



OpenShift Dedicated 4

Registry

Understanding registries for OpenShift Dedicated 4

OpenShift Dedicated 4 Registry

Understanding registries for OpenShift Dedicated 4

Legal Notice

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document provides provides a general overview of registries associated with OpenShift Dedicated 4.

Table of Contents

CHAPTER 1. IMAGE REGISTRY	3
1.1. INTEGRATED OPENSIFT DEDICATED REGISTRY	3
CHAPTER 2. ACCESSING THE REGISTRY	4
2.1. ACCESSING REGISTRY DIRECTLY FROM THE CLUSTER	4

CHAPTER 1. IMAGE REGISTRY

1.1. INTEGRATED OPENSIFT DEDICATED REGISTRY

OpenShift Dedicated provides a built in container image registry which runs as a standard workload on the cluster. The registry is configured and managed by an infrastructure operator. It provides an out of the box solution for users to manage the images that run their workloads, and runs on top of the existing cluster infrastructure. In addition, it is integrated into the cluster user authentication and authorization system which means that access to create and retrieve images is controlled by defining user permissions on the image resources.

The registry is typically used as a publication target for images built on the cluster as well as a source of images for workloads running on the cluster. When a new image is pushed to the registry, the cluster is notified of the new image and other components can react to and consume the updated image.

The actual image data is stored in a Red Hat managed s3 bucket. The image metadata, which is exposed by the standard cluster APIs and is used to perform access control, is stored as standard API resources, specifically images and imagestreams.

CHAPTER 2. ACCESSING THE REGISTRY

Use the following section for instructions on accessing the registry.

You can access the registry directly to invoke **podman** commands. This allows you to push images to or pull them from the integrated registry directly using operations like **podman push** or **podman pull**. To do so, you must be logged in to the registry using the **oc login** command. The operations you can perform depend on your user permissions, as described in the following sections.

2.1. ACCESSING REGISTRY DIRECTLY FROM THE CLUSTER

You can access the registry from inside the cluster.

Procedure

Access the registry from the cluster by using internal routes:

1. Log in to the container image registry by using your access token:

```
$ podman login -u $(oc whoami) -p $(oc whoami -t) $(oc -n openshift-image-registry get route
default-route -o jsonpath='{.spec.host}')
```

You should see a message confirming login, such as:

```
Login Succeeded!
```



NOTE

You can pass any value for the user name; the token contains all necessary information. Passing a user name that contains colons will result in a login failure.

Since the Image Registry Operator creates the route, it will likely be similar to **default-route-openshift-image-registry.<cluster_name>**.

2. Perform **podman pull** and **podman push** operations against your registry:



IMPORTANT

You can pull arbitrary images, but if you have the **system:registry** role added, you can only push images to the registry in your project.

In the following examples, use:

Component	Value
<registry_url>	oc -n openshift-image-registry get route default-route -o jsonpath='{.spec.host}'
<project>	openshift
<image>	image

Component	Value
<tag>	omitted (defaults to latest)

- a. Pull an arbitrary image:

```
$ podman pull name.io/image
```

- b. Tag the new image with the form **<registry_url>:<project>/<image>**. The project name must appear in this pull specification for OpenShift Dedicated to correctly place and later access the image in the registry:

```
$ podman tag name.io/image $(oc -n openshift-image-registry get route default-route -o jsonpath='{.spec.host}')/openshift/image
```



NOTE

You must have the **system:image-builder** role for the specified project, which allows the user to write or push an image. Otherwise, the **podman push** in the next step will fail. To test, you can create a new project to push the image.

- c. Push the newly-tagged image to your registry:

```
$ podman push $(oc -n openshift-image-registry get route default-route -o jsonpath='{.spec.host}')/openshift/image
```