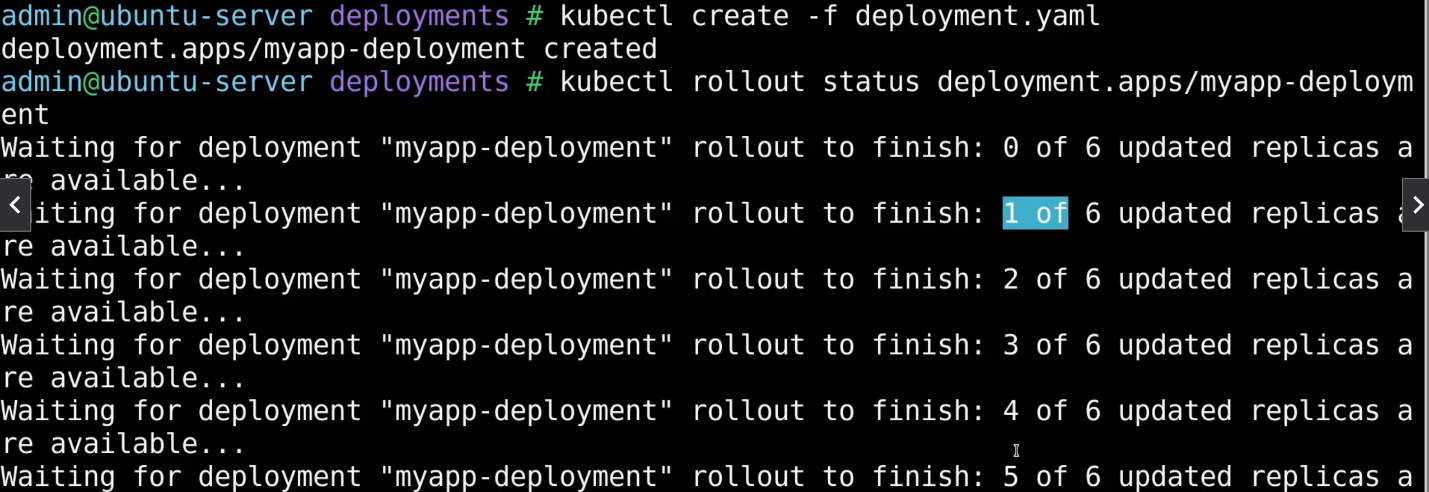
To check the status of the deployment, use the below command:



If we run this command right after running the create command of Kubectl, the rollout command will show status like below:



Until all pods are up and running, you’ll see a status message related to that. And only after all the pods have been deployed successfully, Kubernetes will consider the deployment to be successful.

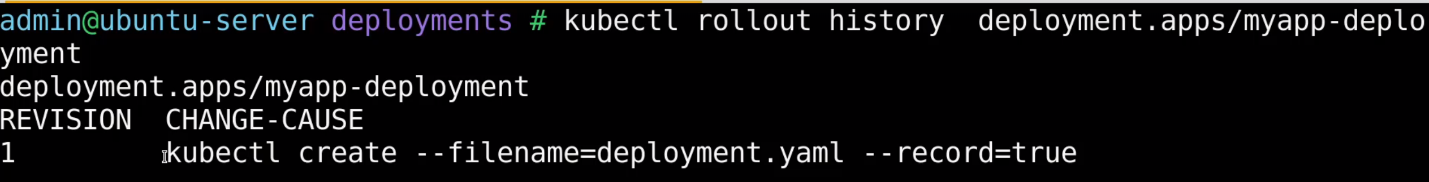
If you want to see the history of the deployment, you will use the same command except for status we will use history.



As of now, we have not entered any cause for the deployment change. To fix that use the below command:



The record option instructs Kubernetes to record the cause of change. If you now run the history command again you will see that against the revision there is a change-cause.



To change the image of the container you have to use Kubectl edit command:



This will open the container definition and make the required changes to change the image name. Once you are done with the update and then run the status command again, you will see old pods are getting terminated first and then newer ones are created as the deployment is set to rolling update.

Other way to update the image is to use the below command:



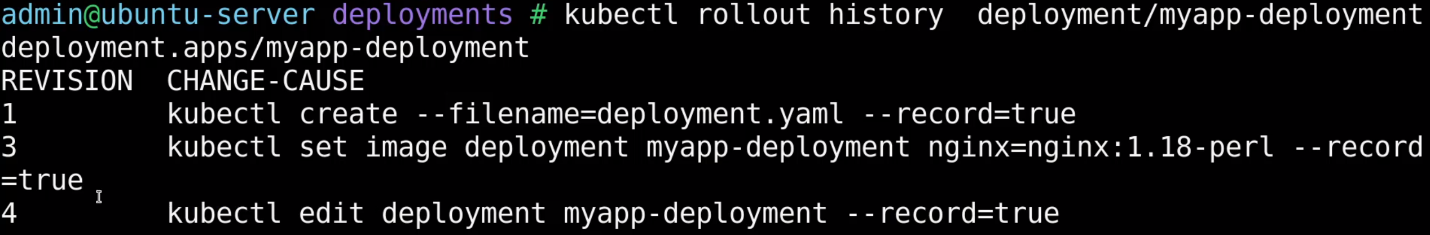
And if you run the status command again, you will see that the older pods are getting terminated while newer ones are getting created. And history will show you the change-cause as well.



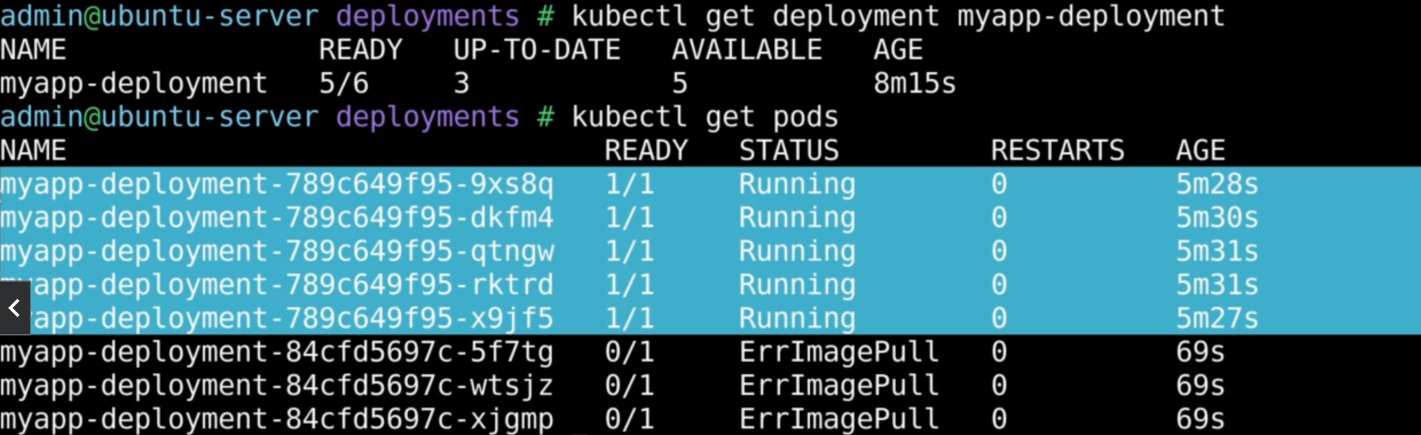
To revert back to the previous version, use the undo command:



And if you check the history now, you will see the revision number 2 is gone and we still have only 3 revisions in total. Essentially, revision 2 has become 4 (latest revision).



In another scenario, let’s say you updated the image name to that which does not exist by using the edit command and immediately after that you run the status command. You will see the output is stuck at a particular point. And if you run get deployment or get pods command you will notice only 5 containers are available out of 6.



Basically, Kubernetes tried to create 3 new pods by deleting one pod but due to the error in image the deployment was unsuccessful and got stuck. In all this the application is not impacted and it is running. If you run the undo command, it will terminate the 3 new ones and create one pod with the previous image.