Guided Exercise: Container Image Identity and Tags

Update an application by changing its deployment to reference a newer image tag, and find the hashes of the old and new application images.

Outcomes

Inspect container images, list images of containers that run on compute nodes, and deploy applications by using image tags or SHA IDs.

As the `student` user on the `workstation` machine, use the `lab` command to prepare your system for this exercise.

This command ensures that all resources are available for this exercise. It also creates the `updates-ids` project and the `/home/student/DO180/labs/updates-ids/resources.txt` file. The `resources.txt` file contains the name of the images and some commands that you use during the exercise. You can use the file to copy and paste these image names and commands.

```
[student@workstation ~]$ lab start updates-ids
```

Instructions

1. Log in to the OpenShift cluster as the `developer` user with `developer` as the password. Use the `updates-ids` project.

   Log in to the OpenShift cluster.

   ```
   [student@workstation ~]$ oc login -u developer -p developer \
     https://api.ocp4.example.com:6443
   Login successful.
   ...output omitted...
   ```

   Set the `updates-ids` project as the active project.

   ```
   [student@workstation ~]$ oc project updates-ids
   ...output omitted...
   ```

2. Inspect the two versions of the `registry.ocp4.example.com:8443/ubi8/httpd-24` image from the classroom container registry. The classroom setup copied that image from the Red Hat Ecosystem Catalog. The original image is `registry.access.redhat.com/ubi8/httpd-24`.

   Use the `oc image info` command to inspect the image version that the `1-209` tag references. Notice the unique SHA ID that identifies the image version.

   > **NOTE**
   >
   > To improve readability, the instructions truncate the SHA-256 strings.
   >
   > On your system, the commands return the full SHA-256 strings. Also, you must type the full SHA-256 string, to provide such a parameter to a command.

   ```
   [student@workstation ~]$ oc image info \
     registry.ocp4.example.com:8443/ubi8/httpd-24:1-209 | head
   Name:          registry.ocp4.example.com:8443/ubi8/httpd-24:1-209
   Digest:        sha256:b1e3...f876
   ...output omitted...
   ```

   Inspect the image version that the `1-215` tag references. Notice that the SHA ID, or digest, differs from the preceding image version.

   ```
   [student@workstation ~]$ oc image info \
     registry.ocp4.example.com:8443/ubi8/httpd-24:1-215 | head
   Name:          registry.ocp4.example.com:8443/ubi8/httpd-24:1-215
   Digest:        sha256:91ad...fd83
   ...output omitted...
   ```

   Use the `skopeo inspect` command to inspect images. The output format differs from the `oc image info` command, although both commands report similar data.

   Use the `skopeo login` command to log in to the registry as the `developer` user with `developer` as the password.

```
[student@workstation ~]$ skopeo login registry.ocp4.example.com:8443 -u developer
Password: developer
Login Succeeded!
```

Use the `skopeo inspect` command to inspect the `1-215` image tag.

```
[student@workstation ~]$ skopeo inspect \
  docker://registry.ocp4.example.com:8443/ubi8/httpd-24:1-215 | head
{
    "Name": "registry.ocp4.example.com:8443/ubi8/httpd-24",
    "Digest": "sha256:91ad...fd83",
    "RepoTags": [
        "1-209",
        "1-215",
        "latest"
    ],
...output omitted...
}
```

The `skopeo inspect` command also shows other existing image tags.

3. Deploy an application from the image version that the `1-209` tag references.

   Use the `oc create deployment` command to deploy the application. Set the name of the deployment to `httpd1`.

```
[student@workstation ~]$ oc create deployment httpd1 \
  --image registry.ocp4.example.com:8443/ubi8/httpd-24:1-209
deployment.apps/httpd1 created
```

   Wait for the pod to start, and then retrieve the name of the cluster node that runs it. You might have to rerun the command several times for the pod to report the `Running` status. The name of the pod on your system probably differs.

```
[student@workstation ~]$ oc get pods -o wide
NAME                     READY  STATUS   RESTARTS  AGE  IP          NODE     ...
httpd1-6dff796d99-pm2x6  1/1    Running  0         19s  10.8.0.104  master01 ...
```

   Retrieve the name of the container that is running inside the pod. The `crictl ps` command that you run in a following step takes the container name as an argument.

```
[student@workstation ~]$ oc get deployment httpd1 -o wide
NAME     READY   UP-TO-DATE   AVAILABLE   AGE     CONTAINERS  ...
httpd1   1/1     1            1           1m10s   httpd-24    ...
```

4. Access the cluster node and retrieve the image that the container is using.

   Log in to the OpenShift cluster as the `admin` user with `redhatocp` as the password.

```
[student@workstation ~]$ oc login -u admin -p redhatocp
Login successful.
...output omitted...
```

   Use the `oc debug node` command to access the cluster node.

```
[student@workstation ~]$ oc debug node/master01
Temporary namespace openshift-debug-flz4d is created for debugging node...
Starting pod/master01-debug-w9cqk ...
To use host binaries, run `chroot /host`
Pod IP: 192.168.50.10
If you don't see a command prompt, try pressing enter.
sh-5.1#
```

   In the remote shell, run the `chroot /host` command to use host binaries.

```
sh-5.1# chroot /host
sh-5.1#
```

   Use the `crictl ps` command to confirm that the `httpd-24` container is running. Add the `-o yaml` option to display the container details in YAML format.

```
sh-4.4# crictl ps --name httpd-24 -o yaml
containers:
- annotations:
...output omitted...
  image:
    annotations: {}
    image: registry.ocp4.example.com:8443/ubi8/httpd-24@sha256:b1e3...f876
...output omitted...
  imageRef: registry.ocp4.example.com:8443/ubi8/httpd-24@sha256:b1e3...f876
  labels:
...output omitted...
  state: CONTAINER_RUNNING
```

Notice that the command refers to the image by its SHA ID, and not by the tag that you specified when you created the deployment resource.

Use the `crictl images` command to list the locally available images on the node.
The `registry.ocp4.example.com:8443/ubi8/httpd-24:1-209` image is in that list, because the local container engine pulled it when you deployed the `httpd1` application.

> **NOTE**
>
> The `IMAGE ID` column displays the local image identifier that the container engine assigns to the image. This identifier is not related to the SHA image ID that the container registry assigned to the image.
>
> Most `crictl` commands, such as `crictl images` or `crictl rmi`, accept a local image identifier instead of the full image name. For example, you can run the `crictl images 8ee59251acc93` command as a short version of the `crictl images registry.ocp4.example.com:8443/ubi8/httpd-24:1-209` command.

```
sh-5.1# crictl images
IMAGE                                     TAG     IMAGE ID       SIZE
...output omitted...
quay.io/openshift-release-dev/ocp-release       <none>  d29424caa281a      517MB
quay.io/openshift-release-dev/ocp-v4.0-art-dev  <none>  42a50bb73e2fb      588MB
quay.io/openshift-release-dev/ocp-v4.0-art-dev  <none>  df18dddad77e7      510MB
quay.io/openshift-release-dev/ocp-v4.0-art-dev  <none>  388587ea693e1      447MB
...output omitted...
registry.ocp4.example.com:8443/ubi8/httpd-24    1-209   8ee59251acc93      461MB
...output omitted...
```

The `crictl images` command does not display the SHA image IDs by default. Rerun the command and add the `--digests` option to display the SHA IDs. Also, add the local image ID to the command to limit the output to the `registry.ocp4.example.com:8443/ubi8/httpd-24:1-209` image.

The command reports only the first characters of the SHA image ID. These characters match the SHA ID of the image that the `httpd-24` container is using. Therefore, the `httpd-24` container is using the expected image.

```
sh-5.1# crictl images --digests 8ee59251acc93
IMAGE                                     TAG   DIGEST        IMAGE ID      ...
registry.ocp4.example.com:8443/ubi8/httpd-24 1-209 b1e3c572516d1 8ee59251acc93 ...
```

Disconnect from the cluster node.

```
sh-5.1# exit
exit
sh-5.1# exit
exit

Removing debug pod ...
Temporary namespace openshift-debug-flz4d was removed.
[student@workstation ~]$
```

5. Log in as the `developer` user and deploy another application by using the SHA ID of the image as the digest.

   Log in to the OpenShift cluster as the `developer` user with `developer` as the password.

```
[student@workstation ~]$ oc login -u developer -p developer
Login successful.
...output omitted...
```

Use the `oc image info` command to retrieve the SHA ID of the image version that the `1-209` tag references. Specify the JSON format for the command output. Parse the JSON output with the `jq -r` command to retrieve the value of the `.digest` object.

```
[student@workstation ~]$ oc image info \
  registry.ocp4.example.com:8443/ubi8/httpd-24:1-209 -o json | \
  jq -r .digest
sha256:b1e3...f876
```

Export the SHA ID as the `$IMAGE` environment variable.

```
[student@workstation ~]$ IMAGE=sha256:b1e3...f876
```

Use the `oc create deployment` command to deploy the application. Set the name of the deployment to `httpd2`.

```
[student@workstation ~]$ oc create deployment httpd2 \
  --image registry.ocp4.example.com:8443/ubi8/httpd-24@$IMAGE
deployment.apps/httpd2 created
```

Confirm that the new deployment refers to the image version by its SHA ID.

```
[student@workstation ~]$ oc get deployment httpd2 -o wide
NAME     READY  ...  CONTAINERS   IMAGES ...
httpd2   1/1    ...  httpd-24     registry.../ubi8/httpd-24@sha256:b1e3...f876 ...
```

6. Update the `httpd2` application by using a more recent image version.

   In the `httpd2` deployment, update the `httpd-24` container to use the image version that the `1-215` tag references.

```
[student@workstation ~]$ oc set image deployment/httpd2 \
  httpd-24=registry.ocp4.example.com:8443/ubi8/httpd-24:1-215
deployment.apps/httpd2 image updated
```

   Confirm that the deployment refers to the new image version.

```
[student@workstation ~]$ oc get deployment httpd2 -o wide
NAME     READY   ...    IMAGES ...
httpd2   1/1     ...    registry.ocp4.example.com:8443/ubi8/httpd-24:1-215 ...
```

   Confirm that the deployment finished redeploying the pod. You might have to rerun the command several times for the pod to report a `Running` status. The pod names probably differ on your system.

```
[student@workstation ~]$ oc get pods
NAME                     READY   STATUS    RESTARTS   AGE
httpd1-6dff796d99-pm2x6  1/1     Running   0          118m
httpd2-998d9b9b9-5859j   1/1     Running   0          21s
```

   Inspect the pod to confirm that the container is using the new image. Replace the pod name with your own pod from the previous step.

```
[student@workstation ~]$ oc get pod httpd2-998d9b9b9-5859j \
  -o jsonpath='{.spec.containers[0].image}{"\n"}'
registry.ocp4.example.com:8443/ubi8/httpd-24:1-215
```

7. Add the `latest` tag to the image version that the `1-209` tag already references. Deploy an application from the image with the `latest` tag.

   Use the `skopeo login` command to log in to the classroom container registry as the `developer` user with `developer` as the password.

```
[student@workstation ~]$ skopeo login -u developer -p developer \
  registry.ocp4.example.com:8443
Login Succeeded!
```

   Use the `skopeo copy` command to add the `latest` tag to the image.

```
[student@workstation ~]$ skopeo copy \
  docker://registry.ocp4.example.com:8443/ubi8/httpd-24:1-209 \
  docker://registry.ocp4.example.com:8443/ubi8/httpd-24:latest
Getting image source signatures
...output omitted...
Writing manifest to image destination
```

Use the `oc image info` command to confirm that both tags refer to the same image. The two commands report the same SHA image ID, which indicates that the tags point to the same image version.

```
[student@workstation ~]$ oc image info \
  registry.ocp4.example.com:8443/ubi8/httpd-24:1-209 | head
Name:          registry.ocp4.example.com:8443/ubi8/httpd-24:1-209
Digest:        sha256:b1e3...f876
...output omitted...
```

```
[student@workstation ~]$ oc image info \
  registry.ocp4.example.com:8443/ubi8/httpd-24:latest | head
Name:          registry.ocp4.example.com:8443/ubi8/httpd-24:latest
Digest:        sha256:b1e3...f876
...output omitted...
```

Use the `oc create deployment` command to deploy another application. Set the name of the deployment to `httpd3`. To confirm that by default the command selects the `latest` tag, do not provide the tag part in the image name.

```
[student@workstation ~]$ oc create deployment httpd3 \
  --image registry.ocp4.example.com:8443/ubi8/httpd-24
deployment.apps/httpd3 created
```

Confirm that the pod is running. You might have to rerun the command several times for the pod to report the `Running` status. The pod names probably differ on your system.

```
[student@workstation ~]$ oc get pods
NAME                     READY   STATUS    RESTARTS   AGE
httpd1-6dff796d99-pm2x6  1/1     Running   0          150m
httpd2-998d9b9b9-5859j   1/1     Running   0          32m
httpd3-85b978d758-fvqdr  1/1     Running   0          42s
```

Confirm that the pod is using the expected image. Notice that the SHA image ID corresponds to the image that the `1-209` tag references. You retrieved that SHA image ID in a preceding step when you ran the `oc image info` command.

```
[student@workstation ~]$ oc describe pod httpd3-85b978d758-fvqdr | \
  grep ID
    Container ID: cri-o://baa3...7c90
    Image ID:     registry.ocp4.example.com:8443/ubi8/httpd-24@sha256:b1e3...f876
```

8. Assign the `latest` tag to a different image version. This operation simulates a developer who pushes a new version of an image and assigns the `latest` tag to that new image version.

   Use the `skopeo copy` command to add the `latest` tag to the image version that the `1-215` tag already references. The command automatically removes the `latest` tag from the earlier image.

```
[student@workstation ~]$ skopeo copy \
  docker://registry.ocp4.example.com:8443/ubi8/httpd-24:1-215 \
  docker://registry.ocp4.example.com:8443/ubi8/httpd-24:latest
Getting image source signatures
...output omitted...
Writing manifest to image destination
Storing signatures
```

   Log out of the classroom container registry.

```
[student@workstation ~]$ skopeo logout registry.ocp4.example.com:8443
Removed login credentials for registry.ocp4.example.com:8443
```

> **NOTE**
>
> The `skopeo logout` command logs out of a specified registry server by deleting the cached credentials in the `${XDG_RUNTIME_DIR}/containers/auth.json` file.

Red Hat recommends removing cached credentials that are no longer required.

Even though the `latest` tag is now referencing a different image version, OpenShift does not redeploy the pods that are running with the previous image version.

Rerun the `oc describe pod` command to confirm that the pod still uses the preceding image.

```
[student@workstation ~]$ oc describe pod httpd3-85b978d758-fvqdr | \
  grep ID
    Container ID: cri-o://2cee...3a68
    Image ID:     registry.ocp4.example.com:8443/ubi8/httpd-24@sha256:b1e3...f876
```

9. Scale the `httpd3` deployment to two pods.

   Use the `oc scale` command to add a new pod to the deployment.

```
[student@workstation ~]$ oc scale deployment/httpd3 --replicas 2
deployment.apps/httpd3 scaled
```

   List the pods to confirm that two pods are running for the `httpd3` deployment. The pod names probably differ on your system.

```
[student@workstation ~]$ oc get pods
httpd1-6dff796d99-pm2x6   1/1     Running   0          75m
httpd2-998d9b9b9-5859j    1/1     Running   0          30m
httpd3-85b978d758-f98jh   1/1     Running   0          54s
httpd3-85b978d758-fvqdr   1/1     Running   0          11m
```

   Retrieve the SHA image ID for the pod that the deployment initially created. The ID did not change. The container is still using the original image version.

```
[student@workstation ~]$ oc describe pod httpd3-85b978d758-fvqdr | \
  grep ID
    Container ID: cri-o://2cee...3a68
    Image ID:     registry.ocp4.example.com:8443/ubi8/httpd-24@sha256:b1e3...f876
```

   Retrieve the SHA image ID for the additional pod. Notice that the ID is different. The additional pod is using the image that the `latest` tag is currently referencing.

```
[student@workstation ~]$ oc describe pod httpd3-85b978d758-f98jh | \
  grep ID
    Container ID: cri-o://d254...c893
    Image ID:     registry.ocp4.example.com:8443/ubi8/httpd-24@sha256:91ad...fd83
```

   The state of the deployment is inconsistent. The two replicated pods use a different image version. Consequently, the scaled application might not behave correctly. Red Hat recommends that you use a less volatile tag than `latest` in production environments, or that you tightly control the tag assignments in your container registry.

Finish

On the `workstation` machine, use the `lab` command to complete this exercise. This step is important to ensure that resources from previous exercises do not impact upcoming exercises.

```
[student@workstation ~]$ lab finish updates-ids
```