



Red Hat CodeReady Workspaces 2.0

Installation Guide

Installing Red Hat CodeReady Workspaces 2.0

Red Hat CodeReady Workspaces 2.0 Installation Guide

Installing Red Hat CodeReady Workspaces 2.0

Supriya Takkhi

Robert Kratky
rkratky@redhat.com

Michal Maléř
mmaler@redhat.com

Fabrice Flore-Thébault
ffloreth@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

Information for administrators installing Red Hat CodeReady Workspaces.

Table of Contents

CHAPTER 1. INSTALLING CODEREADY WORKSPACES ON OPENSIFT CONTAINER PLATFORM 4	4
1.1. INSTALLING CODEREADY WORKSPACES ON OPENSIFT 4 FROM OPERATORHUB		4
1.1.1. Creating the CodeReady Workspaces project in OpenShift 4 web console		4
Prerequisites		4
Procedure		4
1.1.2. Installing the CodeReady Workspaces Operator in OpenShift 4 web console		5
Prerequisites		5
Procedure		5
1.1.3. Installing CodeReady Workspaces using the CodeReady Workspaces Operator in OpenShift 4 web console		5
Prerequisites		5
Procedure		6
1.1.4. Viewing the state of the CodeReady Workspaces cluster deployment in OpenShift 4 web console		6
Prerequisites		6
Procedure		6
1.1.5. Finding CodeReady Workspaces cluster URL in OpenShift 4 web console		7
Prerequisites		7
Procedure		7
1.1.6. Viewing the state of the CodeReady Workspaces cluster deployment using OpenShift 4 CLI tools		8
Prerequisites		8
Procedure		8
1.1.7. Finding CodeReady Workspaces cluster URL using OpenShift 4 CLI tools		9
Prerequisites		9
Procedure		9
1.1.8. Enabling SSL on OpenShift 4		9
Prerequisites		9
Procedure		9
1.1.9. Logging in to CodeReady Workspaces on OpenShift for the first time using OAuth		10
Prerequisites		10
Procedure		10
1.1.10. Logging in to CodeReady Workspaces on OpenShift for the first time registering as a new user		10
Prerequisites		11
Procedure		11
1.2. INSTALLING CODEREADY WORKSPACES USING CLI MANAGEMENT TOOL		11
1.2.1. Installing the crwctl CLI management tool		11
1.2.2. Installing CodeReady Workspaces using CodeReady Workspaces CLI management tool		11
1.2.2.1. Installing with default settings		12
1.2.2.2. Installing with custom settings		12
CHAPTER 2. INSTALLING CODEREADY WORKSPACES ON OPENSIFT 3 USING THE OPERATOR	13
2.1. PREPARING OPENSIFT 3 FOR INSTALLING CODEREADY WORKSPACES		13
2.2. INSTALLING CODEREADY WORKSPACES ON OPENSIFT 3 USING THE OPERATOR		14
2.3. INSTALLING CODEREADY WORKSPACES ON OPENSIFT 3 USING THE OPERATOR AND SSL		14
CHAPTER 3. INSTALLING CODEREADY WORKSPACES IN A RESTRICTED ENVIROMENT	16
3.1. PREPARING AN IMAGE REGISTRY FOR INSTALLING CODEREADY WORKSPACES IN A RESTRICTED ENVIRONMENT		16
3.2. PREPARING CODEREADY WORKSPACES CUSTOM RESOURCE FOR RESTRICTED ENVIRONMENT		19
3.2.1. Downloading the default CodeReady WorkspacesCluster Custom Resource		19
3.2.2. Customizing the CodeReady WorkspacesCluster Custom Resource for restricted environment		19
3.3. STARTING CODEREADY WORKSPACES INSTALLATION IN A RESTRICTED ENVIRONMENT USING CODEREADY WORKSPACES CLI MANAGEMENT TOOL		20

3.4. STARTING CODEREADY WORKSPACES INSTALLATION IN A RESTRICTED ENVIRONMENT USING OPERATORHUB	20
CHAPTER 4. UPGRADING CODEREADY WORKSPACES	21
4.1. UPGRADING CODEREADY WORKSPACES ON OPENSIFT 4 USING THE WEB CONSOLE	21
4.1.1. Preparing the upgrade of CodeReady Workspaces on OpenShift using the Operator	21
Prerequisites	21
Procedure	21
4.1.2. Creating the CodeReady Workspaces project in OpenShift 4 web console	22
Prerequisites	22
Procedure	22
4.1.3. Installing the CodeReady Workspaces Operator in OpenShift 4 web console	22
Prerequisites	22
Procedure	22
4.1.4. Viewing the state of the CodeReady Workspaces cluster deployment in OpenShift 4 web console	23
Prerequisites	23
Procedure	23
4.1.5. Finishing the upgrade of CodeReady Workspaces on OpenShift using the Operator	24
4.2. UPGRADING CODEREADY WORKSPACES ON OPENSIFT 4 USING THE CLI MANAGEMENT TOOL	25
4.3. UPGRADING CODEREADY WORKSPACES ON OPENSIFT 3 USING THE CLI MANAGEMENT TOOL	25
4.4. KNOWN ISSUES	26
4.4.1. Upgrading from CodeReady Workspaces 1.2 to CodeReady Workspaces 2.0 using the Operator	26
4.4.2. Updating a CodeReady Workspaces installation using the Operator	26
CHAPTER 5. ADVANCED CONFIGURATION OPTIONS	28
5.1. CODEREADY WORKSPACES CONFIGMAPS AND THEIR BEHAVIOR	28
5.1.1. CodeReady Workspaces installed using an Operator	28
5.2. CONFIGURING NAMESPACE STRATEGIES	29
5.2.1. One namespace per workspace strategy	29
5.2.2. One namespace for all workspaces strategy	29
5.2.3. One namespace per user strategy	30
5.2.4. Allowing user-defined workspace namespaces	30
5.3. DEPLOYING CODEREADY WORKSPACES WITH SUPPORT FOR GIT REPOSITORIES WITH SELF-SIGNED CERTIFICATES	31
Configuring support for self-signed Git repositories on OpenShift	31
5.4. CODEREADY WORKSPACES CONFIGMAPS FIELDS REFERENCE	31
5.4.1. server settings related to the CodeReady Workspaces server	31
5.4.2. database configuration settings related to the database used by CodeReady Workspaces	35
5.4.3. auth configuration settings related to authentication used by CodeReady Workspaces installation	36
5.4.4. storage configuration settings related to persistent storage used by CodeReady Workspaces	37
5.4.5. k8s configuration settings specific to CodeReady Workspaces installations on OpenShift	38
5.4.6. installation defines the observed state of CodeReady Workspaces installation	38
5.4.7. Limits for workspaces	39
5.4.8. Limits for the workspaces of an user	39
5.4.9. Limits for for the workspaces of an organization	40
CHAPTER 6. UNINSTALLING CODEREADY WORKSPACES	41
6.1. UNINSTALLING CODEREADY WORKSPACES AFTER OPERATORHUB INSTALLATION	41
6.1.1. Uninstalling CodeReady Workspaces using the OpenShift web console	41
6.1.2. Uninstalling CodeReady Workspaces using oc commands	42
6.2. UNINSTALLING CODEREADY WORKSPACES AFTER CRWCTL INSTALLATION	43

CHAPTER 1. INSTALLING CODEREADY WORKSPACES ON OPENSIFT CONTAINER PLATFORM 4

1.1. INSTALLING CODEREADY WORKSPACES ON OPENSIFT 4 FROM OPERATORHUB

On OpenShift, Red Hat CodeReady Workspaces can be installed using the OperatorHub Catalog present in the OpenShift web console. Following steps are described:

- [Section 1.1.1, "Creating the CodeReady Workspaces project in OpenShift 4 web console"](#) .
- [Section 1.1.2, "Installing the CodeReady Workspaces Operator in OpenShift 4 web console"](#) .
- [Section 1.1.3, "Installing CodeReady Workspaces using the CodeReady Workspaces Operator in OpenShift 4 web console"](#).
- [Section 1.1.4, "Viewing the state of the CodeReady Workspaces cluster deployment in OpenShift 4 web console"](#).
- [Section 1.1.6, "Viewing the state of the CodeReady Workspaces cluster deployment using OpenShift 4 CLI tools"](#).
- [Section 1.1.5, "Finding CodeReady Workspaces cluster URL in OpenShift 4 web console"](#) .
- [Section 1.1.7, "Finding CodeReady Workspaces cluster URL using OpenShift 4 CLI tools"](#) .
- [Section 1.1.8, "Enabling SSL on OpenShift 4"](#) .
- [Section 1.1.9, "Logging in to CodeReady Workspaces on OpenShift for the first time using OAuth"](#)
- [Section 1.1.10, "Logging in to CodeReady Workspaces on OpenShift for the first time registering as a new user"](#).

1.1.1. Creating the CodeReady Workspaces project in OpenShift 4 web console

This section describes how to create the **CodeReady Workspaces** project in OpenShift 4 web console.

Prerequisites

- An administrator account on a running instance of OpenShift 4.

Procedure

1. Open the OpenShift web console.
2. In the left panel, navigate to **Projects**.
3. Click the **Create Project** button.
4. Enter the project details:
 - In the **Name** field, type **workspaces**.
 - In the **Display Name** field, type **Red Hat CodeReady Workspaces**.

5. Click the **Create** button.

1.1.2. Installing the CodeReady Workspaces Operator in OpenShift 4 web console

This section describes how to install the CodeReady Workspaces Operator in OpenShift 4 web console.

Prerequisites

- An administrator account on a running instance of OpenShift 4.
- Administrative rights on an existing project named **workspaces** on this instance of OpenShift 4. See [Section 1.1.1, “Creating the CodeReady Workspaces project in OpenShift 4 web console”](#) .
- The Red Hat CodeReady Workspaces 1.2 Operator is not installed.

Procedure

1. Open the OpenShift web console.
2. In the left panel, navigate to the **Operators** → **OperatorHub** section.
3. In the **Search by keyword** field, type **Red Hat CodeReady Workspaces**.
4. Click on the **Red Hat CodeReady Workspaces** tile.
5. Click the **Install** button in the **Red Hat CodeReady Workspaces** pop-up window.
6. In the **A specific namespace on the cluster** field, in the cluster drop-down list, select the namespace into which the previous version of the CodeReady Workspaces Operator was installed.
7. Click the **Subscribe** button.
8. In the left panel, navigate to the **Operators** → **Installed Operators** section.
9. Red Hat CodeReady Workspaces is displayed as an installed Operator having the **InstallSucceeded** status.
10. Click on the **Red Hat CodeReady Workspaces** name in the list of installed operators.
11. Navigate to the **Overview** tab.
12. In the **Conditions** sections at the bottom of the page, wait for this message: **install strategy completed with no errors**.
13. Navigate to the **Events** tab.
14. Wait for this message: **install strategy completed with no errors**.

1.1.3. Installing CodeReady Workspaces using the CodeReady Workspaces Operator in OpenShift 4 web console

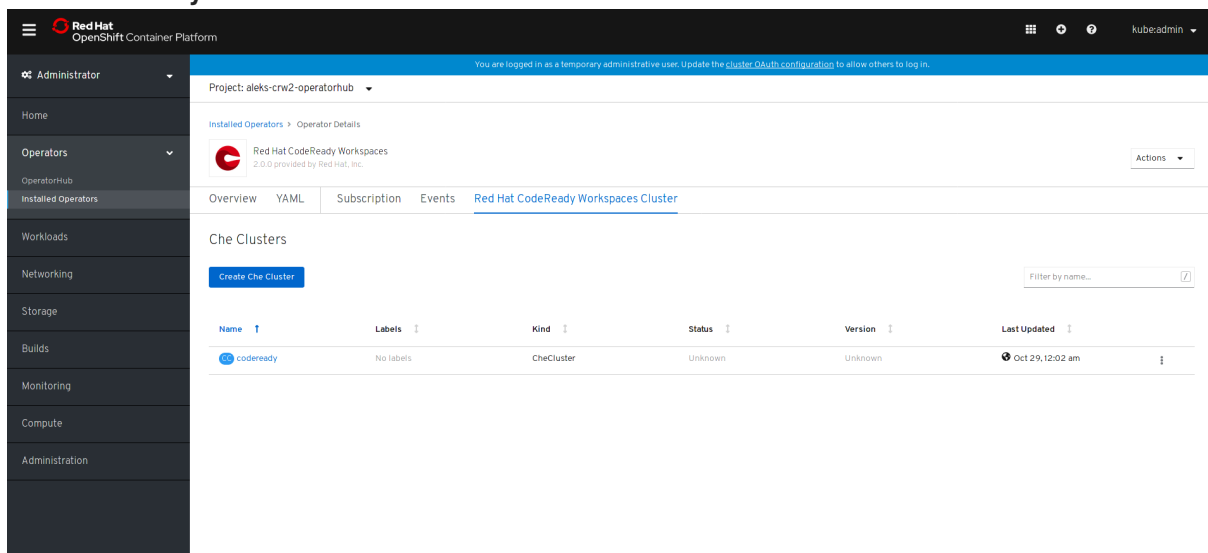
This section describes how to install CodeReady Workspaces using the CodeReady Workspaces Operator in OpenShift 4 web console.

Prerequisites

- An administrator account on a running instance of OpenShift 4.
- At least one OAuth user provisioned on this instance of OpenShift 4.
- The CodeReady Workspaces Operator is installed on this instance of OpenShift 4. See [Section 1.1.2, “Installing the CodeReady Workspaces Operator in OpenShift 4 web console”](#)

Procedure

1. Open the OpenShift web console.
2. Navigate to the **Operators** → **Installed Operators** section.
3. Click **Red Hat CodeReady Workspaces** in the list of installed operators.
4. Click the **Create Instance** link in **Provided APIs** section.
5. The **Create CodeReady Workspaces Cluster** page is displayed.
6. Leave the default values as they are.
7. Click the **Create** button in the bottom-left corner of the window.
8. The **codeready** cluster is created.



1.1.4. Viewing the state of the CodeReady Workspaces cluster deployment in OpenShift 4 web console

This section describes how to view the state of the CodeReady Workspaces cluster deployment in OpenShift 4 web console.

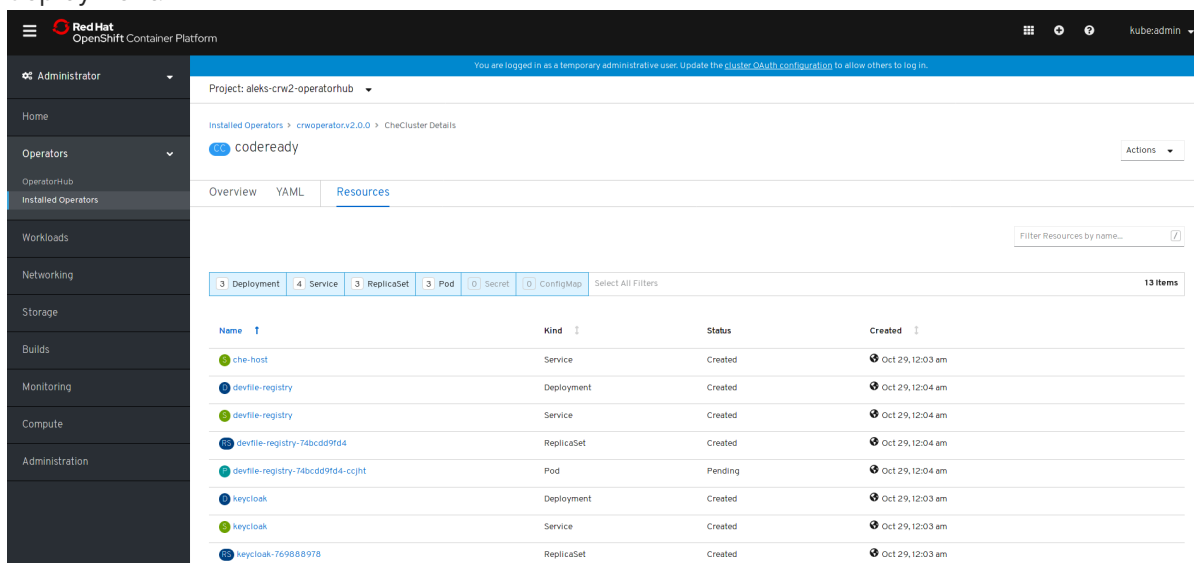
Prerequisites

- An administrator account on a running instance of OpenShift 4.
- A CodeReady Workspaces cluster is deployed on this instance of OpenShift 4.

Procedure

1. Open the OpenShift web console.
2. Navigate to the **Operators** → **Installed Operators** section.

3. Click **Red Hat CodeReady Workspaces** in the list of installed operators.
4. Navigate to the **Red Hat CodeReady Workspaces Cluster** tab.
5. Click on the **eclipse-che** cluster that is listed in the table.
6. Navigate to the **Overview** tab.
7. Watch the content of the **Message** field. The field contain error messages, if any. The expected content is **None**.
8. Navigate to the **Resources** tab.
9. The screen displays the state of the resources assigned to the CodeReady Workspaces cluster deployment.



1.1.5. Finding CodeReady Workspaces cluster URL in OpenShift 4 web console

This section describes how to find the CodeReady Workspaces cluster URL in OpenShift 4 web console.

Prerequisites

- A running Red Hat CodeReady Workspaces cluster.
 - See [Section 1.1.4, “Viewing the state of the CodeReady Workspaces cluster deployment in OpenShift 4 web console”](#).
 - See [Section 1.1.6, “Viewing the state of the CodeReady Workspaces cluster deployment using OpenShift 4 CLI tools”](#).

Procedure

1. Open the OpenShift web console.
2. In the left panel, navigate to the **Operators** → **Installed Operators** section.
3. Click on the **Red Hat CodeReady Workspaces** Operator tile.
4. Click on **eclipse-che** in the table.
5. Navigate to the **Overview** tab.

6. Read the content of the **Red Hat CodeReady Workspaces URL** field.

1.1.6. Viewing the state of the CodeReady Workspaces cluster deployment using OpenShift 4 CLI tools

This section describes how to view the state of the CodeReady Workspaces cluster deployment using OpenShift 4 CLI tools.

Prerequisites

- An installation of an Red Hat CodeReady Workspaces cluster. See [Section 1.1.3, "Installing CodeReady Workspaces using the CodeReady Workspaces Operator in OpenShift 4 web console"](#).

Procedure

1. Run the following commands to select the **workspaces** project:

```
$ oc project workspaces
```

2. Run the following commands to get the name and status of the pods running in the selected project:

```
$ oc get pods
```

3. Check that the status of all pods is **Running**.

```
NAME                                READY   STATUS    RESTARTS   AGE
che-8495f4946b-jrzdc                0/1     Running  0           86s
che-operator-578765d954-99szg       1/1     Running  0           42m
keycloak-74fbfb9654-g9vp5           1/1     Running  0           4m32s
postgres-5d579c6847-w6wx5          1/1     Running  0           5m14s
```

4. Run the following command:

```
$ oc logs --tail=10 -f oc get pods -o name | grep operator
```

5. This is an example output of the command:

```
time="2019-07-12T09:48:29Z" level=info msg="Exec successfully completed"
time="2019-07-12T09:48:29Z" level=info msg="Updating eclipse-che CR with status:
provisioned with OpenShift identity provider: true"
time="2019-07-12T09:48:29Z" level=info msg="Custom resource eclipse-che updated"
time="2019-07-12T09:48:29Z" level=info msg="Creating a new object: ConfigMap, name:
che"
time="2019-07-12T09:48:29Z" level=info msg="Creating a new object: ConfigMap, name:
custom"
time="2019-07-12T09:48:29Z" level=info msg="Creating a new object: Deployment, name:
che"
time="2019-07-12T09:48:30Z" level=info msg="Updating eclipse-che CR with status:
CodeReady Workspaces API: Unavailable"
time="2019-07-12T09:48:30Z" level=info msg="Custom resource eclipse-che updated"
time="2019-07-12T09:48:30Z" level=info msg="Waiting for deployment che. Default timeout:
420 seconds"
```

1.1.7. Finding CodeReady Workspaces cluster URL using OpenShift 4 CLI tools

This section describes how to CodeReady Workspaces cluster URL using OpenShift 4 CLI tool.

Prerequisites

- View the logs of the CodeReady Workspaces Operator. See [Section 1.1.6, “Viewing the state of the CodeReady Workspaces cluster deployment using OpenShift 4 CLI tools”](#).

Procedure

- Wait for the logs to show the message **Red Hat CodeReady Workspaces is now available at:** followed by CodeReady Workspaces URL:

```
time="2019-07-12T09:50:13Z" level=info msg="Updating eclipse-che CR with Red Hat
CodeReady Workspaces server URL: http://che-che.apps.cluster-fre-f0a2.fre-
f0a2.openshiftworkshop.com"
time="2019-07-12T09:50:13Z" level=info msg="Custom resource eclipse-che updated"
time="2019-07-12T09:50:13Z" level=info msg="Red Hat CodeReady Workspaces is now
available at: http://che-che.apps.cluster-fre-f0a2.fre-f0a2.openshiftworkshop.com"
time="2019-07-12T09:50:13Z" level=info msg="Updating eclipse-che CR with version: 7.0.0-
RC-2.0"
```

1.1.8. Enabling SSL on OpenShift 4

Prerequisites

- A running Red Hat CodeReady Workspaces cluster.
 - See [Section 1.1.4, “Viewing the state of the CodeReady Workspaces cluster deployment in OpenShift 4 web console”](#).
 - See [Section 1.1.6, “Viewing the state of the CodeReady Workspaces cluster deployment using OpenShift 4 CLI tools”](#).

Procedure

1. Open the OpenShift web console.
2. In the left panel, navigate to the **Operators** → **Installed Operators** section.
3. Click on the **Red Hat CodeReady Workspaces** Operator tile.
4. Click on **eclipse-che** in the table.
5. Navigate to the **Overview** tab.
6. Toggle the **TLS MODE** switch to **True**.
7. Click **Confirm change**.



8. Navigate to the **Resources** tab.

9. Wait that the pods are restarted.
10. Navigate to the **Overview** tab.
11. Click the **Red Hat CodeReady Workspaces URL** link.
12. Notice that the link is redirected to HTTPS.
13. The browser displays the Red Hat CodeReady Workspaces **Dashboard** using a valid *Let's Encrypt* certificate.

1.1.9. Logging in to CodeReady Workspaces on OpenShift for the first time using OAuth

This section describes how to log in to CodeReady Workspaces on OpenShift for the first time using OAuth.

Prerequisites

- A running Red Hat CodeReady Workspaces cluster.
 - See [Section 1.1.4, "Viewing the state of the CodeReady Workspaces cluster deployment in OpenShift 4 web console"](#).
 - See [Section 1.1.6, "Viewing the state of the CodeReady Workspaces cluster deployment using OpenShift 4 CLI tools"](#).
- Know the Red Hat CodeReady Workspaces URL.
 - See [Section 1.1.5, "Finding CodeReady Workspaces cluster URL in OpenShift 4 web console"](#).
 - See [Section 1.1.7, "Finding CodeReady Workspaces cluster URL using OpenShift 4 CLI tools"](#).

Procedure

1. Navigate to the **Red Hat CodeReady Workspaces URL** to display the Red Hat CodeReady Workspaces login page.
2. Choose the **OpenShift OAuth** option.
3. The **Authorize Access** page is displayed.
4. Click on the **Allow selected permissions** button.
5. Update the account information: fill in the **Username**, **Email**, **First name** and **Last name** field and click on the **Submit** button.
6. The browser displays the Red Hat CodeReady Workspaces **Dashboard**.

1.1.10. Logging in to CodeReady Workspaces on OpenShift for the first time registering as a new user

This section describes how to log in to CodeReady Workspaces on OpenShift for the first time registering as a new user.

Prerequisites

- A running Red Hat CodeReady Workspaces cluster.
 - See [Section 1.1.4, “Viewing the state of the CodeReady Workspaces cluster deployment in OpenShift 4 web console”](#).
 - See [Section 1.1.6, “Viewing the state of the CodeReady Workspaces cluster deployment using OpenShift 4 CLI tools”](#).
- Know the Red Hat CodeReady Workspaces URL.
 - See [Section 1.1.5, “Finding CodeReady Workspaces cluster URL in OpenShift 4 web console”](#).
 - See [Section 1.1.7, “Finding CodeReady Workspaces cluster URL using OpenShift 4 CLI tools”](#).

Procedure

1. Navigate to the **Red Hat CodeReady Workspaces URL** to display the Red Hat CodeReady Workspaces login page.
2. Choose the **Register as a new user** option.
3. Update the account information: fill in the **Username, Email, First name** and **Last name** field and click on the **Submit** button.
4. The browser displays the Red Hat CodeReady Workspaces **Dashboard**.

1.2. INSTALLING CODEREADY WORKSPACES USING CLI MANAGEMENT TOOL

1.2.1. Installing the crwctl CLI management tool

This section describes how to install crwctl, the CodeReady Workspaces CLI management tool.

Procedure

1. Navigate to <https://developers.redhat.com/products/codeready-workspaces/download>.
2. Download the CodeReady Workspaces CLI management tool archive for version 2.0.
3. Extract the archive.
4. Place the extracted binary in your **\$PATH**.

1.2.2. Installing CodeReady Workspaces using CodeReady Workspaces CLI management tool

This sections describes how to install CodeReady Workspaces using the CodeReady Workspaces CLI management tool.

Prerequisites

- CodeReady Workspaces CLI management tool is installed.
- OpenShift Container Platform 4 CLI is installed.
- Access to an OpenShift Container Platform instance

1.2.2.1. Installing with default settings

Procedure

1. Log in to OpenShift Container Platform 4:

```
$ oc login ${OPENSIFT_API_URL} -u ${OPENSIFT_USERNAME} -p  
${OPENSIFT_PASSWORD}
```

2. Run this command to install Red Hat CodeReady Workspaces with defaults settings:

```
$ crwctl server:start
```



NOTE

crwctl default namespace is **workspaces**. If you use a namespace with a different name, run the command with the **--chnamespace=*your namespace*** flag, for example:

```
$ {prod-cli} server:start --chnamespace=codeready-workspaces
```

1.2.2.2. Installing with custom settings

Procedure

To override specific settings of the red-hat-codeready-workspaces installation, provide a dedicated custom resource when running the above **crwctl** command:

1. Download [the default custom resource YAML file](#).
2. Name the downloaded custom resource **org_v1_che_cr.yaml**, and copy it into the current directory.
3. Modify the **org_v1_che_cr.yaml** file to override or add any field.
4. Run the installation using the **org_v1_che_cr.yaml** file to override the CodeReady Workspaces CLI management tool defaults:

```
$ crwctl server:start --che-operator-cr-yaml=org_v1_che_cr.yaml
```



NOTE

Some basic installation settings can be overridden in a simpler way by using additional **crwctl** arguments. To display the list of available arguments:

```
$ crwctl server:start --help
```


CHAPTER 2. INSTALLING CODEREADY WORKSPACES ON OPENSIFT 3 USING THE OPERATOR

This chapter describes how to install CodeReady Workspaces on OpenShift 3, with the CLI management tool, using the Operator method.

2.1. PREPARING OPENSIFT 3 FOR INSTALLING CODEREADY WORKSPACES

Prerequisites

- A running instance of OpenShift 3.11.
- Administrator rights on this OpenShift 3 instance.
- The **oc** OpenShift 3.11 CLI management tool is installed and configured. See [Installing the OpenShift 3.11 CLI](#).
- The **crwctl** CLI management tool is installed. See [the CodeReady Workspaces 2.0 Installation Guide](#).

Procedure

1. Log in to OpenShift. See [Basic Setup and Login](#).

```
$ oc login
```

2. Run the following command to verify that the version of the **oc** OpenShift CLI management tool is 3.11:

```
$ oc version
oc v3.11.0+0cbc58b
```

3. Run the following commands to create a dummy project to find the URL that this OpenShift instance is using to deploy applications.

```
$ oc new-project hello-world
$ oc new-app centos/httpd-24-centos7~https://github.com/openshift/httpd-ex
$ oc expose svc/httpd-ex
$ oc get route httpd-ex
NAME          HOST/PORT                                     PATH    SERVICES PORT
TERMINATION  WILDCARD
httpd-ex     httpd-ex-hello-world.apps.rhpd311.openshift.opentlc.com httpd-ex 8080-tcp
None
```

4. Extract the domain from **httpd-ex-hello-world.apps.rhpd311.openshift.opentlc.com**. It is the part after the first name: **apps.rhpd311.openshift.opentlc.com**. Remember this URL as **<OPENSIFT_APPS_URL>**.

5. Remove the dummy project:

```
$ oc delete project hello-world
```

2.2. INSTALLING CODEREADY WORKSPACES ON OPENSIFT 3 USING THE OPERATOR

This section describes how to install CodeReady Workspaces on OpenShift 3 with the CLI management tool, using the Operator method.

Prerequisites

- A running instance of OpenShift 3.11.
- Administrator rights on this OpenShift 3 instance.
- The **crwctl** CLI management tool is installed. See [the CodeReady Workspaces 2.0 Installation Guide](#).
- The URL for applications on this OpenShift instance is known as **<OPENSIFT_APPS_URL>**. See [Section 2.1, "Preparing OpenShift 3 for installing CodeReady Workspaces"](#).

Procedure

1. Run the following command to create the CodeReady Workspaces instance:

```
$ crwctl server:start --platform=openshift --installer=operator \  
--domain=<OPENSIFT_APPS_URL>
```

Verification steps

1. The output of the previous command ends with:

```
Command server:start has completed successfully.
```

2. Navigate to the CodeReady Workspaces cluster instance: **http://che-che.<OPENSIFT_APPS_URL>**.

2.3. INSTALLING CODEREADY WORKSPACES ON OPENSIFT 3 USING THE OPERATOR AND SSL

This section describes how to install CodeReady Workspaces on OpenShift 3 with the CLI management tool, using the Operator method and the SSL option.

Prerequisites

- A running instance of OpenShift 3.11.
- Administrator rights on this OpenShift 3 instance.
- The **crwctl** CLI management tool is installed. See [the CodeReady Workspaces 2.0 Installation Guide](#).
- The URL for applications on this OpenShift instance is known as **<OPENSIFT_APPS_URL>**. See [Section 2.1, "Preparing OpenShift 3 for installing CodeReady Workspaces"](#).

Procedure

1. Run the following command to create the CodeReady Workspaces instance:

```
$ crwctl server:start --platform=openshift --installer=operator --domain=  
<OPENSIFT_APPS_URL> --tls
```

Verification steps

1. The output of the previous command ends with:

```
Command server:start has completed successfully.
```

2. Navigate to the CodeReady Workspaces cluster instance. The domain is now prefixed with HTTPS and using *Let's Encrypt* ACME certificates: **http(s)://codeready-workspaces.<OPENSIFT_APPS_URL>**

CHAPTER 3. INSTALLING CODEREADY WORKSPACES IN A RESTRICTED ENVIROMENT

By default, Red Hat CodeReady Workspaces workspaces reference various external resources, mainly container images available in public registries.

To deploy CodeReady Workspaces in an environment where these external resources are not available (for example, on a cluster that is not exposed to the public Internet):

1. Identify the image registry used by the OpenShift cluster, and ensure you can push to it.
2. Push all the images needed for running CodeReady Workspaces to this registry.
3. Configure CodeReady Workspaces to use the images that have been pushed to the registry.
4. Proceed to the CodeReady Workspaces installation.



NOTE

The procedure for installing CodeReady Workspaces in restricted environments applies to both OpenShift 3.11 and 4.x.

Notes on network connectivity in restricted environments

Restricted network environments range from a private subnet in a cloud provider to a separate network owned by a company, disconnected from the public Internet. Regardless of the network configuration, CodeReady Workspaces works **provided that the Ingress and Routes that are created for CodeReady Workspaces components (codeready-workspaces-server, identity provider, devfile and plugin registries) are accessible from inside the OpenShift cluster.**

Take into account the network topology of the environment to determine how best to accomplish this. For example, on a network owned by a company or an organization, the network administrators must ensure that traffic bound from the cluster can be routed to Ingress and Route hostnames. In other cases, for example, on AWS, create a proxy configuration allowing the traffic to leave the node to reach an external-facing Load Balancer.

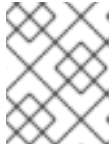
Prerequisites

- A running OpenShift cluster. See the [OpenShift Container Platform 4.2 Documentation](#) for instructions on how to install an OpenShift cluster.

3.1. PREPARING AN IMAGE REGISTRY FOR INSTALLING CODEREADY WORKSPACES IN A RESTRICTED ENVIRONMENT

Prerequisites

- The **oc** tool is installed.
- An image registry that is accessible from the OpenShift cluster. Ensure you can push to it from a location that has, at least temporarily, access to the Internet.
- The **podman** tool is installed.

**NOTE**

When pushing images to other registry than the OpenShift internal registry, and the **podman** tool fails to work, use the **docker** tool instead.

The following placeholders are used in this section.

Table 3.1. Placeholders used in examples

<internal-registry>	host name and port of the container-image registry accessible in the restricted environment
<organization>	organization of the container-image registry

**NOTE**

For the OpenShift internal registry, the placeholder values are typically the following:

Table 3.2. Placeholders for the internal OpenShift registry

<internal-registry>	image-registry.openshift-image-registry.svc:5000
<organization>	openshift

See [OpenShift documentation](#) for more details.

Procedure

1. Define the environment variable with the external endpoint of the image registry:
For the OpenShift internal registry, use:

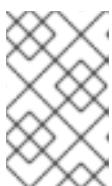
```
$ REGISTRY_ENDPOINT=$(oc get route default-route --namespace openshift-image-registry \
--template='{{ .spec.host }}')
```

For other registries, use the host name and port of the image registry:

```
$ REGISTRY_ENDPOINT=<internal-registry>
```

2. Log into the internal image registry:

```
$ podman login --username <user> --password <password> <internal-registry>
```

**NOTE**

When using the OpenShift internal registry, follow the steps described in the related [OpenShift documentation](#) to first expose the internal registry through a route, and then log in to it.

3. Download, tag, and push the necessary images. Repeat the step for every image in the following lists:

```
$ podman pull <image_name>:<image_tag>
$ podman tag <image_name>:<image_tag>
${REGISTRY_ENDPOINT}/<organization>/<image_name>:<image_tag>
$ podman push ${REGISTRY_ENDPOINT}/<organization>/<image_name>:<image_tag>
```

Essential images

The following infrastructure images are included in every workspace launch:

- registry.redhat.io/codeready-workspaces/server-operator-rhel8:2.0
- registry.redhat.io/codeready-workspaces/server-rhel8:2.0
- registry.redhat.io/codeready-workspaces/pluginregistry-rhel8:2.0
- registry.redhat.io/codeready-workspaces/devfileregistry-rhel8:2.0
- registry.redhat.io/codeready-workspaces/pluginbroker-rhel8:2.0
- registry.redhat.io/codeready-workspaces/pluginbrokerinit-rhel8:2.0
- registry.redhat.io/codeready-workspaces/jwtproxy-rhel8:2.0
- registry.redhat.io/codeready-workspaces/machineexec-rhel8:2.0
- registry.redhat.io/codeready-workspaces/theia-rhel8:2.0
- registry.redhat.io/codeready-workspaces/theia-endpoint-rhel8:2.0
- registry.redhat.io/rhsc/pgsql-96-rhel7:1-47
- registry.redhat.io/redhat-sso-7/sso73-openshift:1.0-15
- registry.redhat.io/ubi8-minimal:8.0-213

Workspace-specific images

These are images that are required for running a workspace. A workspace generally uses only a subset of the images below. It is only necessary to include the images related to required technology stacks.

- registry.redhat.io/codeready-workspaces/stacks-cpp-rhel8:2.0
- registry.redhat.io/codeready-workspaces/stacks-dotnet-rhel8:2.0
- registry.redhat.io/codeready-workspaces/stacks-golang-rhel8:2.0
- registry.redhat.io/codeready-workspaces/stacks-java-rhel8:2.0
- registry.redhat.io/codeready-workspaces/stacks-node-rhel8:2.0
- registry.redhat.io/codeready-workspaces/stacks-php-rhel8:2.0
- registry.redhat.io/codeready-workspaces/stacks-python-rhel8:2.0

- registry.redhat.io/codeready-workspaces/plugin-openshift-rhel8:2.0

3.2. PREPARING CODEREADY WORKSPACES CUSTOM RESOURCE FOR RESTRICTED ENVIRONMENT

When installing CodeReady Workspaces in a restricted environment using **crwctl** or OperatorHub, provide a **CodeReady WorkspacesCluster** custom resource with additional information.

3.2.1. Downloading the default CodeReady WorkspacesCluster Custom Resource

Procedure

1. Download [the default custom resource YAML file](#).
2. Name the downloaded custom resource **org_v1_che_cr.yaml**. Keep it for further modification and usage.

3.2.2. Customizing the CodeReady WorkspacesCluster Custom Resource for restricted environment

Prerequisites

- All required images available in an image registry that is visible to the OpenShift cluster where CodeReady Workspaces is to be deployed. This is described in [Section 3.1, “Preparing an image registry for installing CodeReady Workspaces in a restricted environment”](#), where the placeholders used in the following examples are also defined.

Procedure

1. In the **CodeReady WorkspacesCluster** Custom Resource, which is managed by the CodeReady Workspaces Operator, add the fields used to facilitate deploying an instance of CodeReady Workspaces in a restricted environment:

```
# [...]
spec:
  server:
    airGapContainerRegistryHostname: '<internal-registry>'
    airGapContainerRegistryOrganization: '<organization>'
# [...]
```

Setting these fields in the Custom Resource uses **<internal-registry>** and **<organization>** for all images. This means, for example, that the Operator expects the offline plug-in and devfile registries to be available at:

```
<internal-registry>/<organization>/pluginregistry-rhel8:<ver>
<internal-registry>/<organization>/pluginregistry-rhel8:<ver>
```

For example, to use the OpenShift 4 internal registry as the image registry, define the following fields in the **CodeReady WorkspacesCluster** Custom Resource:

```
# [...]
spec:
```

```
server:
  airGapContainerRegistryHostname: 'image-registry.openshift-image-registry.svc:5000'
  airGapContainerRegistryOrganization: 'openshift'
# [...]
```

2. In the downloaded **CodeReady WorkspacesCluster** Custom Resource, add the two fields described above with the proper values according to the container-image registry with all the mirrored container images.

3.3. STARTING CODEREADY WORKSPACES INSTALLATION IN A RESTRICTED ENVIRONMENT USING CODEREADY WORKSPACES CLI MANAGEMENT TOOL

This sections describes how to start the CodeReady Workspaces installation in a restricted environment using the CodeReady Workspaces CLI management tool.

Prerequisites

- CodeReady Workspaces CLI management tool is installed.
- The **oc** tool is installed.
- Access to an OpenShift instance.

Procedure

1. Log in to OpenShift Container Platform:

```
$ oc login ${OPENSIFT_API_URL} --username ${OPENSIFT_USERNAME} \
  --password ${OPENSIFT_PASSWORD}
```

2. Install CodeReady Workspaces with the customized Custom Resource to add fields related to restricted environment:

```
$ crwctl server:start \
  --che-operator-image=<image-registry>/<organization>/server-operator-rhel8:2.0 \
  --che-operator-cr-yaml=org_v1_che_cr.yaml
```

3.4. STARTING CODEREADY WORKSPACES INSTALLATION IN A RESTRICTED ENVIRONMENT USING OPERATORHUB

Installing CodeReady Workspaces from OperatorHub in a restricted environment is not supported.

CHAPTER 4. UPGRADING CODEREADY WORKSPACES

This chapter describes how to upgrade from CodeReady Workspaces 1.2 instance to CodeReady Workspaces 2.0.

The method that was used to install the CodeReady Workspaces 1.2 instance determines the method to proceed with for the upgrade.

Following scenarios are covered:

- [Section 4.1, “Upgrading CodeReady Workspaces on OpenShift 4 using the web console”](#)
- [Section 4.2, “Upgrading CodeReady Workspaces on OpenShift 4 using the CLI management tool”](#)
- [Section 4.3, “Upgrading CodeReady Workspaces on OpenShift 3 using the CLI management tool”](#)

After a major upgrade of the CodeReady Workspaces instance from 1.2 to 2.0, it is necessary to migrate the workspaces to new devfiles definitions:

See [the CodeReady Workspaces 2.0 End-user Guide](#).

4.1. UPGRADING CODEREADY WORKSPACES ON OPENSIFT 4 USING THE WEB CONSOLE

This section describes how to upgrade from CodeReady Workspaces 1.2 to CodeReady Workspaces 2.0 on OpenShift 4 using the OpenShift web console. This method is using the Operator from OperatorHub.

4.1.1. Preparing the upgrade of CodeReady Workspaces on OpenShift using the Operator

This section describes how to prepare upgrade from CodeReady Workspaces 1.2 to CodeReady Workspaces 2.0 on OpenShift using the Operator.

Uninstalling the Red Hat CodeReady Workspaces 1.2 Operator means that the OpenShift cluster do not receive any updates of the CodeReady Workspaces 1.2 Operator. As a consequence, the existing CodeReady Workspaces instances do not receive any update. The CodeReady Workspaces instances that are already present, including their database, remains present.

Prerequisites

- An administrative account on an OpenShift 4 instance.
- Red Hat CodeReady Workspaces 1.2 Operator is installed on this OpenShift 4 instance.
- A running instance of CodeReady Workspaces 1.2 on this OpenShift 4 instance.

Procedure

1. In all running workspaces in the CodeReady Workspaces 1.2 instance, save and push changes back to the git repositories.
2. Shut down all workspaces in the CodeReady Workspaces 1.2 instance.
3. Open the OpenShift console.

4. Navigate to **Operators > Installed Operators**
5. Click on **Red Hat CodeReady Workspaces 1.2**
6. Click on the **Actions** button on the top right.
7. A drop-down menu appears.
8. Click on **Uninstall Operator** to uninstall the CodeReady Workspaces 1.2 Operator.
9. Accept the selected option **Also completely remove the Operator from the selected namespace**.
10. Click on the **Remove** button.
11. Navigate to **Operators > Installed Operators**
12. The Operator is removed from the list of installed operators.

4.1.2. Creating the CodeReady Workspaces project in OpenShift 4 web console

This section describes how to create the **CodeReady Workspaces** project in OpenShift 4 web console.

Prerequisites

- An administrator account on a running instance of OpenShift 4.

Procedure

1. Open the OpenShift web console.
2. In the left panel, navigate to **Projects**.
3. Click the **Create Project** button.
4. Enter the project details:
 - In the **Name** field, type **workspaces**.
 - In the **Display Name** field, type **Red Hat CodeReady Workspaces**.
5. Click the **Create** button.

4.1.3. Installing the CodeReady Workspaces Operator in OpenShift 4 web console

This section describes how to install the CodeReady Workspaces Operator in OpenShift 4 web console.

Prerequisites

- An administrator account on a running instance of OpenShift 4.
- Administrative rights on an existing project named **workspaces** on this instance of OpenShift 4. See [Section 4.1.2, "Creating the CodeReady Workspaces project in OpenShift 4 web console"](#).
- The Red Hat CodeReady Workspaces 1.2 Operator is not installed.

Procedure

1. Open the OpenShift web console.

1. Open the OpenShift web console.
2. In the left panel, navigate to the **Operators** → **OperatorHub** section.
3. In the **Search by keyword** field, type **Red Hat CodeReady Workspaces**.
4. Click on the **Red Hat CodeReady Workspaces** tile.
5. Click the **Install** button in the **Red Hat CodeReady Workspaces** pop-up window.
6. In the **A specific namespace on the cluster** field, in the cluster drop-down list, select the namespace into which the previous version of the CodeReady Workspaces Operator was installed.
7. Click the **Subscribe** button.
8. In the left panel, navigate to the **Operators** → **Installed Operators** section.
9. Red Hat CodeReady Workspaces is displayed as an installed Operator having the **InstallSucceeded** status.
10. Click on the **Red Hat CodeReady Workspaces** name in the list of installed operators.
11. Navigate to the **Overview** tab.
12. In the **Conditions** sections at the bottom of the page, wait for this message: **install strategy completed with no errors**.
13. Navigate to the **Events** tab.
14. Wait for this message: **install strategy completed with no errors**.

4.1.4. Viewing the state of the CodeReady Workspaces cluster deployment in OpenShift 4 web console

This section describes how to view the state of the CodeReady Workspaces cluster deployment in OpenShift 4 web console.

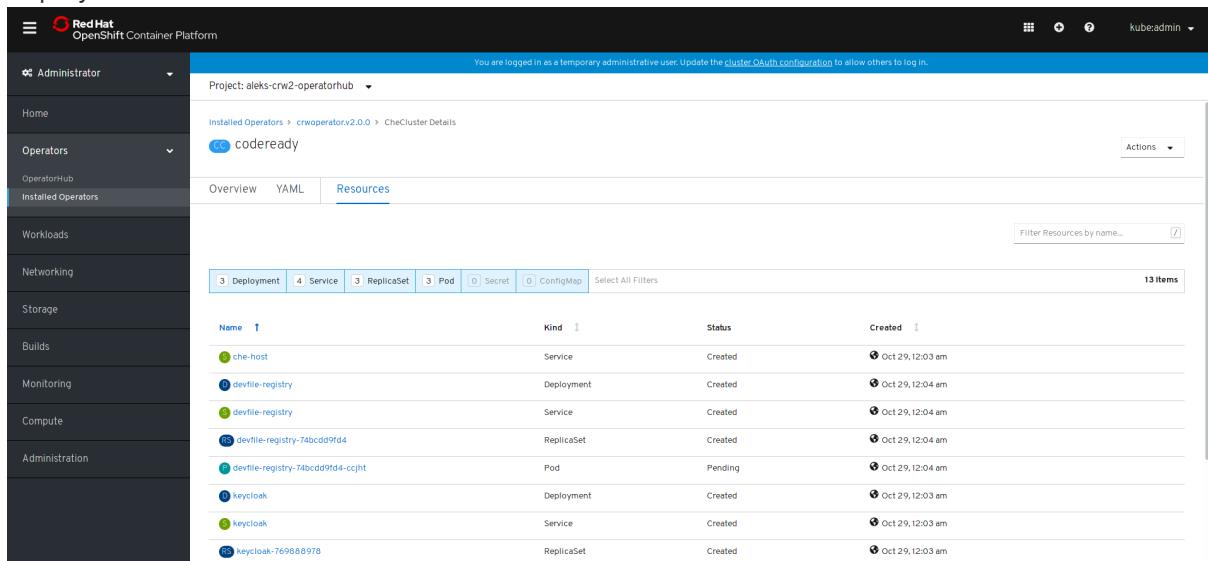
Prerequisites

- An administrator account on a running instance of OpenShift 4.
- A CodeReady Workspaces cluster is deployed on this instance of OpenShift 4.

Procedure

1. Open the OpenShift web console.
2. Navigate to the **Operators** → **Installed Operators** section.
3. Click **Red Hat CodeReady Workspaces** in the list of installed operators.
4. Navigate to the **Red Hat CodeReady Workspaces Cluster** tab.
5. Click on the **eclipse-che** cluster that is listed in the table.
6. Navigate to the **Overview** tab.

- Watch the content of the **Message** field. The field contain error messages, if any. The expected content is **None**.
- Navigate to the **Resources** tab.
- The screen displays the state of the resources assigned to the CodeReady Workspaces cluster deployment.



4.1.5. Finishing the upgrade of CodeReady Workspaces on OpenShift using the Operator

This section describes how to finish the upgrade from CodeReady Workspaces 1.2 to CodeReady Workspaces 2.0 on OpenShift using the Operator.

Prerequisites

- The CodeReady Workspaces 1.2 Operator is removed. See [Section 4.1.1, “Preparing the upgrade of CodeReady Workspaces on OpenShift using the Operator”](#)
- The CodeReady Workspaces 2.0 Operator is deployed. See [Section 4.1.3, “Installing the CodeReady Workspaces Operator in OpenShift 4 web console”](#)
- The CodeReady Workspaces 2.0 Operator has finished upgrading the Red Hat CodeReady Workspaces cluster. See [Section 4.1.4, “Viewing the state of the CodeReady Workspaces cluster deployment in OpenShift 4 web console”](#)

Procedure

- Log into the Red Hat CodeReady Workspaces instance.
- Select an old workspace and click on the **Start** button.
- An error message appears, with a link to the documentation explaining how to migrate a workspace to a devfile.
- For each workspace, create a devfile that represent your workspace. See the CodeReady Workspaces 2.0 End-user Guide.
- Use the devfile to start the new workspace. See [the CodeReady Workspaces 2.0 End-user Guide](#).

4.2. UPGRADING CODEREADY WORKSPACES ON OPENSIFT 4 USING THE CLI MANAGEMENT TOOL

This section describes how to upgrade from CodeReady Workspaces 1.2 to CodeReady Workspaces 2.0 on OpenShift 4 using the CLI management tool.

Prerequisites

- An administrative account on an OpenShift 4 instance.
- A running instance of Red Hat CodeReady Workspaces running on OpenShift 4.
- This instance was installed using the CLI management tool, not using OperatorHub.
- The **crwctl** management tool is installed. See [the CodeReady Workspaces 2.0 Installation Guide](#).

Procedure

1. In all running workspaces in the CodeReady Workspaces 1.2 instance, save and push changes back to the Git repositories.
2. Shut down all workspaces in the CodeReady Workspaces 1.2 instance.
3. Run following command:

```
$ crwctl server:update
```

Verification steps

1. Log in to the CodeReady Workspaces instance.
2. The 2.0 version number is displayed at the bottom of the page.

4.3. UPGRADING CODEREADY WORKSPACES ON OPENSIFT 3 USING THE CLI MANAGEMENT TOOL

This section describes how to upgrade from CodeReady Workspaces 1.2 to CodeReady Workspaces 2.0 on OpenShift 3 using the CLI management tool.

Prerequisites

- An administrative account on an OpenShift 3 instance.
- A running instance of Red Hat CodeReady Workspaces running on OpenShift 3.
- This instance was installed using the CLI management tool.
- The **crwctl** management tool is installed. See [the CodeReady Workspaces 2.0 Installation Guide](#).

Procedure

1. In all running workspaces in the CodeReady Workspaces 1.2 instance, save and push changes back to the Git repositories.
2. Shut down all workspaces in the CodeReady Workspaces 1.2 instance.
3. Run the following command:

```
$ crwctl server:update
```

Verification steps

1. Log in to the CodeReady Workspaces instance.
2. The 2.0 version number is displayed at the bottom of the page.

4.4. KNOWN ISSUES

4.4.1. Upgrading from CodeReady Workspaces 1.2 to CodeReady Workspaces 2.0 using the Operator

When upgrading the Operator of a CodeReady Workspaces 1.2 installation to CodeReady Workspaces 2.0, some components may not be automatically updated. To fix this, manually update the images listed in the **checluster** custom resource, which describes the CodeReady Workspaces installation. To reset the images to their default versions for given Operator version, set the image properties to empty strings. This forces the CodeReady Workspaces Operator to use the default versions it was built with.

Review the following components:

Component	CR attribute name
Postgresql	spec.database.postgresImage
Keycloak	spec.auth.identityProviderImage
PVC Jobs	spec.storage.pvcJobsImage

4.4.2. Updating a CodeReady Workspaces installation using the Operator

When making changes to the **checluster** custom resource, use patching to make updates to it. For example:

On OpenShift, run:

```
$ oc patch checluster <my-che> --type=json -n <che-namespace> -p '...'
```

On OpenShift, run:

```
$ oc patch checluster <my-che> --type=json -n <che-namespace> -p '...'
```

**WARNING**

Making local updates to the YAML file of the **checluster** resource and then applying such changed resource to the cluster using **oc apply -f** or **oc apply -f** can result in an invalidation of the CodeReady Workspaces installation.

CHAPTER 5. ADVANCED CONFIGURATION OPTIONS

The following section describes advanced deployment and configuration methods for Red Hat CodeReady Workspaces.

5.1. CODEREADY WORKSPACES CONFIGMAPS AND THEIR BEHAVIOR

The following section describes CodeReady Workspaces **configMaps** and how they behave.

A **configMap** is provided as an editable file that lists options to customize the CodeReady Workspaces environment. Based on the CodeReady Workspaces installation method, **configMaps** can be used to customize the working environment. The type of configMaps available in your CodeReady Workspaces environment varies based on the method used for installing CodeReady Workspaces.

5.1.1. CodeReady Workspaces installed using an Operator

Operators are software extensions to OpenShift that use **custom resources** to manage applications and their components.

CodeReady Workspaces installed using the Operator provides the user with an automatically generated **configMap** called **workspaces**.

The **workspaces configMap** contains the main properties for the CodeReady Workspaces server, and is in sync with the information stored in the CheCluster Custom Resource file. User modifications of the **workspaces configMap** after installing CodeReady Workspaces using the Operator are automatically overwritten by values that the Operator obtains from the **CheCluster** Custom Resource.

To edit the **workspaces configMap**, edit the Custom Resource manually. The **configMap** derives values from the **CheCluster** field. User modifications of the **CheCluster** Custom Resource field cause the Operator to change the attributes of the **workspaces configMap** accordingly. The **configMap** changes automatically trigger a restart of the CodeReady Workspaces Pod.

To add custom properties to the CodeReady Workspaces server, such as environment variables that are not automatically generated in the **workspaces configMap** by the Operator, or to override automatically generated properties, the **CheCluster** Custom Resource has a **customCheProperties** field, which expects a map.

For example, to override the default memory limit for workspaces, add the **CHE_WORKSPACE_DEFAULT_MEMORY_LIMIT_MB** property to **customCheProperties**:

```
apiVersion: org.eclipse.che/v1
kind: CheCluster
metadata:
  name: eclipse-che
  namespace: che
spec:
  server:
    cheImageTag: "
    devfileRegistryImage: "
    pluginRegistryImage: "
    tlsSupport: false
    selfSignedCert: false
    customCheProperties:
```



```
CHE_WORKSPACE_DEFAULT_MEMORY_LIMIT_MB: "2048"
auth:
...
```

Previous versions of the CodeReady Workspaces Operator had a configMap named **custom** to fulfill this role. If the CodeReady Workspaces Operator finds a **configMap** with the name **custom**, it adds the data it contains into the **customCheProperties** field, redeploys CodeReady Workspaces, and deletes the **custom configMap**.

5.2. CONFIGURING NAMESPACE STRATEGIES



NOTE

The term *namespace* (Kubernetes) is used interchangeably with *project* (OpenShift).

The namespace strategies are configured using the **CHE_INFRA_KUBERNETES_NAMESPACE_DEFAULT** environment variable.



WARNING

There are legacy variables **CHE_INFRA_KUBERNETES_NAMESPACE** and **CHE_INFRA_OPENSHIFT_PROJECT**. These should be left unset for new installations. Changing these variables during update can lead to data loss.

5.2.1. One namespace per workspace strategy

The strategy creates a new namespace for each new workspace.

To use the strategy, the **CHE_INFRA_KUBERNETES_NAMESPACE_DEFAULT** variable value must contain the **<workspaceid>** identifier. It can be used alone or combined with other identifiers or any string.

Example 5.1. One namespace per workspace

To assign namespace names composed of a **che-ws** prefix and workspace id, set:

```
CHE_INFRA_KUBERNETES_NAMESPACE_DEFAULT=che-ws-<workspaceid>
```

5.2.2. One namespace for all workspaces strategy

The strategy uses one predefined namespace for all workspaces.

To use the strategy, the **CHE_INFRA_KUBERNETES_NAMESPACE_DEFAULT** variable value must be the name of the desired namespace to use.

Example 5.2. One namespace for all workspaces

To have all workspaces created in **che-workspaces** namespace, set:

```
CHE_INFRA_KUBERNETES_NAMESPACE_DEFAULT=che-workspaces
```



IMPORTANT

To run more than one workspace at a time when using this strategy together with the **common** PVC strategy, configure persistent volumes to use **ReadWriteMany** access mode.

5.2.3. One namespace per user strategy

The strategy isolates each user in their own namespace.

To use the strategy, the **CHE_INFRA_KUBERNETES_NAMESPACE_DEFAULT** variable value must contain one or more user identifiers. Currently supported identifiers are **<username>** and **<userid>**.

Example 5.3. One namespace per user

To assign namespace names composed of a **che-ws** prefix and individual usernames (**che-ws-user1**, **che-ws-user2**), set:

```
CHE_INFRA_KUBERNETES_NAMESPACE_DEFAULT=che-ws-<username>
```



IMPORTANT

To run more than one workspace at a time when using this strategy together with the **common** PVC strategy, configure persistent volumes to use **ReadWriteMany** access mode.

To limit the number of concurrently running workspaces per user to one, set the **CHE_LIMITS_USER_WORKSPACES_RUN_COUNT** environment variable to **1**.

To limit the number of concurrently running workspaces per user to one (1):

- For Helm Chart deployments: set the **.global.workspace.number** parameter to **1**.
- For Operator deployments: set the **spec.server.cheCustomProperties.CHE_LIMITS_USER_WORKSPACE_RUN_COUNT** variable of the CheCluster Custom Resource (CR) to **1**.

5.2.4. Allowing user-defined workspace namespaces

The server can be configured to honor the user selection of a namespace when a workspace is created. This feature is disabled by default. To allow user-defined workspace namespaces, set:

```
CHE_INFRA_KUBERNETES_NAMESPACE_ALLOWUSERDEFINED=true
```

5.3. DEPLOYING CODEREADY WORKSPACES WITH SUPPORT FOR GIT REPOSITORIES WITH SELF-SIGNED CERTIFICATES

This procedure describes how to configure CodeReady Workspaces for deployment with support for Git operations on repositories that use self-signed certificates.

Prerequisites

- Git version 2 or later.

Configuring support for self-signed Git repositories on OpenShift

Deploying CodeReady Workspaces using a Helm Chart

1. Configure the workspace exposure strategy using the **global.useGitSelfSignedCerts** property. To do that, add the following option to the **helm upgrade** command:

```
$ helm upgrade che --set global.useGitSelfSignedCerts=true
```

2. Create a new **configMap** with details about the Git server:

```
$ oc create configmap che-git-self-signed-cert --from-file=<ca.crt> \
  --from-literal=githost=<host:port> -n=che
```

In the command, substitute **<ca.crt>** for the self-signed certificate and `<host:port>` for the host and port of the HTTPS connection on the Git server (optional).



NOTE

When **githost** is not specified, the given certificate is used for all HTTPS repositories.

Create and start new workspace. Every container used by the workspace mounts a special volume that contains a file with the self-signed certificate. The repository's **.git/config** file contains information about the Git server host (its URL) and the path to the certificate in the **http** section (see Git documentation about [git-config](#)). For example:

```
[http "https://10.33.177.118:3000"]
  sslCAInfo = /etc/che/git/cert/ca.crt
```

5.4. CODEREADY WORKSPACES CONFIGMAPS FIELDS REFERENCE

5.4.1. server settings related to the CodeReady Workspaces server

Property	Default value	Description
----------	---------------	-------------

Property	Default value	Description
airGapContainerRegistryHostname	omit	Optional hostname or URL to an alternate container registry to pull images from. This value overrides the container registry hostname defined in all default container images involved in a CodeReady Workspaces deployment. This is particularly useful to install CodeReady Workspaces in an air-gapped environment.
airGapContainerRegistryOrganization	omit	Optional repository name of an alternate container registry to pull images from. This value overrides the container registry organization defined in all the default container images involved in a CodeReady Workspaces deployment. This is particularly useful to install CodeReady Workspaces in an air-gapped environment.
cheDebug	false	Enables the debug mode for CodeReady Workspaces server.
cheFlavor	codeready-workspaces	Flavor of the installation.
cheHost	The Operator automatically sets the value.	Public hostname of the installed CodeReady Workspaces server.
cheImagePullPolicy	Always for nightly or latest images, and IfNotPresent in other cases	Overrides the image pull policy used in CodeReady Workspaces deployment.
cheImageTag	omit	Overrides the tag of the container image used in CodeReady Workspaces deployment. Omit it or leave it empty to use the default image tag provided by the Operator.
cheImage	omit	Overrides the container image used in CodeReady Workspaces deployment. This does not include the container image tag. Omit it or leave it empty to use the default container image provided by the Operator.
cheLogLevel	INFO	Log level for the CodeReady Workspaces server: INFO or DEBUG .
cheWorkspaceClusterRole	omit	Custom cluster role bound to the user for the workspaces. Omit or leave empty to use the default roles.

Property	Default value	Description
customCheProperties	omit	Map of additional environment variables that will be applied in the generated codeready-workspaces config map to be used by the CodeReady Workspaces server, in addition to the values already generated from other fields of the CheCluster custom resource (CR). If customCheProperties contains a property that would be normally generated in codeready-workspaces config map from other CR fields, then the value defined in the customCheProperties will be used instead.
devfileRegistryImage	omit	Overrides the container image used in the Devfile registry deployment. This includes the image tag. Omit it or leave it empty to use the default container image provided by the Operator.
devfileRegistryMemoryLimit	256Mi	Overrides the memory limit used in the Devfile registry deployment.
devfileRegistryMemoryRequest	16Mi	Overrides the memory request used in the Devfile registry deployment.
devfileRegistryPullPolicy	Always for nightly or latest images, and IfNotPresent in other cases	Overrides the image pull policy used in the Devfile registry deployment.
devfileRegistryUrl	The Operator automatically sets the value.	Public URL of the Devfile registry that serves sample, ready-to-use devfiles. Set it if you use an external devfile registry (see the externalDevfileRegistry field).
externalDevfileRegistry	false	Instructs the Operator to deploy a dedicated Devfile registry server. By default a dedicated devfile registry server is started. If externalDevfileRegistry set to true , the Operator does not start a dedicated registry server automatically and you need to set the devfileRegistryUrl field manually.
externalPluginRegistry	false	Instructs the Operator to deploy a dedicated Plugin registry server. By default, a dedicated plug-in registry server is started. If externalPluginRegistry set to true , the Operator does not deploy a dedicated server automatically and you need to set the pluginRegistryUrl field manually.
nonProxyHosts	omit	List of hosts that should not use the configured proxy. Use <code> </code> as delimiter, eg localhost my.host.com 123.42.12.32 Only use when configuring a proxy is required (see also the proxyURL field).

Property	Default value	Description
pluginRegistryImage	omit	Overrides the container image used in the Plugin registry deployment. This includes the image tag. Omit it or leave it empty to use the default container image provided by the Operator.
pluginRegistryMemoryLimit	256Mi	Overrides the memory limit used in the Plugin registry deployment.
pluginRegistryMemoryRequest	16Mi	Overrides the memory request used in the Plugin registry deployment.
pluginRegistryPullPolicy	Always for nightly or latest images, and IfNotPresent in other cases	Overrides the image pull policy used in the Plugin registry deployment.
pluginRegistryURL	the Operator sets the value automatically	Public URL of the Plugin registry that serves sample ready-to-use devfiles. Set it only when using an external devfile registry (see the externalPluginRegistry field).
proxyPassword	omit	Password of the proxy server. Only use when proxy configuration is required.
proxyPort	omit	Port of the proxy server. Only use when configuring a proxy is required (see also the proxyURL field).
proxyURL	omit	URL (protocol+hostname) of the proxy server. This drives the appropriate changes in the JAVA_OPTS and https(s)_proxy variables in the CodeReady Workspaces server and workspaces containers. Only use when configuring a proxy is required.
proxyUser	omit	User name of the proxy server. Only use when configuring a proxy is required (see also the proxyURL field).
selfSignedCert	false	Enables the support of OpenShift clusters with routers that use self-signed certificates. When enabled, the Operator retrieves the default self-signed certificate of OpenShift routes and adds it to the Java trust store of the CodeReady Workspaces server. Required when activating the tlsSupport field on demo OpenShift clusters that have not been setup with a valid certificate for the routes.
serverMemoryLimit	1Gi	Overrides the memory limit used in the CodeReady Workspaces server deployment.
serverMemoryRequest	512Mi	Overrides the memory request used in the CodeReady Workspaces server deployment.

Property	Default value	Description
tlsSupport	false	Instructs the Operator to deploy CodeReady Workspaces in TLS mode. Enabling TLS requires enabling the selfSignedCert field.

5.4.2. database configuration settings related to the database used by CodeReady Workspaces

Property	Default value	Description
chePostgresDb	dbche	Postgres database name that the CodeReady Workspaces server uses to connect to the database.
chePostgresHostName	the Operator sets the value automatically	Postgres Database hostname that the CodeReady Workspaces server uses to connect to. Defaults to postgres. Override this value only when using an external database. (See the field externalDb .)
chePostgresPassword	auto-generated value	Postgres password that the CodeReady Workspaces server uses to connect to the database.
chePostgresPort	5432	Postgres Database port that the CodeReady Workspaces server uses to connect to. Override this value only when using an external database (see field externalDb).
chePostgresUser	pgche	Postgres user that the CodeReady Workspaces server uses to connect to the database.
externalDb	false	Instructs the Operator to deploy a dedicated database. By default, a dedicated Postgres database is deployed as part of the CodeReady Workspaces installation. If set to true , the Operator does not deploy a dedicated database automatically, you need to provide connection details to an external database. See all the fields starting with: chePostgres .
postgresImagePullPolicy	Always` for nightly or latest images, and IfNotPresent in other cases	Overrides the image pull policy used in the Postgres database deployment.
postgresImage	omit	Overrides the container image used in the Postgres database deployment. This includes the image tag. Omit it or leave it empty to use the default container image provided by the Operator.

5.4.3. auth configuration settings related to authentication used by CodeReady Workspaces installation

Property	Default value	Description
externalIdentityProvider	false	By default, a dedicated Identity Provider server is deployed as part of the CodeReady Workspaces installation. But if externalIdentityProvider is true , then no dedicated identity provider will be deployed by the Operator and you might need to provide details about the external identity provider you want to use. See also all the other fields starting with: identityProvider .
identityProviderAdminUserName	admin	Overrides the name of the Identity Provider admin user.
identityProviderClientid	omit	Name of an Identity provider (Keycloak / RH SSO) client-id that should be used for CodeReady Workspaces. This is useful to override it ONLY if you use an external Identity Provider (see the externalIdentityProvider field). If omitted or left blank, it will be set to the value of the flavor field suffixed with -public .
identityProviderImagePullPolicy	Always for nightly or latest images, and IfNotPresent in other cases	Overrides the image pull policy used in the Identity Provider (Keycloak / RH SSO) deployment.
identityProviderImage	omit	Overrides the container image used in the Identity Provider (Keycloak / RH SSO) deployment. This includes the image tag. Omit it or leave it empty to use the default container image provided by the Operator.
identityProviderPassword	omit	Overrides the password of Keycloak admin user. Override it only when using an external Identity Provider (see the externalIdentityProvider field). Omit or leave empty to set an auto-generated password.
identityProviderPostgresPassword	the Operator sets the value automatically	Password for The Identity Provider (Keycloak / RH SSO) to connect to the database. This is useful to override it ONLY if you use an external Identity Provider (see the externalIdentityProvider field).
identityProviderRealm	omit	Name of an Identity provider (Keycloak / RH SSO) realm. Override it only when using an external Identity Provider (see the externalIdentityProvider field). Omit or leave empty blank to set it to the value of the flavor field.

Property	Default value	Description
identityProviderURL	the Operator sets the value automatically	Instructs the Operator to deploy a dedicated Identity Provider (Keycloak or RH SSO instance). Public URL of the Identity Provider server (Keycloak / RH SSO server). Set it only when using an external Identity Provider (see the externalIdentityProvider field).
oAuthClientName	the Operator sets the value automatically	Name of the OpenShift OAuthClient resource used to setup identity federation on the OpenShift side. See also the OpenShifttoAuth field.
oAuthSecret	the Operator sets the value automatically	Name of the secret set in the OpenShift OAuthClient resource used to setup identity federation on the OpenShift side. See also the OAuthClientName field.
openShifttoAuth	true on OpenShift	Enables the integration of the identity provider (Keycloak / RHSSO) with OpenShift OAuth. This allows users to login with their Openshift login and have their workspaces created under personal OpenShift namespaces. The kubeadmin user is not supported, and logging through does not allow access to the CodeReady Workspaces Dashboard.
updateAdminPassword	false	Forces the default admin CodeReady Workspaces user to update password on first login.

5.4.4. storage configuration settings related to persistent storage used by CodeReady Workspaces

Property	Default value	Description
postgresPVCStorageClassName	omit	Storage class for the Persistent Volume Claim dedicated to the Postgres database. Omitted or leave empty to use a default storage class.
preCreateSubPaths	false	Instructs the CodeReady Workspaces server to launch a special Pod to pre-create a subpath in the Persistent Volumes. Enable it according to the configuration of your K8S cluster.
pvcClaimSize	1Gi	Size of the persistent volume claim for workspaces.
pvcJobsImage	omit	Overrides the container image used to create sub-paths in the Persistent Volumes. This includes the image tag. Omit it or leave it empty to use the default container image provided by the Operator. See also the preCreateSubPaths field.
pvcStrategy	common	Available options: <code>common</code> (all workspaces PVCs in one volume), per-workspace (one PVC per workspace for all declared volumes) and unique (one PVC per declared volume).

Property	Default value	Description
workspacePVCStorageClassName	omit	Storage class for the Persistent Volume Claims dedicated to the workspaces. Omit or leave empty to use a default storage class.

5.4.5. k8s configuration settings specific to CodeReady Workspaces installations on OpenShift

Property	Default value	Description
ingressClass	nginx	Ingress class that defines which controller manages ingresses.
ingressDomain	omit	Global ingress domain for a K8S cluster. This field must be explicitly specified. This drives the is.kubernetes.io/ingress.class annotation on CodeReady Workspaces-related ingresses.
ingressStrategy	multi-host	Strategy for ingress creation. This can be multi-host (host is explicitly provided in ingress), single-host (host is provided, path-based rules) and default-host.* (no host is provided, path-based rules).
securityContextFsGroup,omite mpty	1724	FSGroup the CodeReady Workspaces Pod and Workspace Pods containers should run in.
securityContextRunAsUser	1724	ID of the user the CodeReady Workspaces Pod and Workspace Pods containers should run as.
tlsSecretName	omit	Name of a secret that is used to set ingress TLS termination if TLS is enabled. See also the tlsSupport field.

5.4.6. installation defines the observed state of CodeReady Workspaces installation

Property	Description
cheClusterRunning	Status of a CodeReady Workspaces installation. Can be Available, Unavailable, or Available, Rolling Update in Progress .
cheURL	Public URL to the CodeReady Workspaces server.
cheVersion	Currently installed CodeReady Workspaces version.
dbProvisioned	Indicates whether a Postgres instance has been correctly provisioned.
devfileRegistryURL	Public URL to the Devfile registry.

Property	Description
helpLink	A URL to where to find help related to the current Operator status.
keycloakProvisioned	Indicates whether an Identity Provider instance (Keycloak / RH SSO) has been provisioned with realm, client and user.
keycloakURL	Public URL to the Identity Provider server (Keycloak / RH SSO).
message	A human-readable message with details about why the Pod is in this state.
openShifttoAuthProvisioned	Indicates whether an Identity Provider instance (Keycloak / RH SSO) has been configured to integrate with the OpenShift OAuth.
pluginRegistryURL	Public URL to the Plugin registry.
reason	A brief CamelCase message with details about why the Pod is in this state.

5.4.7. Limits for workspaces

Property	Default value	Description
che.limits.workspace.env.ram	16gb	The maximum amount of RAM that a user can allocate to a workspace when they create a new workspace. The RAM slider is adjusted to this maximum value.
che.limits.workspace.idle.timeout	1800000	The length of time that a user is idle with their workspace when the system will suspend the workspace and then stopping it. Idleness is the length of time that the user has not interacted with the workspace, meaning that one of our agents has not received interaction. Leaving a browser window open counts toward idleness.

5.4.8. Limits for the workspaces of an user

Property	Default value	Description
che.limits.user.workspaces.ram	16gb	The total amount of RAM that a single user is allowed to allocate to running workspaces. A user can allocate this RAM to a single workspace or spread it across multiple workspaces.
che.limits.user.workspaces.count	1800000	The maximum number of workspaces that a user is allowed to create. The user will be presented with an error message if they try to create additional workspaces. This applies to the total number of both running and stopped workspaces.

Property	Default value	Description
che.limits.user.workspaces.run.count	1	The maximum number of running workspaces that a single user is allowed to have. If the user has reached this threshold and they try to start an additional workspace, they will be prompted with an error message. The user will need to stop a running workspace to activate another.

5.4.9. Limits for for the workspaces of an organization

Property	Default value	Description
che.limits.organization.workspaces.ram	-1	The total amount of RAM that a single organization (team) is allowed to allocate to running workspaces. An organization owner can allocate this RAM however they see fit across the team's workspaces.
che.limits.organization.workspaces.count	-1	The maximum number of workspaces that a organization is allowed to own. The organization will be presented an error message if they try to create additional workspaces. This applies to the total number of both running and stopped workspaces.
che.limits.organization.workspaces.run.count	-1	The maximum number of running workspaces that a single organization is allowed. If the organization has reached this threshold and they try to start an additional workspace, they will be prompted with an error message. The organization will need to stop a running workspace to activate another.

CHAPTER 6. UNINSTALLING CODEREADY WORKSPACES

This section describes uninstallation procedures for Red Hat CodeReady Workspaces installed on OpenShift. The uninstallation process leads to a complete removal of CodeReady Workspaces-related user data. The appropriate uninstallation method depends on what method was used to install the CodeReady Workspaces instance.

- For CodeReady Workspaces installed using OperatorHub, see [Section 6.1, “Uninstalling CodeReady Workspaces after OperatorHub installation”](#).
- For CodeReady Workspaces installed using `crwctl`, see [Section 6.2, “Uninstalling CodeReady Workspaces after `crwctl` installation”](#)

6.1. UNINSTALLING CODEREADY WORKSPACES AFTER OPERATORHUB INSTALLATION

Users have two options for uninstalling CodeReady Workspaces from an OpenShift cluster. The following sections describe both of these methods:

- Using the OpenShift Administrator Perspective web UI
- Using `oc` commands from the terminal

6.1.1. Uninstalling CodeReady Workspaces using the OpenShift web console

This section describes how to uninstall CodeReady Workspaces from a cluster using the OpenShift Administrator Perspective main menu.

Prerequisites

- Che was installed on an OpenShift cluster using OperatorHub.

Procedure: deleting the CodeReady Workspaces deployment

1. Open the OpenShift web console.
2. Navigate to the **Operators > Installed Operators** section.
3. Click **Red Hat CodeReady Workspaces** in the list of installed operators.
4. Navigate to the **Red Hat CodeReady Workspaces Cluster** tab.
5. In the row that displays information about the specific CodeReady Workspaces cluster, delete the CodeReady Workspaces Cluster deployment using the drop-down menu illustrated as three horizontal dots situated on the right side of the screen.
6. Alternatively, delete the CodeReady Workspaces deployment by clicking the displayed Red Hat CodeReady Workspaces Cluster, **red-hat-codeready-workspaces**, and select the **Delete cluster** option in the **Actions** drop-down menu on the top right.

Procedure: deleting the CodeReady Workspaces Operator

1. Open the OpenShift web console.
2. Navigate to the **Operators > Installed Operators** section in OpenShift Developer Perspective.

3. In the row that displays information about the specific Red Hat CodeReady Workspaces Operator, uninstall the CodeReady Workspaces Operator using the drop-down menu illustrated as three horizontal dots situated on the right side of the screen.
4. Accept the selected option, **Also completely remove the Operator from the selected namespace**.
5. Alternatively, uninstall the Red Hat CodeReady Workspaces Operator by clicking the displayed Red Hat CodeReady Workspaces Operator, **Eclipse Che**, followed by selecting the **Uninstall Operator** option in the **Actions** drop-down menu on the top right.

6.1.2. Uninstalling CodeReady Workspaces using oc commands

This section provides instructions on how to uninstall a CodeReady Workspaces instance using **oc** commands.

Prerequisites

- CodeReady Workspaces was installed on an OpenShift cluster using OperatorHub.
- OpenShift command-line tools (**oc**) are installed on the local workstation.

Procedure

The following procedure provides command-line outputs as examples. Note that output in the user terminal may differ.

To uninstall a CodeReady Workspaces instance from a cluster:

1. Sign in to the cluster:

```
$ oc login -u <username> -p <password> <cluster_URL>
```

2. Switch to the project where the CodeReady Workspaces instance is deployed:

```
$ oc project <codeready-workspaces_project>
```

3. Obtain the CodeReady Workspaces cluster name. The following shows a cluster named **red-hat-codeready-workspaces**:

```
$ oc get codeready-workspacescluster
NAME          AGE
red-hat-codeready-workspaces 27m
```

4. Delete the CodeReady Workspaces cluster:

```
$ oc delete codeready-workspacescluster red-hat-codeready-workspaces
checluster.org.eclipse.che "red-hat-codeready-workspaces" deleted
```

5. Obtain the name of the CodeReady Workspaces cluster service version (CSV) module. The following detects a CSV module named **red-hat-codeready-workspaces.v2.0.0**:

```
$ oc get csv
NAME          DISPLAY          VERSION  REPLACES          PHASE
```

```
red-hat-codeready-workspaces.v2.0.0 Red Hat CodeReady Workspaces 2.0.0 red-hat-codeready-workspaces.v1.2.0 Succeeded
```

6. Delete the CodeReady Workspaces CSV:

```
$ oc delete csv red-hat-codeready-workspaces.v2.0.0
clusterserviceversion.operators.coreos.com "red-hat-codeready-workspaces.v2.0.0" deleted
```

6.2. UNINSTALLING CODEREADY WORKSPACES AFTER CRWCTL INSTALLATION

This section describes how to uninstall an instance of Red Hat CodeReady Workspaces that was installed using the **crwctl** tool.



IMPORTANT

- For CodeReady Workspaces installed using the **crwctl server:start** command and the **-n** argument (custom namespace specified), use the **-n** argument also to uninstall the CodeReady Workspaces instance.
- For installations that did not use the **-n** argument, the created namespace is named **workspaces** by default.

Prerequisites

- CodeReady Workspaces was installed on an OpenShift cluster using **crwctl**.
- OpenShift command-line tools (**oc**) and **crwctl** are installed on the local workstation.
- The user is logged in a CodeReady Workspaces cluster using **oc**.

Procedure

1. Stop the Red Hat CodeReady Workspaces Server:

```
$ crwctl server:stop
```

2. Obtain the name of the CodeReady Workspaces namespace:

```
$ oc get checluster --all-namespaces -o=jsonpath="{.items[*].metadata.namespace}"
```

3. Remove CodeReady Workspaces from the cluster:

```
$ crwctl server:delete -n <namespace>
```

This removes all CodeReady Workspaces installations from the cluster.