Guided Exercise: Reproducible Deployments with OpenShift Image Streams

Deploy an application that references container images indirectly by using image streams.

Outcomes

You should be able to create image streams and image stream tags, and deploy applications that use image stream tags.

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-imagestreams` project and the `/home/student/DO180/labs/updates-imagestreams/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-imagestreams
```

Instructions

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

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

2. Create the `versioned-hello` image stream and the `v1.0` image stream tag from the `registry.ocp4.example.com:8443/redhattraining/versioned-hello:v1.0` image.

   Use the `oc create is` command to create the image stream.

   ```
   [student@workstation ~]$ oc create is versioned-hello
   imagestream.image.openshift.io/versioned-hello created
   ```

   Use the `oc tag` command to create the image stream tag.

   ```
   [student@workstation ~]$ oc tag \
     registry.ocp4.example.com:8443/redhattraining/versioned-hello:v1.0 \
     versioned-hello:v1.0
   Tag versioned-hello:v1.0 set to registry.ocp4.example.com:8443/redhattraining/versioned-hello:v1.0.
   ```

3. Enable image stream resolution for the `versioned-hello` image stream so that Kubernetes resources in the current project can use it.

   Use the `oc set image-lookup` command to enable image lookup resolution.

   ```
   [student@workstation ~]$ oc set image-lookup versioned-hello
   imagestream.image.openshift.io/versioned-hello image lookup updated
   ```

   Run the `oc set image-lookup` command without any arguments to verify your work.

   ```
   [student@workstation ~]$ oc set image-lookup
   NAME             LOCAL
   versioned-hello  true
   ```

4. Review the image stream and confirm that the image stream tag refers to the source image by its SHA ID. Verify that the source image in the `registry.ocp4.example.com:8443` registry has the same SHA ID.

   Retrieve the details of the `versioned-hello` image stream.

NOTE

To improve readability, the instructions truncate the SHA-256 strings.

```
[student@workstation ~]$ oc describe is versioned-hello
Name:            versioned-hello
Namespace:       updates-imagestreams
Created:         7 minutes ago
...output omitted...

v1.0
  tagged from registry.ocp4.example.com:8443/redhattraining/versioned-hello:v1.0

  * registry.ocp4.example.com:8443/.../versioned-hello@sha256:66e0...105e
      7 minutes ago
```

Use the `oc image info` command to query the image from the classroom container registry. The SHA image ID is the same as the one from the image stream tag.

```
[student@workstation ~]$ oc image info \
  registry.ocp4.example.com:8443/redhattraining/versioned-hello:v1.0 | head
Name:           registry.ocp4.example.com:8443/redhattraining/versioned-hello:v1.0
Digest:         sha256:66e0...105e
Media Type:     application/vnd.docker.distribution.manifest.v2+json
...output omitted...
```

5. Create a deployment named `version` that uses the `versioned-hello:v1.0` image stream tag.

   Use the `oc create deployment` command to create the object.

```
[student@workstation ~]$ oc create deployment version --image versioned-hello:v1.0
deployment.apps/version created
```

   Wait for the pod to start. You might have to rerun the command several times for the pod to report a `Running` status. The name of the pod on your system probably differs.

```
[student@workstation ~]$ oc get pods
NAME                        READY   STATUS    RESTARTS   AGE
version-744bf7694b-bzhd2    1/1     Running   0          2m11s
```

6. Confirm that both the deployment and the pod refer to the image by its SHA ID.

   Retrieve the image that the deployment uses. The deployment refers to the image from the source registry by its SHA ID. The `v1.0` image stream tag also points to that SHA image ID.

```
[student@workstation ~]$ oc get deployment -o wide
... IMAGES ...
... registry.ocp4.example.com:8443/.../versioned-hello@sha256:66e0...105e ...
```

   Retrieve the image that the pod is using. The pod is also referring to the image by its SHA ID.

```
[student@workstation ~]$ oc get pod version-744bf7694b-bzhd2 \
  -o jsonpath='{.spec.containers[0].image}{"\n"}'
registry.ocp4.example.com:8443/redhattraining/versioned-hello@sha256:66e0...105e
```

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-imagestreams
```