Lesson 3 : Container Image Identity and Tags

- Ensure reproducibility of application deployments by using image streams and short image names.

Image Streams (IS)

- Provide stable, short name to reference to container image stored in
  - internal openshift registry or external registry server
- Native to OpenShift
- K8s resources reference container images directly
- OpenShift resources reference Image Streams
- Reproducible, stable deployments, rollbacks deployment to latest known-good state

Example 1: use of Image Streams

- Developer deploy application using --images to external registry server
- OpenShift download the image to internal registry
- OpenShift create image stream refers to the local image
- Another developer then can use the image stream

```
Developer 1
$ oc new-project demo1
$ oc new-app --name dev1-app1 \
--image quay.io/jason_wong76/webserver
$ oc get is
webserver image-registry.openshift-image..webserver


Developer 2
$ oc new-project demo2
$ oc new-app  --name dev2-app1 \
-i demo1/webserver
```

Example 2:

- Alternative image with better security configuration exists
- Now you want to change the direction point to the alternative

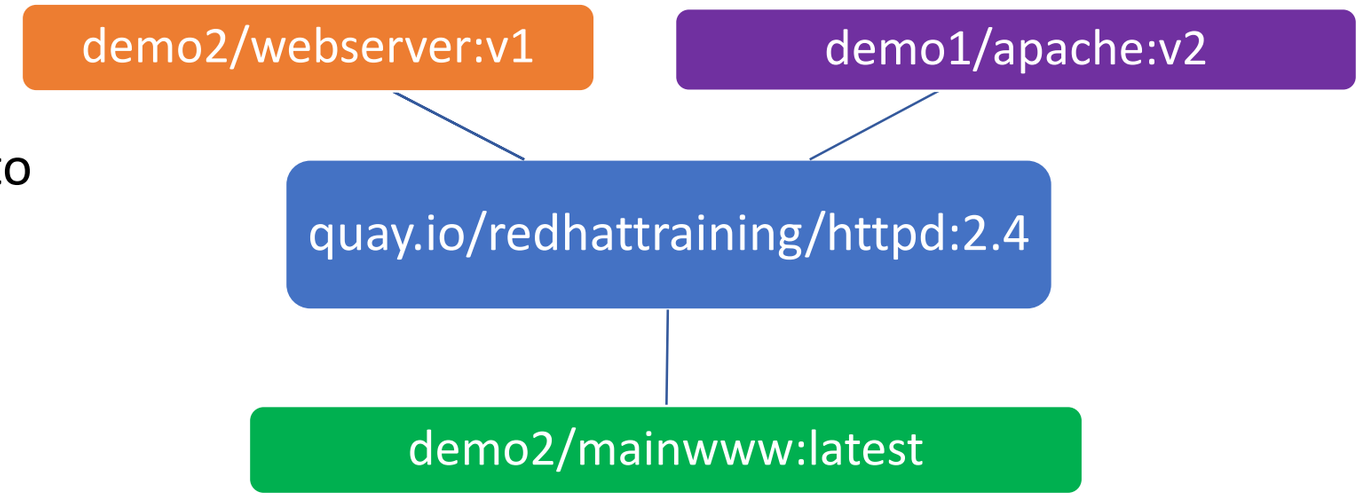
- oc import-image webserver –n demo1 \

--confirm –from quay.io/redhattraining/httpd-24

- Rebuild or redeploy application but still pointing to same image stream name

Image Stream Tags

- Is named pointer to an image in an image stream (local or external image)
- Can use abbreviated – istag
- Multiple tag reference to same image
- Image stream provides default configuration to tags
  - Image tags can overwrite configurations
- Tag stores copy of metadata about its current image. Allow faster search and inspection of image



Explore Image Streams from openshift namespace

- List all image streams : only showing name field

```
[user@host ~]$ oc get is -n openshift -o name
...output omitted...
imagestream.image.openshift.io/nodejs
imagestream.image.openshift.io/perl
imagestream.image.openshift.io/php
imagestream.image.openshift.io/postgresql
imagestream.image.openshift.io/python
...output omitted...
```

- List all istags for php image stream

```
[user@host ~]$ oc get istag -n openshift | grep php
8.0-ubi9      image-registry ...    6 days ago
8.0-ubi8      image-registry ...    6 days ago
7.4-ubi8      image-registry ...    6 days ago
7.3-ubi7      image-registry ...    6 days ago
```

Explore Image Streams from openshift namespace

- Get more information from both image stream and tags

```
[user@host ~]$ oc describe is php -n openshift
Name:                   php
Namespace:              openshift
...output omitted...
Tags:                   5

8.0-ubi9
  tagged from registry.access.redhat.com/ubi9/php-80:latest
...output omitted...

8.0-ubi8 (latest)
  tagged from registry.access.redhat.com/ubi8/php-80:latest
...output omitted...

7.4-ubi8
  tagged from registry.access.redhat.com/ubi8/php-74:latest
```

Image Names, Tags, and IDs

- Name of image is in following format

  registry-host-name/repository-or-organization-or-user-name/image-name:tag-name

- SHA image ID is immutable unique identifiers for image in SHA-256 hash value
- OpenShift does not monitor external registries for changes
- Can schedule image stream tag to check external registry for updates

Image Names, Tags, and IDs

```
[user@host ~]$ oc describe is php -n openshift
Name:                   php
Namespace:              openshift
...output omitted...

8.0-ubi9
  tagged from registry.access.redhat.com/ubi9/php-80:latest
...output omitted...
  * registry.access.redhat.com/ubi9/php-80@sha256:2b82...f544
      2 days ago
    registry.access.redhat.com/ubi9/php-80@sha256:8840...94f0
      5 days ago
    registry.access.redhat.com/ubi9/php-80@sha256:506c...5d90
      9 days ago
```

* shows current one for each image stream tag

- Verify and compares SHA Image ID with external registry

Create image stream tag to external registries

- Try avoid the latest tag

```
[user@host ~]$ oc create istag keycloak:20.0 \
  --from-image quay.io/keycloak/keycloak:20.0.2
```

Repeat the preceding command if you need more image stream tags:

```
[user@host ~]$ oc create istag keycloak:19.0 \
  --from-image quay.io/keycloak/keycloak:19.0
```

- Further create another tag

```
[user@host ~]$ oc tag quay.io/keycloak/keycloak:20.0.3 keycloak:20.0
```

Verify image stream tags points to SHA ID

```
[user@host ~]$ oc describe is keycloak
Name:                   keycloak
Namespace:              myproject
Created:                5 minutes ago
Labels:                 <none>
Annotations:            openshift.io/image.dockerRepositoryCheck=2023-01-31T11:12:44Z
Image Repository:       image-registry.openshift-image-registry.svc:5000/.../keycloak
Image Lookup:           local=false
Unique Images:          3
Tags:                   2

20.0
  tagged from quay.io/keycloak/keycloak:20.0.3

  * quay.io/keycloak/keycloak@sha256:c167...62e9
      47 seconds ago
    quay.io/keycloak/keycloak@sha256:5569...b311
      5 minutes ago

19.0
  tagged from quay.io/keycloak/keycloak:19.0

  * quay.io/keycloak/keycloak@sha256:40cc...ffde
      5 minutes ago
```

Importing Image Stream Tags Periodically

- OpenShift does not check for images updates on external registry
- Manually update image stream tag point to new version
- Configure image stream tags to automatically refresh

```
[user@host ~]$ oc tag quay.io/keycloak/keycloak:20.0.3 keycloak:20.0 --scheduled
```

- Default: every 15 minutes. Cluster admin can change the setting
- To remove periodic check – re-run above command without --scheduled option

```
[user@host ~]$ oc tag quay.io/keycloak/keycloak:20.0.3 keycloak:20.0
```

Configuring Image Pull-through

- When uses image stream tag, it pulls image from external registry
  - this can take time or even fail (in case network outage/registries bandwidth throttling)
- Configure image stream to cache images into OpenShift internal registry

```
[user@host ~]$ oc tag quay.io/keycloak/keycloak:20.0.3 keycloak:20.0 \
  --reference-policy local
```

- Initial OpenShift pulls image, subsequent application deployment will points to the internal registry
- Oblivious to Application developers

Using Image Streams in Deployments

- When deployment is using external image
  1. Create image stream object in same project
  2. Enable local lookup policy
- Use –n to refer to image stream in another project
- Can use short form when referring to image stream tag
  - keycloak:20.0

Enabling Local Lookup Policy

- By default look all image stream in current project
- Disable: search in allowed local registries (include openshift namespace)

```
[user@host ~]$ oc set image-lookup keycloak
```

Use the `oc describe is` command to verify that the policy is active:

```
[user@host ~]$ oc describe is keycloak
Name:              keycloak
Namespace:         myproject
Created:           3 hours ago
Labels:            <none>
Annotations:       openshift.io/image.dockerRepositoryCheck=2023-01-31T11:12:44Z
Image Repository: image-registry.openshift-image-registry.svc:5000/.../keycloak
Image Lookup:      local=true
Unique Images:     3
Tags:              2
...output omitted...
```

Disable local lookup

- Disable local lookup for following image tag

```
[user@host ~]$ oc set image-lookup nagios --enabled=false
```

- Verify

```
[user@host ~]$ oc set image-lookup
NAME          LOCAL
keycloak      true
zabbix-agent  false
nagios        false
```

Configuring Image Streams in Deployments

- Use the **--image** or **–i** option

```
[user@host ~]$ oc run NAME --image IMAGE-STREAM-TAG
```

```
[user@host ~]$ oc create deployment mykeycloak --image keycloak:20.0
```

```
[user@host ~]$ oc create job NAME --image IMAGE-STREAM-TAG -- COMMAND
```

```
[user@host ~]$ oc create cronjob NAME --image IMAGE-STREAM-TAG \
  --schedule CRON-SYNTAX -- COMMAND
```

- IF shorthand is used, OpenShift looksfor matching IS in current project
  - provided the image is enable for local lookup
  - Otherwise search in allowed container registries

Guided Exercises: Reproducible Deployments with OpenShift Image Streams

You should be able to:

- Create image streams.
- Create image stream tags.
- Deploy applications that use image stream tags.

Lesson 4 : Automatic Image Updates with OpenShift Image Change Triggers

Ensure automatic update of application pods by using image streams with Kubernetes workload resources.

Using Triggers to Manage Images

- ISTag points to immutable image
- New version becomes available
  - Updating image stream tag point to new image is tedious (sometimes won't works)
  - Configure Deployment with image trigger
- Possibly revert image stream tag, image trigger automatically rollback to previous image
- Image Trigger supports other workload: Pod, CronJob, Job

Configuring Image Trigger for Deployments

- Ensure Deployment is using image stream tag

```
[user@host ~]$ oc get deployment mykeycloak -o wide
NAME         READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS ...
mykeycloak   0/1     1            0           6s    keycloak   ...
```

- Use oc set triggers command

```
[user@host ~]$ oc set triggers deployment/mykeycloak --from-image keycloak:20 \
  --containers keycloak
```

- The **--container** is alternative option to point to specific container
- DeploymentConfig natively support image streams and triggers
- Deployment is native to K8s: doesn't support image stream and triggers
  - OpenShift enhances by adding annotation
  - Update SHA ID whenever changes

```
[user@host ~]$ oc get deployment mykeycloak \
  -o jsonpath='{.metadata.annotations.image\.openshift\.io/triggers}' | jq .
[
  {
    "from": {
      "kind": "ImageStreamTag",
      "name": "keycloak:20"
    },
    "fieldPath": "spec.template.spec.containers[?(@.name==\"keycloak\")].image"
  }
```

View triggers

For a more concise view, use the `oc set triggers` command with the name of the `Deployment` object as an argument:

```
[user@host ~]$ oc set triggers deployment/mykeycloak
NAME                    TYPE    VALUE                   AUTO
deployments/mykeycloak  config                          true (1)
deployments/mykeycloak  image   keycloak:20 (keycloak)  true (2)
```

1. OpenShift uses the configuration trigger to roll out the deployment whenever you change its configuration, such as to update environment variables or to configure the readiness probe.
2. OpenShift watches the keycloak:20 image stream tag that the keycloak container uses

Enable/Disable Triggers

- Disable trigger

```
[user@host ~]$ oc set triggers deployment/mykeycloak --manual \
  --from-image keycloak:20 --containers keycloak
```

- Re-enable trigger

```
[user@host ~]$ oc set triggers deployment/mykeycloak --auto \
  --from-image keycloak:20 --containers keycloak
```

- Remove all triggers in deployment

```
[user@host ~]$ oc set triggers deployment/mykeycloak --remove-all
```

Rolling out and back Deployments

- DeploymentConfig with Image trigger automatically rolls out when image tag changes
  - Changes to image tag can be manual or via oc tag --scheduled option
- Rollback DeploymentConfig
  - In event of malfunctioning image
  - Use oc rollout undo command
    - ✓ Rolls back objects and disables image trigger
    - ✓ If necessary, re-enable image trigger after fix issue
- For Kubernetes deployment objects
  - Cannot revert using oc rollout undo command
  - Instead manually revert image stream tag
  - OpenShift rolls out new ReplicaSets with reverted image

Managing Image Stream Tags

```
[user@host ~]$ oc create istag keycloak:20.0.2 \
  --from-image quay.io/keycloak/keycloak:20.0.2

[user@host ~]$ oc import-image keycloak:20.0.2 \
  --from quay.io/keycloak/keycloak:20.0.2 --confirm

[user@host ~]$ oc tag quay.io/keycloak/keycloak:20.0.2 keycloak:20.0.2
```

- Several approach but achieve same result
- All above commands: Create image stream if it does not exists, and create keyclock:20.0.2 image stream tag

Update image stream tag from different source image

- Rerun following command to update image from different source image
  - oc import-image with --confirm option
  - oc tag
- Using aliases similar to floating tag for container images

```
[user@host ~]$ oc tag --alias keycloak:20.0.2 keycloak:20
```

The `oc describe is` command reports that both tags point to the same image:

```
[user@host ~]$ oc describe is keycloak
Name:       keycloak
Namespace:  myproject
...output omitted...

20.0.2 (20)
  tagged from quay.io/keycloak/keycloak:20.0.2

  * quay.io/keycloak/keycloak@sha256:5569...b311
      3 minutes ago
```

Use **oc create istag** command

- To initially create image stream tag
- Cannot update tags
- keycloak:20 image stream tag does not change. Therefore Deployment objects that uses this tag do not roll out new version of image

```
[user@host ~]$ oc create istag keycloak:20.0.3 \
  --from-image quay.io/keycloak/keycloak:20.0.3
imagestreamtag.image.openshift.io/keycloak:20.0.3 cre
[user@host ~]$ oc describe is keycloak
Name:       keycloak
Namespace:  myproject
...output omitted...

20.0.3
  tagged from quay.io/keycloak/keycloak:20.0.3

  * quay.io/keycloak/keycloak@sha256:c167...62e9
      36 seconds ago

20.0.2 (20)
  tagged from quay.io/keycloak/keycloak:20.0.2

  * quay.io/keycloak/keycloak@sha256:5569...b311
      About an hour ago
```

Use **oc create istag** command

- After testing new image, you can then move keycloak:20 tag point to new image
- OpenShift rolls out all Deployment objects that uses the tag to new image version

```
[user@host ~]$ oc tag --alias keycloak:20.0.3 keycloak:20
Tag keycloak:20 set up to track keycloak:20.0.3.
[user@host ~]$ oc describe is keycloak
Name:       keycloak
Namespace:  myproject
...output omitted...

20.0.3 (20)
  tagged from quay.io/keycloak/keycloak:20.0.3

  * quay.io/keycloak/keycloak@sha256:c167...62e9
      10 minutes ago

20.0.2
  tagged from quay.io/keycloak/keycloak:20.0.2

  * quay.io/keycloak/keycloak@sha256:5569...b311
      About an hour ago
```

If new image version not working

- Roll back deployments

```
[user@host ~]$ oc tag --alias keycloak:20.0.2 keycloak:20
```

- By providing a level of indirection, image streams give you control over managing the container images that you use in your OpenShift cluster.

Guided Exercises: Automatic Image Updates with OpenShift Image Change Triggers

You should be able to:

- Add an image trigger to a deployment.
- Modify an image stream tag to point to a new image.
- Watch the rollout of the application.
- Roll back a deployment to the previous image.

Lab: Manage Application Updates

You should be able to:

- Configure Deployment objects with images and triggers,
- Configure image stream tags and aliases.