Red Hat Developer Hub 1.8

Installing Red Hat Developer Hub on OpenShift Container Platform

Running Red Hat Developer Hub on Red Hat OpenShift Container Platform by using either the Operator or Helm chart

Last Updated: 2025-11-25

Red Hat Developer Hub 1.8 Installing Red Hat Developer Hub on OpenShift Container Platform

Running Red Hat Developer Hub on Red Hat OpenShift Container Platform by using either the Operator or Helm chart

Legal Notice

Abstract

Platform administrators can configure roles, permissions, and other settings to enable other authorized users to deploy a Red Hat Developer Hub (RHDH) instance on Red Hat OpenShift Container Platform using either the Operator or Helm chart.

Table of Contents

PREFACE

You can install Red Hat Developer Hub on OpenShift Container Platform by using one of the following installers:

The Red Hat Developer Hub Operator

- Ready for immediate use in OpenShift Container Platform after an administrator installs it with OperatorHub
- Uses Operator Lifecycle Management (OLM) to manage automated subscription updates on OpenShift Container Platform
- Requires preinstallation of Operator Lifecycle Management (OLM) to manage automated subscription updates on Kubernetes

The Red Hat Developer Hub Helm chart

- Ready for immediate use in both OpenShift Container Platform and Kubernetes
- Requires manual installation and management

IMPORTANT

You must set the **baseUrl** in **app-config.yaml** to match the external URL of your Developer Hub instance (for example, **https://rhdh.example.com**). This value is required for the Red Hat Developer Hub to function correctly. If it is not set, frontend and backend services cannot communicate properly, and features may not work as expected.

Use the installation method that best meets your needs and preferences.

Additional resources

- Choosing an installation method: Helm Charts vs. Operators
- Working with Operators
- Working with Helm charts

CHAPTER 1. INSTALLING RED HAT DEVELOPER HUB ON OPENSHIFT CONTAINER PLATFORM WITH THE OPERATOR

You can install Red Hat Developer Hub on OpenShift Container Platform by using the Red Hat Developer Hub Operator in the OpenShift Container Platform console.

1.1. INSTALLING THE RED HAT DEVELOPER HUB OPERATOR

As an administrator, you can install the Red Hat Developer Hub Operator. Authorized users can use the Operator to install Red Hat Developer Hub on Red Hat OpenShift Container Platform (OpenShift Container Platform) and supported Kubernetes platforms. For more information on supported platforms and versions, see the Red Hat Developer Hub Life Cycle page.

Containers are available for the following CPU architectures:

- AMD64 and Intel 64 (**x86_64**)

Prerequisites

- You are logged in as an administrator on the OpenShift Container Platform web console.
- You have configured the appropriate roles and permissions within your project to create or access an application. For more information, see the Red Hat OpenShift Container Platform documentation on Building applications.
- You have installed Red Hat OpenShift Container Platform 4.16 to 4.19.
- Make sure that your system meets the minimum sizing requirements. See Sizing requirements for Red Hat Developer Hub.

Procedure

1. In the navigation menu of the OpenShift Container Platform console, click **Operators > OperatorHub**.
2. In the **Filter by keyword** box, enter Developer Hub and click the **Red Hat Developer Hub Operator** card.
3. On the **Red Hat Developer Hub Operator**page, read the information about the Operator and click **Install** to open the **Install Operator** page.
4. After the Operator is successfully installed, provision your custom configuration: Before you create a Developer Hub instance, you must create the required config map and Secret resources in your project. These include the **baseUrl** and service-to-service authentication secrets.

   For detailed steps, see Provisioning your custom Red Hat Developer Hub configuration .
5. From the **Update channel** drop-down menu, select the update channel that you want to use, for example, **fast** or **fast-1.8**.

IMPORTANT

The `**fast** channel includes all of the updates available for a particular version. Any update might introduce unexpected changes in your Red Hat Developer Hub deployment. Check the release notes for details about any potentially breaking changes.

The **fast-1.8** channel only provides z-stream updates, for example, updating from version 1.8.1 to 1.8.2. If you want to update the Red Hat Developer Hub y-version in the future, for example, updating from 1.8 to 1.9, you must switch to the **fast-1.9** channel manually.

6. From the **Version** drop-down menu, select the version of the Red Hat Developer Hub Operator that you want to install. The default version is the latest version available in the selected channel.

7. Select the Operator **Installation mode**.

NOTE

The **All namespaces on the cluster (default)**option is selected by default. The **Specific namespace on the cluster**option is not currently supported.

8. In the **Installed Namespace** field, do one of the following actions:

   - Select **Operator recommended Namespace** to create and use the **rhdh-operator** namespace. This option is selected by default.

   - Select **Select a Namespace** to use an alternative namespace.

     - From the **Select Project** drop-down menu, do one of the following actions:

       - Select an existing project.

       - Select **Create Project** to create a new project for the Operator.

         - On the **Create Project** dialog, enter text into the required fields and click **Create**.

> IMPORTANT
>
> For enhanced security, better control over the Operator lifecycle, and preventing potential privilege escalation, install the Red Hat Developer Hub Operator in a dedicated default **rhdh-operator** namespace. You can restrict other users' access to the Operator resources through role bindings or cluster role bindings.
>
> You can also install the Operator in another namespace by creating the necessary resources, such as an Operator group. For more information, see Installing global Operators in custom namespaces.
>
> However, if the Red Hat Developer Hub Operator shares a namespace with other Operators, then it shares the same update policy as well, preventing the customization of the update policy. For example, if one Operator is set to manual updates, the Red Hat Developer Hub Operator update policy is also set to manual. For more information, see Colocation of Operators in a namespace.

9. Select the **Update approval** method for the Operator.
   - If you select the **Automatic** option, the Operator is updated without requiring manual confirmation.
   - If you select the **Manual** option, a notification opens when a new update is released in the update channel. The update must be manually approved by an administrator before installation can begin.
10. Click **Install**.

> NOTE
>
> If you selected a **Manual** approval strategy, the upgrade status of the subscription remains **Upgrading** until you review and approve the install plan. After you click **Approve** on the **Install Plan** page, the subscription upgrade status changes to **Up to date**.
>
> If you selected an **Automatic** approval strategy, the upgrade status should resolve to **Up to date** without intervention.

Verification

- Immediately after the Operator is installed, the dialog box on the **OperatorHub** page displays the **Installed operator: ready for use**message.
- From the dialog box, do one of the following actions:
  - Click **View Operator** to open the **Operator details** page for the Red Hat Developer Hub Operator.
  - Click **View all installed operators** to open the **Installed Operators** page.

  - From the list of installed Operators, locate the Red Hat Developer Hub Operator name and details.
  - Click **Red Hat Developer Hub Operator**to open the **Operator details** page for the Red Hat Developer Hub Operator.

1.2. DEPLOYING RED HAT DEVELOPER HUB ON OPENSHIFT CONTAINER PLATFORM WITH THE OPERATOR

As a developer, you can deploy a Red Hat Developer Hub instance on OpenShift Container Platform by using the **Developer Catalog** in the Red Hat OpenShift Container Platform web console. This deployment method uses the Red Hat Developer Hub Operator.

Prerequisites

- You have set the **baseUrl** in your **app-config.yaml** to match the external URL of your Developer Hub instance. Without it, frontend and backend services cannot communicate, and features might not work as expected.
- An OpenShift Container Platform administrator has installed the Red Hat Developer Hub Operator.
- You have provisioned your custom config maps and secrets in your ***<my-rhdh-project>*** project.
- You have authored your Backstage custom resource .

Procedure

1. In the OpenShift Container Platform web console, select your ***<{my_product_namespace}>*** project, then click **Add**.
2. From the **Developer Catalog** panel, click **Operator Backed**.
3. In the **Filter by keyword** box, enter *Developer Hub* and click the **Red Hat Developer Hub**card.
4. Provision your custom configuration using the following template:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config-rhdh
data:
  "app-config.yaml": |
    app:
      title: Red Hat Developer Hub
      baseUrl: https://__&lt;my_developer_hub_domain&gt;__
    backend:
      auth:
        externalAccess:
            - type: legacy
              options:
                subject: legacy-default-config
                secret: "${BACKEND_SECRET}"
      baseUrl: https://__&lt;my_developer_hub_domain&gt;__
      cors:
        origin: https://__&lt;my_developer_hub_domain&gt;__
```

Use a config map named **app-config-rhdh** to provide your **app-config.yaml** file, and a Secret for service-to-service authentication (such as **BACKEND_SECRET**).

To create and apply these configuration resources, follow the steps in Provisioning your custom Red Hat Developer Hub configuration for the full procedure.

NOTE

The **app-config-rhdh** config map must include your customized **app-config.yaml** file. This config map is mounted into the Developer Hub container at runtime.

5. Create a secret named **my-rhdh-secrets** and add a key named **BACKEND_SECRET** with a **Base64-encoded** string as value, as shown in the following example:

```
apiVersion: v1
kind: Secret
metadata:
  name: my-rhdh-secrets
stringData:
  # TODO: Add the necessary auth secrets for service-to-service auth setup
  BACKEND_SECRET: "xxx" # Replace with your Base64-encoded secret
```

NOTE

Ensure that your secret includes **BACKEND_SECRET**, used for service-to-service authentication. For structure and provisioning steps, see Provisioning your custom Red Hat Developer Hub configuration.

6. Click **Create**.

7. Add your Backstage custom resource content .

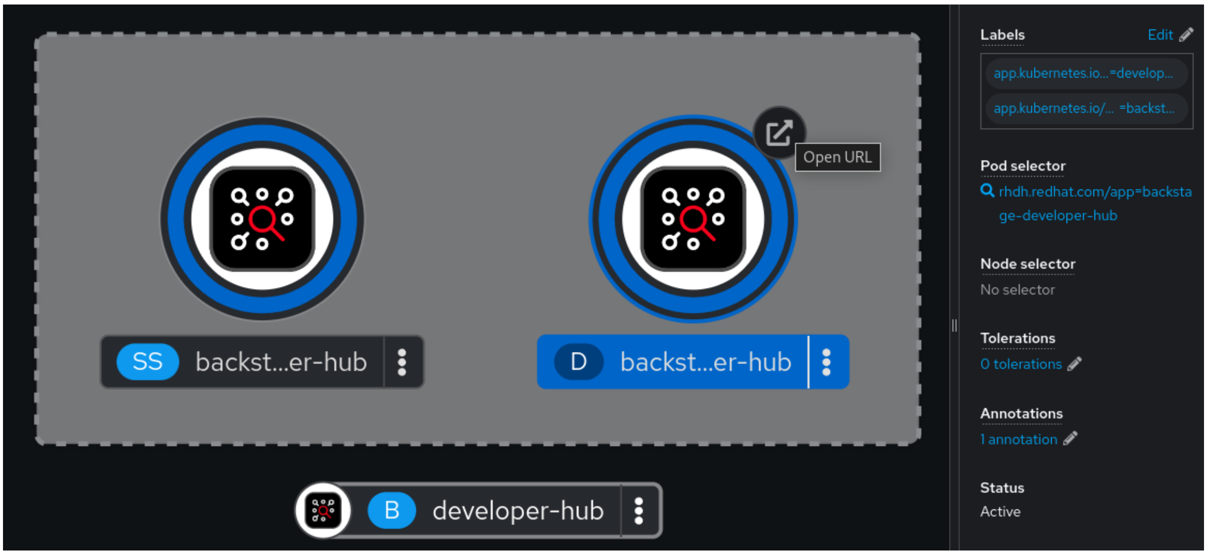
8. On the **Create Backstage** page, click **Create**.

Verification

After the pods are ready, you can access the Red Hat Developer Hub platform by opening the URL.

1. Confirm that the pods are ready by clicking the pod in the **Topology** view and confirming the **Status** in the **Details** panel. The pod status is **Active** when the pod is ready.

2. From the **Topology** view, click the **Open URL** icon on the Developer Hub pod.



Labels Edit

app.kubernetes.io...=develop...

app.kubernetes.io/... =backst...

Pod selector

rhdh.redhat.com/app=backstage-developer-hub

Node selector

No selector

Tolerations

0 tolerations

Annotations

1 annotation

Status

Active

Additional resources

- OpenShift Container Platform - Building applications overview

CHAPTER 2. INSTALLING RED HAT DEVELOPER HUB ON OPENSHIFT CONTAINER PLATFORM WITH THE HELM CHART

You can install Red Hat Developer Hub on OpenShift Container Platform by using the Helm chart with one of the following methods:

- The OpenShift Container Platform console
- The Helm CLI

2.1. DEPLOYING DEVELOPER HUB FROM THE OPENSHIFT CONTAINER PLATFORM WEB CONSOLE WITH THE HELM CHART

You can use a Helm chart to install Developer Hub on the Red Hat OpenShift Container Platform web console.

Helm is a package manager on OpenShift Container Platform that provides the following features:

- Applies regular application updates using custom hooks
- Manages the installation of complex applications
- Provides charts that you can host on public and private servers
- Supports rolling back to previous application versions

The Red Hat Developer Hub Helm chart is available in the Helm catalog on OpenShift Dedicated and OpenShift Container Platform.

Prerequisites

- You are logged in to your OpenShift Container Platform account.
- A user with the OpenShift Container Platform **admin** role has configured the appropriate roles and permissions within your project to create an application. For more information about OpenShift Container Platform roles, see Using RBAC to define and apply permissions .
- You have created a project in OpenShift Container Platform. For more information about creating a project in OpenShift Container Platform, see Red Hat OpenShift Container Platform documentation.
- Make sure that your system meets the minimum sizing requirements. See Sizing requirements for Red Hat Developer Hub.

Procedure

1. From the **Developer** perspective on the Developer Hub web console, click **+Add**.
2. From the **Developer Catalog** panel, click **Helm Chart**.
3. In the **Filter by keyword** box, enter *Developer Hub* and click the **Red Hat Developer Hub**card.
4. From the Red Hat Developer Hub page, click **Create**.

5. From your cluster, copy the OpenShift Container Platform router host (for example: **apps.<clusterName>.com**).

6. Select the radio button to configure the Developer Hub instance with either the form view or YAML view. The Form view is selected by default.

   - Using **Form view**

     a. To configure the instance with the Form view, go to **Root Schema** → **global** → **Enable service authentication within Backstage instance** and paste your OpenShift Container Platform router host into the field on the form.

   - Using **YAML view**

     a. To configure the instance with the YAML view, paste your OpenShift Container Platform router hostname in the **global.clusterRouterBase** parameter value as shown in the following example:

     ```
     global:
       auth:
         backend:
           enabled: true
       clusterRouterBase: apps.<clusterName>.com
     ```

7. Edit the other values if needed.

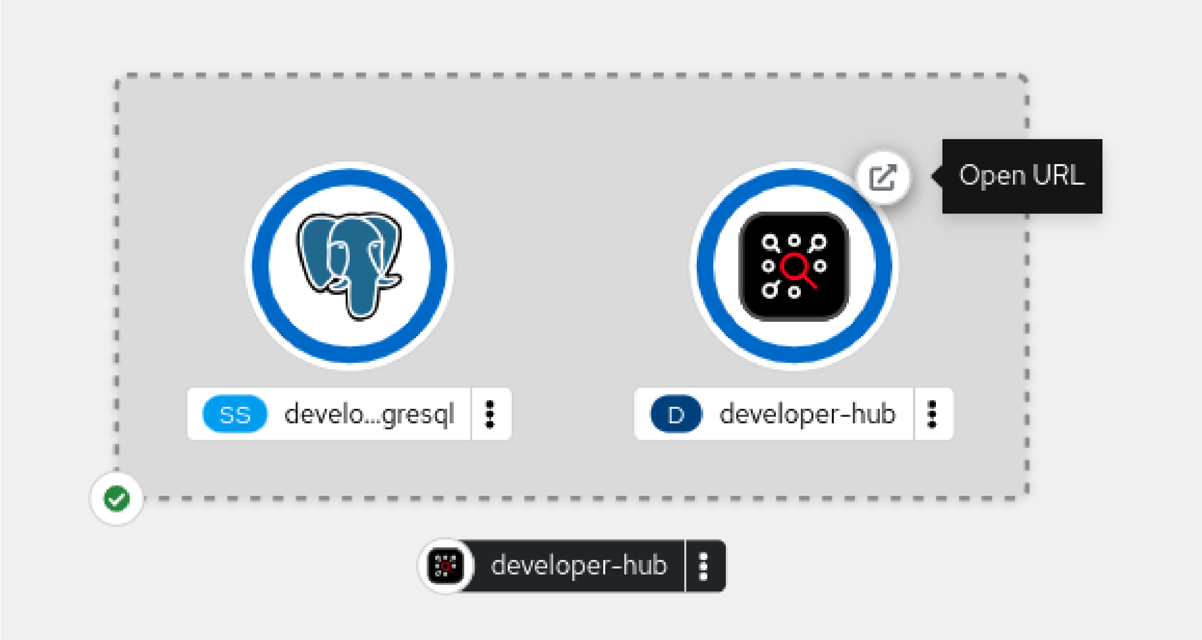
   NOTE

   The information about the host is copied and can be accessed by the Developer Hub backend.

   When an OpenShift Container Platform route is generated automatically, the host value for the route is inferred and the same host information is sent to the Developer Hub. Also, if the Developer Hub is present on a custom domain by setting the host manually using values, the custom host takes precedence.

8. Click **Create** and wait for the database and Developer Hub to start.

9. Click the **Open URL** icon to start using the Developer Hub platform.



NOTE

Your **developer-hub** pod might be in a **CrashLoopBackOff** state if the Developer Hub container cannot access the configuration files. This error is indicated by the following log:

```
Loaded config from app-config-from-configmap.yaml, env
...
2023-07-24T19:44:46.223Z auth info Configuring "database" as KeyStore provider
type=plugin
Backend failed to start up Error: Missing required config value at
'backend.database.client'
```

To resolve the error, verify the configuration files.

2.2. DEPLOYING DEVELOPER HUB ON OPENSHIFT CONTAINER PLATFORM WITH THE HELM CLI

You can use the Helm CLI to install Red Hat Developer Hub on Red Hat OpenShift Container Platform.

Prerequisites

- You have installed the OpenShift CLI (**oc**) on your workstation.
- You are logged in to your OpenShift Container Platform account.
- A user with the OpenShift Container Platform admin role has configured the appropriate roles and permissions within your project to create an application. For more information about OpenShift Container Platform roles, see Using RBAC to define and apply permissions .
- You have created a project in OpenShift Container Platform. For more information about creating a project in OpenShift Container Platform, see Red Hat OpenShift Container Platform documentation.
- You have installed the Helm CLI tool.

Procedure

1. Create and activate the *<my-rhdh-project>* OpenShift Container Platform project:

```
NAMESPACE=<emphasis><rhdh></emphasis>
oc new-project ${NAMESPACE} || oc project ${NAMESPACE}
```

2. Install the Red Hat Developer Hub Helm chart:

```
helm upgrade redhat-developer-hub -i https://github.com/openshift-helm-
charts/charts/releases/download/redhat-redhat-developer-hub-1.8.0/redhat-developer-hub-
1.8.0.tgz
```

3. Configure your Developer Hub Helm chart instance with the Developer Hub database password and router base URL values from your OpenShift Container Platform cluster:

```
PASSWORD=$(oc get secret redhat-developer-hub-postgresql -o jsonpath="
{.data.password}" | base64 -d)
CLUSTER_ROUTER_BASE=$(oc get route console -n openshift-console -
o=jsonpath='{.spec.host}' | sed 's/^[^.]*\.//')
helm upgrade redhat-developer-hub -i "https://github.com/openshift-helm-
charts/charts/releases/download/redhat-redhat-developer-hub-1.8.0/redhat-developer-hub-
1.8.0.tgz" \
    --set global.clusterRouterBase="$CLUSTER_ROUTER_BASE" \
    --set global.postgresql.auth.password="$PASSWORD"
```

4. Display the running Developer Hub instance URL:

```
echo "https://redhat-developer-hub-$NAMESPACE.$CLUSTER_ROUTER_BASE"
```

Verification

- Open the running Developer Hub instance URL in your browser to use Developer Hub.

Additional resources

- Installing Helm