This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Setting up Genesys Multicloud CX Private Edition

Setting up an OpenShift Container Registry

2/21/2026

Contents

- 1 What is OpenShift Container Registry?
- 2 How to set up an OpenShift Container Registry in Genesys Multicloud CX private edition environment?
  - 2.1 Exposing OpenShift Container Registry
  - 2.2 Connecting the registry
- 3 How to push an image into the registry?
- 4 How to pull images from the registry?
  - 4.1 Pulling images from registry during Helm installations
  - 4.2 Pulling images across projects

Instructions to set up an OpenShift Container Registry in your environment.

Related documentation:

-
-

RSS:

- For private edition

This page describes only the procedure to set up an OpenShift Container Registry with generic examples. You still have to manually download the artifacts of your Genesys Multicloud CX service from the JFrog Artifactory Edge repository.

What is OpenShift Container Registry?

OpenShift Container Registry is an in-built container image registry that is available by default as an integrated solution with OpenShift Container Platform. You can configure OpenShift Container Registry to maintain the source images of all Genesys Multicloud CX services running in your clusters.

To get started with OpenShift Container Registry, refer to the OpenShift documentation.

How to set up an OpenShift Container Registry in Genesys Multicloud CX private edition environment?

You can set up an OpenShift Container Registry by exposing its default route and connecting the registry.

Exposing OpenShift Container Registry

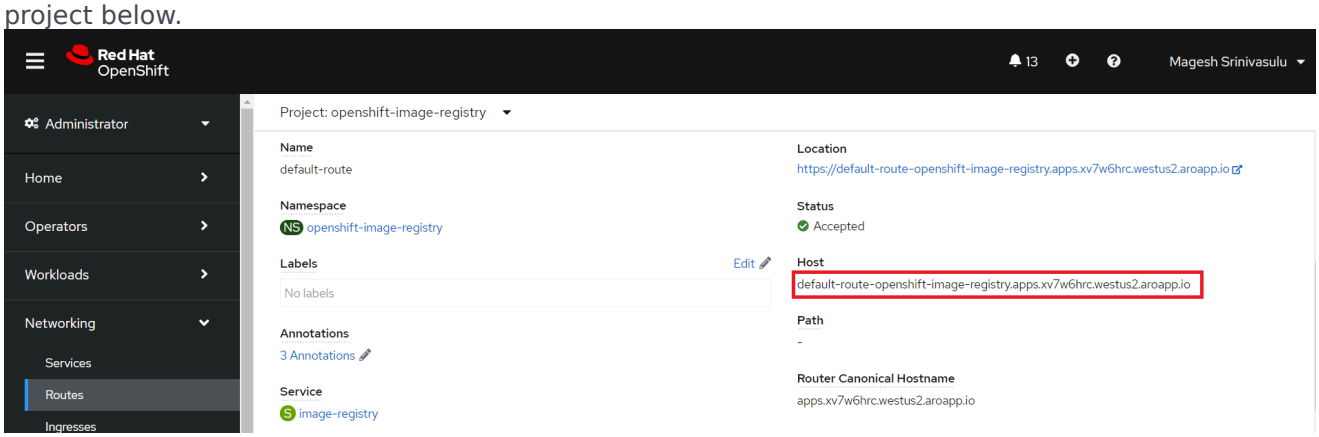
The first step in setting up an OpenShift Container Registry is to expose the registry through the default or customized route. You can do so by running the following command.

```
oc patch configs.imageregistry.operator.openshift.io/cluster --patch
'{"spec":{"defaultRoute":true}}' --type=merge
```

Once you run the above command, you can find the default route getting updated in the **Host** field. The default route is the location of the image registry which you can connect to perform image related operations such as `pull`, `push`, etc.

See an example screenshot showing **default route** in the **Host** field for **openshift-image-registry**

project below.



Additional information on 'Exposing the registry' is available at OpenShift documentation.

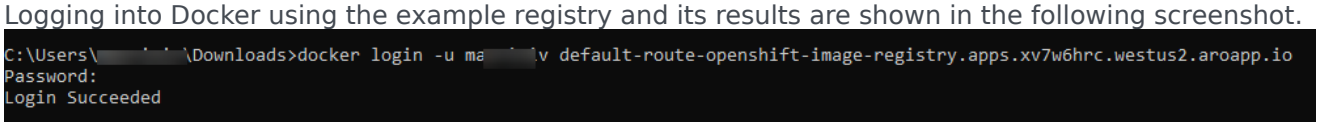
Connecting the registry

Once you expose the registry, you can connect to it by using the `docker login` command given below by providing the **Host** value.

```
docker login -u > >
```

Running the above command prompts for password. Provide the password details and proceed further.

Logging into Docker using the example registry and its results are shown in the following screenshot.



How to push an image into the registry?

Pushing an image to OpenShift Container Registry is a two step process - tagging and pushing. It is similar to pushing an image using `docker push` command.

> **Important**
>
> Practically, you will place the downloaded images in a dedicated quarantine location for security scans. Once the image passes the security scans defined by your organization, you will decide to push the image into your container registry.

An example of pushing the latest version of postgreSQL is shown in the

screenshots.

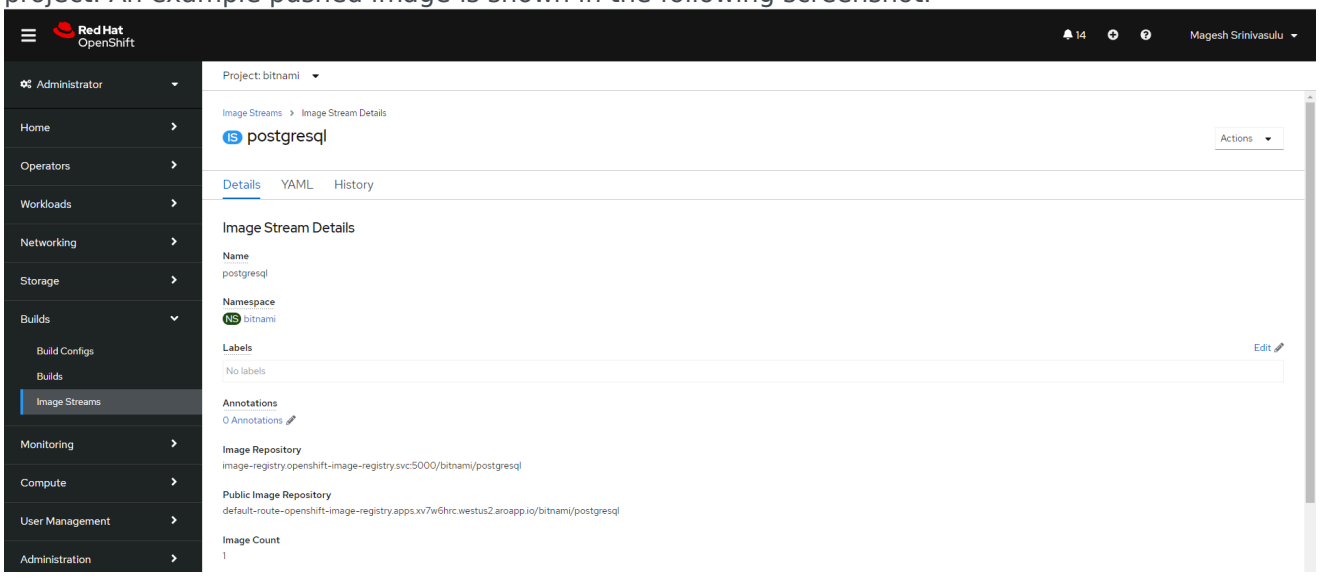
- **Tagging** - the first step is to tag the image with appropriate version name.
  `docker tag >:> >/>/>:>`

```
C:\Users\masriniv\Downloads>docker tag bitnami/postgresql:latest default-route-openshift-image-registry.apps.xv7w6hrc.westus2.aroapp.io/bitnami/postgresql:latest

C:\Users\masriniv\Downloads>docker images
REPOSITORY                                                                      TAG       IMAGE ID       CREATED        SIZE
default-route-openshift-image-registry.apps.xv7w6hrc.westus2.aroapp.io/bitnami/postgresql   latest    42267548cd2a   2 days ago     258MB
bitnami/postgresql                                                              latest    42267548cd2a   2 days ago     258MB
gcr.io/k8s-minikube/kicbase                                                     v0.0.17   a9b1f16d8ece   2 months ago   985MB
```

- **Pushing** - the second step is to push the tagged image into the registry.
  `docker push >/>/>:>`

```
C:\Users\masriniv\Downloads>docker images
REPOSITORY                                                                      TAG       IMAGE ID       CREATED        SIZE
bitnami/postgresql                                                              latest    42267548cd2a   2 days ago     258MB
default-route-openshift-image-registry.apps.xv7w6hrc.westus2.aroapp.io/bitnami/postgresql   latest    42267548cd2a   2 days ago     258MB
gcr.io/k8s-minikube/kicbase                                                     v0.0.17   a9b1f16d8ece   2 months ago   985MB

C:\Users\masriniv\Downloads>docker push default-route-openshift-image-registry.apps.xv7w6hrc.westus2.aroapp.io/bitnami/postgresql
Using default tag: latest
The push refers to repository [default-route-openshift-image-registry.apps.xv7w6hrc.westus2.aroapp.io/bitnami/postgresql]
6a5db3294032: Layer already exists
e271d9bc294a: Layer already exists
ff44c061c7f7: Layer already exists
536f65d434c8: Layer already exists
586bf4d8e4cb: Layer already exists
8eee54dba937: Layer already exists
144c8eed628e: Layer already exists
065032d7482f: Layer already exists
0a22b631b162: Layer already exists
c2bd774a7d1c: Layer already exists
e371050776ae: Layer already exists
b4dcf0ba5e60: Layer already exists
f3871f5dd620: Layer already exists
latest: digest: sha256:5e2e868fb3489042221549e664ed542ea7020d959c9d9ef449e62de96e91f867 size: 3046
```

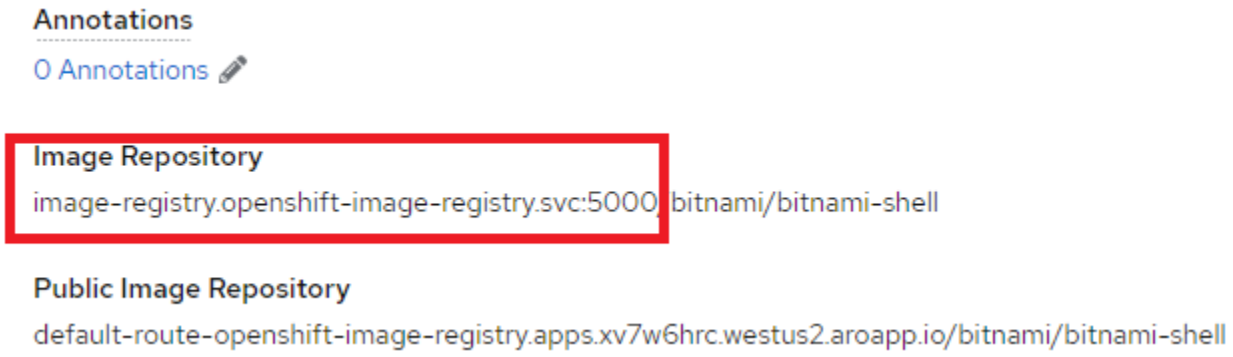
You can see the image pushed into the registry on the **Image Streams** tab of the corresponding project. An example pushed image is shown in the following screenshot.



How to pull images from the registry?

Pulling images from registry during Helm installations

You can pull the image from the OpenShift Container Registry during Helm installations. You can do so by overriding the corresponding Helm parameter with the internal registry details. The internal registry value is highlighted in the example screenshot given below.



Pulling images across projects

You can refer the image created under one project across different projects by creating policies. https://access.redhat.com/documentation/en-us/openshift_container_platform/4.1/html/images/managing-images#images-allow-pods-to-reference-images-across-projects_using-image-pull-secrets.

The command to create policy is as follows:

```
oc policy add-role-to-user system:image-puller system:serviceaccount:>:default --
namespace=>
```

> **Tip**
> You can verify that image is being pulled and used in Helm installation by either navigating to the **Workloads >> Pods >> Events** tab in UI or by running the command `oc describe pod` in the command line.