

A **rolling update** allows a Deployment update to take place with zero downtime



Rolling updates allow the following actions:

- Promote an application from one environment to another (via container image updates)
- Rollback to previous versions
- Continuous Integration and Continuous Delivery of applications with zero downtime



Here is a hands-on task that describes how the rolling updates work

A) Creating the deployment and NodePort service

Create a deployment named myapp with the required container image

```
controlplane $ kubectl create deployment myapp --image docker.io/vaishnavivyawahare26/myapp:v1
deployment.apps/myapp created
```

Here, I have used an image from my docker hub repository

Deployment “myapp” successfully created

```
controlplane $ kubectl get deployments.apps
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
myapp	1/1	1	1	9s

Pod managed by the deployment “myapp”

```
controlplane $ kubectl get pod
```

NAME	READY	STATUS	RESTARTS	AGE
myapp-57f8bf559d-hdm6q	1/1	Running	0	15s

Now, expose the deployment “myapp” to create a NodePort service. This will make the application accessible outside the cluster

```
controlplane $ kubectl expose deployment myapp --port 80 --target-port 80 --type NodePort
service/myapp exposed
controlplane $ kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	15d
myapp	NodePort	10.99.248.47	<none>	80:32701/TCP	7s

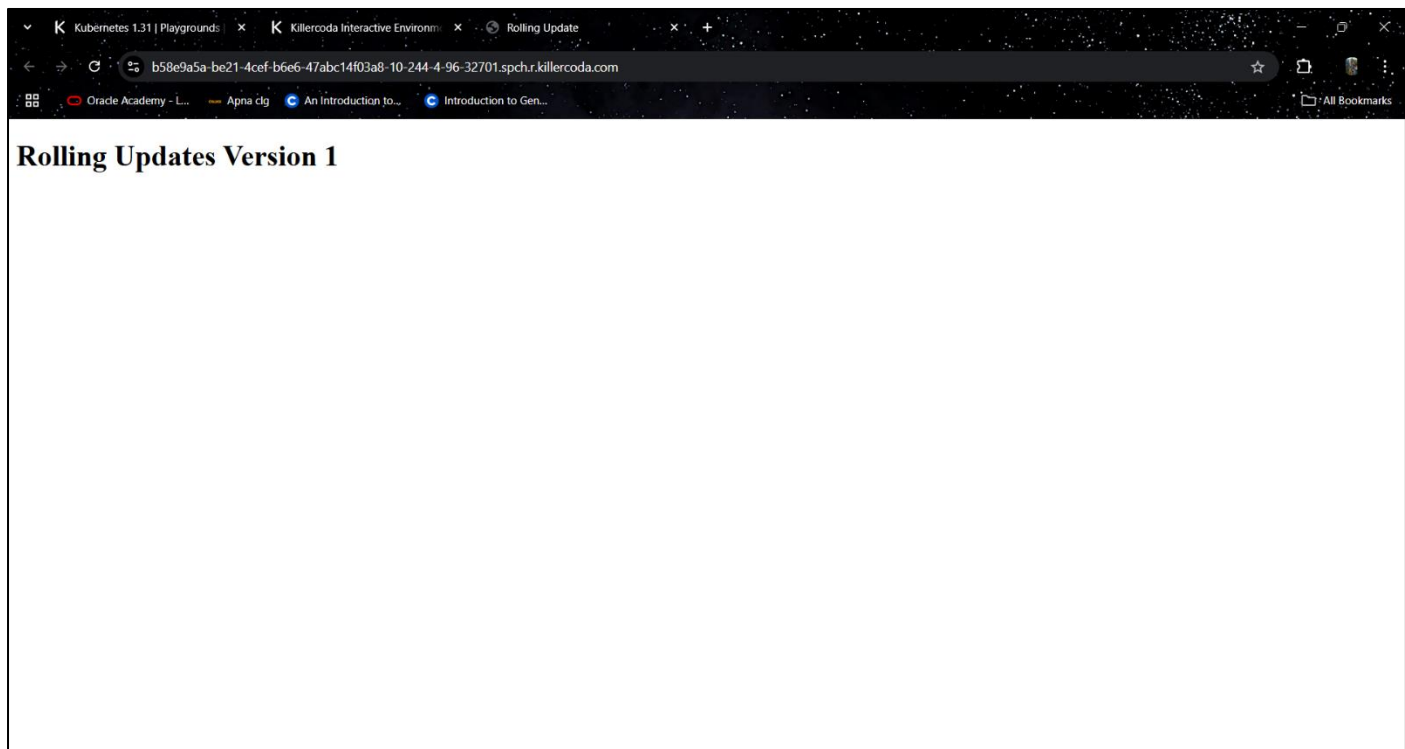


Successfully created NodePort service for “myapp” deployment

```
controlplane $ kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	15d
myapp	NodePort	10.99.248.47	<none>	80:32701/TCP	7s

Application is now accessible





B) Rollout the deployment to version 2

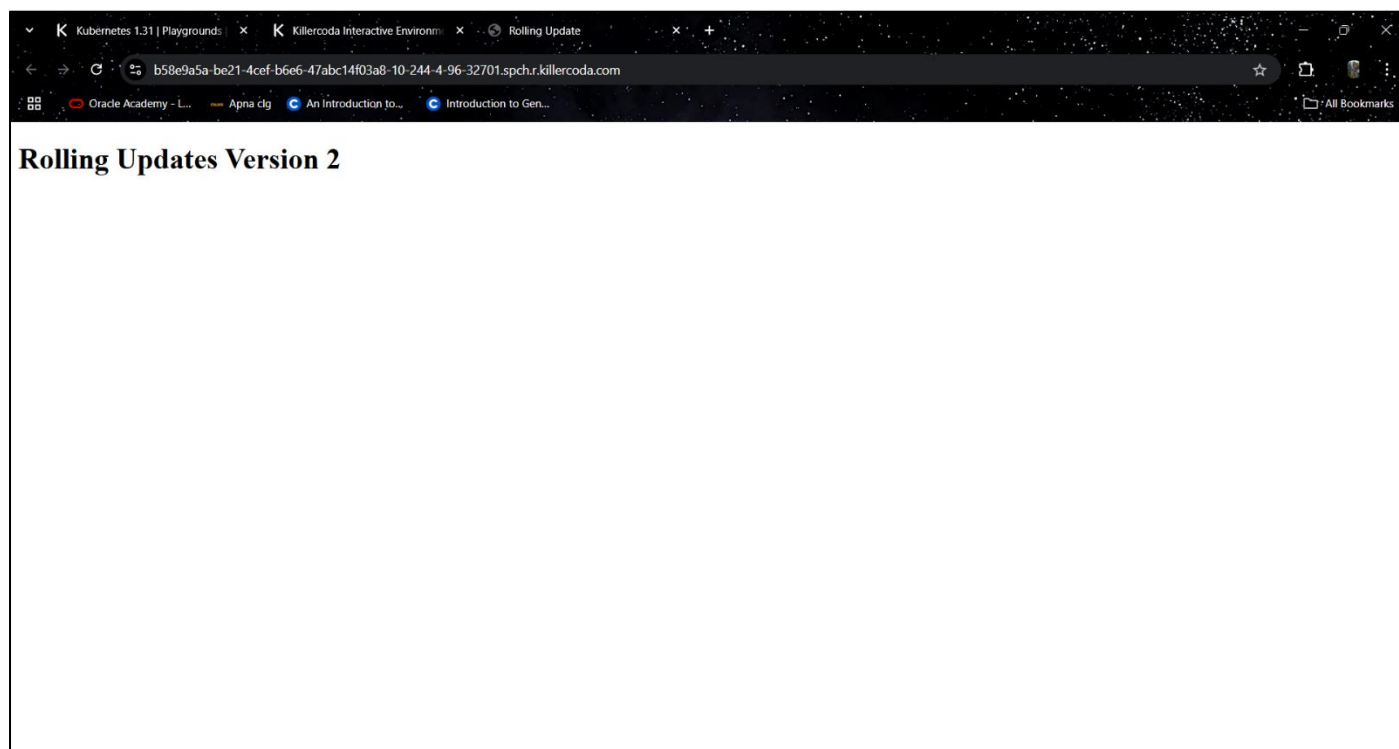
To update the deployment we will use

```
kubectl set image deployments/<deployment_name>  
container_name=<image>
```

here * is used for all the containers and the dockerhub image with updated version

```
controlplane $ kubectl set image deployments/myapp *=docker.io/vaishnaviyawahare26/myapp:v2  
deployment.apps/myapp image updated
```

Application is successfully updated to version2





The rollout history shows the number of versions updated for the required deployment

Check the rollout history using

Kubectl rollout history deployment <deployment_name>

```
controlplane $ kubectl rollout history deployment myapp
deployment.apps/myapp
REVISION    CHANGE-CAUSE
1           <none>
2           <none>
```

List Revisions: Displays all revisions of the specified Deployment along with their revision numbers.

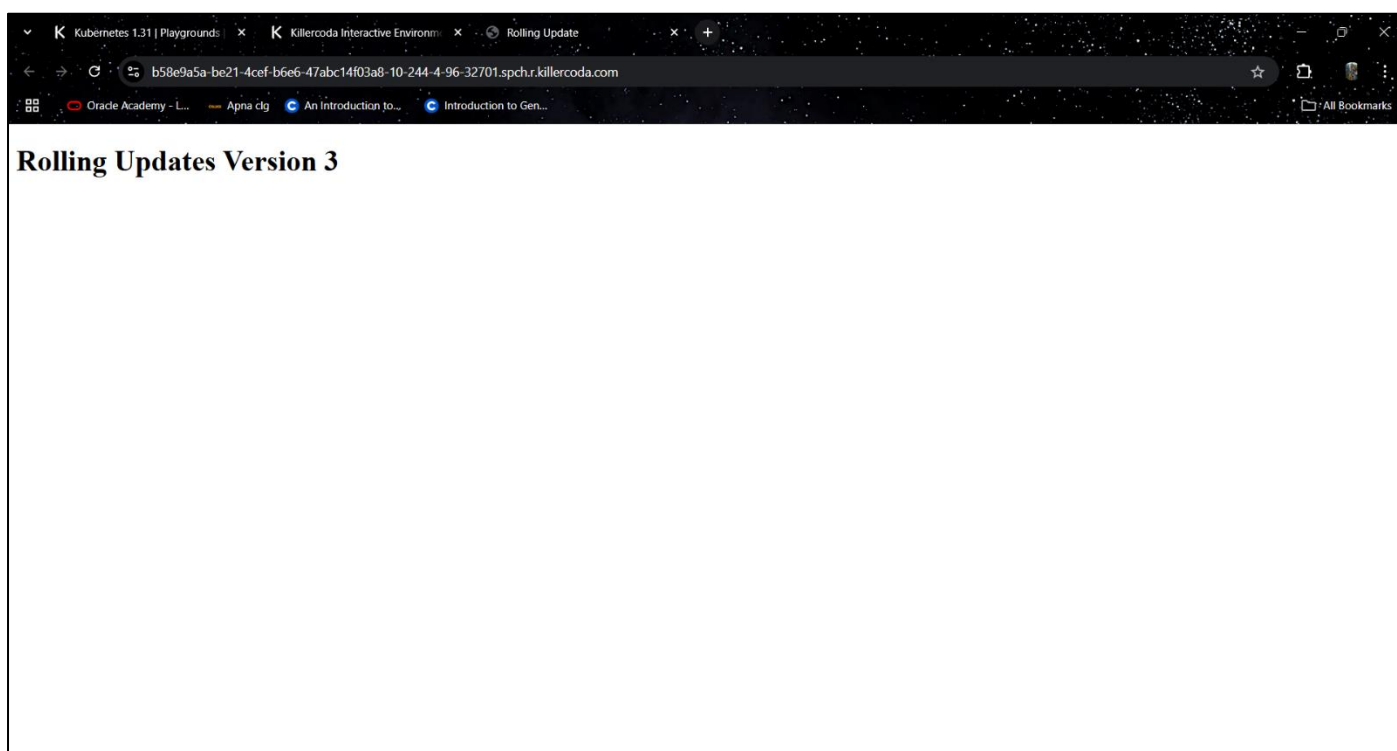


C) Rollout the deployment to version 3

Again to update the application set the image with version 3

```
controlplane $ kubectl set image deployments/myapp *=docker.io/vaishnavivyawahare26/myapp:v3
deployment.apps/myapp image updated
```

Application successfully updated to version3



Now, Revision 3 is the current version for the deployment

```
controlplane $ kubectl rollout history deployment myapp
deployment.apps/myapp
REVISION  CHANGE-CAUSE
1          <none>
2          <none>
3          <none>
```



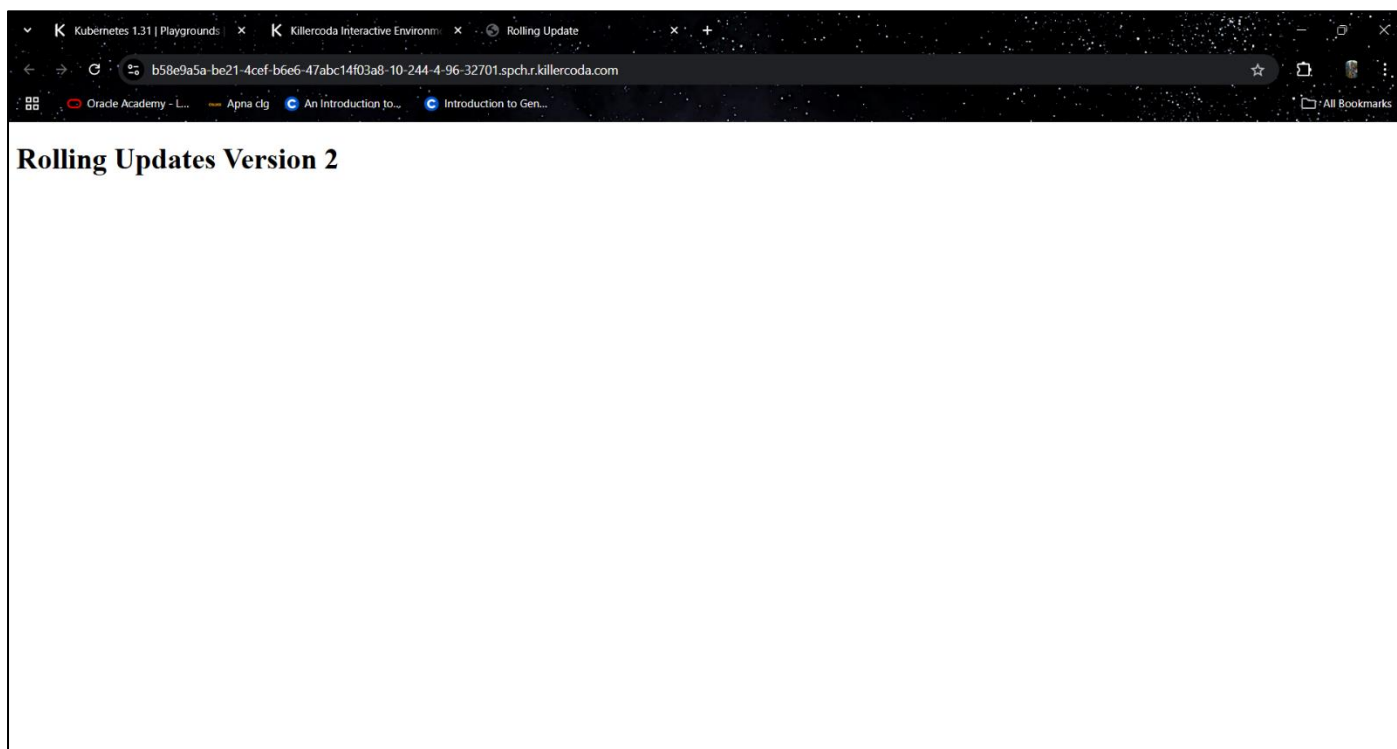
D) Rollback the deployment to version 2

We can also roll back the update to the previous version by using

kubectl rollout undo deployment <deployment_name> --to-revision <revision_number>

```
controlplane $ kubectl rollout undo deployment myapp --to-revision 2
deployment.apps/myapp rolled back
```

Deployment is successfully rolled back to version 2





This will also update the rollout history marking the current version as revision 4

```
controlplane $ kubectl rollout history deployment myapp
deployment.apps/myapp
REVISION  CHANGE-CAUSE
1          <none>
3          <none>
4          <none>
```

E) Rollback the deployment to version1

Rollback the deployment again to set the version 1

```
controlplane $ kubectl rollout undo deployment myapp --to-revision 1
deployment.apps/myapp rolled back
```

Deployment is successfully rolled back to version 1

