Exercise 5.4: Rolling Updates and Rollbacks

We created our `simpleapp` image using **podman**, and will create an update to the container. Remember that if you are using **docker** the command syntax should be the same.

1. Make a slight change to our source and create a new image. We will use updates and rollbacks with our application. Adding a comment to the last line should be enough for a new image to be generated.

```
student@cp:~$ cd $HOME/app1

student@cp:~/app1$ vim simple.py

<output_omitted>
## Sleep for five seconds then continue the loop

  time.sleep(5)

## Adding a new comment so image is different.
```

2. View the current images, tags and how long ago the images were created.

```
student@cp:~/app1$ sudo podman images |grep simple
```

```
10.97.40.62:5000/simpleapp         latest    fd6e141c3a2e  14 hours ago   925 MB
localhost/simpleapp                  latest    fd6e141c3a2e  14 hours ago   925 MB
```

3. Build the image again. A new container and image will be created. Verify when successful. There should be a different image ID and a recent creation time for the local image.

```
student@cp:~/app1$ sudo podman build -t simpleapp .
```

```
STEP 1: FROM python:2
STEP 2: ADD simple.py /
--> c1620c253cb
STEP 3: CMD [ "python", "./simple.py" ]
STEP 4: COMMIT simpleapp
--> fd46f4115eb
fd46f4115ebc87a064c15c57b08779b02deec97ab1abe92713689342ece496bd
```

```
student@cp:~/app1$ sudo podman images | grep simple
```

```
localhost/simpleapp                  latest    fd46f4115ebc  43 seconds ago  925 MB
10.97.40.62:5000/simpleapp         latest    fd6e141c3a2e  14 hours ago    925 MB
```

4. Tag and push the updated image to your locally hosted registry. Use the tag `v2` this time at the end of the image name.

```
student@cp:~/app1$ sudo podman tag simpleapp $repo/simpleapp:v2

student@cp:~/app1$ sudo podman push $repo/simpleapp:v2
```

```
Getting image source signatures
Copying blob 46829331b1e4 skipped: already exists
Copying blob 47458fb45d99 skipped: already exists
Copying blob 461719022993 skipped: already exists
Copying blob a3c1026c6bcc skipped: already exists
```

```
Copying blob d35c5bda4793 skipped: already exists
Copying blob f1d420c2af1a skipped: already exists
Copying blob bbcb11d3fa81 done
Copying blob ceee8816bb96 skipped: already exists
Copying blob da7b0a80a4f2 skipped: already exists
Copying blob e571d2d3c73c skipped: already exists
Copying config fd46f4115e done
Writing manifest to image destination
Storing signatures
```

5. Check your images again, there should be an additional entry with same image ID, but with the tag of `v2`.

```
student@cp:~/app1$ sudo podman images |grep simple
```

```
10.97.40.62:5000/simpleapp         v2        fd46f4115ebc  3 minutes ago  925 MB
localhost/simpleapp                  latest    fd46f4115ebc  3 minutes ago  925 MB
10.97.40.62:5000/simpleapp         latest    fd6e141c3a2e  14 hours ago   925 MB
```

6. Connect to a terminal running on your worker node. Pull the image without asking for a version, which one would expect to pull the `latest` image, then pull `v2`. Note the default did not pull the new version of the image. Literally the `latest` tag has no relationship to the latest image.

```
student@worker:~$ sudo podman pull $repo/simpleapp
```

```
Trying to pull 10.97.40.62:5000/simpleapp:latest...
Getting image source signatures
Copying blob ca488b1eb9fa skipped: already exists
Copying blob 1f22e54987ac skipped: already exists
Copying blob 33be93024b52 skipped: already exists
Copying blob 41b428188c3d skipped: already exists
Copying blob 6dae3f0239bb skipped: already exists
Copying blob 637977187da0 skipped: already exists
Copying blob 3263dd1bdf84 [--------------------------------------] 0.0b / 0.0b
Copying blob 7506b856c5df [--------------------------------------] 0.0b / 0.0b
Copying blob 19aa3a14ed80 [--------------------------------------] 0.0b / 0.0b
Copying blob c45f0d38c649 [--------------------------------------] 0.0b / 0.0b
Copying config fd6e141c3a done
Writing manifest to image destination
Storing signatures
fd6e141c3a2e2d663f56c0fa7a4d056f73f6d1f63da6e36b19d7ef006dd8fff6
```

```
student@worker:~$ sudo podman pull $repo/simpleapp:v2
```

```
Trying to pull 10.97.40.62:5000/simpleapp:v2...
Getting image source signatures
Copying blob ca488b1eb9fa skipped: already exists
Copying blob 1f22e54987ac skipped: already exists
Copying blob 6dae3f0239bb skipped: already exists
Copying blob 41b428188c3d skipped: already exists
Copying blob 637977187da0 skipped: already exists
Copying blob 33be93024b52 skipped: already exists
Copying blob 3263dd1bdf84 skipped: already exists
Copying blob 7506b856c5df skipped: already exists
Copying blob c45f0d38c649 skipped: already exists
Copying blob 4da1215fbbb8 done
Copying config fd46f4115e done
Writing manifest to image destination
Storing signatures
fd46f4115ebc87a064c15c57b08779b02deec97ab1abe92713689342ece496bd
```

7. Return to your cp node, use **kubectl edit** to update the image for the `try1` deployment to use `v2`. As we are only

changing one parameter we could also use the **kubectl set** command. Note that the configuration file has not been updated, so a delete or a replace command would not include the new version. It can take the pods up to a minute to delete and to recreate each pod in sequence.

```
student@cp:~/app1$ kubectl edit deployment try1 --record=true
```

```
....
      containers:
      - image: 10.97.40.62:5000/simpleapp:v2   #<-- Edit tag
        imagePullPolicy: Always
....
```

8. Verify each of the pods has been recreated and is using the new version of the image. Note some messages will show the scaling down of the old **replicaset**, others should show the scaling up using the new image.

```
student@cp:~/app1$ kubectl get events
```

```
42m         Normal    ScalingReplicaSet   Deployment   Scaled up replica set try1-7fdbb5d557 to 6
32s         Normal    ScalingReplicaSet   Deployment   Scaled up replica set try1-7fd7459fc6 to 2
32s         Normal    ScalingReplicaSet   Deployment   Scaled down replica set try1-7fdbb5d557 to
↪  5
32s         Normal    ScalingReplicaSet   Deployment   Scaled up replica set try1-7fd7459fc6 to 3
23s         Normal    ScalingReplicaSet   Deployment   Scaled down replica set try1-7fdbb5d557 to
↪  4
23s         Normal    ScalingReplicaSet   Deployment   Scaled up replica set try1-7fd7459fc6 to 4
22s         Normal    ScalingReplicaSet   Deployment   Scaled down replica set try1-7fdbb5d557 to
↪  3
22s         Normal    ScalingReplicaSet   Deployment   Scaled up replica set try1-7fd7459fc6 to 5
18s         Normal    ScalingReplicaSet   Deployment   Scaled down replica set try1-7fdbb5d557 to
↪  2
18s         Normal    ScalingReplicaSet   Deployment   Scaled up replica set try1-7fd7459fc6 to 6
8s          Normal    ScalingReplicaSet   Deployment   (combined from similar events):
Scaled down replica set try1-7fdbb5d557 to 0
```

9. View the images of a Pod in the deployment. Narrow the output to just view the images. The `goproxy` remains unchanged, but the `simpleapp` should now be `v2`.

```
student@cp:~/app1$ kubectl describe pod try1-895fccfb-ttqdn |grep Image
```

```
    Image:          10.97.40.62:5000/simpleapp:v2
    Image ID:       10.97.40.62:5000/simpleapp@sha256:c3848fd3b5cdf5f241ceb4aa7e96c5c2be9f09
f23eea1afe946f8507f3fcbc29
    Image:          registry.k8s.io/goproxy:0.1
    Image ID:       docker.io/library/goproxy@sha256:5334c7ad43048e3538775c
b09aaf184f5e8c
c8f1d93c209865c8f1d93c2098
```

10. View the update history of the deployment.

```
student@cp:~/app1$ kubectl rollout history deployment try1
```

```
deployment.apps/try1
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
```

11. Compare the output of the **rollout history** for the two revisions. Images and labels should be different, with the image `v2` being the change we made.

```
student@cp:~/app1$ kubectl rollout history deployment try1 --revision=1 > one.out

student@cp:~/app1$ kubectl rollout history deployment try1 --revision=2 > two.out
```

```
student@cp:~/app1:$ diff one.out two.out
```

```
1c1
< deployment.apps/try1 with revision #1
---
> deployment.apps/try1 with revision #2
4c4
<           pod-template-hash=1509661973
---
>           pod-template-hash=45197796
7c7
<     Image:        10.97.40.62:5000/simpleapp
---
>     Image:        10.97.40.62:5000/simpleapp:v2
```

12. View what would be undone using the **–dry-run** option while undoing the rollout. This allows us to see the new template prior to using it.

```
student@cp:~/app1$ kubectl rollout undo --dry-run=client deployment/try1
```

```
deployment.apps/try1 Pod Template:
  Labels:       app=try1
        pod-template-hash=754bf9c75b
  Containers:
   simpleapp:
    Image:        10.97.40.62:5000/simpleapp
    Port:         <none>
    Host Port:        <none>
<output_omitted>
```

13. View the pods. Depending on how fast you type the `try1` pods should be about 2 minutes old.

```
student@cp:~/app1$ kubectl get pods
```

```
NAME                          READY     STATUS    RESTARTS   AGE
nginx-6b58d9cdfd-9fnl4        1/1       Running   1          5d
registry-795c6c8b8f-hl5wf     1/1       Running   2          5d
try1-594fbb5fc7-7dl7c         2/2       Running   0          2m
try1-594fbb5fc7-8mxlb         2/2       Running   0          2m
try1-594fbb5fc7-jr7h7         2/2       Running   0          2m
try1-594fbb5fc7-s24wt         2/2       Running   0          2m
try1-594fbb5fc7-xfffg         2/2       Running   0          2m
try1-594fbb5fc7-zfmz8         2/2       Running   0          2m
```

14. In our case there are only two revisions, which is also the default number kept. Were there more we could choose a particular version. The following command would have the same effect as the previous, without the **–dry-run** option.

```
student@cp:~/app1$ kubectl rollout undo deployment try1 --to-revision=1
```

```
deployment.apps/try1 rolled back
```

15. Again, it can take a bit for the pods to be terminated and re-created. Keep checking back until they are all running again.

```
student@cp:~/app1$ kubectl get pods
```

```
NAME                          READY     STATUS        RESTARTS   AGE
nginx-6b58d9cdfd-9fnl4        1/1       Running       1          5d
registry-795c6c8b8f-hl5wf     1/1       Running       2          5d
try1-594fbb5fc7-7dl7c         2/2       Terminating   0          3m
try1-594fbb5fc7-8mxlb         0/2       Terminating   0          2m
try1-594fbb5fc7-jr7h7         2/2       Terminating   0          3m
```

```
try1-594fbb5fc7-s24wt       2/2       Terminating   0          2m
try1-594fbb5fc7-xfffg       2/2       Terminating   0          3m
try1-594fbb5fc7-zfmz8       1/2       Terminating   0          2m
try1-895fccfb-8dn4b         2/2       Running       0          22s
try1-895fccfb-kz72j         2/2       Running       0          10s
try1-895fccfb-rxxtw         2/2       Running       0          24s
try1-895fccfb-srwq4         1/2       Running       0          11s
try1-895fccfb-vkvmb         2/2       Running       0          31s
try1-895fccfb-z46qr         2/2       Running       0          31s
```