



Red Hat CodeReady Workspaces 2.2

Installation Guide

Installing Red Hat CodeReady Workspaces 2.2

Red Hat CodeReady Workspaces 2.2 Installation Guide

Installing Red Hat CodeReady Workspaces 2.2

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. CONFIGURING THE CODEREADY WORKSPACES INSTALLATION	3
1.1. UNDERSTANDING THE CHECLUSTER CUSTOM RESOURCE	3
1.2. CHECLUSTER CUSTOM RESOURCE FIELDS REFERENCE	3
CHAPTER 2. INSTALLING CODEREADY WORKSPACES ON OPENSIFT CONTAINER PLATFORM	12
2.1. INSTALLING CODEREADY WORKSPACES USING THE CODEREADY WORKSPACES OPERATOR IN OPENSIFT 4 WEB CONSOLE	12
2.2. INSTALLING CODEREADY WORKSPACES USING THE CLI MANAGEMENT TOOL ON OPENSIFT CONTAINER PLATFORM 3.11	14
2.2.1. Installing the crwctl CLI management tool	14
2.2.2. Installing CodeReady Workspaces on OpenShift 3 using the Operator	14
CHAPTER 3. INSTALLING CODEREADY WORKSPACES IN A RESTRICTED ENVIROMENT	17
3.1. INSTALLING CODEREADY WORKSPACES IN A RESTRICTED ENVIROMENT USING OPERATORHUB	17
3.2. INSTALLING CODEREADY WORKSPACES IN A RESTRICTED ENVIROMENT USING CLI MANAGEMENT TOOL	18
3.2.1. Preparing an image registry for installing CodeReady Workspaces in a restricted environment	18
3.2.2. Preparing CodeReady Workspaces Custom Resource for restricted environment	21
3.2.2.1. Downloading the default CheCluster Custom Resource	21
3.2.2.2. Customizing the CheCluster Custom Resource for restricted environment	21
3.2.3. Starting CodeReady Workspaces installation in a restricted environment using CodeReady Workspaces CLI management tool	22
3.3. PREPARING CODEREADY WORKSPACES CUSTOM RESOURCE FOR INSTALLING BEHIND A PROXY	22
CHAPTER 4. UPGRADING CODEREADY WORKSPACES	24
4.1. UPGRADING CODEREADY WORKSPACES USING OPERATORHUB	24
4.2. UPGRADING CODEREADY WORKSPACES USING THE CLI MANAGEMENT TOOL	24
4.3. KNOWN ISSUES	25
4.3.1. Updating a CodeReady Workspaces installation using the Operator	25
CHAPTER 5. ADVANCED CONFIGURATION OPTIONS FOR THE CODEREADY WORKSPACES SERVER COMPONENT	26
5.1. UNDERSTANDING CODEREADY WORKSPACES SERVER ADVANCED CONFIGURATION USING THE OPERATOR	26
5.2. CODEREADY WORKSPACES SERVER COMPONENT SYSTEM PROPERTIES REFERENCE	27
CHAPTER 6. UNINSTALLING CODEREADY WORKSPACES	59
6.1. UNINSTALLING CODEREADY WORKSPACES AFTER OPERATORHUB INSTALLATION	59
6.1.1. Uninstalling CodeReady Workspaces using the OpenShift web console	59
6.1.2. Uninstalling CodeReady Workspaces using oc commands	60
6.2. UNINSTALLING CODEREADY WORKSPACES AFTER CRWCTL INSTALLATION	61

CHAPTER 1. CONFIGURING THE CODEREADY WORKSPACES INSTALLATION

The following section describes configuration options to install Red Hat CodeReady Workspaces using the Operator.

1.1. UNDERSTANDING THE CHECLUSTER CUSTOM RESOURCE

A default deployment of CodeReady Workspaces consist in the application of a parametrized **CheCluster Custom Resource** by the Red Hat CodeReady Workspaces [Operator](#). The **CheCluster Custom Resource** is a YAML document describing the configuration of the overall CodeReady Workspaces installation. The Operator translates the **CheCluster Custom Resource** into configuration usable by each component of the CodeReady Workspaces installation: authentication, database, registry, server, storage.

For instance, to configure the main properties of the CodeReady Workspaces server component, the Operator generates a necessary **ConfigMap**, called **codeready**. And any change in the **ConfigMap** automatically triggers a restart of the CodeReady Workspaces Pod.

Additional resources

- [Understanding Operators](#).
- [Understanding Custom Resources](#).
- To learn how to modify the **CheCluster Custom Resource**, see the chosen installation procedure.

1.2. CHECLUSTER CUSTOM RESOURCE FIELDS REFERENCE

This section describes all fields available to customize the **CheCluster Custom Resource**.

- [Example 1.1, "A minimal **CheCluster Custom Resource** example."](#)
- [Table 1.3, "**CheCluster Custom Resource auth** configuration settings related to authentication used by CodeReady Workspaces installation"](#)
- [Table 1.2, "**CheCluster Custom Resource database** configuration settings related to the database used by CodeReady Workspaces"](#)
- [Table 1.1, "**CheCluster Custom Resource server** settings, related to the CodeReady Workspaces server component."](#)
- [Table 1.4, "**CheCluster Custom Resource storage** configuration settings related to persistent storage used by CodeReady Workspaces"](#)
- [Table 1.5, "**CheCluster Custom Resource k8s** configuration settings specific to CodeReady Workspaces installations on OpenShift"](#)
- [Table 1.6, "**CheCluster Custom Resource status** defines the observed state of CodeReady Workspaces installation"](#)

Example 1.1. A minimal **CheCluster Custom Resource example.**

```

apiVersion: org.eclipse.che/v1
kind: CheCluster
metadata:
  name: codeready-workspaces
spec:
  auth:
    externalIdentityProvider: false
  database:
    externalDb: false
  server:
    selfSignedCert: false
    gitSelfSignedCert: false
    tlsSupport: true
  storage:
    pvcStrategy: 'common'
    pvcClaimSize: '1Gi'

```

Table 1.1. **CheCluster** Custom **ResourceServer** settings, related to the CodeReady Workspaces server component.

Property	Default value	Description
airGapContainerRegistryHostName	omit	An optional host name or URL to an alternative container registry to pull images from. This value overrides the container registry host name 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 alternative 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.	A public host name 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.

Property	Default value	Description
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 CodeReady Workspaces workspaces. Omit or leave empty to use the default roles.
customCheProperties	omit	Map of additional environment variables that will be applied in the generated codeready-workspaces ConfigMap 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 ConfigMap 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).

Property	Default value	Description
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 will not use the configured proxy. Use <code> </code> as delimiter, for example localhost my.host.com 123.42.12.32 Only use when configuring a proxy is required (see also the proxyURL field).
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).

Property	Default value	Description
proxyURL	omit	URL (protocol+host name) 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.
tlsSupport	false	Instructs the Operator to deploy CodeReady Workspaces in TLS mode. Enabling TLS requires enabling the selfSignedCert field.

Table 1.2. CheCluster Custom Resource **database** configuration settings related to the database used by CodeReady Workspaces

Property	Default value	Description
chePostgresDb	dbche	PostgreSQL database name that the CodeReady Workspaces server uses to connect to the database.
chePostgresHostName	the Operator sets the value automatically	PostgreSQL Database host name 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	PostgreSQL password that the CodeReady Workspaces server uses to connect to the database.
chePostgresPort	5432	PostgreSQL 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	PostgreSQL user that the CodeReady Workspaces server uses to connect to the database.

Property	Default value	Description
externalDb	false	Instructs the Operator to deploy a dedicated database. By default, a dedicated PostgreSQL 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 PostgreSQL database deployment.
postgresImage	omit	Overrides the container image used in the PostgreSQL database deployment. This includes the image tag. Omit it or leave it empty to use the default container image provided by the Operator.

Table 1.3. CheCluster Custom Resource **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 must 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.

Property	Default value	Description
identityProvider Image	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.
identityProvider Password	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.
identityProvider PostgresPassword	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).
identityProvider Realm	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.
identityProvider URL	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.
openShiftoAuth	true on OpenShift	Enables the integration of the identity provider (Keycloak / RHSSO) with OpenShift OAuth. This allows users to log in with their OpenShift login and have their workspaces created under personal OpenShift projects. 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.

Table 1.4. CheCluster Custom Resource **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 PostgreSQL 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).
workspacePVCStorageClassName	omit	Storage class for the Persistent Volume Claims dedicated to the CodeReady Workspaces workspaces. Omit or leave empty to use a default storage class.

Table 1.5. **CheCluster Custom Resource 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).
securityContext FsGroup,omite mpty	1724	FSGroup the CodeReady Workspaces Pod and Workspace Pods containers run in.
securityContext RunAsUser	1724	ID of the user the CodeReady Workspaces Pod and Workspace Pods containers run as.

Property	Default value	Description
tlsSecretName	omit	Name of a secret that is used to set ingress TLS termination if TLS is enabled. See also the tlsSupport field.

Table 1.6. **CheCluster Custom Resource** **status** 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 PostgreSQL instance has been correctly provisioned.
devfileRegistryURL	Public URL to the Devfile registry.
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.
openShiftAuthProvisioned	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.

CHAPTER 2. INSTALLING CODEREADY WORKSPACES ON OPENSIFT CONTAINER PLATFORM

2.1. INSTALLING CODEREADY WORKSPACES USING THE CODEREADY WORKSPACES OPERATOR IN OPENSIFT 4 WEB CONSOLE

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

Operators are a method of packaging, deploying, and managing a OpenShift application which also provide the following:

- Repeatability of installation and upgrade.
- Constant health checks of every system component.
- Over-the-air (OTA) updates for OpenShift components and independent software vendor (ISV) content.
- A place to encapsulate knowledge from field engineers and spread it to all users.

Prerequisites

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

Procedure

1. Open the OpenShift web console.
2. To create the **Red Hat CodeReady Workspaces** project, in the left panel, navigate to the **Home → Projects** section.
3. Click the **Create Project** button.
4. In the **Create Project** pop-up window, enter the project details and validate.
 - **Name: CodeReady Workspaces.**
 - **Display Name: Red Hat CodeReady Workspaces.**
 - **Description: Red Hat CodeReady Workspaces.**
5. To install the **Red Hat CodeReady Workspaces** Operator, in the left panel, navigate to the **Operators → OperatorHub** section.
6. In the **Filter by keyword** field, type **Red Hat CodeReady Workspaces**.
7. Click the **Red Hat CodeReady Workspaces** tile.
8. In the **Red Hat CodeReady Workspaces** pop-up window, click the **Install** button .
9. On the **Install Operator** screen, choose following options and validate:
 - **Installation mode: A specific project on the cluster.**

- **Installed Namespace: CodeReady Workspaces.**

10. To create an instance of the **Red Hat CodeReady Workspaces** Operator, in the left panel, navigate to the **Operators → Installed Operators** section.
11. In the **Installed Operators** screen, click the **Red Hat CodeReady Workspaces** name.
12. In the **Operator Details** screen, in the **Details** tab, inside of the **Provided APIs** section, click the **Create Instance** link.
13. The **Create CheCluster** page contains the configuration of the overall CodeReady Workspaces instance to create. It is the **CheCluster** Custom Resource. For an installation using the default configuration, keep the default values. To modify the configuration, see [Configuring the CodeReady Workspaces installation](#).
14. To create the **codeready-workspaces** cluster, click the **Create** button in the lower left corner of the window.
15. On the **Operator Details** screen, in the **Red Hat CodeReady Workspaces Cluster** tab, click on the **codeready-workspaces** link.
16. To navigate to the **codeready-workspaces** instance, click the link under **Red Hat CodeReady Workspaces URL**.

Validation steps

1. To validate the installation of the **Red Hat CodeReady Workspaces** Operator, in the left panel, navigate to the **Operators → Installed Operators** section.
2. In the **Installed Operators** screen, click on the **Red Hat CodeReady Workspaces** name.
3. Navigate to the **Details** tab.
4. In the **ClusterServiceVersion Details** section at the bottom of the page, wait for these messages:
 - **Status: Succeeded.**
 - **Status Reason: install strategy completed with no errors.**
5. Navigate to the **Events** tab.
6. Wait for this message: **install strategy completed with no errors.**
7. To validate the installation of the **Red Hat CodeReady Workspaces** instance, navigate to the **CodeReady Workspaces Cluster** tab.
8. The **CheClusters** screen displays the list of **Red Hat CodeReady Workspaces** instances and their status.
9. Click **codeready-workspaces CheCluster** in the table.
10. Navigate to the **Details** tab.
11. Watch the content of following fields:
 - **Message:** the field contains error messages, if any. The expected content is **None**.

- **Red Hat CodeReady Workspaces URL:** displays the URL of the **Red Hat CodeReady Workspaces** instance, once the deployment is successful. An empty field means the deployment has not succeeded.
12. Navigate to the **Resources** tab.
 13. The screen displays the list of the resources assigned to the CodeReady Workspaces deployment.
 14. To see more details about the state of a resource, click its name and inspect the content of the available tabs.

Additional resources

- [the CodeReady Workspaces 2.2 End-user Guide](#).
- [the CodeReady Workspaces 2.2 Administration Guide](#).
- It is possible to use the **crwctl** utility script for deploying CodeReady Workspaces on OpenShift Container Platform and OpenShift Dedicated versions 4.4. This method is unofficial and serves as a backup installation method for situations where the installation method using OperatorHub is not available. See the [Installing CodeReady Workspaces on OpenShift 3 using the Operator](#) section.

2.2. INSTALLING CODEREADY WORKSPACES USING THE CLI MANAGEMENT TOOL ON OPENSIFT CONTAINER PLATFORM 3.11

2.2.1. 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 2.2.
3. Extract the archive to a folder, such as **\$(HOME)/crwctl** or **/opt/crwctl**.
4. Run the **crwctl** executable from the extracted folder. In this example, **\$(HOME)/crwctl/bin/crwctl version**.
5. Optionally, add the **bin** folder to your **\$PATH**, for example, **PATH=\$(PATH):\$(HOME)/crwctl/bin** to enable running **crwctl** without the full path specification.

Verification step

Running **crwctl version** displays the current version of the tool.

2.2.2. Installing CodeReady Workspaces on OpenShift 3 using the Operator

This section describes how to install CodeReady Workspaces on OpenShift 3 with the **crwctl** CLI management tool. The method of installation is using the Operator and enable TLS (HTTPS).

Operators are a method of packaging, deploying, and managing a OpenShift application which also provide the following:

- Repeatability of installation and upgrade.
- Constant health checks of every system component.
- Over-the-air (OTA) updates for OpenShift components and independent software vendor (ISV) content.
- A place to encapsulate knowledge from field engineers and spread it to all users.

TIP

This approach is only supported for use with OpenShift Container Platform and OpenShift Dedicated version 3.11, but also work for newer versions of OpenShift Container Platform and OpenShift Dedicated, and serves as a backup installation method for situations when the installation method using OperatorHub is not available.

Prerequisites

- Administrator rights on a running instance of OpenShift 3.11.
- An installation of the **oc** OpenShift 3.11 CLI management tool. See [Installing the OpenShift 3.11 CLI](#).
- An installation of the **crwctl** management tool. See [Using the **crwctl** management tool](#).
- To apply settings that the main **crwctl** command-line parameters cannot set, prepare a configuration file **operator-cr-patch.yaml** that will override the default values in the **CheCluster** Custom Resource used by the Operator. See [Configuring the CodeReady Workspaces installation](#).
- `<workspaces>` represents the project of the target installation.

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.rhpds311.openshift.opentlc.com httpd-ex      8080-tcp
None
```

4. Extract the domain from **httpd-ex-hello-world.apps.rhpds311.openshift.opentlc.com**. It is the part after the first name: **apps.rhpds311.openshift.opentlc.com**. Remember this URL as **<OPENSHIFT_APPS_URL>**.
5. Remove the dummy project:

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

6. To upgrade from a previous CodeReady Workspaces installation in the same OpenShift Container Platform 3.11 cluster, remove the Custom Resource Definition and the Cluster Roles:

```
$ oc delete customresourcedefinition/checlusters.org.eclipse.che
$ oc patch customresourcedefinition/checlusters.org.eclipse.che \
  --type merge \
  -p '{"metadata": {"finalizers": null }}'
$ oc delete clusterrole codeready-operator
```

7. To have multiple CodeReady Workspaces deployments in parallel using different versions in the same OpenShift Container Platform 3.11 cluster, create a new service account for the new deployment. It is, however, strongly recommended that you update all your old CodeReady Workspaces deployments to the latest version instead, as this mix of versions may cause unexpected and unsupported results.

```
$ oc patch clusterrolebinding codeready-operator \
  --type=json \
  -p '[{"op": "add", "path": "/subjects/0", "value": {"kind": "ServiceAccount", "namespace":
"<workspaces>", "name": "codeready-operator"}}]'
```

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

```
$ crwctl server:start -n <workspaces> --che-operator-cr-patch-yaml=operator-cr-patch.yaml
```

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: **https://codeready-<openshift_deployment_name>.<domain_name>**. The domain uses *Let's Encrypt* ACME certificates.

CHAPTER 3. INSTALLING CODEREADY WORKSPACES IN A RESTRICTED ENVIROMENT

By default, Red Hat CodeReady Workspaces uses 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.

The procedure for installing CodeReady Workspaces in restricted environments is different based on the installation method you use:

- [Installation using OperatorHub on Openshift 4.3](#) and above
- [Installation using the crwctl management tool on both OpenShift 3.11 or 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 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 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.

When the restricted network involves a proxy, follow the instructions provided in [Section 3.3, "Preparing CodeReady Workspaces Custom Resource for installing behind a proxy"](#).

3.1. INSTALLING CODEREADY WORKSPACES IN A RESTRICTED ENVIROMENT USING OPERATORHUB

Prerequisites

- A running OpenShift cluster. See the [OpenShift Container Platform 4.3 documentation](#) for instructions on how to install an OpenShift cluster on a restricted network.
- Access to the mirror registry used to installed the OpenShift disconnected cluster in restricted network. See the [Related OpenShift Container Platform 4.3 documentation about creating a mirror registry for installation in a restricted network.](#)

On disconnected OpenShift 4 clusters running on restricted networks, an Operator can be successfully installed from OperatorHub only if it meets the additional requirements defined in [Enabling your Operator for restricted network environments](#).

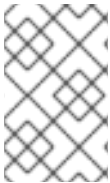
The CodeReady Workspaces operator meets these requirements and is therefore compatible with the [official documentation about OLM on a restricted network](#).

Procedure

To install CodeReady Workspaces from OperatorHub:

1. Build a **redhat-operators** catalog image. See [Building an Operator catalog image](#).
2. Configure OperatorHub to use this catalog image for operator installations. See [Configuring OperatorHub for restricted networks](#).
3. Proceed to the CodeReady Workspaces installation as usual as described in [Section 2.1, "Installing CodeReady Workspaces using the CodeReady Workspaces Operator in OpenShift 4 web console"](#).

3.2. INSTALLING CODEREADY WORKSPACES IN A RESTRICTED ENVIROMENT USING CLI MANAGEMENT TOOL



NOTE

Use CodeReady Workspaces CLI management tool to install CodeReady Workspaces on restricted networks only if installation through OperatorHub is not available. This method is not officially supported for OpenShift Container Platform 4.1 or later.

Prerequisites

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

3.2.1. 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

Every workspace launch requires infrastructure images from the following list:

- CodeReady Workspaces deployment and workspace support
 - registry.redhat.io/codeready-workspaces/crw-2-rhel8-operator:2.2
 - registry.redhat.io/codeready-workspaces/crw-2-rhel8-operator-metadata:2.2
 - registry.redhat.io/codeready-workspaces/devfileregistry-rhel8:2.2
 - registry.redhat.io/codeready-workspaces/server-rhel8:2.2
 - registry.redhat.io/codeready-workspaces/imagepuller-rhel8:2.2
 - registry.redhat.io/codeready-workspaces/jwtproxy-rhel8:2.2
 - registry.redhat.io/codeready-workspaces/pluginbroker-artifacts-rhel8:2.2
 - registry.redhat.io/codeready-workspaces/pluginbroker-metadata-rhel8:2.2
 - registry.redhat.io/codeready-workspaces/pluginregistry-rhel8:2.2
 - registry.redhat.io/redhat-sso-7/sso74-openshift-rhel8:7.4
 - registry.access.redhat.com/ubi8-minimal:8.2
- Plugins and editors
 - registry.redhat.io/codeready-workspaces/machineexec-rhel8:2.2
 - registry.redhat.io/codeready-workspaces/theia-rhel8:2.2
 - registry.redhat.io/codeready-workspaces/theia-endpoint-rhel8:2.2
- Workspace tooling
 - registry.redhat.io/rhsc/jboss-eap-7/eap73-openjdk8-openshift-rhel7:7.3.1
 - registry.redhat.io/rhel8/postgresql-96:1

Workspace-specific images

CodeReady Workspaces uses a subset of the following images to run a workspace. It is only necessary to include the images related to required technology stacks.

- Plugins
 - registry.redhat.io/codeready-workspaces/plugin-java8-rhel8:2.2
 - registry.redhat.io/codeready-workspaces/plugin-java11-rhel8:2.2
 - registry.redhat.io/codeready-workspaces/plugin-kubernetes-rhel8:2.2
 - registry.redhat.io/codeready-workspaces/plugin-openshift-rhel8:2.2
- Stacks
 - registry.redhat.io/codeready-workspaces/stacks-cpp-rhel8:2.2

- registry.redhat.io/codeready-workspaces/stacks-dotnet-rhel8:2.2
- registry.redhat.io/codeready-workspaces/stacks-golang-rhel8:2.2
- registry.redhat.io/codeready-workspaces/stacks-php-rhel8:2.2
- Workspace tooling
 - registry.redhat.io/rhsc/mongodb-36-rhel7:1-50

3.2.2. Preparing CodeReady Workspaces Custom Resource for restricted environment

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

3.2.2.1. Downloading the default CheCluster 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.

3.2.2.2. Customizing the CheCluster 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 3.2.1, “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 **CheCluster** 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 **CheCluster** Custom Resource:

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

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

3.2.3. 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-rhel8:2.2 \
  --che-operator-cr-yaml=org_v1_che_cr.yaml
```

3.3. PREPARING CODEREADY WORKSPACES CUSTOM RESOURCE FOR INSTALLING BEHIND A PROXY

This procedure describes how to provide necessary additional information to the **CheCluster** custom resource when installing CodeReady Workspaces behind a proxy.

Procedure

1. In the **CheCluster** 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:
    proxyURL: '<URL of the proxy, with the http protocol, and without the port>'
    proxyPort: '<Port of proxy, typically 3128>'
# [...]
```

- In addition to those basic settings, the proxy configuration usually requires adding the host of the external OpenShift cluster API URL in the list of the hosts to be accessed from CodeReady Workspaces without using the proxy.

To retrieve this cluster API host, run the following command against the OpenShift cluster:

```
$ oc whoami --show-server | sed 's#https://##' | sed 's#:.*$##'
```

The corresponding field of the **CheCluster** Custom Resource is **nonProxyHosts**. If a host already exists in this field, use | as a delimiter to add the cluster API host:

```
# [...]
spec:
  server:
    nonProxyHosts: 'anotherExistingHost|<cluster api host>'
# [...]
```

CHAPTER 4. UPGRADING CODEREADY WORKSPACES

This chapter describes how to upgrade a CodeReady Workspaces instance from previous minor version to CodeReady Workspaces 2.2.

The method used to install the CodeReady Workspaces instance determines the method to proceed with for the upgrade:

- [Section 4.1, “Upgrading CodeReady Workspaces using OperatorHub”](#)
- [Section 4.2, “Upgrading CodeReady Workspaces using the CLI management tool”](#)

4.1. UPGRADING CODEREADY WORKSPACES USING OPERATORHUB

This section describes how to upgrade from a previous minor version using the Operator from OperatorHub in the OpenShift web console.

Prerequisites

- An administrator account on an OpenShift instance.
- An instance of a previous minor version of CodeReady Workspaces, installed using the Operator from OperatorHub on the same instance of OpenShift.

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 the installed Operators.
4. Navigate to the **Subscription** tab and enable the following options:
 - **Channel: latest**
 - **Approval: Automatic**

Verification steps

1. Navigate to the CodeReady Workspaces instance.
2. The 2.2 version number is visible at the bottom of the page.

4.2. UPGRADING CODEREADY WORKSPACES USING THE CLI MANAGEMENT TOOL

This section describes how to upgrade from previous minor version using the CLI management tool.

Prerequisites

- An administrative account on an OpenShift instance.

- A running instance of a previous minor version of Red Hat CodeReady Workspaces, installed using the CLI management tool on the same instance of OpenShift.
- An installation of the **crwctl** management tool. See [Using the **crwctl** management tool](#).

Procedure

1. In all running workspaces in the CodeReady Workspaces 2.1 instance, save and push changes back to the Git repositories.
2. Shut down all workspaces in the CodeReady Workspaces 2.1 instance.
3. Run the following command:

```
$ crwctl server:update
```

Verification steps

1. Navigate to the CodeReady Workspaces instance.
2. The 2.2 version number is visible at the bottom of the page.

4.3. KNOWN ISSUES

4.3.1. 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 <codeready-cluster> --type=json -n <codeready-namespace> --patch '<requested-patch>'
```



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 FOR THE CODEREADY WORKSPACES SERVER COMPONENT

The following section describes advanced deployment and configuration methods for the CodeReady Workspaces server component.

5.1. UNDERSTANDING CODEREADY WORKSPACES SERVER ADVANCED CONFIGURATION USING THE OPERATOR

The following section describes the CodeReady Workspaces server component advanced configuration method for a deployment using the Operator.

Advanced configuration is necessary to:

- Add environment variables not automatically generated by the Operator from the standard **CheCluster** Custom Resource fields.
- Override the properties automatically generated by the Operator from the standard **CheCluster** Custom Resource fields.

The **customCheProperties** field, part of the **CheCluster** Custom Resource **server** settings, contains a map of additional environment variables to apply to the CodeReady Workspaces server component.

Example 5.1. 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: codeready-workspaces
  namespace: <workspaces>
spec:
  server:
    cheImageTag: ""
    devfileRegistryImage: ""
    pluginRegistryImage: ""
    tlsSupport: true
    selfSignedCert: false
    customCheProperties:
      CHE_WORKSPACE_DEFAULTMEMORYLIMIT__MB: "2048"
  auth:
# [...]
```

NOTE

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**.

Additional resources

- For the list of all parameters available in the **CheCluster** Custom Resource, see [Configuring the CodeReady Workspaces installation](#).
- For the list of all parameters available to configure **customCheProperties**, see [Section 5.2, “CodeReady Workspaces server component system properties reference”](#).

5.2. CODEREADY WORKSPACES SERVER COMPONENT SYSTEM PROPERTIES REFERENCE

The following document describes all possible configuration properties of the CodeReady Workspaces server component.

Table 5.1. Che server

Environment Variable Name	Default value	Description
CHE_DATABASE	<code>`\${che.home}/storage`</code>	Folder where CodeReady Workspaces will store internal data objects
CHE_API	<code>http://`\${CHE_HOST}`:`\${CHE_PORT}`/api</code>	API service. Browsers initiate REST communications to CodeReady Workspaces server with this URL
CHE_WEBSOCKET_ENDPOINT	<code>ws://`\${CHE_HOST}`:`\${CHE_PORT}`/api/websocket</code>	CodeReady Workspaces websocket major endpoint. Provides basic communication endpoint for major websocket interaction/messaging.
CHE_WEBSOCKET_ENDPOINT_MINOR	<code>ws://`\${CHE_HOST}`:`\${CHE_PORT}`/api/websocket-minor</code>	CodeReady Workspaces websocket minor endpoint. Provides basic communication endpoint for minor websocket interaction/messaging.
CHE_WORKSPACE_STORAGE	<code>`\${che.home}/workspaces`</code>	Your projects are synchronized from the CodeReady Workspaces server into the machine running each workspace. This is the directory in the ws runtime where your projects are mounted.
CHE_WORKSPACE_PROJECTS_STORAGE	<code>/projects</code>	Your projects are synchronized from the CodeReady Workspaces server into the machine running each workspace. This is the directory in the machine where your projects are placed.

Environment Variable Name	Default value	Description
CHE_WORKSPACE_PROJECTS_STORAGE_DEFAULT_SIZE	1Gi	Used when devfile OpenShift/os type components requests project PVC creation (applied in case of unique and perWorkspace PVC strategy. In case of common PVC strategy, it will be rewritten with value of <code>che.infra.kubernetes.pvc.quantity</code> property)
CHE_WORKSPACE_LOGS_ROOT_DIR	/workspace_logs	Defines the directory inside the machine where all the workspace logs are placed. The value of this folder should be provided into machine e.g. like environment variable so agents developers can use this directory for backup agents logs.
CHE_WORKSPACE_HTTP_PROXY	++	Configures proxies used by runtimes powering workspaces
CHE_WORKSPACE_HTTPS_PROXY	++	Configures proxies used by runtimes powering workspaces
CHE_WORKSPACE_NO_PROXY	++	Configures proxies used by runtimes powering workspaces
CHE_WORKSPACE_AUTO_START	true	By default, when users access to a workspace with its URL the workspace automatically starts if it is stopped. You can set this to false to disable this.
CHE_WORKSPACE_POOL_TYPE	fixed	Workspace threads pool configuration, this pool is used for workspace related operations that require asynchronous execution e.g. starting/stopping. Possible values are 'fixed', 'cached'
CHE_WORKSPACE_POOL_EXACT_SIZE	30	This property is ignored when pool type is different from 'fixed'. Configures the exact size of the pool, if it's set multiplier property is ignored. If this property is not set(0, < 0, NULL) then pool sized to number of cores, it can be modified within multiplier

Environment Variable Name	Default value	Description
CHE_WORKSPACE_POOL_CORES__MULTIPLIER	2	This property is ignored when pool type is different from 'fixed' or exact pool size is set. If it's set the pool size will be N_CORES * multiplier
CHE_WORKSPACE_PROBE__POOL__SIZE	10	This property specifies how much threads to use for workspaces servers liveness probes
CHE_WORKSPACE_HTTP__PROXY__JAVA__OPTIONS	NULL	Http proxy setting for workspace JVM
CHE_WORKSPACE_JAVA__OPTIONS	-XX:MaxRAM=150m - XX:MaxRAMFraction=2 - XX:+UseParallelGC - XX:MinHeapFreeRatio=10 - XX:MaxHeapFreeRatio=20 - XX:GCTimeRatio=4 - XX:AdaptiveSizePolicyWeight=90 - Dsun.zip.disableMemoryMapping=true -Xms20m - Djava.security.egd=file:/dev/. /urandom	Java command line options to be added to JVM's that running within workspaces.
CHE_WORKSPACE_MAVEN__OPTIONS	-XX:MaxRAM=150m - XX:MaxRAMFraction=2 - XX:+UseParallelGC - XX:MinHeapFreeRatio=10 - XX:MaxHeapFreeRatio=20 - XX:GCTimeRatio=4 - XX:AdaptiveSizePolicyWeight=90 - Dsun.zip.disableMemoryMapping=true -Xms20m - Djava.security.egd=file:/dev/. /urandom	Maven command line options added to JVM's that run agents within workspaces.

Environment Variable Name	Default value	Description
CHE_WORKSPACE_MAVEN_SERVER_JAVA_OPTIONS	-XX:MaxRAM=128m - XX:MaxRAMFraction=1 - XX:+UseParallelGC - XX:MinHeapFreeRatio=10 - XX:MaxHeapFreeRatio=20 - XX:GCTimeRatio=4 - XX:AdaptiveSizePolicyWeight=90 - Dsun.zip.disableMemoryMapping=true -Xms20m - Djava.security.egd=file:/dev/. /urandom	Default java command line options to be added to JVM that run maven server.
CHE_WORKSPACE_DEFAULT_MEMORY_LIMIT_MB	1024	RAM limit default for each machine that has no RAM settings in environment. Value less or equal to 0 interpreted as limit disabling.
CHE_WORKSPACE_DEFAULT_MEMORY_REQUEST_MB	200	RAM request default for each container that has no explicit RAM settings in environment. this amount will be allocated on workspace container creation this property might not be supported by all infrastructure implementations: currently it is supported by OpenShift and OpenShift Container Platform if default memory request is more than the memory limit, request will be ignored, and only limit will be used. Value less or equal to 0 interpreted as disabling request.
CHE_WORKSPACE_DEFAULT_CPU_LIMIT_CORES	-1	CPU limit default for each container that has no CPU settings in environment. Can be specified either in floating point cores number, e.g. 0.125 or in K8S format integer millicores e.g. 125m Value less or equal to 0 interpreted as limit disabling.

Environment Variable Name	Default value	Description
CHE_WORKSPACE_DEFAULT_CPU_REQUEST_CORES	-1	CPU request default for each container that has no CPU settings in environment. if default CPU request is more than the CPU limit, request will be ignored, and only limit will be used. Value less or equal to 0 interpreted as disabling this request.
CHE_WORKSPACE_SIDECAR_DEFAULT_MEMORY_LIMIT_MB	128	RAM limit and request default for each sidecar that has no RAM settings in CodeReady Workspaces plugin configuration. Value less or equal to 0 interpreted as limit disabling.
CHE_WORKSPACE_SIDECAR_DEFAULT_MEMORY_REQUEST_MB	64	RAM limit and request default for each sidecar that has no RAM settings in {prod-short} plugin configuration. Value less or equal to 0 interpreted as limit disabling.
CHE_WORKSPACE_SIDECAR_DEFAULT_CPU_LIMIT_CORES	-1	CPU limit and request default for each sidecar that has no CPU settings in CodeReady Workspaces plugin configuration. Can be specified either in floating point cores number, e.g. 0.125 or in K8S format integer millicores e.g. 125m Value less or equal to 0 interpreted as disabling limit.
CHE_WORKSPACE_SIDECAR_DEFAULT_CPU_REQUEST_CORES	-1	CPU limit and request default for each sidecar that has no CPU settings in {prod-short} plugin configuration. Can be specified either in floating point cores number, e.g. 0.125 or in K8S format integer millicores e.g. 125m Value less or equal to 0 interpreted as disabling limit.

Environment Variable Name	Default value	Description
CHE_WORKSPACE_SIDECA R_IMAGE_PULL_POLICY	Always	Define image pulling strategy for sidecars. Possible values are: Always, Never, IfNotPresent. Any other value will be interpreted as unspecified policy (Always if :latest tag is specified, or IfNotPresent otherwise.)
CHE_WORKSPACE_ACTIVIT Y_CHECK_SCHEDULER_ PERIOD__S	60	Period of inactive workspaces suspend job execution.
CHE_WORKSPACE_ACTIVIT Y_CLEANUP_SCHEDULER __PERIOD__S	3600	The period of the cleanup of the activity table. The activity table can contain invalid or stale data if some unforeseen errors happen, like a server crash at a peculiar point in time. The default is to run the cleanup job every hour.
CHE_WORKSPACE_ACTIVIT Y_CLEANUP_SCHEDULER __INITIAL_DELAY__S	60	The delay after server startup to start the first activity clean up job.
CHE_WORKSPACE_ACTIVIT Y_CHECK_SCHEDULER_ DELAY__S	180	Delay before first workspace idleness check job started to avoid mass suspend if ws master was unavailable for period close to inactivity timeout.
CHE_WORKSPACE_CLEANU P_TEMPORARY_INITIAL_ DELAY__MIN	5	Period of stopped temporary workspaces cleanup job execution.
CHE_WORKSPACE_CLEANU P_TEMPORARY_PERIOD_ __MIN	180	Period of stopped temporary workspaces cleanup job execution.
CHE_WORKSPACE_SERVER _PING_SUCCESS_THRES HOLD	1	Number of sequential successful pings to server after which it is treated as available. Note: the property is common for all servers e.g. workspace agent, terminal, exec etc.
CHE_WORKSPACE_SERVER _PING_INTERVAL_MILLIS ECONDS	3000	Interval, in milliseconds, between successive pings to workspace server.

Environment Variable Name	Default value	Description
CHE_WORKSPACE_SERVER_LIVENESS_PROBES	wsagent/http,exec-agent/http,terminal,theia,jupyter,dirigible,cloud-shell	List of servers names which require liveness probes
CHE_WORKSPACE_STARTUP_DEBUG_LOG_LIMIT_BYTES	10485760	Limit size of the logs collected from single container that can be observed by che-server when debugging workspace startup. default 10MB=10485760
CHE_WORKSPACE_STOP_ROLE_ENABLED	true	If true, 'stop-workspace' role with the edit privileges will be granted to the 'che' ServiceAccount if OpenShift OAuth is enabled. This configuration is mainly required for workspace idling when the OpenShift OAuth is enabled.

Table 5.2. Templates

Environment Variable Name	Default value	Description
CHE_TEMPLATE_STORAGE	\${che.home}/templates	Folder that contains JSON files with code templates and samples

Table 5.3. Authentication parameters

Environment Variable Name	Default value	Description
CHE_AUTH_USER_SELF_CREATION	false	CodeReady Workspaces has a single identity implementation, so this does not change the user experience. If true, enables user creation at API level
CHE_AUTH_ACCESS_DENIED_ERROR_PAGE	/error-oauth	Authentication error page address
CHE_AUTH_RESERVED_USER_NAMES	++	Reserved user names
CHE_OAUTH_GITHUB_CLIENTID	NULL	You can setup GitHub OAuth to automate authentication to remote repositories. You need to first register this application with GitHub OAuth.

Environment Variable Name	Default value	Description
CHE_OAUTH_GITHUB_CLIENTSECRET	NULL	You can setup GitHub OAuth to automate authentication to remote repositories. You need to first register this application with GitHub OAuth.
CHE_OAUTH_GITHUB_AUTH_URI	https://github.com/login/oauth/authorize	You can setup GitHub OAuth to automate authentication to remote repositories. You need to first register this application with GitHub OAuth.
CHE_OAUTH_GITHUB_TOKEN_URI	https://github.com/login/oauth/access_token	You can setup GitHub OAuth to automate authentication to remote repositories. You need to first register this application with GitHub OAuth.
CHE_OAUTH_GITHUB_REDIRECT_URI	http://localhost:\${CHE_PORT}/api/oauth/callback	You can setup GitHub OAuth to automate authentication to remote repositories. You need to first register this application with GitHub OAuth.
CHE_OAUTH_OPENSHIFT_CLIENTID	NULL	Configuration of OpenShift OAuth client. Used to obtain OpenShift OAuth token.
CHE_OAUTH_OPENSHIFT_CLIENTSECRET	NULL	Configuration of OpenShift OAuth client. Used to obtain OpenShift OAuth token.
CHE_OAUTH_OPENSHIFT_OAUTH_ENDPOINT	NULL	Configuration of OpenShift OAuth client. Used to obtain OpenShift OAuth token.
CHE_OAUTH_OPENSHIFT_VERIFY_TOKEN_URL	NULL	Configuration of OpenShift OAuth client. Used to obtain OpenShift OAuth token.

Table 5.4. Internal

Environment Variable Name	Default value	Description
---------------------------	---------------	-------------

Environment Variable Name	Default value	Description
SCHEDULE_CORE__POOL__ _SIZE	10	CodeReady Workspaces extensions can be scheduled executions on a time basis. This configures the size of the thread pool allocated to extensions that are launched on a recurring schedule.
ORG_EVERREST_ASYNCCHR ONOUS	false	Everrest is a Java Web Services toolkit that manages JAX-RS & web socket communications. Users should rarely need to configure this. Disable asynchronous mechanism that is embedded in everrest.
ORG_EVERREST_ASYNCCHR ONOUS_POOL_SIZE	20	Quantity of asynchronous requests which may be processed at the same time
ORG_EVERREST_ASYNCCHR ONOUS_QUEUE_SIZE	500	Size of queue. If asynchronous request can't be processed after consuming it will be added in queue.
ORG_EVERREST_ASYNCCHR ONOUS_JOB_TIMEOUT	10	Timeout in minutes for request. If after timeout request is not done or client did not come yet to get result of request it may be discarded.
ORG_EVERREST_ASYNCCHR ONOUS_CACHE_SIZE	1024	Size of cache for waiting, running and ended request.
ORG_EVERREST_ASYNCCHR ONOUS_SERVICE_PATH	/async/	Path to asynchronous service
DB_SCHEMA_FLYWAY_BAS ELINE_ENABLED	true	DB initialization and migration configuration
DB_SCHEMA_FLYWAY_BAS ELINE_VERSION	5.0.0.8.1	DB initialization and migration configuration
DB_SCHEMA_FLYWAY_SCRI PTS_PREFIX	++	DB initialization and migration configuration
DB_SCHEMA_FLYWAY_SCRI PTS_SUFFIX	.sql	DB initialization and migration configuration

Environment Variable Name	Default value	Description
DB_SCHEMA_FLYWAY_SCRIPTS_VERSION_SEPARATOR	—	DBinitializationandmigrationconfiguration
DB_SCHEMA_FLYWAY_SCRIPTS_LOCATIONS	classpath:che-schema	DBinitializationandmigrationconfiguration

Table 5.5. Kubernetes Infra parameters

Environment Variable Name	Default value	Description
CHE_INFRA_KUBERNETES_MASTER_URL	++	Configuration of Kubernetes client that Infra will use
CHE_INFRA_KUBERNETES_TRUST_CERTS	++	Configuration of Kubernetes client that Infra will use
CHE_INFRA_KUBERNETES_SERVER_STRATEGY	default-host	Defines the way how servers are exposed to the world in OpenShift infra. List of strategies implemented in CodeReady Workspaces: default-host, multi-host, single-host
CHE_INFRA_KUBERNETES_INGRESS_DOMAIN	++	Used to generate domain for a server in a workspace in case property che.infra.kubernetes.server_strategy is set to multi-host

Environment Variable Name	Default value	Description
CHE_INFRA_KUBERNETES_NAMESPACE	++	DEPRECATED - please do not change the value of this property otherwise the existing workspaces will loose data. Do not set it on new installations. Defines Kubernetes namespace in which all workspaces will be created. If not set, every workspace will be created in a new namespace, where namespace = workspace id It's possible to use <username> and <userid> placeholders (e.g.: che-workspace-<username>). In that case, new namespace will be created for each user. Service account with permission to create new namespace must be used. Ignored for OpenShift infra. Use che.infra.openshift.project instead If the namespace pointed to by this property exists, it will be used for all workspaces. If it does not exist, the namespace specified by the che.infra.kubernetes.namespace.default will be created and used.
CHE_INFRA_KUBERNETES_NAMESPACE_DEFAULT	<username>-che	Defines Kubernetes default namespace in which user's workspaces are created if user does not override it. It's possible to use <username>, <userid> and <workspaceid> placeholders (e.g.: che-workspace-<username>). In that case, new namespace will be created for each user (or workspace). Is used by OpenShift infra as well to specify Project
CHE_INFRA_KUBERNETES_NAMESPACE_ALLOW_USER_DEFINED	false	Defines if a user is able to specify Kubernetes namespace (or OpenShift project) different from the default. It's NOT RECOMMENDED to configured true without OAuth configured. This property is also used by the OpenShift infra.

Environment Variable Name	Default value	Description
CHE_INFRA_KUBERNETES_SERVICE_ACCOUNT_NAME	NULL	Defines Kubernetes Service Account name which should be specified to be bound to all workspaces pods. Note that Kubernetes Infrastructure won't create the service account and it should exist. OpenShift infrastructure will check if project is predefined(if che.infra.openshift.project is not empty): - if it is predefined then service account must exist there - if it is 'NULL' or empty string then infrastructure will create new OpenShift project per workspace and prepare workspace service account with needed roles there
CHE_INFRA_KUBERNETES_WORKSPACE_SA_CLUSTER_ROLES	NULL	Specifies optional, additional cluster roles to use with the workspace service account. Note that the cluster role names must already exist, and the CodeReady Workspaces service account needs to be able to create a Role Binding to associate these cluster roles with the workspace service account. The names are comma separated. This property deprecates 'che.infra.kubernetes.cluster_role_name'.
CHE_INFRA_KUBERNETES_WORKSPACE_START_TIMEOUT_MIN	8	Defines time frame that limits the Kubernetes workspace start time
CHE_INFRA_KUBERNETES_INGRESS_START_TIMEOUT_MIN	5	Defines the timeout in minutes that limits the period for which Kubernetes Ingress become ready

Environment Variable Name	Default value	Description
CHE_INFRA_KUBERNETES_WORKSPACE__UNRECOVERABLE__EVENTS	FailedMount,FailedScheduling,MountVolume.SetUpfailed,Failed to pull image,FailedCreate	If during workspace startup an unrecoverable event defined in the property occurs, terminate workspace immediately instead of waiting until timeout Note that this SHOULD NOT include a mere 'Failed' reason, because that might catch events that are not unrecoverable. A failed container startup is handled explicitly by CodeReady Workspaces server.
CHE_INFRA_KUBERNETES_PVC_ENABLED	true	Defines whether use the Persistent Volume Claim for the workspace needs e.g backup projects, logs etc or disable it.
CHE_INFRA_KUBERNETES_PVC_STRATEGY	common	Defined which strategy will be used while choosing PVC for workspaces. Supported strategies: - 'common' All workspaces in the same Kubernetes Namespace will reuse the same PVC. Name of PVC may be configured with 'che.infra.kubernetes.pvc.name'. Existing PVC will be used or new one will be created if it doesn't exist. - 'unique' Separate PVC for each workspace's volume will be used. Name of PVC is evaluated as '{che.infra.kubernetes.pvc.name} + '-' + `{generated_8_chars}`'. Existing PVC will be used or a new one will be created if it doesn't exist. - 'per-workspace' Separate PVC for each workspace will be used. Name of PVC is evaluated as '{che.infra.kubernetes.pvc.name} + '-' + `{WORKSPACE_ID}`'. Existing PVC will be used or a new one will be created if it doesn't exist.

Environment Variable Name	Default value	Description
CHE_INFRA_KUBERNETES_PVC_PRECREATE__SUBPATHS	true	Defines whether to run a job that creates workspace's subpath directories in persistent volume for the 'common' strategy before launching a workspace. Necessary in some versions of OpenShift/Kubernetes as workspace subpath volume mounts are created with root permissions, and thus cannot be modified by workspaces running as a user (presents an error importing projects into a workspace in CodeReady Workspaces). The default is 'true', but should be set to false if the version of Openshift/Kubernetes creates subdirectories with user permissions. Relevant issue: https://github.com/kubernetes/kubernetes/issues/41638 Note that this property has effect only if the 'common' PVC strategy used.
CHE_INFRA_KUBERNETES_PVC_NAME	claim-che-workspace	Defines the settings of PVC name for che workspaces. Each PVC strategy supplies this value differently. See doc for <code>che.infra.kubernetes.pvc.strategy</code> property
CHE_INFRA_KUBERNETES_PVC_STORAGE__CLASS__NAME	++	Defines the storage class of Persistent Volume Claim for the workspaces. Empty strings means 'use default'.
CHE_INFRA_KUBERNETES_PVC_QUANTITY	10Gi	Defines the size of Persistent Volume Claim of che workspace. Format described here: https://docs.openshift.com/container-platform/4.4/storage/understanding-persistent-storage.html
CHE_INFRA_KUBERNETES_PVC_JOBS_IMAGE	centos:centos7	Pod that is launched when performing persistent volume claim maintenance jobs on OpenShift

Environment Variable Name	Default value	Description
CHE_INFRA_KUBERNETES_PVC_JOBS_IMAGE_PULL_POLICY	IfNotPresent	Image pull policy of container that used for the maintenance jobs on Kubernetes/OpenShift cluster
CHE_INFRA_KUBERNETES_PVC_JOBS_MEMORYLIMIT	250Mi	Defines pod memory limit for persistent volume claim maintenance jobs
CHE_INFRA_KUBERNETES_PVC_ACCESS_MODE	ReadWriteOnce	Defines Persistent Volume Claim access mode. Note that for common PVC strategy changing of access mode affects the number of simultaneously running workspaces. If OpenShift flavor where che running is using PVs with RWX access mode then a limit of running workspaces at the same time bounded only by che limits configuration like(RAM, CPU etc). Detailed information about access mode is described here: https://docs.openshift.com/container-platform/4.4/storage/understanding-persistent-storage.html
CHE_INFRA_KUBERNETES_PVC_WAIT_BOUND	true	Defines whether CodeReady Workspaces Server should wait workspaces PVCs to become bound after creating. It's used by all PVC strategies. It should be set to false in case if volumeBindingMode is configured to WaitForFirstConsumer otherwise workspace starts will hangs up on phase of waiting PVCs. Default value is true (means that PVCs should be waited to be bound)
CHE_INFRA_KUBERNETES_INSTALLER_SERVER_MIN_PORT	10000	Defined range of ports for installers servers By default, installer will use own port, but if it conflicts with another installer servers then OpenShift infrastructure will reconfigure installer to use first available from this range

Environment Variable Name	Default value	Description
CHE_INFRA_KUBERNETES_INSTALLER_SERVER_MAX_PORT	20000	Defined range of ports for installers servers. By default, installer will use own port, but if it conflicts with another installer servers then OpenShift infrastructure will reconfigure installer to use first available from this range.
CHE_INFRA_KUBERNETES_INGRESS_ANNOTATIONS_JSON	NULL	<p>Defines annotations for ingresses which are used for servers exposing. Value depends on the kind of ingress controller. OpenShift infrastructure ignores this property because it uses Routes instead of ingresses. Note that for a single-host deployment strategy to work, a controller supporting URL rewriting has to be used (so that URLs can point to different servers while the servers don't need to support changing the app root). The <code>che.infra.kubernetes.ingress.path.rewrite_transform</code> property defines how the path of the ingress should be transformed to support the URL rewriting and this property defines the set of annotations on the ingress itself that instruct the chosen ingress controller to actually do the URL rewriting, potentially building on the path transformation (if required by the chosen ingress controller). For example for nginx ingress controller 0.22.0 and later the following value is recommended:</p> <pre>{'ingress.kubernetes.io/rewrite-target': '/\$1','ingress.kubernetes.io/ssl-redirect': 'false',\ 'ingress.kubernetes.io/proxy-connect-timeout': '3600','ingress.kubernetes.io/proxy-read-timeout': '3600'}</pre> <p>and the <code>che.infra.kubernetes.ingress.path.rewrite_transform</code> should be set to <code>'%(s.*)'</code>. For nginx ingress controller older than 0.22.0, the <code>rewrite-target</code> should be set to</p>

Environment Variable Name	Default value	Description
		<p>merely '/' and the path transform to '%s' (see the the che.infra.kubernetes.ingress.path.rewrite_transform property). Please consult the nginx ingress controller documentation for the explanation of how the ingress controller uses the regular expression present in the ingress path and how it achieves the URL rewriting.</p>
CHE_INFRA_KUBERNETES_INGRESS_PATH_TRANSFORM	NULL	<p>Defines a 'recipe' on how to declare the path of the ingress that should expose a server. The '%s' represents the base public URL of the server and is guaranteed to end with a forward slash. This property must be a valid input to the String.format() method and contain exactly one reference to '%s'. Please see the description of the che.infra.kubernetes.ingress.annotations_json property to see how these two properties interplay when specifying the ingress annotations and path. If not defined, this property defaults to '%s' (without the quotes) which means that the path is not transformed in any way for use with the ingress controller.</p>
CHE_INFRA_KUBERNETES_POD_SECURITY_CONTEXT_RUN_AS_USER	NULL	<p>Defines security context for pods that will be created by Kubernetes Infra This is ignored by OpenShift infra</p>
CHE_INFRA_KUBERNETES_POD_SECURITY_CONTEXT_FS_GROUP	NULL	<p>Defines security context for pods that will be created by Kubernetes Infra This is ignored by OpenShift infra</p>

Environment Variable Name	Default value	Description
CHE_INFRA_KUBERNETES_POD_TERMINATION_GRACE_PERIOD_SEC	0	Defines grace termination period for pods that will be created by Kubernetes / OpenShift infrastructures Grace termination period of Kubernetes / OpenShift workspace's pods defaults '0', which allows to terminate pods almost instantly and significantly decrease the time required for stopping a workspace. Note: if terminationGracePeriodSeconds have been explicitly set in Kubernetes / OpenShift recipe it will not be overridden.
CHE_INFRA_KUBERNETES_CLIENT_HTTP_ASYNC_REQUESTS_MAX	1000	Number of maximum concurrent async web requests (http requests or ongoing web socket calls) supported in the underlying shared http client of the KubernetesClient instances. Default values are 64, and 5 per-host, which doesn't seem correct for multi-user scenarios knowing that CodeReady Workspaces keeps a number of connections opened (e.g. for command or ws-agent logs)
CHE_INFRA_KUBERNETES_CLIENT_HTTP_ASYNC_REQUESTS_MAX_PER_HOST	1000	Number of maximum concurrent async web requests (http requests or ongoing web socket calls) supported in the underlying shared http client of the KubernetesClient instances. Default values are 64, and 5 per-host, which doesn't seem correct for multi-user scenarios knowing that {prod-short} keeps a number of connections opened (e.g. for command or ws-agent logs)
CHE_INFRA_KUBERNETES_CLIENT_HTTP_CONNECTION_POOL_MAX_IDLE	5	Max number of idle connections in the connection pool of the Kubernetes-client shared http client

Environment Variable Name	Default value	Description
CHE_INFRA_KUBERNETES_CLIENT_HTTP_CONNECTION_POOL_KEEP_ALIVE_MIN	5	Keep-alive timeout of the connection pool of the Kubernetes-client shared http client in minutes
CHE_INFRA_KUBERNETES_TLS_ENABLED	false	Creates Ingresses with Transport Layer Security (TLS) enabled In OpenShift infrastructure, Routes will be TLS-enabled
CHE_INFRA_KUBERNETES_TLS_SECRET	++	Name of a secret that should be used when creating workspace ingresses with TLS Ignored by OpenShift infrastructure
CHE_INFRA_KUBERNETES_TLS_KEY	NULL	Data for TLS Secret that should be used for workspaces Ingresses cert and key should be encoded with Base64 algorithm These properties are ignored by OpenShift infrastructure
CHE_INFRA_KUBERNETES_TLS_CERT	NULL	Datafor TLS Secret that should be used for workspaces Ingresses cert and key should be encoded with Base64 algorithm These properties are ignored by OpenShift infrastructure

Environment Variable Name	Default value	Description
CHE_INFRA_KUBERNETES_RUNTIMES_CONSISTENCY_CHECK_PERIOD_MIN	-1	Defines the period with which runtimes consistency checks will be performed. If runtime has inconsistent state then runtime will be stopped automatically. Value must be more than 0 or -1 , where -1 means that checks won't be performed at all. It is disabled by default because there is possible CodeReady Workspaces Server configuration when CodeReady Workspaces Server doesn't have an ability to interact with Kubernetes API when operation is not invoked by user. It DOES work on the following configurations: - workspaces objects are created in the same namespace where CodeReady Workspaces Server is located; - cluster-admin service account token is mount to CodeReady Workspaces Server pod; It DOES NOT work on the following configurations: - CodeReady Workspaces Server communicates with Kubernetes API using token from OAuth provider;

Table 5.6. OpenShift Infra parameters

Environment Variable Name	Default value	Description
---------------------------	---------------	-------------

Environment Variable Name	Default value	Description
CHE_INFRA_OPENSIFT_PROJECT	++	DEPRECATED - please do not change the value of this property otherwise the existing workspaces will loose data. Do not set it on new installations. Defines OpenShift namespace in which all workspaces will be created. If not set, every workspace will be created in a new project, where project name = workspace id It's possible to use <username> and <userid> placeholders (e.g.: che-workspace-<username>). In that case, new project will be created for each user. OpenShift oauth or service account with permission to create new projects must be used. If the project pointed to by this property exists, it will be used for all workspaces. If it does not exist, the namespace specified by the che.infra.kubernetes.namespace.default will be created and used.
CHE_SINGLEPORT_WILDCARD__DOMAIN_HOST	NULL	Single port mode wildcard domain host & port. nip.io is used by default
CHE_SINGLEPORT_WILDCARD__DOMAIN_PORT	NULL	Singleport mode wildcard domain host & port. nip.io is used by default
CHE_SINGLEPORT_WILDCARD__DOMAIN_IPLESS	false	Enable single port custom DNS without inserting the IP

Table 5.7. Experimental properties

Environment Variable Name	Default value	Description
CHE_WORKSPACE_PLUGIN__BROKER_METADATA_IMAGE	quay.io/eclipse/che-plugin-metadata-broker:v3.2.0	Docker image of CodeReady Workspaces plugin broker app that resolves workspace tooling configuration and copies plugins dependencies to a workspace

Environment Variable Name	Default value	Description
CHE_WORKSPACE_PLUGIN__BROKER_ARTIFACTS_IMAGE	quay.io/eclipse/che-plugin-artifacts-broker:v3.2.0	Dockerimage of {prod-short} plugin broker app that resolves workspace tooling configuration and copies plugins dependencies to a workspace
CHE_WORKSPACE_PLUGIN__BROKER_PULL_POLICY	Always	Docker image of CodeReady Workspaces plugin broker app that resolves workspace tooling configuration and copies plugins dependencies to a workspace
CHE_WORKSPACE_PLUGIN__BROKER_WAIT_TIMEOUT_MIN	3	Defines the timeout in minutes that limits the max period of result waiting for plugin broker.
CHE_WORKSPACE_PLUGIN__REGISTRY_URL	https://che-plugin-registry.prod-preview.openshift.io/v3	Workspace tooling plugins registry endpoint. Should be a valid HTTP URL. Example: http://che-plugin-registry-eclipse-che.192.168.65.2.nip.io In case CodeReady Workspaces plugins tooling is not needed value 'NULL' should be used
CHE_WORKSPACE_DEVFILE__REGISTRY_URL	https://che-devfile-registry.prod-preview.openshift.io/	Devfile Registry endpoint. Should be a valid HTTP URL. Example: http://che-devfile-registry-eclipse-che.192.168.65.2.nip.io In case CodeReady Workspaces plugins tooling is not needed value 'NULL' should be used
CHE_WORKSPACE_PERSIST__VOLUMES_DEFAULT	true	Defines a default value for persist volumes that clients like Dashboard should propose for users during workspace creation. Possible values: true or false In case of true - PersistentVolumeClaims are used by declared volumes by user and plugins. true value is supposed not to be set explicitly in Devfile attributes since it's default fixed behaviour. In case of false - emptyDir is used instead of PVCs. Note that data will be lost after workspace restart.

Environment Variable Name	Default value	Description
CHE_SERVER_SECURE_EX POSER	default	Configures in which way secure servers will be protected with authentication. Suitable values: - 'default': jwtproxy is configured in a pass-through mode. So, servers should authenticate requests themselves. - 'jwtproxy': jwtproxy will authenticate requests. So, servers will receive only authenticated ones.
CHE_SERVER_SECURE_EX POSER_JWTPROXY_TOKEN _ISSUER	wsmaster	Jwtproxy issuer string, token lifetime and optional auth page path to route unsigned requests to.
CHE_SERVER_SECURE_EX POSER_JWTPROXY_TOKEN _TTL	8800h	Jwtproxyissuer string, token lifetime and optional auth page path to route unsigned requests to.
CHE_SERVER_SECURE_EX POSER_JWTPROXY_AUTH_ LOADER_PATH	/_app/loader.html	Jwtproxyissuerstring, token lifetime and optional auth page path to route unsigned requests to.
CHE_SERVER_SECURE_EX POSER_JWTPROXY_IMAGE	quay.io/eclipse/che- jwtproxy:fd94e60	Jwtproxyissuerstring,token lifetime and optional auth page path to route unsigned requests to.
CHE_SERVER_SECURE_EX POSER_JWTPROXY_MEMOR Y_LIMIT	128mb	Jwtproxyissuerstring,tokenlifetim e and optional auth page path to route unsigned requests to.
CHE_SERVER_SECURE_EX POSER_JWTPROXY_CPU_ LIMIT	0.5	Jwtproxyissuerstring,tokenlifetim eand optional auth page path to route unsigned requests to.

Table 5.8. Configuration of major "/websocket" endpoint

Environment Variable Name	Default value	Description
CHE_CORE_JSONRPC_PRO CESSOR_MAX_POOL_ SIZE	50	Maximum size of the JSON RPC processing pool in case if pool size would be exceeded message execution will be rejected

Environment Variable Name	Default value	Description
CHE_CORE_JSONRPC_PROCESSOR_CORE_POOL_SIZE	5	Initial json processing pool. Minimum number of threads that used to process major JSON RPC messages.
CHE_CORE_JSONRPC_PROCESSOR_QUEUE_CAPACITY	100000	Configuration of queue used to process Json RPC messages.

Table 5.9. Configuration of major "/websocket-minor" endpoint

Environment Variable Name	Default value	Description
CHE_CORE_JSONRPC_MINOR_PROCESSOR_MAX_POOL_SIZE	100	Maximum size of the JSON RPC processing pool in case if pool size would be exceeded message execution will be rejected
CHE_CORE_JSONRPC_MINOR_PROCESSOR_CORE_POOL_SIZE	15	Initial json processing pool. Minimum number of threads that used to process minor JSON RPC messages.
CHE_CORE_JSONRPC_MINOR_PROCESSOR_QUEUE_CAPACITY	10000	Configuration of queue used to process Json RPC messages.
CHE_METRICS_PORT	8087	Port the the http server endpoint that would be exposed with Prometheus metrics

Table 5.10. CORS settings

Environment Variable Name	Default value	Description
CHE_CORS_ALLOWED_ORIGINS	*	CORS filter on WS Master is turned off by default. Use environment variable 'CHE_CORS_ENABLED=true' to turn it on 'cors.allowed.origins' indicates which request origins are allowed
CHE_CORS_ALLOW_CREDENTIALS	false	'cors.support.credentials' indicates if it allows processing of requests with credentials (in cookies, headers, TLS client certificates)

Table 5.11. Factory defaults

Environment Variable Name	Default value	Description
CHE_FACTORY_DEFAULT__EDITOR	eclipse/che-theia/next	Editor and plugin which will be used for factories which are created from remote git repository which doesn't contain any CodeReady Workspaces-specific workspace descriptors (like .devfile or .factory.json) Multiple plugins must be comma-separated, for example: pluginFooPublisher/pluginFooName/pluginFooVersion,pluginBarPublisher/pluginBarName/pluginBarVersion
CHE_FACTORY_DEFAULT__PLUGINS	eclipse/che-machine-exec-plugin/nightly	Editor and plugin which will be used for factories which are created from remote git repository which doesn't contain any {prod-short} -specific workspace descriptors (like .devfile or .factory.json) Multiple plugins must be comma-separated, for example: pluginFooPublisher/pluginFooName/pluginFooVersion,pluginBarPublisher/pluginBarName/pluginBarVersion

Table 5.12. Devfile defaults

Environment Variable Name	Default value	Description
CHE_WORKSPACE_DEVFILE__DEFAULT__EDITOR	eclipse/che-theia/next	Default Editor that should be provisioned into Devfile if there is no specified Editor Format is editorPublisher/editorName/editorVersion value. NULL or absence of value means that default editor should not be provisioned.

Environment Variable Name	Default value	Description
CHE_WORKSPACE_DEVFILE_DEFAULT_EDITOR_PLUGINS	eclipse/che-machine-exec-plugin/nightly	Default Plugins which should be provisioned for Default Editor. All the plugins from this list that are not explicitly mentioned in the user-defined devfile will be provisioned but only when the default editor is used or if the user-defined editor is the same as the default one (even if in different version). Format is comma-separated pluginPublisher/pluginName/pluginVersion values, and URLs. For example: eclipse/che-theia-exec-plugin/0.0.1,eclipse/che-theia-terminal-plugin/0.0.1,https://cdn.pluginregistry.com/vi-mode/meta.yaml If the plugin is a URL, the plugin's meta.yaml is retrieved from that URL.
CHE_WORKSPACE_PROVISION_SECRET_LABELS	app.kubernetes.io/part-of=che.eclipse.org,app.kubernetes.io/component=workspace-secret#	Defines comma-separated list of labels for selecting secrets from a user namespace, which will be mount into workspace containers as a files or env variables. Only secrets that match ALL given labels will be selected.

Table 5.13. Che system

Environment Variable Name	Default value	Description
CHE_SYSTEM_SUPER_PRIVILEGED_MODE	false	System Super Privileged Mode. Grants users with the manageSystem permission additional permissions for getByKey, getByNameSpace, stopWorkspaces, and getResourcesInformation. These are not given to admins by default and these permissions allow admins gain visibility to any workspace along with naming themselves with admin privileges to those workspaces.

Environment Variable Name	Default value	Description
CHE_SYSTEM_ADMIN_NAME	admin	Grant system permission for 'che.admin.name' user. If the user already exists it'll happen on component startup, if not - during the first login when user is persisted in the database.

Table 5.14. Workspace limits

Environment Variable Name	Default value	Description
CHE_LIMITS_WORKSPACE_ENV_RAM	16gb	Workspaces are the fundamental runtime for users when doing development. You can set parameters that limit how workspaces are created and the resources that are consumed. 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	180000	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.

Table 5.15. Users workspace limits

Environment Variable Name	Default value	Description
CHE_LIMITS_USER_WORKSPACES_RAM	-1	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.

Environment Variable Name	Default value	Description
CHE_LIMITS_USER_WORKSPACES_COUNT	-1	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.
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.

Table 5.16. Organizations workspace limits

Environment Variable Name	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.

Environment Variable Name	Default value	Description
CHE_LIMITS_ORGANIZATION_WORKSPACES_RUNNING_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.
CHE_MAIL_FROM_EMAIL_ADDRESS	che@noreply.com	Address that will be used as from email for email notifications

Table 5.17. Organizations notifications settings

Environment Variable Name	Default value	Description
CHE_ORGANIZATION_EMAIL_MEMBER_ADDED_SUBJECT	You've been added to a Che Organization	Organization notifications subjects and templates
CHE_ORGANIZATION_EMAIL_MEMBER_ADDED_TEMPLATE	st-html-templates/user_added_to_organization	Organization notifications subjects and templates
CHE_ORGANIZATION_EMAIL_MEMBER_REMOVED_SUBJECT	You've been removed from a Che Organization	
CHE_ORGANIZATION_EMAIL_MEMBER_REMOVED_TEMPLATE	st-html-templates/user_removed_from_organization	
CHE_ORGANIZATION_EMAIL_ORG_REMOVED_SUBJECT	CheOrganization deleted	
CHE_ORGANIZATION_EMAIL_ORG_REMOVED_TEMPLATE	st-html-templates/organization_deleted	
CHE_ORGANIZATION_EMAIL_ORG_RENAMED_SUBJECT	CheOrganization renamed	

Environment Variable Name	Default value	Description
CHE_ORGANIZATION_EMAIL_ORG_RENAMED_TEMP_LATE	st-html-templates/organization_renamed	

Table 5.18. Multi-user-specific OpenShift infrastructure configuration

Environment Variable Name	Default value	Description
CHE_INFRA_OPENSHIFT_OAUTH_IDENTITY_PROVIDER	NULL	Alias of the Openshift identity provider registered in Keycloak, that should be used to create workspace OpenShift resources in Openshift namespaces owned by the current CodeReady Workspaces user. Should be set to NULL if che.infra.openshift.project is set to a non-empty value. For more information see the following documentation: https://www.keycloak.org/docs/latest/server_admin/index.html#openshift-4

Table 5.19. Keycloak configuration

Environment Variable Name	Default value	Description
CHE_KEYCLOAK_AUTH_SERVER_URL	http://\${CHE_HOST}:5050/auth	Url to keycloak identity provider server Can be set to NULL only if che.keycloak.oidcProvider is used
CHE_KEYCLOAK_REALM	che	Keycloak realm is used to authenticate users Can be set to NULL only if che.keycloak.oidcProvider is used
CHE_KEYCLOAK_CLIENT_ID	che-public	Keycloak client id in che.keycloak.realm that is used by dashboard, ide and cli to authenticate users

Table 5.20. RedHat Che specific configuration

Environment Variable Name	Default value	Description
CHE_KEYCLOAK_OSO_ENDPOINT	NULL	URL to access OSO oauth tokens
CHE_KEYCLOAK_GITHUB_ENDPOINT	NULL	URL to access Github oauth tokens
CHE_KEYCLOAK_ALLOWED_CLOCK_SKEW_SEC	3	The number of seconds to tolerate for clock skew when verifying exp or nbf claims.
CHE_KEYCLOAK_USE_NONCE	true	Use the OIDC optional nonce feature to increase security.
CHE_KEYCLOAK_JS_ADAPTER_URL	NULL	URL to the Keycloak Javascript adapter we want to use. if set to NULL, then the default used value is <code>\${che.keycloak.auth_server_url}/js/keycloak.js</code> , or <code><che-server>/api/keycloak/OIDCKeycloak.js</code> if an alternate oidc_provider is used
CHE_KEYCLOAK_OIDC_PROVIDER	NULL	Base URL of an alternate OIDC provider that provides a discovery endpoint as detailed in the following specification https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderConfig
CHE_KEYCLOAK_USE_FIXED_REDIRECT_URLS	false	Set to true when using an alternate OIDC provider that only supports fixed redirect Urls This property is ignored when che.keycloak.oidc_provider is NULL
CHE_KEYCLOAK_USERNAME_CLAIM	NULL	Username claim to be used as user display name when parsing JWT token if not defined the fallback value is 'preferred_username'

Environment Variable Name	Default value	Description
CHE_OAUTH_SERVICE_MODE	delegated	Configuration of OAuth Authentication Service that can be used in 'embedded' or 'delegated' mode. If set to 'embedded', then the service work as a wrapper to CodeReady Workspaces's OAuthAuthenticator (as in Single User mode). If set to 'delegated', then the service will use Keycloak IdentityProvider mechanism. Runtime Exception wii be thrown, in case if this property is not set properly.

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 the following 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

- CodeReady Workspaces 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, **Red Hat CodeReady Workspaces**, 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 checluster
NAME          AGE
red-hat-codeready-workspaces 27m
```

4. Delete the CodeReady Workspaces cluster:

```
$ oc delete checluster 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.v2.2**:

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



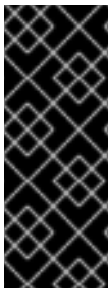
```
red-hat-codeready-workspaces.v2.2 Red Hat CodeReady Workspaces 2.2 red-hat-
codeready-workspaces.v2.1 Succeeded
```

6. Delete the CodeReady Workspaces CSV:

```
$ oc delete csv red-hat-codeready-workspaces.v2.2
clusterserviceversion.operators.coreos.com "red-hat-codeready-workspaces.v2.2" 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 project specified), use the **-n** argument also to uninstall the CodeReady Workspaces instance.
- For installations that did not use the **-n** argument, the created project 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.