Exercise 5.4: Rolling Updates and Rollbacks

When we started working with `simpleapp` we used a **Docker** tag called `latest`. While this is the default tag when pulling an image, and commonly used, it remains just a string, it may not be the actual latest version of the image.

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@master:~$ cd ~/app1

student@master:~/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. 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.

```
student@master:~/app1$ sudo docker build -t simpleapp .
```

```
Sending build context to Docker daemon 7.168 kB
Step 1/3 : FROM python:2
 ---> 2863c80c418c
Step 2/3 : ADD simple.py /
 ---> cde8ecf8492b
Removing intermediate container 3e908b76b5b4
Step 3/3 : CMD python ./simple.py
 ---> Running in 354620c97bf5
 ---> cc6bba0ea213
Removing intermediate container 354620c97bf5
Successfully built cc6bba0ea213
```

```
student@master:~/app1$ sudo docker images
```

```
REPOSITORY                            TAG
 IMAGE ID            CREATED       SIZE
simpleapp                             latest
 cc6bba0ea213    8 seconds ago     886 MB
10.105.119.236:5000/simpleapp         latest
15b5ad19d313     4 days ago        886 MB
<output_omitted>
```

3. Tag and push the updated image to your locally hosted registry. A reminder your IP address will be different than the example below. Use the tag `v2` this time instead of `latest`.

```
student@master:~/app1$ sudo docker tag simpleapp \
   10.105.119.236:5000/simpleapp:v2

student@master:~/app1$ sudo docker push 10.105.119.236:5000/simpleapp:v2
```

```
The push refers to a repository [10.105.119.236:5000/simpleapp]
d6153c8cc7c3: Pushed
ca82a2274c57: Layer already exists
de2fbb43bd2a: Layer already exists
4e32c2de91a6: Layer already exists
6e1b48dc2ccc: Layer already exists
ff57bdb79ac8: Layer already exists
6e5e20cbf4a7: Layer already exists
86985c679800: Layer already exists
8fad67424c4e: Layer already exists
v2: digest: sha256:6cf74051d09463d89f1531fceb9c44cbf99006f8d9b407
dd91d8f07baeee7e9c size: 2218
```

4. Connect to a terminal running on your second node. Pull the `latest` image, then pull `v2`. Note the latest did not pull the new version of the image. Again, remember to use the IP for your locally hosted registry. You'll note the digest is different.

```
student@worker:~$ sudo docker pull 10.105.119.236:5000/simpleapp
```

```
Using default tag: latest
latest: Pulling from simpleapp
Digest: sha256:cefa3305c36101d32399baf0919d3482ae8a53c926688be33
86f9bbc04e490a5
Status: Image is up to date for 10.105.119.236:5000/simpleapp:latest
```

```
student@worker:~$ sudo docker pull 10.105.119.236:5000/simpleapp:v2
```

```
v2: Pulling from simpleapp
f65523718fc5: Already exists
1d2dd88bf649: Already exists
c09558828658: Already exists
0e1d7c9e6c06: Already exists
c6b6fe164861: Already exists
45097146116f: Already exists
f21f8abae4c4: Already exists
1c39556edcd0: Already exists
fa67749bf47d: Pull complete
Digest: sha256:6cf74051d09463d89f1531fceb9c44cbf99006f8d9b407dd91d8
f07baeee7e9c
Status: Downloaded newer image for 10.105.119.236:5000/simpleapp:v2
```

5. 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@master:~/app1$ kubectl edit deployment try1
```

```
....
      containers:
      - image: 10.105.119.236:5000/simpleapp:v2   #<-- Edit tag
        imagePullPolicy: Always
....
```

6. 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@master:~/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

```

7. 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@master:~/app1$ kubectl describe pod try1-895fccfb-ttqdn |grep Image
```

```
    Image:          10.105.119.236:5000/simpleapp:v2
    Image ID:\
       docker-pullable://10.105.119.236:5000/simpleapp@sha256:6cf74051d09
463d89f1531fceb9c44cbf99006f8d9b407dd91d8f07baeee7e9c
    Image:          k8s.gcr.io/goproxy:0.1
    Image ID:\
       docker-pullable://k8s.gcr.io/goproxy@sha256:5334c7ad43048e3538775c
b09aaf184f5e8acf4b0ea60e3bc8f1d93c209865a5
```

8. View the update history of the deployment.

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

```
deployments "try1"
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
```

9. 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@master:~/app1$ kubectl rollout history deployment try1 --revision=1 > one.out

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

student@ckad-/app11:~$ diff one.out two.out
```

```
1c1
< deployments "try1" with revision #1
---
> deployments "try1" with revision #2
3c3
<   Labels:        pod-template-hash=1509661973
---
>   Labels:        pod-template-hash=45197796
7c7
<     Image:        10.105.119.236:5000/simpleapp
---
>     Image:        10.105.119.236:5000/simpleapp:v2
```

10. 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@master:~/app1$ kubectl rollout undo --dry-run=client deployment/try1
```

```
deployment.apps/try1
Pod Template:
  Labels:        pod-template-hash=1509661973
        run=try1
  Containers:
   try1:
    Image:        10.105.119.236:5000/simpleapp:latest
    Port:        <none>
<output_omitted>
```

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

```
student@master:~/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
```

12. 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@master:~/app1$ kubectl rollout undo deployment try1 --to-revision=1
```

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

13. 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@master:~/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
```