Red Hat CodeReady Studio 12.19

Getting Started with Container and Cloud-based Development

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

Last Updated: 2021-04-20

Red Hat CodeReady Studio 12.19 Getting Started with Container and Cloud-based Development

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

Eva-Lotte Gebhardt
egebhard@redhat.com

Levi Valeeva
levi@redhat.com

Supriya Takkhi
sbharadw@redhat.com

Yana Hontyk
yhontyk@redhat.com

Legal Notice

Abstract

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

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

CHAPTER 1. DEVELOPING USING CONTAINERS AND THE CLOUD IN CODEREADY STUDIO

1.1. USING RED HAT CODEREADY CONTAINERS TOOLS 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.

1.1.1. Downloading and installing Red Hat CodeReady Containers

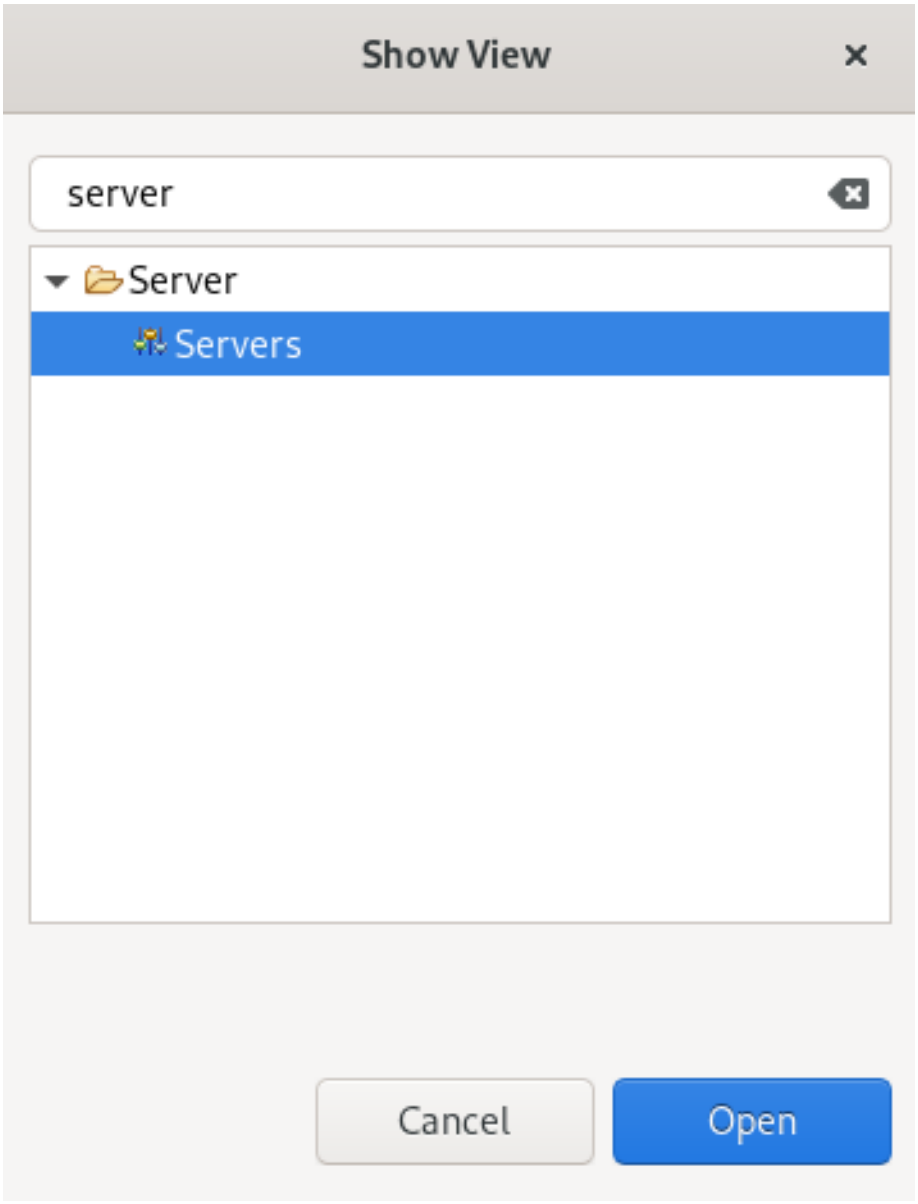
The following section describes how to set up CodeReady Containers in CodeReady Studio.

Prerequisites

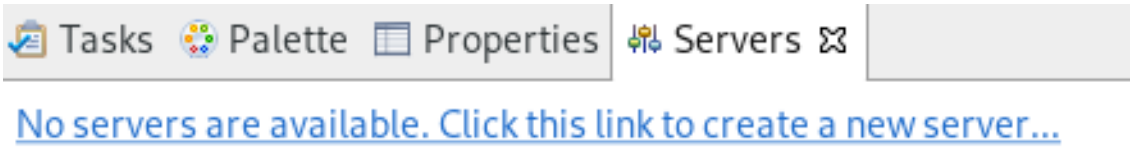
1. Download the latest release of CodeReady Containers and the pull secret .
2. Extract the CRC file.
   For more information on how to install and set up CRC, see the Installation chapter of the Getting started with CodeReady Containers Guide.

Procedure

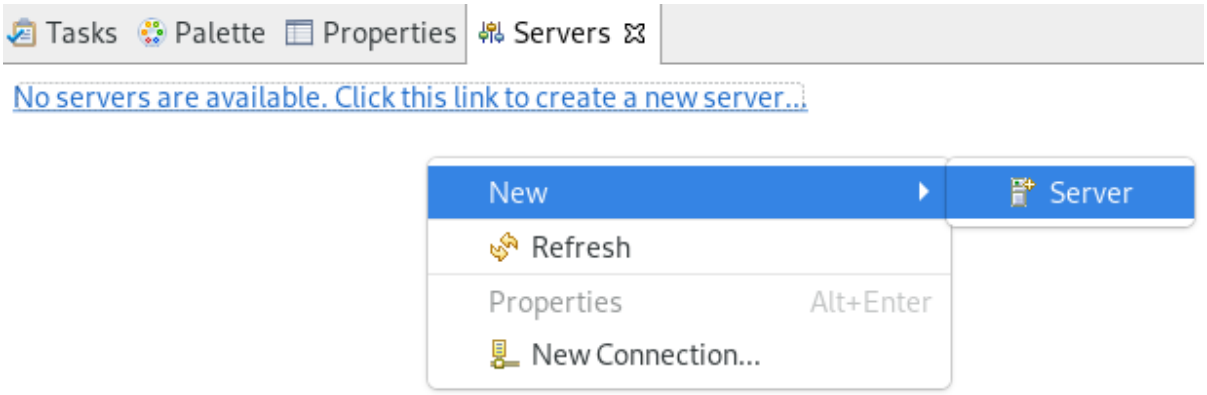
1. Start CodeReady Studio.
2. Click **Window → Show View → Other**.
   The **Show View** window appears.



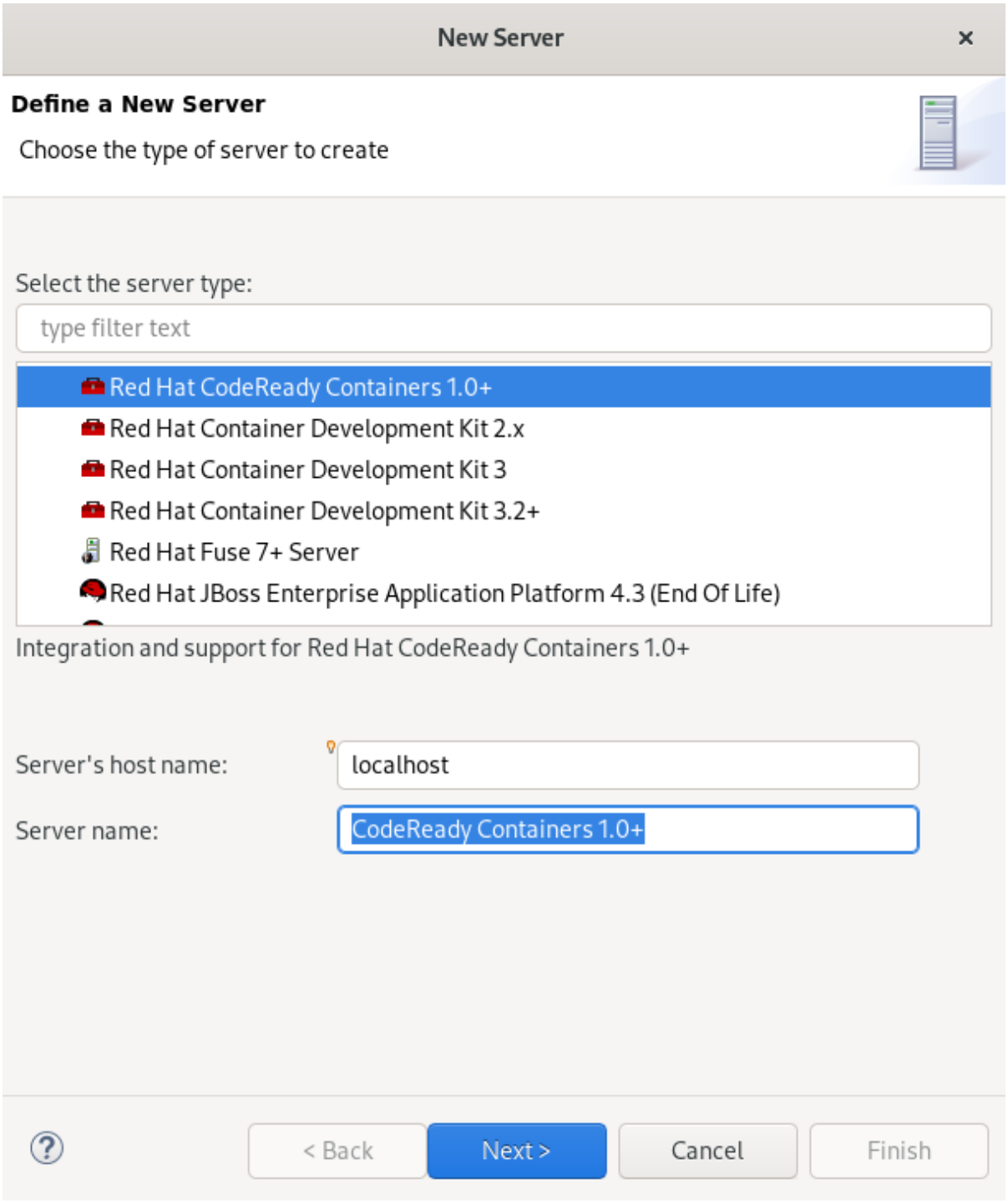
3. Enter **Server** in the search field.
4. Select **Servers**.
5. Click **Open**.
   The **Servers** view appears.



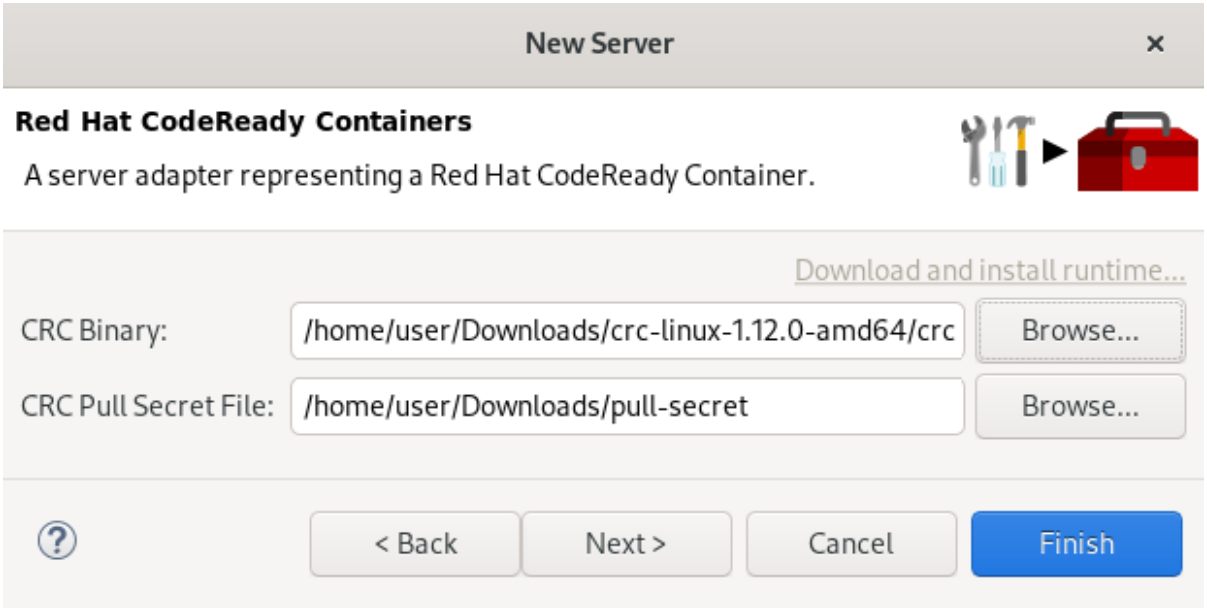
6. Right-click any area in the **Servers** view.



7. Click **New → Server**.
The **Define a New Server** window appears.

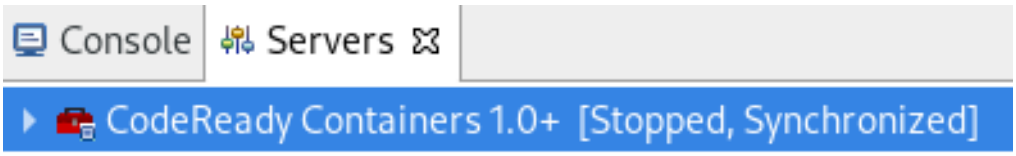


8. Select **CodeReady Containers 1.0+**.
9. Click **Next**.
   The **CodeReady Containers** window appears.



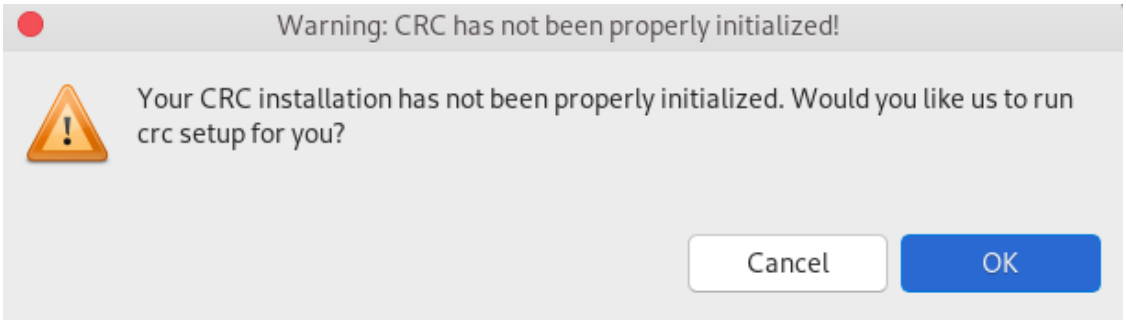
10. Click **Browse** to locate the **CRC binary**.
11. Click **Browse** to locate the **CRC Pull Secret File**
12. Click **Finish**.

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



NOTE

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



Follow the on-screen instructions to initialize CRC.

The instructions prompt you for optional, anonymous usage data collection to assist with development. No personally identifiable information is collected. For information on changing your settings later, visit Red Hat CodeReady Containers - Consent for telemetry data collection.

1.1.2. Using OpenShift Container Platform tools

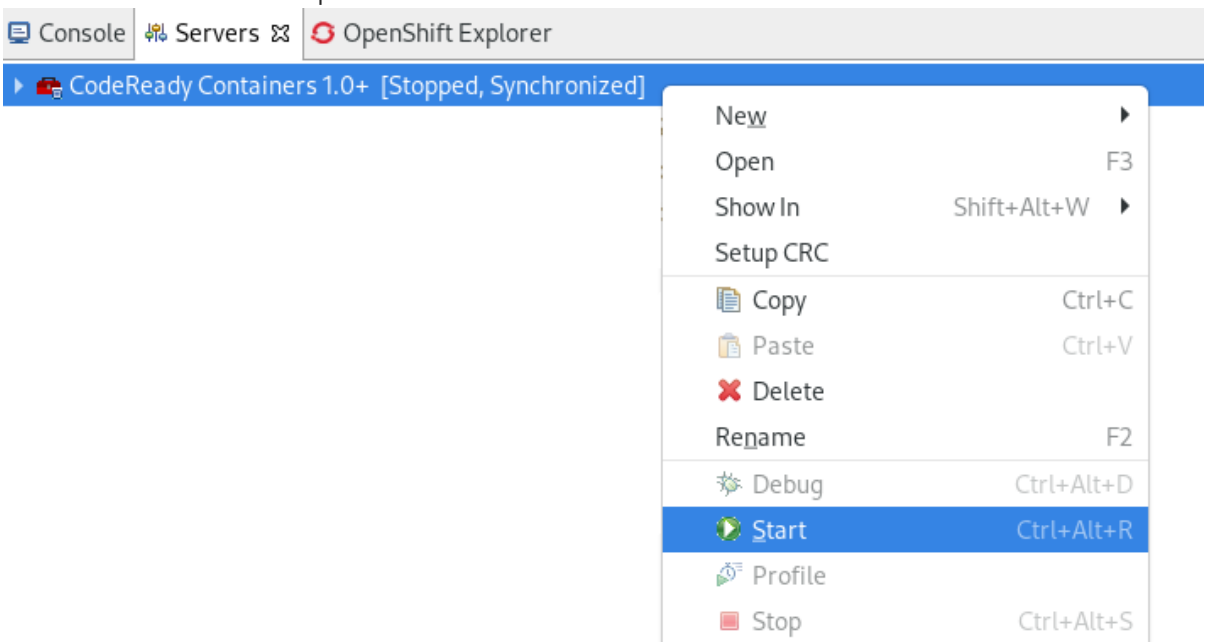
The following section describes how to use OpenShift Containers in CodeReady Studio.

Prerequisites

- The CRC server adapter is set up and configured.
  For more information, see Downloading and installing CRC.

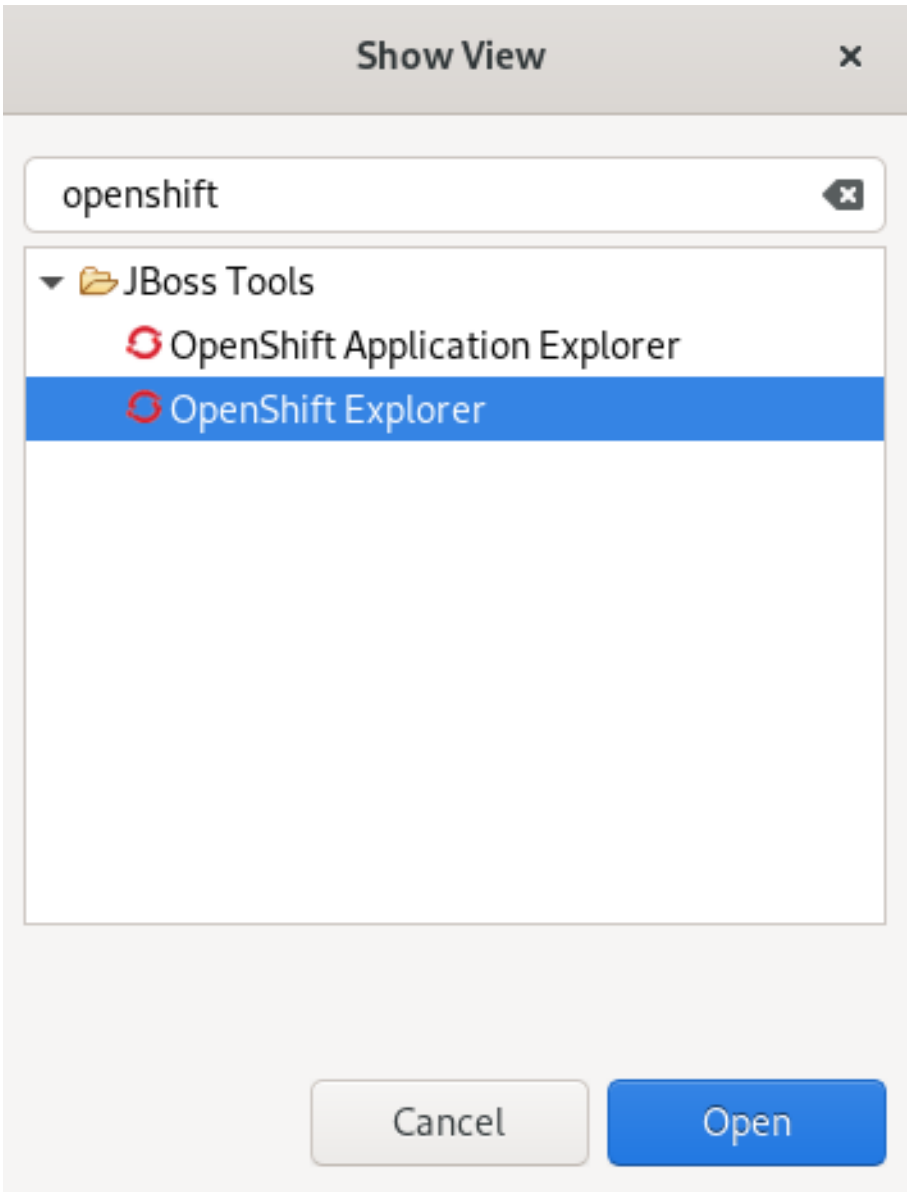
Procedure

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

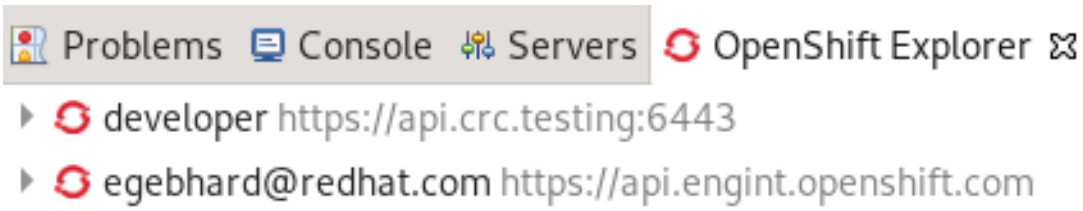


3. Click **Window → Show View → Other**.

The **Show View** window appears.

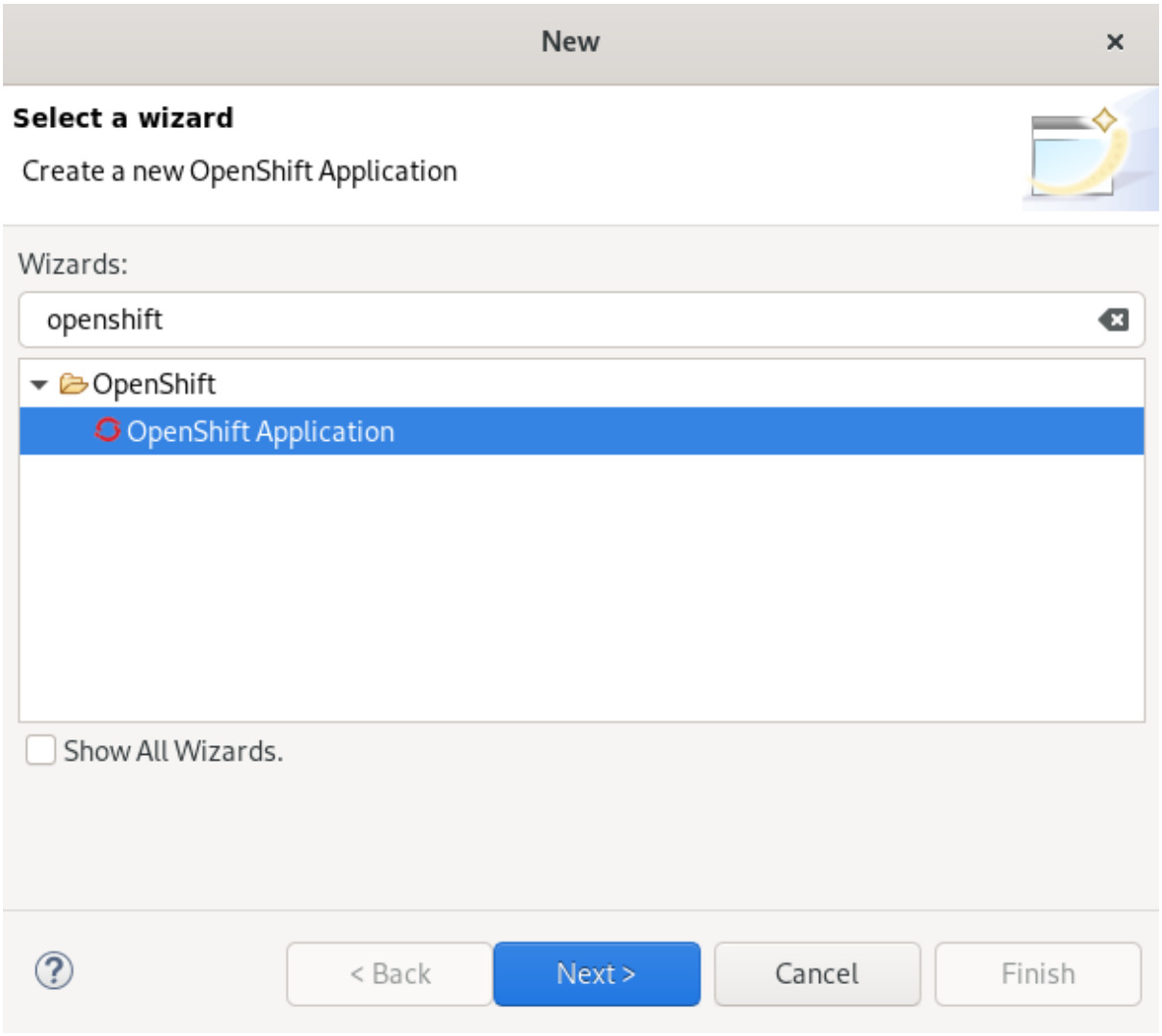


4. Enter **OpenShift** in the search field.
5. Select **OpenShift Explorer**.
6. Click **Open**.
The **OpenShift Explorer** view appears.

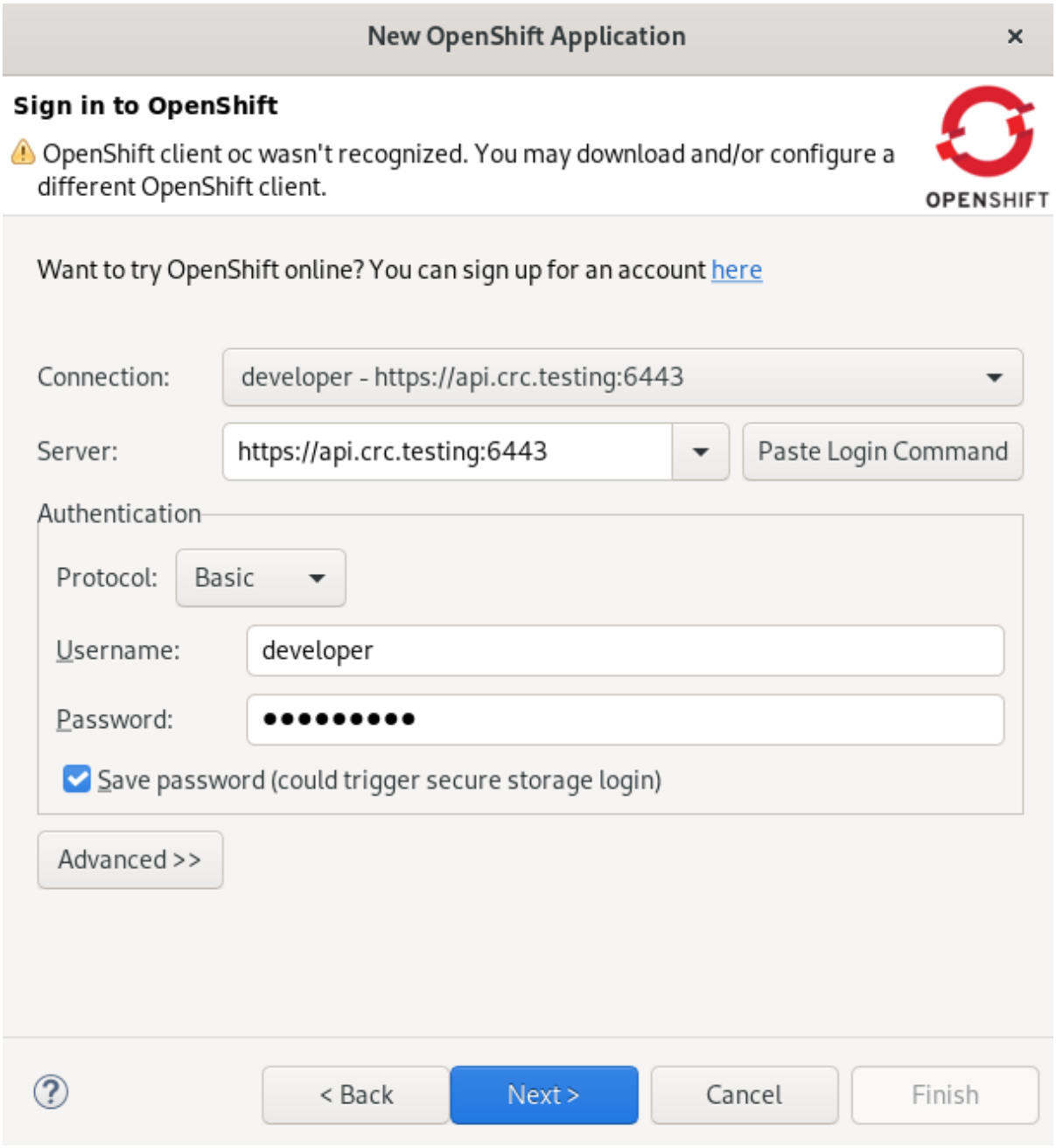


For information on how to set up a new OpenShift connection, visit Creating a new OpenShift Container Platform connection.

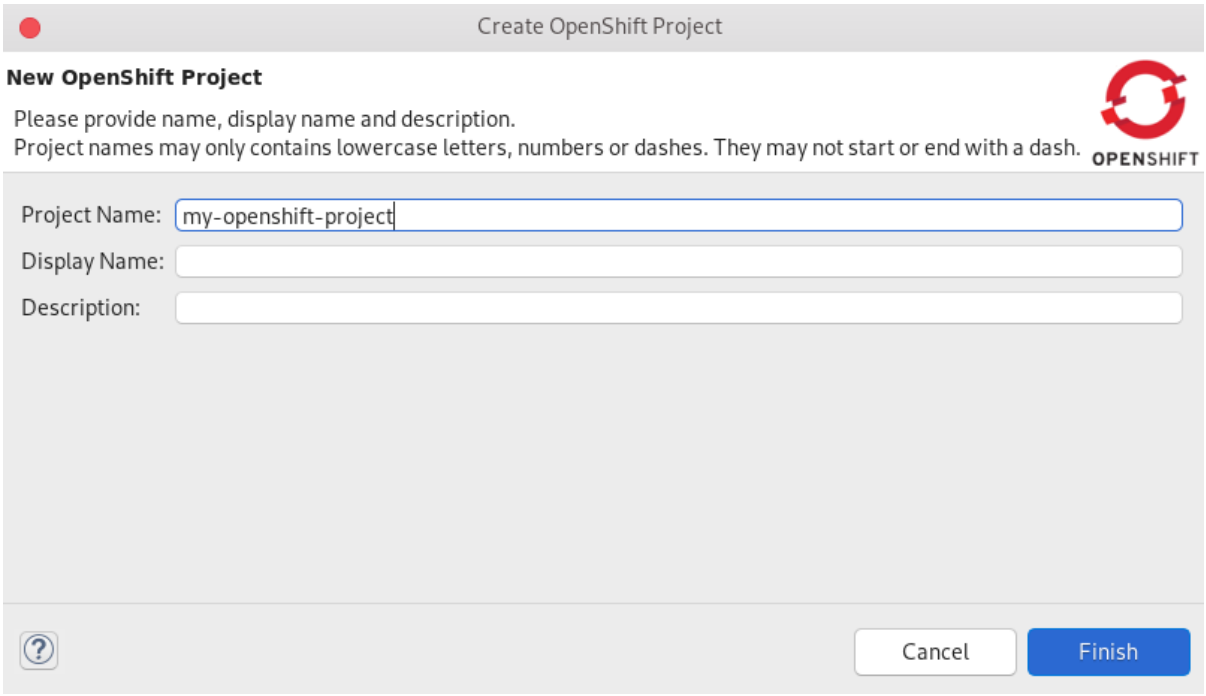
7. Press **Ctrl**+**N**.
The **Select a wizard** window appears.



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

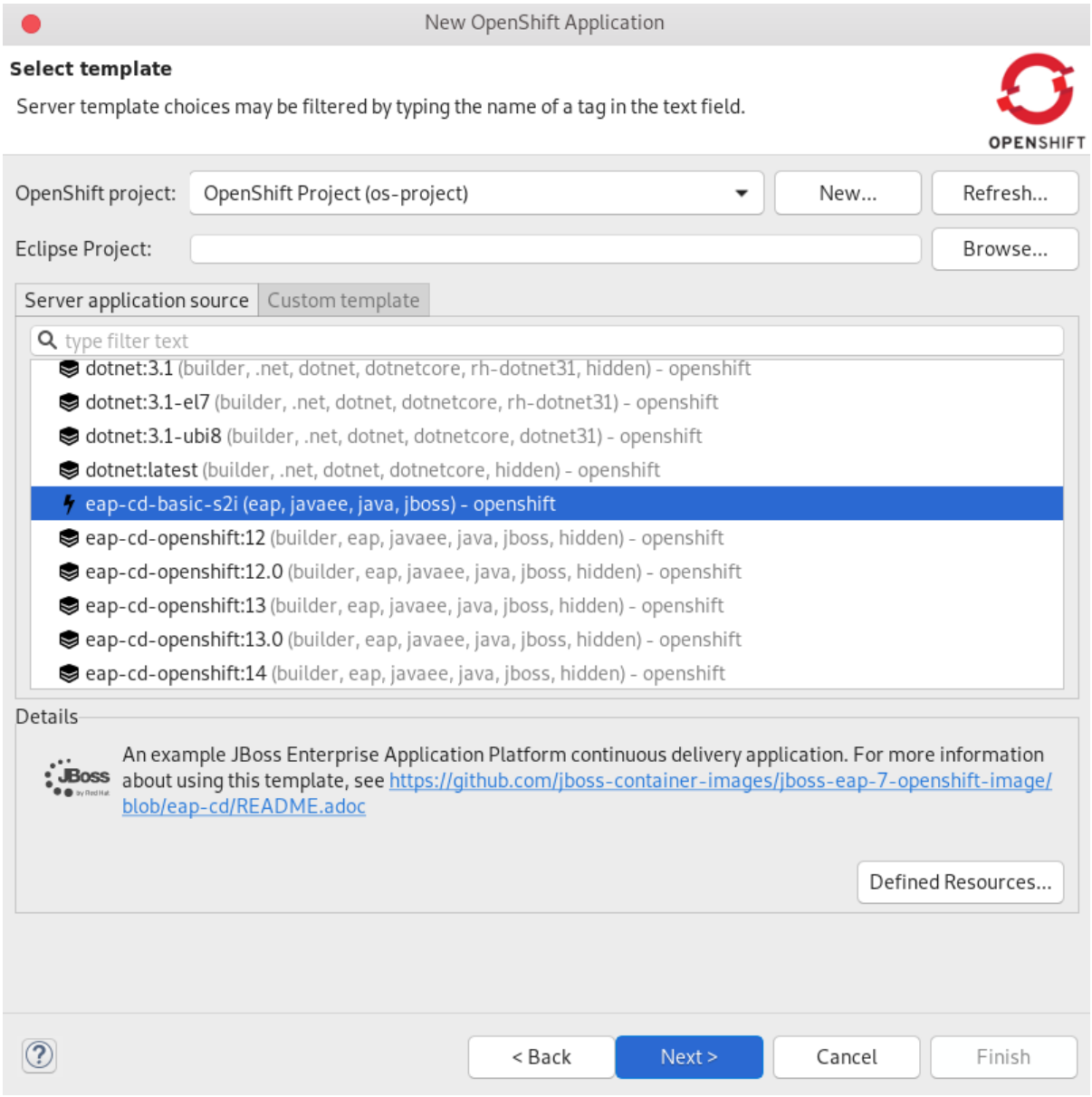


11. Provide your credentials and click **Next**.
The **Create OpenShift Project** window appears.



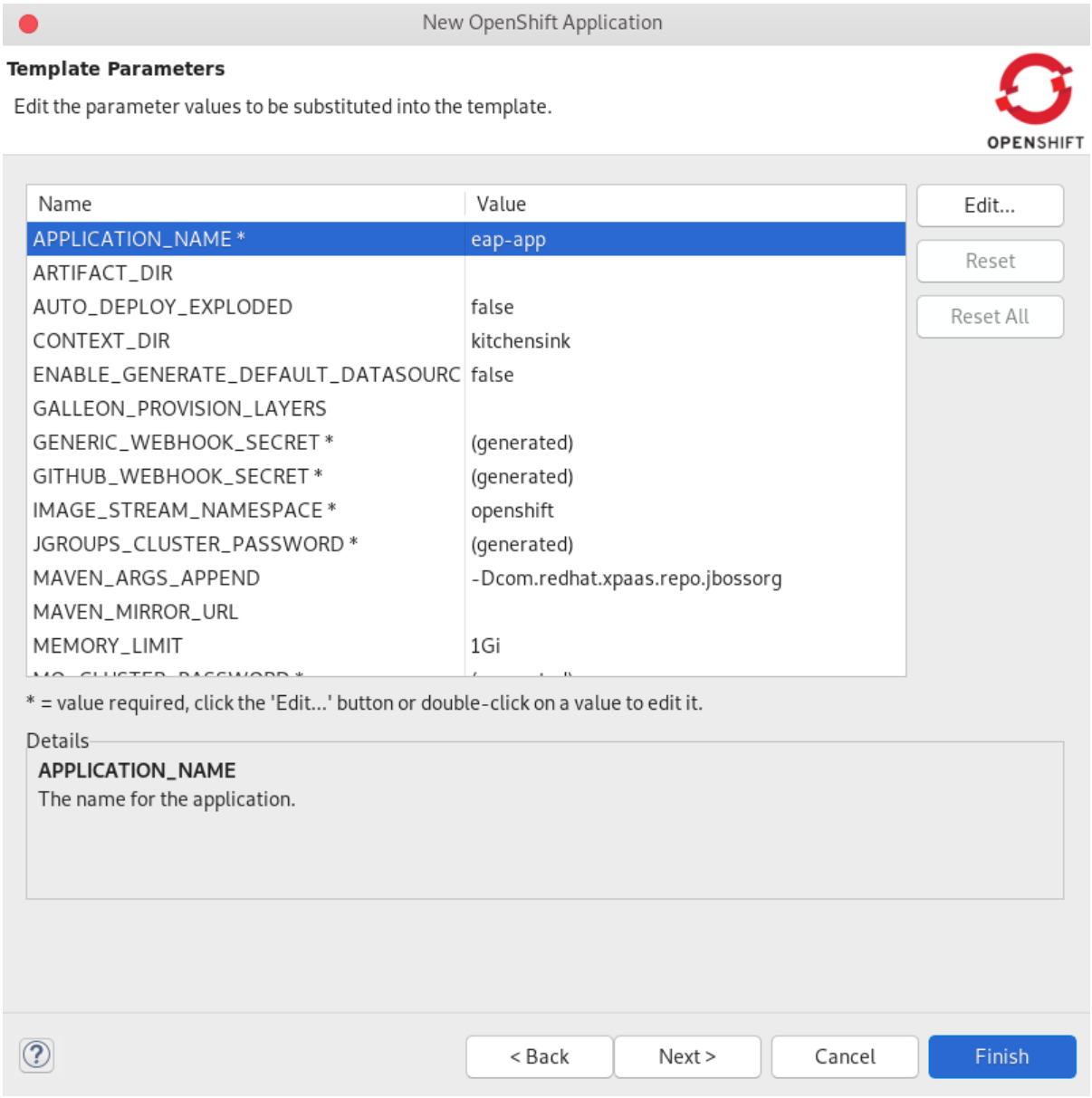
12. Name your project.

13. Click **Finish**.
    The **Select template** window appears.



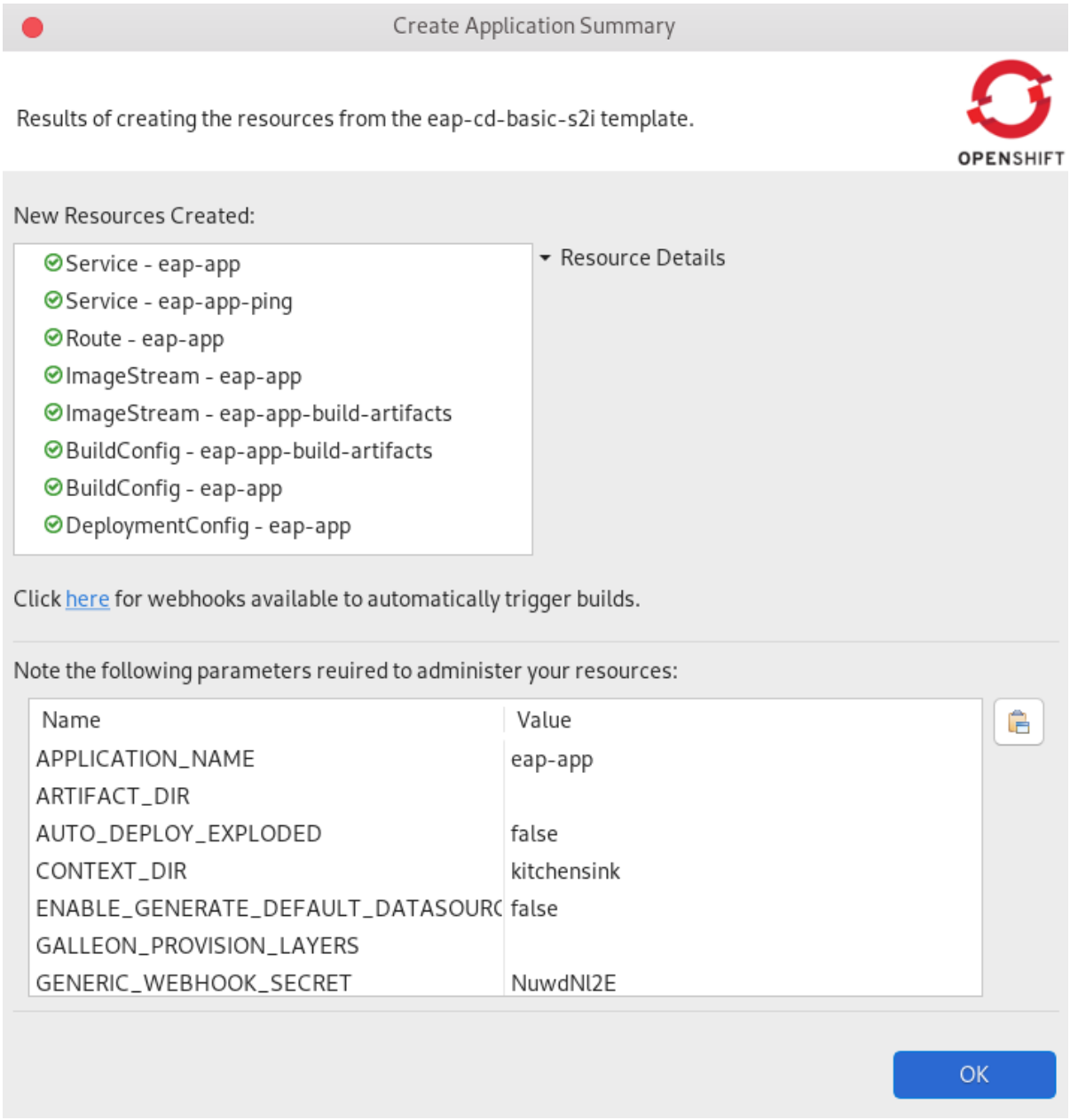
14. Select a template.

15. Click **Next**.
    The **Template Parameters** window appears.



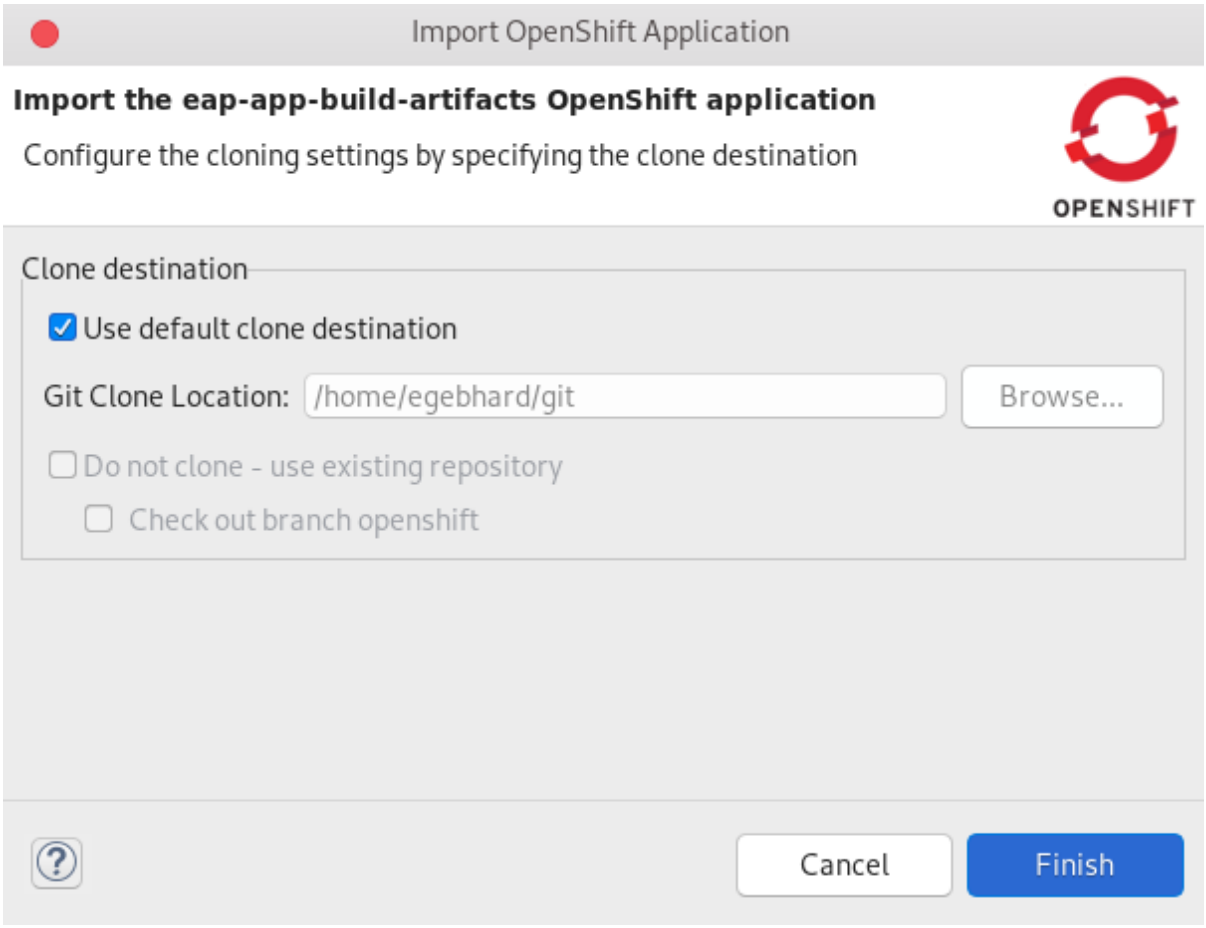
16. Ensure that the template parameters are correct.

17. Click **Finish**.
    The **Create Application Summary** window appears.



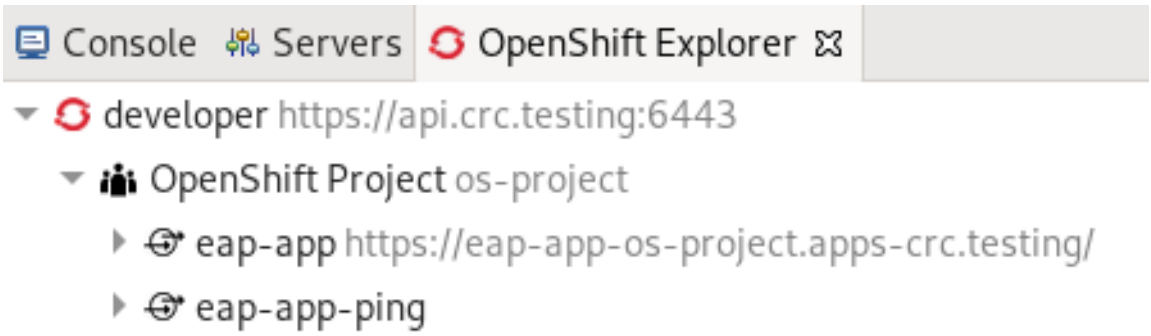
18. Ensure that the application details are correct.

19. Click **OK**.
    The **Import OpenShift Application** window appears.



20. Choose the location for your git repository clone.
21. Click **Finish**.

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



Additional resources

- For more information on how to perform additional tasks with OpenShift Container Platform projects and applications, see Developing for the Cloud with OpenShift in CodeReady Studio .

1.2. USING RED HAT CONTAINER DEVELOPMENT KIT TOOLS 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.

1.2.1. Installing Container Development Kit

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

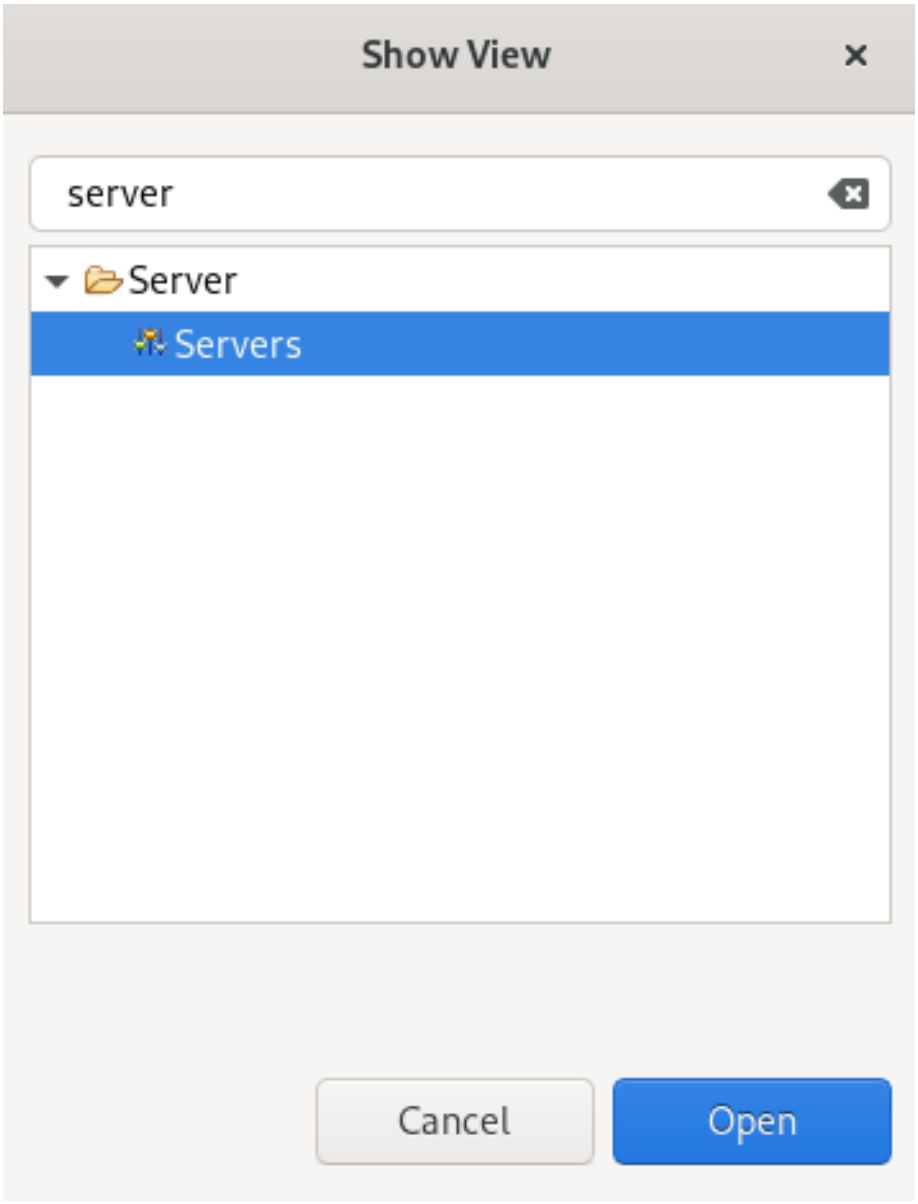
Prerequisites

- 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.
  To create a new account, visit developers.redhat.com.

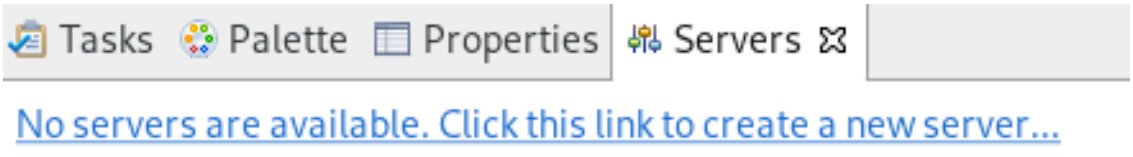
For more information on CDK, see the Red Hat Container Development Kit Getting Started Guide .

Procedure

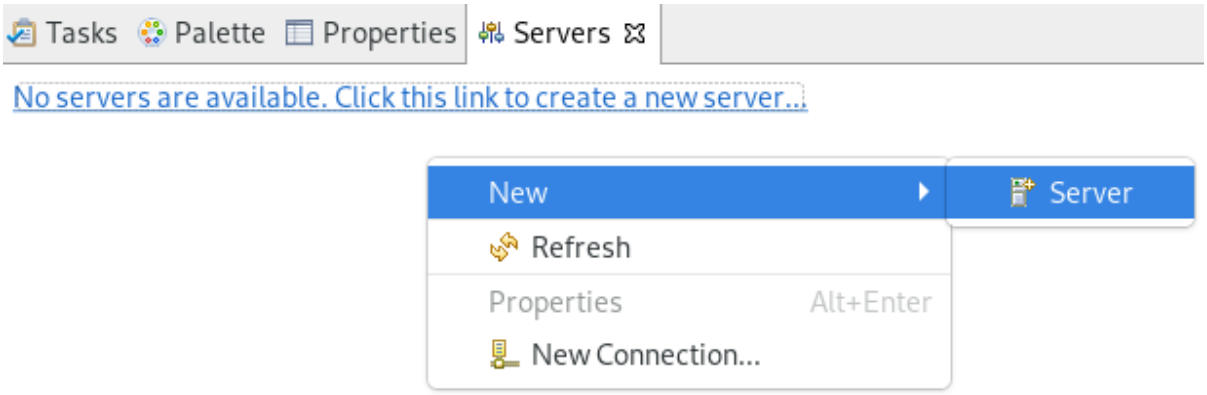
1. Start CodeReady Studio.
2. Click **Window → Show View → Other**.
   The **Show View** window appears.



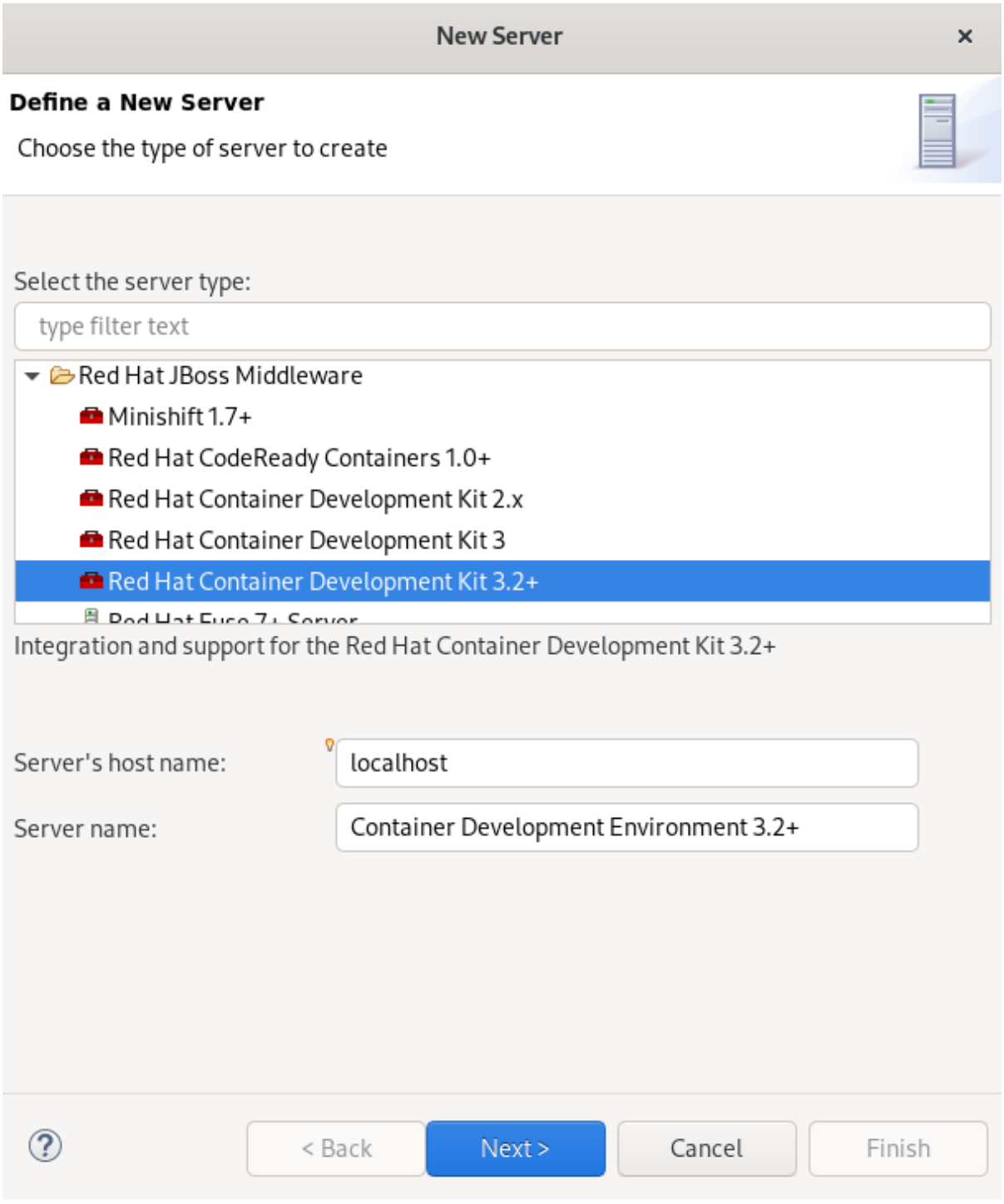
3. Enter **Server** in the search field.
4. Select **Servers**.
5. Click **Open**.
   The **Servers** view appears.



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

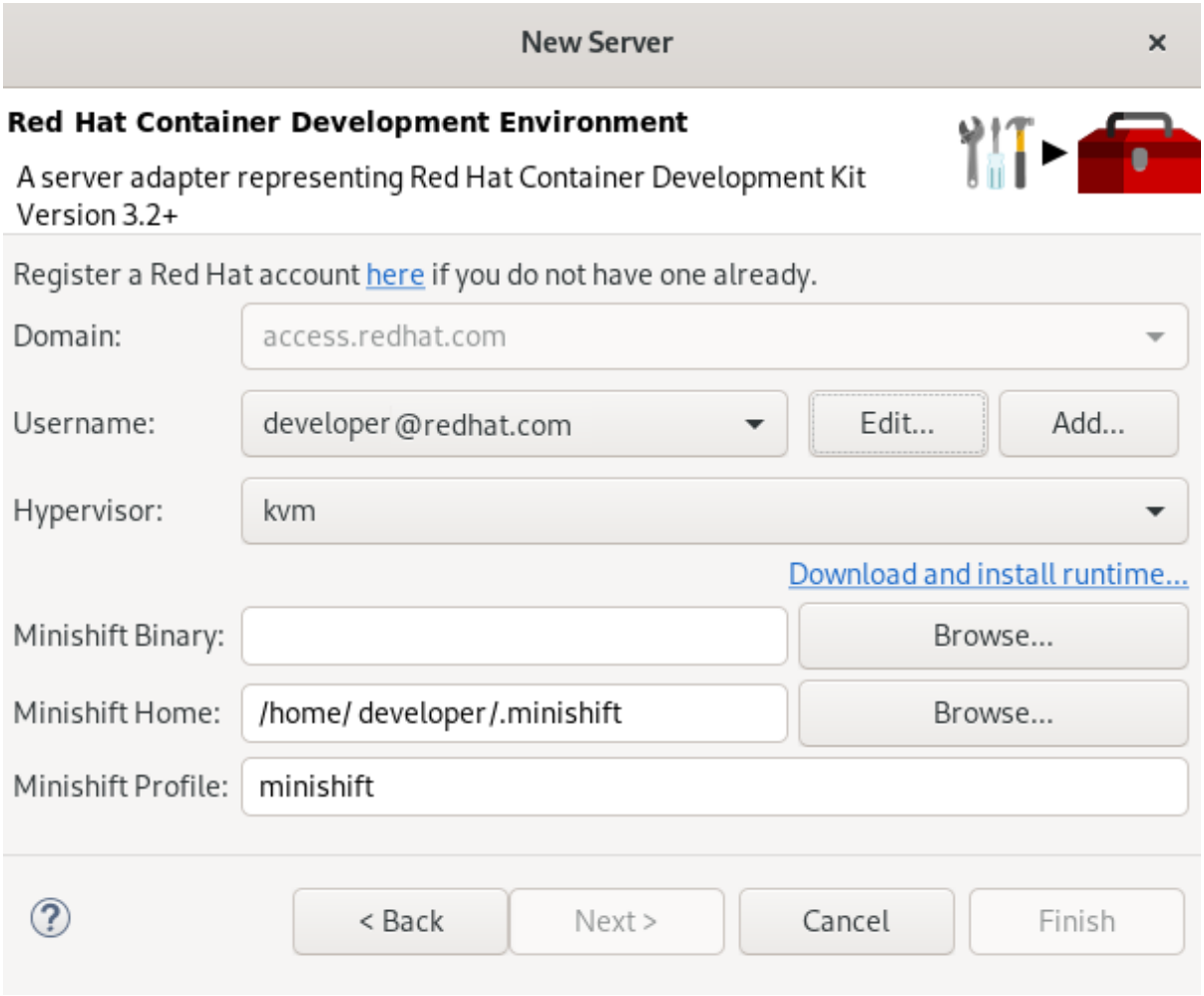


7. Click **New → Server**.
The **Define a New Server** window appears.

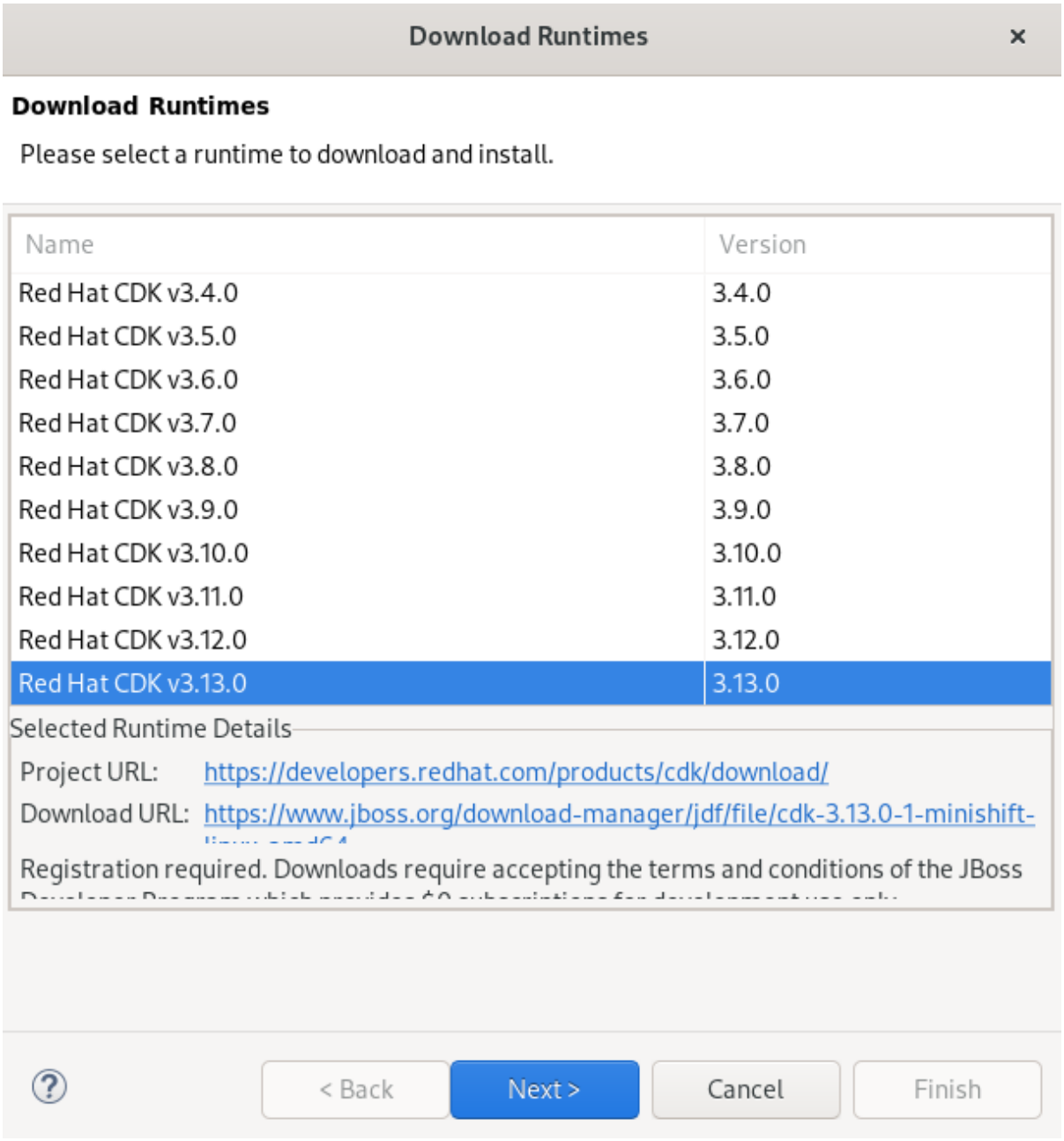


8. Select **Red Hat Container Development Kit 3.2+**.

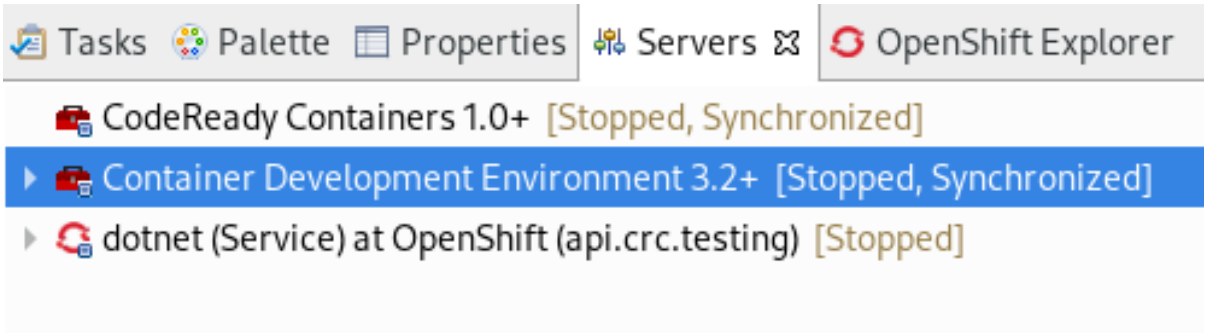
9. Click **Next**.
   The **Red Hat Container Development Environment**window appears.



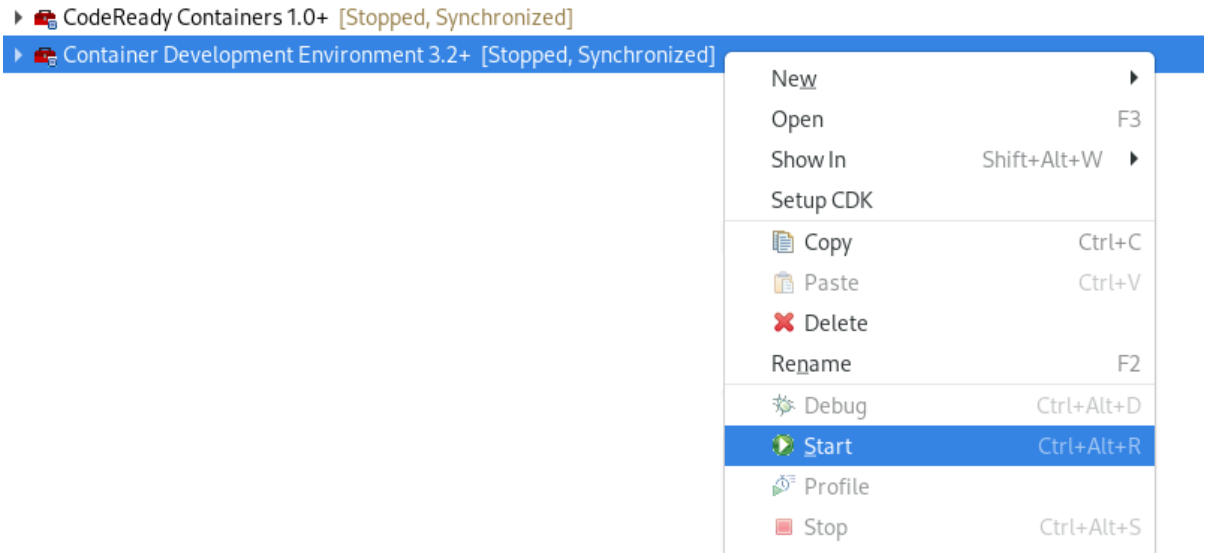
10. Click **Download and install runtime**.
    The **Download Runtimes** window appears.



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 tools

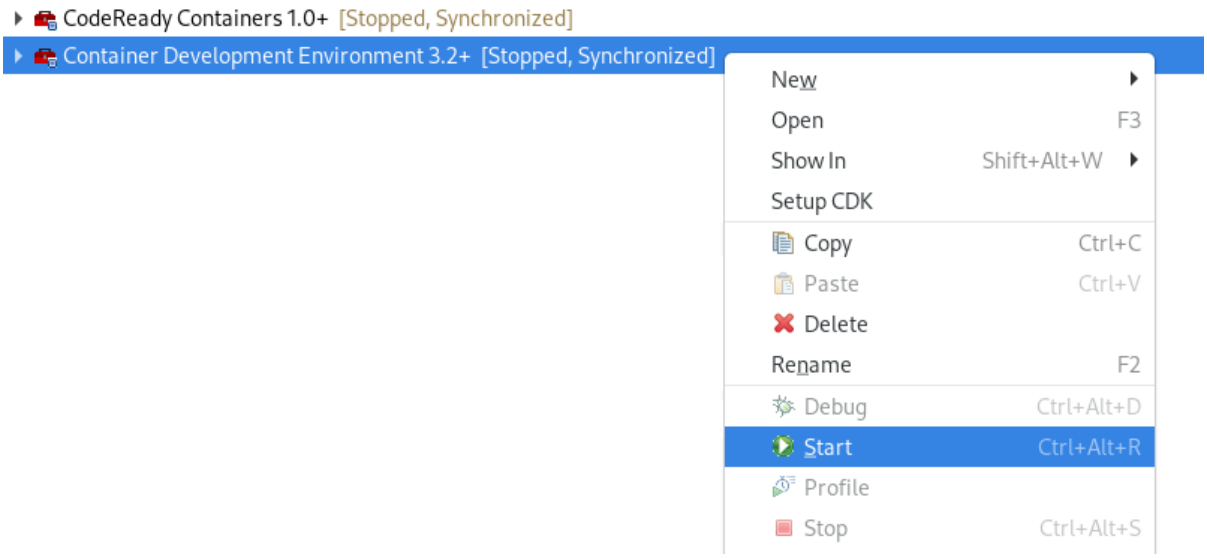
1.2.2.1. Creating a Dockerfile

Prerequisites

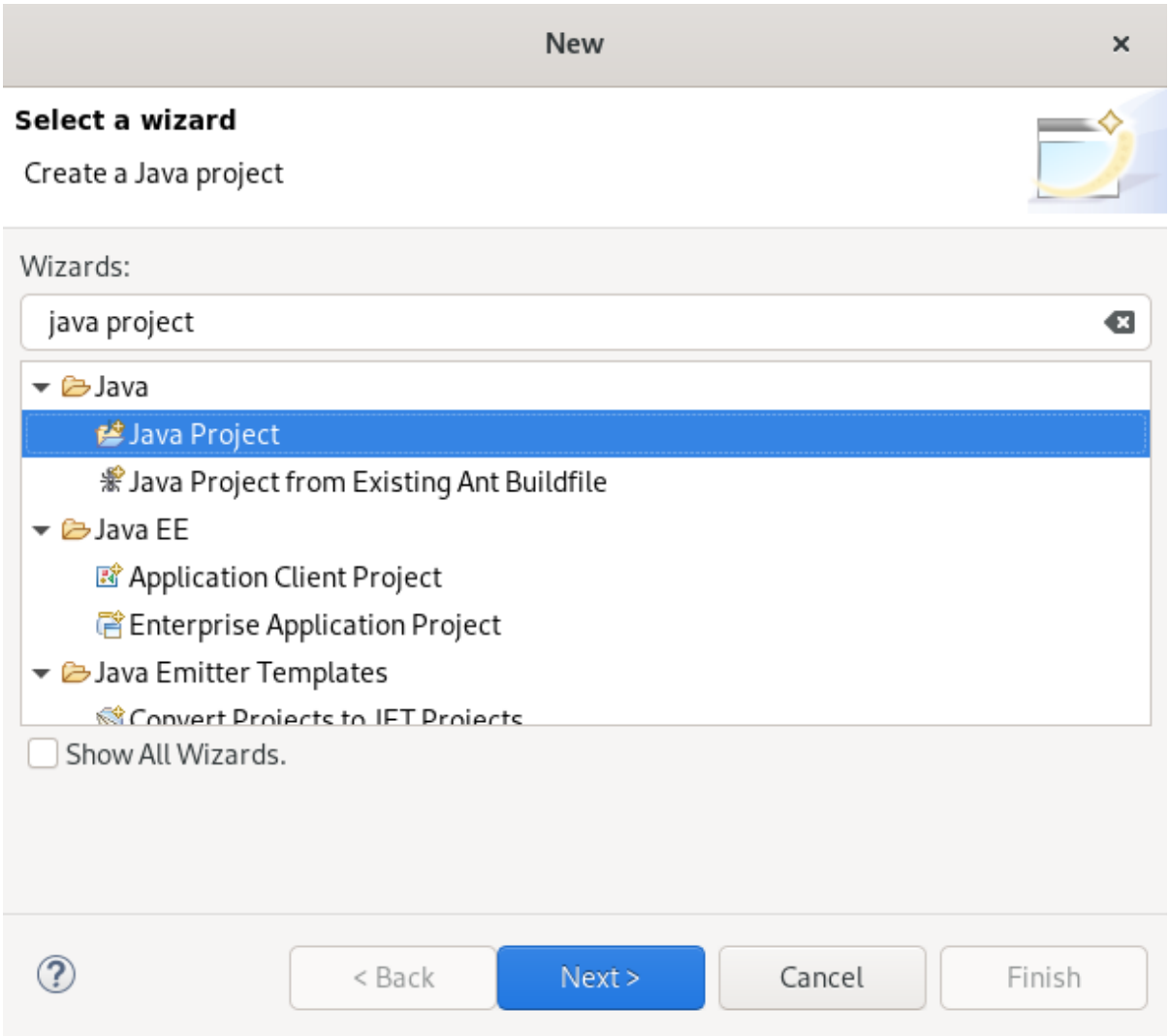
- The CDK server adapter is set up and configured.
  For more information, see Installing CDK.

Procedure

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



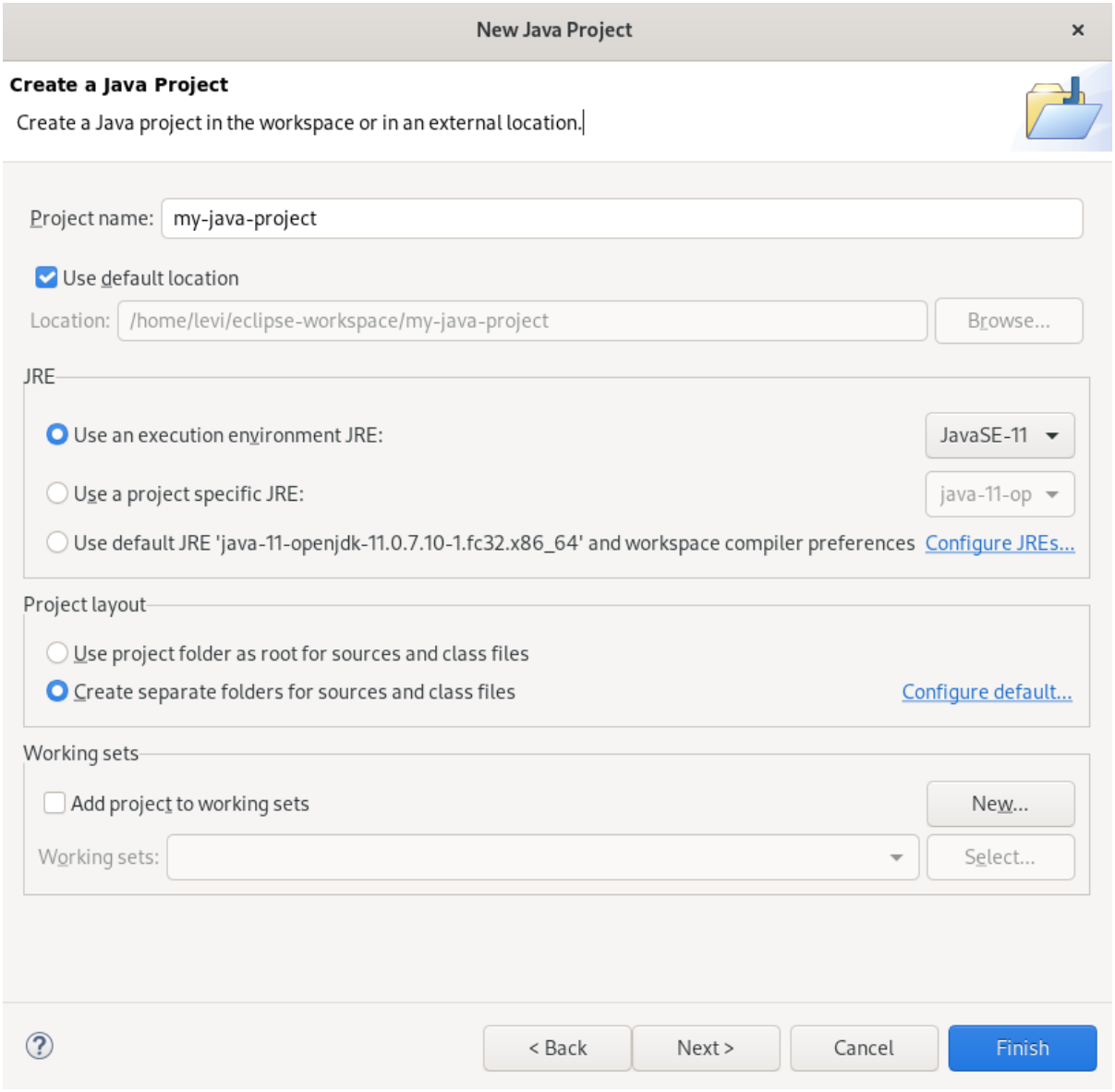
3. Press **Ctrl**+**N**.
The **Select a wizard** window appears.



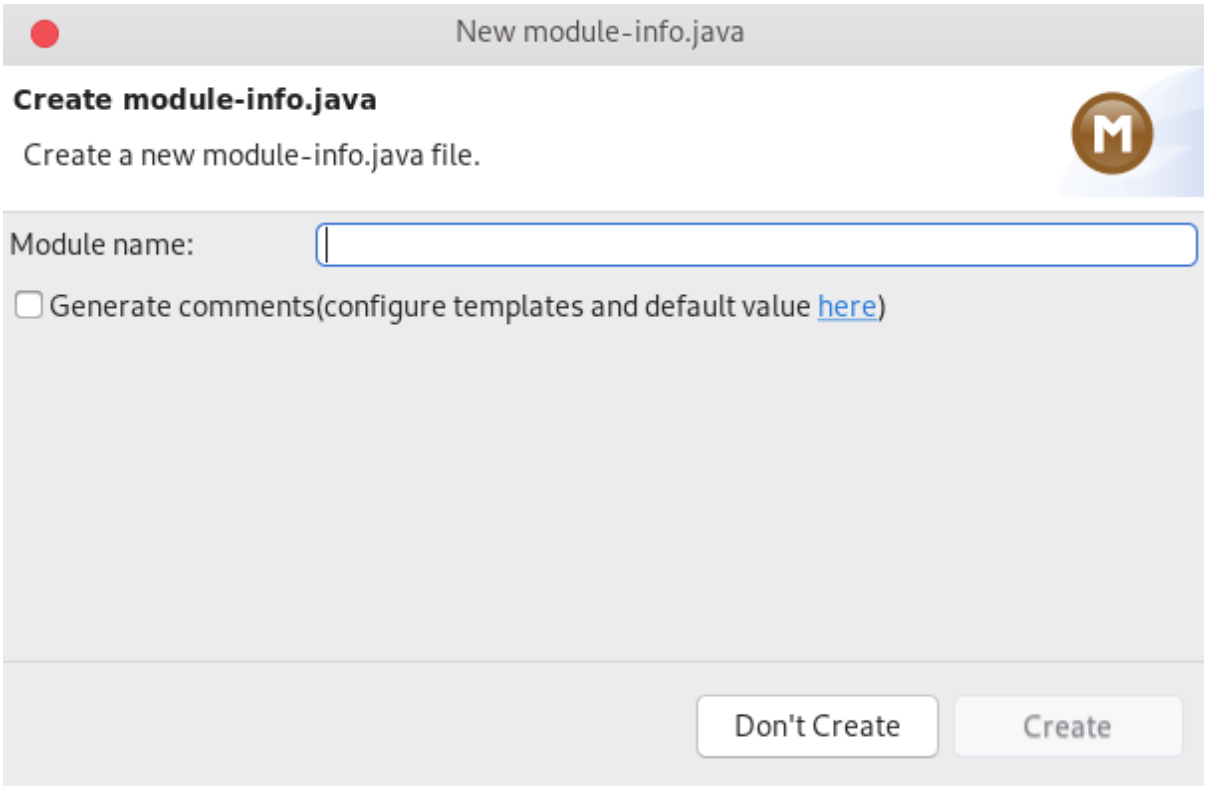
4. Enter **Java Project** in the search field.

5. Select **Java Project**.

6. Click **Next**.
   The **New Java Project** window appears.

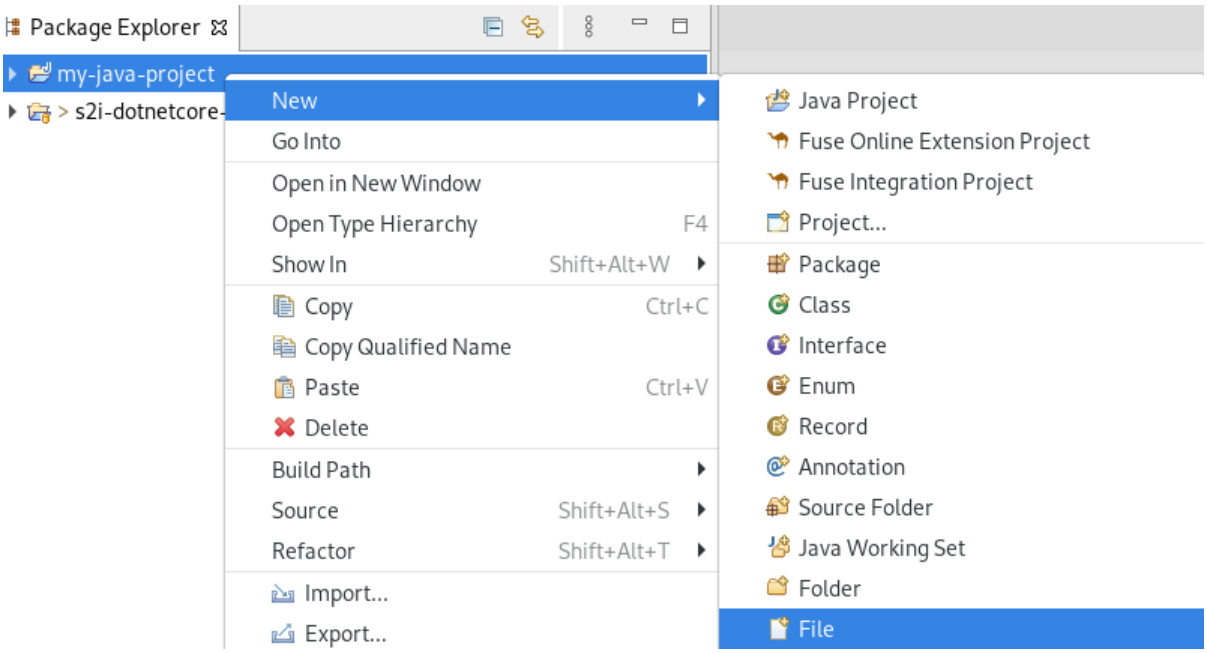


7. Name your project.
   The **New module-info.java** window appears.



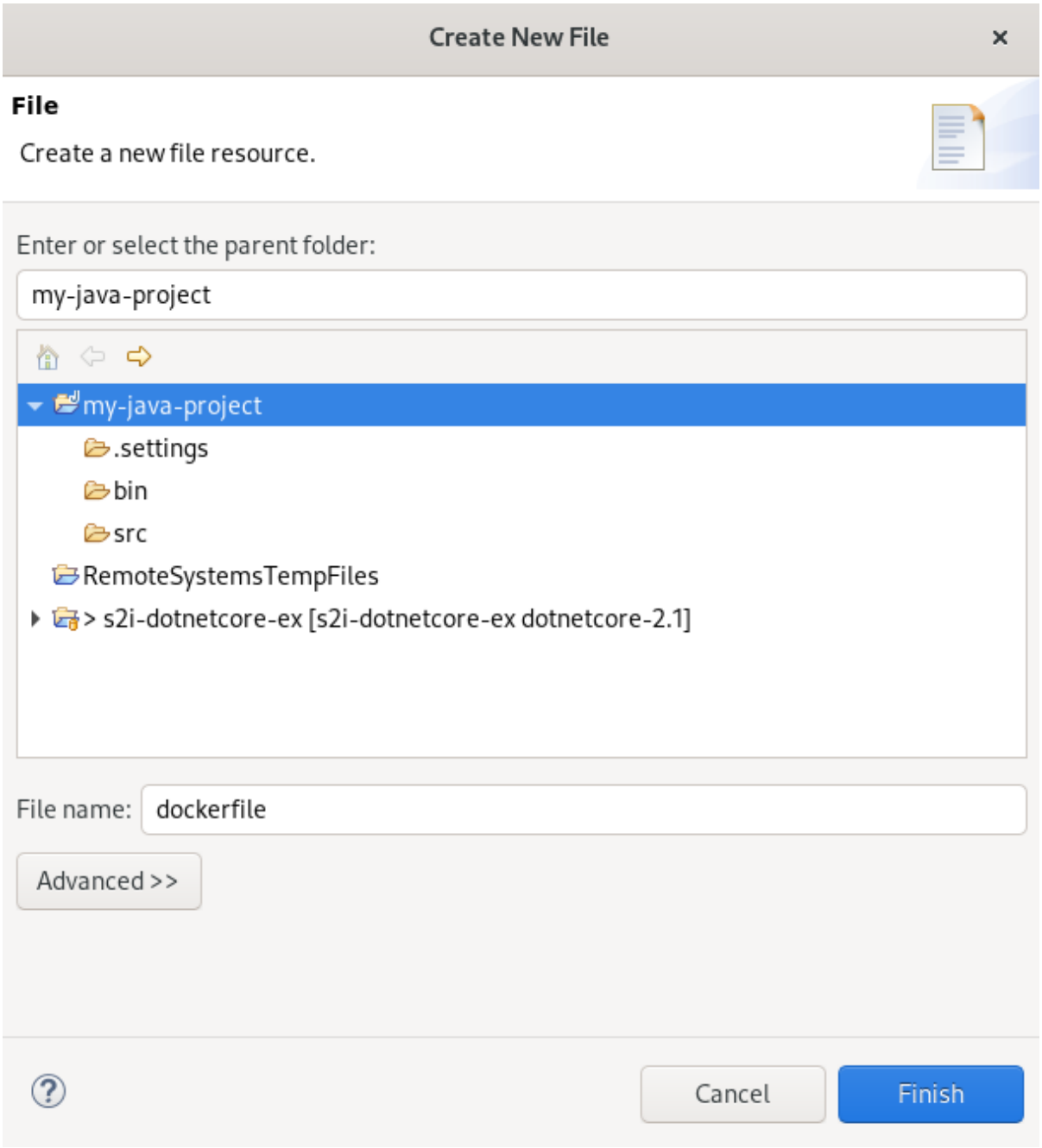
8. Create a new **module-info.java** file or click **Don't Create**.

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.

Your Dockerfile has been created.

Additional resources

- For more information about Dockerfiles, see Dockerfile reference.

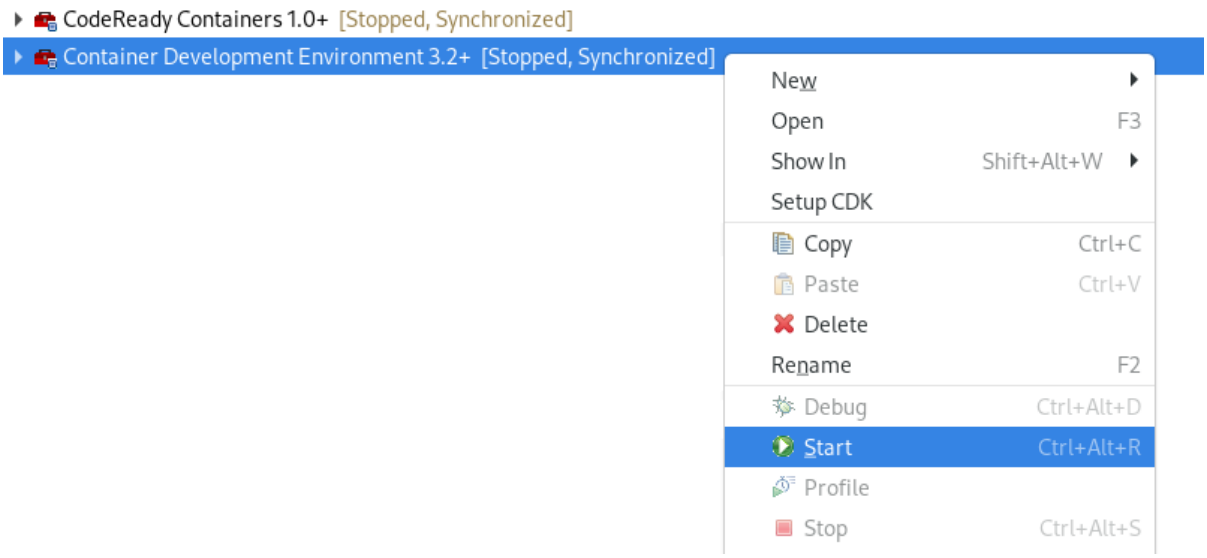
1.2.2.2. Building a Docker image Using Red Hat Container Development Kit

Prerequisites

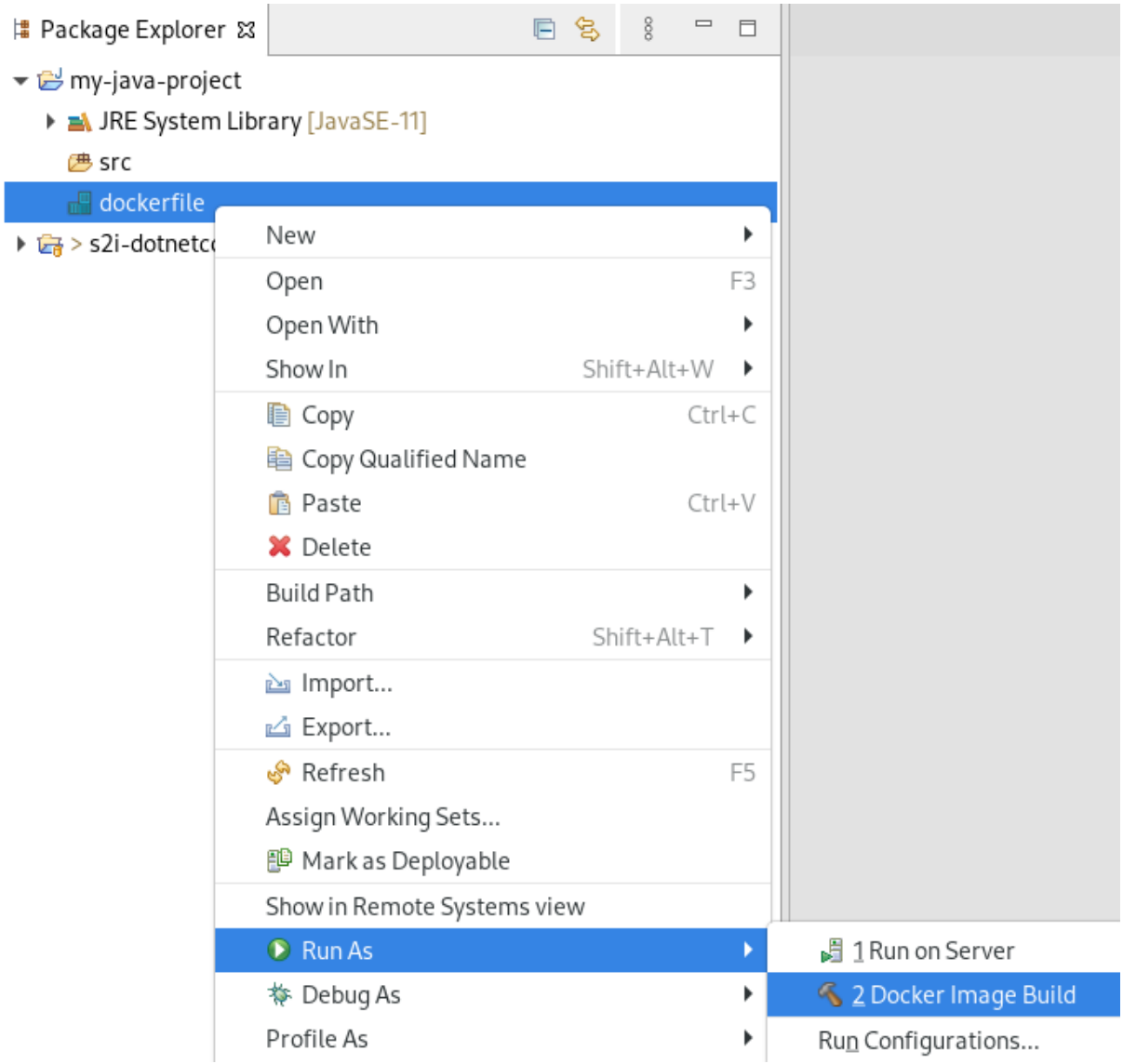
- The CDK server adapter is set up and configured.
  For more information, see Installing CDK.
- A Java project and a Dockerfile.
  For more information, see Creating a Dockerfile.

Procedure

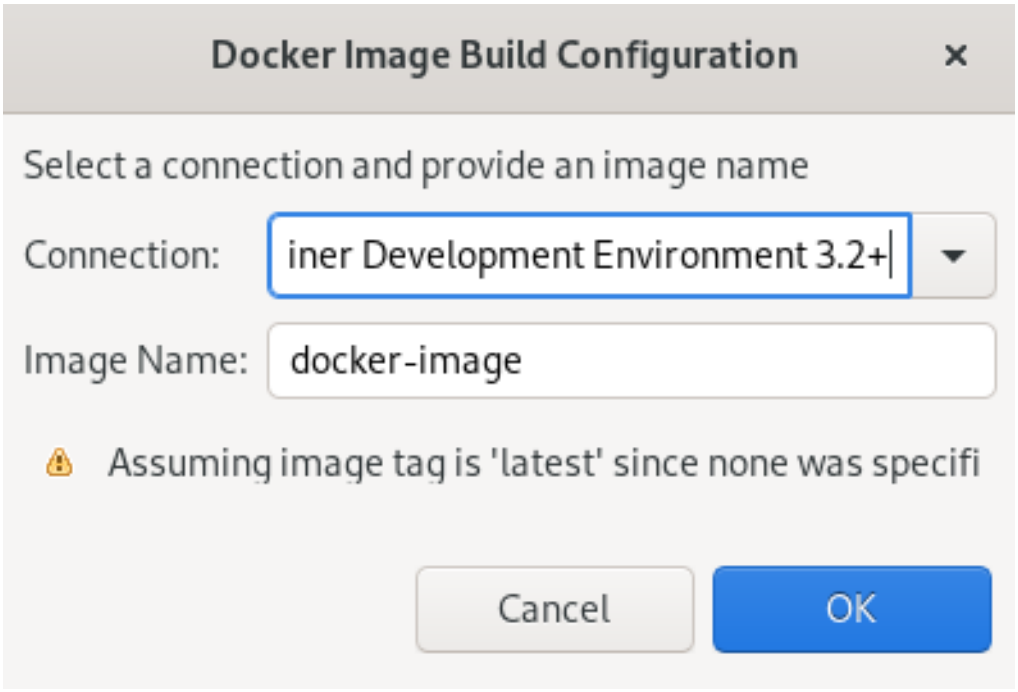
1. Start CodeReady Studio.
2. Start the CDK server adapter.



3. Expand your Java project.



4. Right-click **Dockerfile → Run as → Docker Image Build**.
The **Docker Image Build Configuration** window appears.



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.

1.2.3. Additional resources

- For more information on how to perform tasks using OpenShift Container Platform tools, see Developing for the Cloud with OpenShift in CodeReady Studio .
- For more information on how to use OpenShift in CodeReady Studio, see OpenShift basics in CodeReady Studio.

CHAPTER 2. DEVELOPING FOR THE CLOUD WITH OPENSHIFT IN CODEREADY STUDIO

2.1. CREATING AN OPENSHIFT CONTAINER PLATFORM APPLICATION IN CODEREADY STUDIO

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

2.1.1. Creating a new OpenShift Container Platform connection

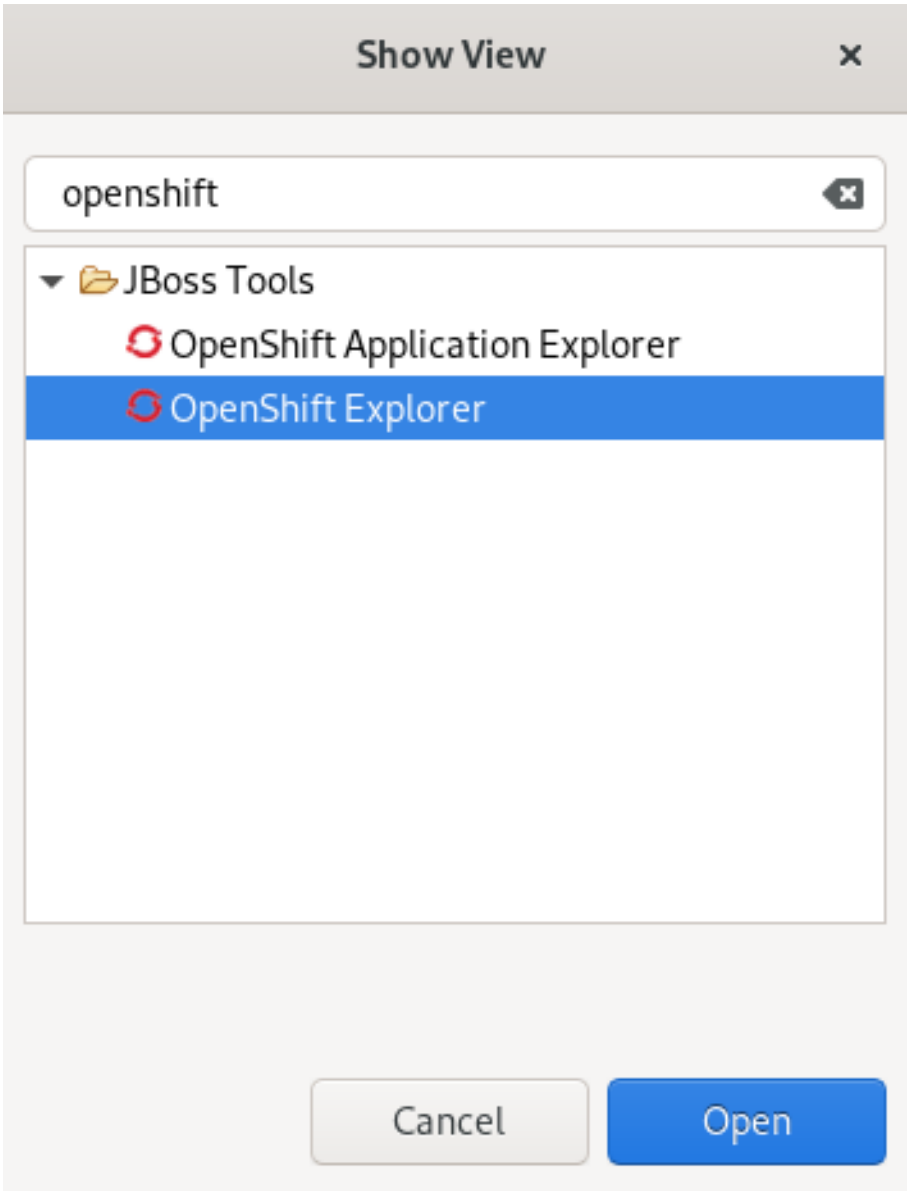
To use OpenShift tools in CodeReady Studio, you must create an OpenShift connection in the **OpenShift Explorer** view. An OpenShift connection connects CodeReady Studio to an OpenShift instance (based on CDK, OpenShift Online, Kubernetes or minishift). The connection is listed in the **OpenShift Explorer**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

Prerequisites

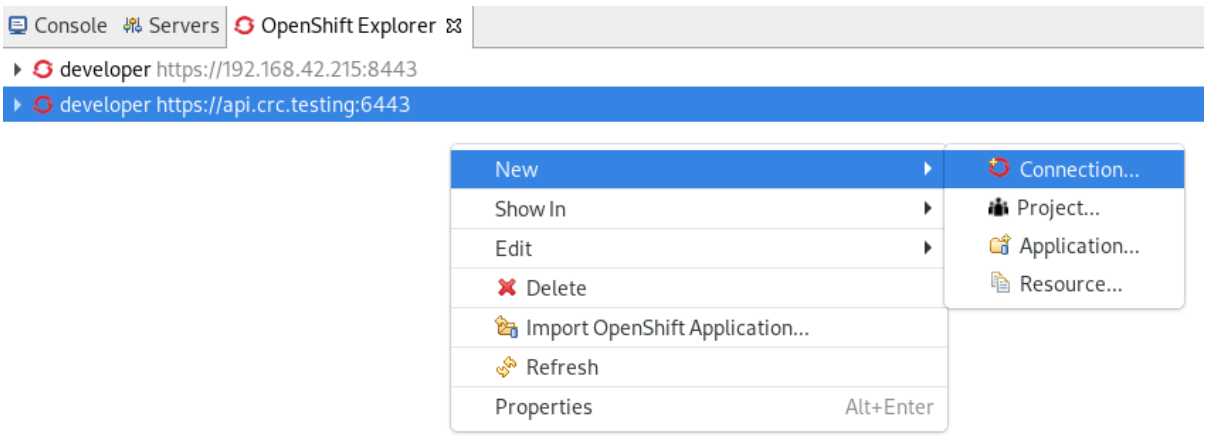
1. The CodeReady Containers server adapter is running.

Procedure

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

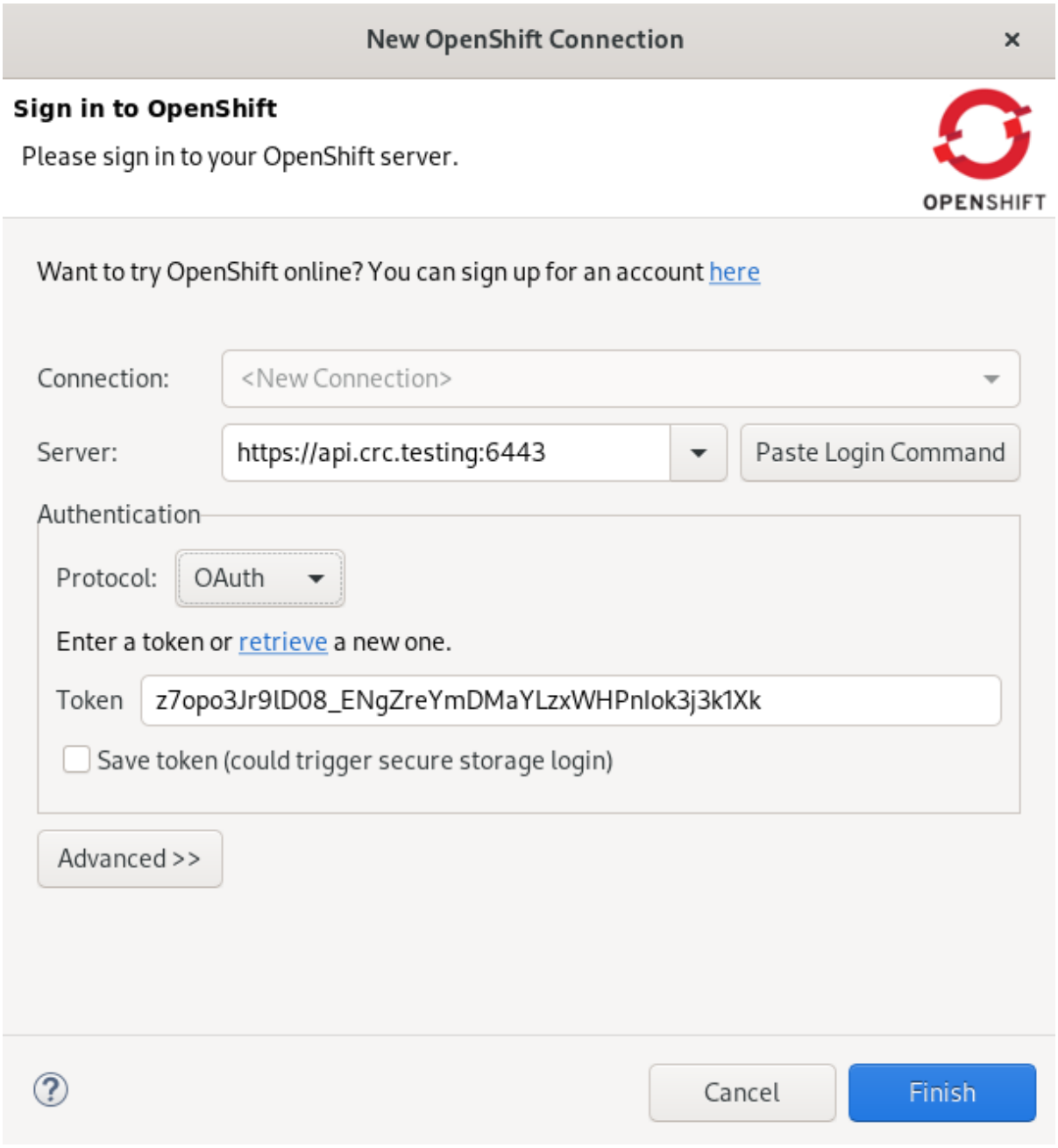


3. Enter **OpenShift** in the search field.
4. Select **OpenShift Explorer**.
5. Click **Open**.
   The **OpenShift Explorer** view appears.



6. Right-click any area in the **OpenShift Explorer**.

7. Click **New → Connection**.
   The **Sign in to OpenShift** window appears.



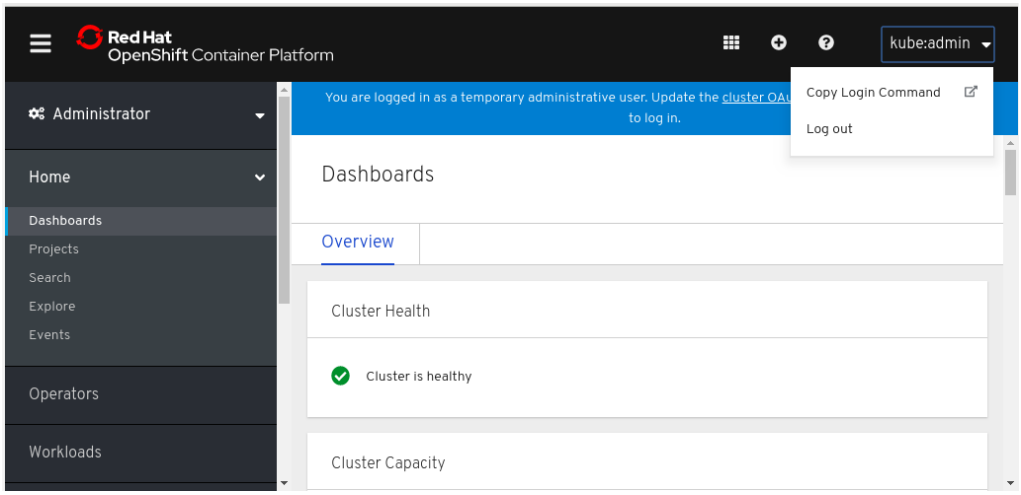
8. Paste the URL of your OpenShift server into the **Server** field.

9. Authenticate with a 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

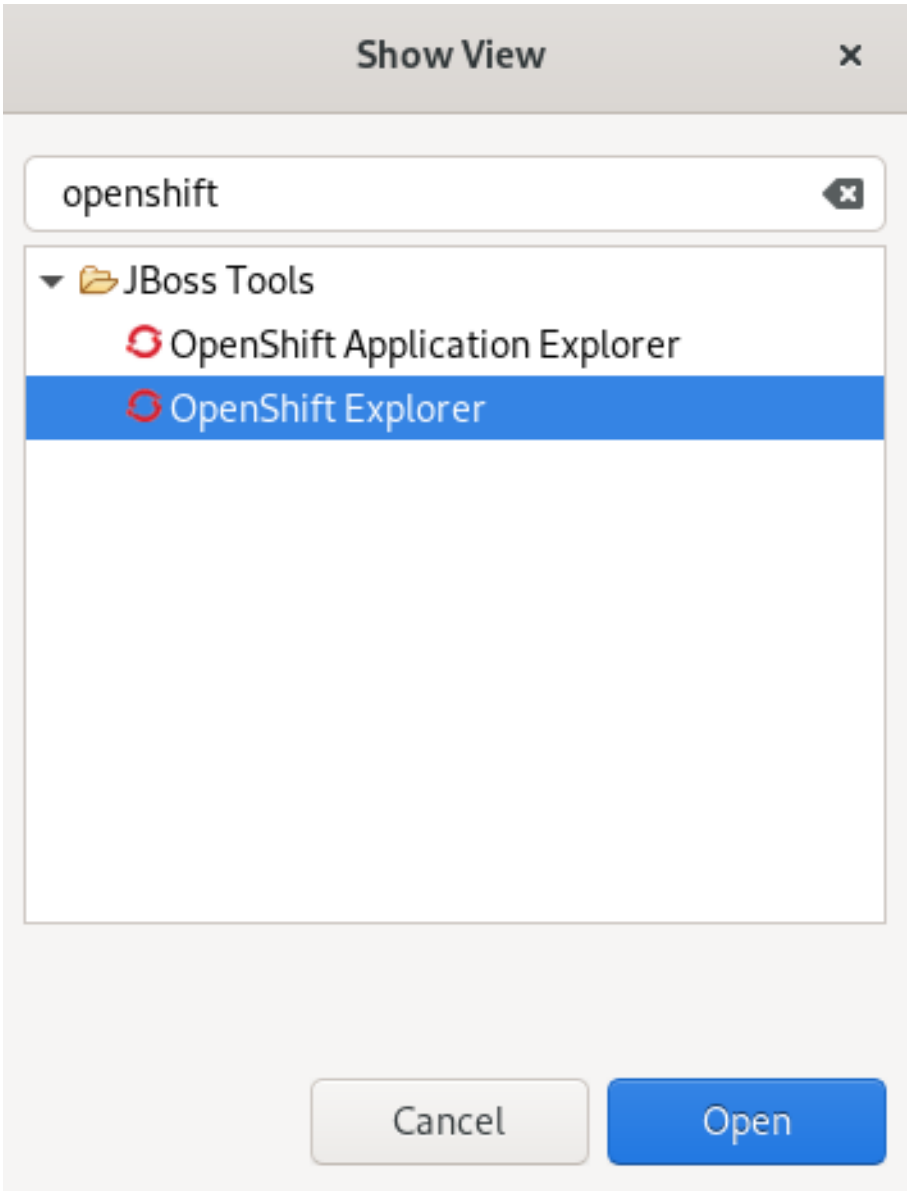
You must create an OpenShift Container Platform project, which essentially is a namespace with additional annotations, to centrally manage the access to resources for regular users of your OpenShift Container Platform.

Prerequisites

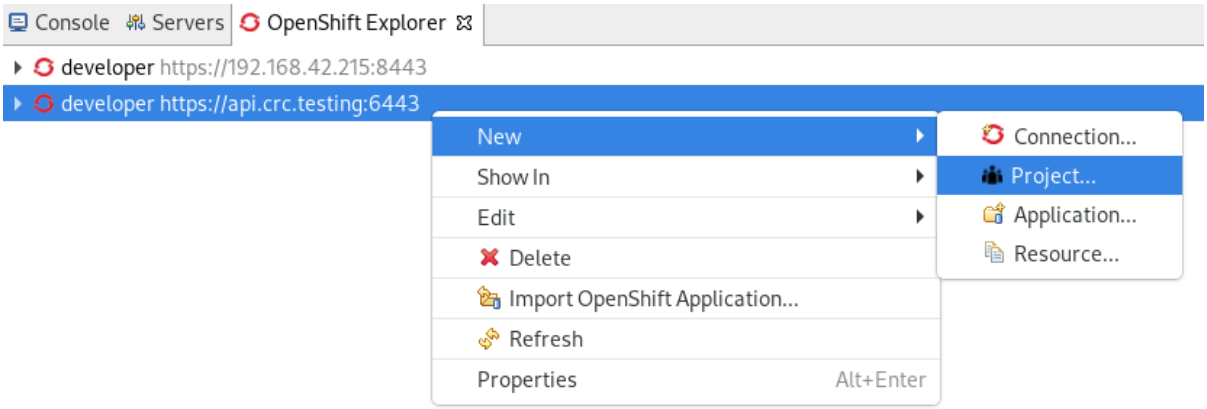
- An OpenShift Container Platform connection.
  For more information on how to create a new OpenShift Container Platform connection, see Creating a new OpenShift Container Platform connection .

Procedure

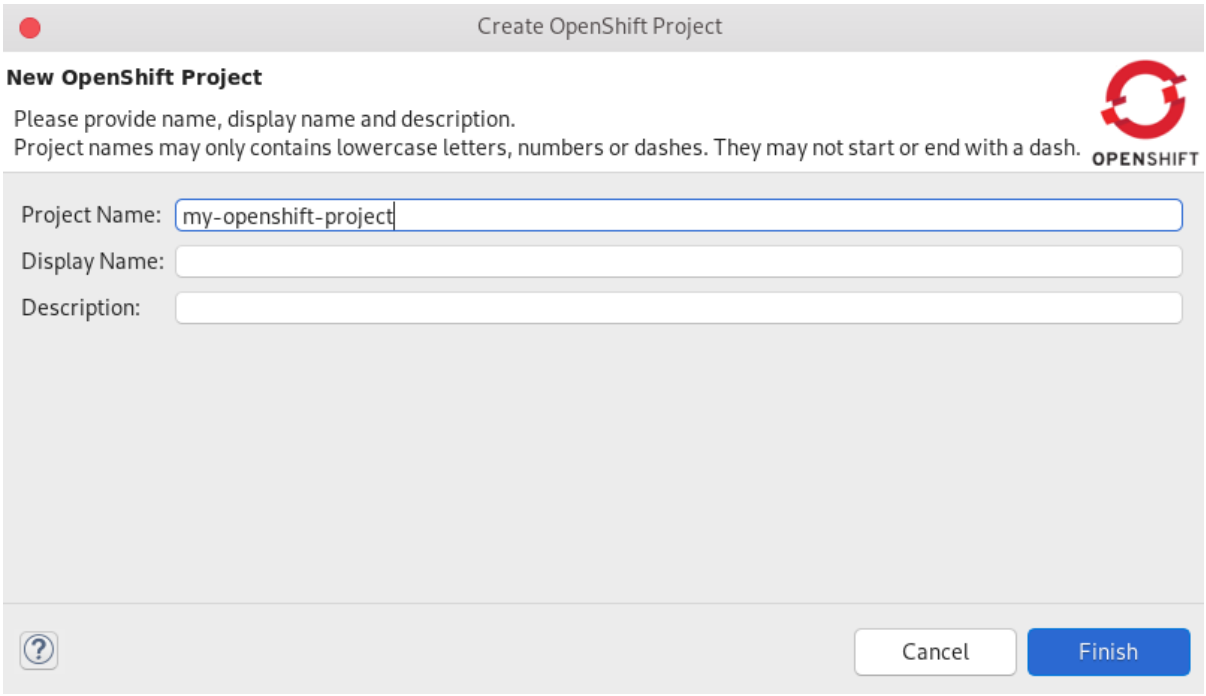
1. Start CodeReady Studio.
2. Click **Window → Show View → Other**.
   The **Show View** window appears.



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

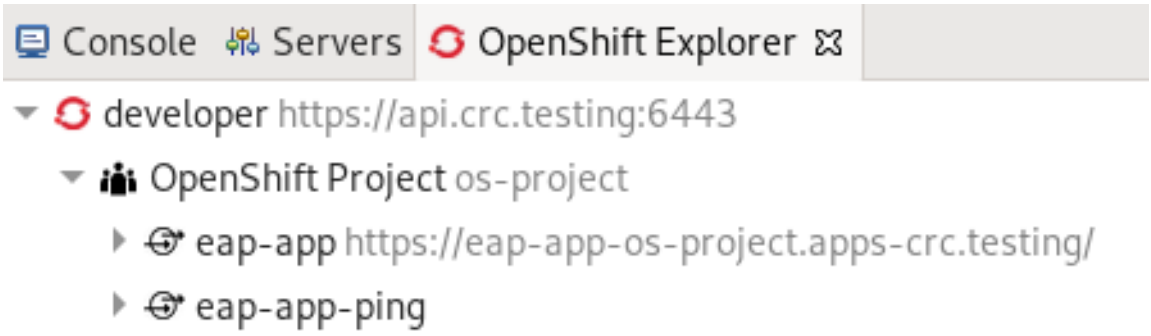


6. Right-click the **OpenShift Container Platform connection → New → Project**.
   The **New OpenShift Project** window appears.



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

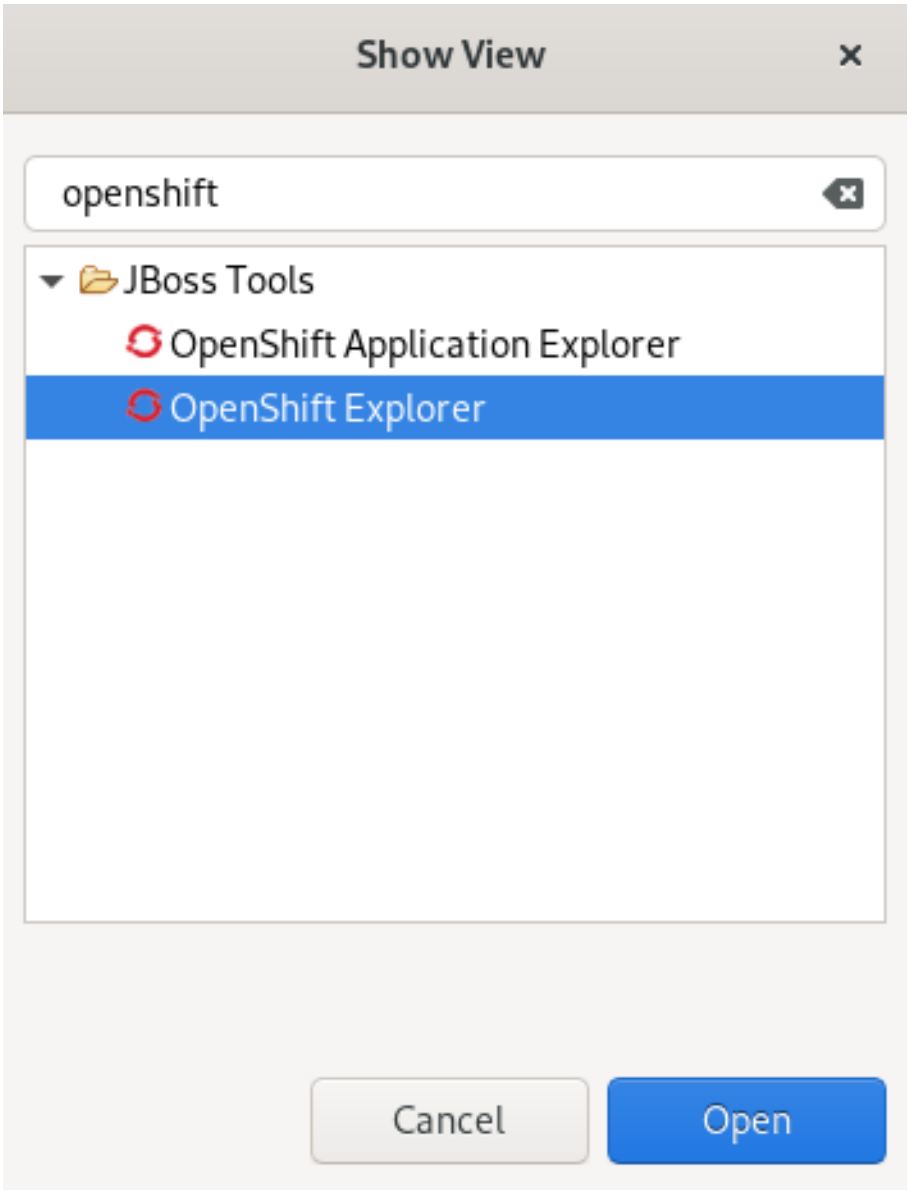
You can use the **OpenShift Application** wizard in CodeReady Studio to create OpenShift Container Platform applications from default or custom templates.

Prerequisites

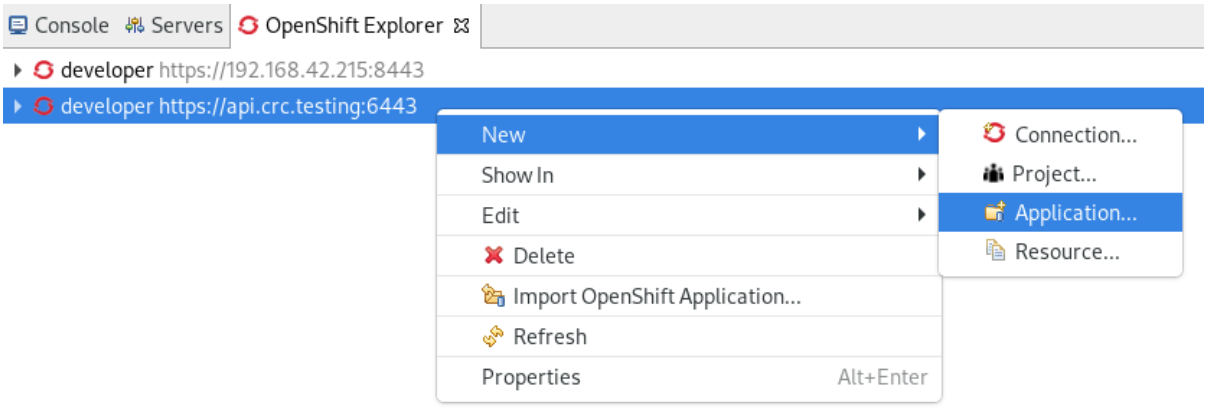
- An OpenShift Container Platform connection.
  For more information on how to create a new OpenShift Container Platform connection, see Creating a new OpenShift Container Platform connection .
- An OpenShift Container Platform project.
  For more information on how to create a new OpenShift Container Platform project, see Creating a new OpenShift Container Platform project .

Procedure

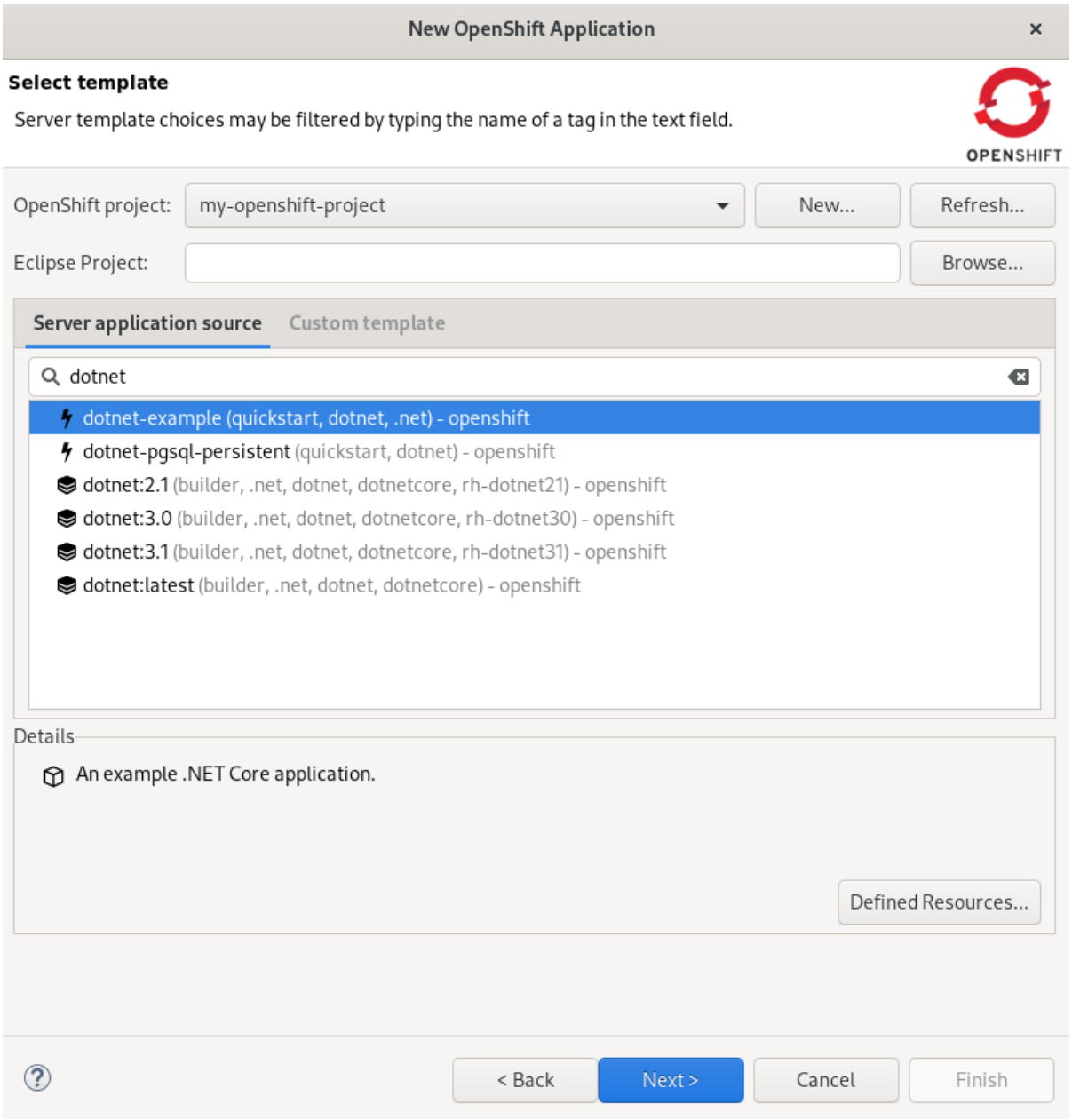
1. Start CodeReady Studio.
2. Click **Window → Show View → Other**.
   The **Show View** window appears.



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

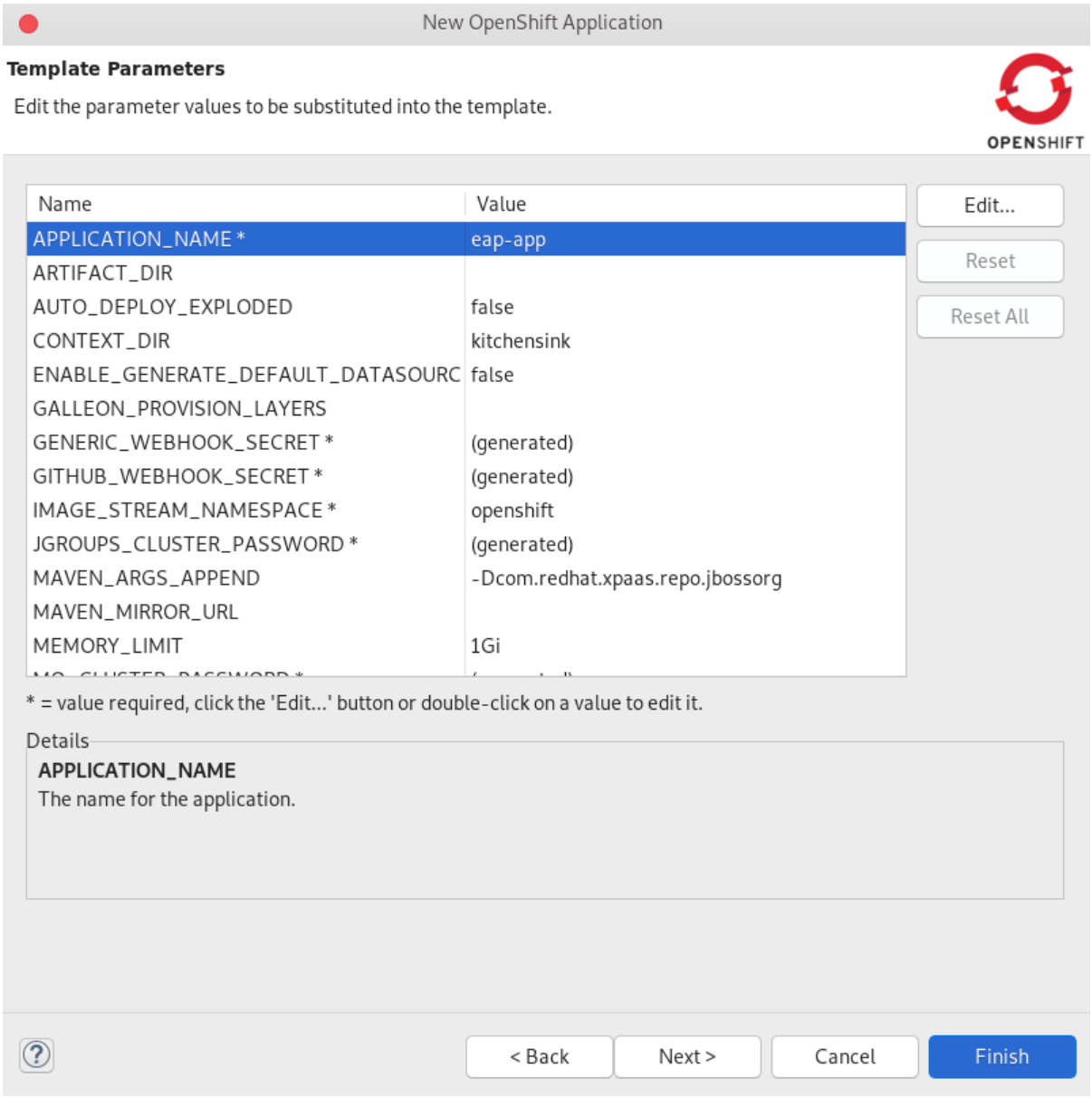


6. Right-click the **OpenShift Container Platform connection → New → Application**.
The **Select template** window appears.



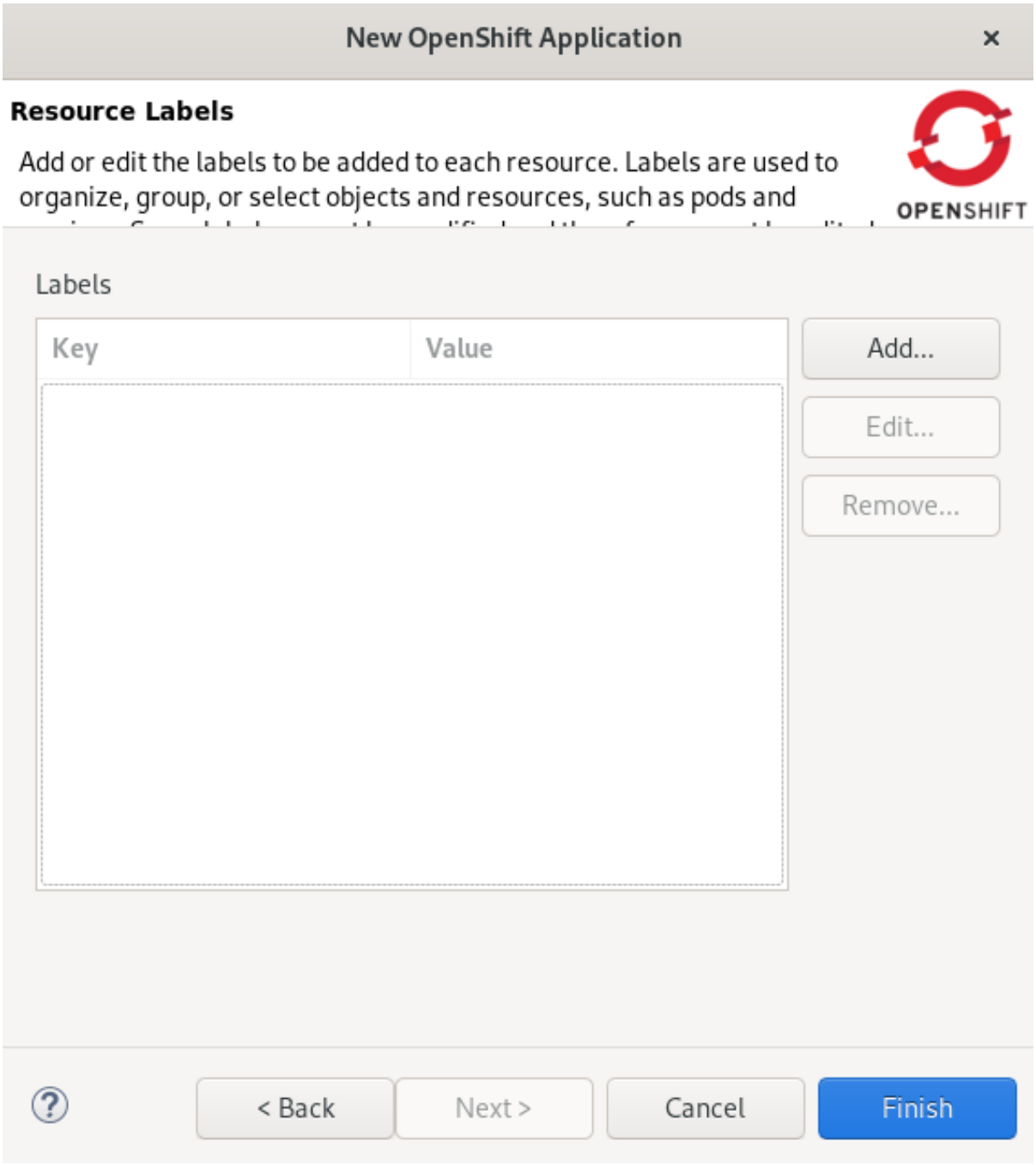
7. Select a template.

8. Click **Next**.
The **Template Parameters** window appears.



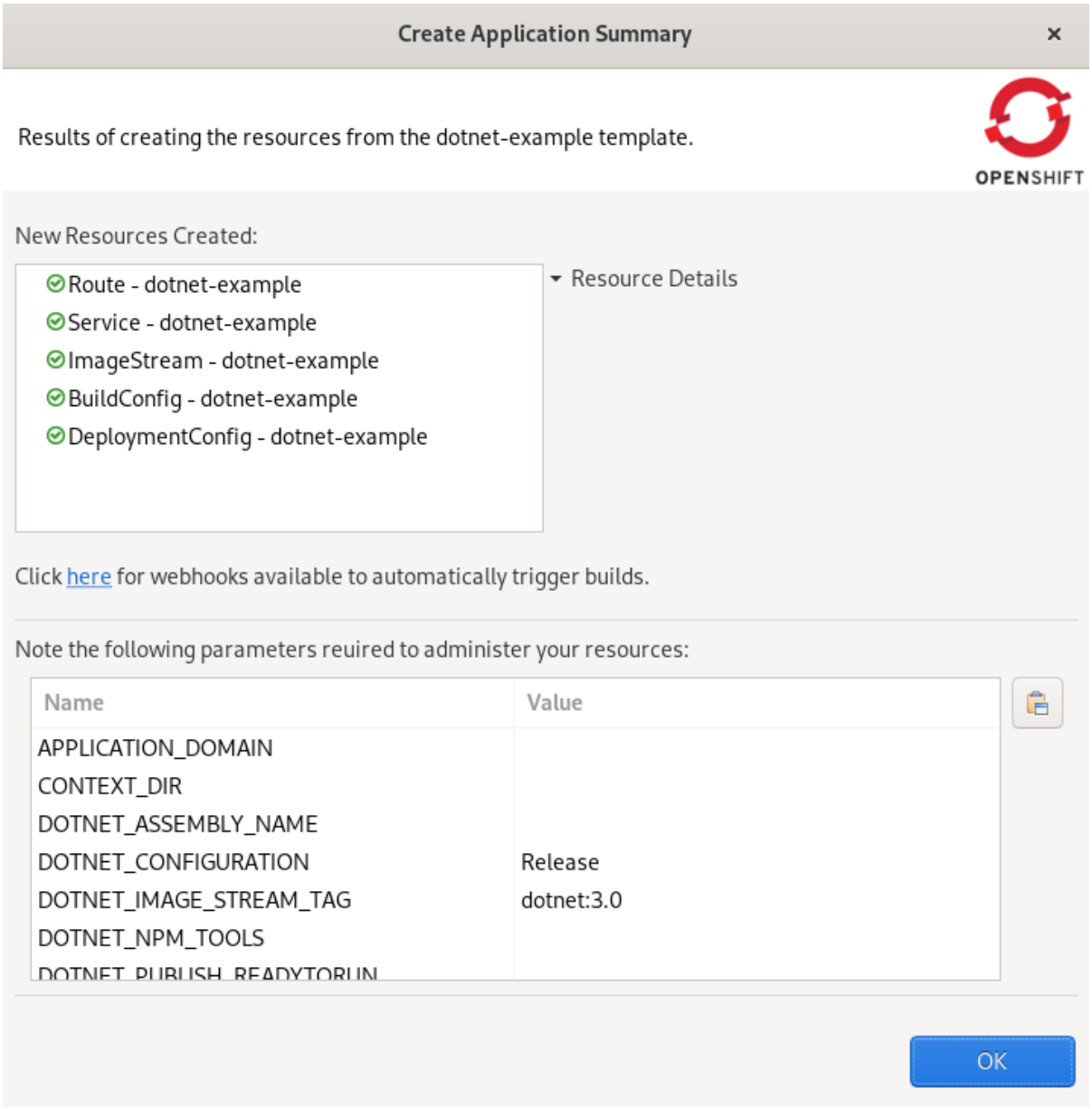
9. Ensure that the template parameters are correct.

10. Click **Next**.
    The **Resource Labels** window appears.



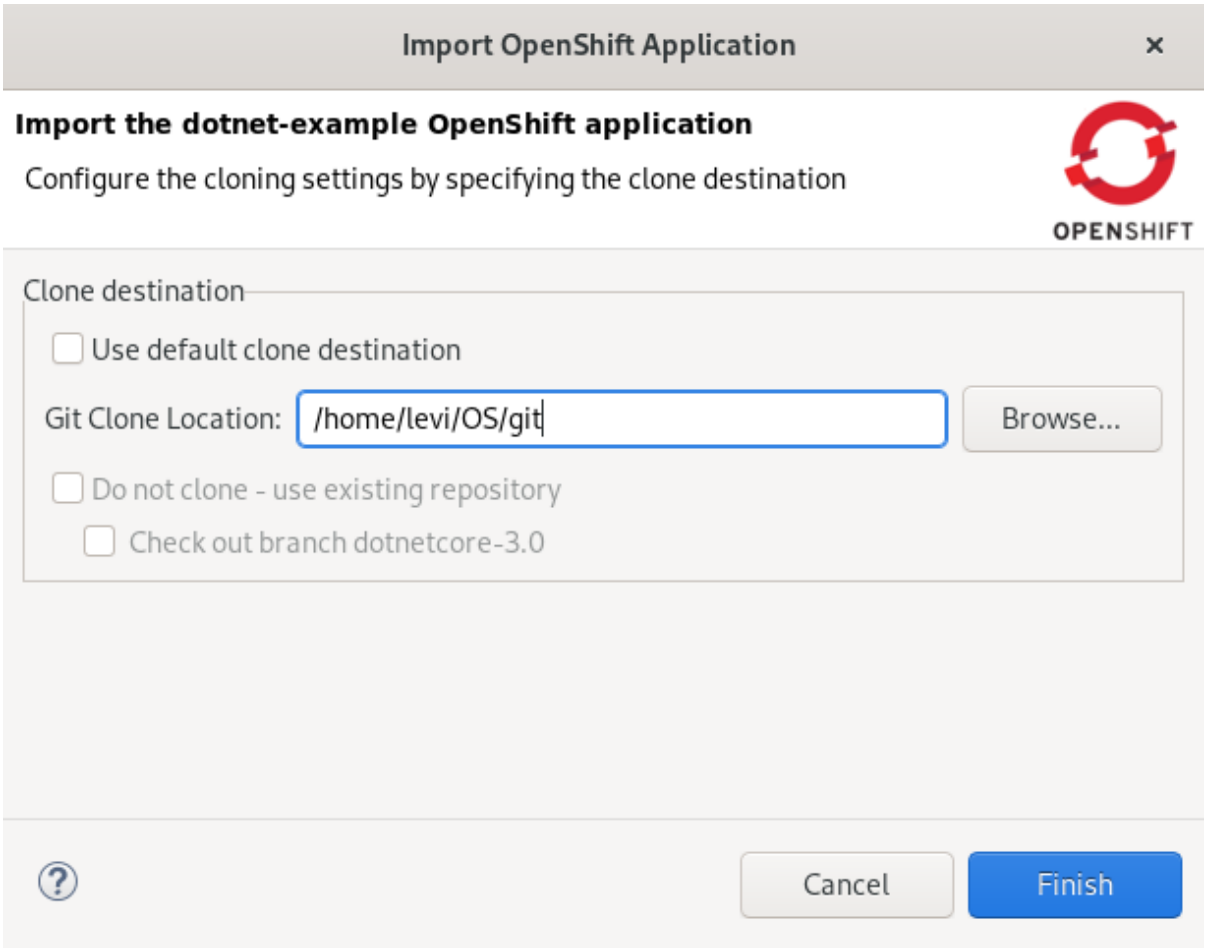
11. Click **Add** to add labels.

12. Click **Finish**.
The **Create Application Summary** window appears.



13. Ensure that the application details are correct.

14. Click **OK**.
    The **Import OpenShift Application** window appears.



15. Choose the location for your git repository clone.
16. 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 the upstream documentation Official OKD documentation, Using templates .

2.1.4. Importing an existing OpenShift Container Platform application into CodeReady Studio

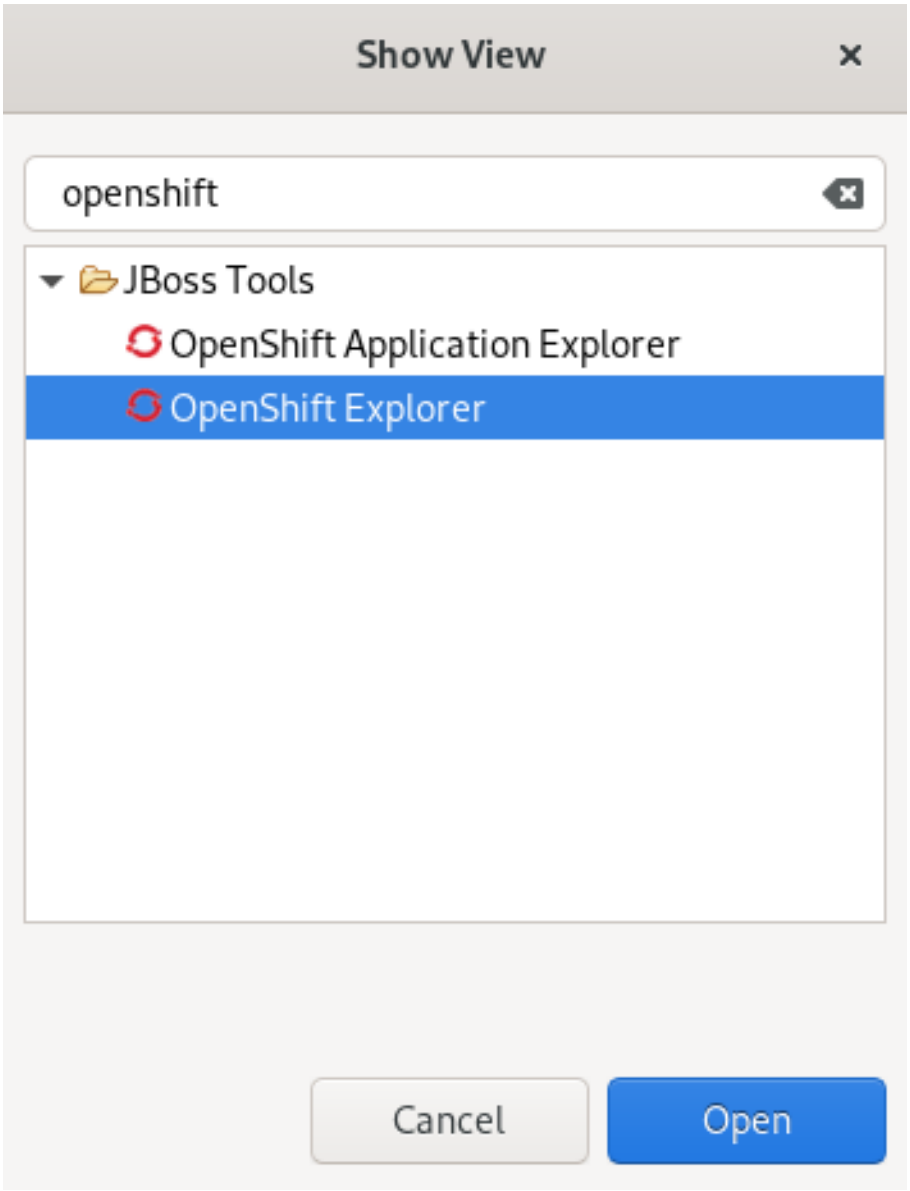
The **OpenShift Explorer** view in CodeReady Studio lists applications associated with your OpenShift Container Platform accounts. You can import the source code for these applications individually into CodeReady Studio using the **Import OpenShift Application** wizard. After the application is imported,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

Prerequisites

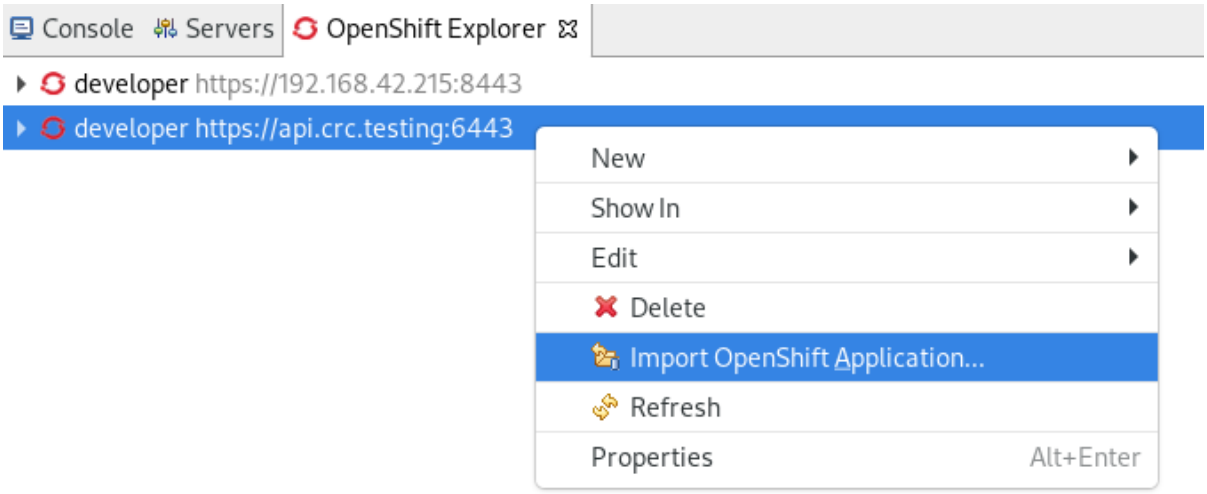
- The application that you are importing into CodeReady Studio has its source specified in the **build config** file.
- An OpenShift Container Platform connection.
  For more information on how to create an OpenShift Container Platform connection, see Creating a new OpenShift Container Platform connection .

Procedure

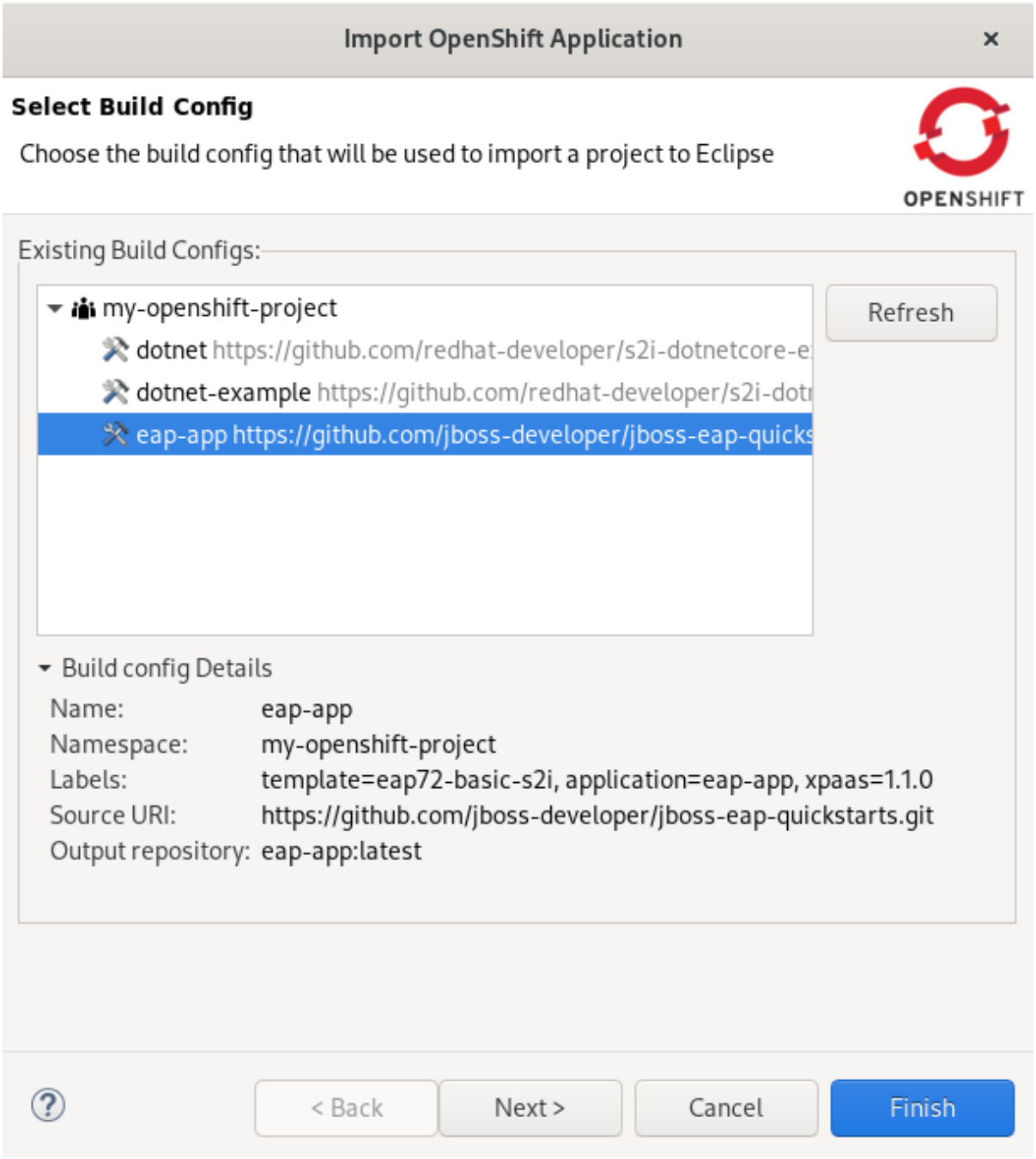
1. Start CodeReady Studio.
2. Click **Window → Show View → Other**.
   The **Show View** window appears.



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

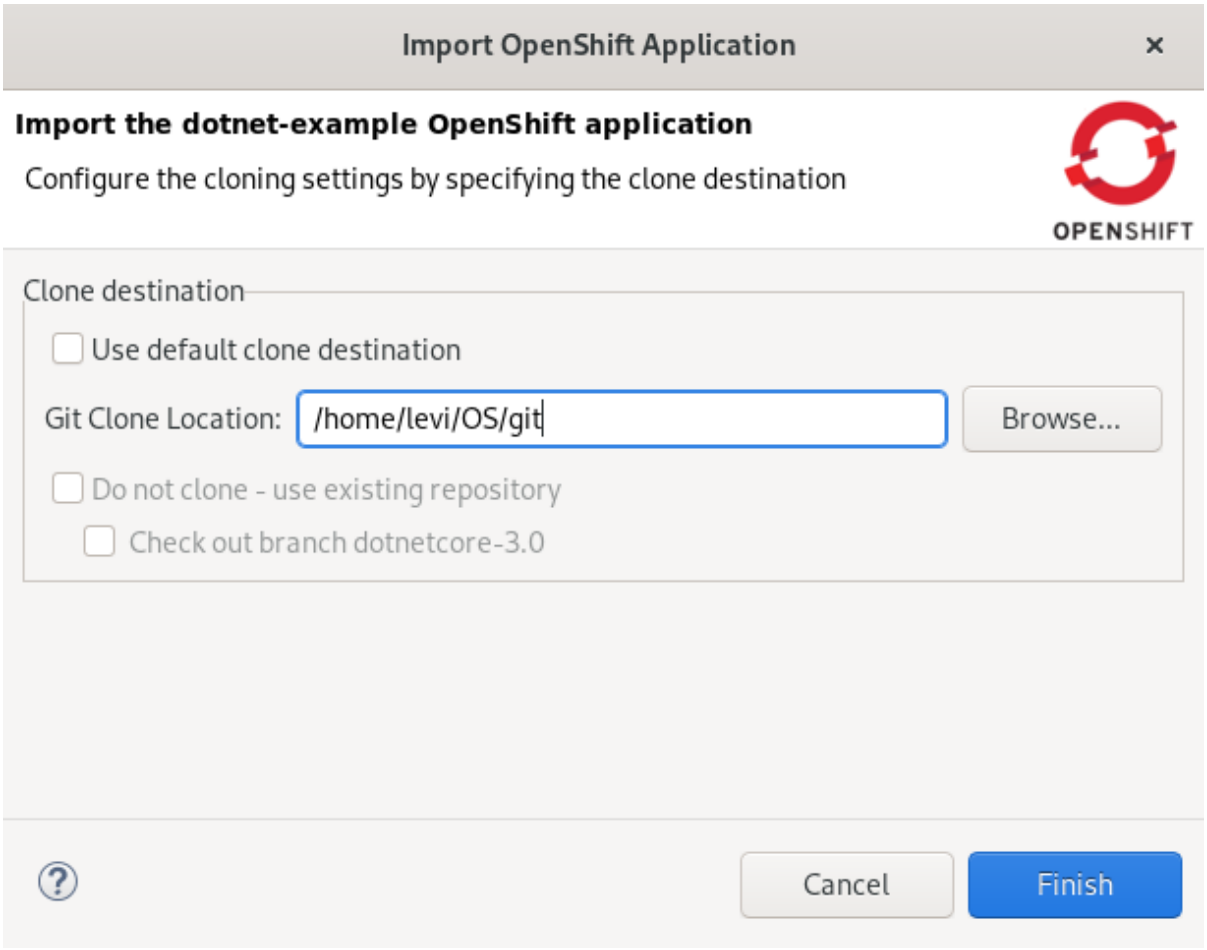


6. Right-click your **OpenShift Container Platform connection** → **Import OpenShift Application**. The **Select Build Config** window appears.



7. Select the application you want to import.

8. Click **Next**.
   The **Import OpenShift Application** window appears.



9. Select your **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

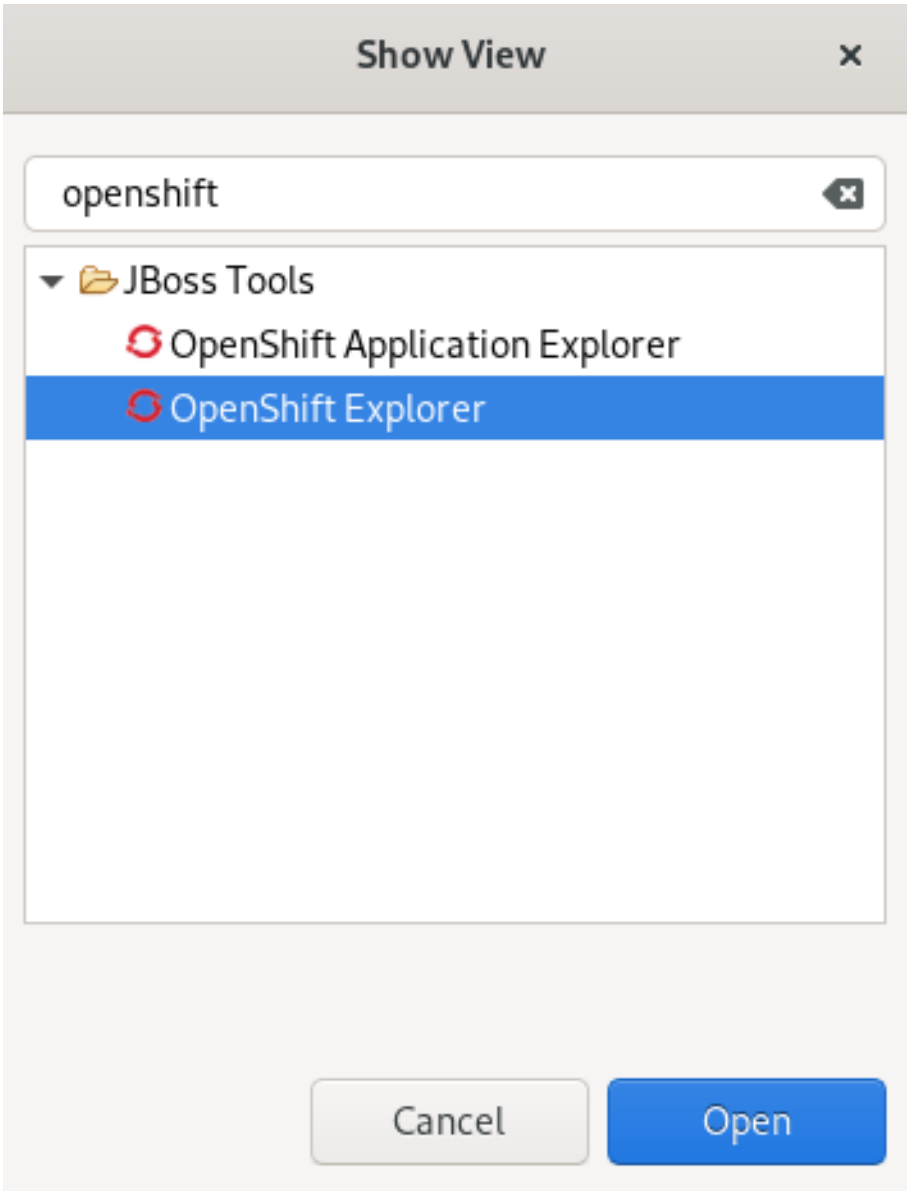
The server adapter enables you to publish the changes that you made in your workspace project to the running OpenShift application on the OpenShift instance. It enables incremental deployment of applications directly into the deployed pods on OpenShift. You can use the server adapter to push changes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

Prerequisites

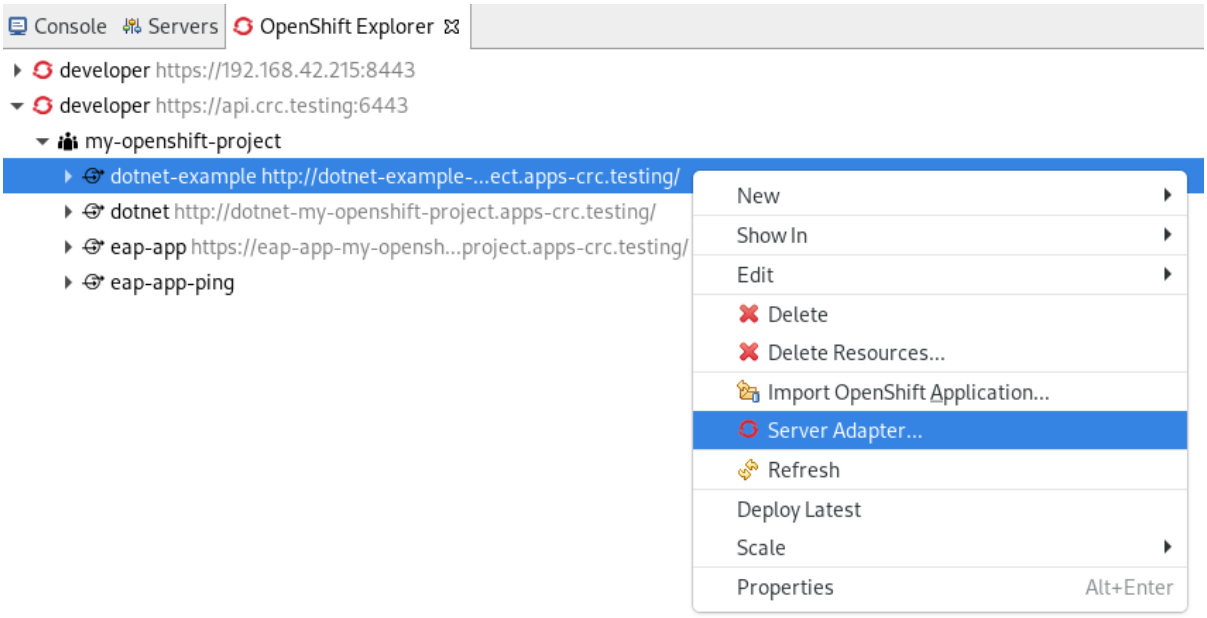
- An OpenShift Container Platform connection.
  For more information on how to create an OpenShift Container Platform connection, see Creating a new OpenShift Container Platform connection .

Procedure

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

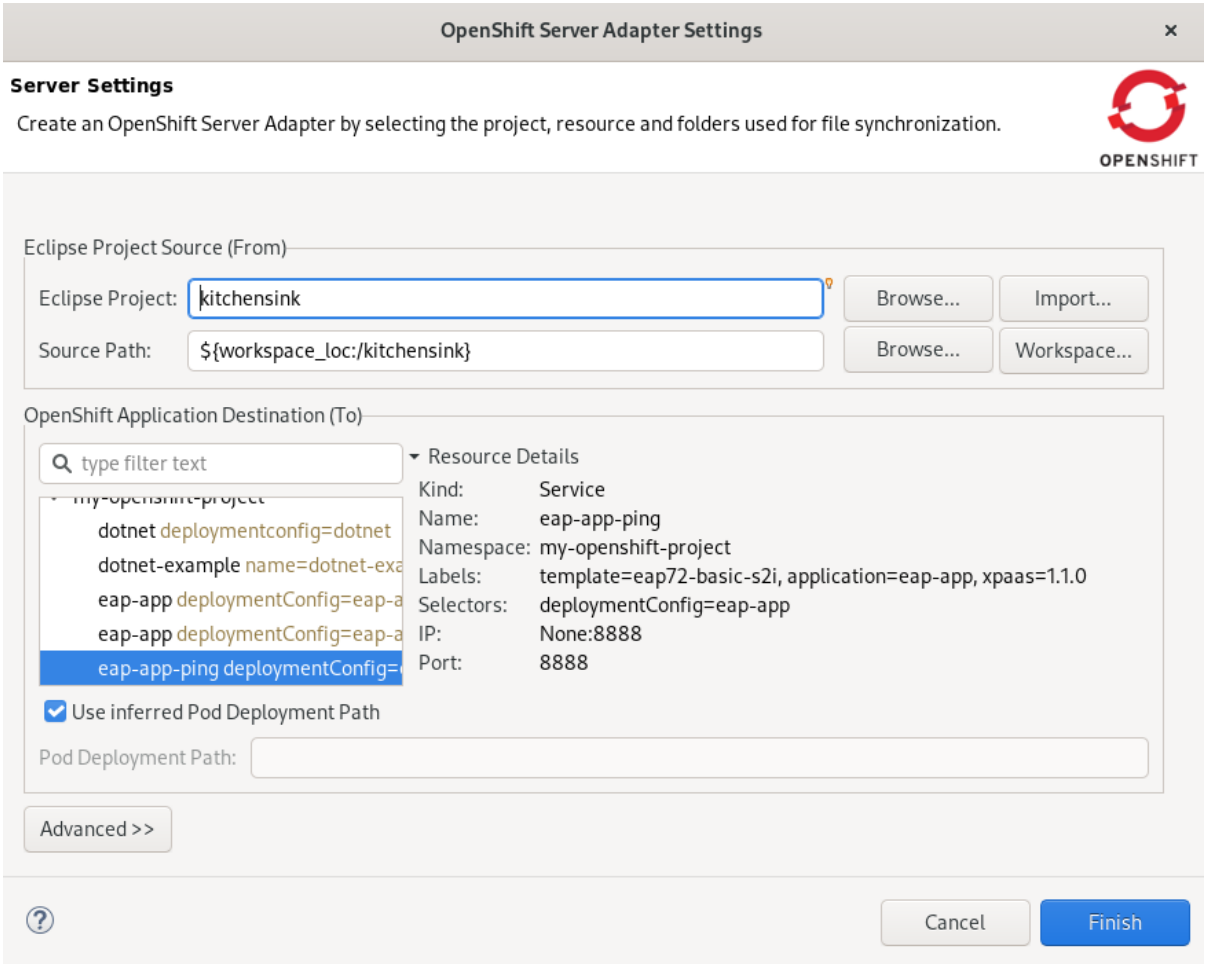


3. Enter **OpenShift** in the search field.
4. Select **OpenShift Explorer**.
5. Click **Open**.
   The **OpenShift Explorer** view appears.



6. Expand the OpenShift Container Platform connection.

7. Right-click your **application → Server Adapter**.
The **Server Settings** window appears.



NOTE

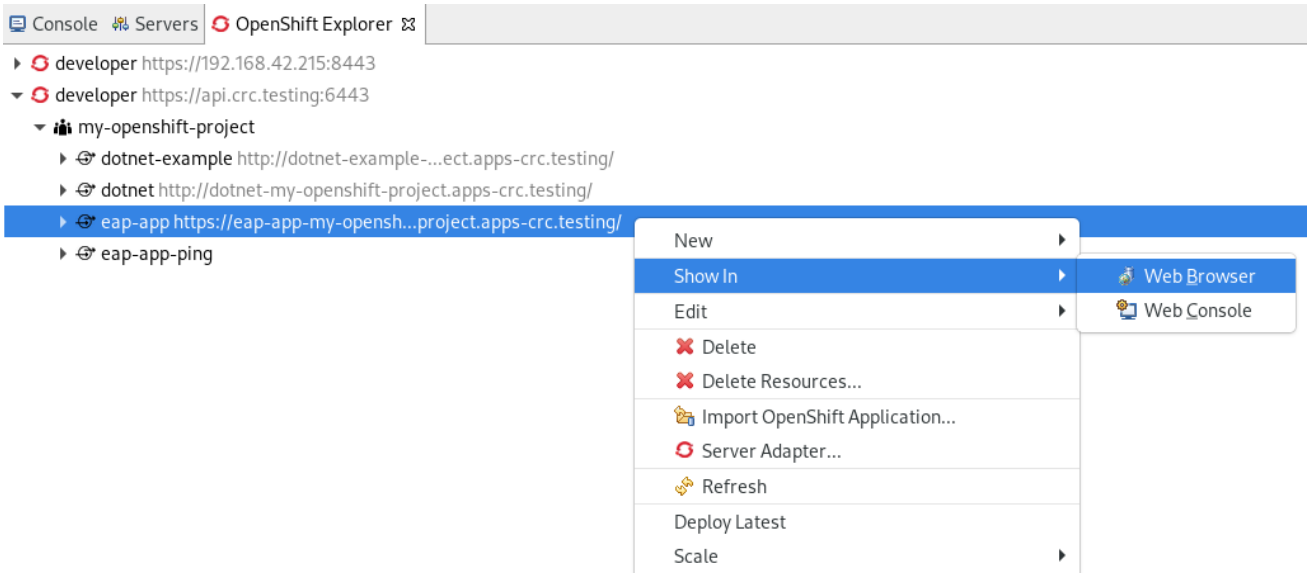
If you are using EAP 7.3, you need to set the path for the deployment of your server adapter due to changes in the templates.

To do so, uncheck the **Use inferred Pod Deployment Path** checkbox and set the Pod Deployment Path field to **/opt/eap/standalone/deployments/**.

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**.



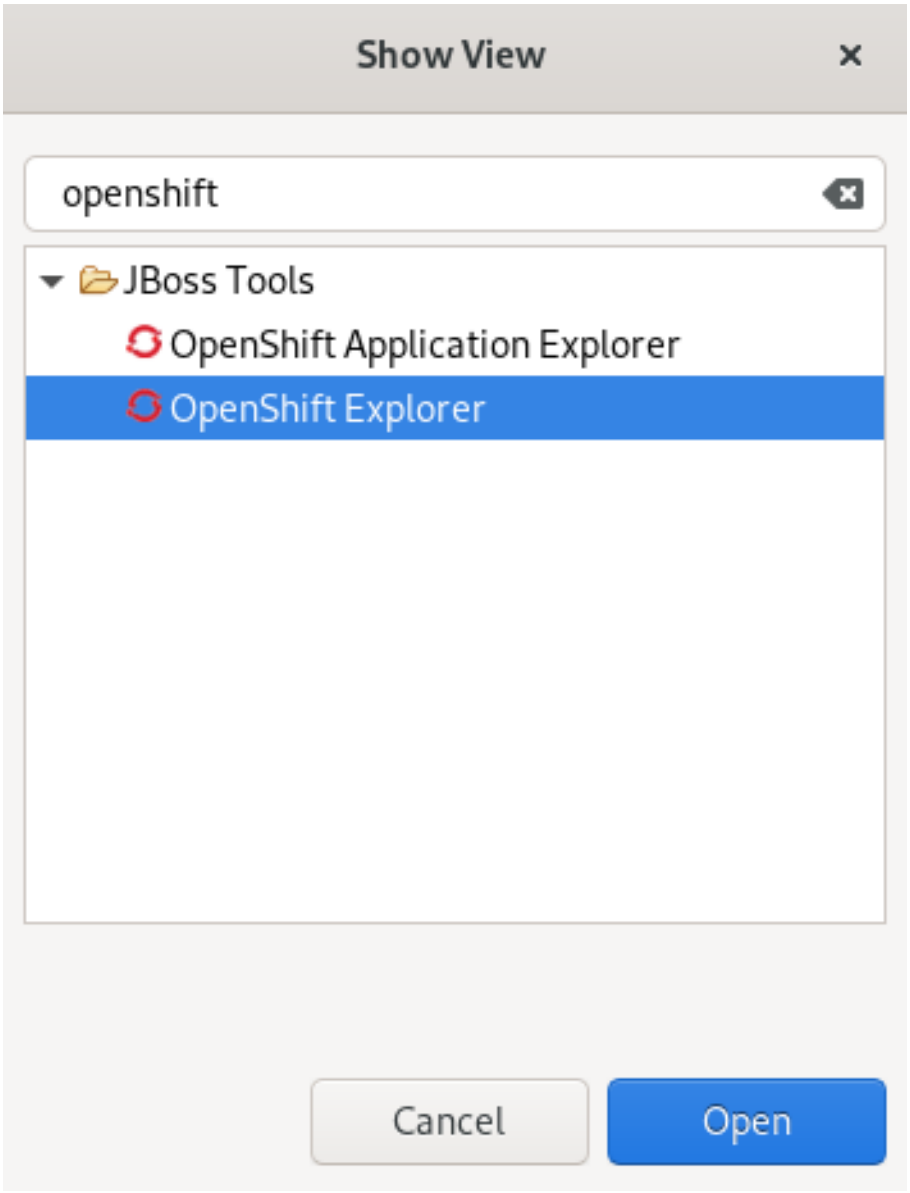
The CodeReady Studio built-in web browser opens, displaying your application.

2.1.6. Deleting an OpenShift Container Platform project

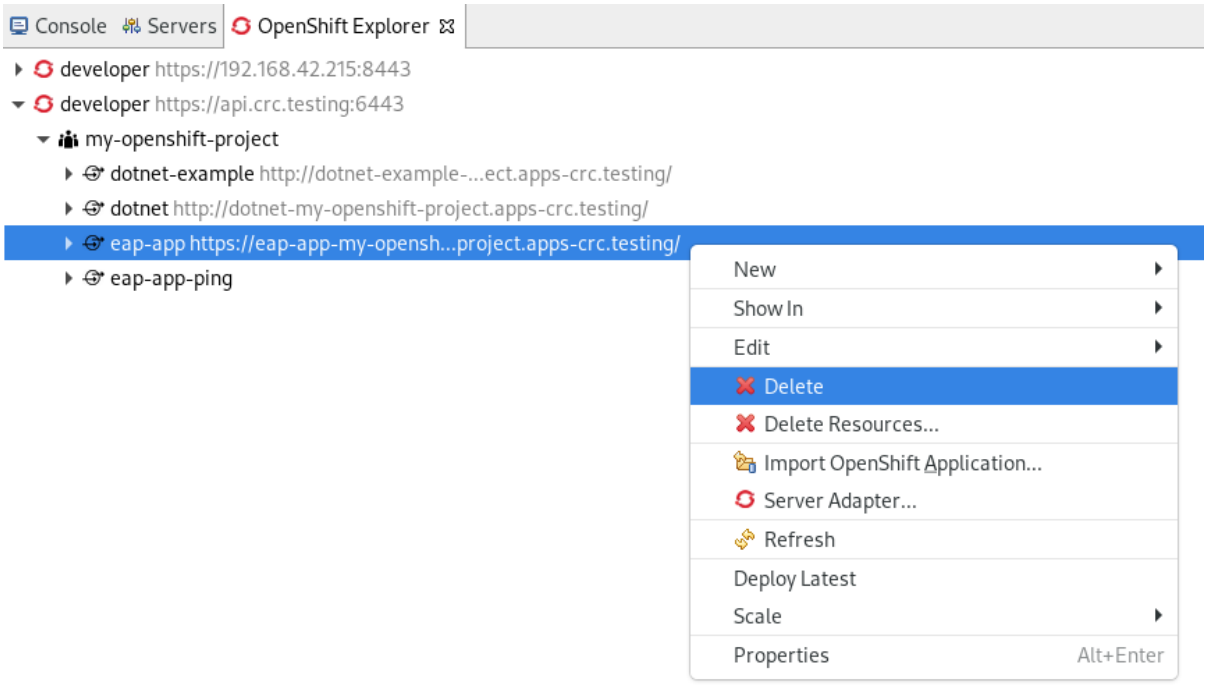
You may choose to delete a project from the workspace for a fresh start in project development or after you have concluded development in a project. When you delete a project, all resources associated with the project are deleted as well.

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**.
   A **Delete OpenShift Resource** window prompts you for consent.

8. Click **OK**.

Your project is now deleted.

2.2. SETTING UP AND REMOTELY MONITORING AN OPENSHIFT CONTAINER PLATFORM APPLICATION IN CODEREADY STUDIO

CodeReady Studio allows users to set up a connection to a remote instance of OpenShift Container Platform and use application and build logs to troubleshoot and monitor running applications.

2.2.1. Setting up OpenShift Client Binaries

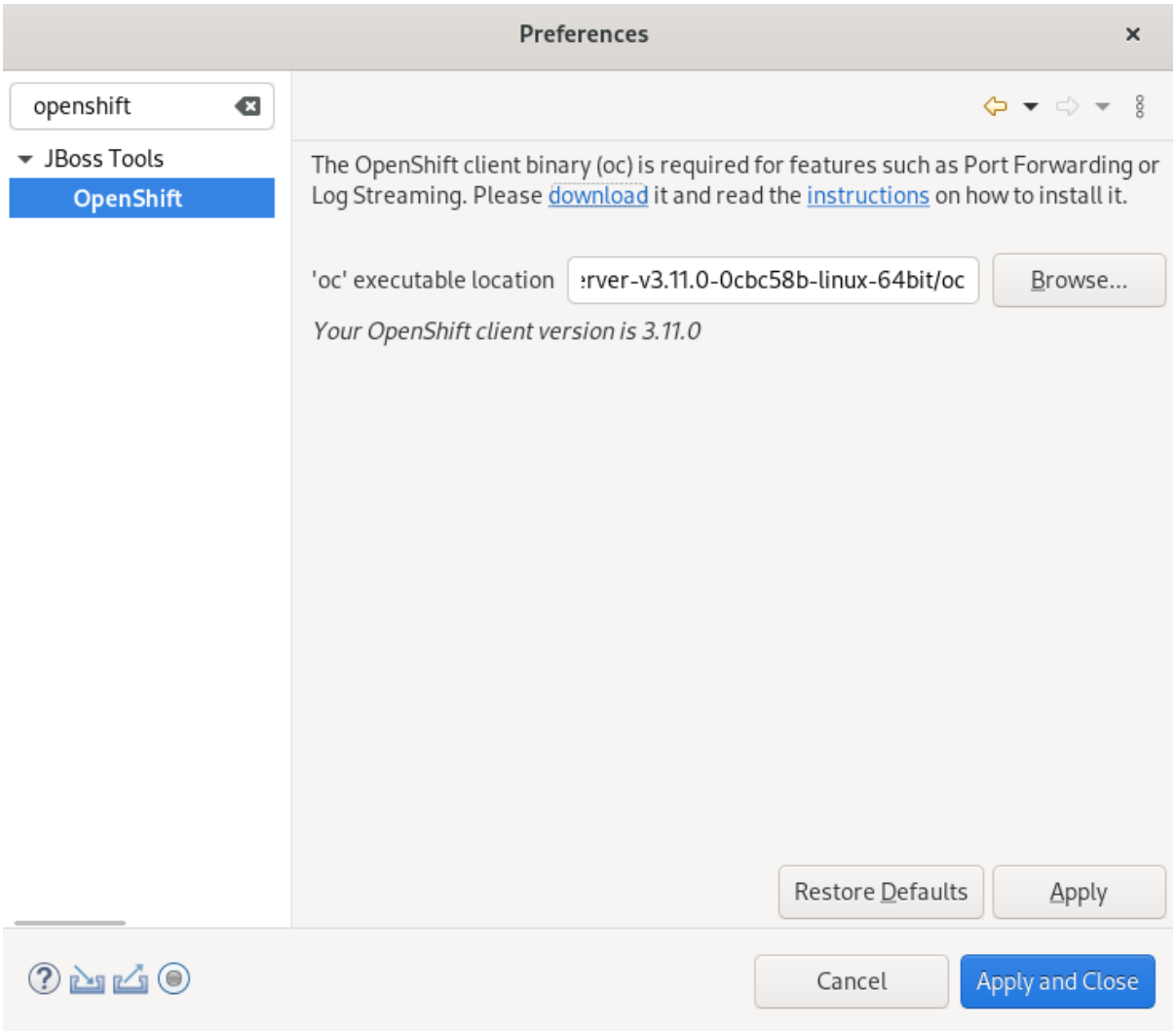
Prerequisites

Before setting up port forwarding or streaming application and build logs, 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 local ports to their remote counterparts to access data or debug your application.

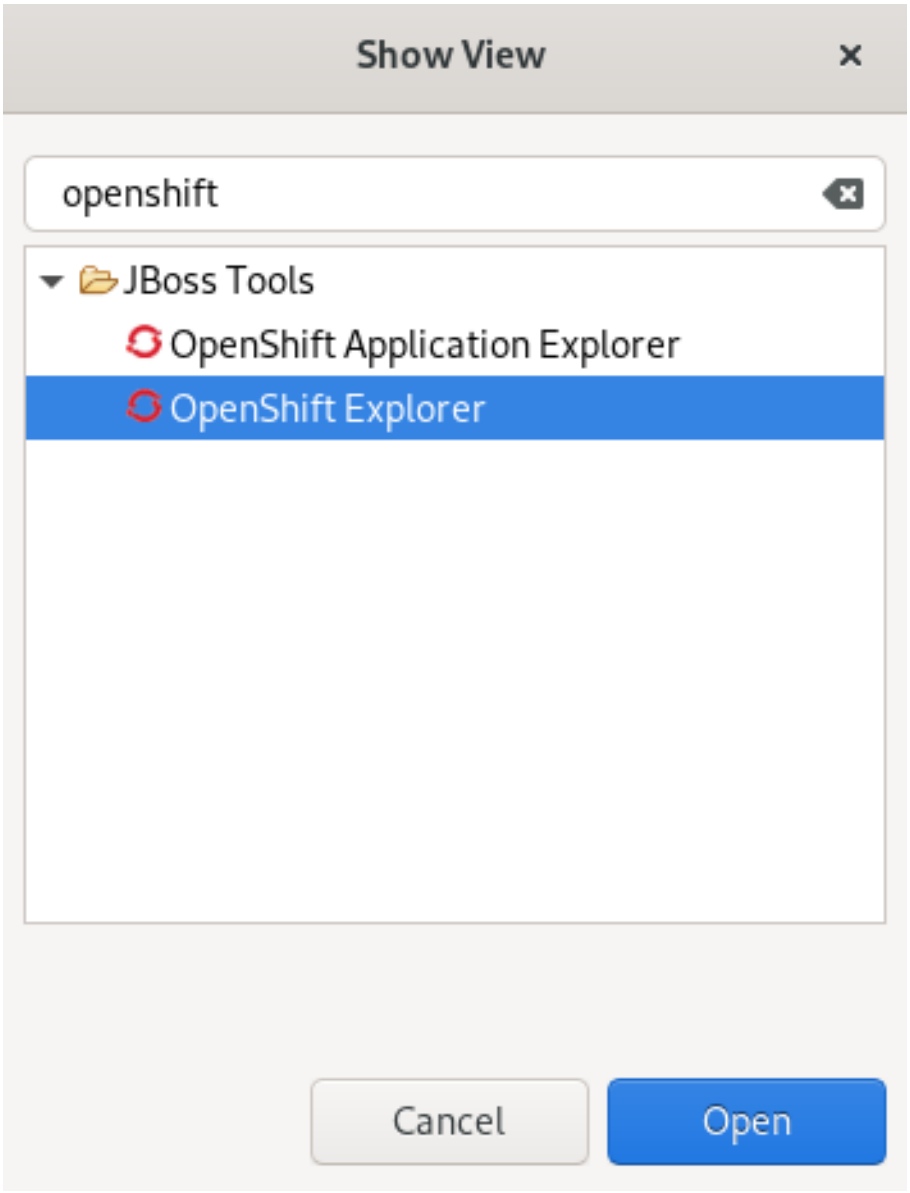
Port forwarding automatically stops due to any of the following reasons:

- The OpenShift Container Platform connection terminates
- CodeReady Studio shuts down
- The workspace is changed

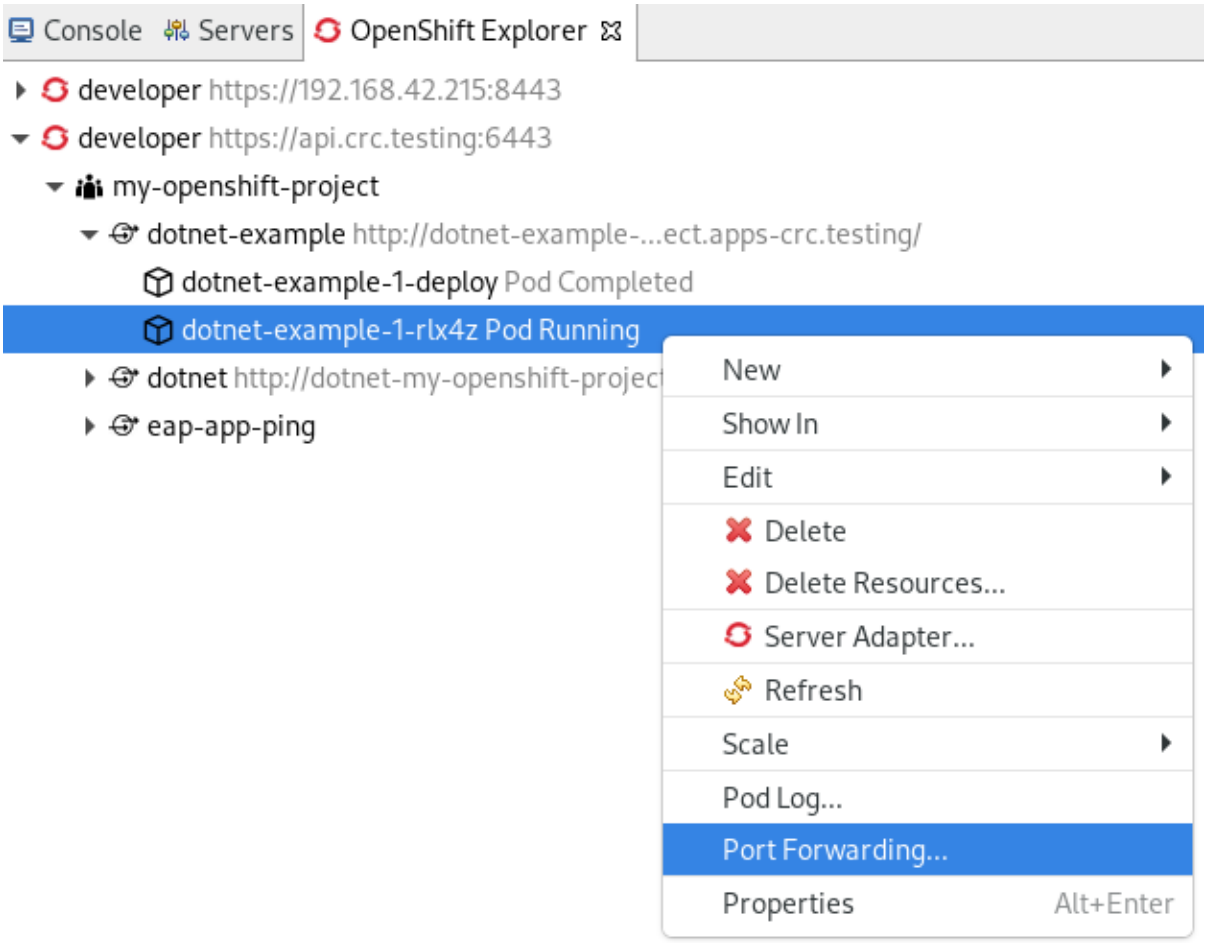
Port forwarding must be enabled each time to connect to OpenShift Container Platform from CodeReady Studio.

Procedure

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

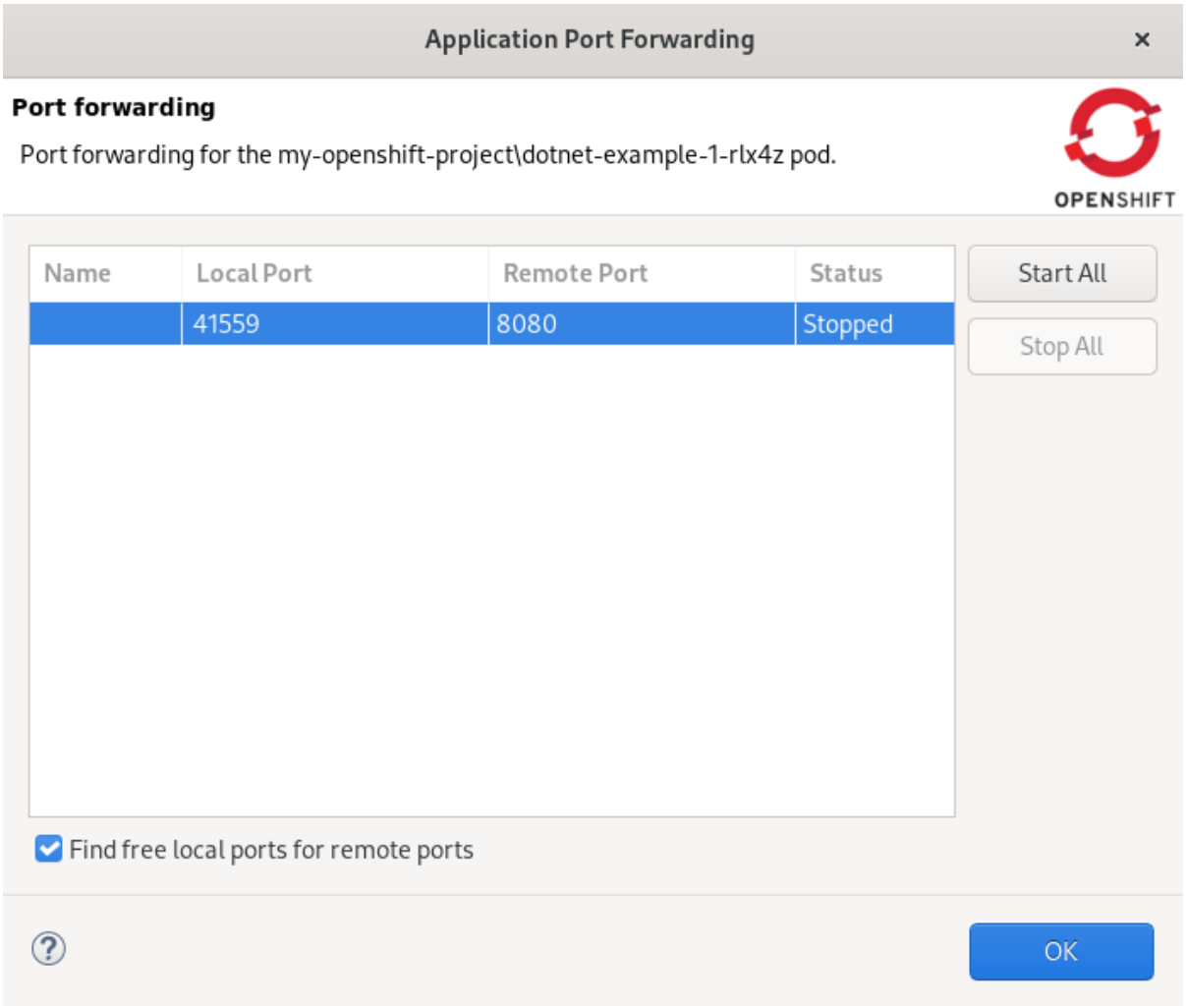


3. Enter **OpenShift** in the search field.
4. Select **OpenShift Explorer**.
5. Click **Open**.
   The **OpenShift Explorer** view appears.



6. Expand the OpenShift Container Platform connection.

7. Right-click your **application → 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**.

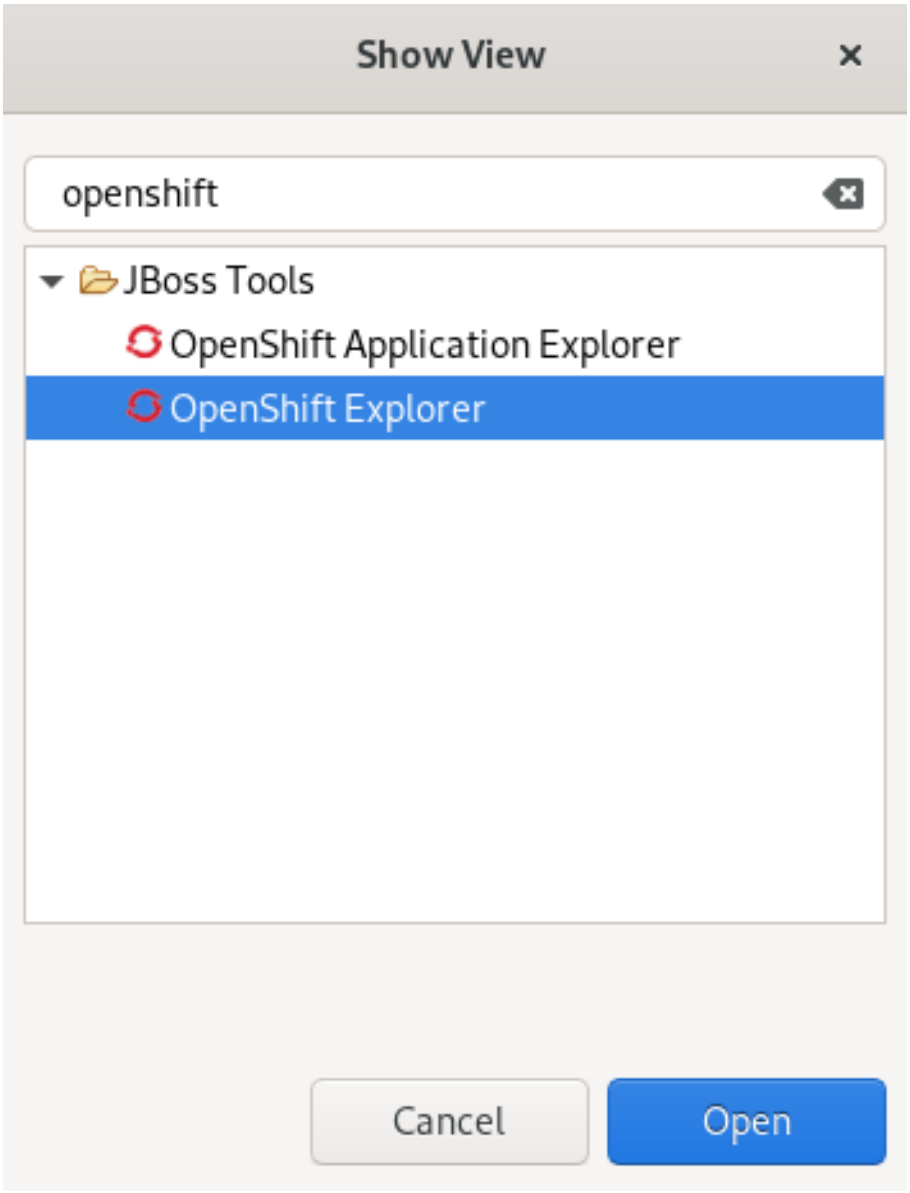
The **Console** view appears showing the port-forwarding starting process.

2.2.3. Streaming Pod Logs

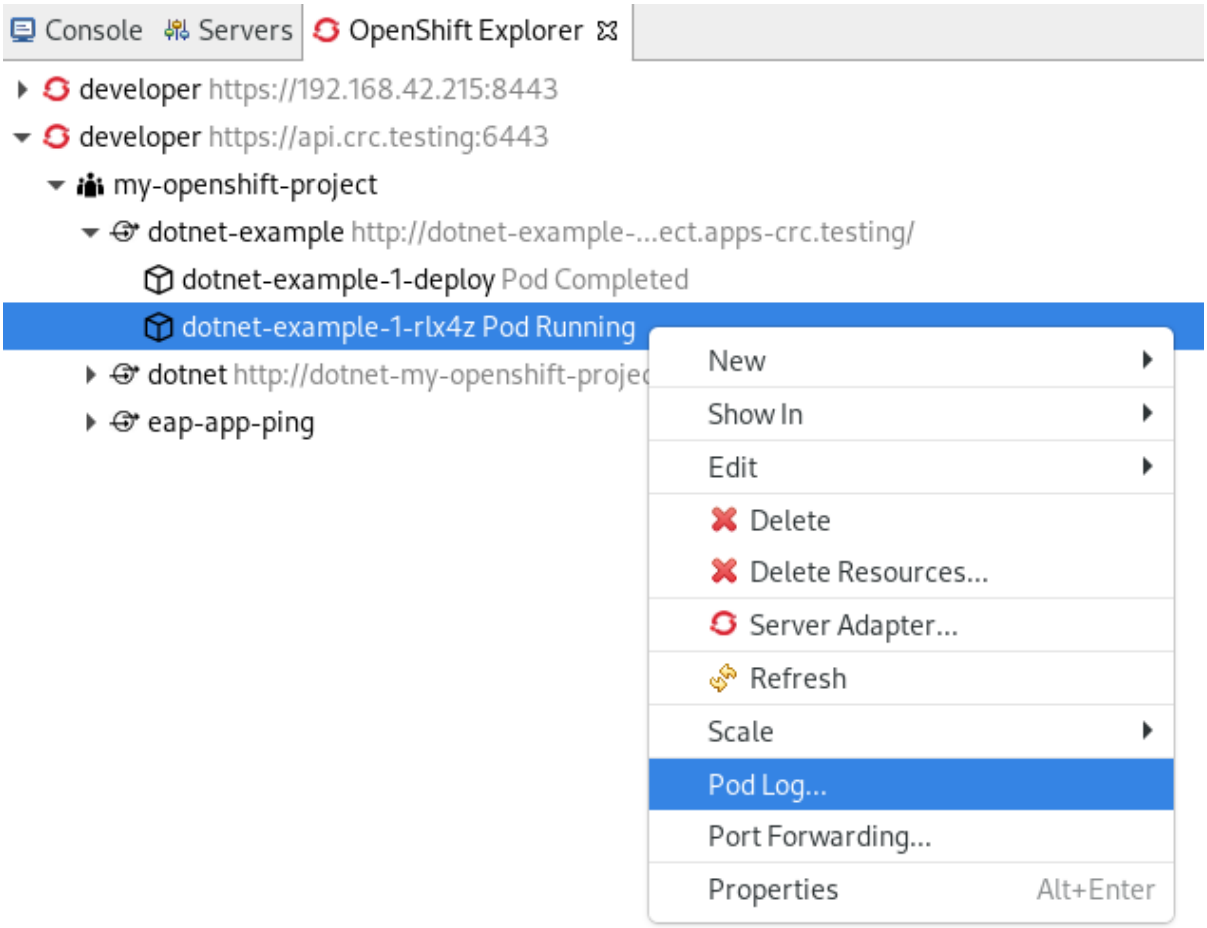
Pod logs are general logs for an application running on a remote OpenShift Container Platform instance. The streaming pod logs feature in CodeReady Studio is used to monitor applications and use the previous pod log to troubleshoot if the application fails or returns errors.

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 → Pod Log**.

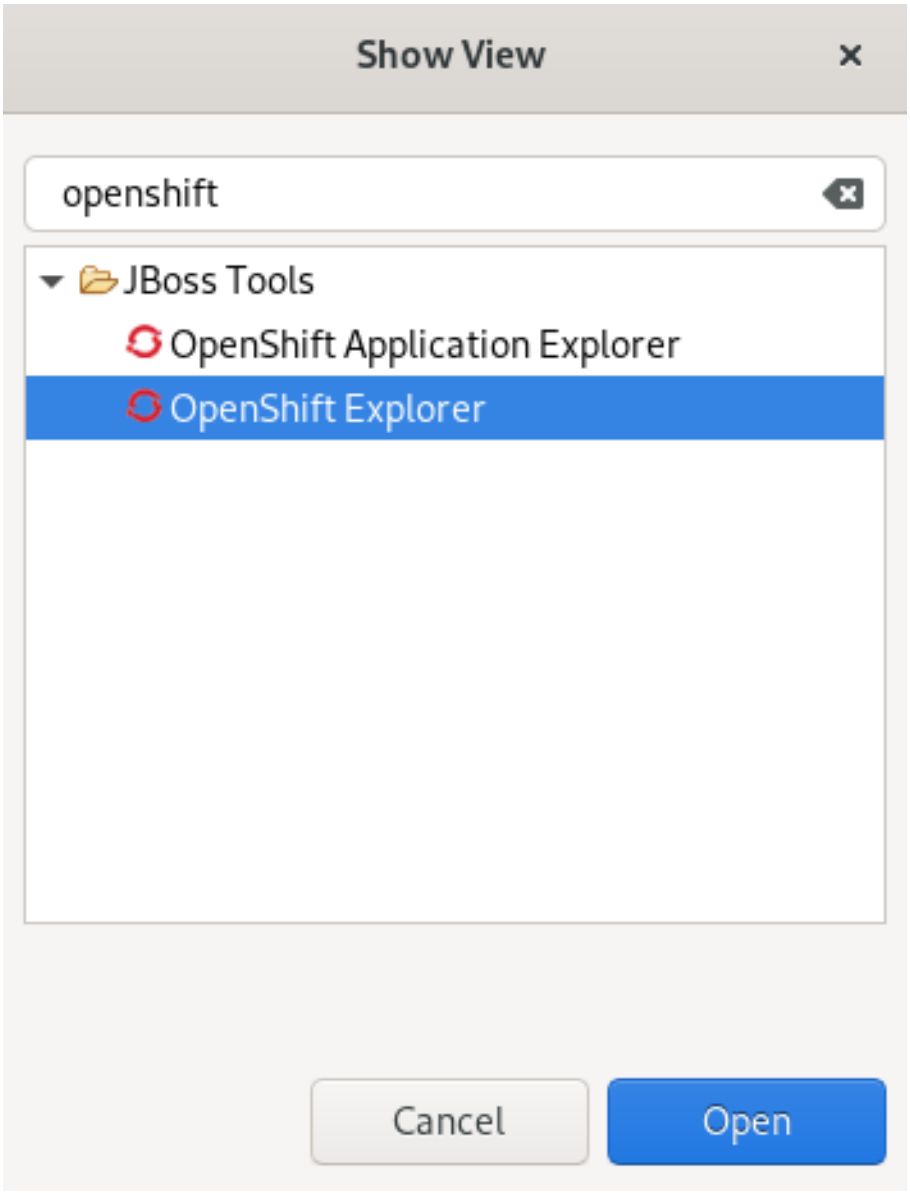
The **Console** view appears displaying the Pod Log.

2.2.4. Streaming Build Logs

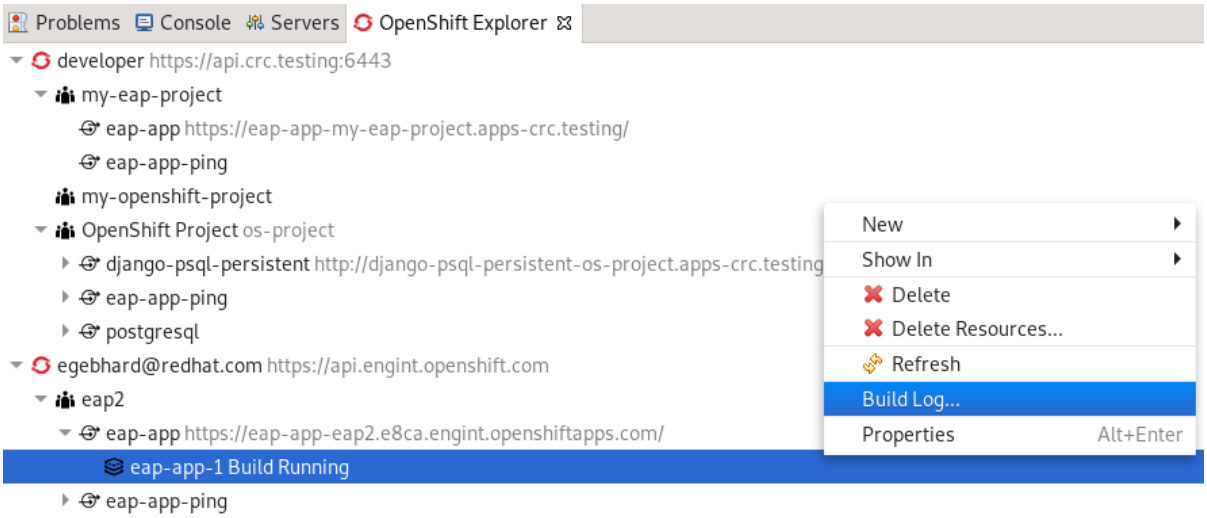
Build logs are logs that document changes to applications running on a remote OpenShift Container Platform instance. The streaming build logs feature in CodeReady Studio is used to view the progress of the application build process and to debug the application.

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 IN CODEREADY STUDIO

3.1. MANAGING DOCKER CONNECTIONS

3.1.1. Setting up a Docker account

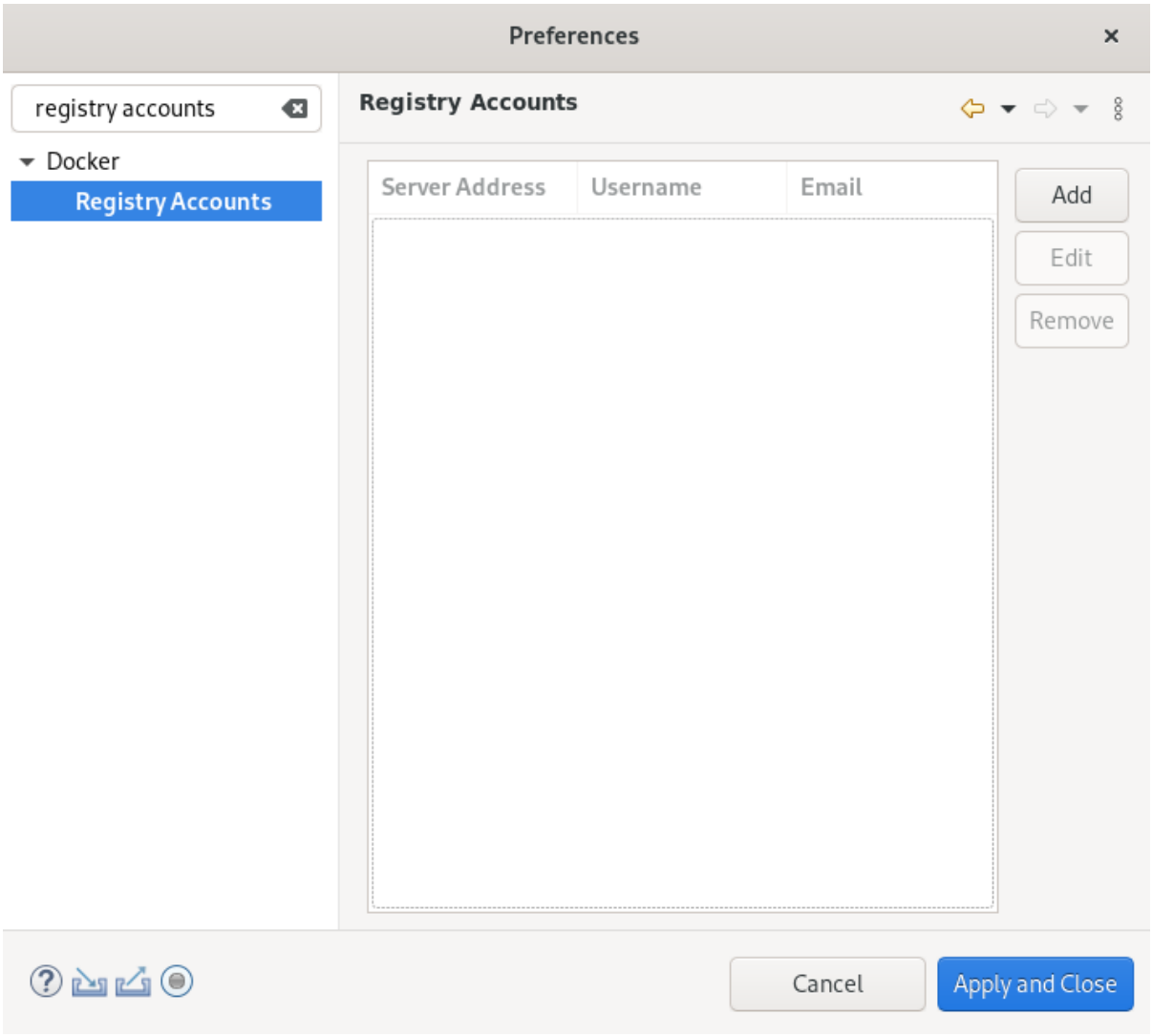
The following section describes how to set up a Docker account in CodeReady Studio.

Prerequisites

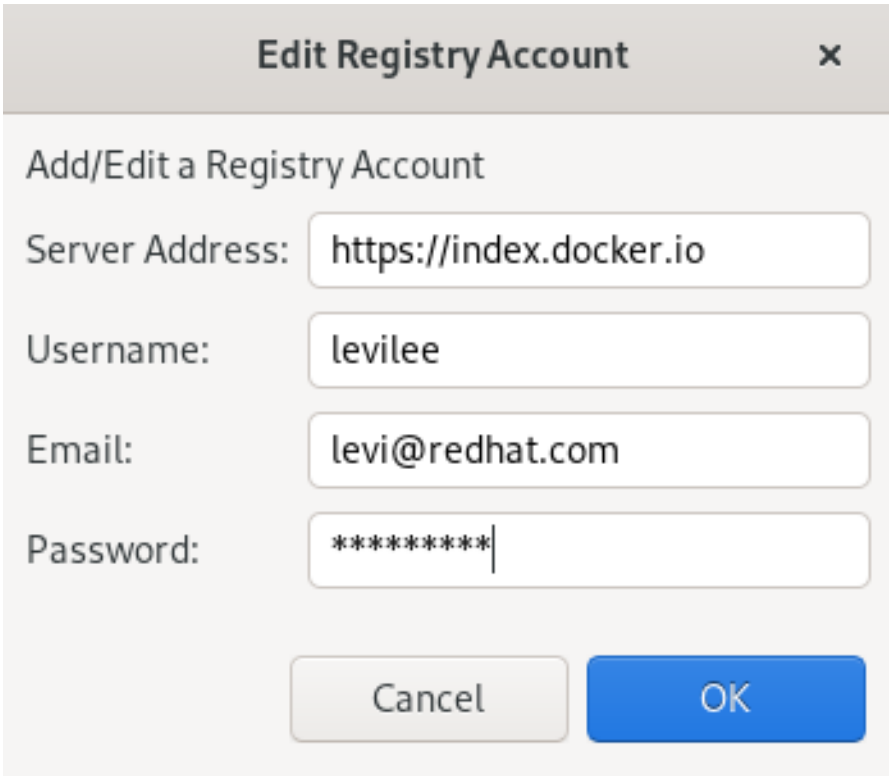
- Docker is installed on your system.
  For more information on how to install Docker, see Docker Docs - Get Docker.
- You have a Docker ID.
  For more information on how to get a Docker ID, see Register for a Docker ID.

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.



6. Enter the **Server Address**.
7. Enter your Doker 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**.

Your Docker account has been set up.

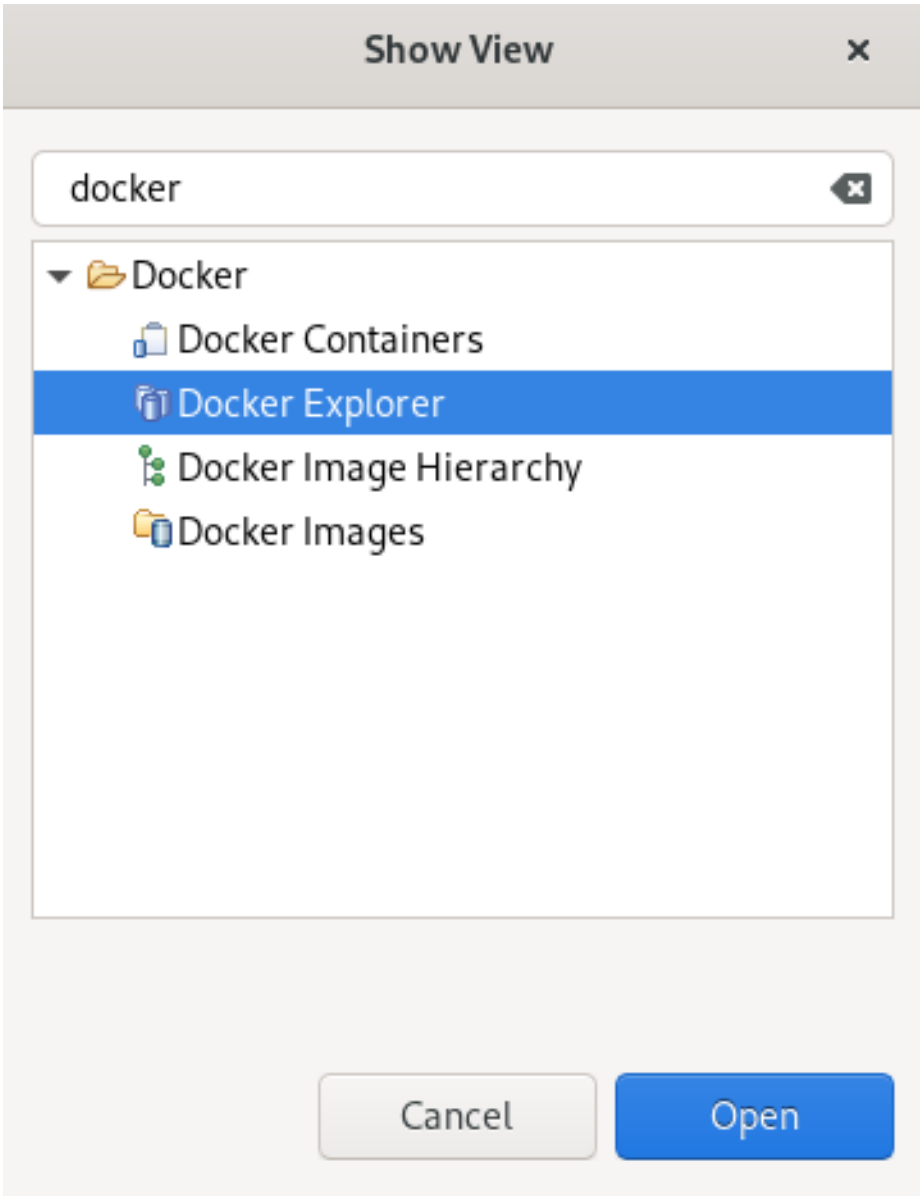
3.1.2. Testing an existing Docker connection

Prerequisites

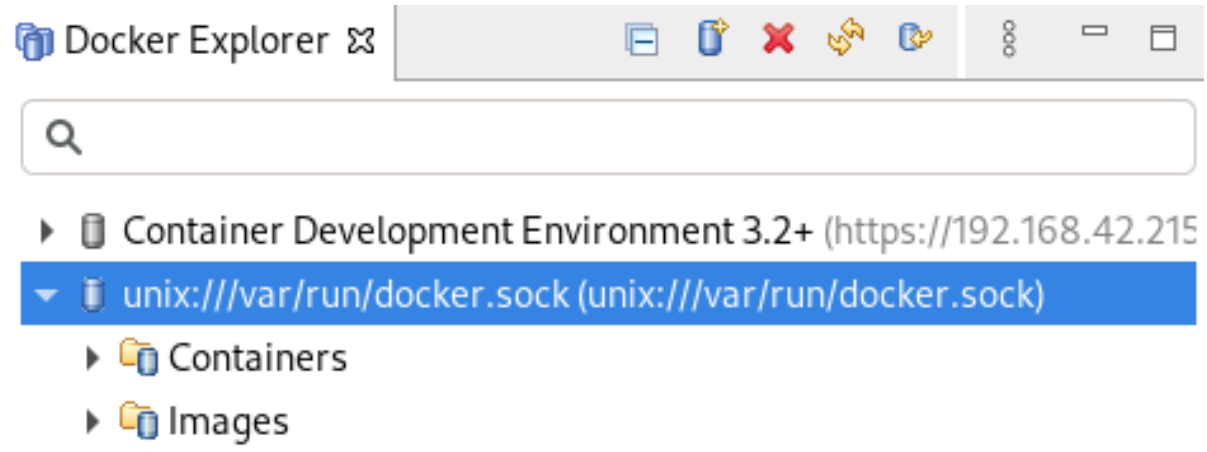
- Your Docker account in CodeReady Studio is set up.
  For more information on how to set up a Docker account in CodeReady Studio, see Setting up a Docker account.

Procedure

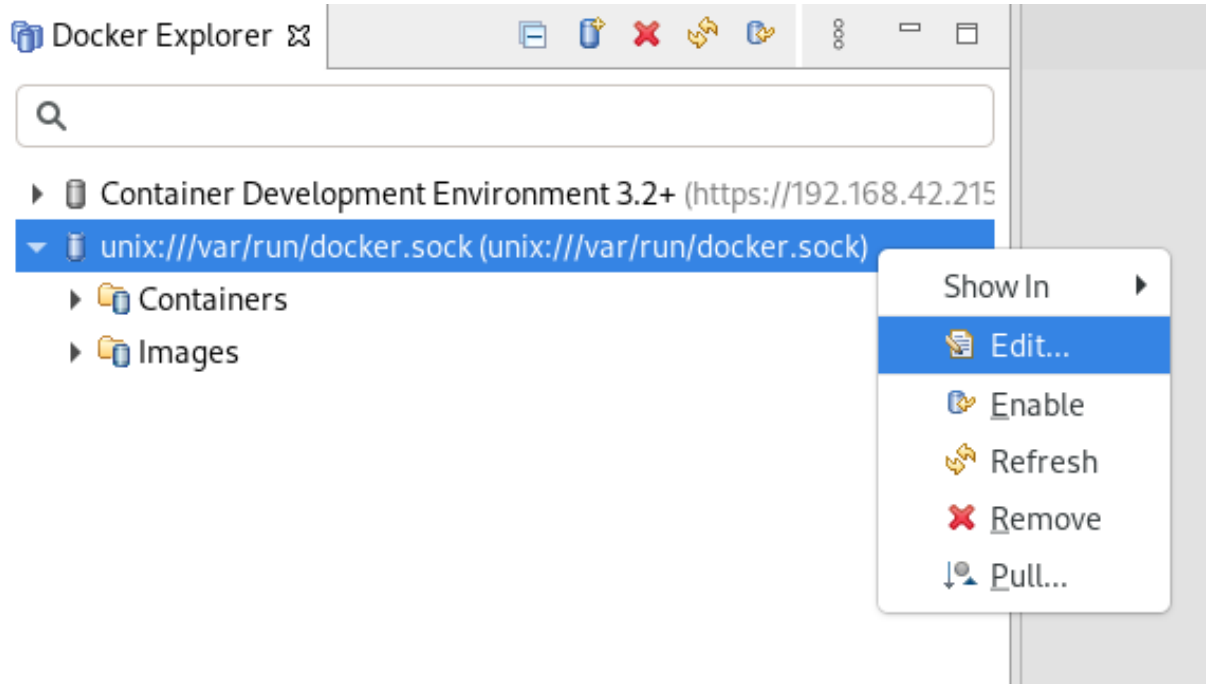
1. Start CodeReady Studio.
2. Click **Window → Show View → Other**.
   The **Show View** window appears.



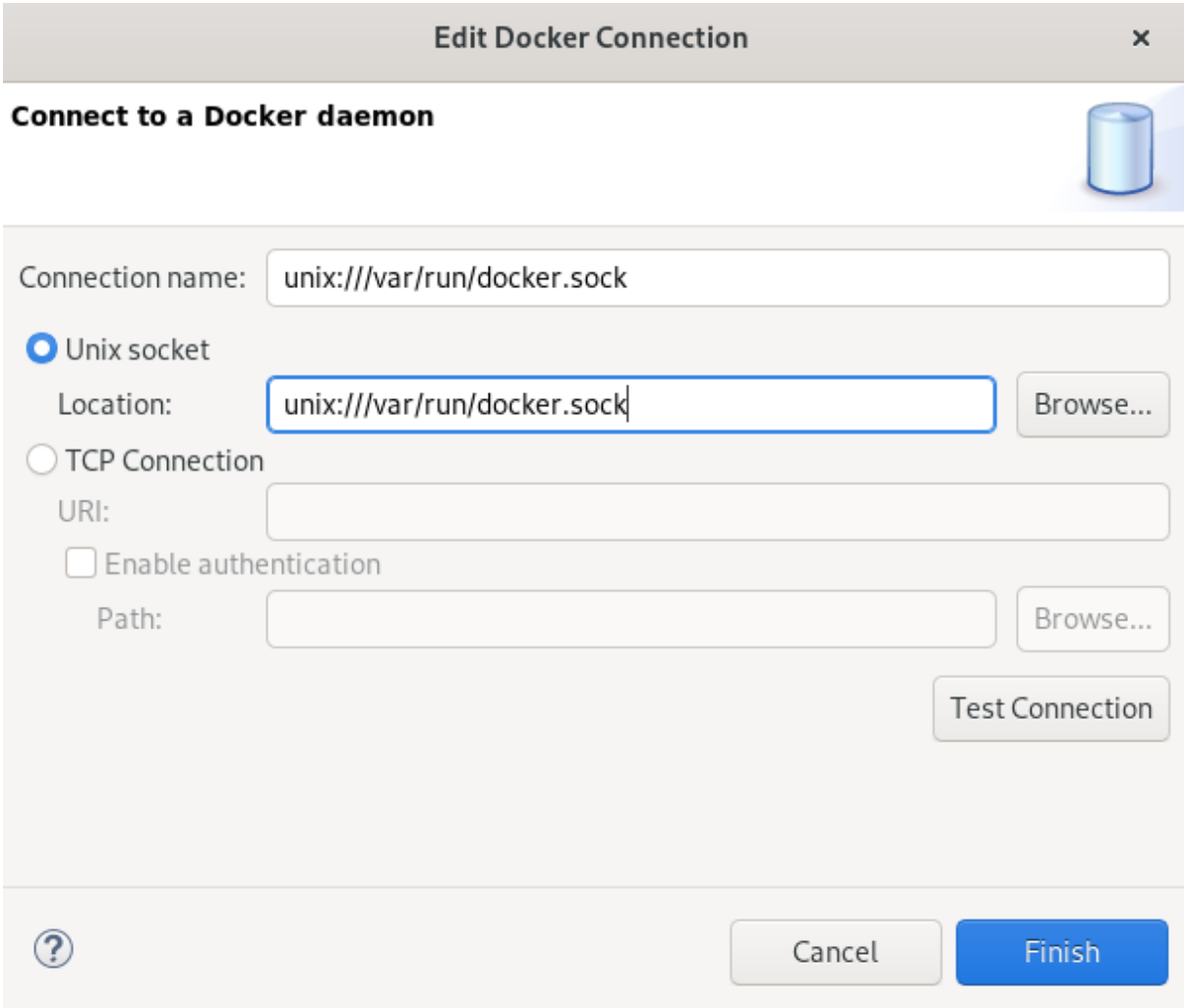
3. Enter **Docker** in the search field.
4. Select **Docker Explorer**.
5. Click **Open**.
   The **Docker Explorer** view appears.



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 **Ping succeeded!** appears.

8. Click **OK**.

9. Click **Finish**.

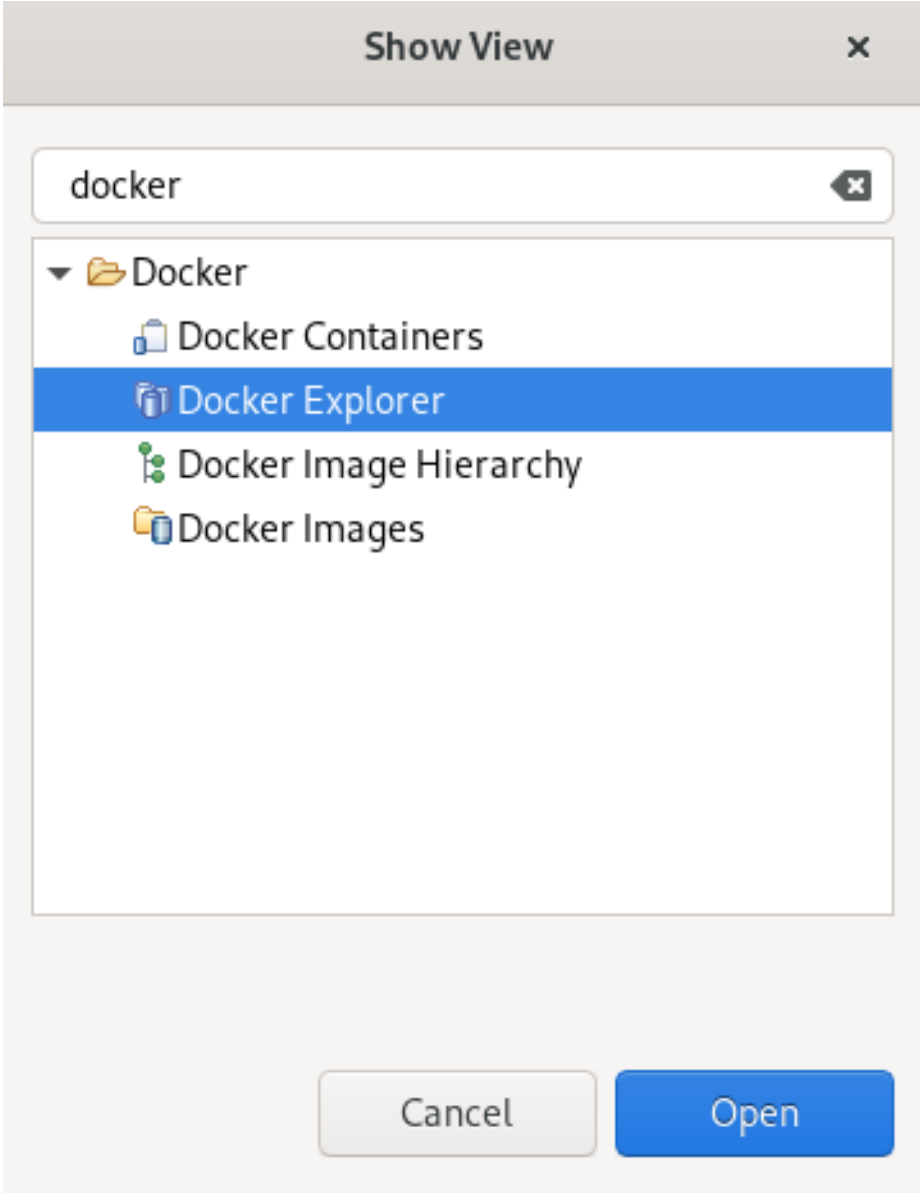
3.1.3. Editing a Docker connection

Prerequisites

- A set up Docker account in CodeReady Studio.
  For more information on how to set up a Docker account in CodeReady Studio, see Setting up a Docker account.

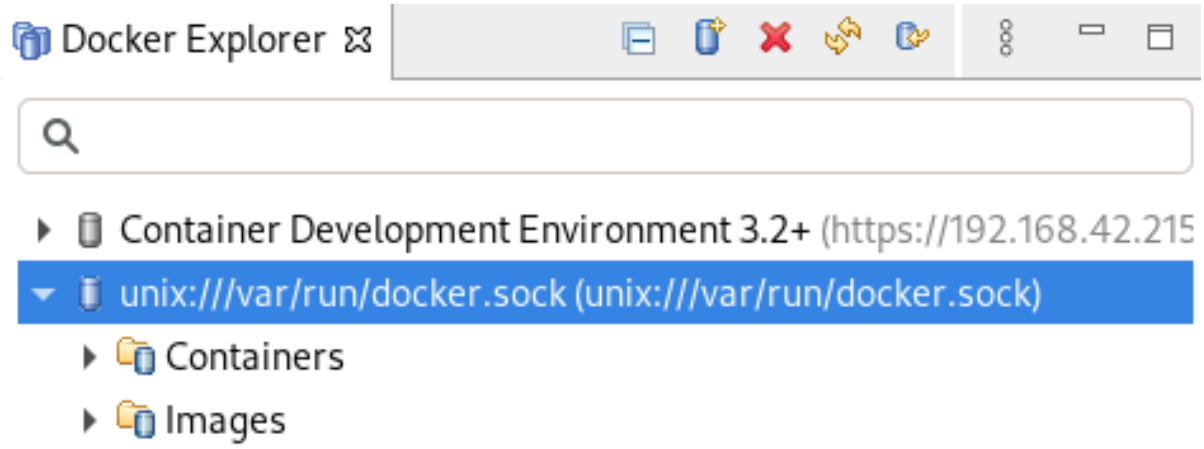
Procedure

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

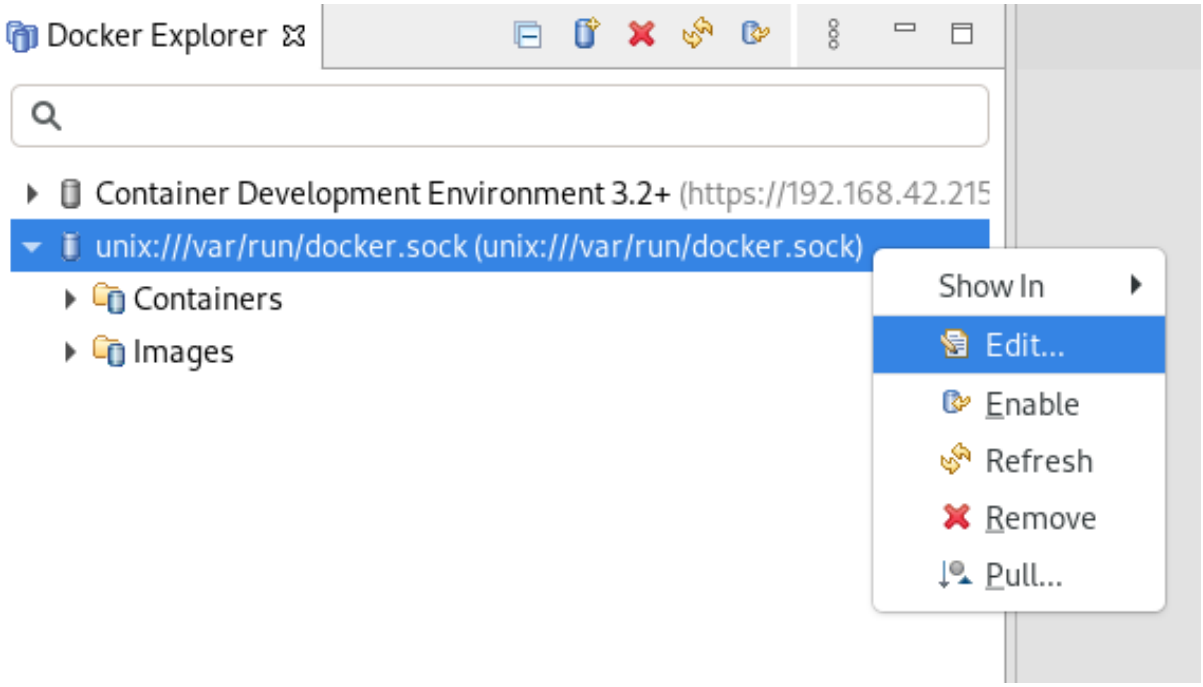


3. Enter **Docker** in the search field.
4. Select **Docker Explorer**.

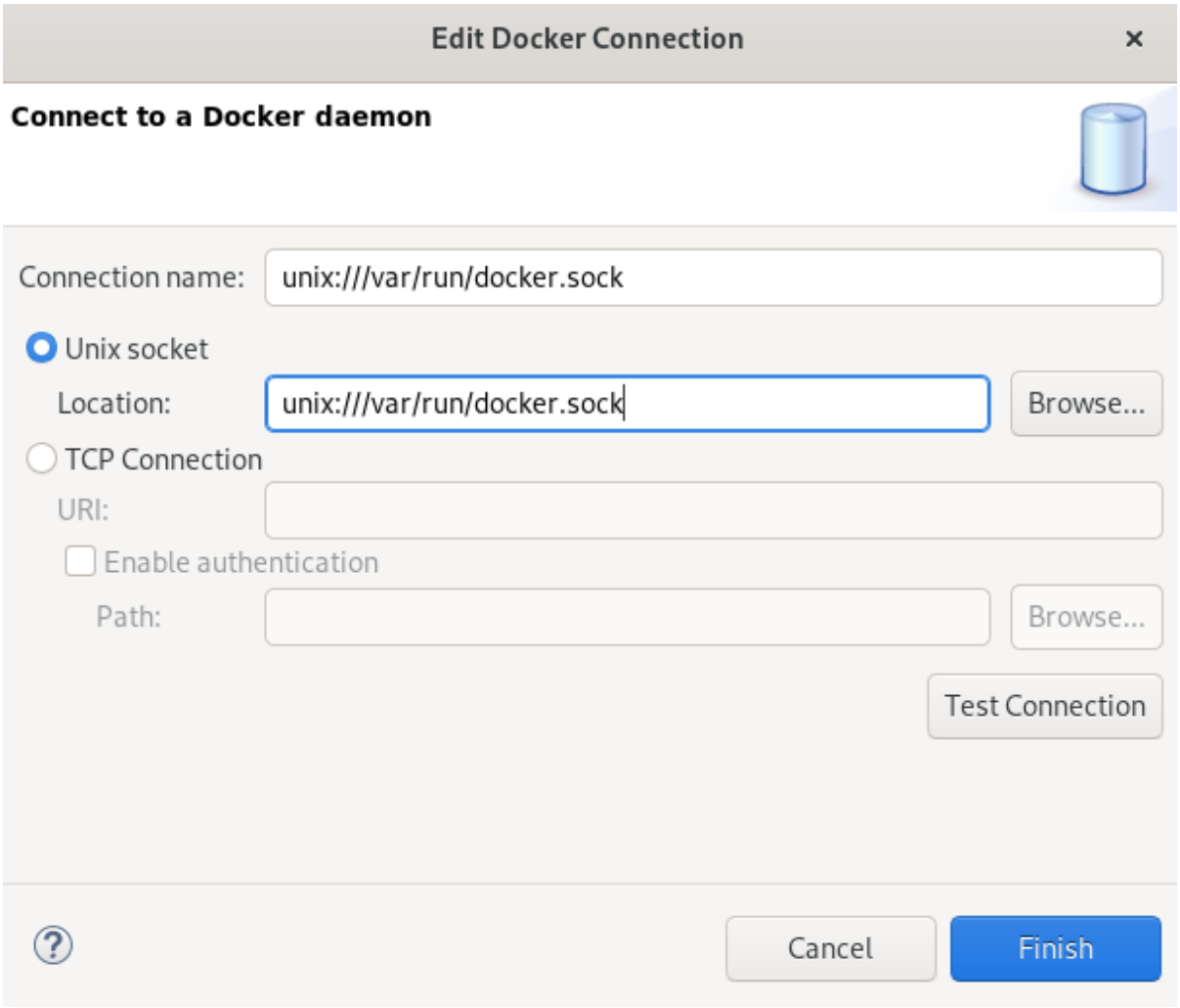
5. Click **Open**.
   The **Docker Explorer** view appears.



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



   The **Edit Docker Connection** window appears.



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

8. Click **Finish**.

Your docker connection has been edited.

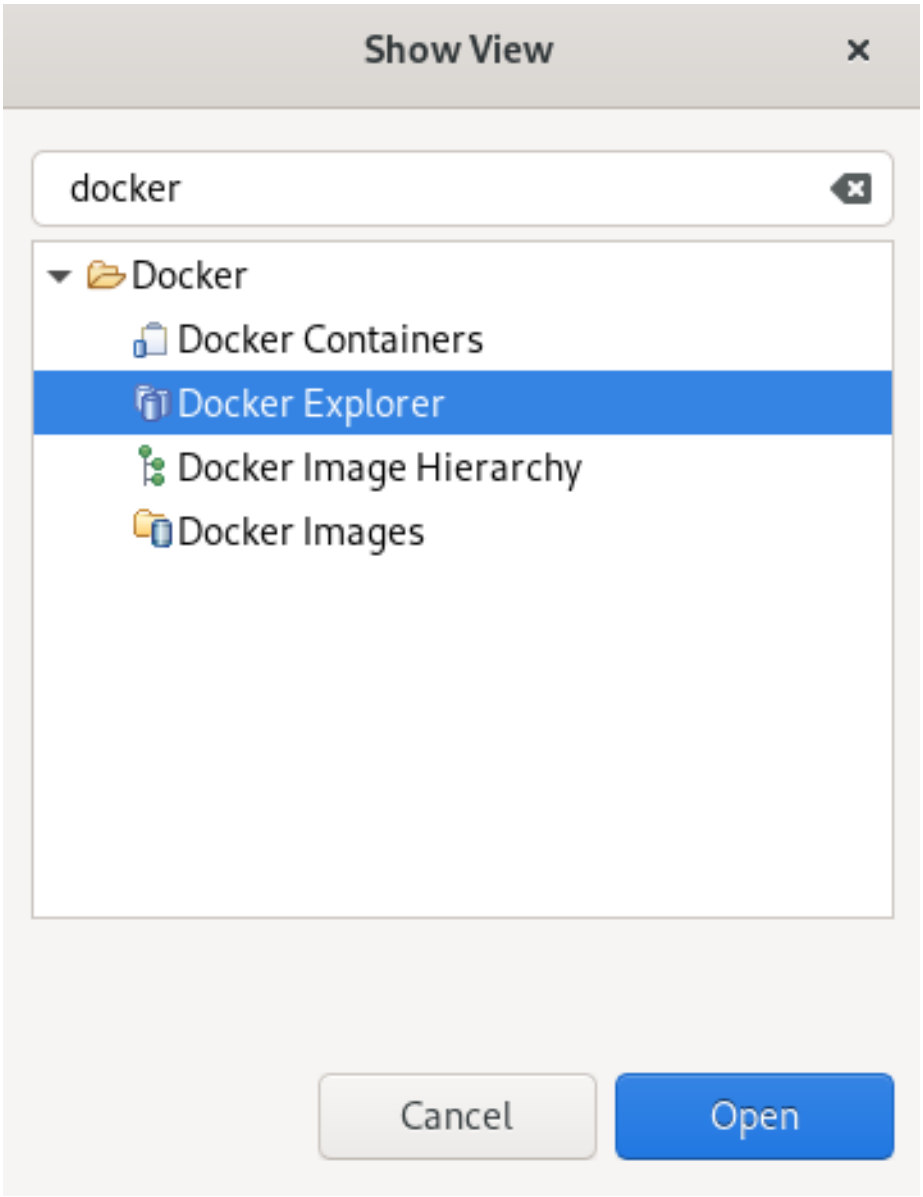
3.2. MANAGING DOCKER IMAGES

3.2.1. Pulling Docker images

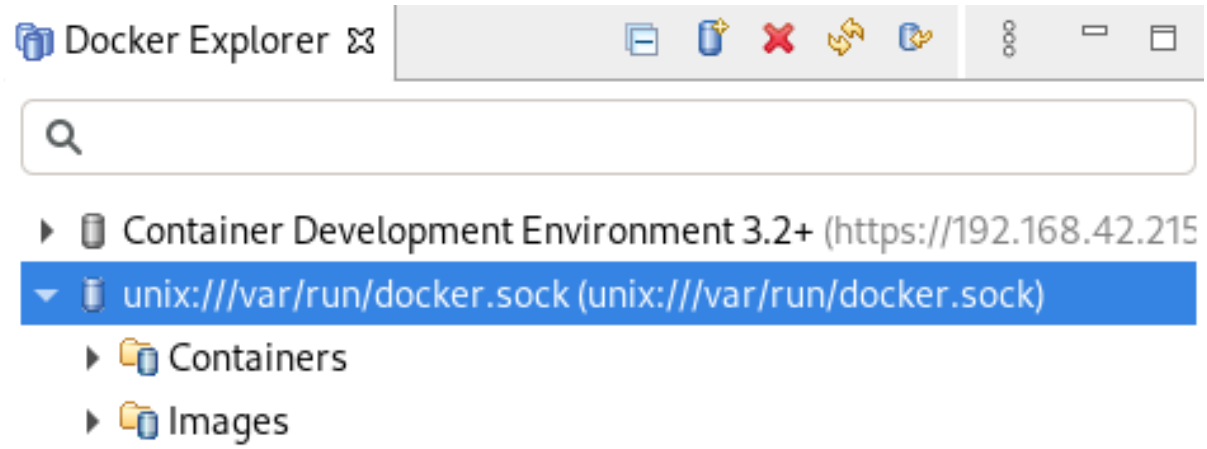
Procedure

1. Start CodeReady Studio.

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

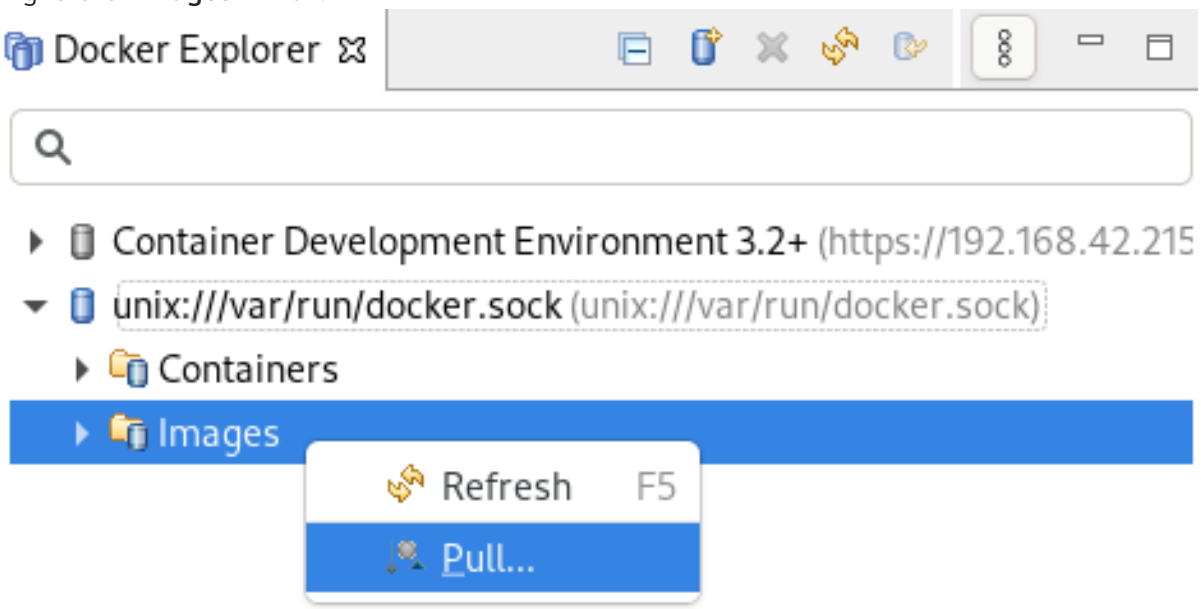


3. Enter **Docker** in the search field.
4. Select **Docker Explorer**.
5. Click **Open**.
   The **Docker Explorer** view appears.

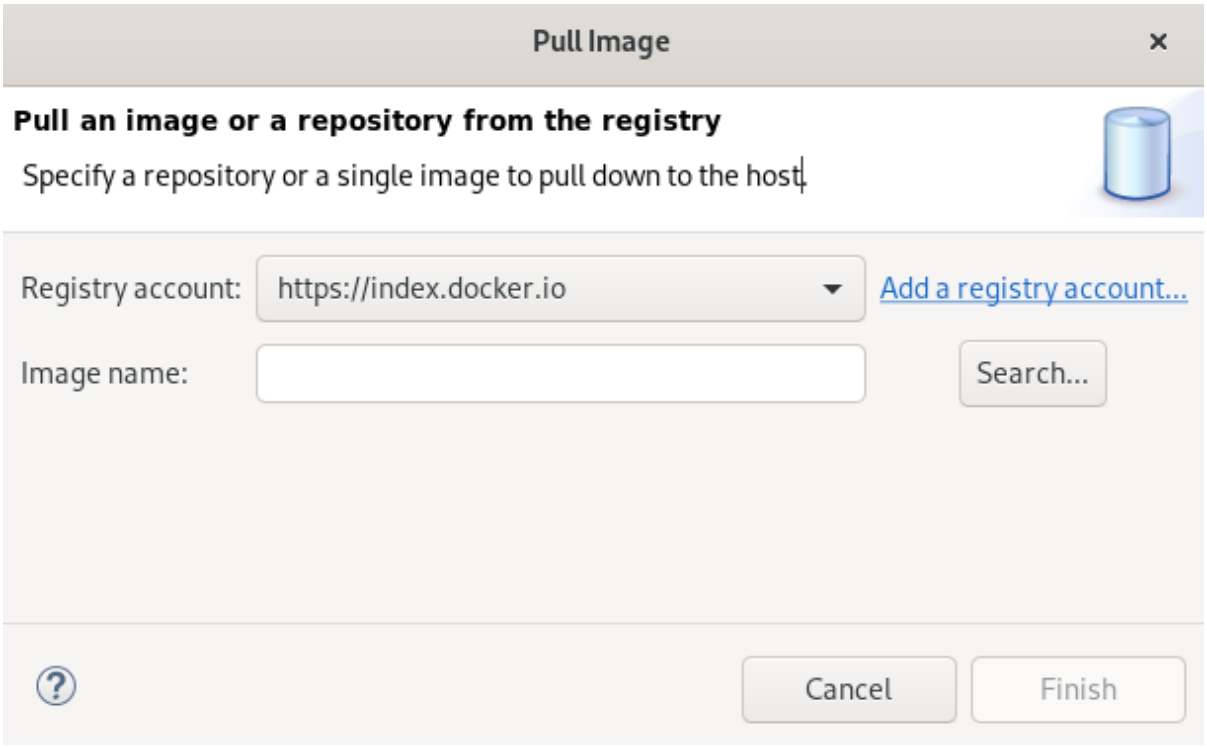


6. Expand the **Docker socket** folder.

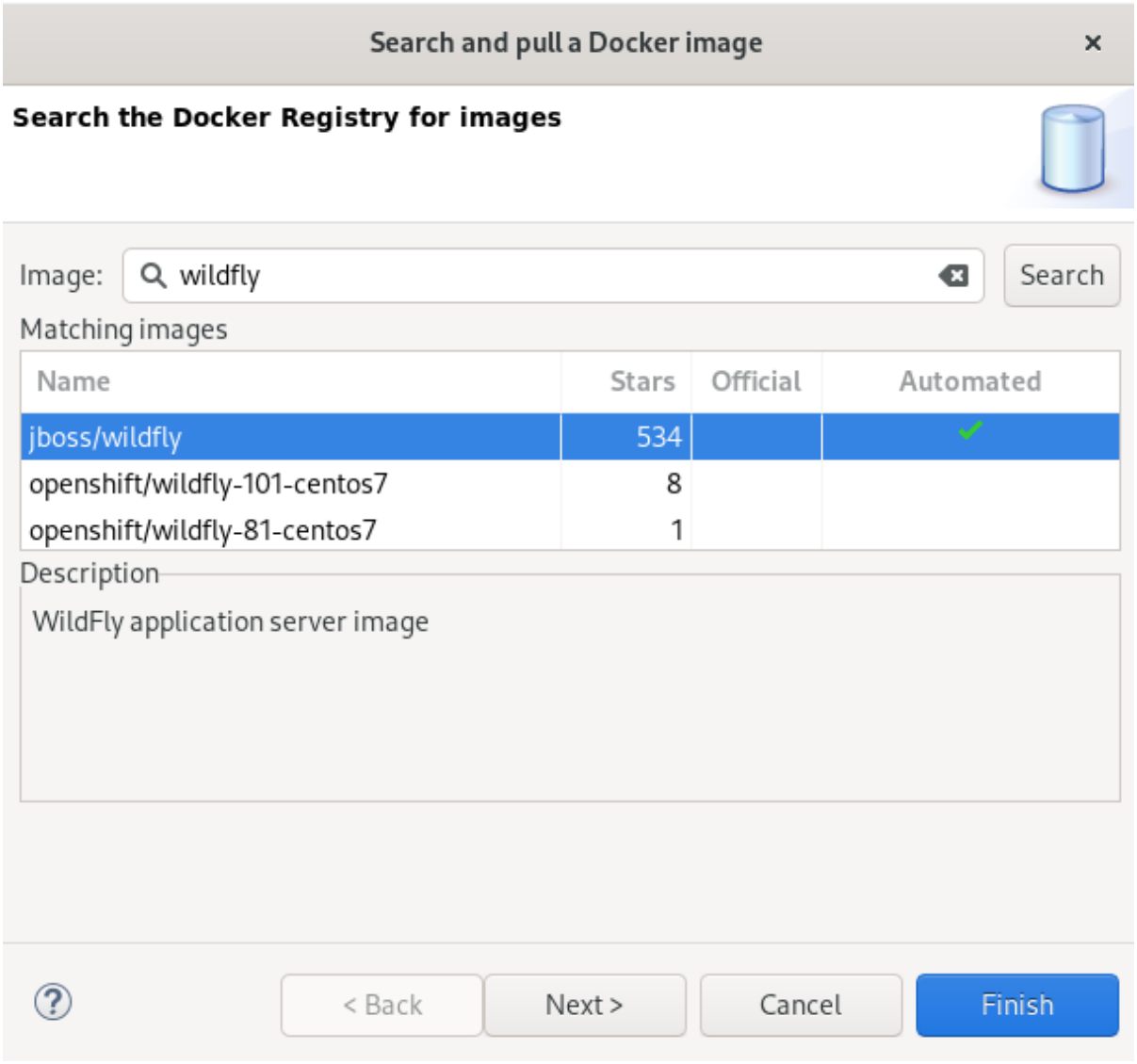
7. Right-click **Images → Pull**.



The **Pull Image** window appears.

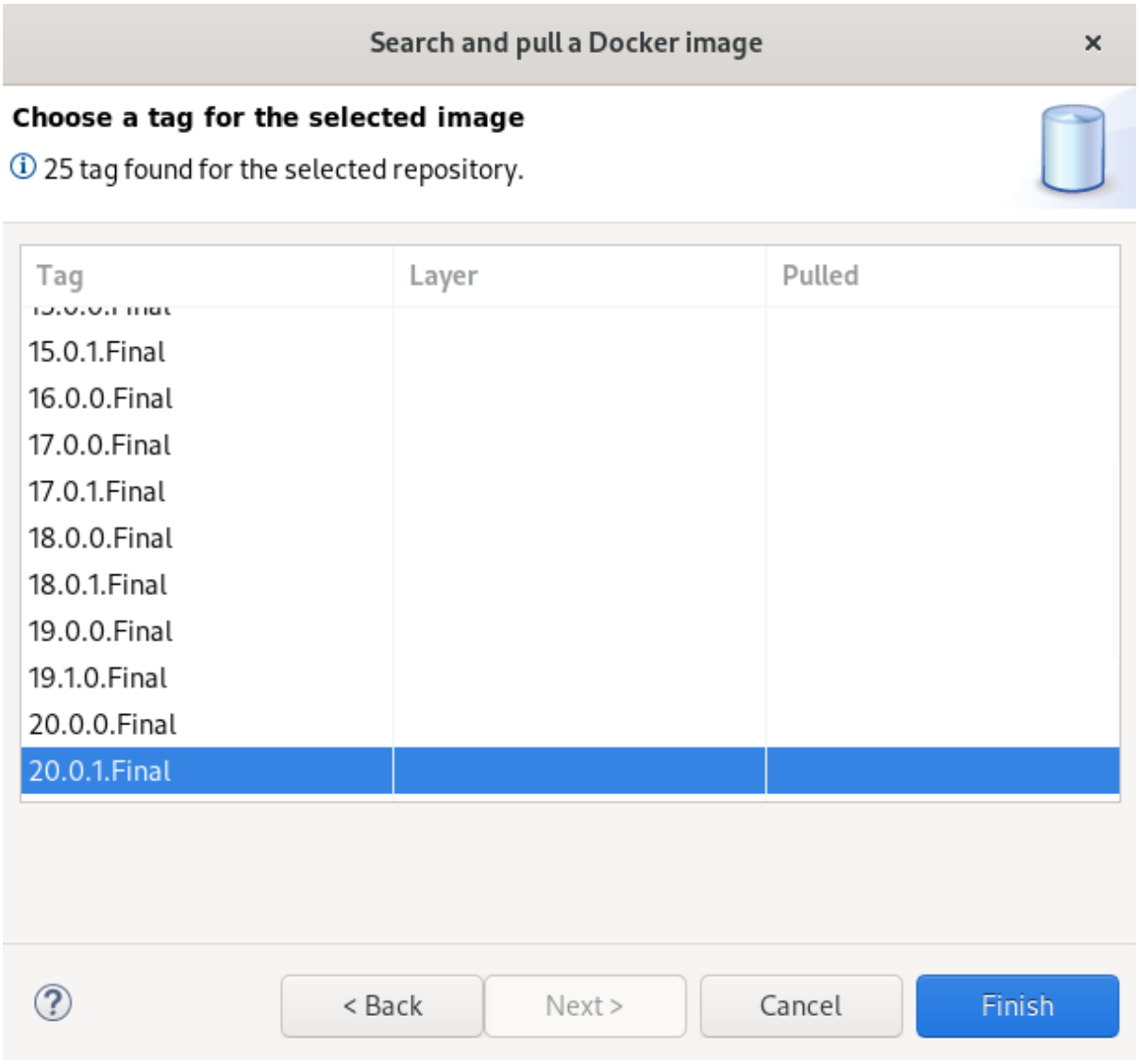


8. Click **Search**.
The **Search the Docker Registry for images**window appears.



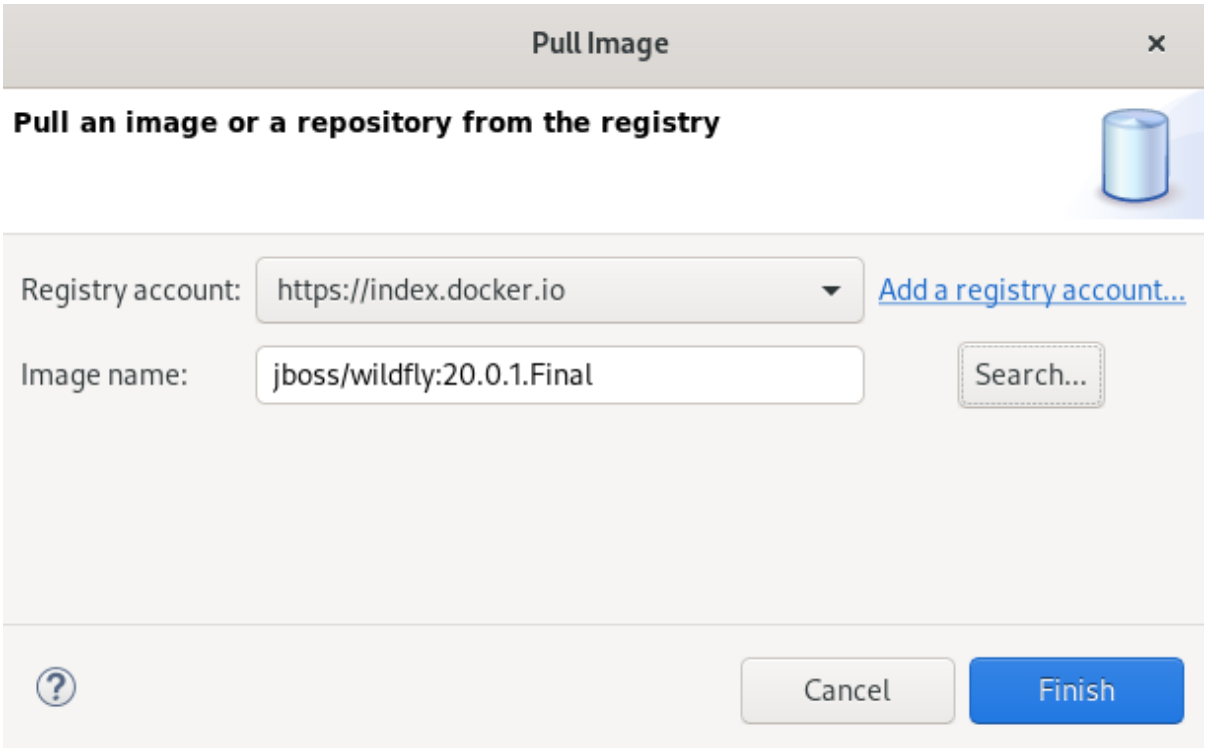
9. Enter your image name into the search field.

10. Click **Next**.
The **Choose a tag for the selected image** window appears.



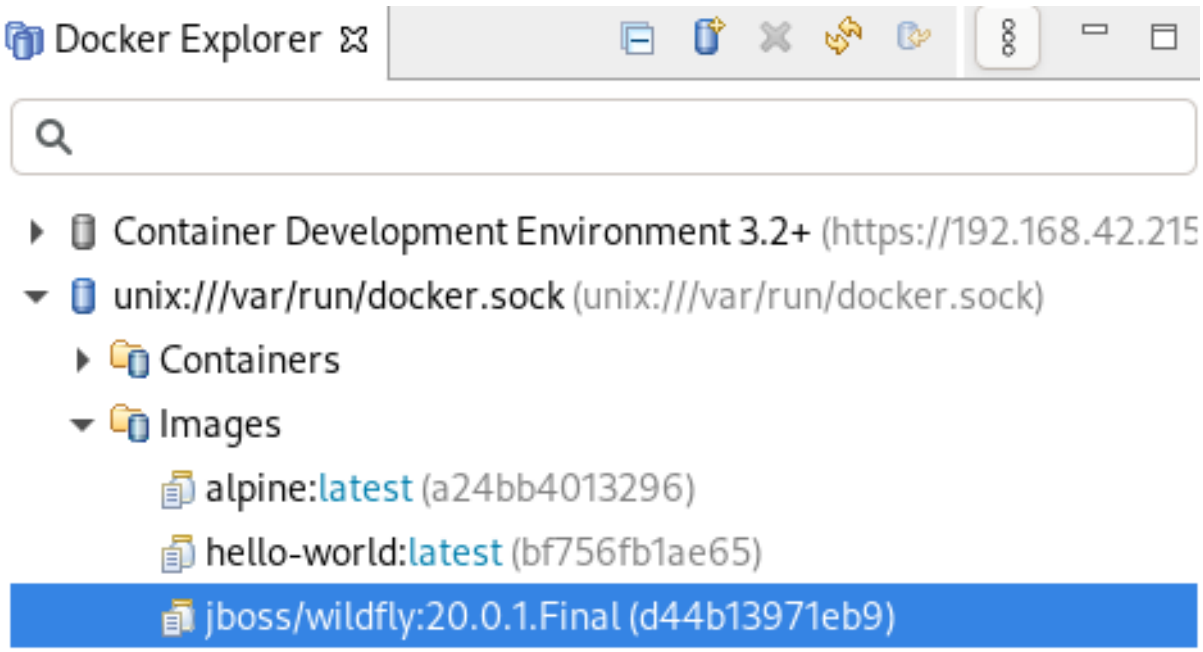
11. Choose a tag for your image.

12. Click **Finish**.
The **Pull an image or a repository from the registry** window appears.



13. Click **Finish**.

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



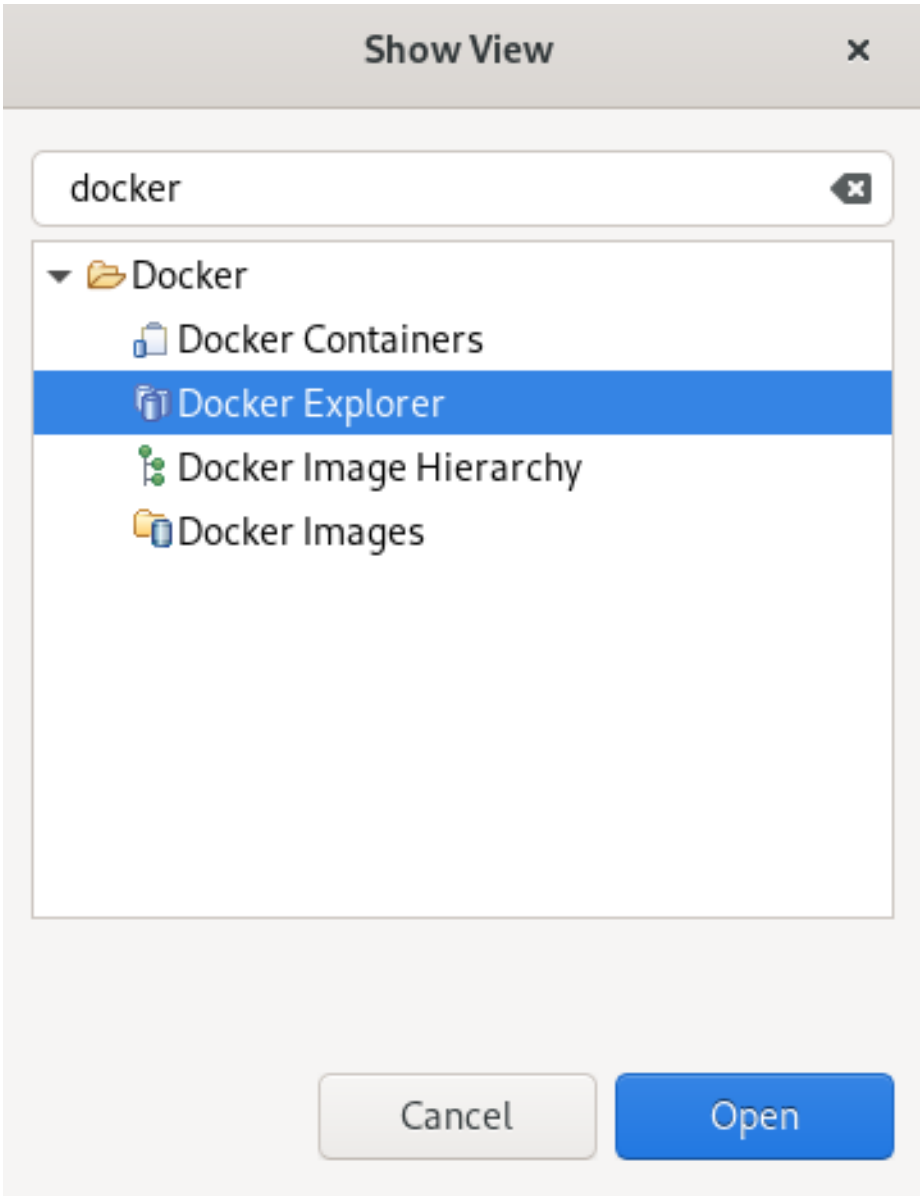
3.2.2. Pushing Docker images

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

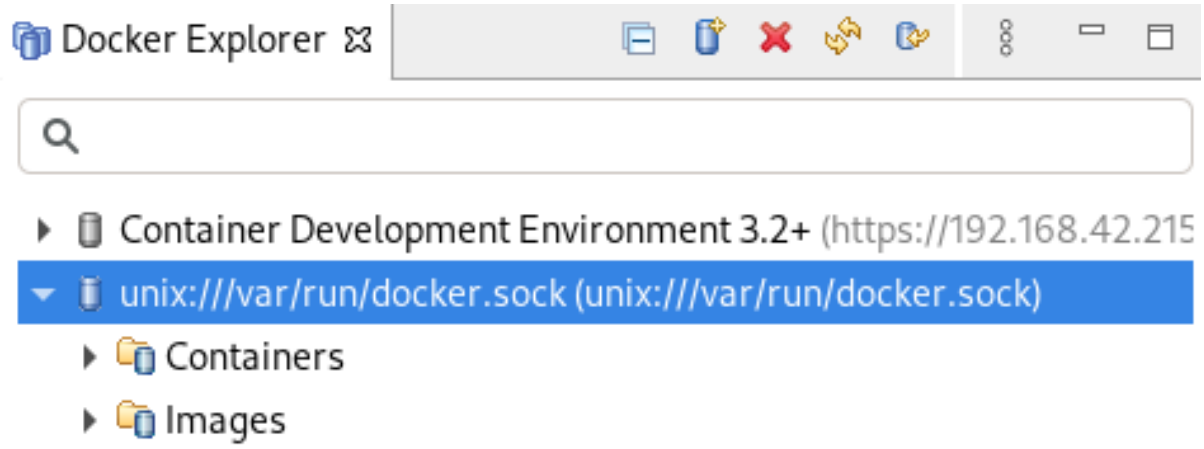
Procedure

1. Start CodeReady Studio.

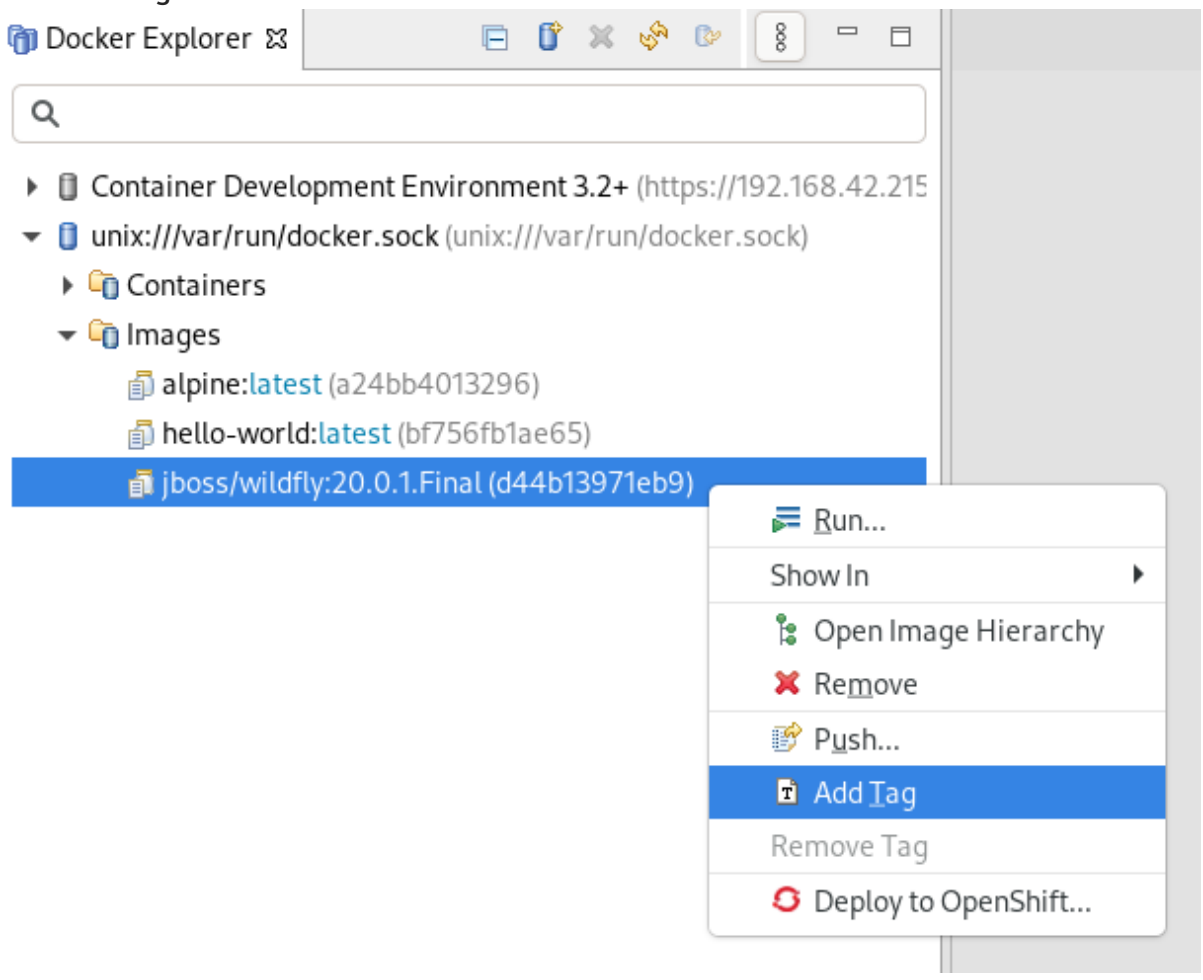
2. Click **Window → Show View → Other**.
   The **Show View** window appears.



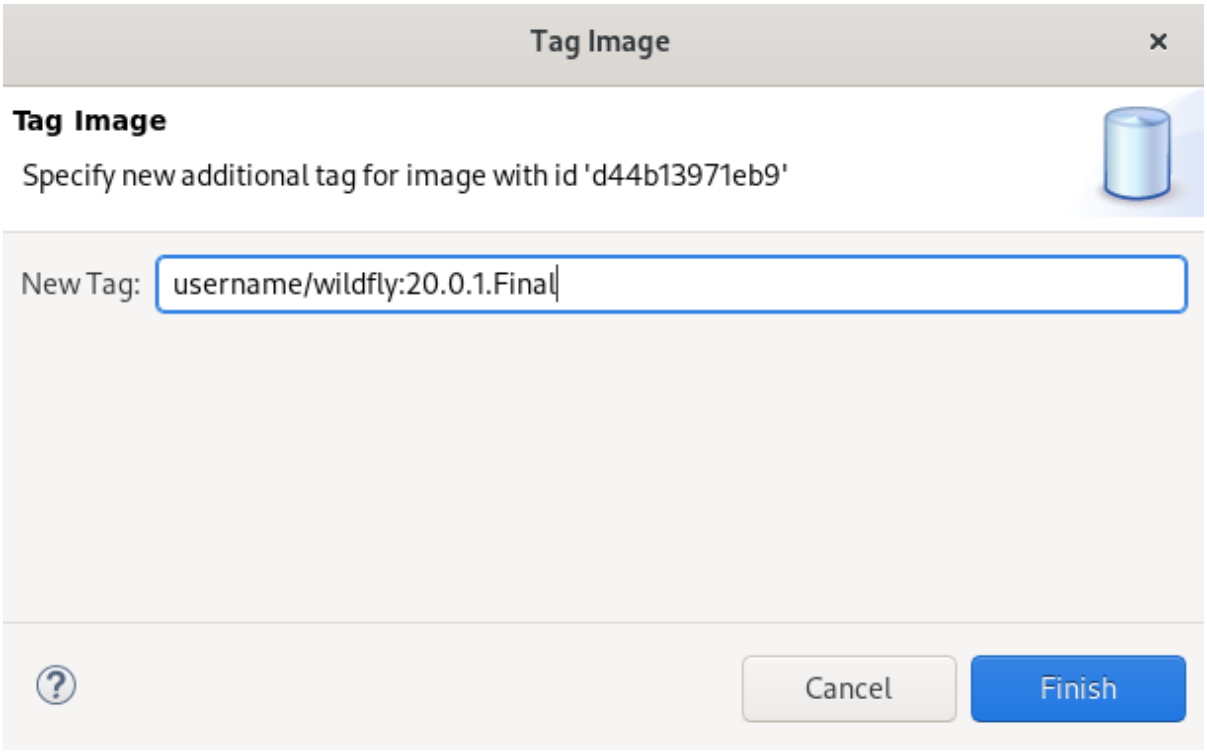
3. Enter **Docker** in the search field.
4. Select **Docker Explorer**.
5. Click **Open**.
   The **Docker Explorer** view appears.



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



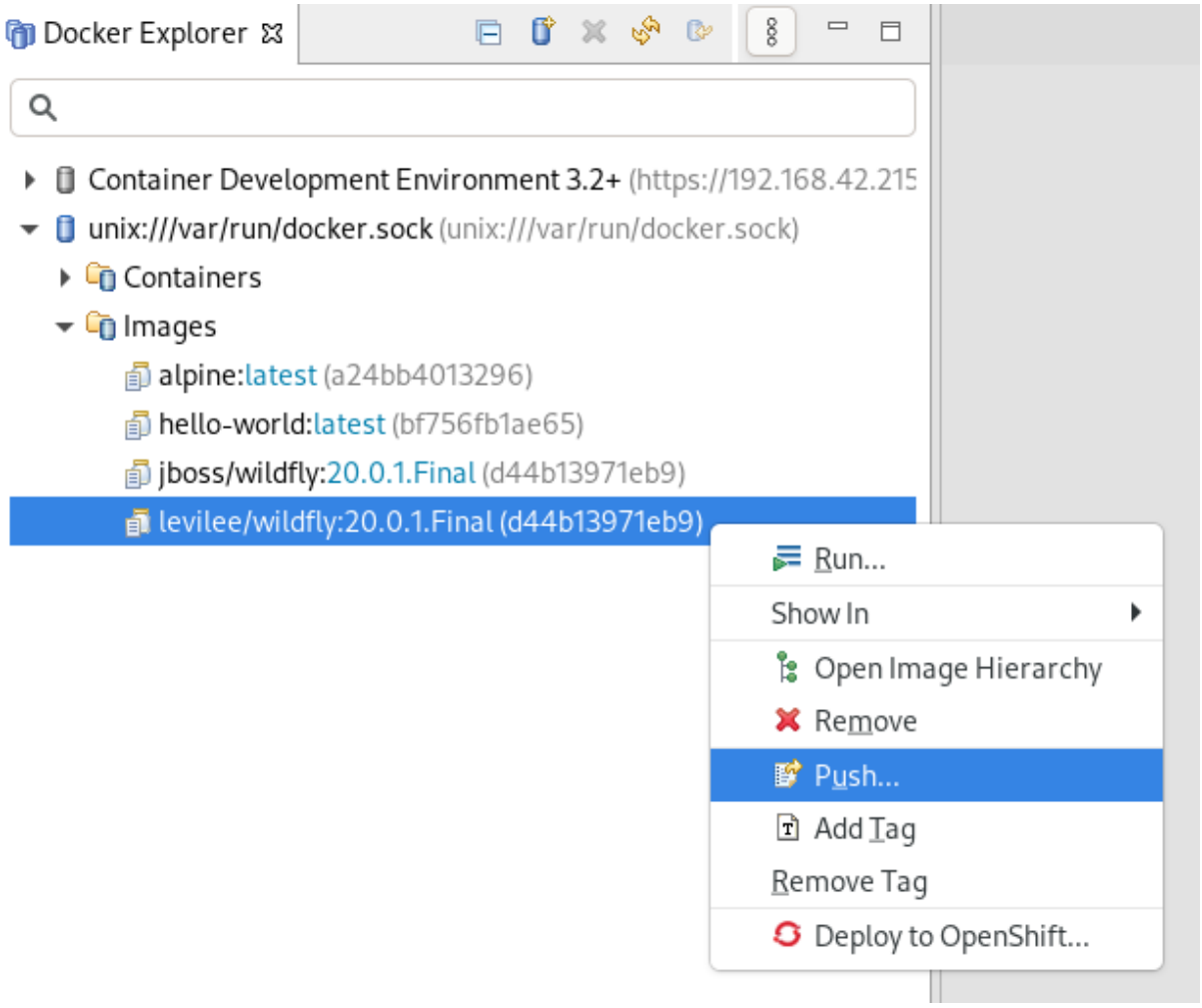
The **Tag Image** window appears.



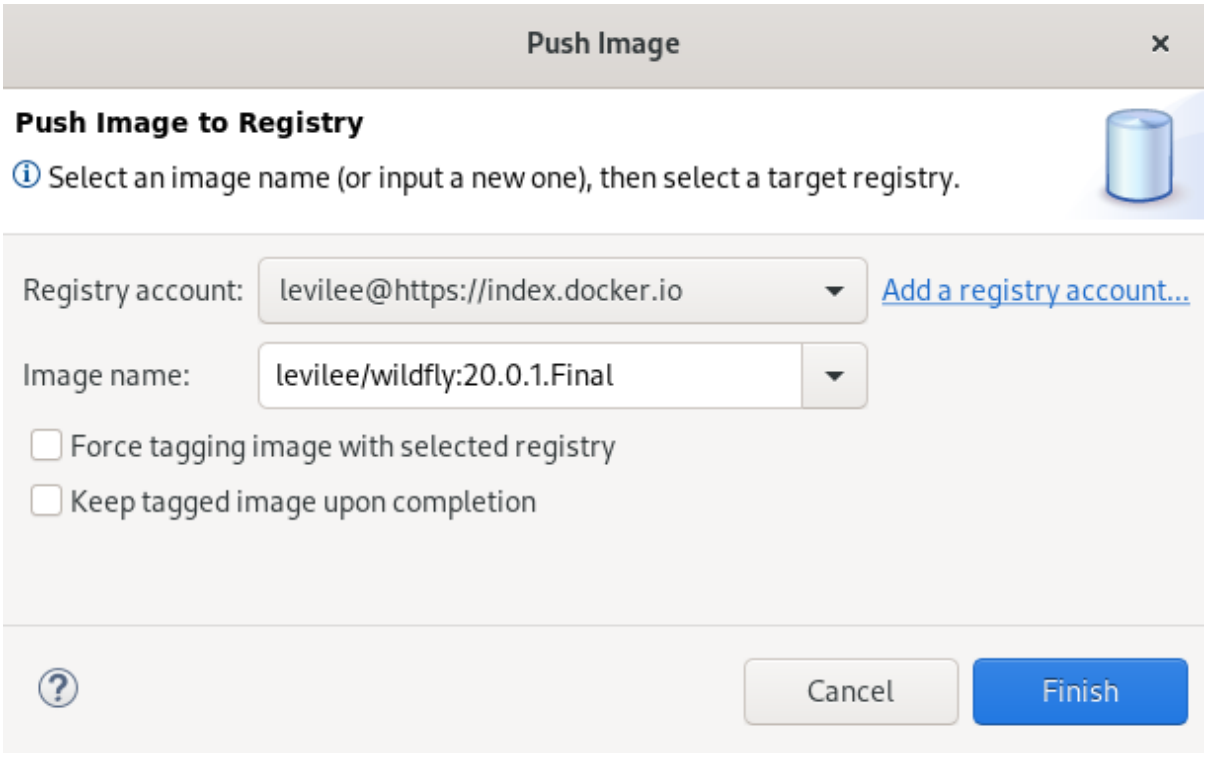
9. Enter your tag in the **New Tag** field.
   The tag should be in the form of **username/image_name:tag_name**, where **username** is your Docker ID on hub.docker.com, **image_name** is the name of your image, and **tag_name** is the version of your image.

10. Click **Finish**.

11. Right-click the **tagged image → Push**.



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



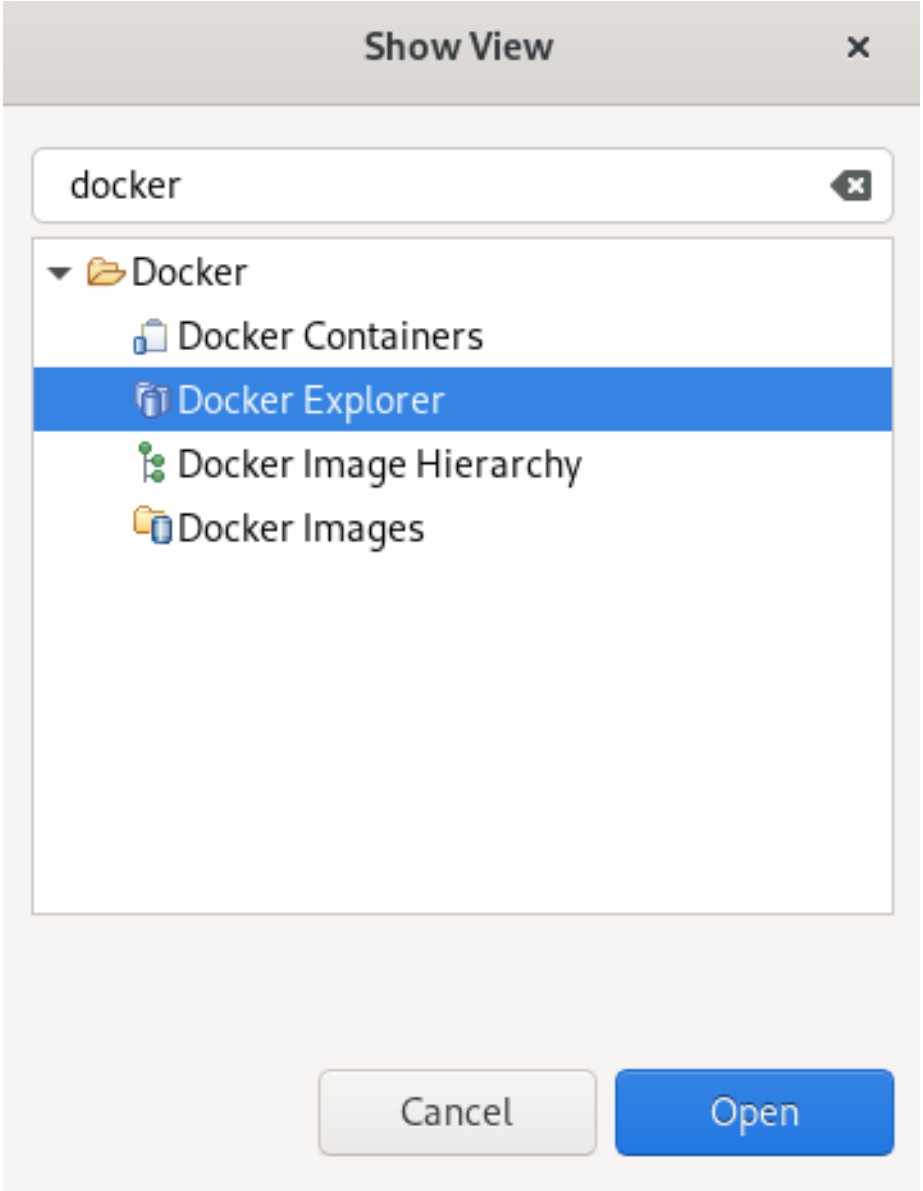
12. Select the **Registry Account** that starts with your Docker ID.
13. Click **Finish**.

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

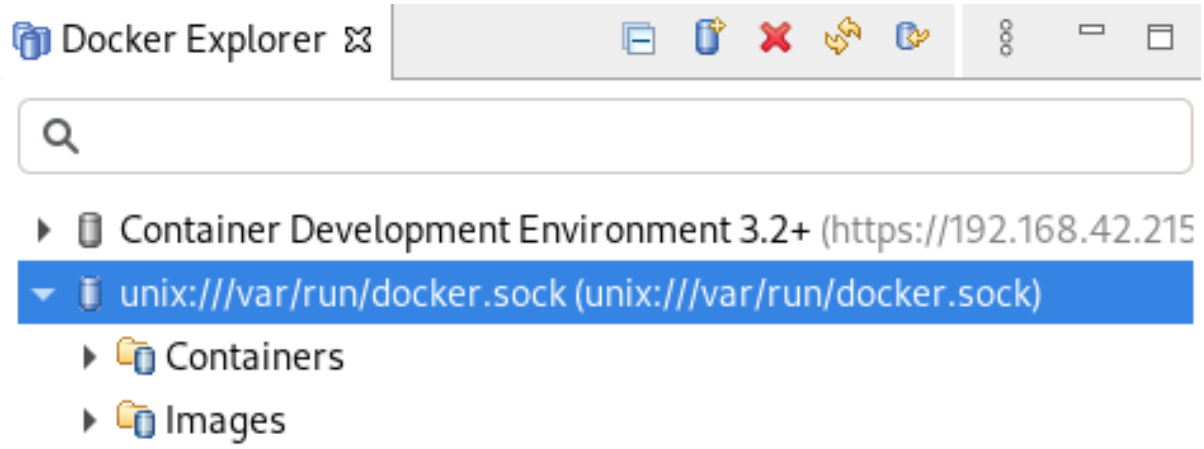
3.2.3. Running Docker images

Procedure

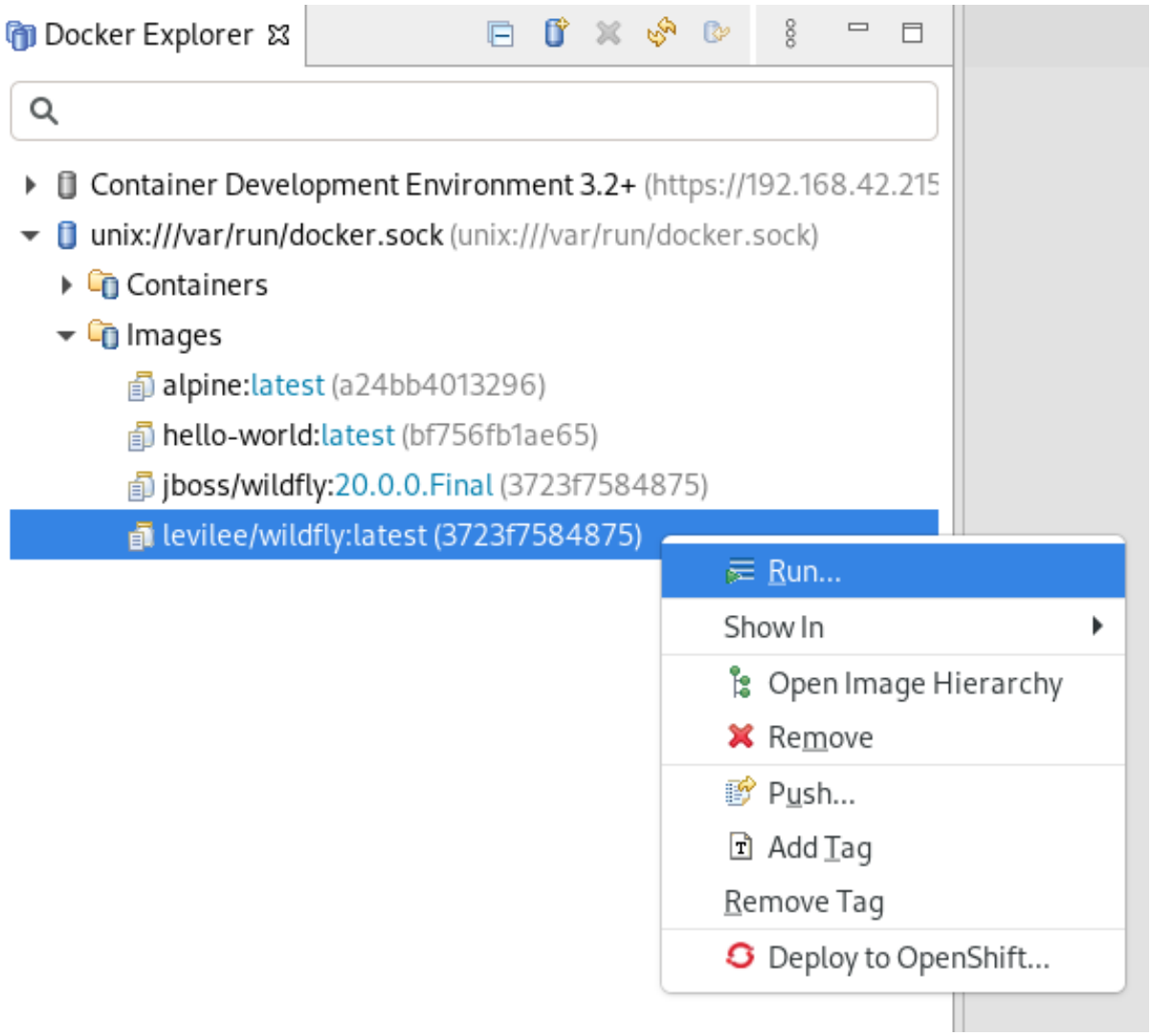
1. Start CodeReady Studio.
2. Click **Window → Show View → Other**.
   The **Show View** window appears.



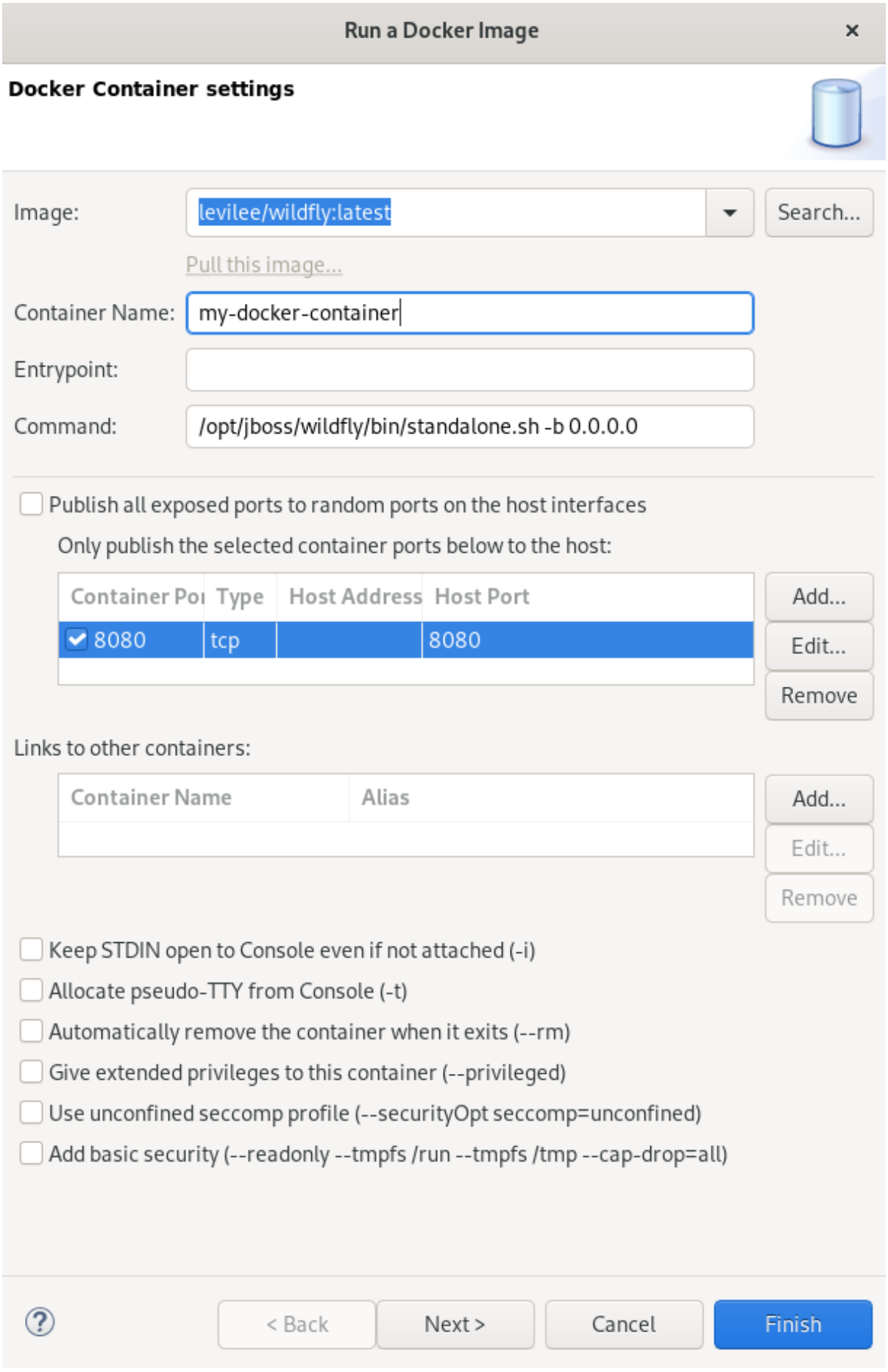
3. Enter **Docker** in the search field.
4. Select **Docker Explorer**.
5. Click **Open**.
   The **Docker Explorer** view appears.



6. Expand **Docker socket → Images**.
7. Right-click an image you want to run.
8. Click **Run**.



The **Docker Container settings** window appears.



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

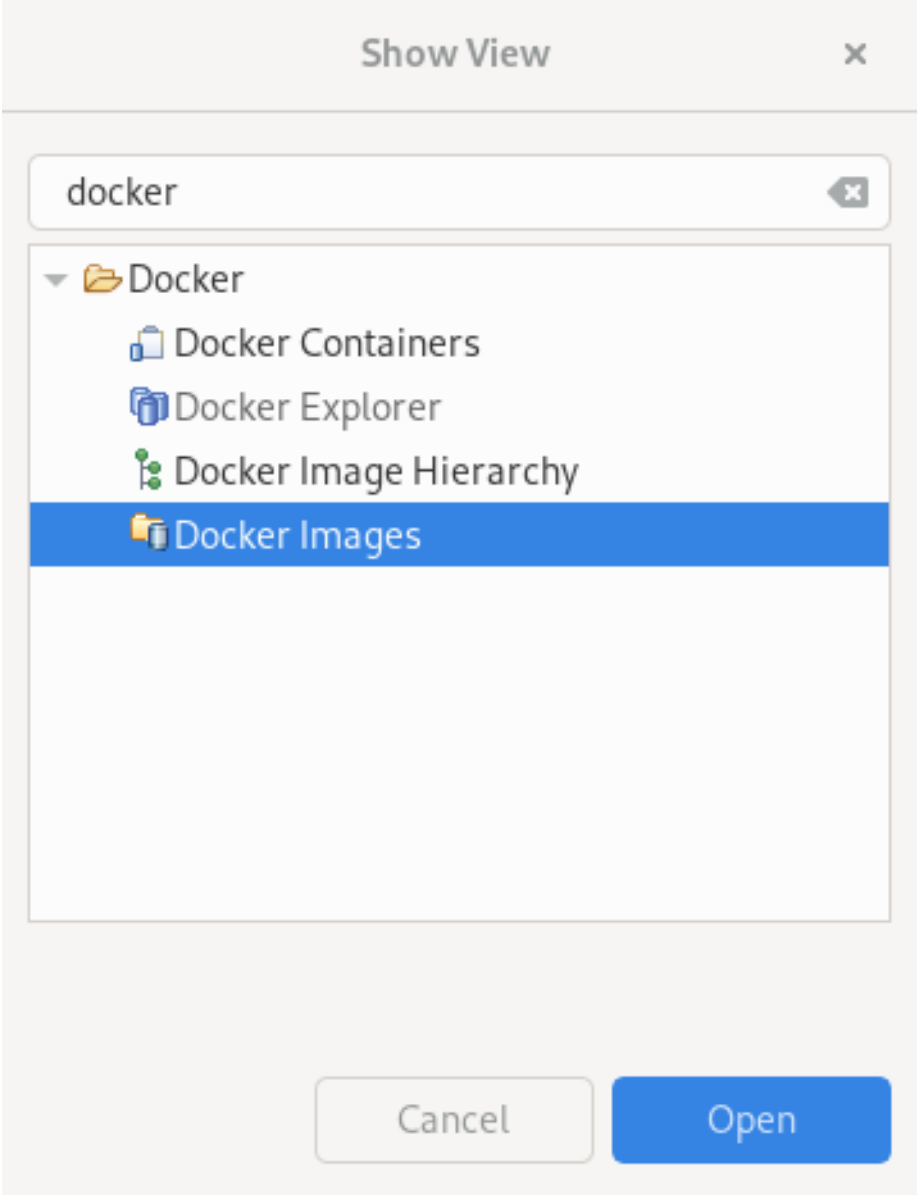
Prerequisites

- You must have a Dockerfile created on your local machine.
  For more information on how to create a Dockerfile, see Creating a Dockerfile.

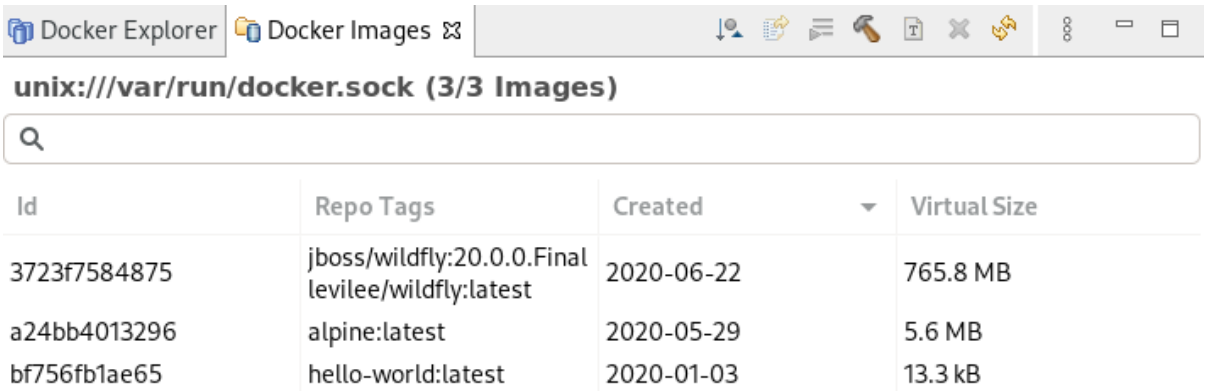
Procedure

1. Start CodeReady Studio.

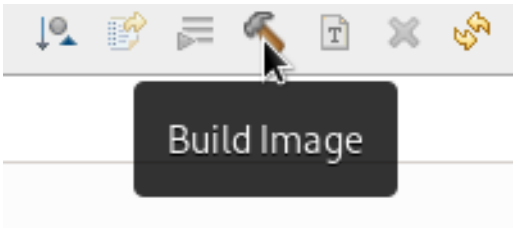
2. Click **Window → Show View → Other**.
   The **Show View** window appears.



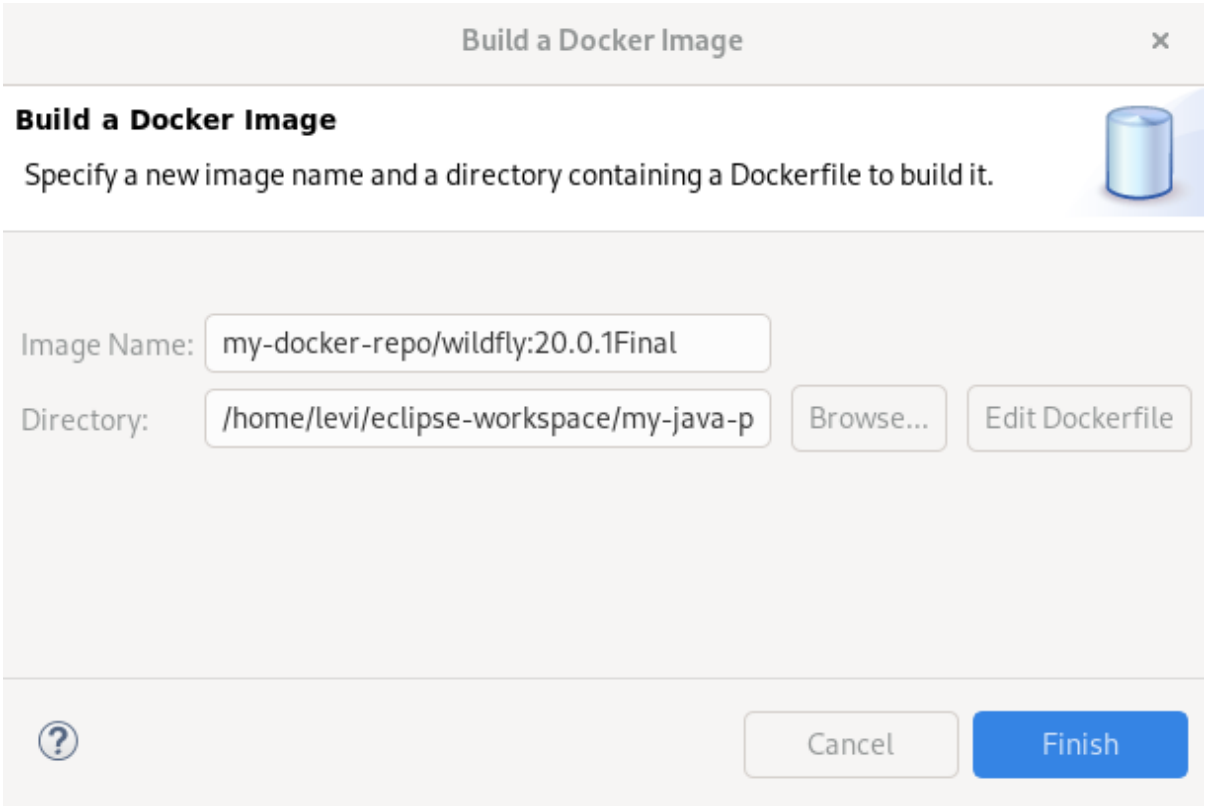
3. Enter **Docker** in the search field.
4. Select **Docker Images**.
5. Click **Open**.
   The **Docker Images** view appears.



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.

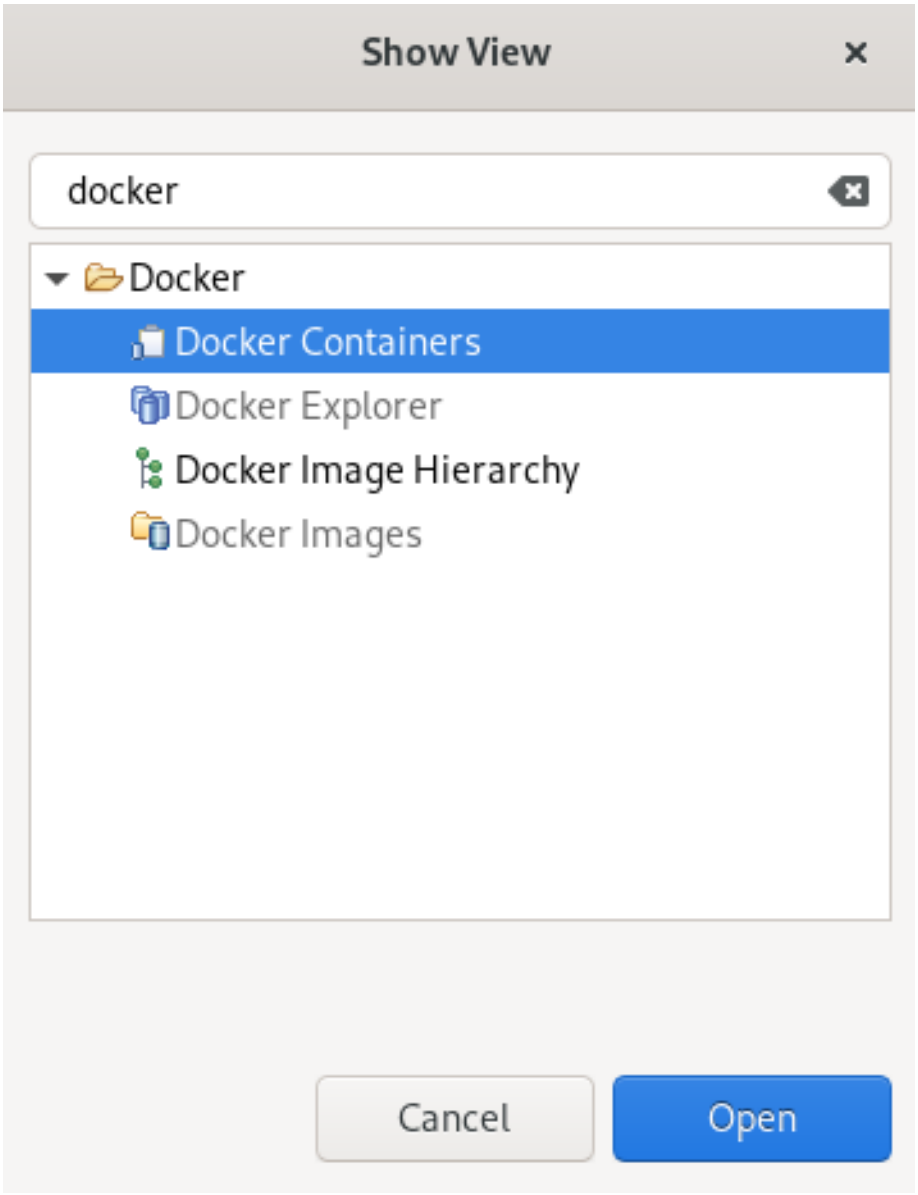
3.3. MANAGING DOCKER CONTAINERS

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.

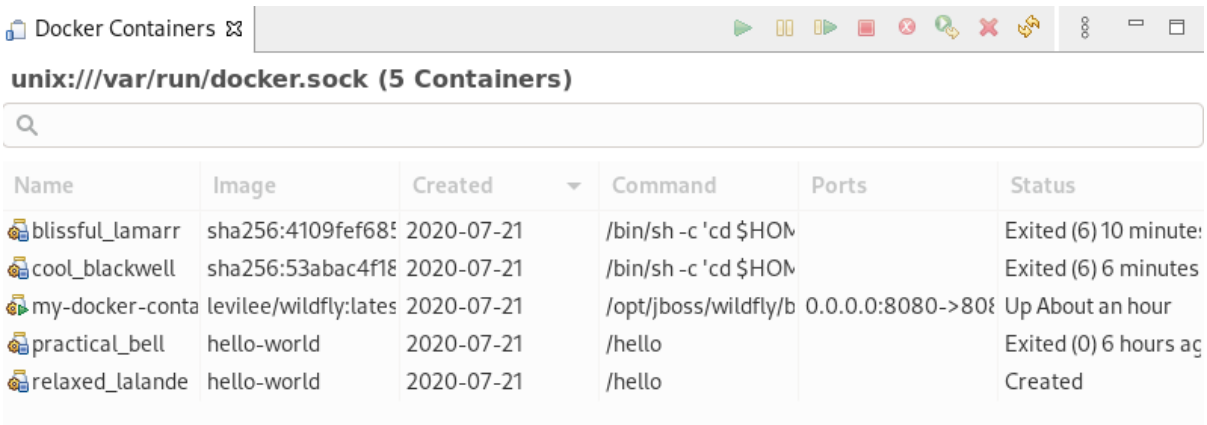
The following section describes how to manage Docker containers in CodeReady Studio.

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.



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