



Red Hat Developer Hub 1.2

Installing Red Hat Developer Hub in an air-gapped environment

Red Hat Developer Hub 1.2 Installing Red Hat Developer Hub in an air-gapped environment

Legal Notice

Copyright © 2024 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

Red Hat Developer Hub is an enterprise-grade platform for building developer portals. Administrative users can configure roles, permissions, and other settings to enable other authorized users to deploy an air-gapped Developer Hub instance on any supported platform using either the Operator or Helm chart.

Table of Contents

CHAPTER 1. AIR-GAPPED ENVIRONMENT	3
CHAPTER 2. INSTALLING RED HAT DEVELOPER HUB IN AN AIR-GAPPED ENVIRONMENT WITH THE OPERATOR	4
CHAPTER 3. INSTALLING RED HAT DEVELOPER HUB IN AN AIR-GAPPED ENVIRONMENT WITH THE HELM CHART	6

CHAPTER 1. AIR-GAPPED ENVIRONMENT

An air-gapped environment, also known as an air-gapped network or isolated network, ensures security by physically segregating the system or network. This isolation is established to prevent unauthorized access, data transfer, or communication between the air-gapped system and external sources.

You can install the Red Hat Developer Hub in an air-gapped environment to ensure security and meet specific regulatory requirements.

CHAPTER 2. INSTALLING RED HAT DEVELOPER HUB IN AN AIR-GAPPED ENVIRONMENT WITH THE OPERATOR

On an OpenShift Container Platform cluster operating on a restricted network, public resources are not available. However, deploying the Red Hat Developer Hub Operator and running Developer Hub requires the following public resources:

- Operator images (bundle, operator, catalog)
- Operands images (RHDH, PostgreSQL)

To make these resources available, replace them with their equivalent resources in a mirror registry accessible to the OpenShift Container Platform cluster.

You can use a helper script that mirrors the necessary images and provides the necessary configuration to ensure those images will be used when installing the Red Hat Developer Hub Operator and creating Developer Hub instances.



NOTE

This script requires a target mirror registry which you should already have installed if your OpenShift Container Platform cluster is ready to operate on a restricted network. However, if you are preparing your cluster for disconnected usage, you can use the script to deploy a mirror registry in the cluster and use it for the mirroring process.

Prerequisites

- You have an active OpenShift CLI (**oc**) session with administrative permissions to the OpenShift Container Platform cluster. See [Getting started with the OpenShift CLI](#).
- You have an active **oc registry** session to the **registry.redhat.io** Red Hat Ecosystem Catalog. See [Red Hat Container Registry Authentication](#).
- The **opm** CLI tool is installed. See [Installing the opm CLI](#).
- The **jq** package is installed. See [Download jq](#).
- Podman is installed. See [Podman Installation Instructions](#).
- Skopeo version 1.14 or higher is installed. See [Installing Skopeo](#).
- If you already have a mirror registry for your cluster, an active Skopeo session with administrative access to this registry is required. See [Authenticating to a registry](#) and [Mirroring images for a disconnected installation](#).



NOTE

The internal OpenShift Container Platform cluster image registry cannot be used as a target mirror registry. See [About the mirror registry](#).

- If you prefer to create your own mirror registry, see [Creating a mirror registry with mirror registry for Red Hat OpenShift](#).
- If you do not already have a mirror registry, you can use the helper script to create one for you and you need the following additional prerequisites:

- The `cURL` package is installed. For Red Hat Enterprise Linux, the `curl` command is available by installing the `curl` package. To use `curl` for other platforms, see the [cURL website](#).
- The `htpasswd` command is available. For Red Hat Enterprise Linux, the `htpasswd` command is available by installing the `httpd-tools` package.

Procedure

1. Download and run the mirroring script to install a custom Operator catalog and mirror the related images: **prepare-restricted-environment.sh** ([source](#)).

```
curl -sLO https://raw.githubusercontent.com/redhat-developer/rhdh-operator/1.2.x/.rhdh/scripts/prepare-restricted-environment.sh

# if you do not already have a target mirror registry
# and want the script to create one for you
# use the following example:
bash prepare-restricted-environment.sh \
  --prod_operator_index "registry.redhat.io/redhat/redhat-operator-index:v4.15" \
  --prod_operator_package_name "rhdh" \
  --prod_operator_bundle_name "rhdh-operator" \
  --prod_operator_version "v1.2.4"

# if you already have a target mirror registry
# use the following example:
bash prepare-restricted-environment.sh \
  --prod_operator_index "registry.redhat.io/redhat/redhat-operator-index:v4.15" \
  --prod_operator_package_name "rhdh" \
  --prod_operator_bundle_name "rhdh-operator" \
  --prod_operator_version "v1.2.4" \
  --use_existing_mirror_registry "my_registry"
```



NOTE

The script can take several minutes to complete as it copies multiple images to the mirror registry.

CHAPTER 3. INSTALLING RED HAT DEVELOPER HUB IN AN AIR-GAPPED ENVIRONMENT WITH THE HELM CHART

An air-gapped environment, also known as an air-gapped network or isolated network, ensures security by physically segregating the system or network. This isolation is established to prevent unauthorized access, data transfer, or communication between the air-gapped system and external sources.

You can install Red Hat Developer Hub in an air-gapped environment to ensure security and meet specific regulatory requirements.

To install Developer Hub in an air-gapped environment, you must have access to the **registry.redhat.io** and the registry for the air-gapped environment.

Prerequisites

- You have installed an Red Hat OpenShift Container Platform 4.13 or later.
- You have access to the **registry.redhat.io**.
- You have access to the Red Hat OpenShift Container Platform image registry of your cluster. For more information about exposing the image registry, see the Red Hat OpenShift Container Platform documentation about [Exposing the registry](#).
- You have installed the OpenShift CLI (**oc**) on your workstation.
- You have installed the **podman** command line tools on your workstation.
- You you have an account in [Red Hat Developer](#) portal.

Procedure

1. Log in to your OpenShift Container Platform account using the OpenShift CLI (**oc**), by running the following command:

```
oc login -u <user> -p <password> https://api.<hostname>:6443
```

2. Log in to the OpenShift Container Platform image registry using the **podman** command line tool, by running the following command:

```
podman login -u kubeadmin -p $(oc whoami -t) default-route-openshift-image-registry.<hostname>
```



NOTE

You can run the following commands to get the full host name of the OpenShift Container Platform image registry, and then use the host name in a command to log in:

```
REGISTRY_HOST=$(oc get route default-route -n openshift-image-registry --template='{{ .spec.host }}')
```

```
podman login -u kubeadmin -p $(oc whoami -t) $REGISTRY_HOST
```

3. Log in to the **registry.redhat.io** in **podman** by running the following command:

```
podman login registry.redhat.io
```

For more information about registry authentication, see [Red Hat Container Registry Authentication](#).

4. Pull Developer Hub and PostgreSQL images from [Red Hat Image registry](#) to your workstation, by running the following commands:

```
podman pull registry.redhat.io/rhdh/rhdh-hub-rhel9:1.2.4
```

```
podman pull registry.redhat.io/rhel9/postgresql-15:latest
```

5. Push both images to the internal OpenShift Container Platform image registry by running the following commands:

```
podman push --remove-signatures registry.redhat.io/rhdh/rhdh-hub-rhel9:1.2.4 default-route-openshift-image-registry.<hostname>/<project_name>/rhdh-hub-rhel9:1.2.4
```

```
podman push --remove-signatures registry.redhat.io/rhel9/postgresql-15:latest default-route-openshift-image-registry.<hostname>/<project_name>/postgresql-15:latest
```

For more information about pushing images directly to the OpenShift Container Platform image registry, see [How do I push an Image directly into the OpenShift 4 registry](#) .



IMPORTANT

If an x509 error occurs, verify that you have [installed the CA certificate used for OpenShift Container Platform routes on your system](#).

6. Use the following command to verify that both images are present in the internal OpenShift Container Platform registry:

```
oc get imagestream -n <project_name>
```

7. Enable local image lookup for both images by running the following commands:

```
oc set image-lookup postgresql-15
```

```
oc set image-lookup rhdh-hub-rhel9
```

8. Go to **YAML view** and update the **image** section for **backstage** and **postgresql** using the following values:

Example values for Developer Hub image

```
upstream:
  backstage:
    image:
```

```
registry: ""  
repository: rhdh-hub-rhel9  
tag: latest
```

Example values for PostgreSQL image

```
upstream:  
  postgresql:  
    image:  
      registry: ""  
      repository: postgresql-15  
      tag: latest
```

9. Install the Red Hat Developer Hub using Helm chart.