



Red Hat CodeReady Studio 12.16

Getting Started with Container and Cloud-based Development

Starting Development of Container and Cloud-based Applications Using Red Hat CodeReady Studio

Red Hat CodeReady Studio 12.16 Getting Started with Container and Cloud-based Development

Starting Development of Container and Cloud-based Applications Using Red Hat CodeReady Studio

Levi Valeeva
levi@redhat.com

Supriya Takkhi
sbharadw@redhat.com

Yana Hontyk
yhontyk@redhat.com

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 compilation of topics contains information on how to start developing containerized applications and applications for cloud deployment.

Table of Contents

CHAPTER 1. DEVELOPING USING CONTAINERS AND THE CLOUD	3
1.1. USING RED HAT CODEREADY CONTAINERS TOOLING IN CODEREADY STUDIO	3
1.1.1. Downloading and installing Red Hat CodeReady Containers in CodeReady Studio	3
1.1.2. Using the OpenShift Container Platform tooling	6
1.2. USING RED HAT CONTAINER DEVELOPMENT KIT TOOLING IN CODEREADY STUDIO	12
1.2.1. Installing Container Development Kit in CodeReady Studio	12
1.2.2. Using Docker tooling	17
1.2.2.1. Creating a Dockerfile	17
1.2.2.2. Building the Docker image Using the Container Development Environment	23
1.2.2.3. Additional resources	25
1.2.3. Using the OpenShift Container Platform tooling	25
1.2.4. Additional resources	31
CHAPTER 2. DEVELOPING FOR THE CLOUD WITH OPENSIFT	32
2.1. CREATING AN OPENSIFT CONTAINER PLATFORM APPLICATION IN CODEREADY STUDIO	32
2.1.1. Creating a new OpenShift Container Platform connection	32
2.1.2. Creating a new OpenShift Container Platform project	35
2.1.3. Creating a new OpenShift Container Platform application	37
2.1.4. Importing an existing OpenShift Container Platform application into the IDE	42
2.1.5. Deploying an application using the server adapter	45
2.1.6. Deleting an OpenShift Container Platform project	48
2.2. SETTING UP AND REMOTELY MONITORING AN OPENSIFT CONTAINER PLATFORM APPLICATION	50
2.2.1. Setting up OpenShift Client Binaries	50
2.2.2. Setting up Port Forwarding	51
2.2.3. Streaming Pod Logs	54
2.2.4. Streaming Build Logs	56
2.3. ADDITIONAL RESOURCES	58
CHAPTER 3. DEVELOPING WITH DOCKER	59
3.1. MANAGING DOCKER CONNECTIONS	59
3.1.1. Setting up a Docker account in CodeReady Studio	59
3.1.2. Testing the Docker connection	60
3.1.3. Editing the Docker connection	63
3.2. MANAGING DOCKER IMAGES	65
3.2.1. Pulling Docker images	65
3.2.2. Pushing Docker images	70
3.2.3. Running Docker images	74
3.2.4. Building images with Dockerfile	78
3.3. MANAGING DOCKER CONTAINERS	80

CHAPTER 1. DEVELOPING USING CONTAINERS AND THE CLOUD

1.1. USING RED HAT CODEREADY CONTAINERS TOOLING IN CODEREADY STUDIO

Red Hat CodeReady Containers (CRC) brings a minimal OpenShift 4 cluster to your local computer. This cluster provides a minimal environment for development and testing purposes. It is mainly targeted at running on developers' desktops. For other use cases, such as headless, multi-developer or team-based setups, use of the [full-fledged OpenShift installer](#) is recommended.

For a more in-depth introduction to OpenShift, see [OpenShift documentation](#).

Prerequisites

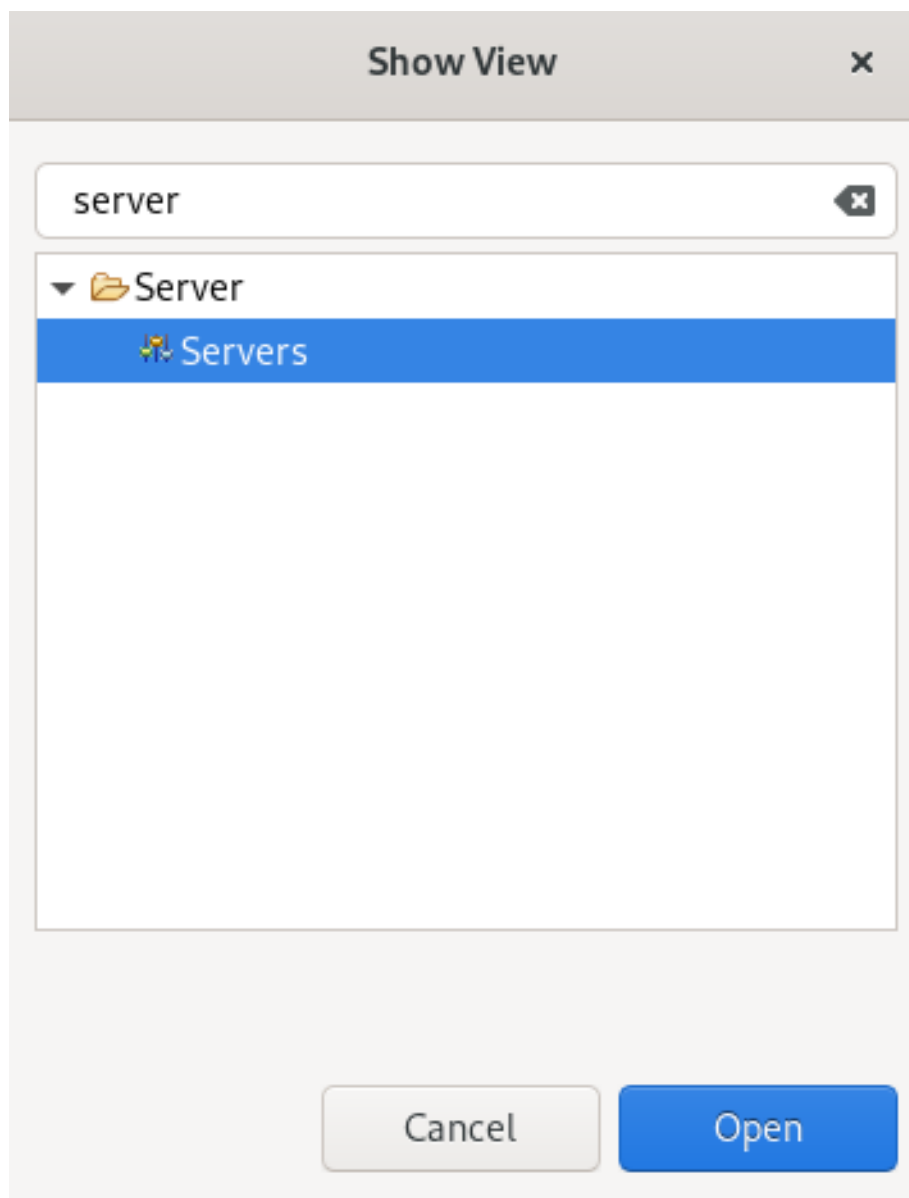
1. [Download the latest release of CodeReady Containers and the pull secret](#).
2. Extract the CRC file.
For more information on how to install and set up CRC, see the [Installation](#) chapter of the Getting started with CodeReady Containers Guide.

1.1.1. Downloading and installing Red Hat CodeReady Containers in CodeReady Studio

The following section describes how to set up CodeReady Containers in CodeReady Studio. The section assumes you completed the steps listed in the prerequisites section for this chapter.

Procedure

1. Start CodeReady Studio.
2. Click **Window** → **Show View** → **Other**.
The **Show View** window appears.



3. Enter **Server** in the search field.

4. Select **Servers**.

5. Click **Open**.

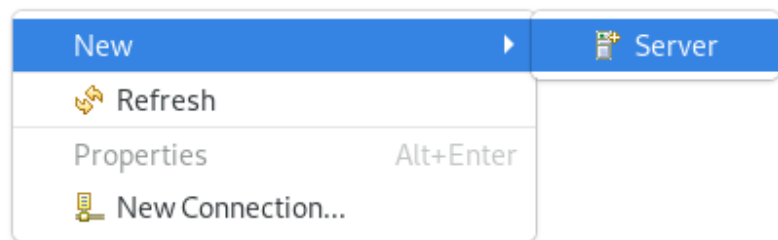
The **Servers** view appears.



6. Right-click any area in the **Servers** view.

Tasks Palette Properties Servers

[No servers are available. Click this link to create a new server...](#)



- Click **New** → **Server**.

The **Define a New Server** window appears.

New Server

Define a New Server

Choose the type of server to create

Select the server type:

type filter text

Red Hat CodeReady Containers 1.0+

Red Hat Container Development Kit 2.x

Red Hat Container Development Kit 3

Red Hat Container Development Kit 3.2+

Red Hat Fuse 7+ Server

Red Hat JBoss Enterprise Application Platform 4.3 (End Of Life)

Integration and support for Red Hat CodeReady Containers 1.0+

Server's host name: localhost

Server name: CodeReady Containers 1.0+

?

< Back

Next >

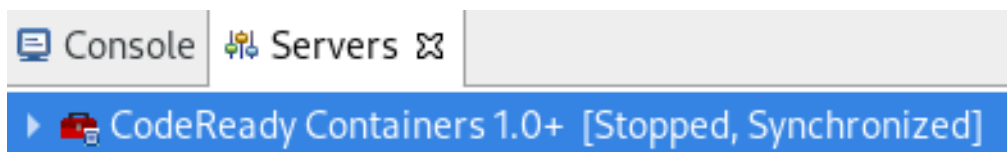
Cancel

Finish

8. Select **CodeReady Containers 1.0+**.
9. Click **Next**.
The **CodeReady Containers** window appears.

10. Click **Browse** to locate the **CRC binary**.
11. Click **Browse** to locate the **CRC Pull Secret File**.
12. Click **Finish**.

Your newly added CodeReady Containers 1.0+ server adapter is now listed in the **Servers** view.



1.1.2. Using the OpenShift Container Platform tooling

The following section describes how to use OpenShift Containers in CodeReady Studio.

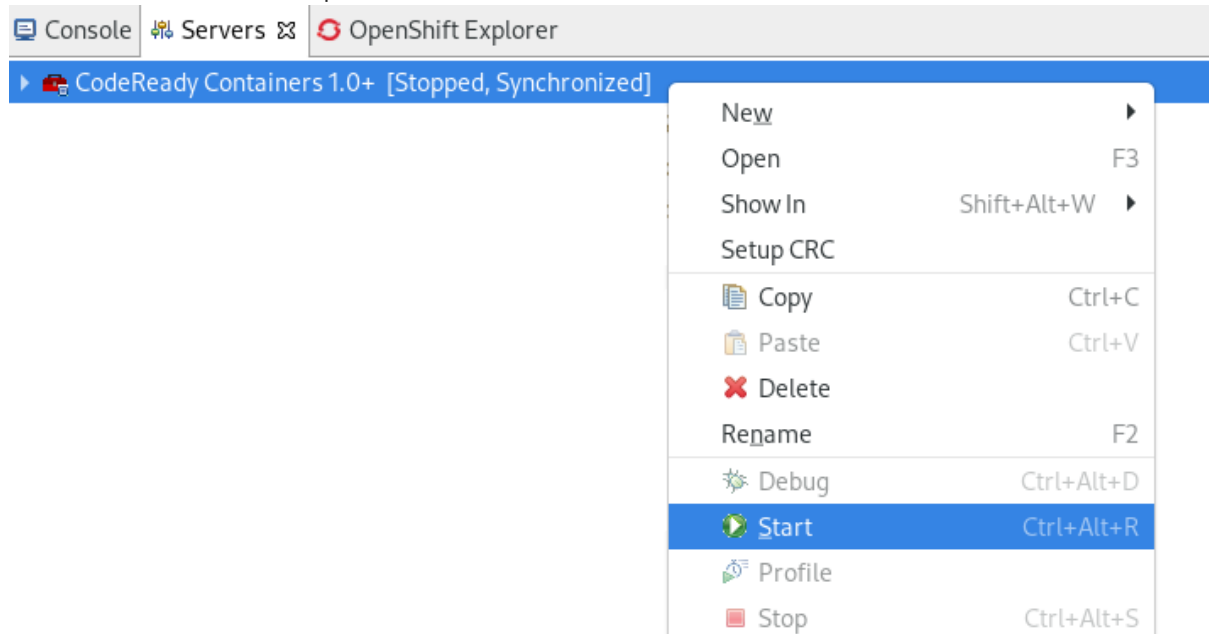
Prerequisites

- Set up and configure the CRC server adapter.
For more information, see [Section 1.1.1, "Downloading and installing Red Hat CodeReady Containers in CodeReady Studio"](#).

Procedure

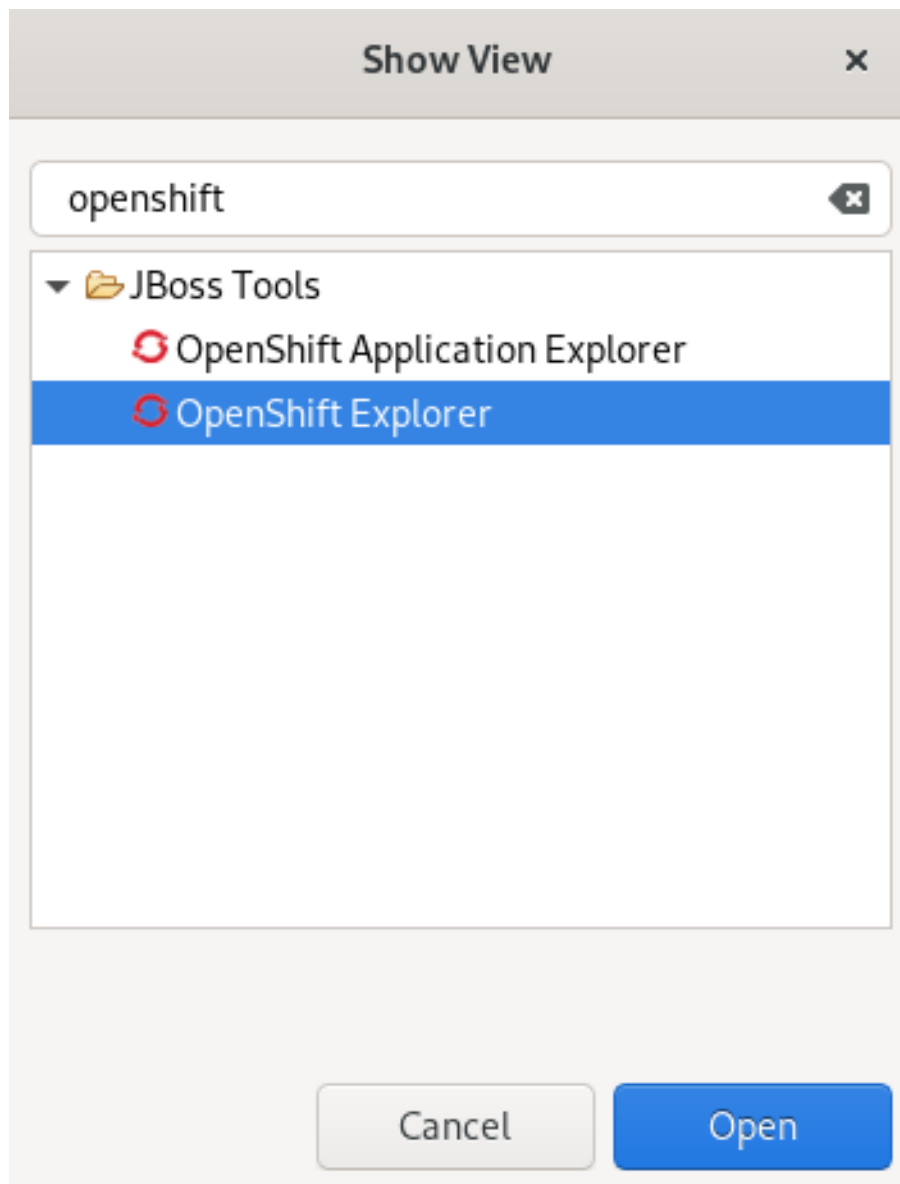
1. Start CodeReady Studio.

2. Start the CRC server adapter.

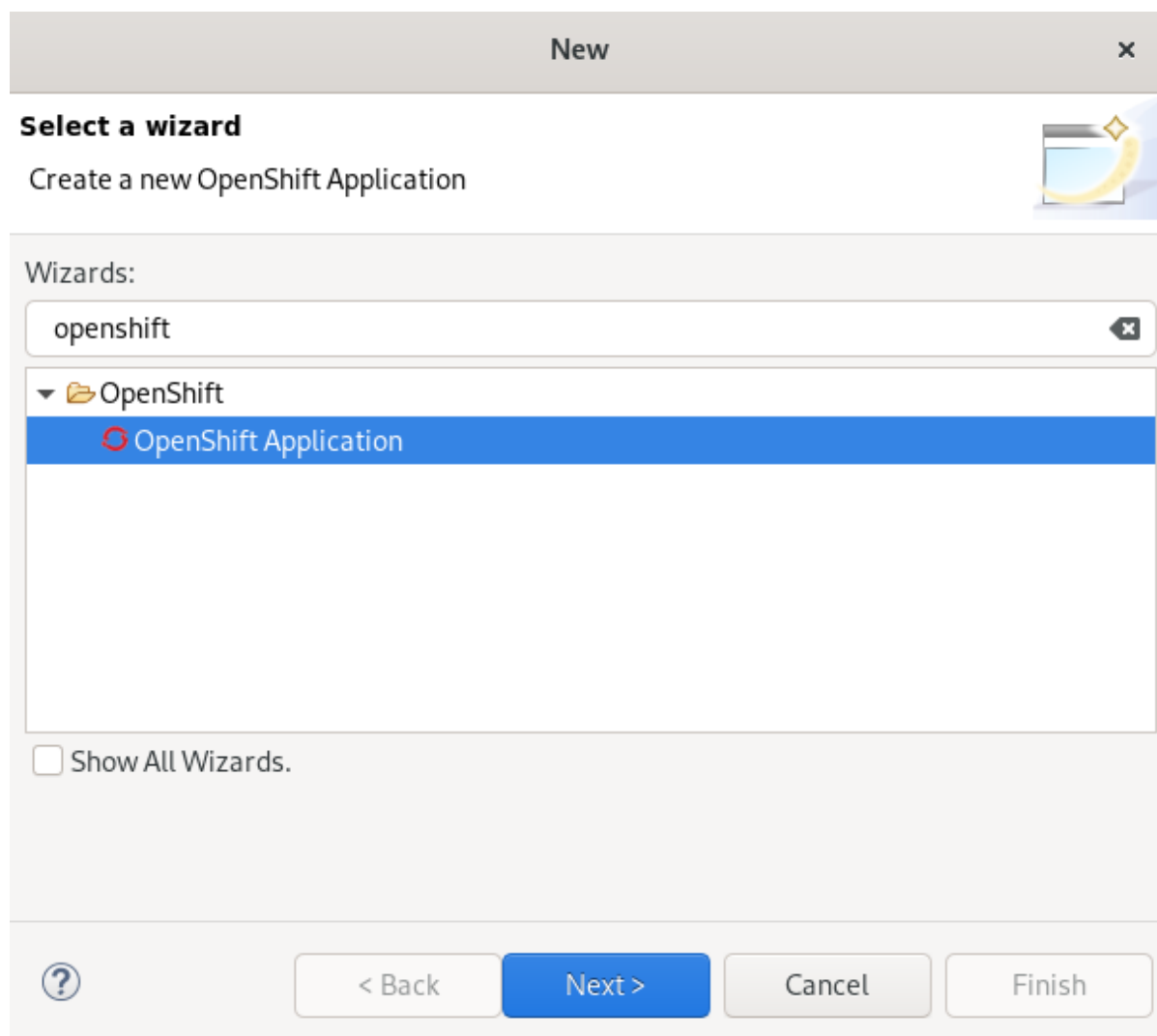


3. Click **Window** → **Show View** → **Other**.

The **Show View** window appears.




4. Enter **OpenShift** in the search field.
5. Select **OpenShift Explorer**.
6. Click **Open**.
The **OpenShift Explorer** view appears.
7. Press **Ctrl+N**.
The **Select a wizard** window appears.




8. Enter **OpenShift** in the search field.
9. Select **OpenShift Application**.
10. Click **Next**.
The **Sign in to OpenShift** window appears.

New OpenShift Application

Sign in to OpenShift


OPENSIFT

 OpenShift client oc wasn't recognized. You may download and/or configure a different OpenShift client.

Want to try OpenShift online? You can sign up for an account [here](#)

Connection: developer - https://api.crc.testing:6443

Server: https://api.crc.testing:6443

Paste Login Command

Authentication


Protocol: Basic

Username: developer

Password: ●●●●●●●●

☒ Save password (could trigger secure storage login)

Advanced >>



< Back

Next >

Cancel


Finish

11. Click **Next**.
The **Create OpenShift Project** window appears.
12. Name your project.
13. Click **Finish**.
The **Select template** window appears.

New OpenShift Application

Select template

Server template choices may be filtered by typing the name of a tag in the text field.


OPENSHIFT

OpenShift project: my-openshift-project

New...

Refresh...

Eclipse Project:

Browse...

Server application source

Custom template

dotnet

⚡ dotnet-example (quickstart, dotnet, .net) - openshift

⚡ dotnet-pgsql-persistent (quickstart, dotnet) - openshift

📦 dotnet:2.1 (builder, .net, dotnet, dotnetcore, rh-dotnet21) - openshift

📦 dotnet:3.0 (builder, .net, dotnet, dotnetcore, rh-dotnet30) - openshift

📦 dotnet:3.1 (builder, .net, dotnet, dotnetcore, rh-dotnet31) - openshift

📦 dotnet:latest (builder, .net, dotnet, dotnetcore) - openshift

Details

📦 An example .NET Core application.

Defined Resources...

?

< Back

Next >

Cancel

Finish

14. Select the template.


15. Click **Next**.

The **Build Configuration** window appears.

10

New OpenShift Application
×

Build Configuration


OPENSHIFT

Name:

Git Repository URL:

Git Reference:

Context Directory:

Build Triggers:

- ☒ Configure a webhook build trigger
- ☒ Automatically build a new image when the builder image changes
- ☒ Automatically build a new image when the build configuration changes

Build environment variables (Build and Runtime):

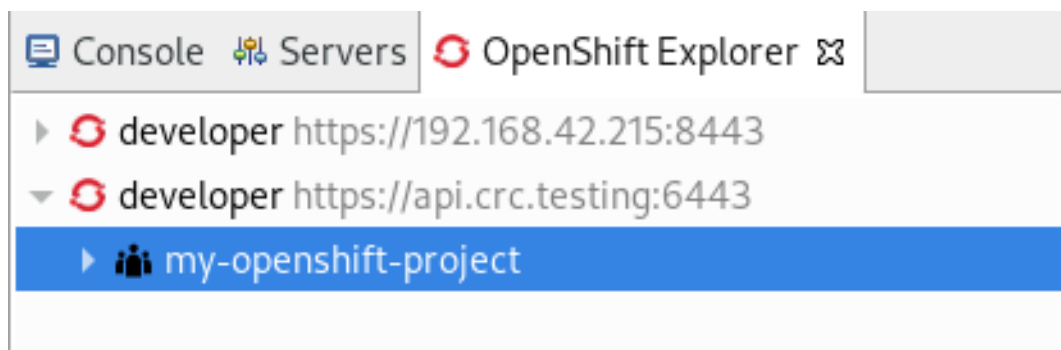
Name	Value

?

16. Ensure that the build configurations are correct.

17. Click **Finish**.

Your newly created OpenShift application project is now listed in the **OpenShift Explorer** view.



Additional resources

- For more information on how to perform additional tasks with the OpenShift Container Platform projects and application, see [Developing for the Cloud with OpenShift](#).

1.2. USING RED HAT CONTAINER DEVELOPMENT KIT TOOLING IN CODEREADY STUDIO

Red Hat Container Development Kit (CDK) is a pre-built container development environment based on Red Hat Enterprise Linux (RHEL). CDK helps you get started with developing container-based applications quickly. You can easily set up CDK within CodeReady Studio.

1.2.1. Installing Container Development Kit in CodeReady Studio

The following section describes how to install CDK from within CodeReady Studio.

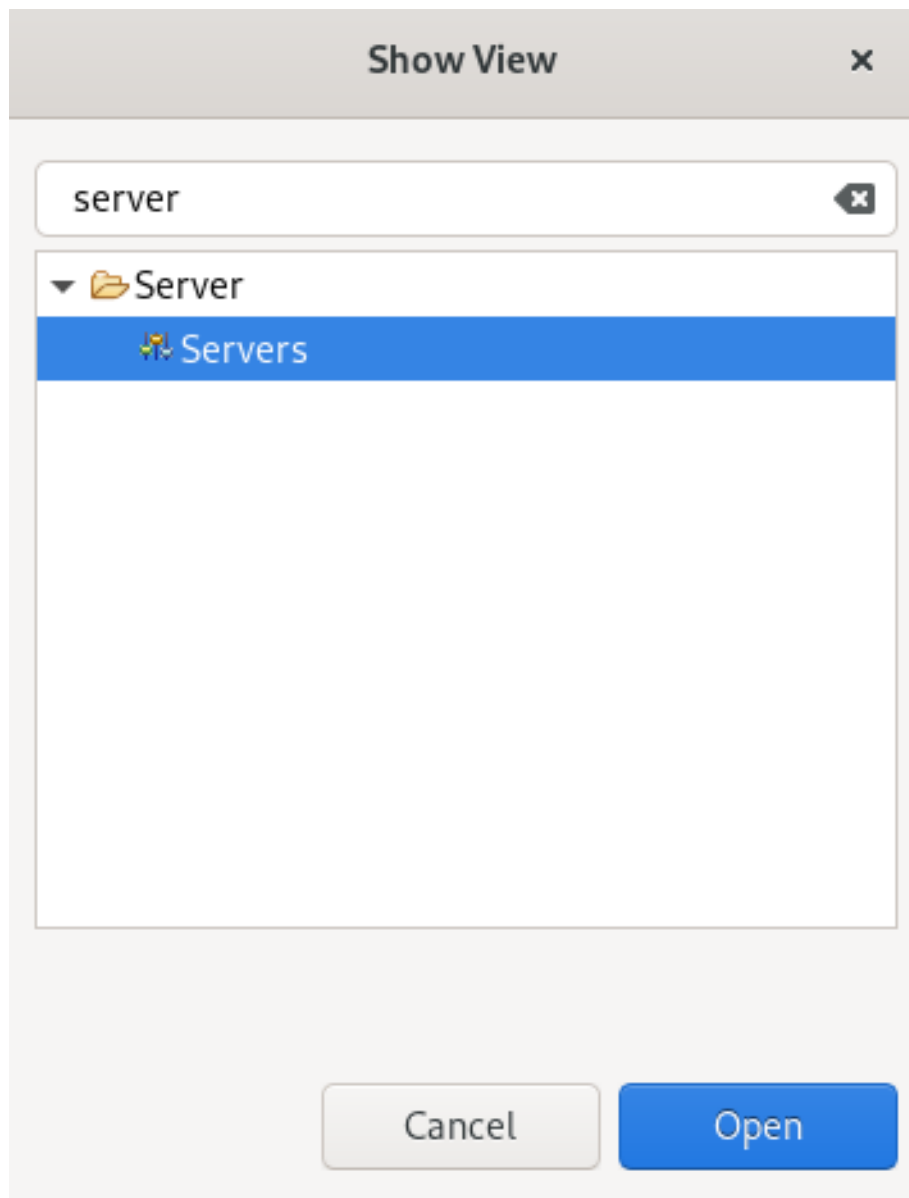
Prerequisites

- Ensure that the Hypervisor is installed and configured on your system:
 - VirtualBox, Linux KVM/libvirt (Linux)
 - xhyve (macOS)
 - Hyper-V (Windows)
- Ensure that hardware virtualization is enabled on your system.
For more information, see [Setting Up the Virtualization Environment](#)
- Ensure that you have a Red Hat Developer account.
For a new account, visit <https://developers.redhat.com/>.

For more information on CDK, see the [Red Hat Container Development Kit Getting Started Guide](#) .

Procedure

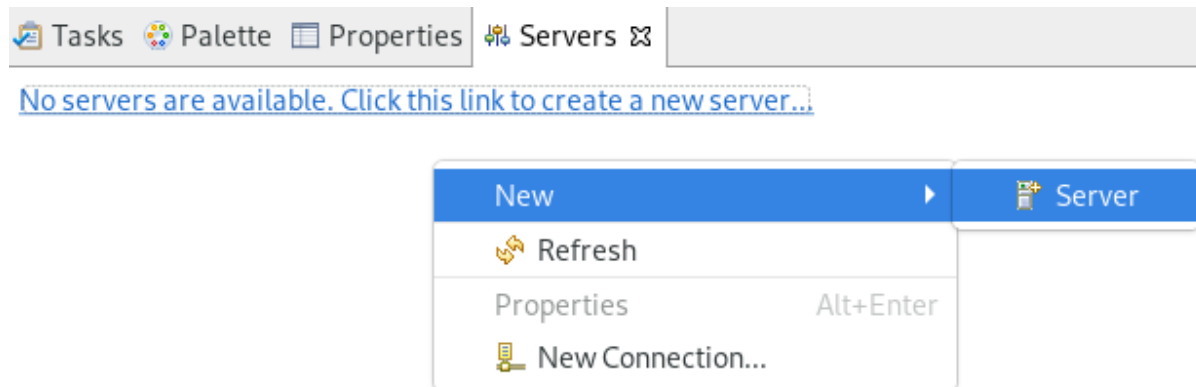
1. Start CodeReady Studio.
2. Click **Window → Show View → Other**.
The **Show View** window appears.



3. Enter **Server** in the search field.
4. Select **Servers**.
5. Click **Open**.
The **Servers** view appears.

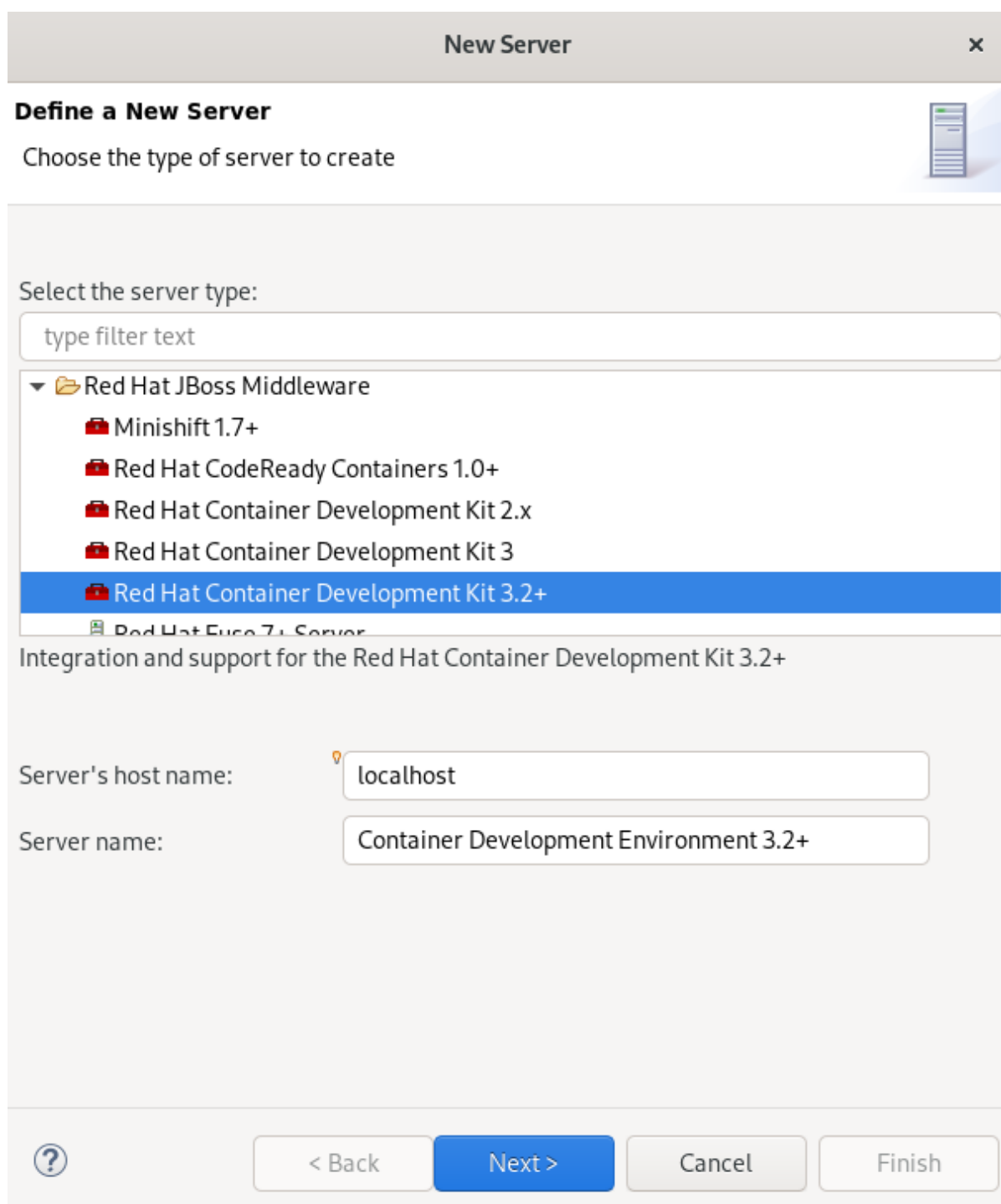


6. Right-click any area in the **Servers** view.



7. Click **New** → **Server**.

The **Define a New Server** window appears.



8. Select **Red Hat Container Development Kit 3.2+**.
9. Click **Next**.
The **Red Hat Container Development Environment** window appears.

here if you do not have one already.' Below this are several input fields: 'Domain:' with a dropdown showing 'access.redhat.com'; 'Username:' with a dropdown showing 'developer@redhat.com', an 'Edit...' button, and an 'Add...' button; 'Hypervisor:' with a dropdown showing 'kvm'; 'Minishift Binary:' with an empty text field and a 'Browse...' button; 'Minishift Home:' with a text field containing '/home/ developer / .minishift' and a 'Browse...' button; and 'Minishift Profile:' with a text field containing 'minishift'. A blue link 'Download and install runtime...' is positioned above the 'Minishift Binary' field. At the bottom, there is a help icon (question mark in a circle) and four buttons: '< Back', 'Next >', 'Cancel', and 'Finish'." data-bbox="154 128 911 566"/>

New Server ×

Red Hat Container Development Environment

A server adapter representing Red Hat Container Development Kit Version 3.2+

Register a Red Hat account [here](#) if you do not have one already.

Domain:

Username:

Hypervisor:

[Download and install runtime...](#)

Minishift Binary:

Minishift Home:

Minishift Profile:

10. Click **Download and install runtime**.
The **Download Runtimes** window appears.

Download Runtimes ✕

Download Runtimes

Please select a runtime to download and install.

Name	Version
Red Hat CDK v3.4.0	3.4.0
Red Hat CDK v3.5.0	3.5.0
Red Hat CDK v3.6.0	3.6.0
Red Hat CDK v3.7.0	3.7.0
Red Hat CDK v3.8.0	3.8.0
Red Hat CDK v3.9.0	3.9.0
Red Hat CDK v3.10.0	3.10.0
Red Hat CDK v3.11.0	3.11.0
Red Hat CDK v3.12.0	3.12.0
Red Hat CDK v3.13.0	3.13.0

Selected Runtime Details

Project URL: <https://developers.redhat.com/products/cdk/download/>

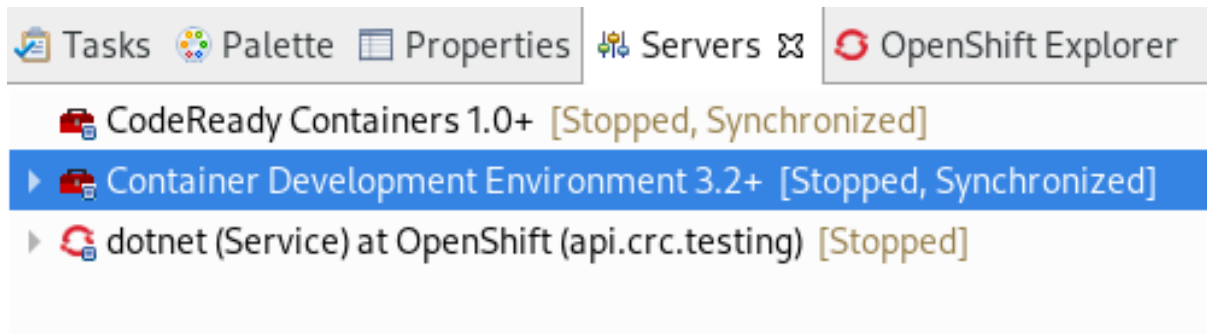
Download URL: <https://www.jboss.org/download-manager/jdf/file/cdk-3.13.0-1-minishift-RedHatCDK4>

Registration required. Downloads require accepting the terms and conditions of the JBoss Developer Program which provides 60 subscriptions for development use only.

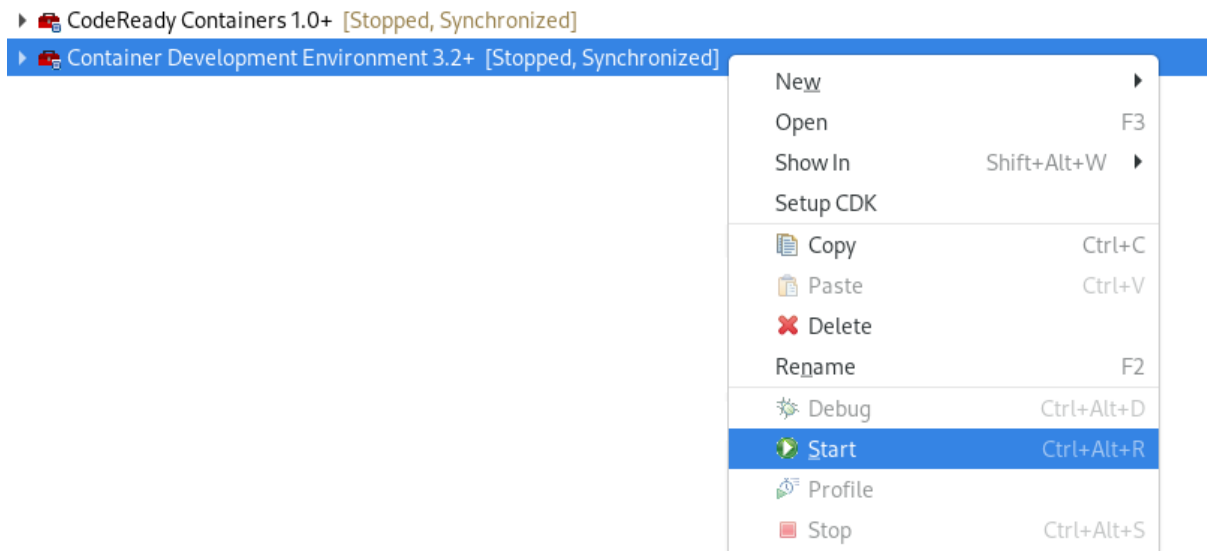
?

< Back
Next >
Cancel
Finish

11. Select a **Red Hat CDK** version.
12. Click **Next**.
13. Ensure that your sign-on credentials for access.redhat.com are correct.
14. Click **Next**.
15. Review and accept the license agreement and click **Next**.
16. Select the installation folder and click **Finish**.
Note that the process of downloading and installing the runtime might take some time to complete.
17. Click **Finish**.
Your newly created **Container Development Environment 3.2+** server is now listed in the **Servers** view.

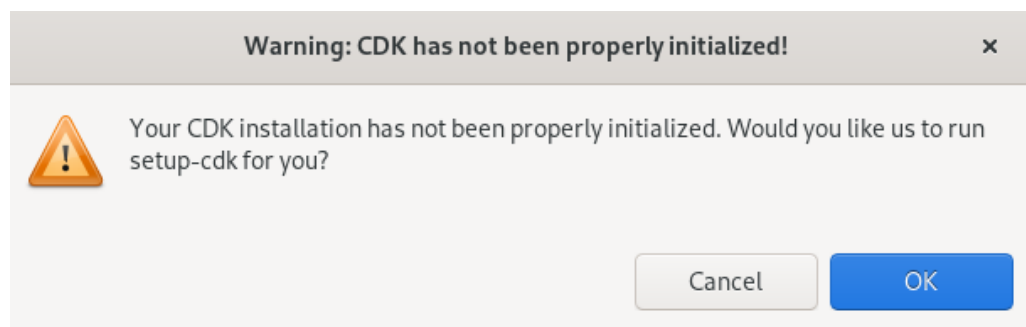


18. Right-click **CDK server adapter** → **Start**.



NOTE

In case you did not set up CDK prior to starting the server adapter, you will see a warning: **CDK has not been properly initialized!**



Follow the on-screen instructions to initialize CDK.

1.2.2. Using Docker tooling

After starting the CDK server in the IDE, you can follow one of the two container development workflows to use the Docker tooling.

1.2.2.1. Creating a Dockerfile

Prerequisites

- Set up and configure the CDK server adapter.

For more information, see [Section 1.2.1, “Installing Container Development Kit in CodeReady Studio”](#).

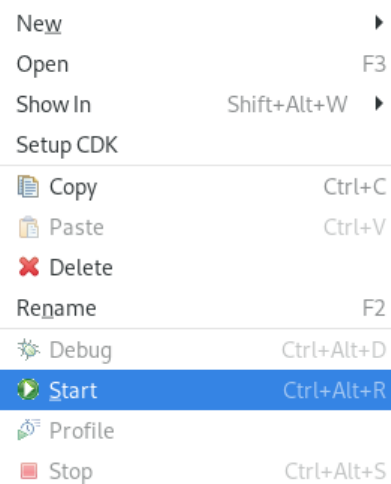
Procedure

1. Start CodeReady Studio.

2. Start the CDK server adapter.

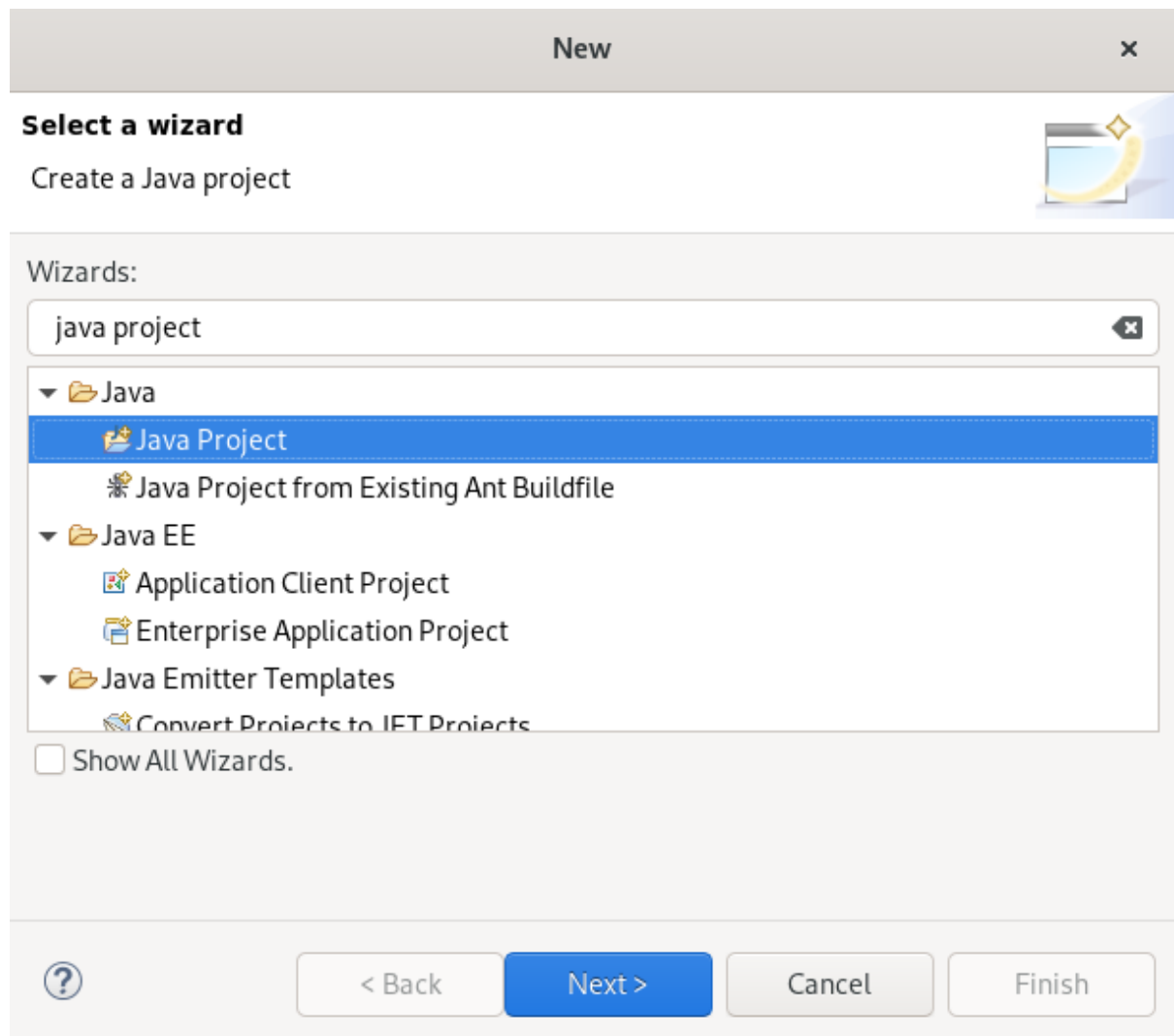
▶  CodeReady Containers 1.0+ [Stopped, Synchronized]

▶  Container Development Environment 3.2+ [Stopped, Synchronized]

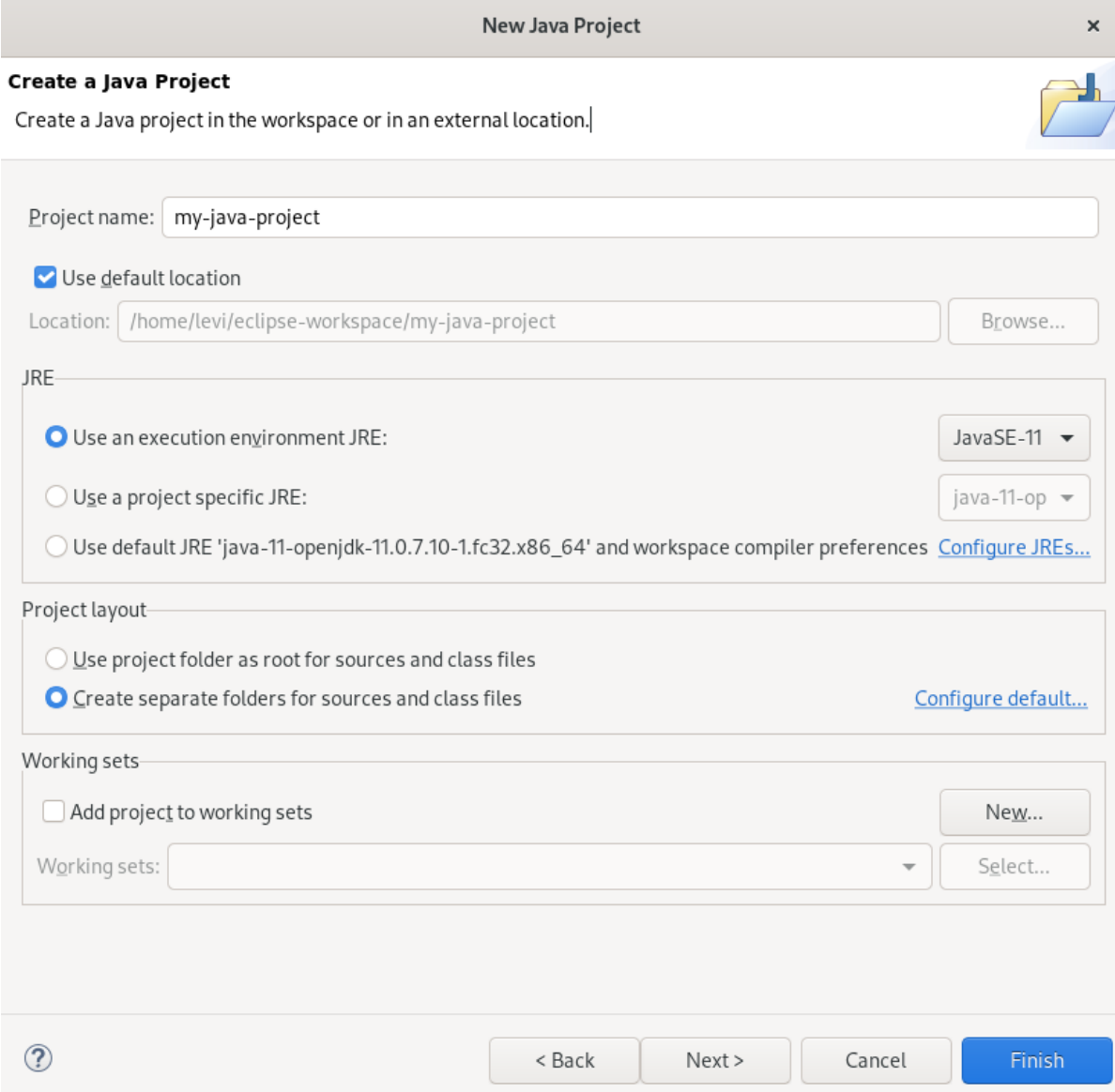


3. Press **Ctrl+N**.

The **Select a wizard** window appears.



4. Enter **Java Project** in the search field.
5. Select **Java Project**.
6. Click **Next**.
The **New Java Project** window appears.



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name:

☒ Use default location

Location:

JRE

☒ Use an execution environment JRE:

☐ Use a project specific JRE:

☐ Use default JRE 'java-11-openjdk-11.0.7.10-1.fc32.x86_64' and workspace compiler preferences [Configure JREs...](#)

Project layout

☐ Use project folder as root for sources and class files

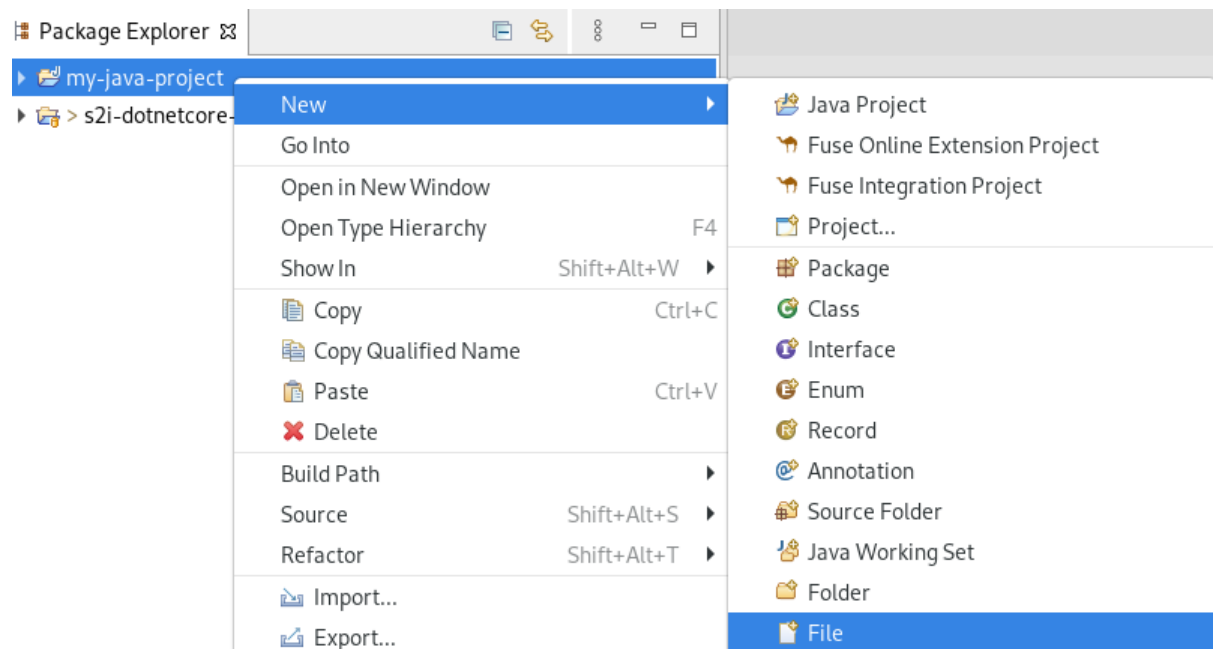
☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

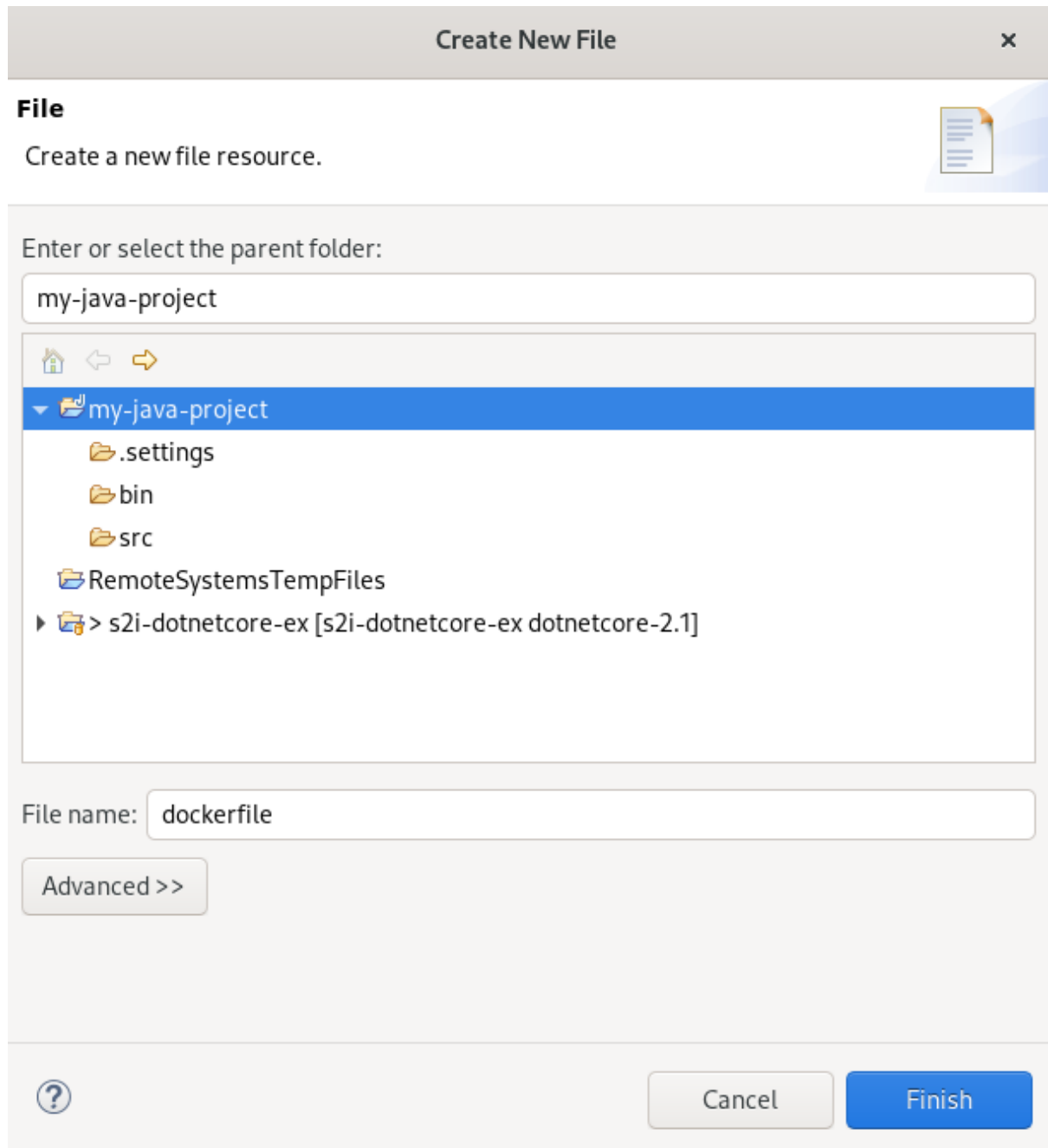
☐ Add project to working sets

Working sets:

7. Name your project.
8. Select the location for your project.
9. Click **Finish**.
Your newly created Java project is now listed in the CodeReady Studio view.
10. Right-click your **Java project** → **New** → **File**.



The **Create New File** window appears.



11. Select the parent folder.
12. Name your file.
13. Click **Finish**.
Your newly created file is now displayed in the CodeReady Studio editor.
14. Paste the following content into your newly created file:

```
# Use latest jboss/base-jdk:8 image as the base
FROM jboss/base-jdk:8

# Set the WILDFLY_VERSION env variable
ENV WILDFLY_VERSION 10.1.0.Final
ENV WILDFLY_SHA1 9ee3c0255e2e6007d502223916cefad2a1a5e333
ENV JBOSS_HOME /opt/jboss/wildfly

USER root
```

```

# Add the WildFly distribution to /opt, and make wildfly the owner of the extracted tar content
# Make sure the distribution is available from a well-known place
RUN cd $HOME \
  && curl -O https://download.jboss.org/wildfly/$WILDFLY_VERSION/wildfly-
$WILDFLY_VERSION.tar.gz \
  && sha1sum wildfly-$WILDFLY_VERSION.tar.gz | grep $WILDFLY_SHA1 \
  && tar xf wildfly-$WILDFLY_VERSION.tar.gz \
  && mv $HOME/wildfly-$WILDFLY_VERSION $JBOSS_HOME \
  && rm wildfly-$WILDFLY_VERSION.tar.gz \
  && chown -R jboss:0 ${JBOSS_HOME} \
  && chmod -R g+rw ${JBOSS_HOME}

# Ensure signals are forwarded to the JVM process correctly for graceful shutdown
ENV LAUNCH_JBOSS_IN_BACKGROUND true

USER jboss

# Expose the ports we're interested in
EXPOSE 8080

# Set the default command to run on boot
# This will boot WildFly in the standalone mode and bind to all interface
CMD ["/opt/jboss/wildfly/bin/standalone.sh", "-b", "0.0.0.0"]

```

15. Press **Ctrl+S** to save the changes.

Additional resources

- For more information about Dockerfiles, see [Dockerfile reference](#).

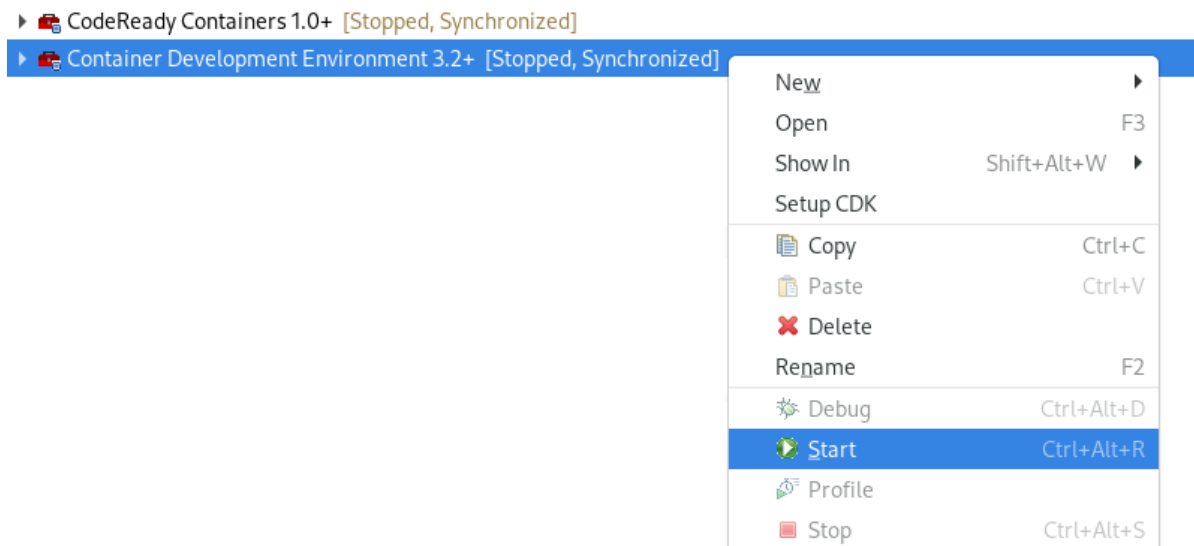
1.2.2.2. Building the Docker image Using the Container Development Environment

Prerequisites

- Set up and configure the CDK server adapter.
For more information, see [Section 1.2.1, "Installing Container Development Kit in CodeReady Studio"](#).
- Create a Java project and a Dockerfile.
For more information, see [Section 1.2.2.1, "Creating a Dockerfile"](#)

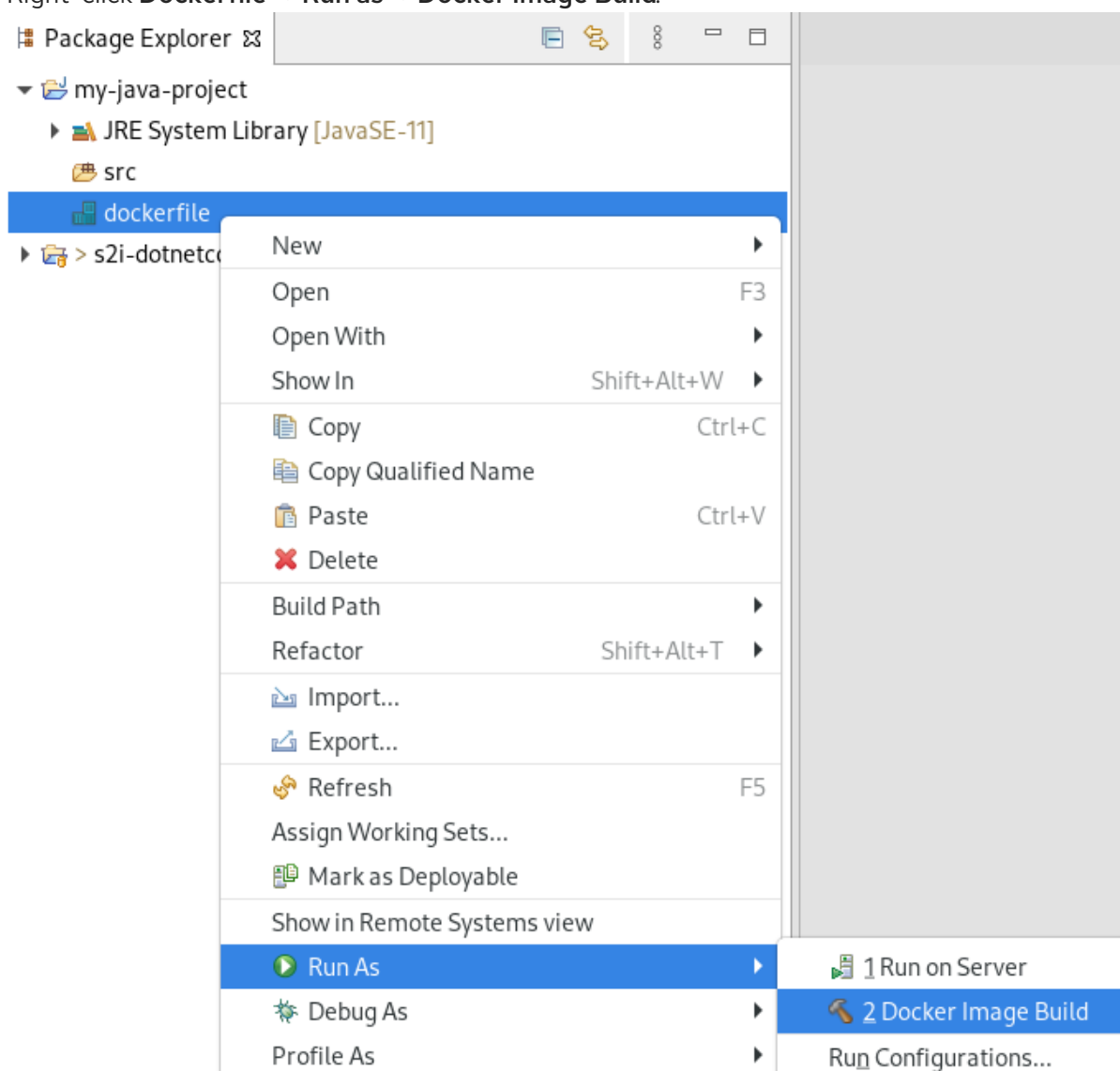
Procedure

1. Start CodeReady Studio.
2. Start the CDK server adapter.

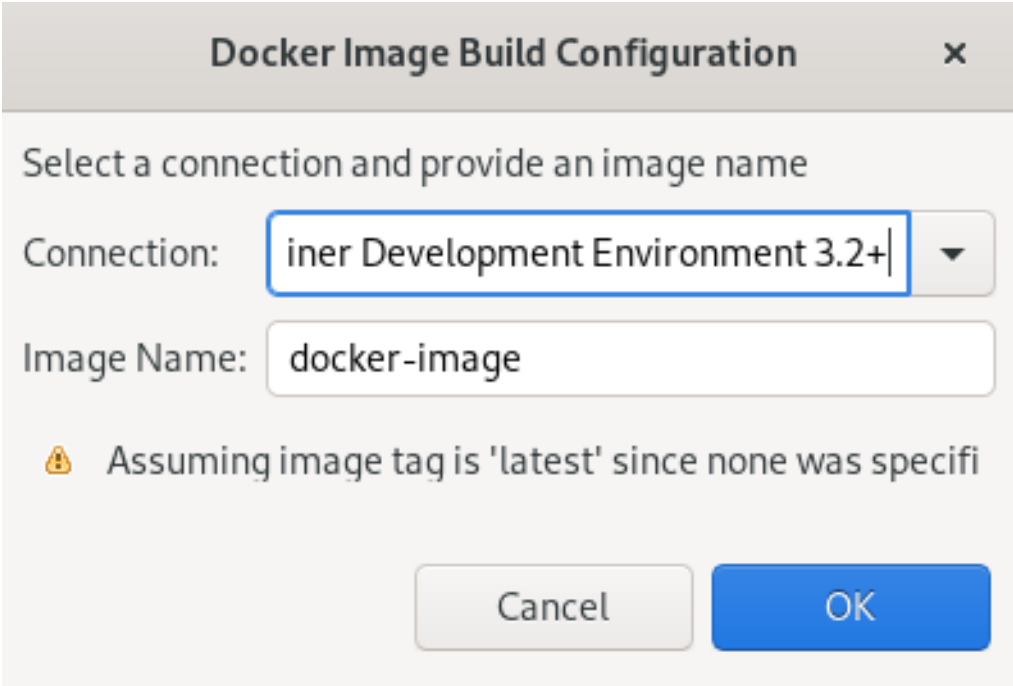


3. Expand your Java project.

4. Right-click **Dockerfile** → **Run as** → **Docker Image Build**.



The **Docker Image Build Configuration** window appears.



The screenshot shows a dialog box titled "Docker Image Build Configuration" with a close button (X) in the top right corner. The dialog contains the instruction "Select a connection and provide an image name". There are two input fields: "Connection:" with a dropdown menu showing "iner Development Environment 3.2+" and "Image Name:" with a text box containing "docker-image". Below these fields is a warning message with a yellow triangle icon: "Assuming image tag is 'latest' since none was specifi". At the bottom are "Cancel" and "OK" buttons.

5. Select your Container Development Environment server adapter as your connection.
6. Name your image.
7. Click **OK**.

The **Console** view appears displaying the docker image building process.

1.2.2.3. Additional resources

- For more information about the basics of Docker Tooling, see [Using Docker Tooling in CodeReady Studio](#).

1.2.3. Using the OpenShift Container Platform tooling

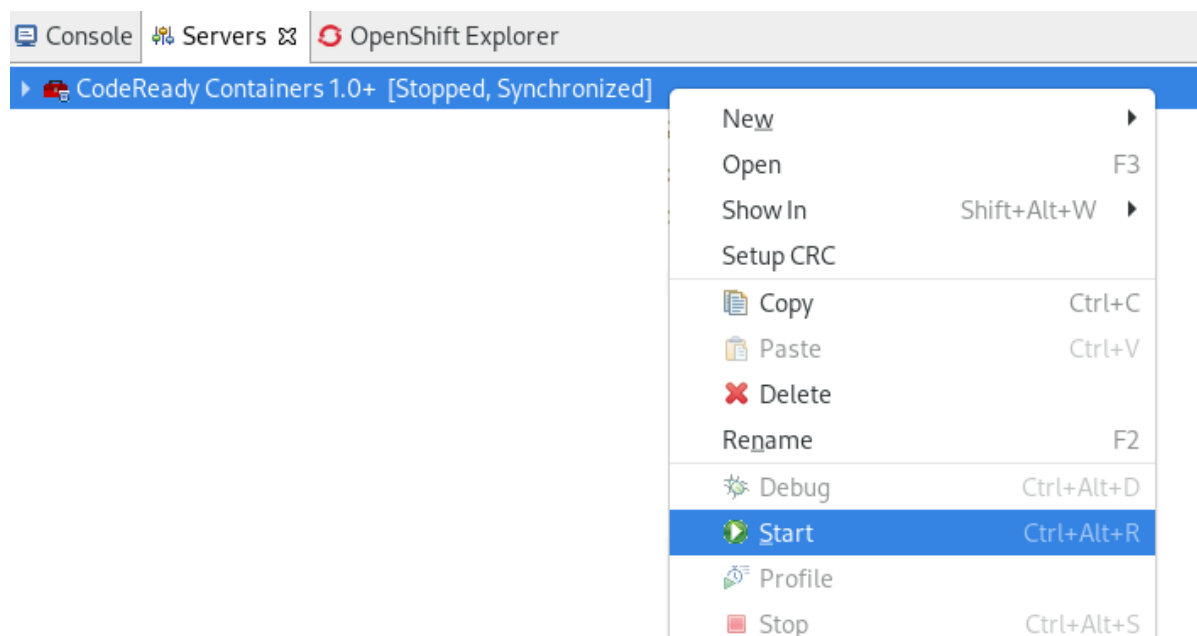
The following section describes how to use OpenShift Containers in CodeReady Studio.

Prerequisites

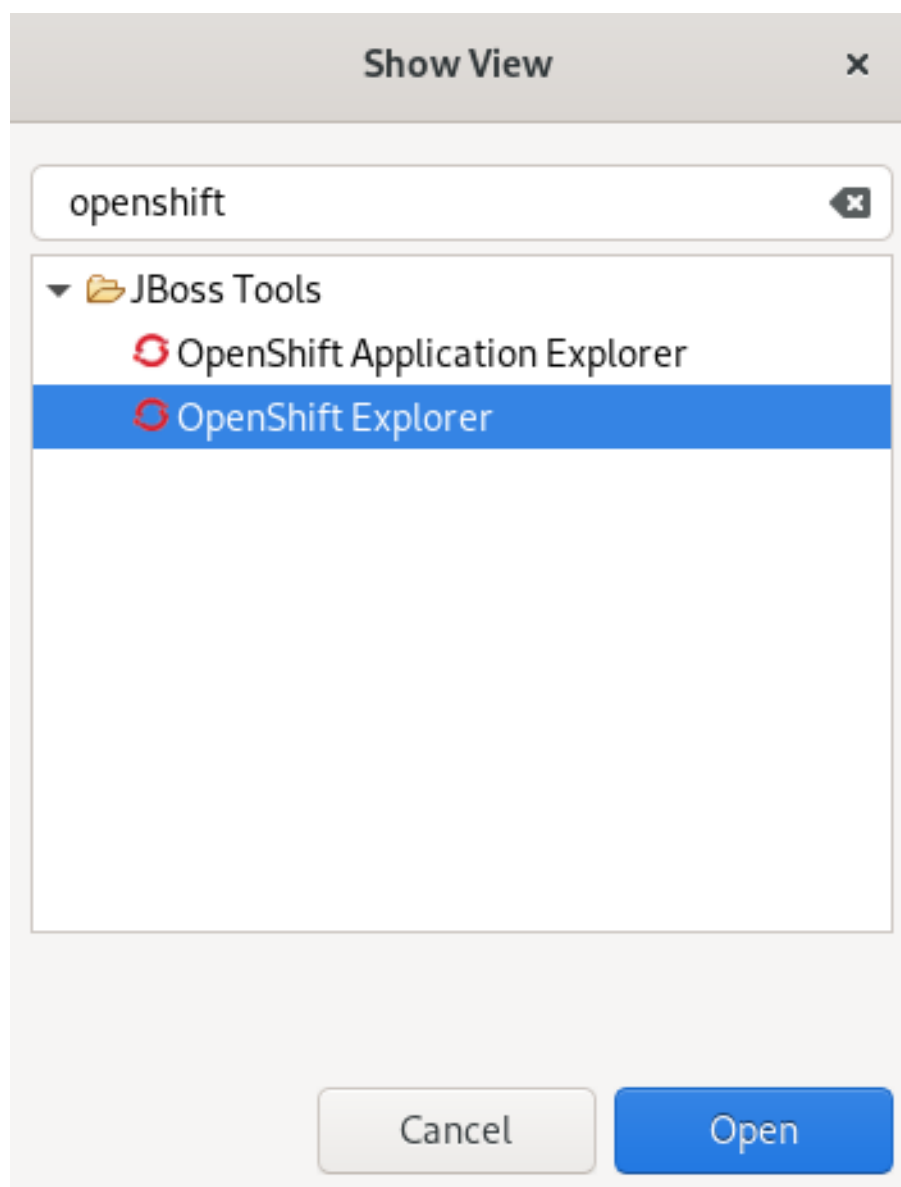
- Set up and configure the CRC server adapter.
For more information, see [Section 1.1.1, "Downloading and installing Red Hat CodeReady Containers in CodeReady Studio"](#).

Procedure

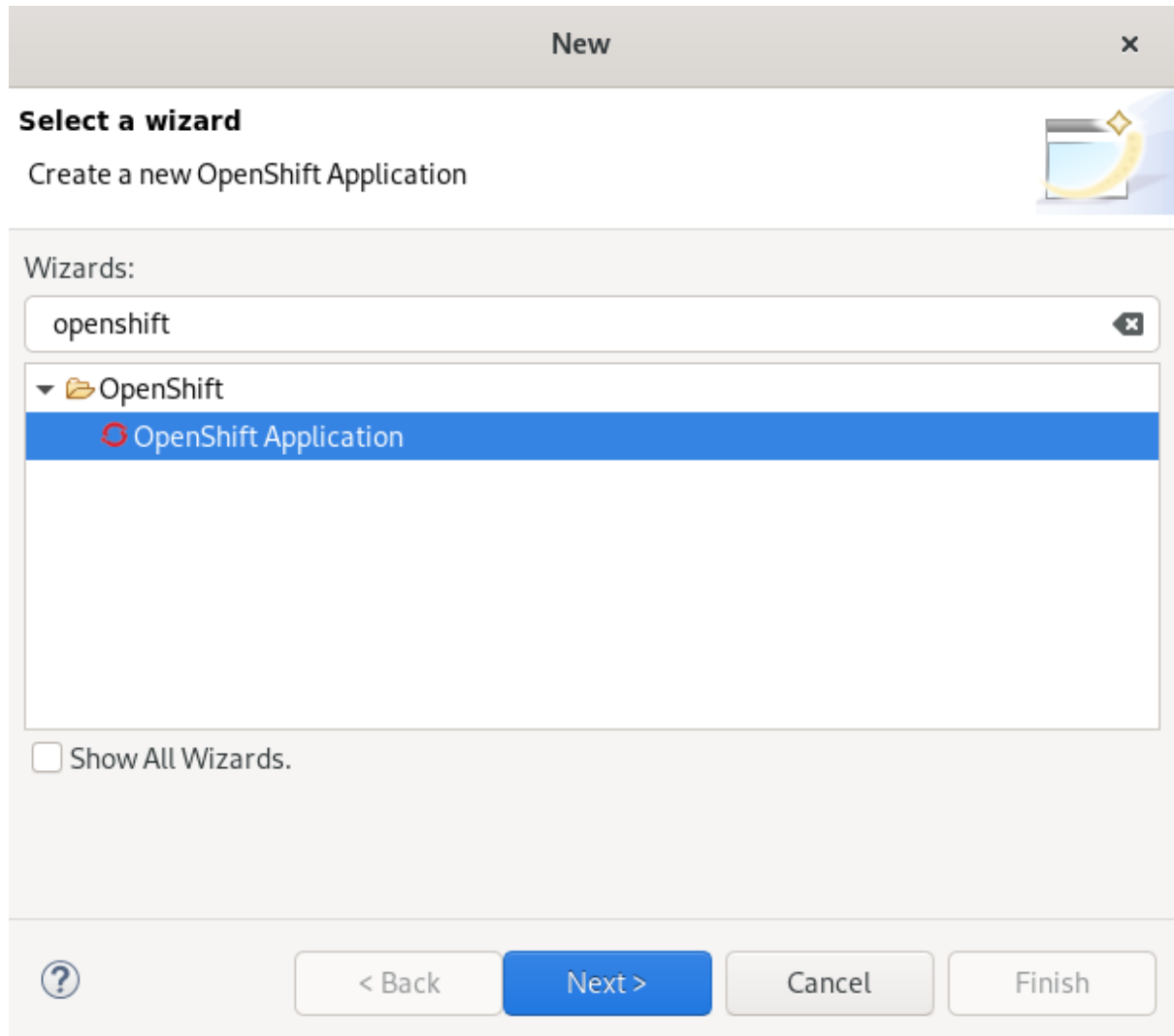
1. Start CodeReady Studio.
2. Start the CRC server adapter.



3. Click **Window** → **Show View** → **Other**.
The **Show View** window appears.




4. Enter **OpenShift** in the search field.
5. Select **OpenShift Explorer**.
6. Click **Open**.
The **OpenShift Explorer** view appears.
7. Press **Ctrl+N**.
The **Select a wizard** window appears.




8. Enter **OpenShift** in the search field.
9. Select **OpenShift Application**.
10. Click **Next**.
The **Sign in to OpenShift** window appears.

New OpenShift Application

Sign in to OpenShift


OPENSIFT

 OpenShift client oc wasn't recognized. You may download and/or configure a different OpenShift client.

Want to try OpenShift online? You can sign up for an account [here](#)

Connection: developer - https://api.crc.testing:6443

Server: https://api.crc.testing:6443 Paste Login Command

Authentication


Protocol: Basic

Username: developer

Password: ●●●●●●●●

☒ Save password (could trigger secure storage login)

Advanced >>



< Back

Next >

Cancel

Finish

11. Click **Next**.
The **Create OpenShift Project** window appears.
12. Name your project.
13. Click **Finish**.
The **Select template** window appears.

28

New OpenShift Application

Select template

Server template choices may be filtered by typing the name of a tag in the text field.

OpenShift project: my-openshift-project

New...

Refresh...

Eclipse Project:

Browse...

Server application source

Custom template

dotnet

dotnet-example (quickstart, dotnet, .net) - openshift

dotnet-pgsql-persistent (quickstart, dotnet) - openshift

dotnet:2.1 (builder, .net, dotnet, dotnetcore, rh-dotnet21) - openshift

dotnet:3.0 (builder, .net, dotnet, dotnetcore, rh-dotnet30) - openshift

dotnet:3.1 (builder, .net, dotnet, dotnetcore, rh-dotnet31) - openshift

dotnet:latest (builder, .net, dotnet, dotnetcore) - openshift

Details

An example .NET Core application.

Defined Resources...

< Back

Next >

Cancel

Finish


14. Select the template.

15. Click **Next**.

The **Build Configuration** window appears.

29

New OpenShift Application
×

Build Configuration


Name:

Git Repository URL:

Git Reference:

Context Directory:

Build Triggers:

- ☒ Configure a webhook build trigger
- ☒ Automatically build a new image when the builder image changes
- ☒ Automatically build a new image when the build configuration changes

Build environment variables (Build and Runtime):

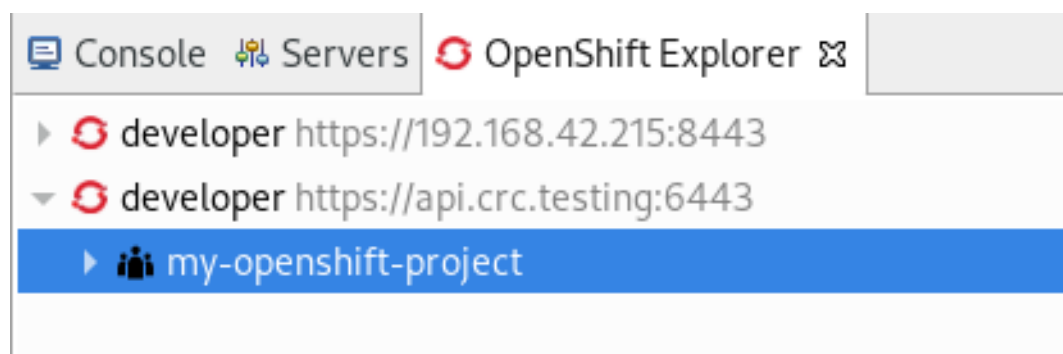
Name	Value

?

16. Ensure that the build configurations are correct.

17. Click **Finish**.

Your newly created OpenShift application project is now listed in the **OpenShift Explorer** view.



Additional resources

- For more information on how to perform additional tasks with the OpenShift Container Platform projects and application, see [Developing for the Cloud with OpenShift](#).

1.2.4. Additional resources

- For more information on how to perform tasks using the OpenShift Container Platform tooling, see [Developing for the Cloud with OpenShift 4](#).
- For more information on how to use OpenShift in CodeReady Studio, see [OpenShift basics in CodeReady Studio](#)

CHAPTER 2. DEVELOPING FOR THE CLOUD WITH OPENSIFT

2.1. CREATING AN OPENSIFT CONTAINER PLATFORM APPLICATION IN CODEREADY STUDIO

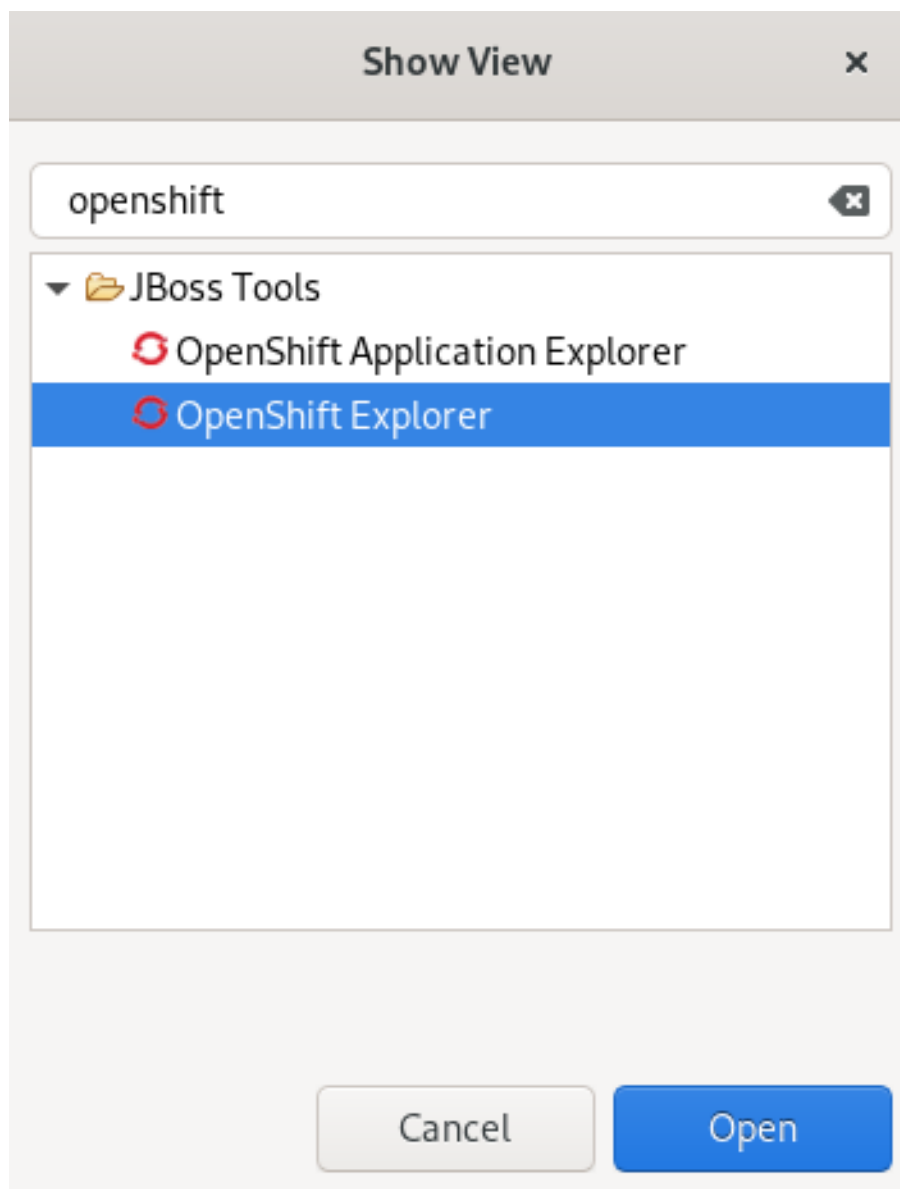
Using the OpenShift Container Platform tooling you can create, import, and modify OpenShift Container Platform applications.

2.1.1. Creating a new OpenShift Container Platform connection

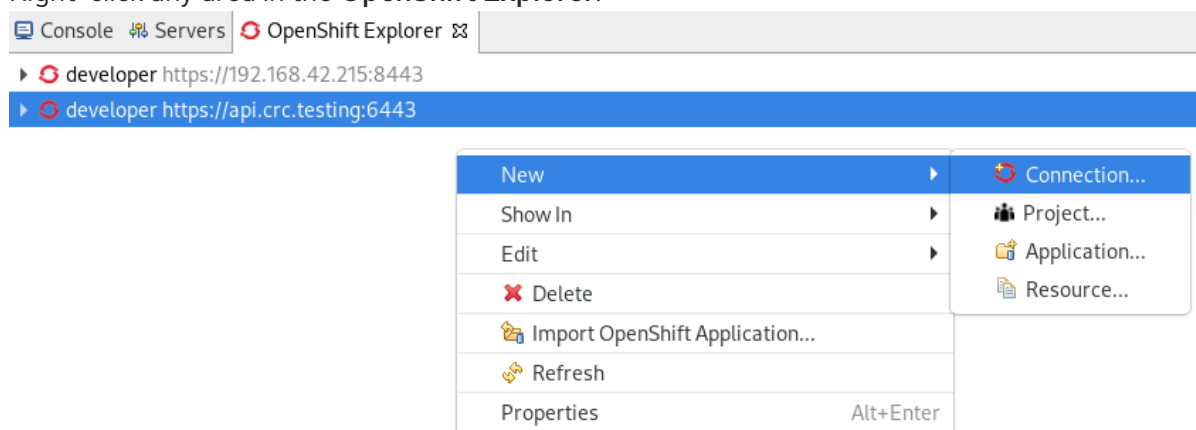
You must create an OpenShift connection in the **OpenShift Explorer** view in CodeReady Studio to use the OpenShift tooling in the IDE. An OpenShift connection connects your IDE to an OpenShift instance (based on CDK, OpenShift Online, Kubernetes, minishift). The connection is listed in the **OpenShift Explorer** view. You can have more than one OpenShift connection configured in the IDE.

Procedure

1. Start CodeReady Studio.
2. Click **Window → Show View → Other**.
The **Show View** window appears.

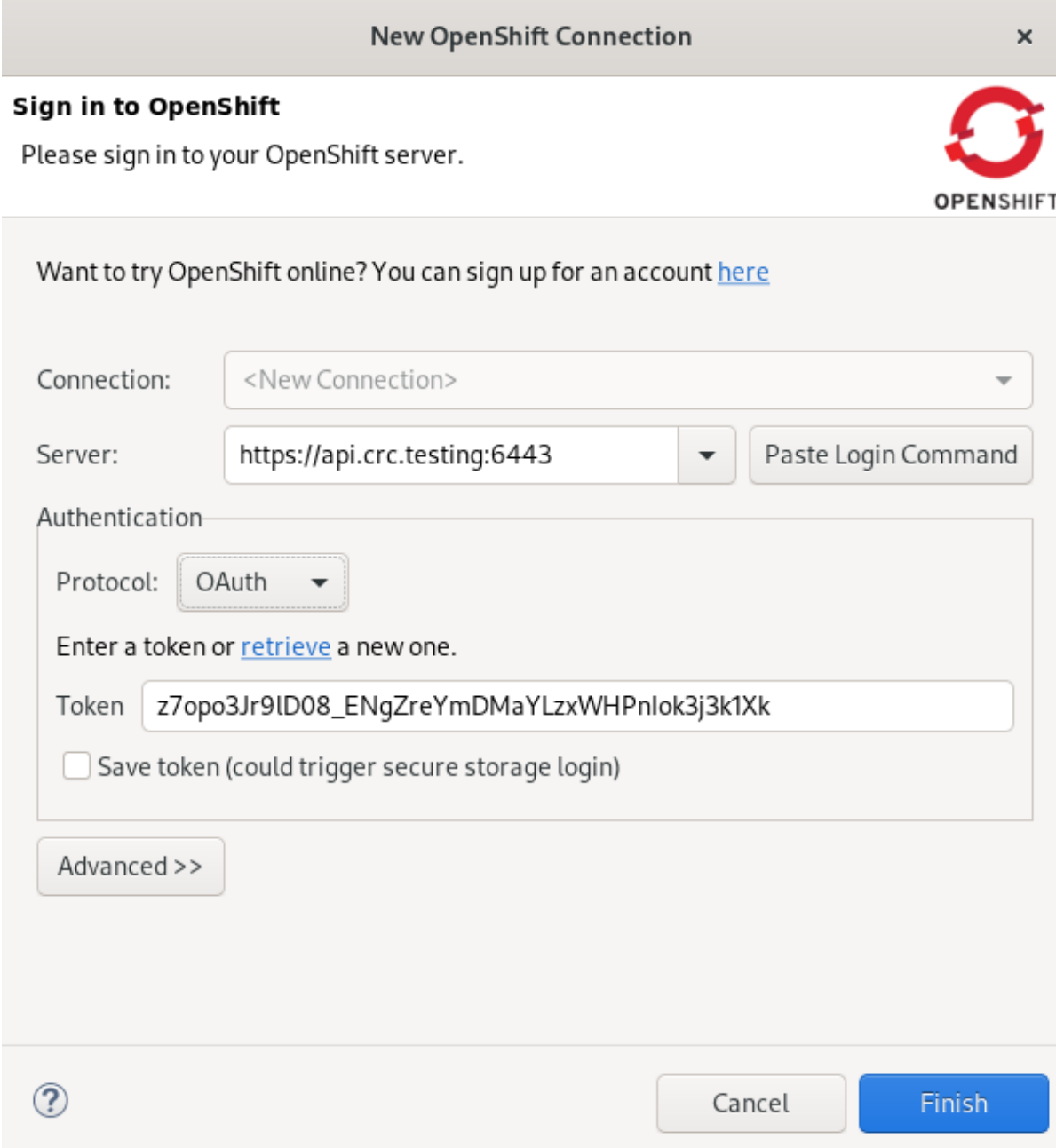


3. Enter **OpenShift** in the search field.
4. Select **OpenShift Explorer**.
5. Click **Open**.
The **OpenShift Explorer** view appears.
6. Right-click any area in the **OpenShift Explorer**.



7. Click **New** → **Connection**.

The **Sign in to OpenShift** window appears.



The image shows a 'New OpenShift Connection' dialog box. At the top, it says 'Sign in to OpenShift' and 'Please sign in to your OpenShift server.' with the OpenShift logo. Below this, there is a link: 'Want to try OpenShift online? You can sign up for an account [here](#)'. The main section contains several fields: 'Connection:' with a dropdown menu showing '<New Connection>', 'Server:' with a text field containing 'https://api.crc.testing:6443' and a 'Paste Login Command' button. Under the 'Authentication' section, there is a 'Protocol:' dropdown set to 'OAuth', a text prompt 'Enter a token or [retrieve](#) a new one.', a 'Token' text field containing 'z7opo3Jr9ID08_ENgZreYmDMaYLzxWHPnlok3j3k1Xk', and a checkbox labeled 'Save token (could trigger secure storage login)'. At the bottom left is an 'Advanced >>' button, and at the bottom right are 'Cancel' and 'Finish' buttons.

New OpenShift Connection

Sign in to OpenShift

Please sign in to your OpenShift server.

Want to try OpenShift online? You can sign up for an account [here](#)

Connection: <New Connection>

Server: https://api.crc.testing:6443 Paste Login Command

Authentication

Protocol: OAuth

Enter a token or [retrieve](#) a new one.

Token z7opo3Jr9ID08_ENgZreYmDMaYLzxWHPnlok3j3k1Xk

☐ Save token (could trigger secure storage login)

Advanced >>

Cancel Finish

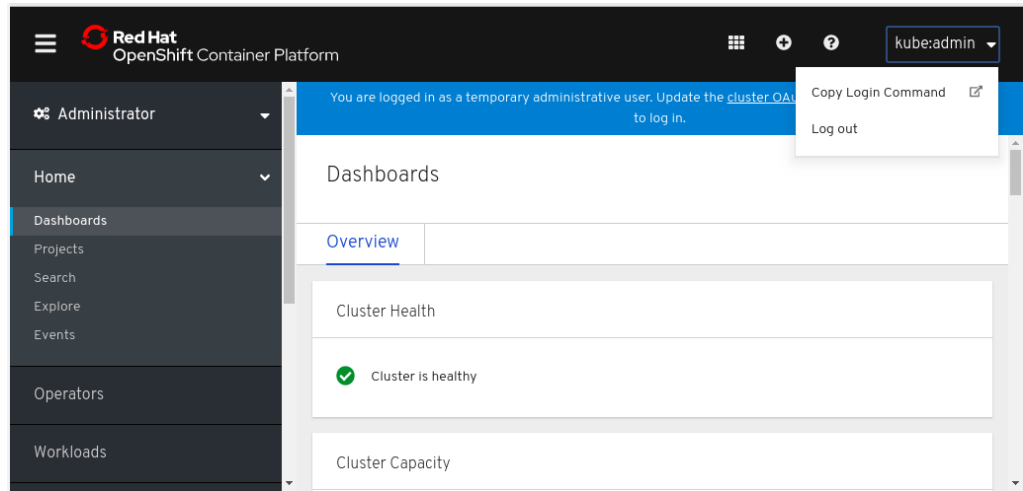
8. Paste the URL for the OpenShift server into the **Server** field.

9. Authenticate with token or login credentials.

NOTE

Alternatively, you can copy the Login Command from the OpenShift Container Platform web UI.

To get login credentials, click the **drop-down menu in the top right corner**→ **Copy Login Command**.



10. Click **Finish**.

Your newly added connection is now listed in the **OpenShift Explorer** view.

2.1.2. Creating a new OpenShift Container Platform project

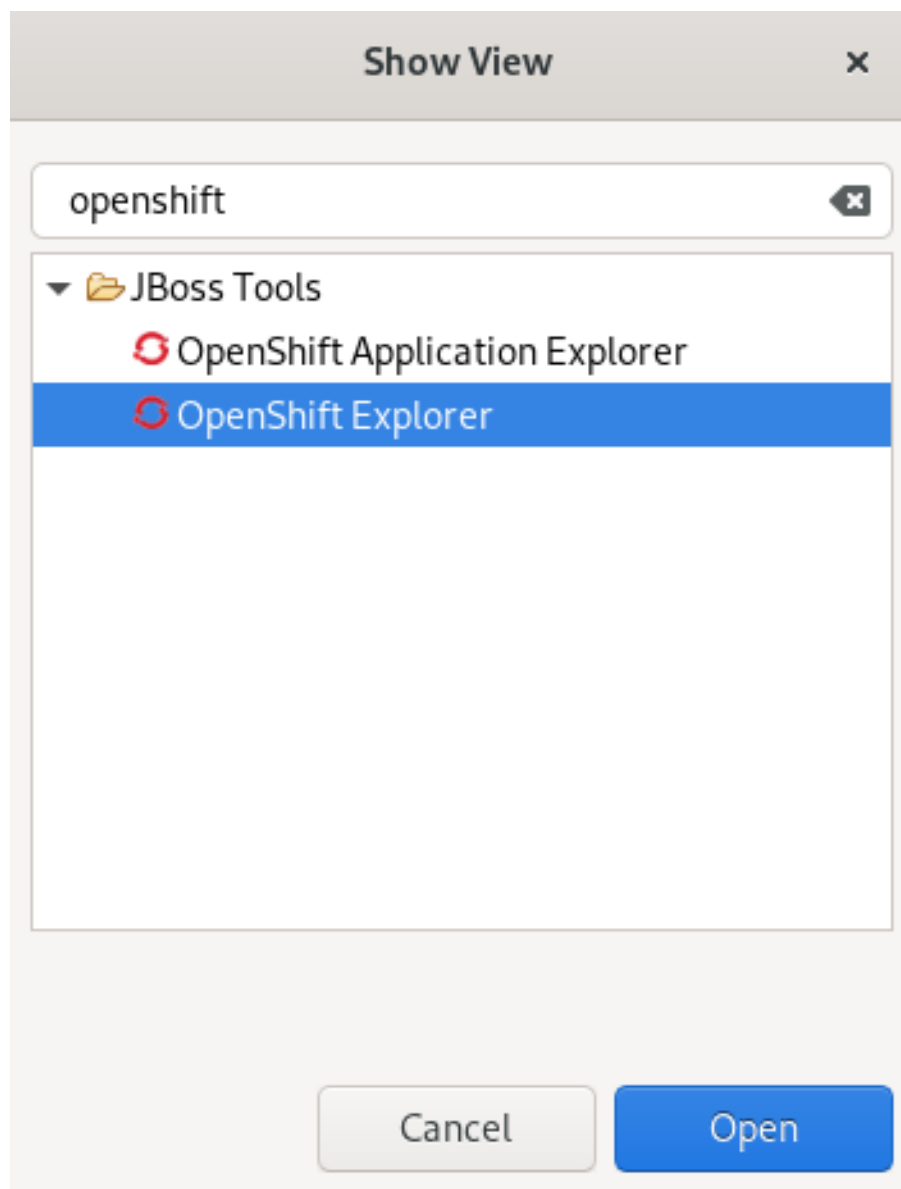
You must create a project, which essentially is a namespace with additional annotations, to centrally manage the access to resources for regular users.

Prerequisites

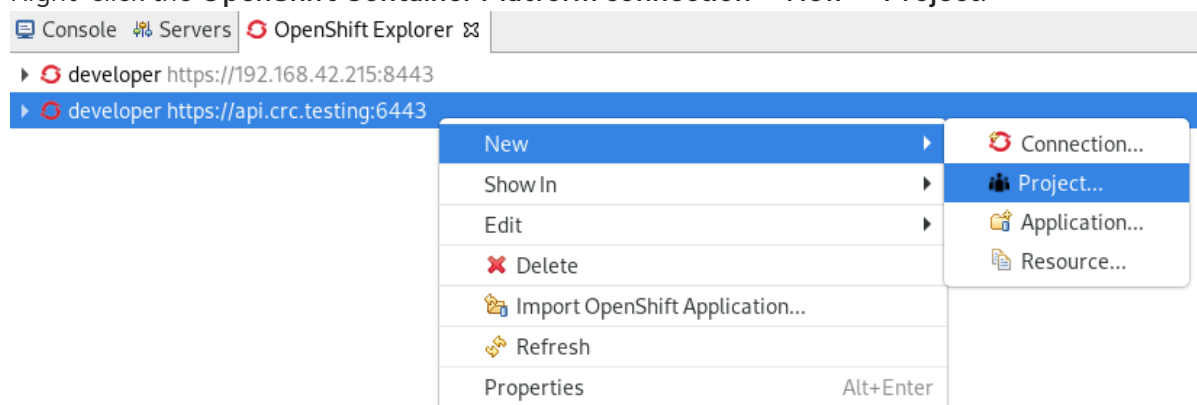
- An OpenShift Container Platform connection.
For more information on how to create a new OpenShift Container Platform connection, see [Section 2.1.1, “Creating a new OpenShift Container Platform connection”](#).

Procedure

1. Start CodeReady Studio.
2. Click **Window** → **Show View** → **Other**.
The **Show View** window appears.



3. Enter **OpenShift** in the search field.
4. Select **OpenShift Explorer**.
5. Click **Open**.
The **OpenShift Explorer** view appears.
6. Right-click the **OpenShift Container Platform** connection → **New** → **Project**.

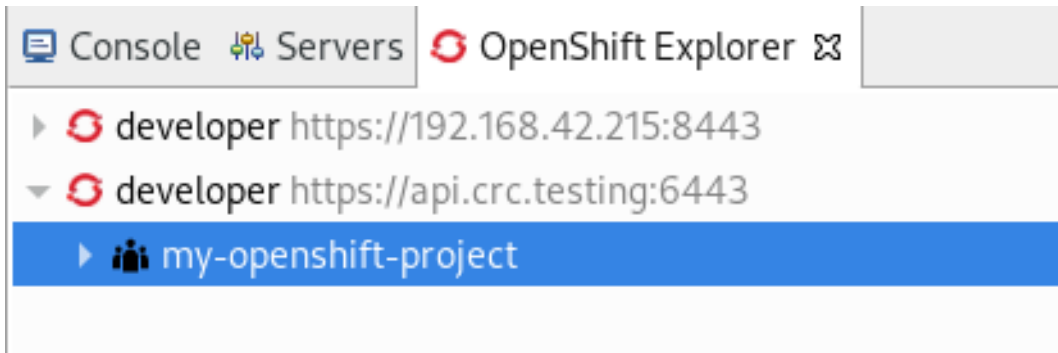


The **New OpenShift Project** window appears.

7. Name your project.

8. Click **Finish**.

Your newly created OpenShift project is now listed in the **OpenShift Explorer** view.



2.1.3. Creating a new OpenShift Container Platform application

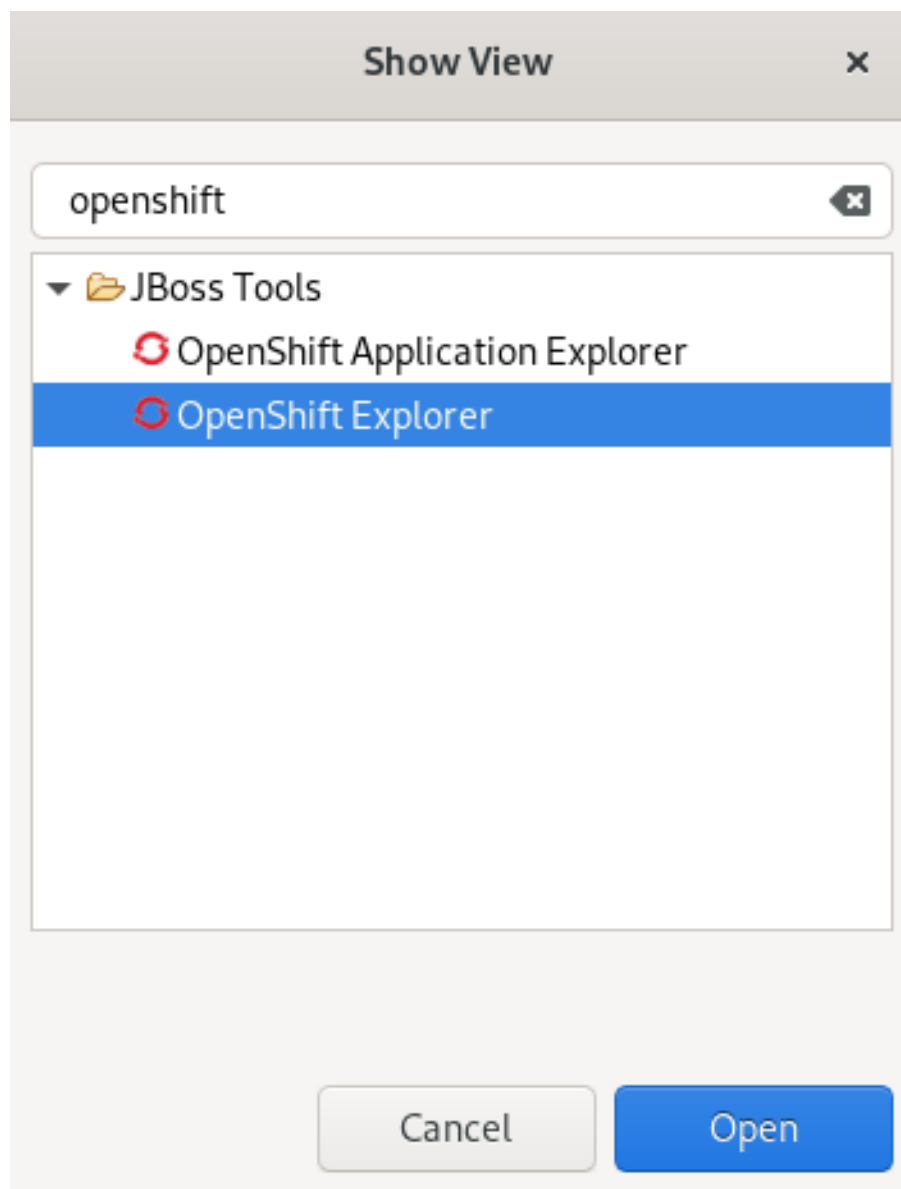
You can use the **OpenShift Application** wizard in the IDE to create OpenShift Container Platform applications from default or custom templates.

Prerequisites

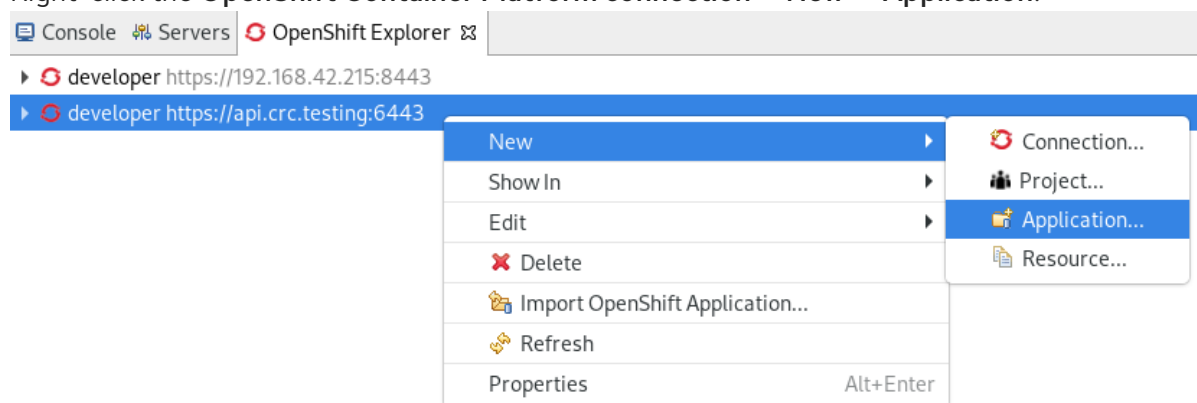
- An OpenShift Container Platform connection.
For more information on how to create a new OpenShift Container Platform connection, see [Section 2.1.1, “Creating a new OpenShift Container Platform connection”](#).
- An OpenShift Container Platform project.
For more information on how to create a new OpenShift Container Platform project, see [Section 2.1.2, “Creating a new OpenShift Container Platform project”](#).

Procedure

1. Start CodeReady Studio.
2. Click **Window** → **Show View** → **Other**.
The **Show View** window appears.



3. Enter **OpenShift** in the search field.
4. Select **OpenShift Explorer**.
5. Click **Open**.
The **OpenShift Explorer** view appears.
6. Right-click the **OpenShift Container Platform** connection → **New** → **Application**.



The **Select template** window appears.

New OpenShift Application

Select template

Server template choices may be filtered by typing the name of a tag in the text field.

OpenShift project: my-openshift-project

New...

Refresh...

Eclipse Project:

Browse...

Server application source

Custom template

dotnet

⚡ dotnet-example (quickstart, dotnet, .net) - openshift

⚡ dotnet-pgsql-persistent (quickstart, dotnet) - openshift

📦 dotnet:2.1 (builder, .net, dotnet, dotnetcore, rh-dotnet21) - openshift

📦 dotnet:3.0 (builder, .net, dotnet, dotnetcore, rh-dotnet30) - openshift

📦 dotnet:3.1 (builder, .net, dotnet, dotnetcore, rh-dotnet31) - openshift

📦 dotnet:latest (builder, .net, dotnet, dotnetcore) - openshift

Details

📦 An example .NET Core application.

Defined Resources...

?

< Back

Next >

Cancel

Finish


7. Select the template.
8. Click **Next**.
The **Template Parameters** window appears.
9. Click **Next**.
The **Resource Labels** window appears.

39

New OpenShift Application

Resource Labels

Add or edit the labels to be added to each resource. Labels are used to organize, group, or select objects and resources, such as pods and


OPENSHIFT


Labels

Key	Value

Add...

Edit...

Remove...



< Back

Next >

Cancel


Finish

10. Click **Add** to add labels.

11. Click **Finish**.

The **Create Application Summary** window appears.

Create Application Summary ✕


OPENSIFT

Results of creating the resources from the dotnet-example template.

New Resources Created:

- ✔ Route - dotnet-example
- ✔ Service - dotnet-example
- ✔ ImageStream - dotnet-example
- ✔ BuildConfig - dotnet-example
- ✔ DeploymentConfig - dotnet-example

▼ Resource Details

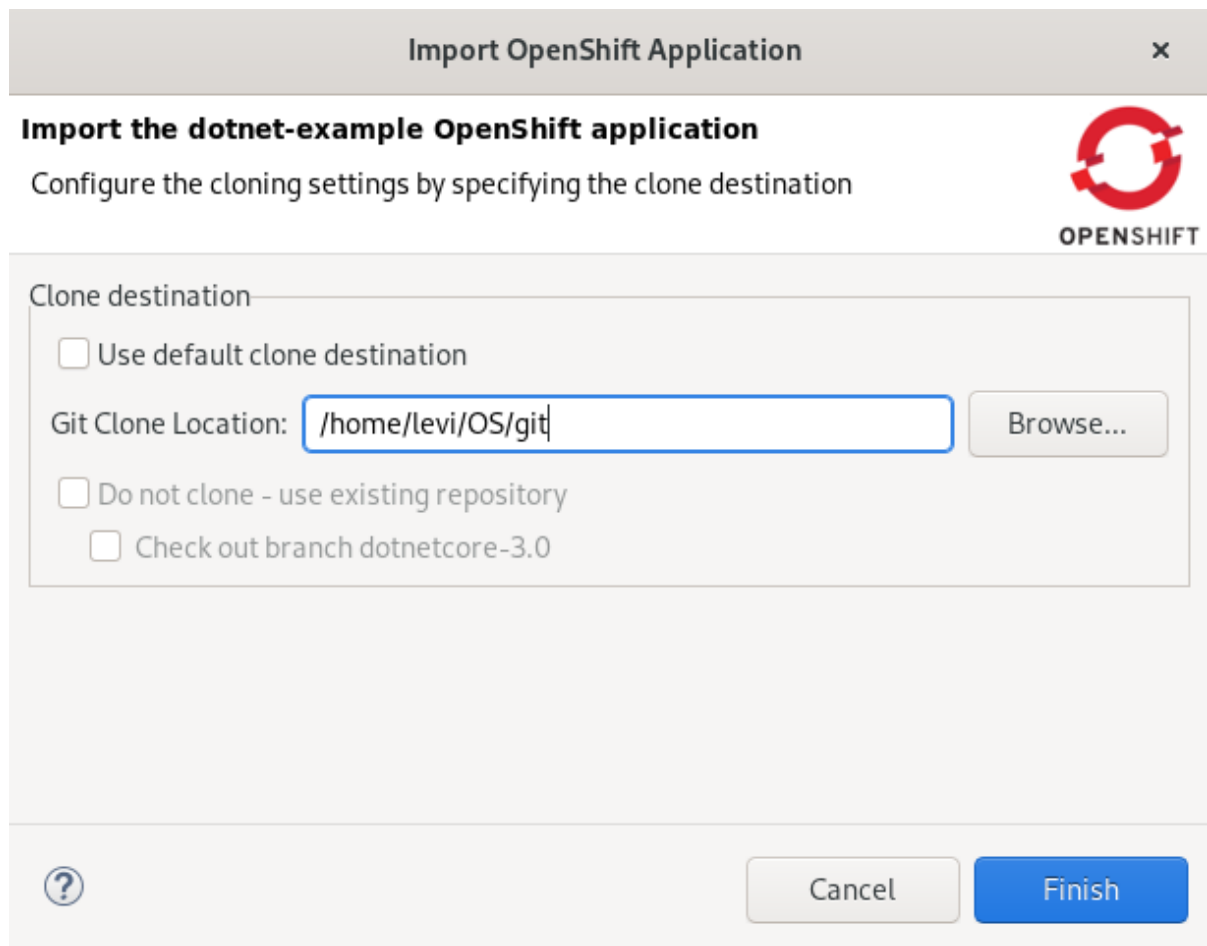
Click [here](#) for webhooks available to automatically trigger builds.

Note the following parameters required to administer your resources:

Name	Value
APPLICATION_DOMAIN	
CONTEXT_DIR	
DOTNET_ASSEMBLY_NAME	
DOTNET_CONFIGURATION	Release
DOTNET_IMAGE_STREAM_TAG	dotnet:3.0
DOTNET_NPM_TOOLS	
DOTNET_PUBLISH_READYTORUN	

OK

12. Click **OK**.
The **Import OpenShift application** window appears.



13. Select the **Git Clone Location**

14. Click **Finish**.

Your newly created OpenShift Container Platform application is now listed in the **OpenShift Explorer** view.

Additional Resources

- For more information about using and creating templates with OpenShift Container Platform, see [Using templates](#).

2.1.4. Importing an existing OpenShift Container Platform application into the IDE

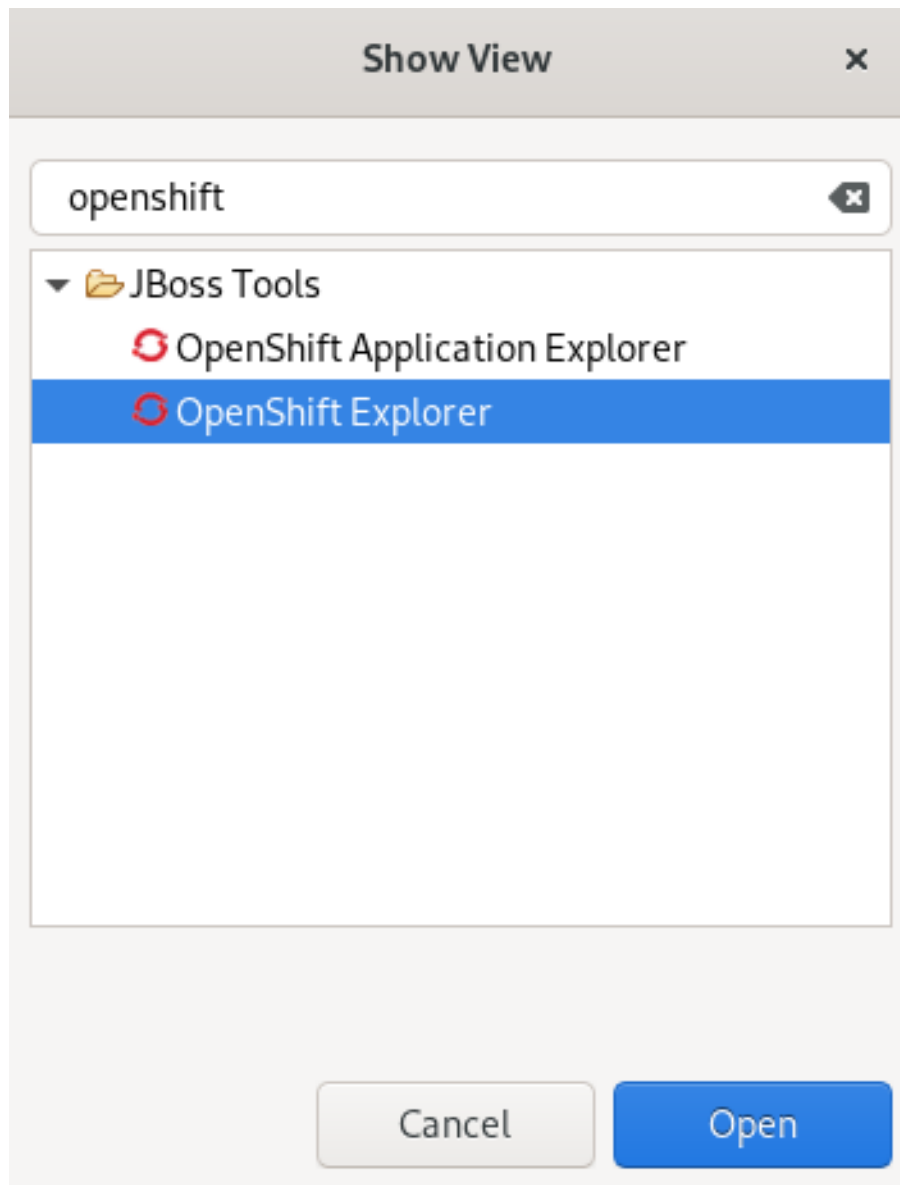
The **OpenShift Explorer** view in the IDE lists applications associated with your OpenShift Container Platform accounts. You can import the source code for these applications individually into the IDE using the **Import OpenShift Application** wizard. After the application is imported, you can easily modify the application source code, build the application, and view it in a web browser.

Prerequisites

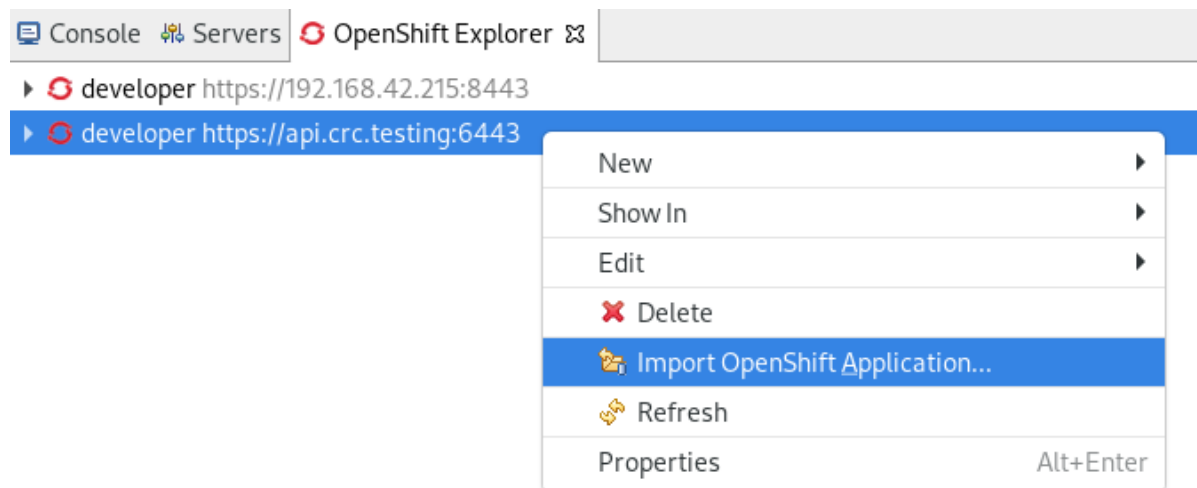
- The application that you are importing to the IDE has its source specified in the **build config** file.
- An OpenShift Container Platform connection.
For more information on how to create a new OpenShift Container Platform connection, see [Section 2.1.1, "Creating a new OpenShift Container Platform connection"](#).

Procedure

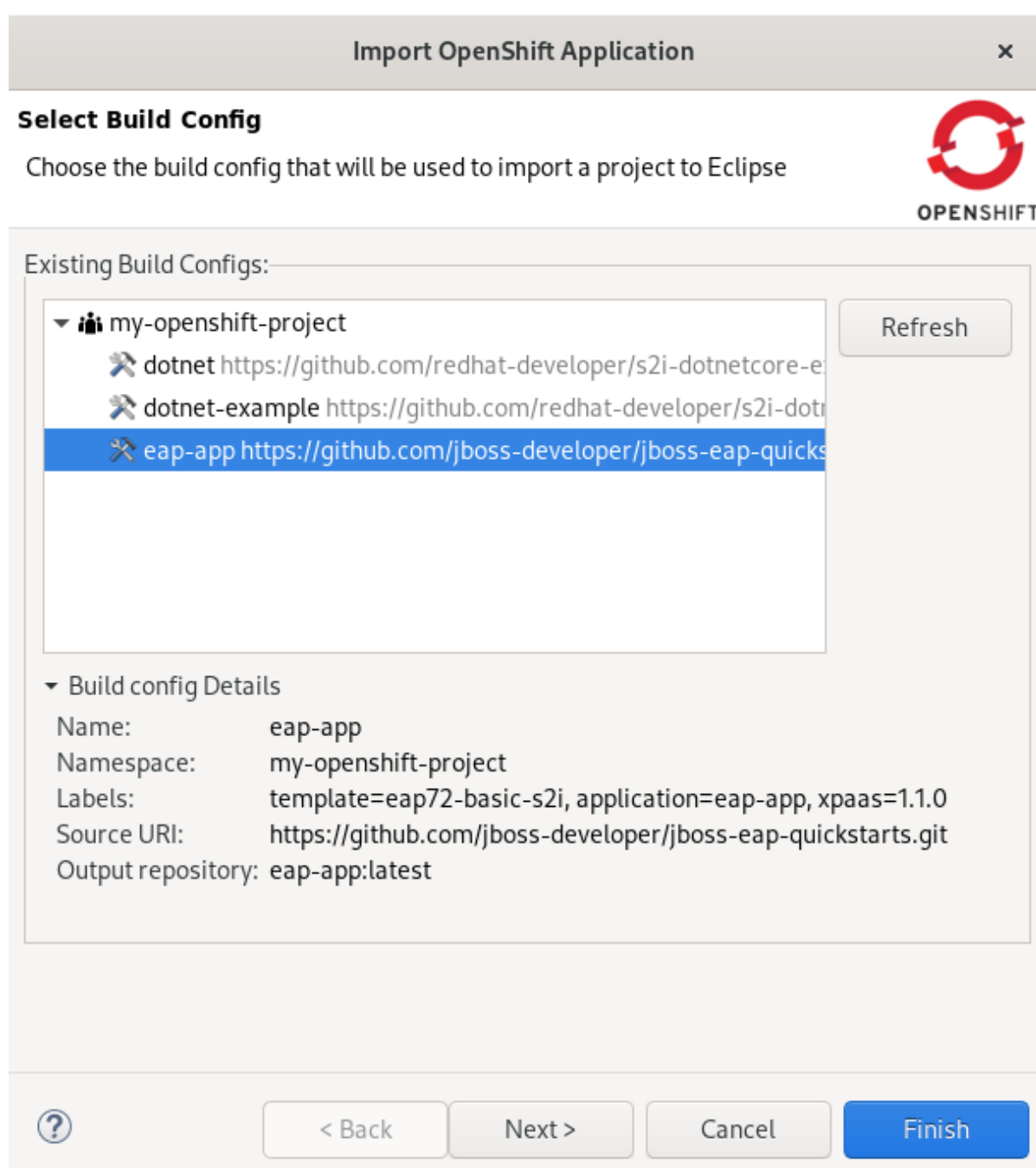
1. Start CodeReady Studio.
2. Click **Window** → **Show View** → **Other**.
The **Show View** window appears.



3. Enter **OpenShift** in the search field.
4. Select **OpenShift Explorer**.
5. Click **Open**.
The **OpenShift Explorer** view appears.
6. Right-click your **OpenShift Container Platform** connection → **Import OpenShift Application**.



The **Select Build Config** window appears.



7. Select the application to import.
8. Click **Next**.
The **Import OpenShift application** window appears.

Import OpenShift Application [X]

Import the dotnet-example OpenShift application

Configure the cloning settings by specifying the clone destination

Clone destination

☐ Use default clone destination

Git Clone Location:

☐ Do not clone - use existing repository

☐ Check out branch dotnetcore-3.0

[?]

9. Select the **Git Clone Location**.
10. Click **Finish**.

Your newly imported OpenShift Container Platform application is now listed in the **OpenShift Explorer** view.

2.1.5. Deploying an application using the server adapter

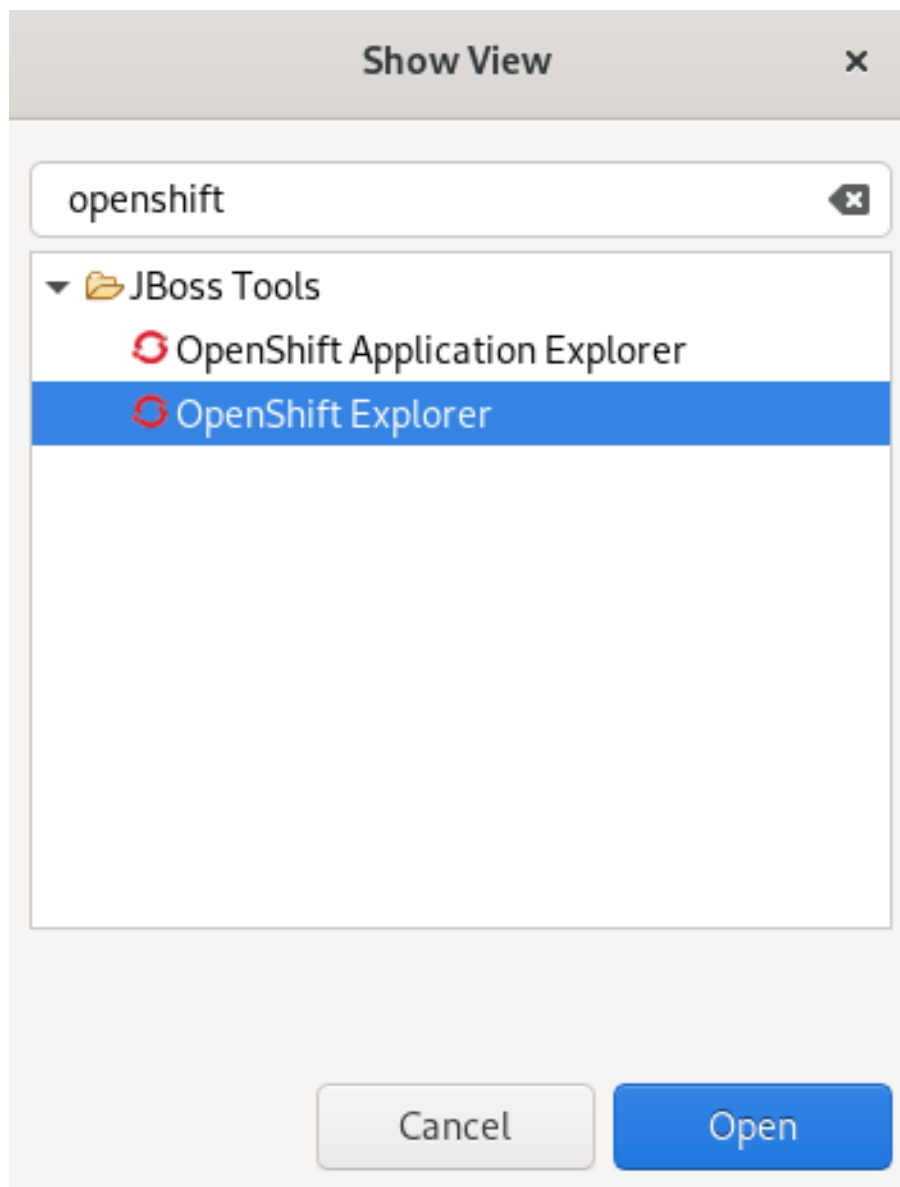
The server adapter enables you to publish the changes that you made in your workspace project to the running OpenShift application on the OpenShift instance. It enables incremental deployment of applications directly into the deployed pods on OpenShift. You can use the server adapter to push changes in your application directly to the running OpenShift application without committing the source code to the Git repository.

Prerequisites

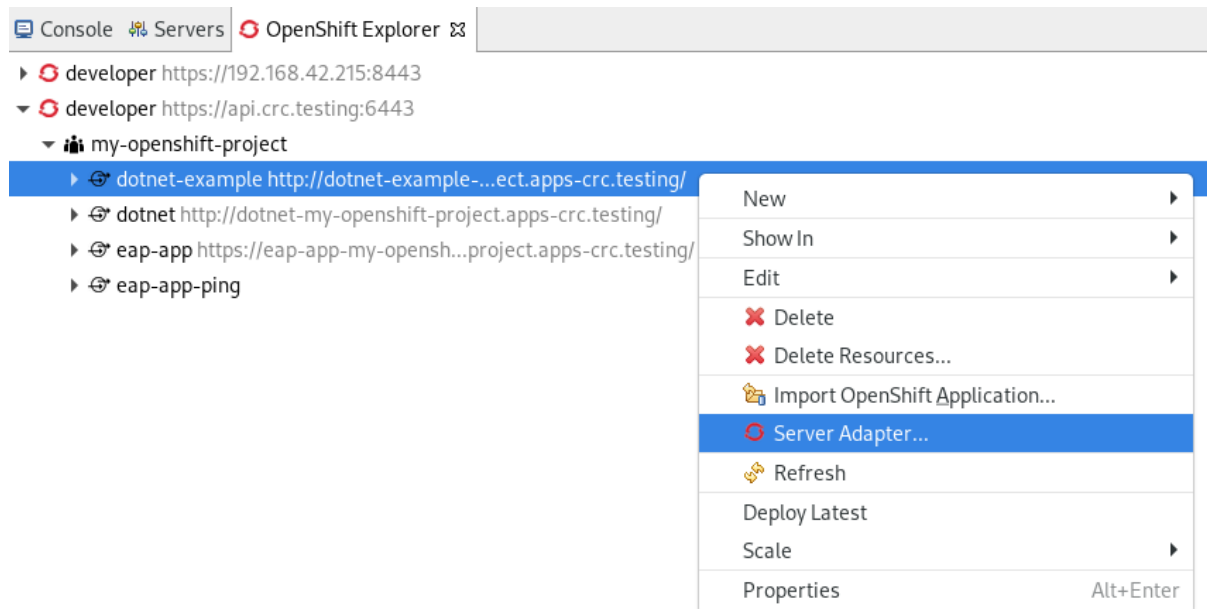
- An OpenShift Container Platform connection.
For more information on how to create a new OpenShift Container Platform connection, see [Section 2.1.1, "Creating a new OpenShift Container Platform connection"](#).

Procedure

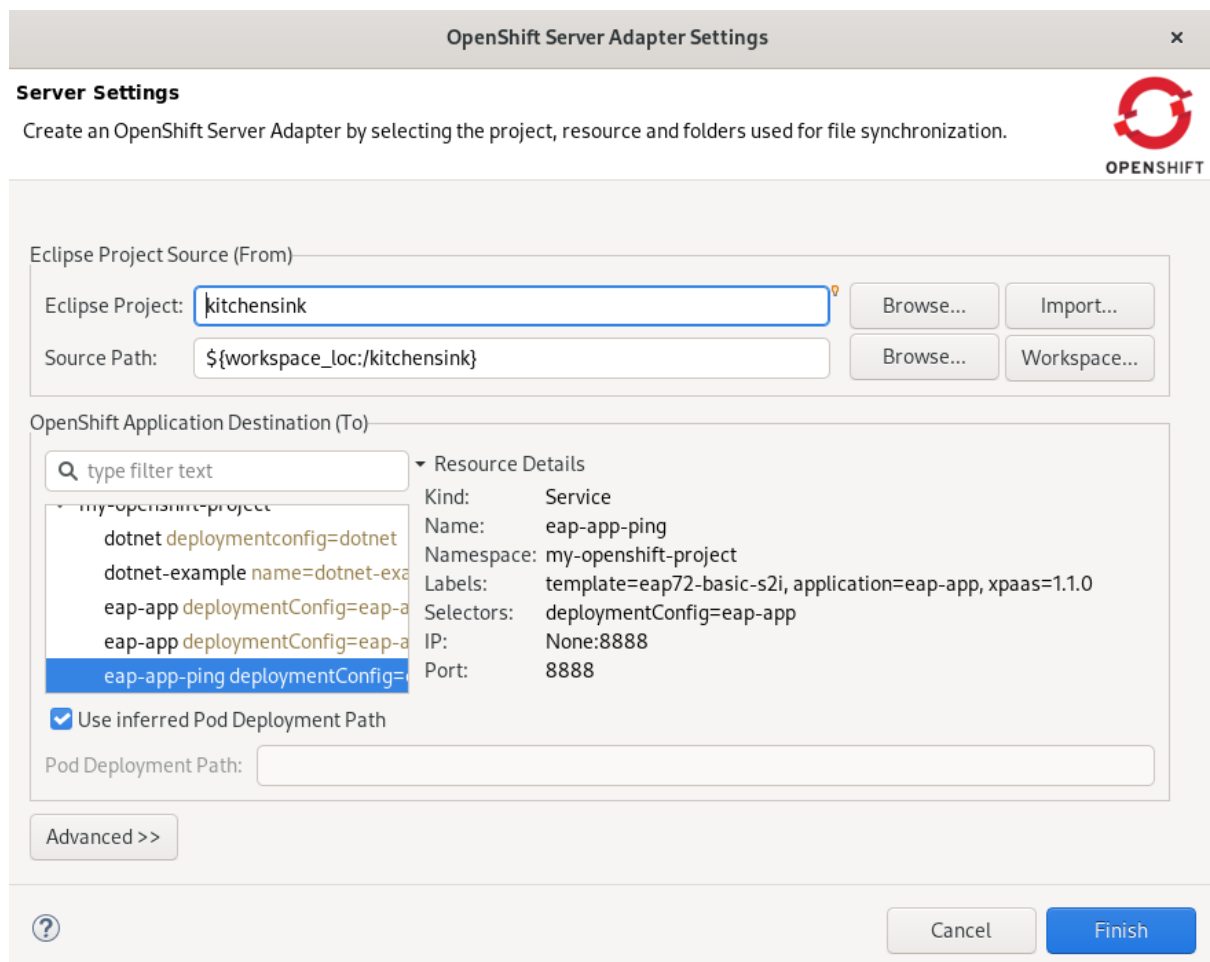
1. Start CodeReady Studio.
2. Click **Window** → **Show View** → **Other**.
The **Show View** window appears.



3. Enter **OpenShift** in the search field.
4. Select **OpenShift Explorer**.
5. Click **Open**.
The **OpenShift Explorer** view appears.
6. Expand the OpenShift Container Platform connection.
7. Right-click your **application** → **Server Adapter**.



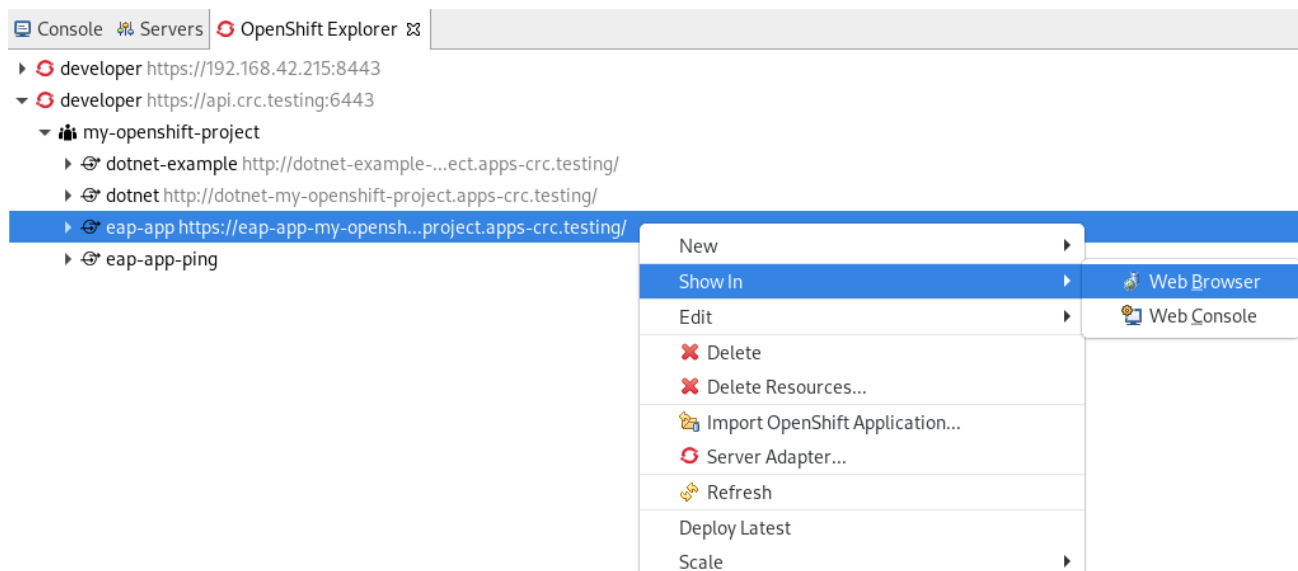
The **Server Settings** window appears.



8. Click **Finish**.

The **Servers** view appears, starting your server adapter.

To open your application in a browser, right-click **application** → **Show In** → **Web Browser**.



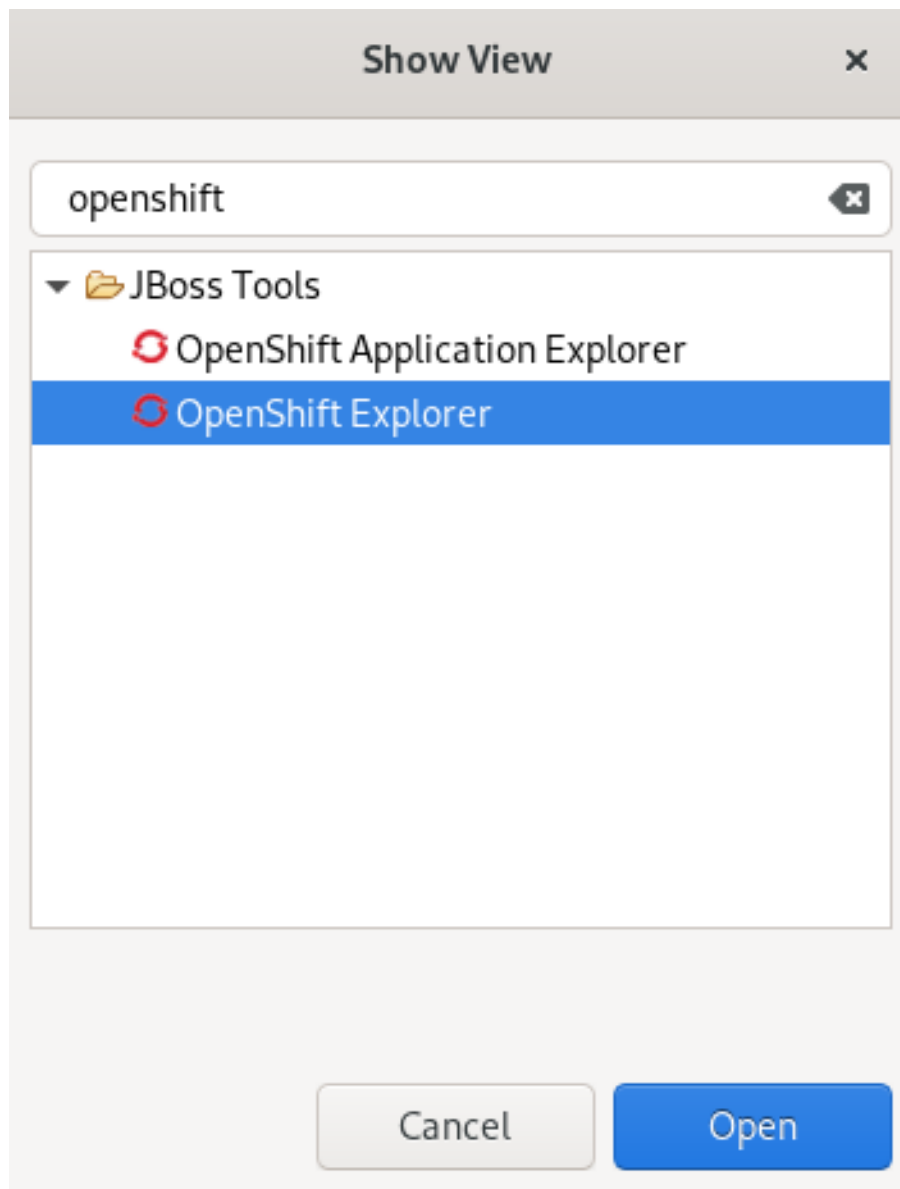
The CodeReady Studio built-in web browser opens, displaying your application.

2.1.6. Deleting an OpenShift Container Platform project

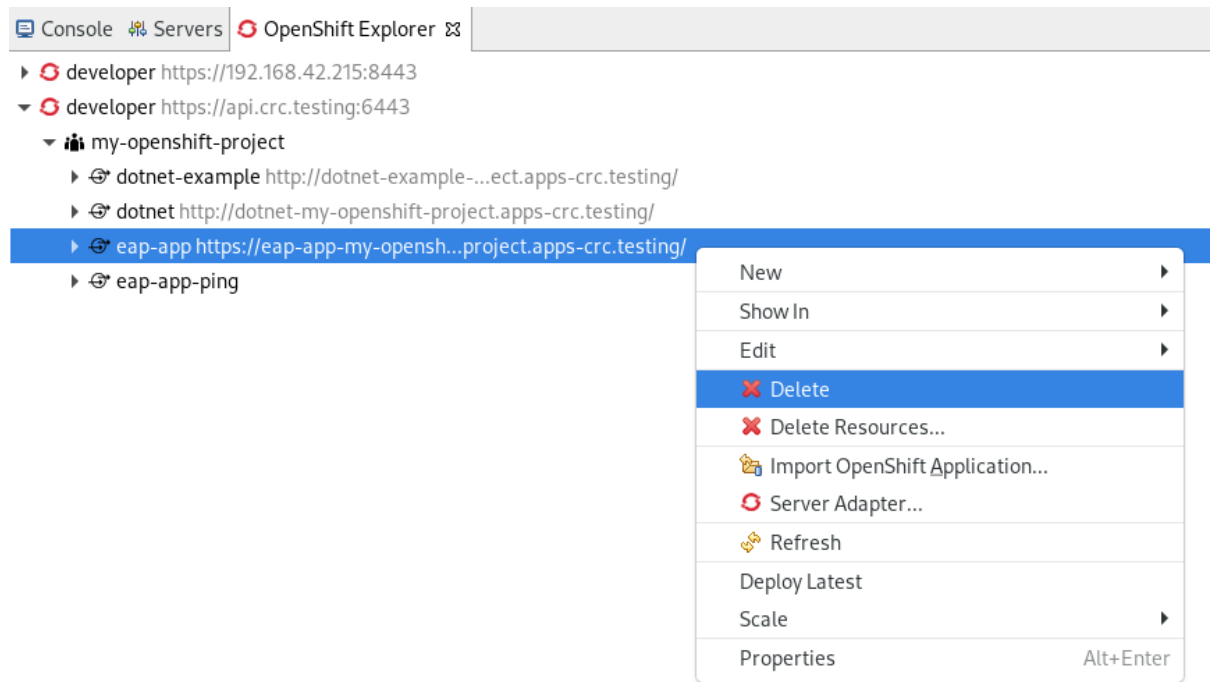
You may choose to delete a project from the workspace for a fresh start in project development or after you have concluded development in a project. When you delete a project, all the resources associated with the project are deleted.

Procedure

1. Start CodeReady Studio.
2. Click **Window** → **Show View** → **Other**.
The **Show View** window appears.



3. Enter **OpenShift** in the search field.
4. Select **OpenShift Explorer**.
5. Click **Open**.
The **OpenShift Explorer** view appears.
6. Expand the OpenShift Container Platform connection.
7. Right-click your **project** → **Delete**.



The **Delete OpenShift Resource** window appears.

8. Click **OK**.

Your project is now deleted.

2.2. SETTING UP AND REMOTELY MONITORING AN OPENSIFT CONTAINER PLATFORM APPLICATION

The IDE allows users to set up a connection to a remote instance of OpenShift Container Platform and then use logs (application logs and build logs) to troubleshoot and monitor running applications.

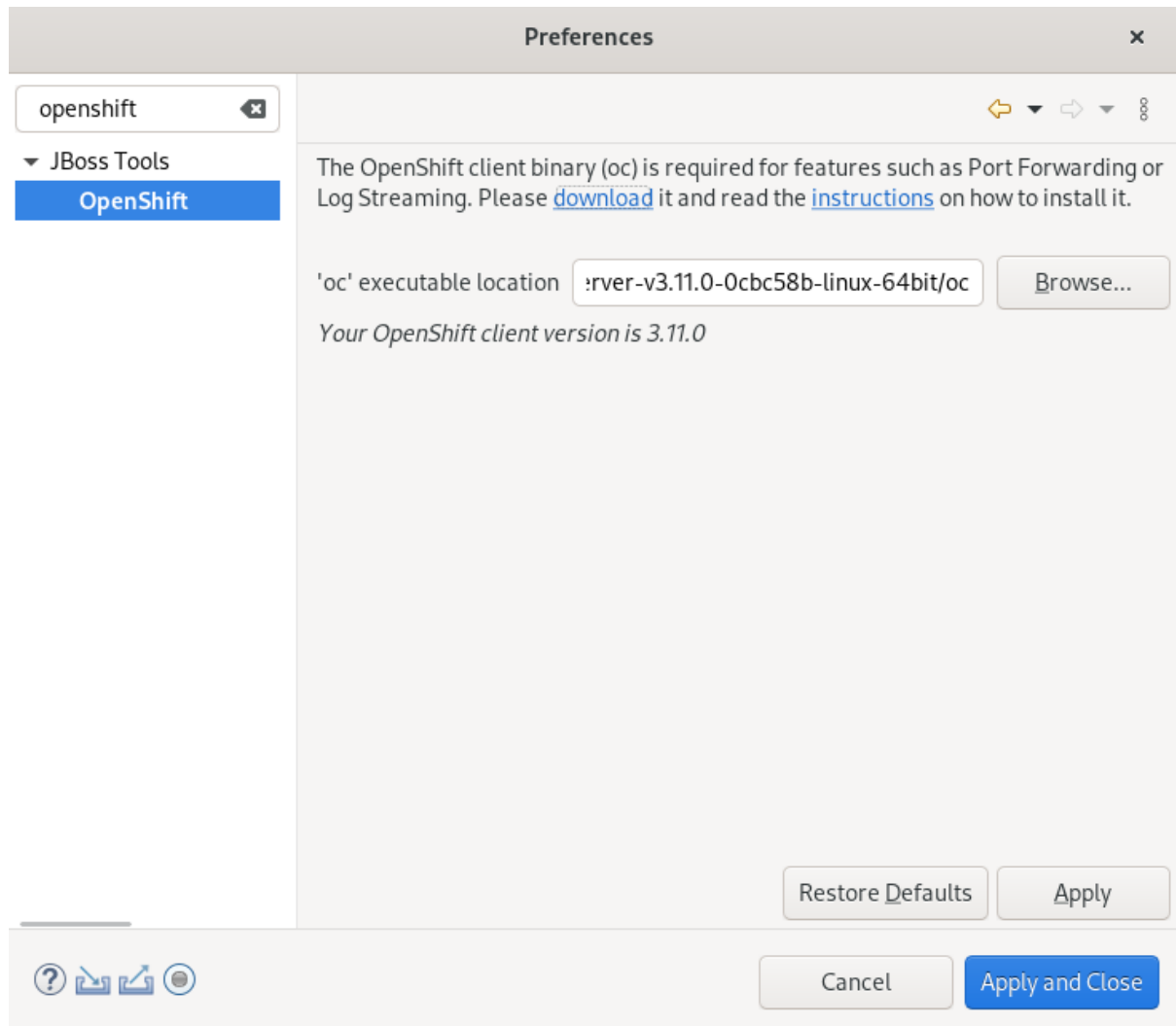
2.2.1. Setting up OpenShift Client Binaries

Prerequisites

Before setting up port forwarding or streaming application and build logs, it is mandatory to set up OpenShift Client Binaries.

Procedure

1. Start CodeReady Studio.
2. Click **Window** → **Preferences**.
The **Preferences** window appears.



3. Enter OpenShift in the search field.
4. Select OpenShift.
5. Click **Browse** to locate the **oc** executable.
6. Click **Apply and Close**.

OpenShift Client Binaries are now set up.

2.2.2. Setting up Port Forwarding

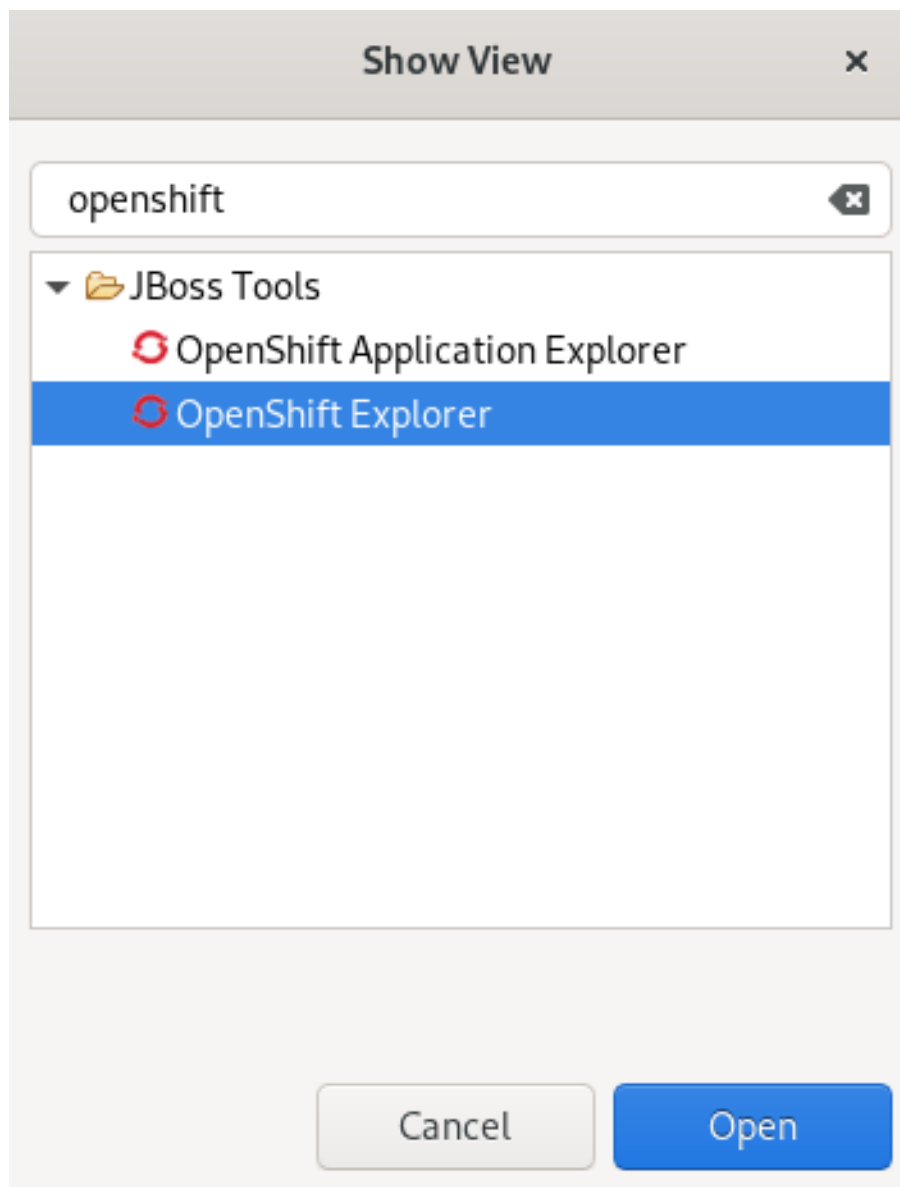
Using the **Application Port Forwarding** window, you can connect the local ports to their remote counterparts to access data or debug the application. Port forwarding automatically stops due to any one of the following reasons:

- The OpenShift Container Platform connection terminates.
- The IDE shuts down.
- The workspace is changed.

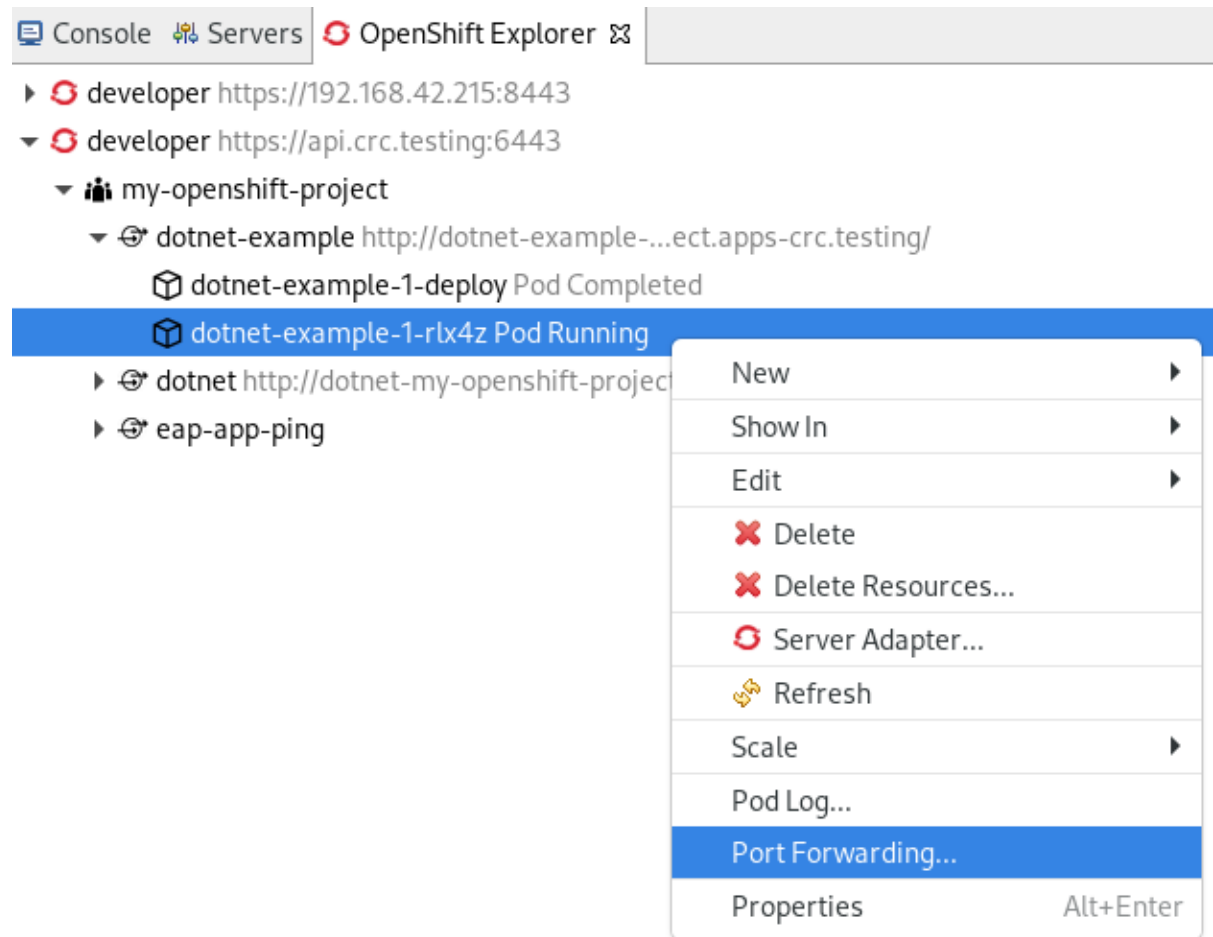
Port forwarding must be enabled each time to connect to OpenShift Container Platform from the IDE.

Procedure

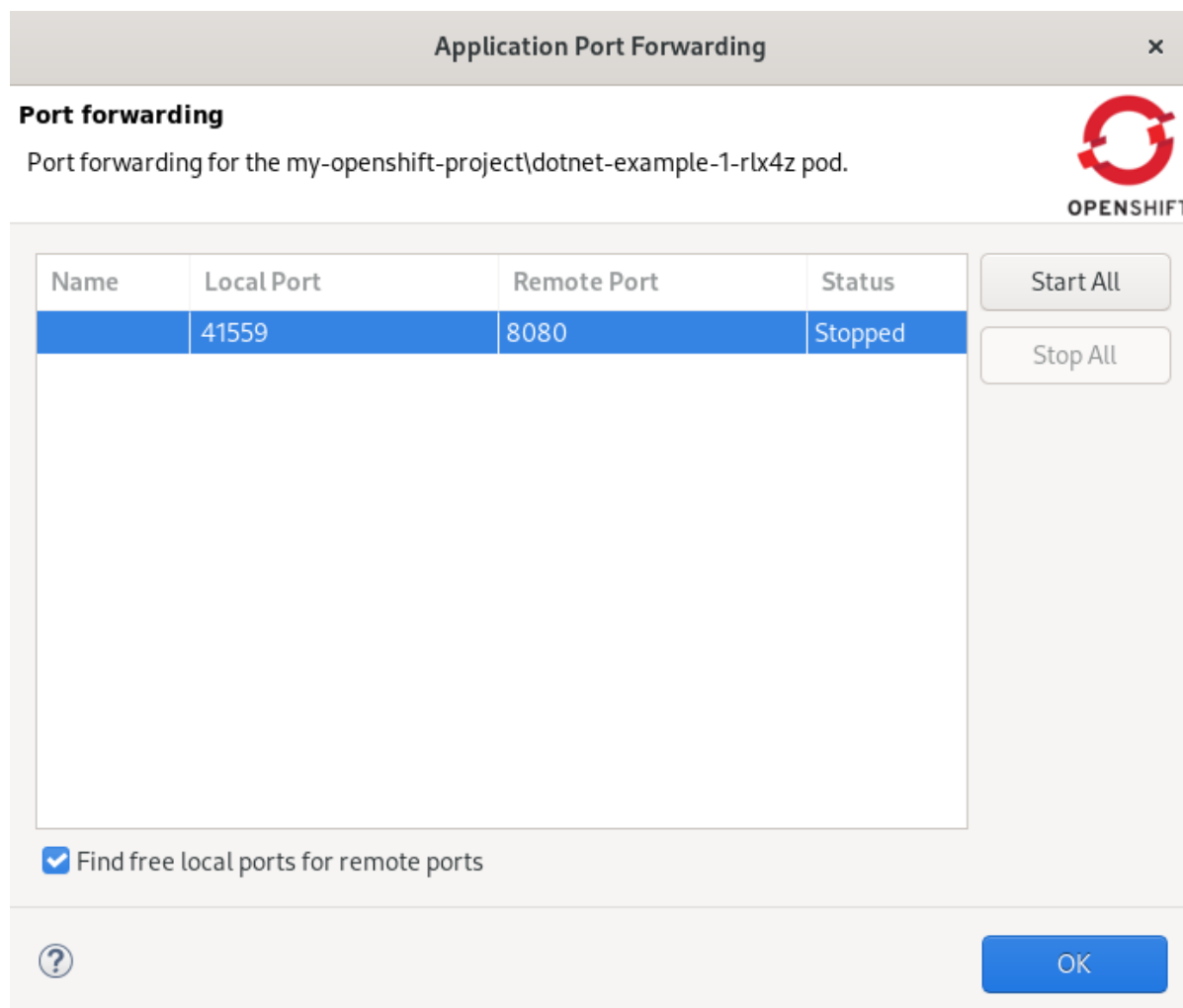
1. Start CodeReady Studio.
2. Click **Window** → **Show View** → **Other**.
The **Show View** window appears.



3. Enter **OpenShift** in the search field.
4. Select **OpenShift Explorer**.
5. Click **Open**.
The **OpenShift Explorer** view appears.
6. Expand the OpenShift Container Platform connection.
7. Right-click the **application** → **Port Forwarding**.



The **Port Forwarding** window appears.



8. Check the **Find free local ports for remote ports** box.
9. Click **Start All**.
10. Click **OK**.

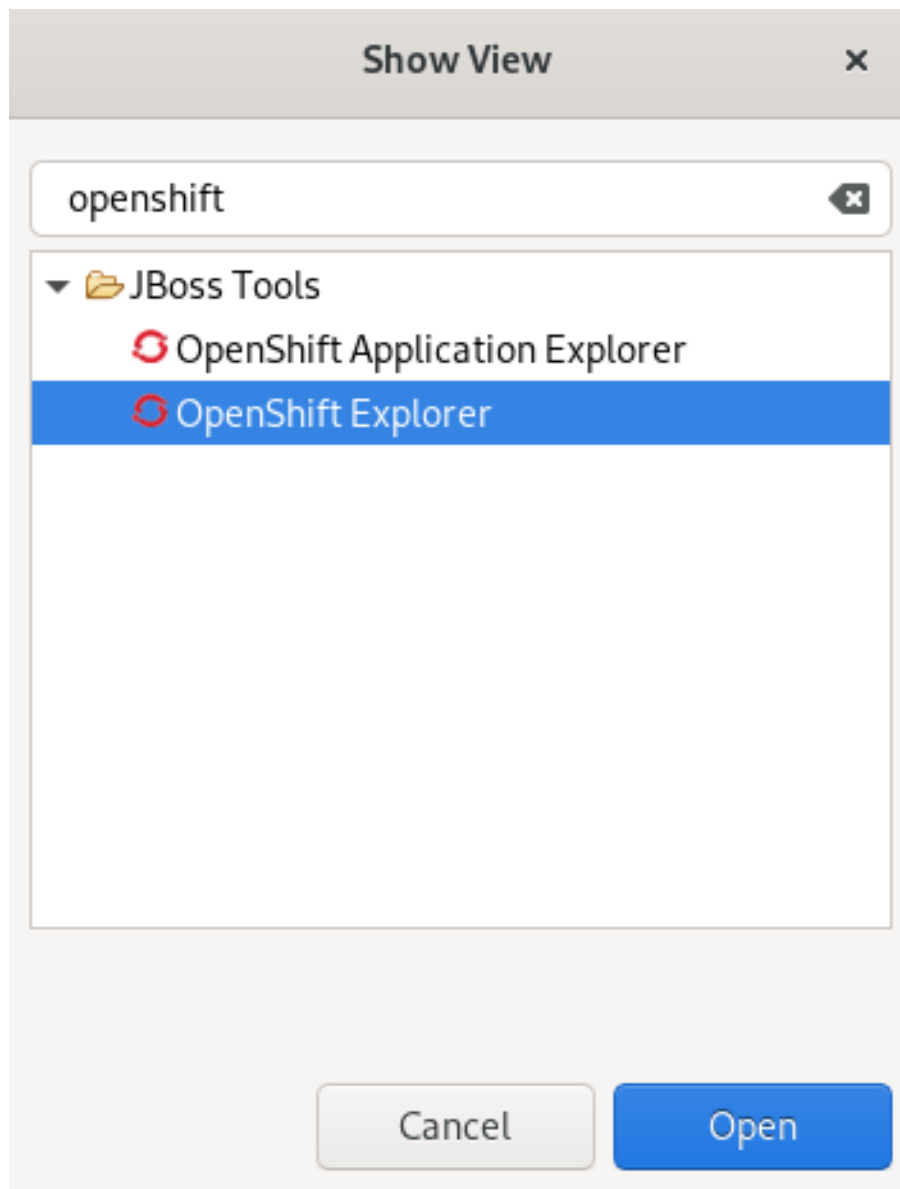
The **Console** view appears showing the port-forwarding starting process.

2.2.3. Streaming Pod Logs

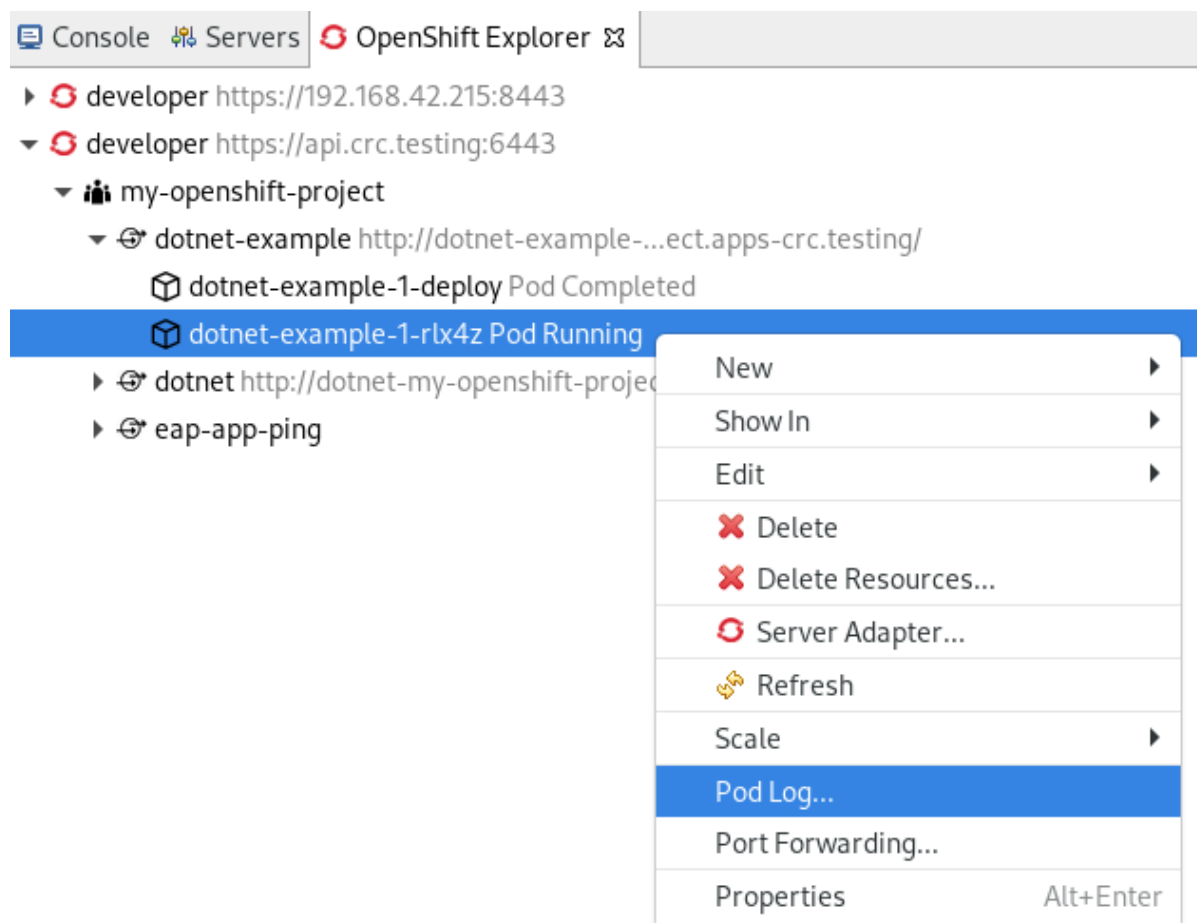
Pod logs are general logs for applications running on a remote OpenShift Container Platform instance. The streaming application logs feature in the IDE is used to monitor applications and use the previous pod log to troubleshoot if the application fails or returns errors.

Procedure

1. Start CodeReady Studio.
2. Click **Window → Show View → Other**.
The **Show View** window appears.



3. Enter **OpenShift** in the search field.
4. Select **OpenShift Explorer**.
5. Click **Open**.
The **OpenShift Explorer** view appears.
6. Expand the OpenShift Container Platform connection.
7. Right-click the **application** → **Port Log**.



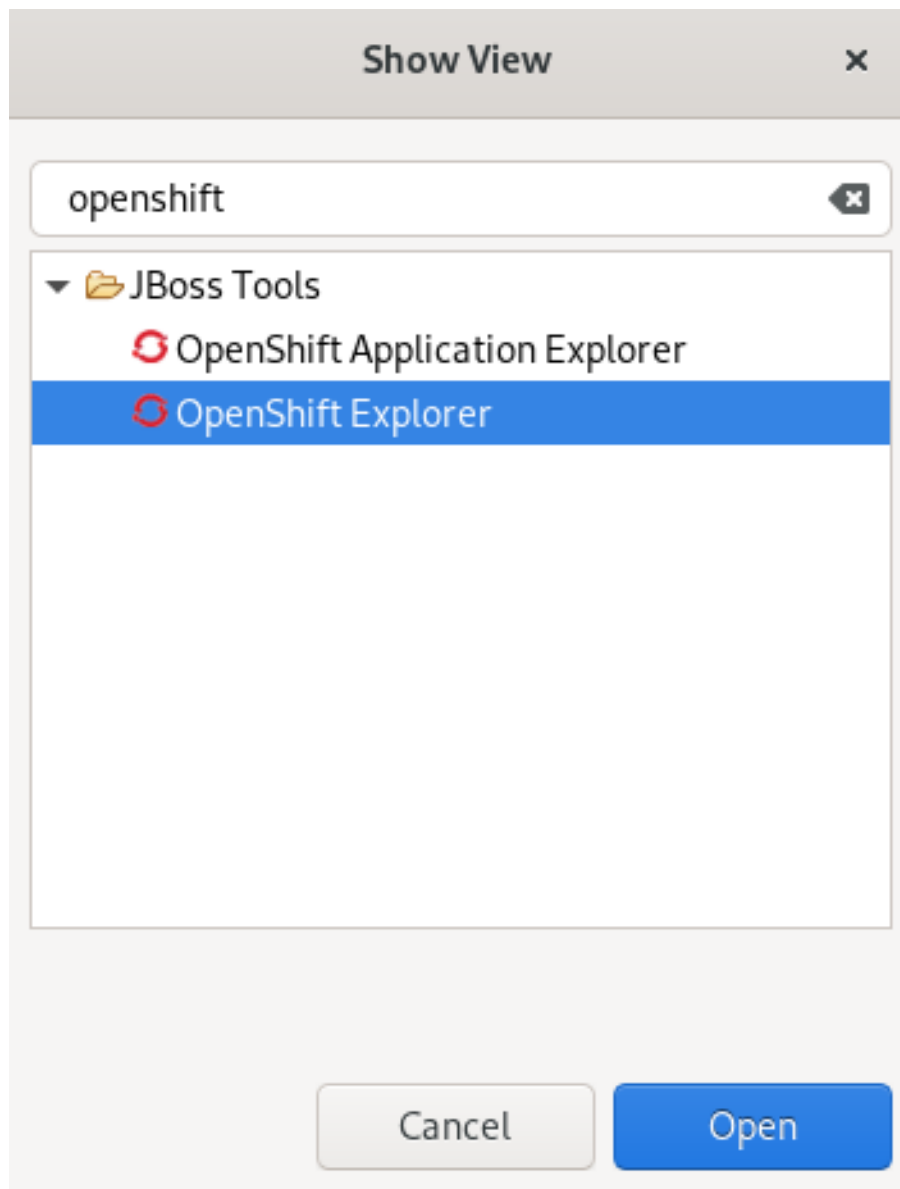
The **Console** view appears displaying the Pod Log.

2.2.4. Streaming Build Logs

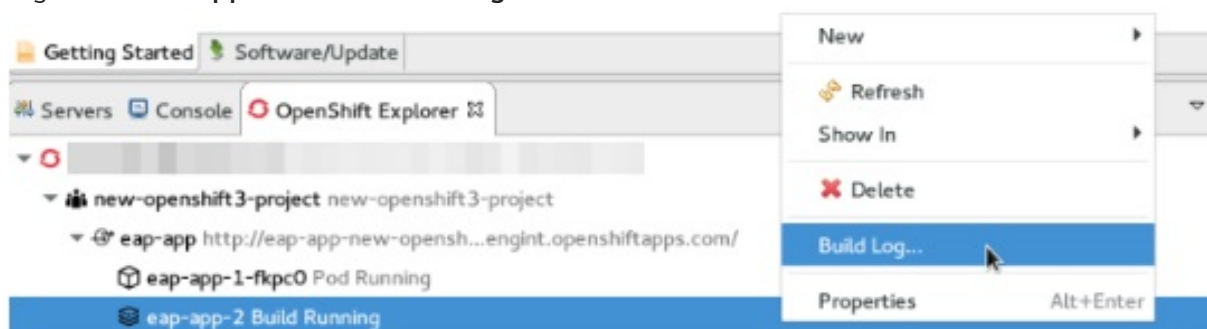
Build logs are logs that document changes to applications running on a remote OpenShift Container Platform instance. The streaming build logs feature in the IDE is used to view the progress of the application build process and to debug the application.

Procedure

1. Start CodeReady Studio.
2. Click **Window** → **Show View** → **Other**.
The **Show View** window appears.



3. Enter **OpenShift** in the search field.
4. Select **OpenShift Explorer**.
5. Click **Open**.
The **OpenShift Explorer** view appears.
6. Expand the OpenShift Container Platform connection.
7. Right-click the **application** → **Build Log**.



The **Console** view appears displaying the Build Log.

2.3. ADDITIONAL RESOURCES

- For more information on OpenShift Application Explorer, see [Getting started with CodeReady Studio Tools](#).

CHAPTER 3. DEVELOPING WITH DOCKER

Prerequisites

- Docker installed on your system.
For more information on how to install Docker, see [Get Docker](#).
- You have obtained the Docker ID.
For more information on how to get a Docker ID, see [Register for a Docker ID](#).

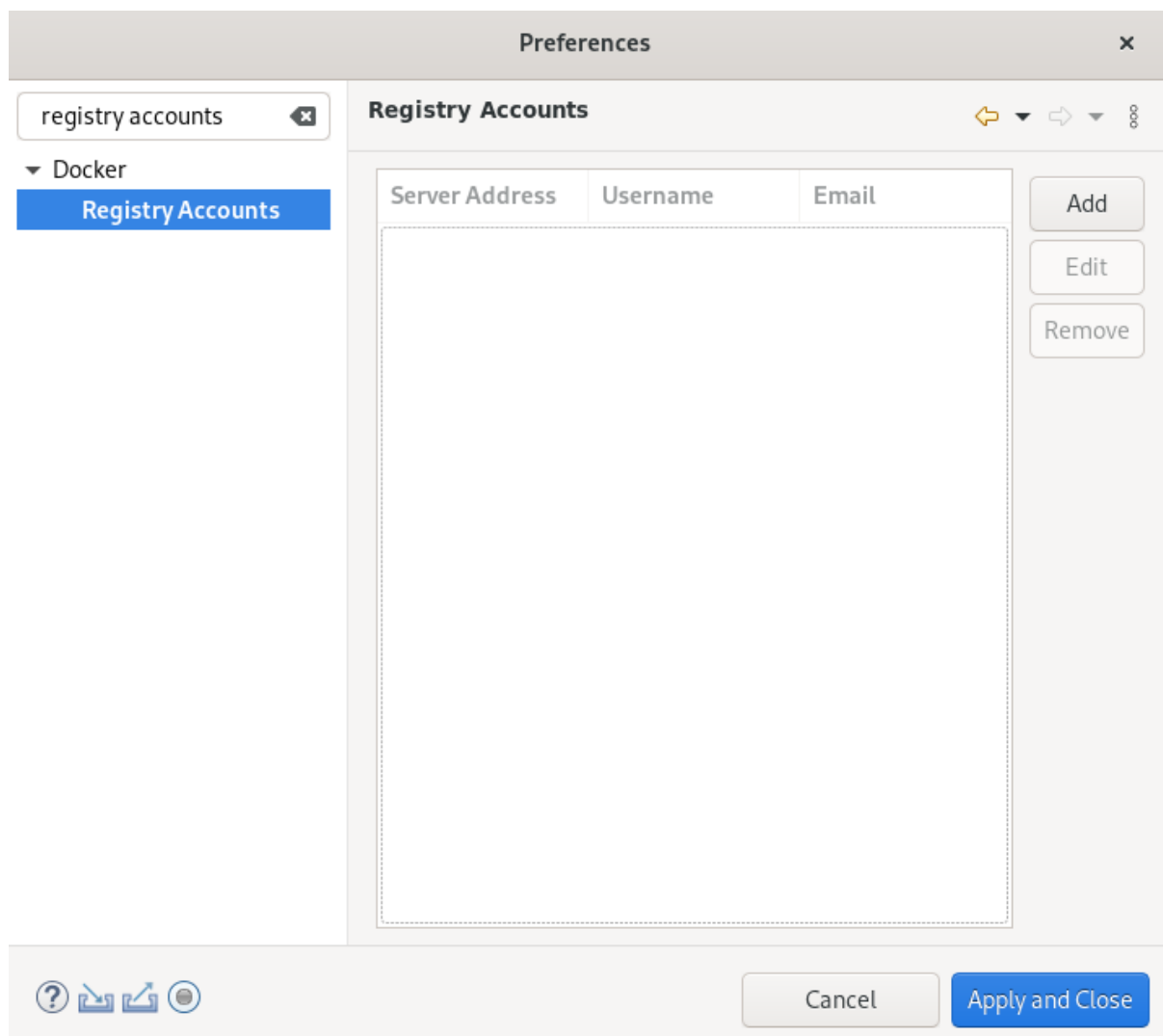
3.1. MANAGING DOCKER CONNECTIONS

3.1.1. Setting up a Docker account in CodeReady Studio

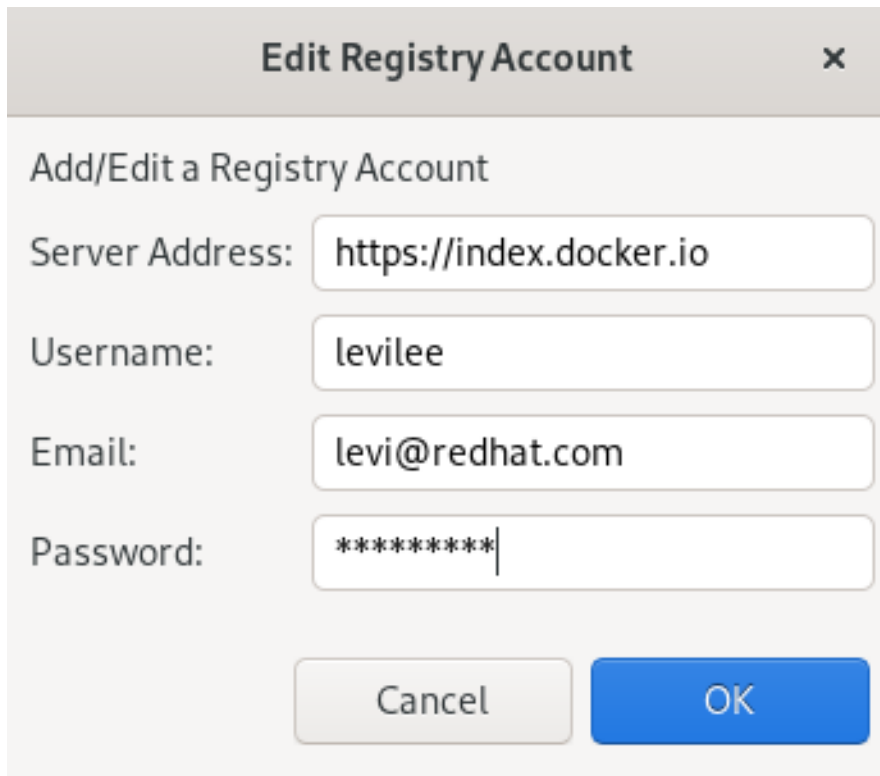
The following section describes how to set up a Docker account in CodeReady Studio. The section assumes you completed the steps listed in the prerequisites section of this chapter.

Procedure

1. Start CodeReady Studio.
2. Click **Window → Preferences**.
The **Preferences** window appears.



3. Enter **Registry Accounts** in the search field.
4. Select **Registry Accounts**.
5. Click **Add**.
The **New Registry Account** window appears.



Edit Registry Account ×

Add/Edit a Registry Account

Server Address:

Username:

Email:

Password:

6. Enter the **Server Address** for the Docker hub.
7. Enter your Docker ID as the **Username**.
8. Enter the email associated with your Docker account.
9. Enter your password.
10. Click **OK**.
11. Click **Apply and Close**.

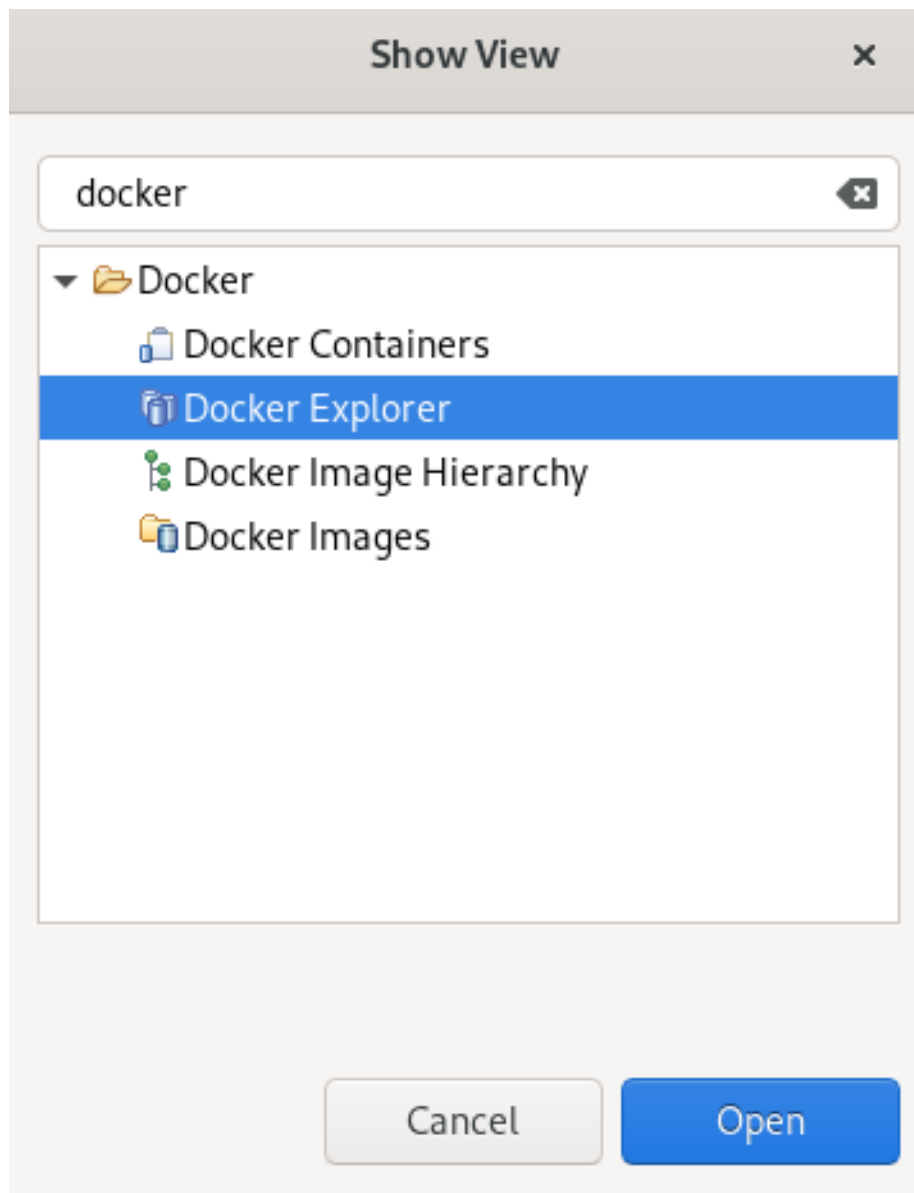
3.1.2. Testing the Docker connection

Prerequisites

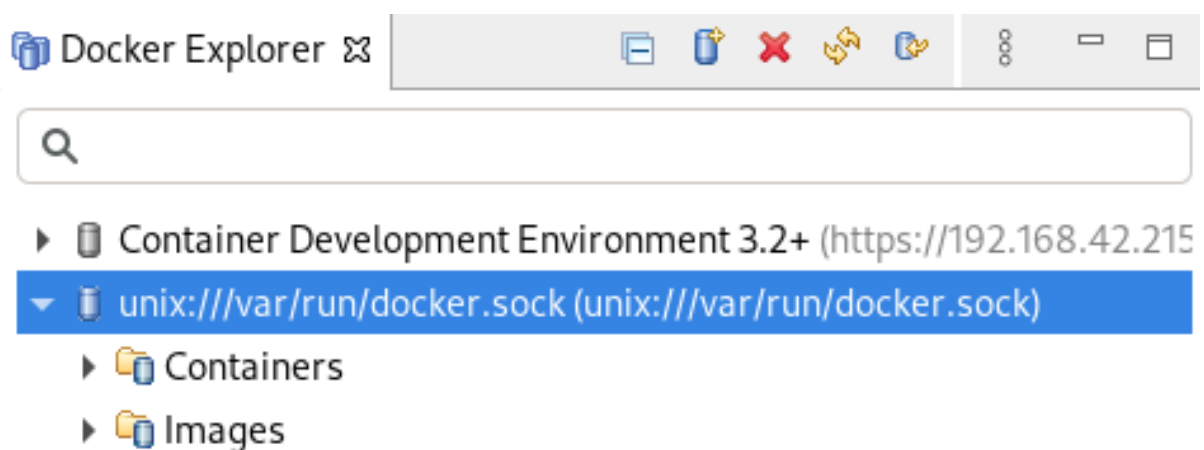
- You have up Docker account set up in CodeReady Studio.
For more information on how to set up a Docker account in CodeReady Studio, see [Section 3.1.1, "Setting up a Docker account in CodeReady Studio"](#)

Procedure

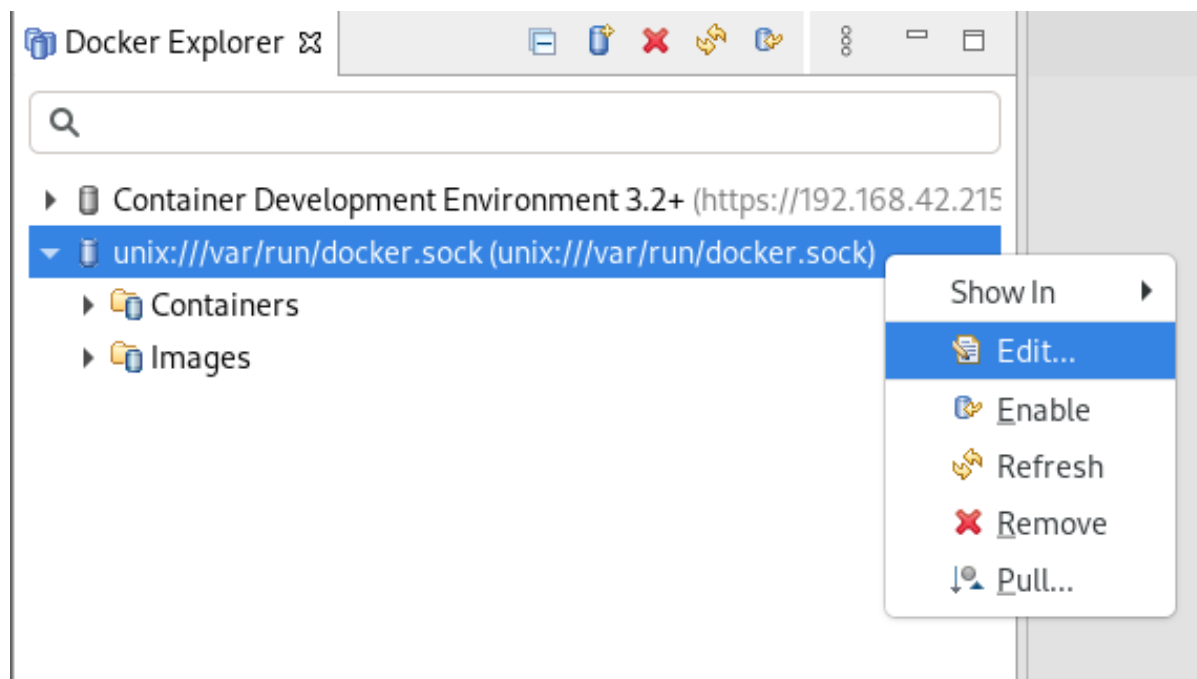
1. Start CodeReady Studio.
2. Click **Window → Show View → Other**.
The **Show View** window appears.



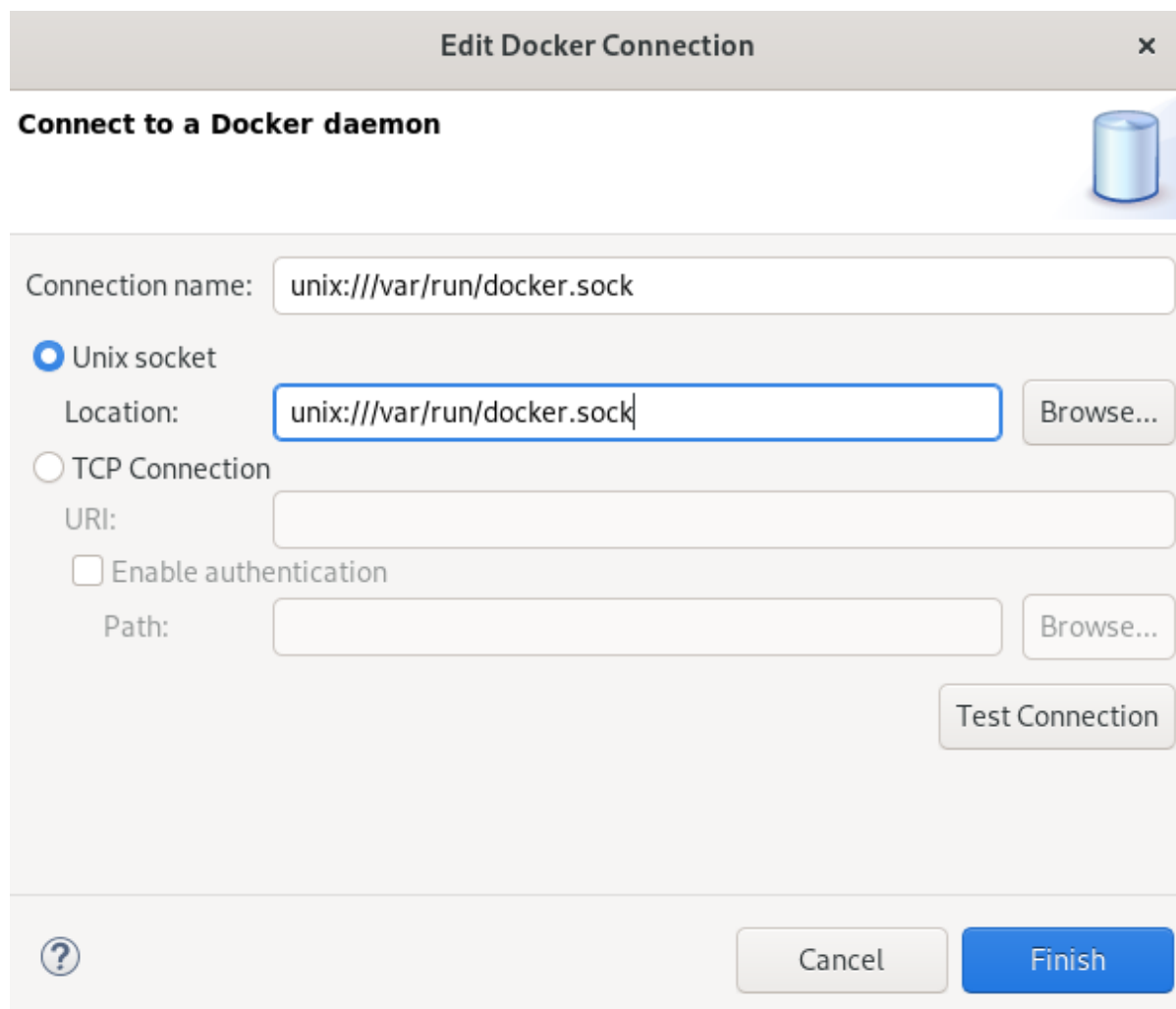
3. Enter **Docker** in the search field.
4. Select **Docker Explorer**.
5. Click **Open**.
The **Docker Explorer** view appears.



6. Right-click **Docker socket** → **Edit**.



The **Edit Docker Connection** window appears.



7. Click **Test Connection**.

If the connection is configured correctly, a window stating that the **Ping succeeded!** appears.

8. Click **OK**.

9. Click **Finish**.

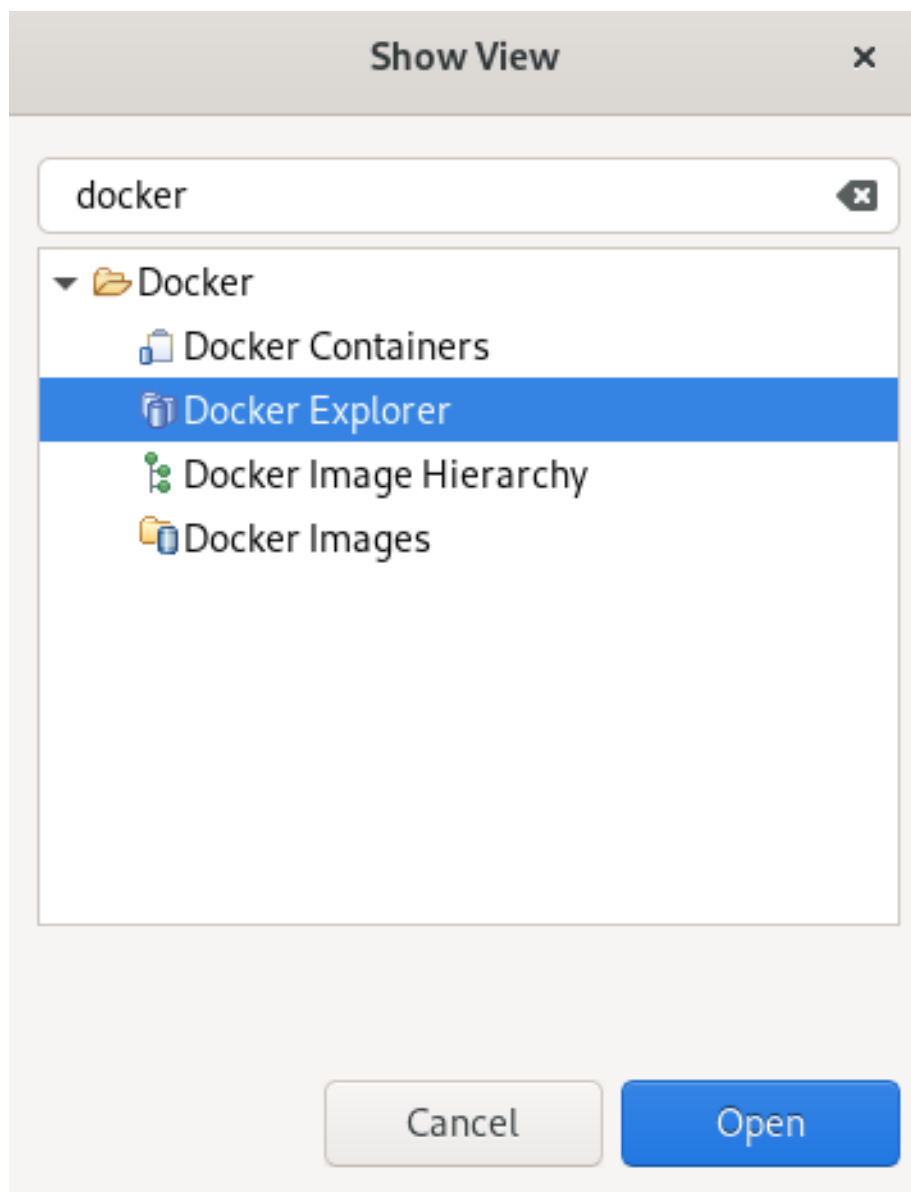
3.1.3. Editing the Docker connection

Prerequisites

- You have a Docker account set up in CodeReady Studio.
For more information on how to set up a Docker account in CodeReady Studio, see [Section 3.1.1, "Setting up a Docker account in CodeReady Studio"](#)

Procedure

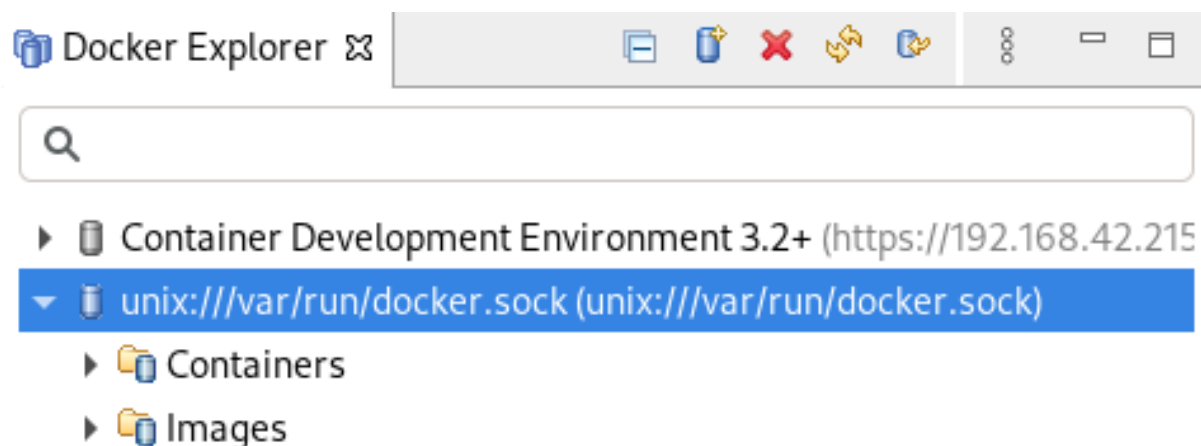
1. Start CodeReady Studio.
2. Click **Window** → **Show View** → **Other**.
The **Show View** window appears.



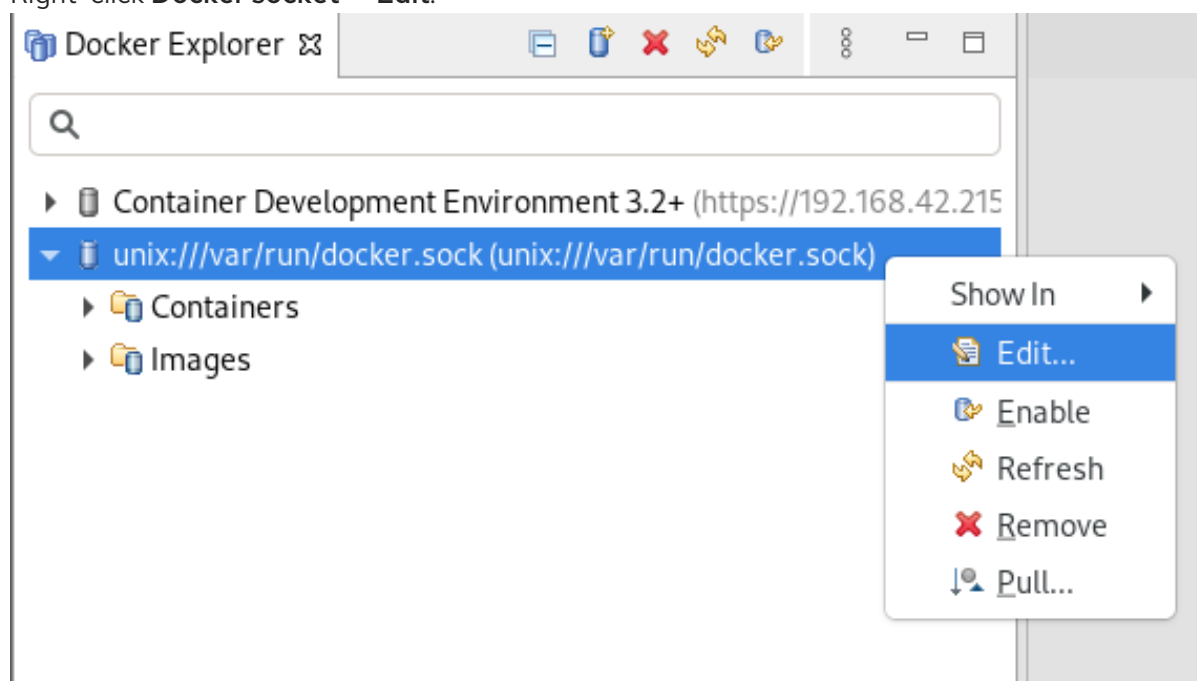
3. Enter **Docker** in the search field.
4. Select **Docker Explorer**.

5. Click **Open**.

The **Docker Explorer** view appears.



6. Right-click **Docker socket** → **Edit**.



The **Edit Docker Connection** window appears.

Edit Docker Connection [X]

Connect to a Docker daemon

Connection name:

☒ **Unix socket**

Location:

☐ **TCP Connection**

URI:

☐ **Enable authentication**

Path:

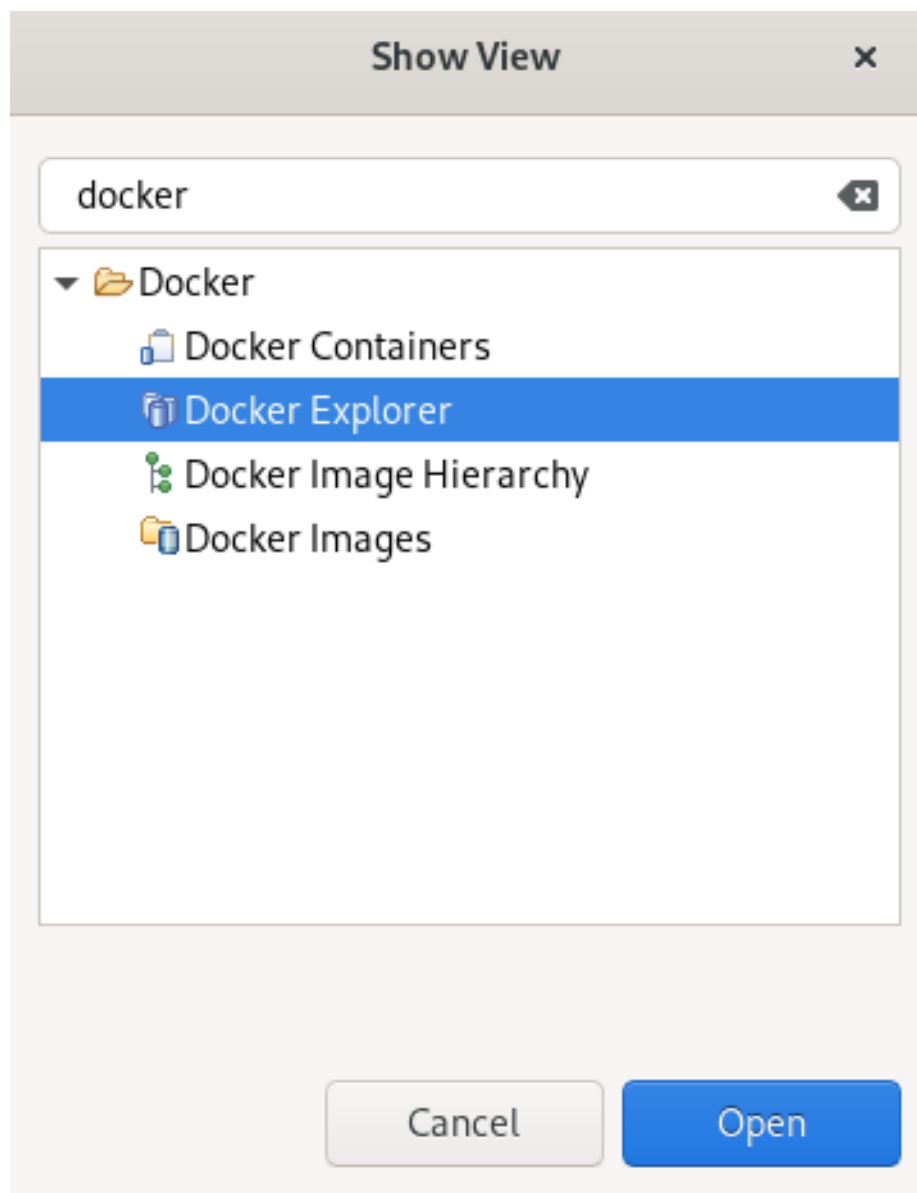
7. Click **Browse** in **Unix socket Location** field to locate a new socket or check the **TCP Connection** option and add the URI.
8. Click **Finish**.

3.2. MANAGING DOCKER IMAGES

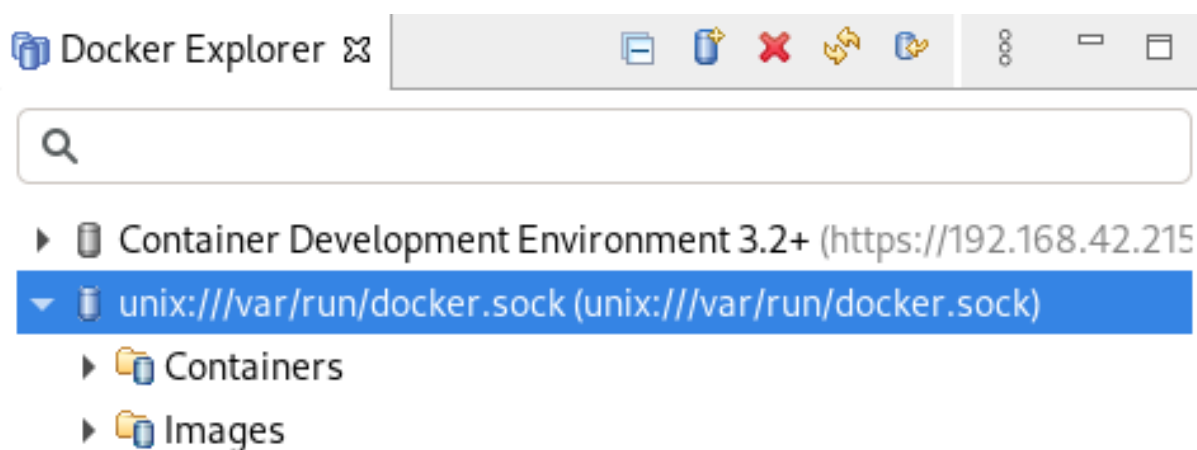
3.2.1. Pulling Docker images

Procedure

1. Start CodeReady Studio.
2. Click **Window** → **Show View** → **Other**.
The **Show View** window appears.

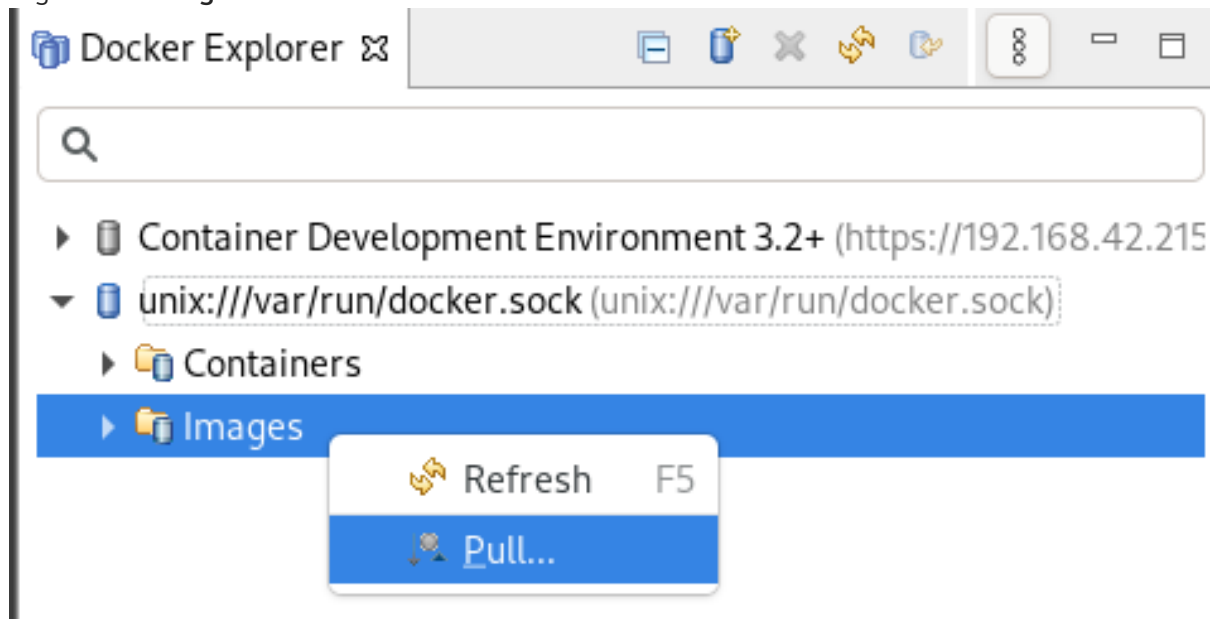


3. Enter **Docker** in the search field.
4. Select **Docker Explorer**.
5. Click **Open**.
The **Docker Explorer** view appears.

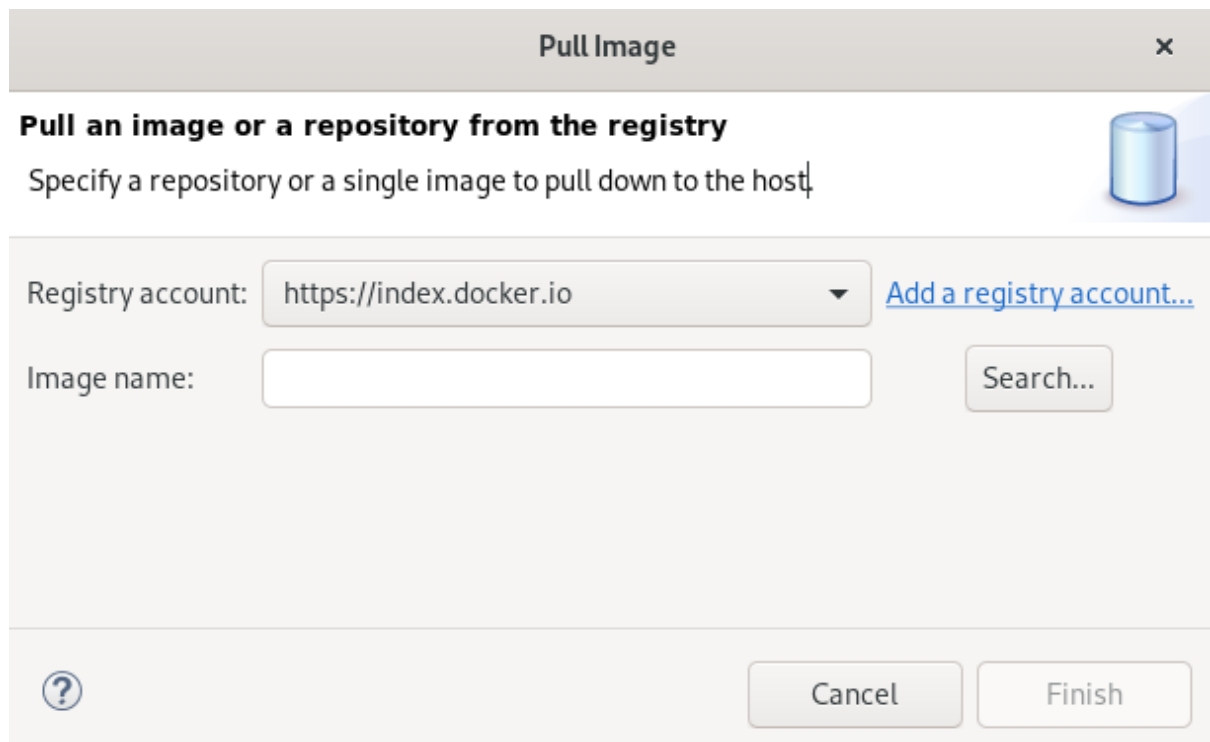


6. Expand **Docker socket** folder.

7. Right-click **Images** → **Pull**.



The **Pull Image** window appears.



8. Click **Search**.

The **Search the Docker Registry for images** window appears.

Search and pull a Docker image

Search the Docker Registry for images






Image: 


Search

Matching images

Name	Stars	Official	Automated
jboss/wildfly	534		
openshift/wildfly-101-centos7	8		
openshift/wildfly-81-centos7	1		

Description

WildFly application server image



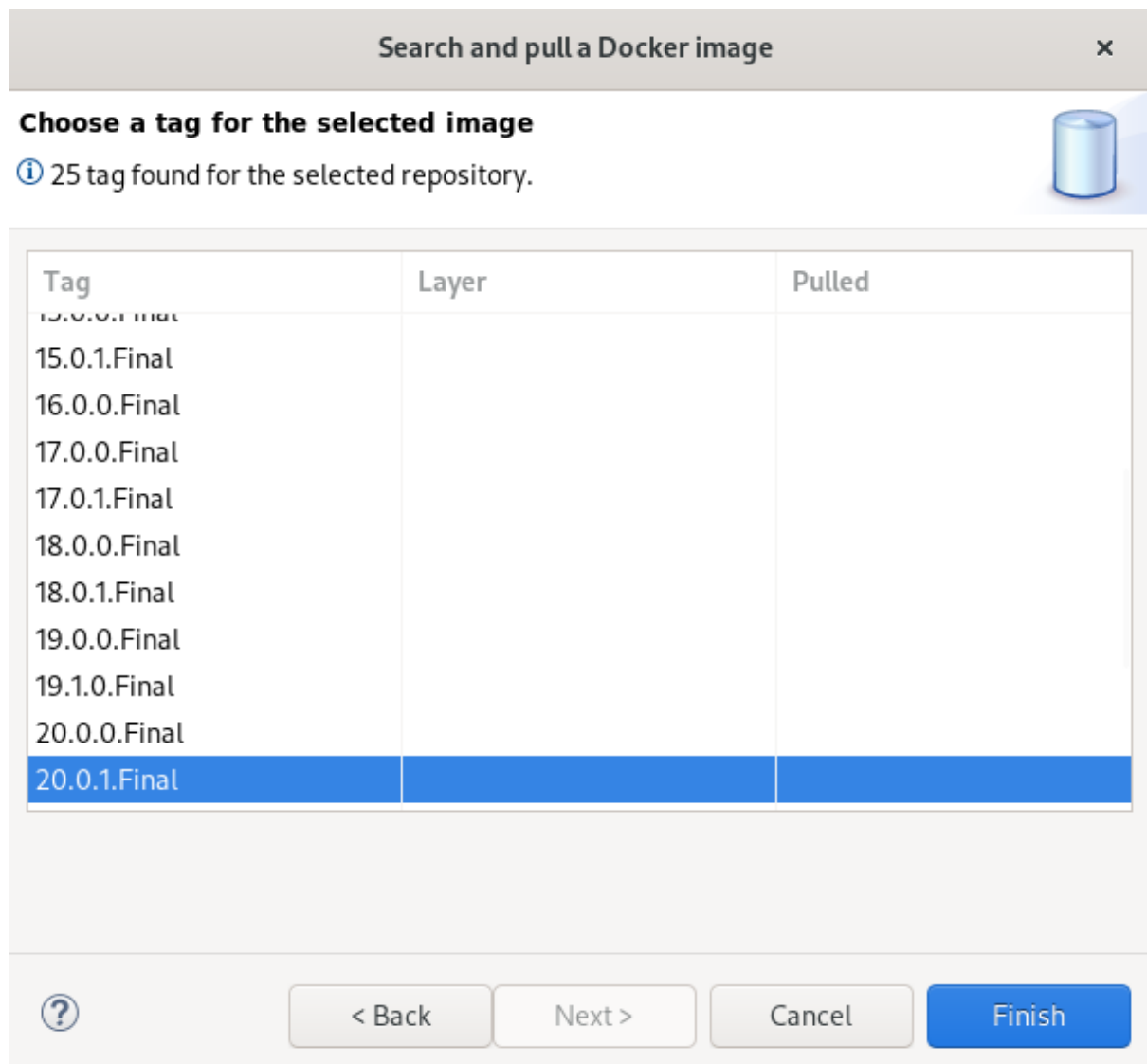
< Back

Next >

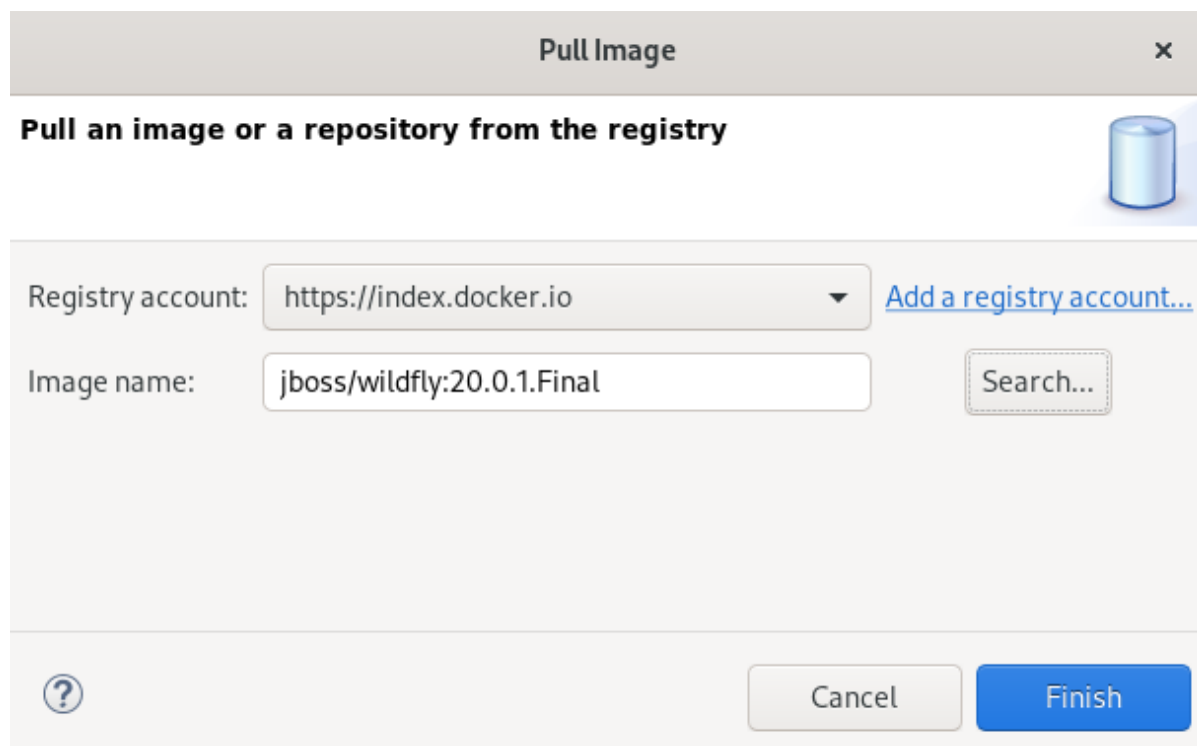
Cancel

Finish

9. Enter the image name in the search field.
10. Click **Next**.
The **Choose a tag for the selected image** window appears.

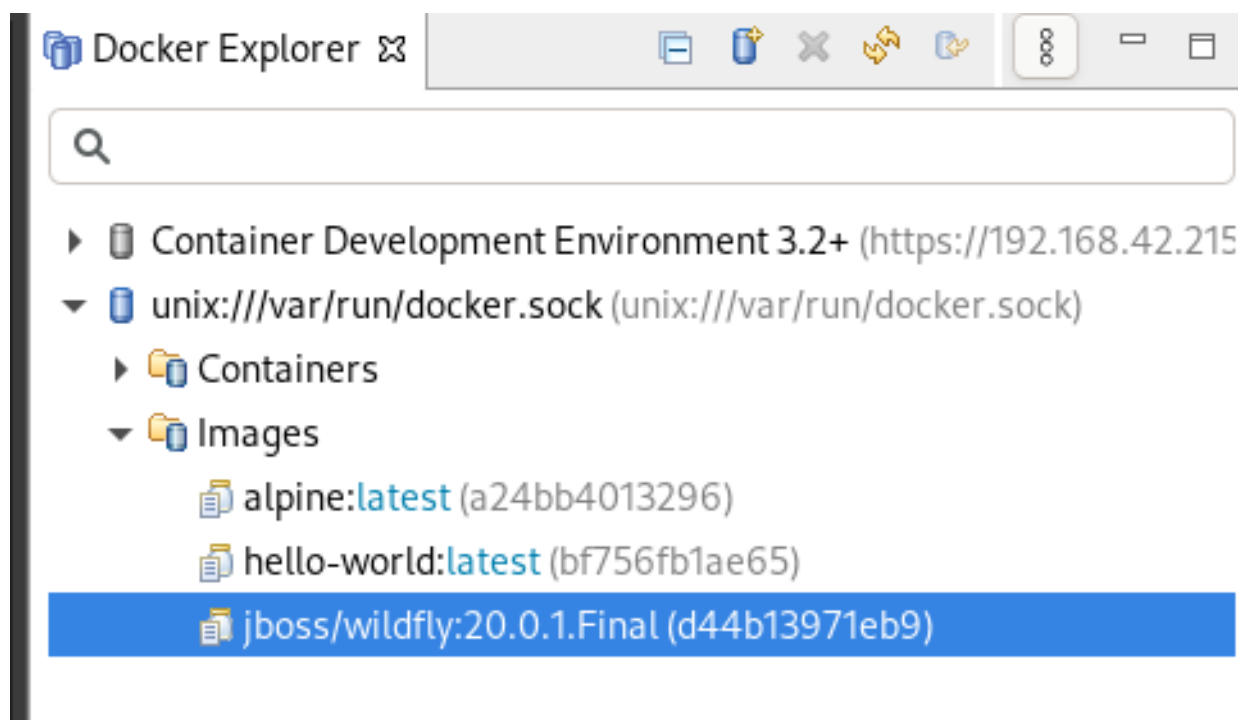


11. Choose a tag for your image.
12. Click **Finish**.
The **Search the Docker Registry for images** window appears.



13. Click **Finish**.

Your new Docker image is now listed in the **Docker Explorer** view.



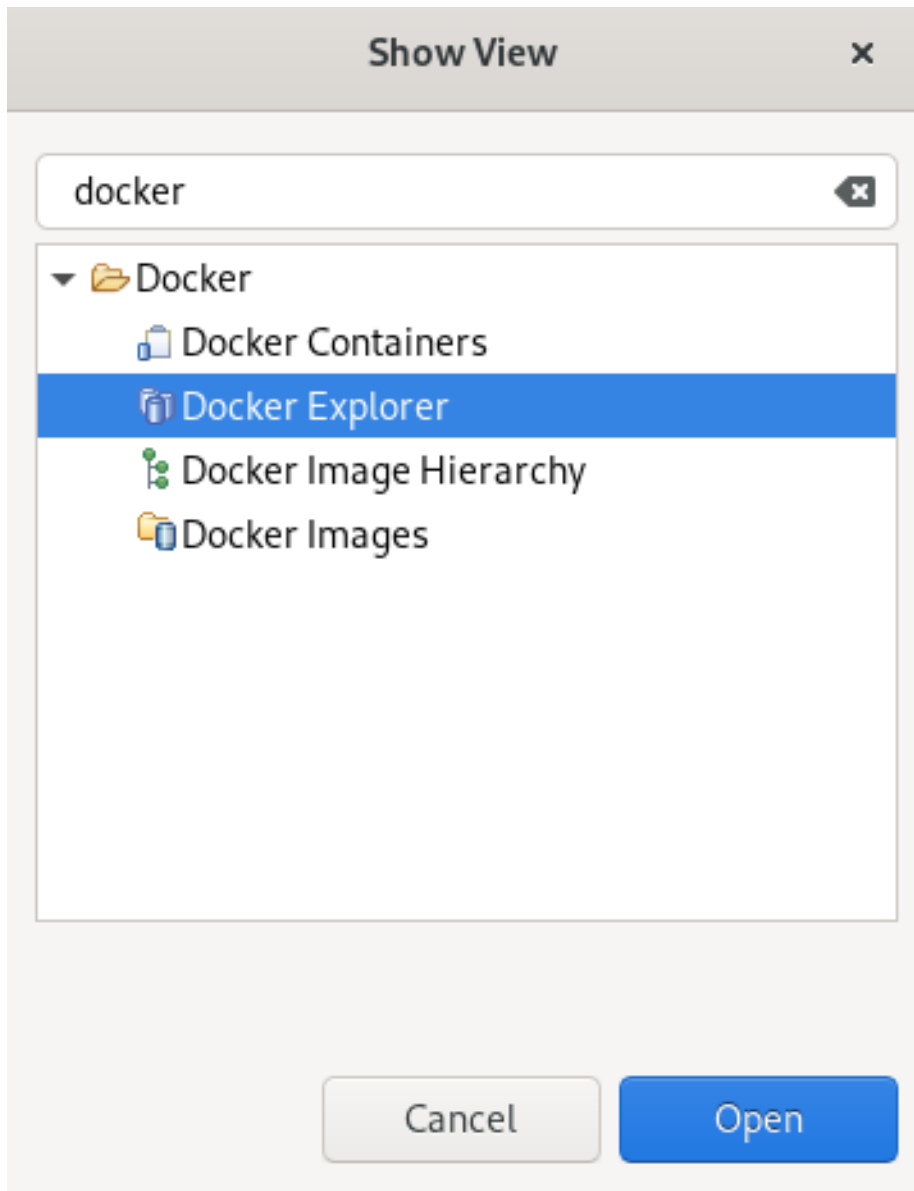
3.2.2. Pushing Docker images

Before pushing a Docker image you must tag it. The following section describes how to tag and push a Docker image in CodeReady Studio.

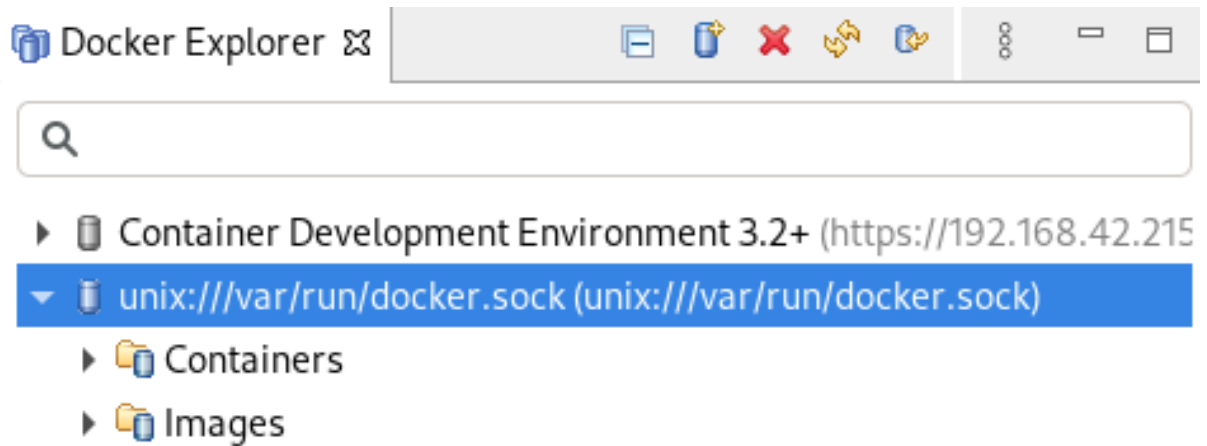
Procedure

1. Start CodeReady Studio.

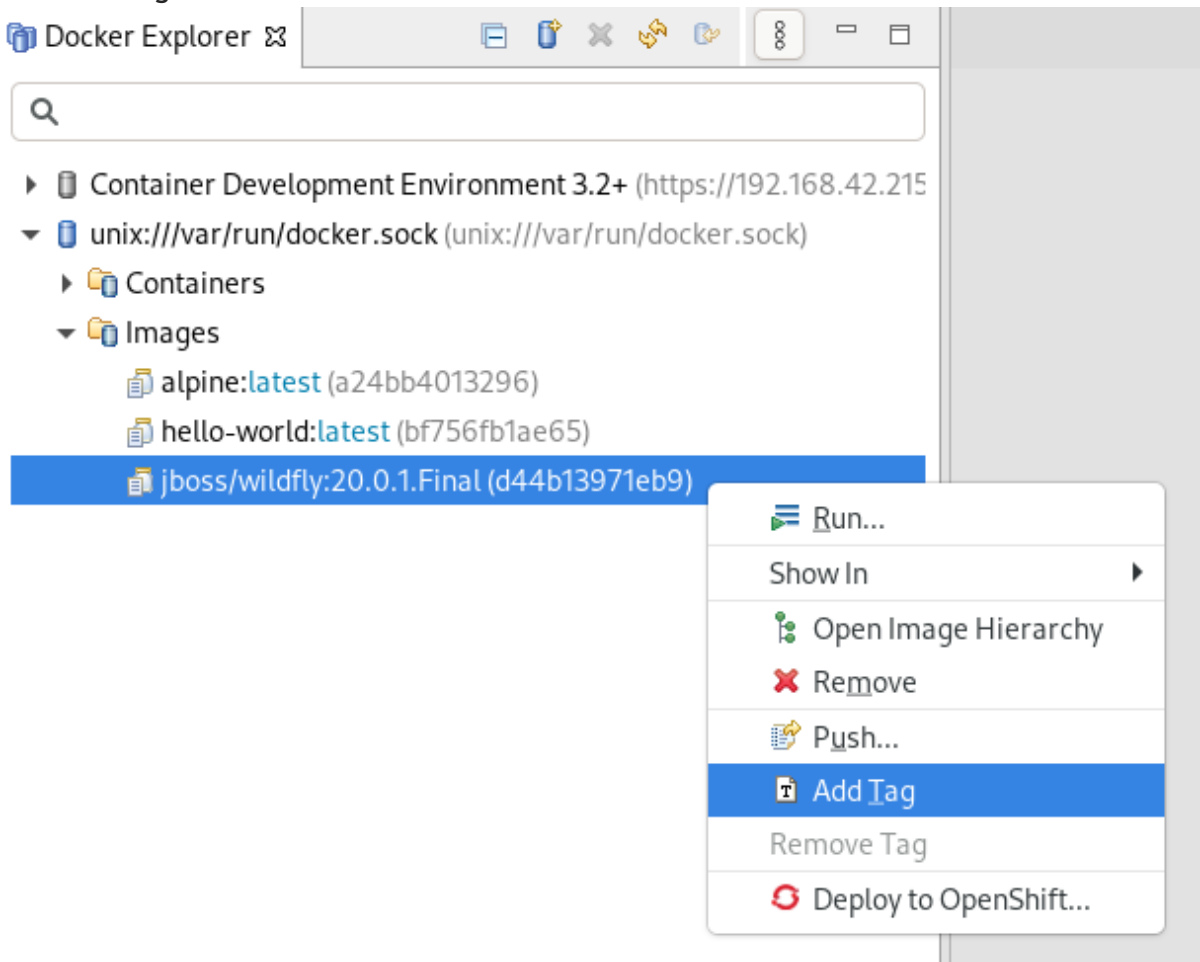
2. Click **Window** → **Show View** → **Other**.
The **Show View** window appears.



3. Enter **Docker** in the search field.
4. Select **Docker Explorer**.
5. Click **Open**.
The **Docker Explorer** view appears.

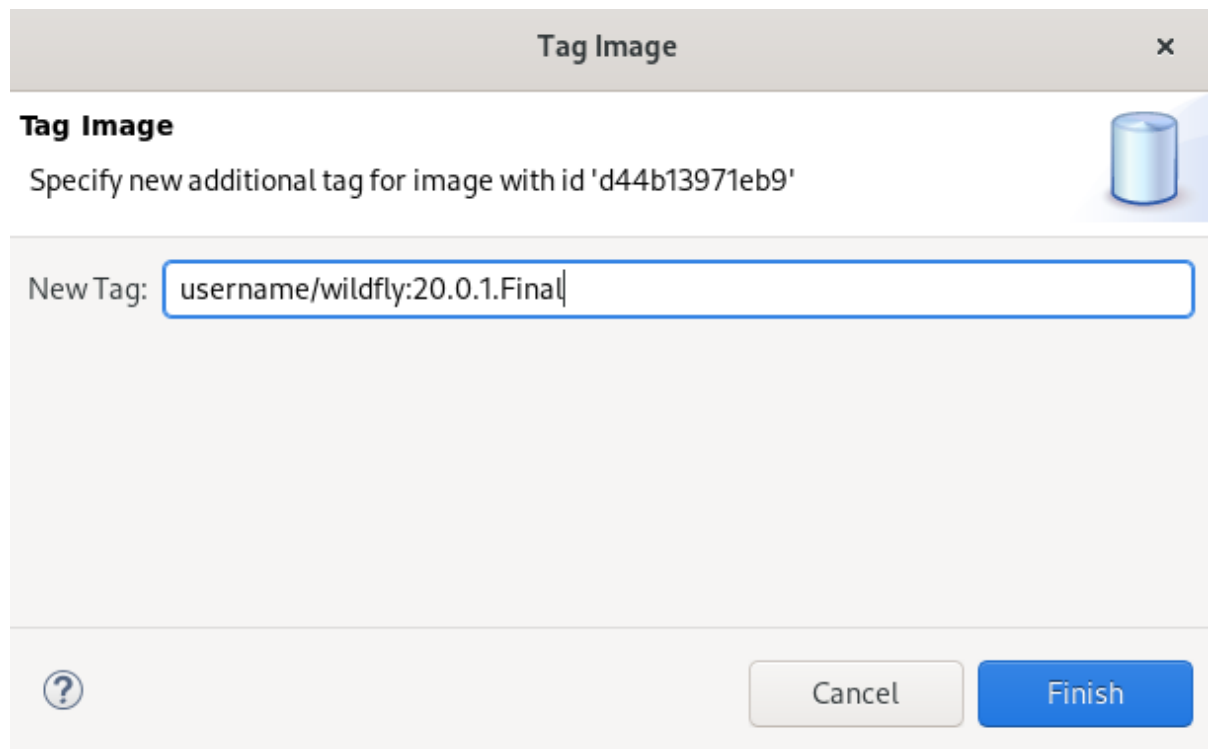


6. Expand **Docker socket** → **Images**.
7. Right-click the image you want to tag.
8. Click **Add tag**.

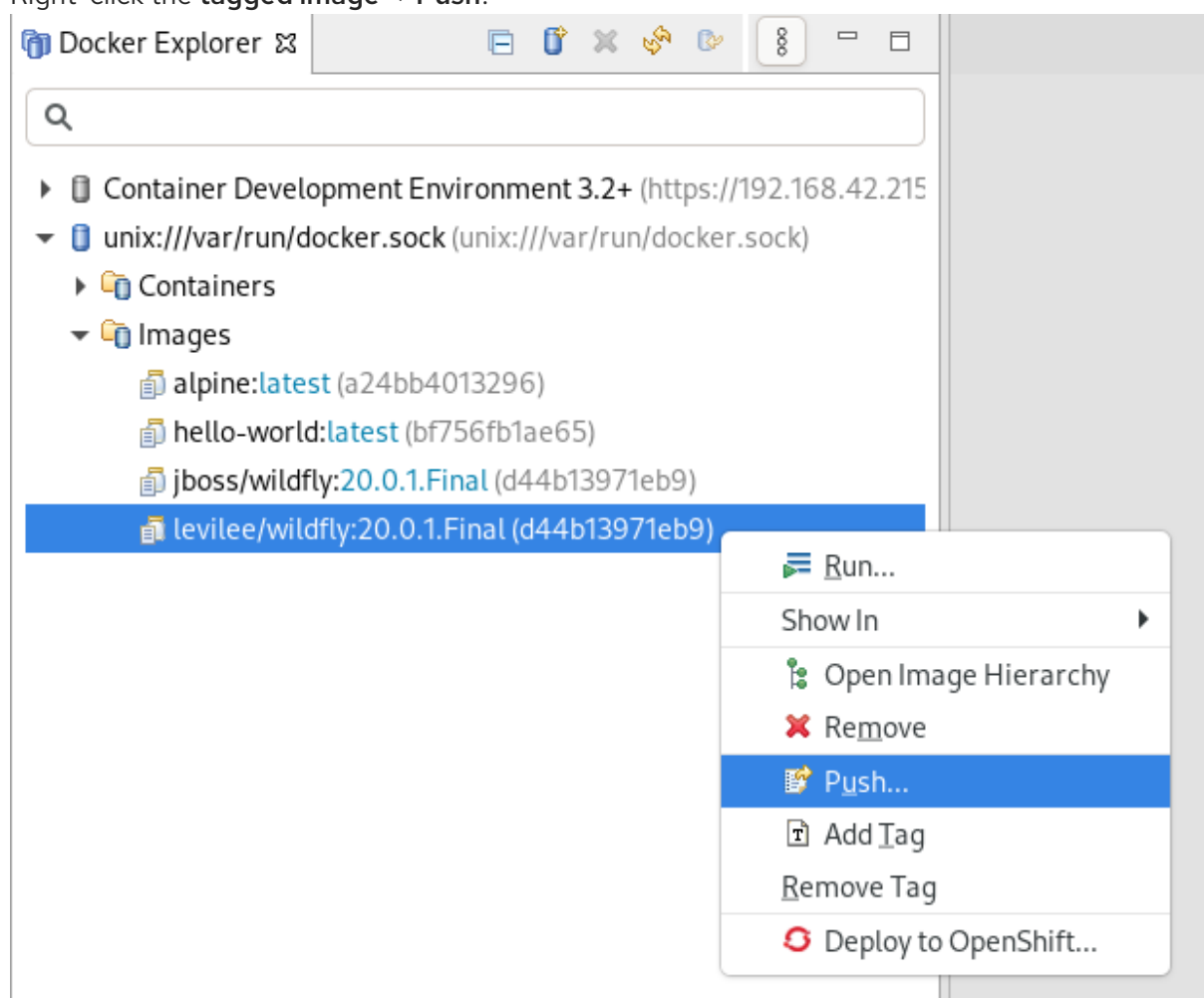


The **Tag Image** window appears.

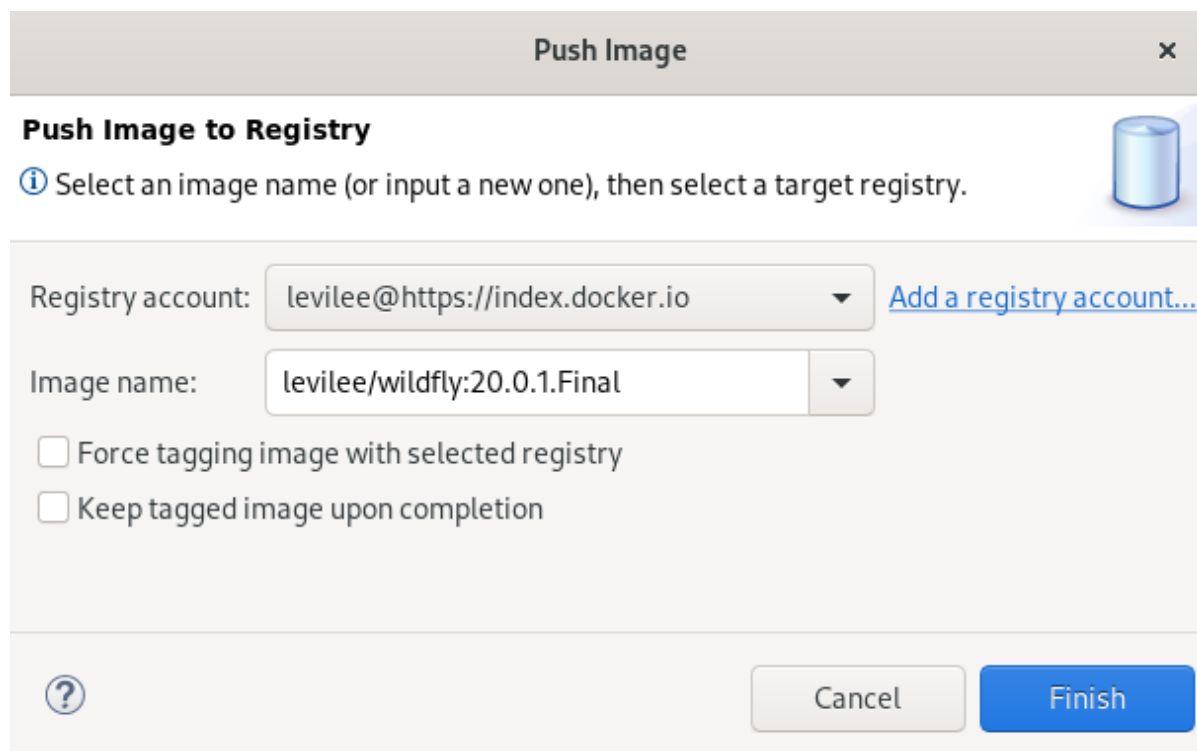
9. Enter the tag in the **New Tag** field.
 The tag should be in the form of **username/image_name:tag_name**, where **username** is your Docker ID on <https://hub.docker.com>, **image_name** is the name of your image, and **tag_name** is the version of the image.



10. Click **Finish**.
11. Right-click the **tagged image** → **Push**.



The **Push image to Registry** window appears.



Push Image ×

Push Image to Registry

❗ Select an image name (or input a new one), then select a target registry.

Registry account: levilee@https://index.docker.io ▼ [Add a registry account...](#)

Image name: levilee/wildfly:20.0.1.Final ▼

☐ Force tagging image with selected registry

☐ Keep tagged image upon completion

ⓘ Cancel Finish

12. Select the **Registry Account** that starts with your Docker ID.

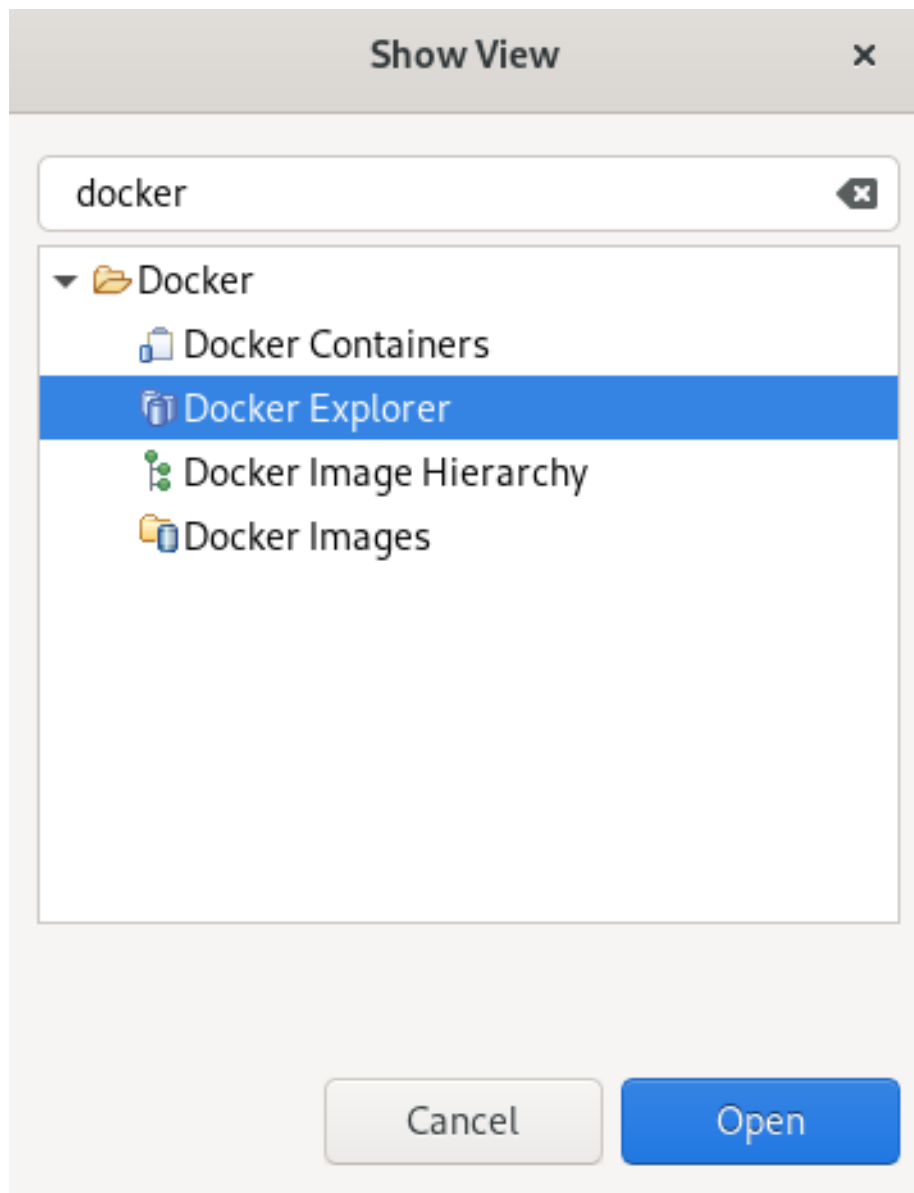
13. Click **Finish**.

After you push the image it appears in the Docker Cloud. This image is then available for other developers to use.

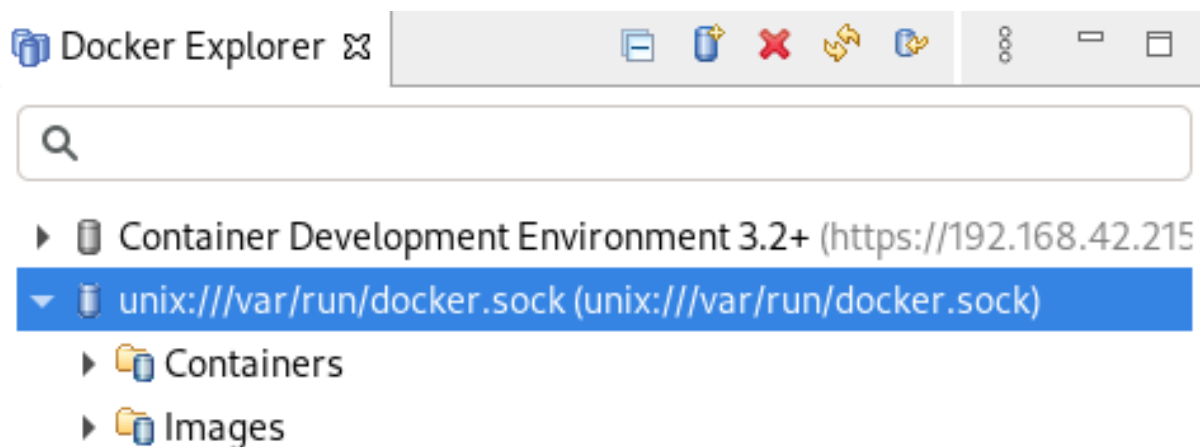
3.2.3. Running Docker images

Procedure

1. Start CodeReady Studio.
2. Click **Window → Show View → Other**.
The **Show View** window appears.



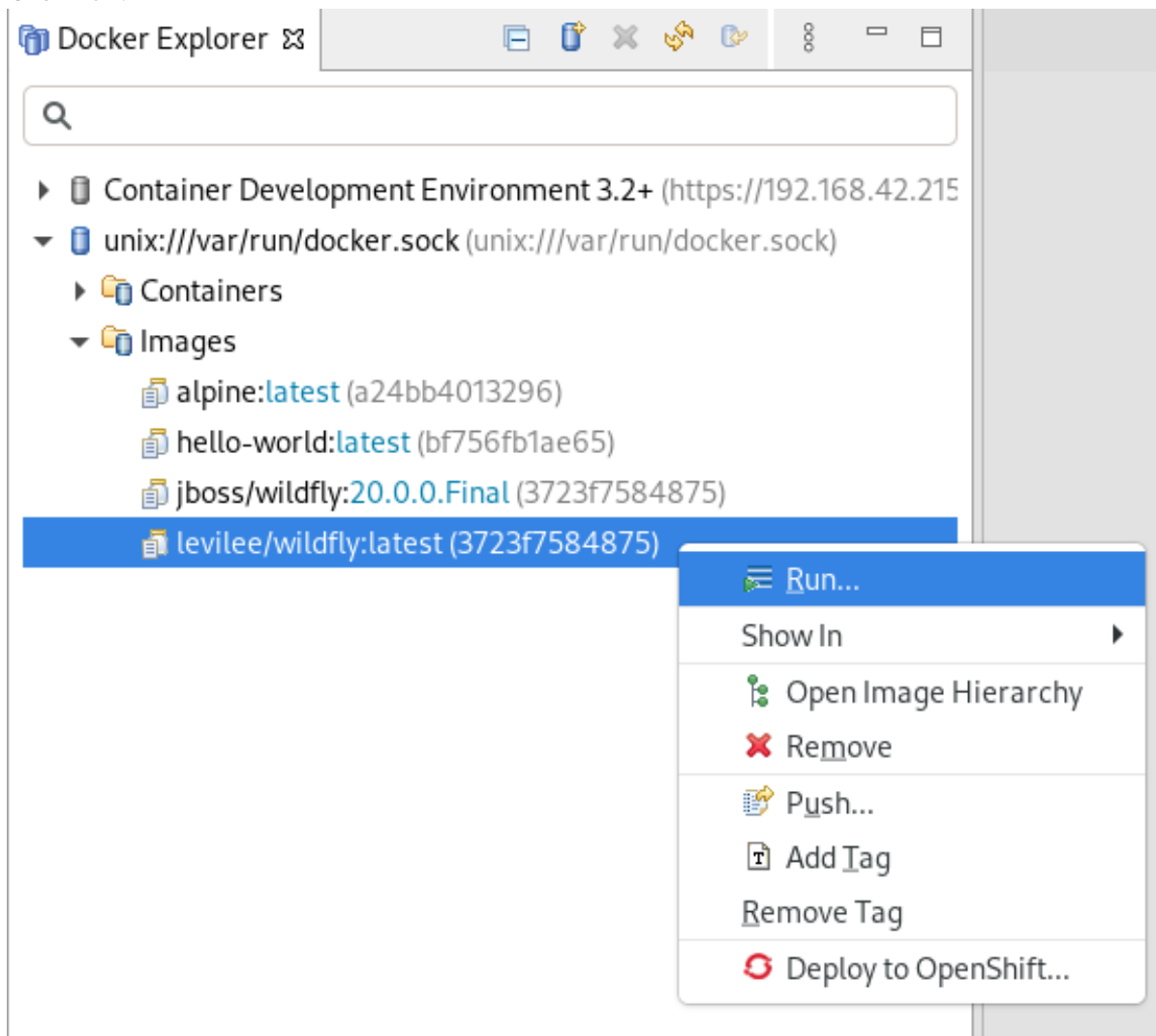
3. Enter **Docker** in the search field.
4. Select **Docker Explorer**.
5. Click **Open**.
The **Docker Explorer** view appears.



6. Expand **Docker socket** → **Images**.

7. Right-click images you want to run.

8. Click **Run**.



The **Docker Container settings** window appears.

Run a Docker Image

Docker Container settings

Image: ▼ Search...

[Pull this image...](#)

Container Name:

Entrypoint:

Command:

☐ Publish all exposed ports to random ports on the host interfaces
 Only publish the selected container ports below to the host:

Container Port	Type	Host Address	Host Port
<input checked="" type="checkbox"/> 8080	tcp		8080

Add...
Edit...
Remove

Links to other containers:

Container Name	Alias
<input type="text"/>	

Add...
Edit...
Remove

☐ Keep STDIN open to Console even if not attached (-i)
☐ Allocate pseudo-TTY from Console (-t)
☐ Automatically remove the container when it exits (--rm)
☐ Give extended privileges to this container (--privileged)
☐ Use unconfined seccomp profile (--securityOpt seccomp=unconfined)
☐ Add basic security (--readonly --tmpfs /run --tmpfs /tmp --cap-drop=all)

?
< Back
Next >
Cancel
Finish

9. Name the container.

10. Clear the **Publish all exposed ports to random ports on the host interfaces** check box.

11. Check the box for **8080** port.
12. Click **Finish**.
The **Console** view appears showing the process of starting the image.
13. In the web browser navigate to <http://localhost:8080/> to see the image running.



3.2.4. Building images with Dockerfile

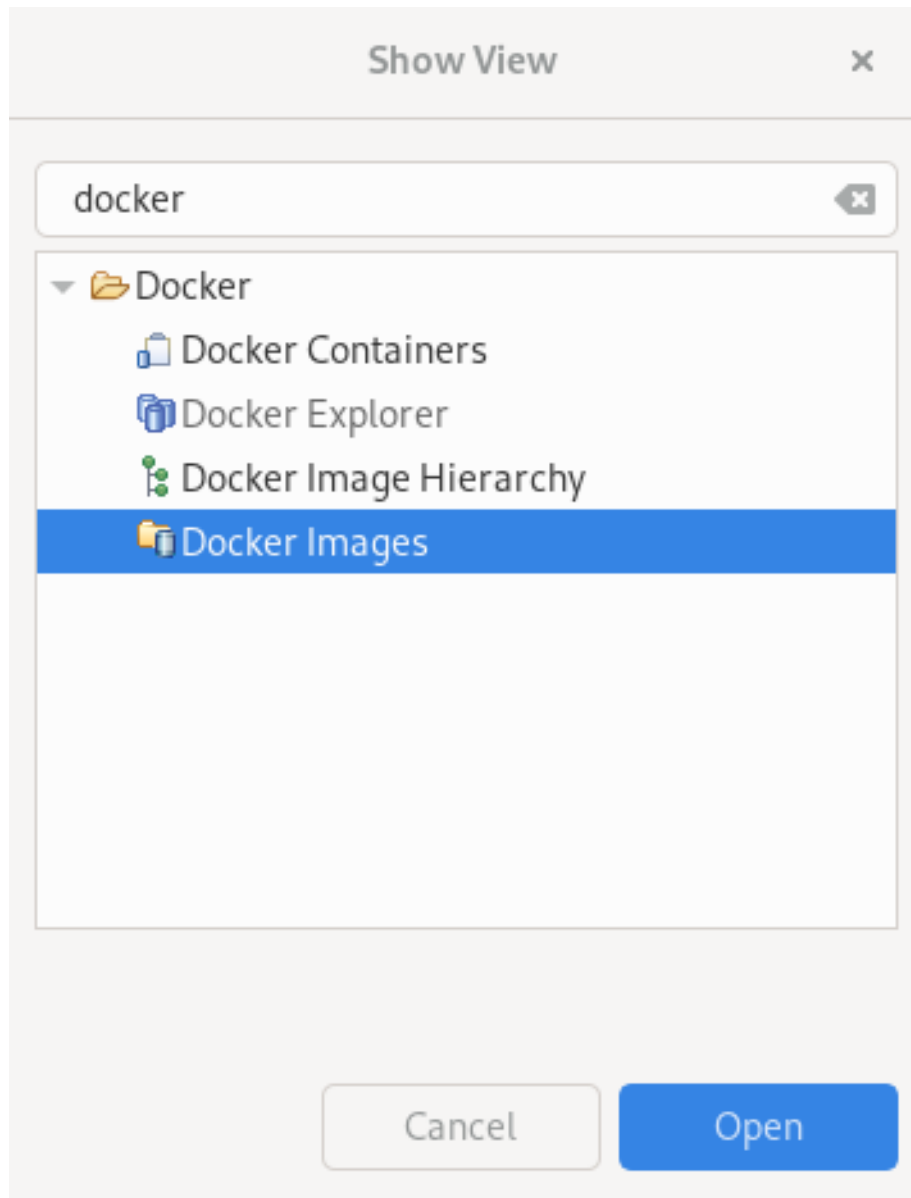
You can build an image or create one by modifying an existing image. Typically, this involves installing new packages. The specification of the new Docker image is done via a the **Dockerfile**

Prerequisites

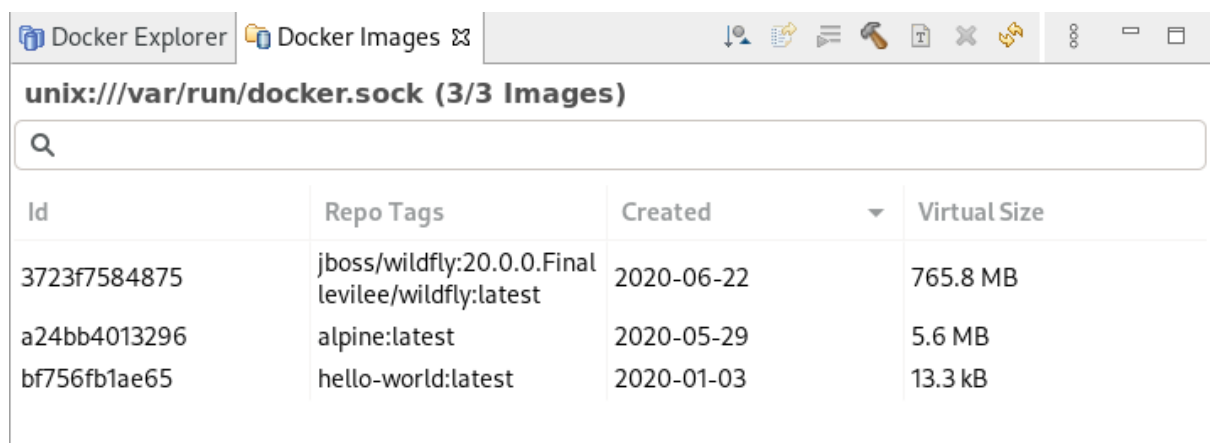
- Dockerfile created on your local machine.
For more information on how to create a Dockerfile, see [Section 1.2.2.1, "Creating a Dockerfile"](#).

Procedure

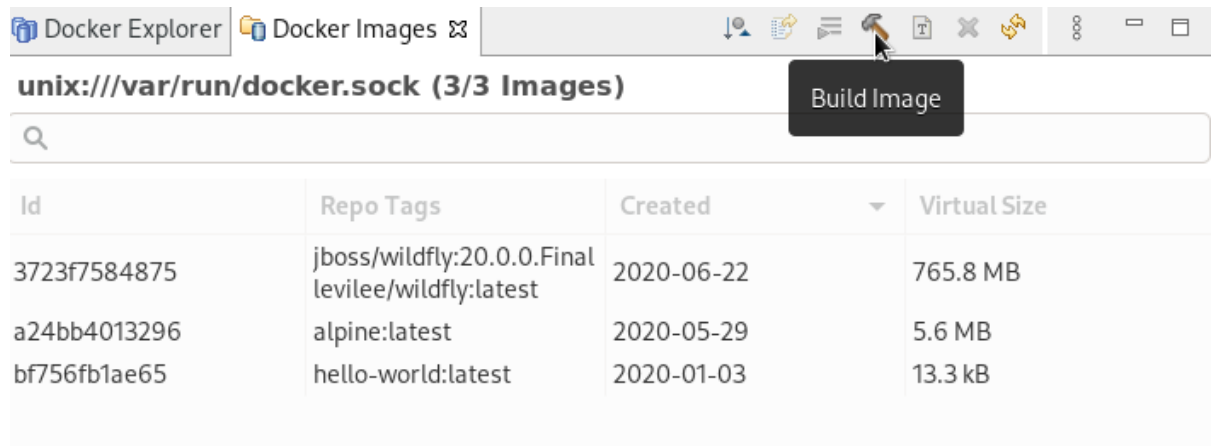
1. Start CodeReady Studio.
2. Click **Window** → **Show View** → **Other**.
The **Show View** window appears.



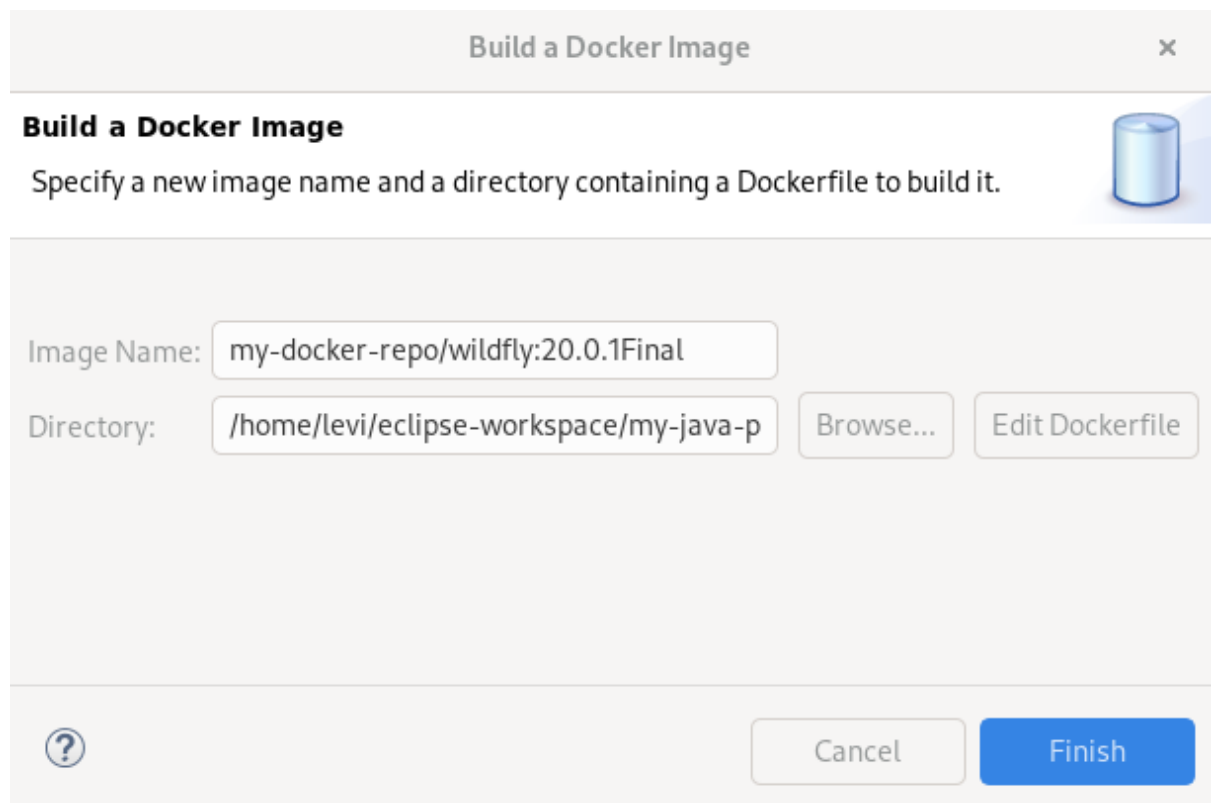
3. Enter **Docker** in the search field.
4. Select **Docker Images**.
5. Click **Open**.
The **Docker Images** view appears.



6. Click the **Build Image** icon.



The **Build a Docker Image** window appears.



7. Name the image in the format of **repo/name:version**.
8. Click **Browse** to locate the Dockerfile.
9. Click **Finish**.

The **Console** view appears displaying the build process.

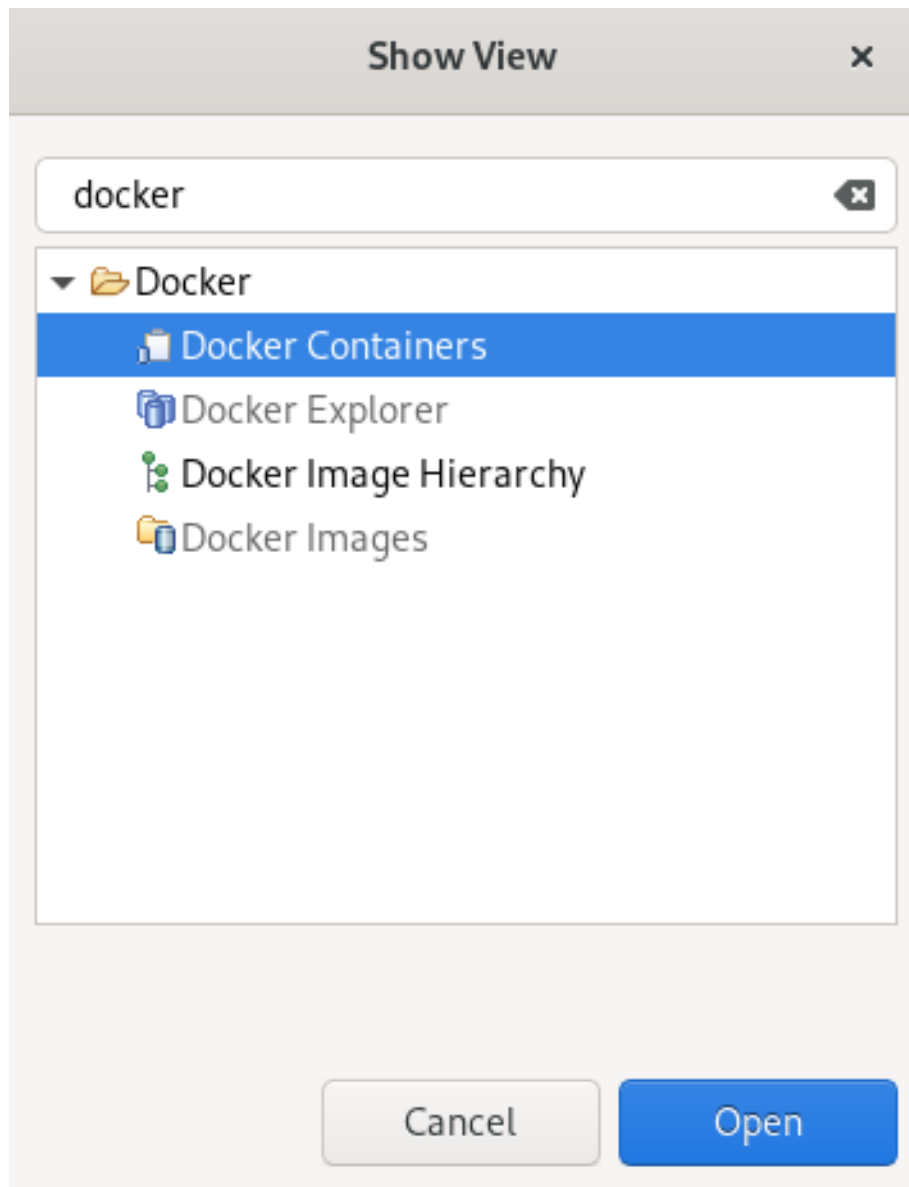
3.3. MANAGING DOCKER CONTAINERS

Docker containers are isolated processes that are based on Docker images. Once created, users can stop, start, pause, unpause, kill, or remove the containers, as well as read their logs.

The following section describes how to manage Docker containers in CodeReady Studio.

Procedure

1. Start CodeReady Studio.
2. Click **Window** → **Show View** → **Other**.
The **Show View** window appears.



3. Enter **Docker** into the search field.
4. Select **Docker Containers**.
5. Click **Open**.
The **Docker Containers** view appears.

Docker Containers ⓘ

unix:///var/run/docker.sock (5 Containers)

Q

Name	Image	Created	Command	Ports	Status
blissful_lamarr	sha256:4109fef685	2020-07-21	/bin/sh -c 'cd \$HOM		Exited (6) 10 minute
cool_blackwell	sha256:53abac4f18	2020-07-21	/bin/sh -c 'cd \$HOM		Exited (6) 6 minutes
my-docker-conta	levilee/wildfly:lates	2020-07-21	/opt/jboss/wildfly/b	0.0.0.0:8080->808	Up About an hour
practical_bell	hello-world	2020-07-21	/hello		Exited (0) 6 hours ag
relaxed_lalande	hello-world	2020-07-21	/hello		Created

You can start, pause, unpause, stop, kill, restart, remove, or refresh the containers by using the panel.

