



# Red Hat CodeReady Workspaces 2.3

## Installation Guide

Installing Red Hat CodeReady Workspaces 2.3



# Red Hat CodeReady Workspaces 2.3 Installation Guide

---

## Installing Red Hat CodeReady Workspaces 2.3

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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

<b>CHAPTER 1. CONFIGURING THE CODEREADY WORKSPACES INSTALLATION</b> .....	<b>3</b>
1.1. UNDERSTANDING THE CHECLUSTER CUSTOM RESOURCE	3
1.2. CHECLUSTER CUSTOM RESOURCE FIELDS REFERENCE	3
<b>CHAPTER 2. INSTALLING CODEREADY WORKSPACES ON OPENSIFT CONTAINER PLATFORM</b> .....	<b>13</b>
2.1. INSTALLING CODEREADY WORKSPACES USING THE CODEREADY WORKSPACES OPERATOR IN OPENSIFT 4 WEB CONSOLE	13
2.2. INSTALLING CODEREADY WORKSPACES USING THE CLI MANAGEMENT TOOL ON OPENSIFT CONTAINER PLATFORM 3.11	15
2.2.1. Installing the crwctl CLI management tool	15
2.2.2. Installing CodeReady Workspaces on OpenShift 3 using the Operator	15
<b>CHAPTER 3. INSTALLING CODEREADY WORKSPACES IN A RESTRICTED ENVIROMENT</b> .....	<b>18</b>
3.1. INSTALLING CODEREADY WORKSPACES IN A RESTRICTED ENVIRONMENT USING OPERATORHUB	18
3.2. INSTALLING CODEREADY WORKSPACES IN A RESTRICTED ENVIRONMENT USING CLI MANAGEMENT TOOL	19
3.2.1. Preparing an image registry for installing CodeReady Workspaces in a restricted environment	19
3.2.2. Preparing CodeReady Workspaces Custom Resource for restricted environment	22
3.2.2.1. Downloading the default CheCluster Custom Resource	22
3.2.2.2. Customizing the CheCluster Custom Resource for restricted environment	22
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	23
<b>CHAPTER 4. UPGRADING CODEREADY WORKSPACES</b> .....	<b>24</b>
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
<b>CHAPTER 5. ADVANCED CONFIGURATION OPTIONS FOR THE CODEREADY WORKSPACES SERVER COMPONENT</b> .....	<b>26</b>
5.1. UNDERSTANDING CODEREADY WORKSPACES SERVER ADVANCED CONFIGURATION USING THE OPERATOR	26
5.2. CODEREADY WORKSPACES SERVER COMPONENT SYSTEM PROPERTIES REFERENCE	27
5.3. CONFIGURING PROJECT STRATEGIES	60
5.3.1. One project per workspace strategy	60
5.3.2. One project for all workspaces strategy	60
5.3.3. One project per user strategy	61
5.3.4. Allowing user-defined workspace projects	61
<b>CHAPTER 6. UNINSTALLING CODEREADY WORKSPACES</b> .....	<b>63</b>
6.1. UNINSTALLING CODEREADY WORKSPACES AFTER OPERATORHUB INSTALLATION USING THE OPENSIFT WEB CONSOLE	63
6.2. UNINSTALLING CODEREADY WORKSPACES AFTER OPERATORHUB INSTALLATION USING OPENSIFT CLI	64
6.3. UNINSTALLING CODEREADY WORKSPACES AFTER CRWCTL INSTALLATION	65



# 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.

### CheCluster Custom Resource

- A YAML document describing the configuration of the overall CodeReady Workspaces installation.
- Contains sections to configure each component: **auth**, **database**, **server**, **storage**.

### Role of the Red Hat CodeReady Workspaces Operator

- To translate the **CheCluster** Custom Resource into configuration (ConfigMap) usable by each component of the CodeReady Workspaces installation.

### Role of OpenShift

- To apply the configuration (ConfigMap) for each component.
- To create the necessary Pods.
- When OpenShift detects a change in the configuration of a component, it restarts the Pods accordingly.

#### Example 1.1. Configuring the main properties of the CodeReady Workspaces server component

1. The user applies a **CheCluster** Custom Resource containing some configuration related to the **server**.
2. The Operator generates a necessary ConfigMap, called **codeready**.
3. OpenShift detects change in the ConfigMap and 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 1.2, “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 1.2. 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 Resource **server** settings, related to the CodeReady Workspaces server component.

Property	Default value	Description
<b>airGapContainerRegistryHostname</b>	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.



Property	Default value	Description
<b>airGapContainerRegistryOrganization</b>	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.
<b>cheDebug</b>	<b>false</b>	Enables the debug mode for CodeReady Workspaces server.
<b>cheFlavor</b>	<b>codeready-workspaces</b>	Flavor of the installation.
<b>cheHost</b>	The Operator automatically sets the value.	A public host name of the installed CodeReady Workspaces server.
<b>cheImagePullPolicy</b>	<b>Always</b> for <b>nightly</b> or <b>latest</b> images, and <b>IfNotPresent</b> in other cases	Overrides the image pull policy used in CodeReady Workspaces deployment.
<b>cheImageTag</b>	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.
<b>cheImage</b>	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.
<b>cheLogLevel</b>	<b>INFO</b>	Log level for the CodeReady Workspaces server: <b>INFO</b> or <b>DEBUG</b> .
<b>cheWorkspaceClusterRole</b>	omit	Custom cluster role bound to the user for the CodeReady Workspaces workspaces. Omit or leave empty to use the default roles.
<b>customCheProperties</b>	omit	Map of additional environment variables that will be applied in the generated <b>codeready-workspaces</b> ConfigMap to be used by the CodeReady Workspaces server, in addition to the values already generated from other fields of the <b>CheCluster</b> Custom Resource (CR). If <b>customCheProperties</b> contains a property that would be normally generated in <b>codeready-workspaces</b> ConfigMap from other CR fields, then the value defined in the <b>customCheProperties</b> will be used instead.

Property	Default value	Description
<b>devfileRegistryImage</b>	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.
<b>devfileRegistryMemoryLimit</b>	<b>256Mi</b>	Overrides the memory limit used in the Devfile registry deployment.
<b>devfileRegistryMemoryRequest</b>	<b>16Mi</b>	Overrides the memory request used in the Devfile registry deployment.
<b>devfileRegistryPullPolicy</b>	<b>Always</b> for <b>nightly</b> or <b>latest</b> images, and <b>IfNotPresent</b> in other cases	Overrides the image pull policy used in the Devfile registry deployment.
<b>devfileRegistryUrl</b>	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 <b>externalDevfileRegistry</b> field).
<b>externalDevfileRegistry</b>	<b>false</b>	Instructs the Operator to deploy a dedicated Devfile registry server. By default a dedicated devfile registry server is started. If <b>externalDevfileRegistry</b> set to <b>true</b> , the Operator does not start a dedicated registry server automatically and you need to set the <b>devfileRegistryUrl</b> field manually.
<b>externalPluginRegistry</b>	<b>false</b>	Instructs the Operator to deploy a dedicated Plugin registry server. By default, a dedicated plug-in registry server is started. If <b>externalPluginRegistry</b> set to <b>true</b> , the Operator does not deploy a dedicated server automatically and you need to set the <b>pluginRegistryUrl</b> field manually.
<b>nonProxyHosts</b>	omit	List of hosts that will not use the configured proxy. Use <code> </code> as delimiter, for example <b>localhost my.host.com 123.42.12.32</b> . Only use when configuring a proxy is required (see also the <b>proxyURL</b> field).
<b>pluginRegistryImage</b>	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.
<b>pluginRegistryMemoryLimit</b>	<b>256Mi</b>	Overrides the memory limit used in the Plugin registry deployment.

Property	Default value	Description
<b>pluginRegistryMemoryRequest</b>	<b>16Mi</b>	Overrides the memory request used in the Plugin registry deployment.
<b>pluginRegistryPullPolicy</b>	<b>Always</b> for <b>nightly</b> or <b>latest</b> images, and <b>IfNotPresent</b> in other cases	Overrides the image pull policy used in the Plugin registry deployment.
<b>pluginRegistryUrl</b>	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 <b>externalPluginRegistry</b> field).
<b>proxyPassword</b>	omit	Password of the proxy server. Only use when proxy configuration is required.
<b>proxyPort</b>	omit	Port of the proxy server. Only use when configuring a proxy is required (see also the <b>proxyURL</b> field).
<b>proxyURL</b>	omit	URL (protocol+host name) of the proxy server. This drives the appropriate changes in the <b>JAVA_OPTS</b> and <b>https(s)_proxy</b> variables in the CodeReady Workspaces server and workspaces containers. Only use when configuring a proxy is required.
<b>proxyUser</b>	omit	User name of the proxy server. Only use when configuring a proxy is required (see also the <b>proxyURL</b> field).
<b>selfSignedCert</b>	<b>false</b>	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 <b>tlsSupport</b> field on demo OpenShift clusters that have not been setup with a valid certificate for the routes.
<b>serverMemoryLimit</b>	<b>1Gi</b>	Overrides the memory limit used in the CodeReady Workspaces server deployment.
<b>serverMemoryRequest</b>	<b>512Mi</b>	Overrides the memory request used in the CodeReady Workspaces server deployment.
<b>tlsSupport</b>	<b>true</b>	Instructs the Operator to deploy CodeReady Workspaces in TLS mode. Enabling TLS requires enabling the <b>selfSignedCert</b> field.

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

Property	Default value	Description
<b>chePostgresDb</b>	<b>dbche</b>	PostgreSQL database name that the CodeReady Workspaces server uses to connect to the database.
<b>chePostgresHostName</b>	the Operator sets the value automatically	PostgreSQL Database host name that the CodeReady Workspaces server uses to connect to. Defaults to <b>postgres</b> . Override this value only when using an external database. (See the field <b>externalDb</b> .)
<b>chePostgresPassword</b>	auto-generated value	PostgreSQL password that the CodeReady Workspaces server uses to connect to the database.
<b>chePostgresPort</b>	<b>5432</b>	PostgreSQL Database port that the CodeReady Workspaces server uses to connect to. Override this value only when using an external database (see field <b>externalDb</b> ).
<b>chePostgresUser</b>	<b>pgche</b>	PostgreSQL user that the CodeReady Workspaces server uses to connect to the database.
<b>externalDb</b>	<b>false</b>	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 <b>true</b> , 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: <b>chePostgres</b> .
<b>postgresImagePullPolicy</b>	Always` for <b>nightly</b> or <b>latest</b> images, and <b>IfNotPresent</b> in other cases	Overrides the image pull policy used in the PostgreSQL database deployment.
<b>postgresImage</b>	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 Resourceauth** configuration settings related to authentication used by CodeReady Workspaces installation

Property	Default value	Description
<b>externalIdentityProvider</b>	<b>false</b>	By default, a dedicated Identity Provider server is deployed as part of the CodeReady Workspaces installation. But if <b>externalIdentityProvider</b> is <b>true</b> , 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: <b>identityProvider</b> .

Property	Default value	Description
<b>identityProviderAdminUserName</b>	<b>admin</b>	Overrides the name of the Identity Provider admin user.
<b>identityProviderClientid</b>	omit	Name of an Identity provider (Keycloak / RH SSO) <b>client-id</b> that must be used for CodeReady Workspaces. This is useful to override it ONLY if you use an external Identity Provider (see the <b>externalIdentityProvider</b> field). If omitted or left blank, it will be set to the value of the <b>flavor</b> field suffixed with <b>-public</b> .
<b>identityProviderImagePullPolicy</b>	<b>Always</b> for <b>nightly</b> or <b>latest</b> images, and <b>IfNotPresent</b> in other cases	Overrides the image pull policy used in the Identity Provider (Keycloak / RH SSO) deployment.
<b>identityProviderImage</b>	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.
<b>identityProviderPassword</b>	omit	Overrides the password of Keycloak admin user. Override it only when using an external Identity Provider (see the <b>externalIdentityProvider</b> field). Omit or leave empty to set an auto-generated password.
<b>identityProviderPostgresPassword</b>	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 <b>externalIdentityProvider</b> field).
<b>identityProviderRealm</b>	omit	Name of an Identity provider (Keycloak / RH SSO) realm. Override it only when using an external Identity Provider (see the <b>externalIdentityProvider</b> field). Omit or leave empty blank to set it to the value of the <b>flavor</b> field.
<b>identityProviderURL</b>	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 <b>externalIdentityProvider</b> field).
<b>oAuthClientName</b>	the Operator sets the value automatically	Name of the OpenShift <b>OAuthClient</b> resource used to setup identity federation on the OpenShift side. See also the <b>OpenShiftoAuth</b> field.
<b>oAuthSecret</b>	the Operator sets the value automatically	Name of the secret set in the OpenShift <b>OAuthClient</b> resource used to setup identity federation on the OpenShift side. See also the <b>OAuthClientName</b> field.

Property	Default value	Description
<b>openShifttoAuth</b>	<b>true</b> 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 <b>kubeadmin</b> user is not supported, and logging through does not allow access to the CodeReady Workspaces Dashboard.
<b>updateAdminPassword</b>	<b>false</b>	Forces the default <b>admin</b> 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
<b>postgresPVCStorageClassName</b>	omit	Storage class for the Persistent Volume Claim dedicated to the PostgreSQL database. Omitted or leave empty to use a default storage class.
<b>preCreateSubPaths</b>	<b>false</b>	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.
<b>pvcClaimSize</b>	<b>1Gi</b>	Size of the persistent volume claim for workspaces.
<b>pvcJobsImage</b>	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 <b>preCreateSubPaths</b> field.
<b>pvcStrategy</b>	<b>common</b>	Available options: <code>common</code> (all workspaces PVCs in one volume), <b>per-workspace</b> (one PVC per workspace for all declared volumes) and <b>unique</b> (one PVC per declared volume).
<b>workspacePVCStorageClassName</b>	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
<b>ingressClass</b>	<b>nginx</b>	Ingress class that defines which controller manages ingresses.

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

Table 1.6. **CheCluster Custom Resource** **status** defines the observed state of CodeReady Workspaces installation

Property	Description
<b>cheClusterRunning</b>	Status of a CodeReady Workspaces installation. Can be <b>Available, Unavailable,</b> or <b>Available, Rolling Update in Progress.</b>
<b>cheURL</b>	Public URL to the CodeReady Workspaces server.
<b>cheVersion</b>	Currently installed CodeReady Workspaces version.
<b>dbProvisioned</b>	Indicates whether a PostgreSQL instance has been correctly provisioned.
<b>devfileRegistryURL</b>	Public URL to the Devfile registry.
<b>helpLink</b>	A URL to where to find help related to the current Operator status.
<b>keycloakProvisioned</b>	Indicates whether an Identity Provider instance (Keycloak / RH SSO) has been provisioned with realm, client and user.
<b>keycloakURL</b>	Public URL to the Identity Provider server (Keycloak / RH SSO).
<b>message</b>	A human-readable message with details about why the Pod is in this state.

Property	Description
<b>openShifttoAuthProvided</b>	Indicates whether an Identity Provider instance (Keycloak / RH SSO) has been configured to integrate with the OpenShift OAuth.
<b>pluginRegistryURL</b>	Public URL to the Plugin registry.
<b>reason</b>	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 2.3 End-user Guide](#).
- [the CodeReady Workspaces 2.3 Administration Guide](#).
- It is possible to use the **crwctl** utility script for deploying CodeReady Workspaces on OpenShift Container Platform and OpenShift Dedicated versions 4.5. 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 2.3.
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).

**NOTE**

Methods for updating from a previous CodeReady Workspaces installation and enabling multiple instances in the same OpenShift Container Platform 3.11 cluster are provided below the installation procedure.

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](#).
- `<namespace>` 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 command to create the CodeReady Workspaces instance

- In the user-defined `<namespace>`:

- In the user-defined `<namespace>`:

```
$ crwctl server:start -n <namespace> -p openshift
```

- In the default project called workspaces:

```
$ crwctl server:start -p openshift
```

### 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.

### Upgrading from a previous CodeReady Workspaces installation

- 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
```

### Having multiple CodeReady Workspaces deployments

- 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"} } ]'
```

## CHAPTER 3. INSTALLING CODEREADY WORKSPACES IN A RESTRICTED ENVIRONMENT

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 ENVIRONMENT 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 ENVIRONMENT USING CLI MANAGEMENT TOOL



### NOTE

Use CodeReady Workspaces CLI management tool to install CodeReady Workspaces on restricted networks if installation through OperatorHub is not available. This method is supported for OpenShift Container Platform 3.11.

### Prerequisites

- A running OpenShift cluster. See the [OpenShift Container Platform 3.11 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.
- The **skopeo** tool, version 0.1.40 or later, is installed.
- The **podman** tool is installed.
- An image registry that is accessible from the OpenShift cluster and supporting the format of the V2 image manifest, schema version 2. Ensure you can push to it from a location that has, at least temporarily, access to the internet.
  - When pushing to an image registry within the restricted environment, the OpenShift cluster must be able to access it.

The following placeholders are used in this section.

#### Table 3.1. Placeholders used in examples

<b>&lt;image-registry&gt;</b>	host name and port of the container-image registry accessible in the restricted environment
<b>&lt;organization&gt;</b>	organization of the container-image registry

## Procedure

1. Log into the internal image registry:

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



### NOTE

If you encounter an error, such as **x509: certificate signed by unknown authority**, when attempting to push to the internal registry, either add the OpenShift cluster's certificate to `/etc/containers/certs.d/<image-registry>`, or add the registry as an insecure registry by adding the following lines to the Podman configuration file located at `/etc/containers/registries.conf`:

```
[registries.insecure]
registries = ['<image-registry>']
```

To ensure that digests are not changed, use the **skopeo copy --all** command to copy images without changing them. Repeat this step for every image in the following lists:

+

```
$ skopeo copy --all docker://<image-name>:<image-tag> docker://<image-registry>/<organization>/<image-name>:<image-tag>
```

1. Check if the images have the same digests:

```
$ skopeo inspect docker://<image-name>:<image-tag>
$ skopeo inspect docker://<image-registry>/<organization>/<image-name>:<image-tag>
```

- Set the digests explicitly when different:

```
$ skopeo copy --all docker://<image_name>:<image_digest> docker://<image-registry>/<organization>/<image-namename>:<image-digest>
```

## Essential images

Every workspace launch requires infrastructure images from the following list:

- CodeReady Workspaces deployment and workspace support
  - registry.redhat.io/codeready-workspaces/{prod-operator-image-name}:2.3
  - registry.redhat.io/codeready-workspaces/crw-2-rhel8-operator-metadata:2.3
  - registry.redhat.io/codeready-workspaces/devfileregistry-rhel8:2.3



- registry.redhat.io/codeready-workspaces/server-rhel8:2.3
- registry.redhat.io/codeready-workspaces/imagepuller-rhel8:2.3
- registry.redhat.io/codeready-workspaces/jwtproxy-rhel8:2.3
- registry.redhat.io/codeready-workspaces/pluginbroker-artifacts-rhel8:2.3
- registry.redhat.io/codeready-workspaces/pluginbroker-metadata-rhel8:2.3
- registry.redhat.io/codeready-workspaces/pluginregistry-rhel8:2.3
- registry.redhat.io/rh-sso-7/sso74-openshift-rhel8:7.4
- registry.redhat.io/ubi8-minimal:8.2
- registry.redhat.io/rhel8/postgresql-96:1
- Plugins and editors
  - registry.redhat.io/codeready-workspaces/machineexec-rhel8:2.3
  - registry.redhat.io/codeready-workspaces/theia-rhel8:2.3
  - registry.redhat.io/codeready-workspaces/theia-endpoint-rhel8:2.3

### 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-rhel8:2.3
  - registry.redhat.io/codeready-workspaces/plugin-java11-rhel8:2.3
  - registry.redhat.io/codeready-workspaces/plugin-kubernetes-rhel8:2.3
  - registry.redhat.io/codeready-workspaces/plugin-openshift-rhel8:2.3
- Stacks
  - registry.redhat.io/codeready-workspaces/stacks-cpp-rhel8:2.3
  - registry.redhat.io/codeready-workspaces/stacks-dotnet-rhel8:2.3
  - registry.redhat.io/codeready-workspaces/stacks-golang-rhel8:2.3
  - registry.redhat.io/codeready-workspaces/stacks-php-rhel8:2.3
  - registry.redhat.io/jboss-eap-7/eap73-openjdk8-openshift-rhel7:7.3.1
- Workspace tooling
  - registry.redhat.io/rhsc1/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: '<image-registry>'
    airGapContainerRegistryOrganization: '<organization>'
# [...]
```

### 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. See [Section 2.2.1, "Installing the crwctl CLI management tool"](#).
- 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 a customized Custom Resource to add fields related to the restricted environment:

```
$ crwctl server:start \
--che-operator-image=<image-registry>/<organization>/crw-2-rhel8-operator:2.3 \
--che-operator-cr-yaml=org_v1_che_cr.yaml
```



#### NOTE

For slow systems or internet connections, add the `--k8spodwaittimeout=1800000` flag option to the `crwctl server:start` command to extend the Pod timeout period to 1800000 ms or longer.

### 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>'
# [...]
```

2. 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 2.3.

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 2.3 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, in the **<workspaces>** project.
- An installation of the **crwctl** 2.3 version management tool. See [Using the crwctl management tool](#).

## Procedure

1. In all running workspaces in the CodeReady Workspaces 2.2 instance, save and push changes back to the Git repositories.
2. Shut down all workspaces in the CodeReady Workspaces 2.2 instance.
3. Run the following command:

```
$ crwctl -n <workspaces> server:update
```



### NOTE

For slow systems or internet connections, add the **--k8spodwaittimeout=1800000** flag option to the **crwctl server:update** command to extend the Pod timeout period to 1800000 ms or longer.

## Verification steps

1. Navigate to the CodeReady Workspaces instance.
2. The 2.3 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
<b>CHE_DATABASE</b>	<b><code>\${che.home}/storage</code></b>	Folder where CodeReady Workspaces will store internal data objects
<b>CHE_API</b>	<b><code>http://\${CHE_HOST}:\${CHE_PORT}/api</code></b>	API service. Browsers initiate REST communications to CodeReady Workspaces server with this URL
<b>CHE_WEBSOCKET_ENDPOINT</b>	<b><code>ws://\${CHE_HOST}:\${CHE_PORT}/api/websocket</code></b>	CodeReady Workspaces websocket major endpoint. Provides basic communication endpoint for major websocket interaction/messaging.
<b>CHE_WEBSOCKET_ENDPOINT_MINOR</b>	<b><code>ws://\${CHE_HOST}:\${CHE_PORT}/api/websocket-minor</code></b>	CodeReady Workspaces websocket minor endpoint. Provides basic communication endpoint for minor websocket interaction/messaging.
<b>CHE_WORKSPACE_STORAGE</b>	<b><code>\${che.home}/workspaces</code></b>	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.
<b>CHE_WORKSPACE_PROJECTS_STORAGE</b>	<b><code>/projects</code></b>	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
<b>CHE_WORKSPACE_PROJECTS_STORAGE_DEFAULT_SIZE</b>	<b>1Gi</b>	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)
<b>CHE_WORKSPACE_LOGS_ROOT_DIR</b>	<b>/workspace_logs</b>	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.
<b>CHE_WORKSPACE_HTTP_PROXY</b>		Configures proxies used by runtimes powering workspaces
<b>CHE_WORKSPACE_HTTPS_PROXY</b>		Configures proxies used by runtimes powering workspaces
<b>CHE_WORKSPACE_NO_PROXY</b>		Configures proxies used by runtimes powering workspaces
<b>CHE_TRUSTED_CA_BUNDLES_CONFIGMAP</b>	<b>NULL</b>	When cluster wide proxy is configured, che-operator creates special configmap and allows OpenShift Network operator to inject ca-bundle into it. In addition, it adds the key <code>CHE_TRUSTEDCABUNDLES_CONFIGMAP</code> with name of this configmap into CodeReady Workspaces server configmap (and corresponding ENV variable). So by its presence we can detect if proxy mode is enabled or not. This property is not supposed to be set manually unless that specifically required.
<b>CHE_WORKSPACE_AUTO_START</b>	<b>true</b>	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.



Environment Variable Name	Default value	Description
<b>CHE_WORKSPACE_POOL_TYPE</b>	<b>fixed</b>	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'
<b>CHE_WORKSPACE_POOL_EXACT_SIZE</b>	<b>30</b>	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
<b>CHE_WORKSPACE_POOL_CORES_MULTIPLIER</b>	<b>2</b>	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
<b>CHE_WORKSPACE_PROBE_POOL_SIZE</b>	<b>10</b>	This property specifies how much threads to use for workspaces servers liveness probes
<b>CHE_WORKSPACE_HTTP_PROXY_JAVA_OPTIONS</b>	<b>NULL</b>	Http proxy setting for workspace JVM
<b>CHE_WORKSPACE_JAVA_OPTIONS</b>	<b>-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</b>	Java command line options to be added to JVM's that running within workspaces.

Environment Variable Name	Default value	Description
<b>CHE_WORKSPACE_MAVEN_OPTIONS</b>	<b>-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</b>	Maven command line options added to JVM's that run agents within workspaces.
<b>CHE_WORKSPACE_MAVEN_SERVER_JAVA_OPTIONS</b>	<b>-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</b>	Default java command line options to be added to JVM that run maven server.
<b>CHE_WORKSPACE_DEFAULT_MEMORY_LIMIT_MB</b>	<b>1024</b>	RAM limit default for each machine that has no RAM settings in environment. Value less or equal to 0 interpreted as limit disabling.
<b>CHE_WORKSPACE_DEFAULT_MEMORY_REQUEST_MB</b>	<b>200</b>	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.

Environment Variable Name	Default value	Description
<b>CHE_WORKSPACE_DEFAULT_CPU_LIMIT_CORES</b>	-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.
<b>CHE_WORKSPACE_DEFAULT_CPU_REQUEST_CORES</b>	-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.
<b>CHE_WORKSPACE_SIDECAR_DEFAULT_MEMORY_LIMIT_MB</b>	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.
<b>CHE_WORKSPACE_SIDECAR_DEFAULT_MEMORY_REQUEST_MB</b>	64	RAM limit and request default for each sidecar that has no RAM settings in <b>{prod-short}</b> plugin configuration. Value less or equal to 0 interpreted as limit disabling.
<b>CHE_WORKSPACE_SIDECAR_DEFAULT_CPU_LIMIT_CORES</b>	-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.

Environment Variable Name	Default value	Description
<b>CHE_WORKSPACE_SIDECAR_DEFAULT_CPU_REQUEST_CORES</b>	<b>-1</b>	CPULimit and request default for each sidecar that has no CPU settings in <b>{prod-short}</b> 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.
<b>CHE_WORKSPACE_SIDECAR_IMAGE_PULL_POLICY</b>	<b>Always</b>	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.)
<b>CHE_WORKSPACE_ACTIVITY_CHECK_SCHEDULER_PERIOD_S</b>	<b>60</b>	Period of inactive workspaces suspend job execution.
<b>CHE_WORKSPACE_ACTIVITY_CLEANUP_SCHEDULER_PERIOD_S</b>	<b>3600</b>	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.
<b>CHE_WORKSPACE_ACTIVITY_CLEANUP_SCHEDULER_INITIAL_DELAY_S</b>	<b>60</b>	The delay after server startup to start the first activity clean up job.
<b>CHE_WORKSPACE_ACTIVITY_CHECK_SCHEDULER_DELAY_S</b>	<b>180</b>	Delay before first workspace idleness check job started to avoid mass suspend if ws master was unavailable for period close to inactivity timeout.
<b>CHE_WORKSPACE_CLEANUP_TEMPORARY_INITIAL_DELAY_MIN</b>	<b>5</b>	Period of stopped temporary workspaces cleanup job execution.
<b>CHE_WORKSPACE_CLEANUP_TEMPORARY_PERIOD_MIN</b>	<b>180</b>	Period of stopped temporary workspaces cleanup job execution.

Environment Variable Name	Default value	Description
<b>CHE_WORKSPACE_SERVER_PING_SUCCESS_THRES_HOLD</b>	<b>1</b>	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.
<b>CHE_WORKSPACE_SERVER_PING_INTERVAL_MILLISECONDS</b>	<b>3000</b>	Interval, in milliseconds, between successive pings to workspace server.
<b>CHE_WORKSPACE_SERVER_LIVENESS_PROBES</b>	<b>wsagent/http,exec-agent/http,terminal,theia,jupyter,dirigible,cloud-shell</b>	List of servers names which require liveness probes
<b>CHE_WORKSPACE_STARTUP_DEBUG_LOG_LIMIT_BYTES</b>	<b>10485760</b>	Limit size of the logs collected from single container that can be observed by che-server when debugging workspace startup. default 10MB=10485760
<b>CHE_WORKSPACE_STOP_ROLE_ENABLED</b>	<b>true</b>	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
<b>CHE_TEMPLATE_STORAGE</b>	<b>`\${che.home}/templates</b>	Folder that contains JSON files with code templates and samples

Table 5.3. Authentication parameters

Environment Variable Name	Default value	Description
<b>CHE_AUTH_USER_SELF_CREATION</b>	<b>false</b>	CodeReady Workspaces has a single identity implementation, so this does not change the user experience. If true, enables user creation at API level
<b>CHE_AUTH_ACCESS_DENIED_ERROR_PAGE</b>	<b>/error-oauth</b>	Authentication error page address

Environment Variable Name	Default value	Description
<b>CHE_AUTH_RESERVED_US ER_NAMES</b>		Reserved user names
<b>CHE_OAUTH_GITHUB_CLIE NTID</b>	<b>NULL</b>	You can setup GitHub OAuth to automate authentication to remote repositories. You need to first register this application with GitHub OAuth.
<b>CHE_OAUTH_GITHUB_CLIE NTSECRET</b>	<b>NULL</b>	You can setup GitHub OAuth to automate authentication to remote repositories. You need to first register this application with GitHub OAuth.
<b>CHE_OAUTH_GITHUB_AUTH URI</b>	<b>https://github.com/login/oauth/authorize</b>	You can setup GitHub OAuth to automate authentication to remote repositories. You need to first register this application with GitHub OAuth.
<b>CHE_OAUTH_GITHUB_TOKE NURI</b>	<b>https://github.com/login/oauth/access_token</b>	You can setup GitHub OAuth to automate authentication to remote repositories. You need to first register this application with GitHub OAuth.
<b>CHE_OAUTH_GITHUB_RED IRECTURIS</b>	<b>http://localhost:\${CHE_PORT}/api/oauth/callback</b>	You can setup GitHub OAuth to automate authentication to remote repositories. You need to first register this application with GitHub OAuth.
<b>CHE_OAUTH_OPENSHIFT_C LIENTID</b>	<b>NULL</b>	Configuration of OpenShift OAuth client. Used to obtain OpenShift OAuth token.
<b>CHE_OAUTH_OPENSHIFT_C LIENTSECRET</b>	<b>NULL</b>	Configuration of OpenShift OAuth client. Used to obtain OpenShift OAuth token.
<b>CHE_OAUTH_OPENSHIFT_O AUTH__ENDPOINT</b>	<b>NULL</b>	Configuration of OpenShift OAuth client. Used to obtain OpenShift OAuth token.
<b>CHE_OAUTH_OPENSHIFT_V ERIFY__TOKEN__URL</b>	<b>NULL</b>	Configuration of OpenShift OAuth client. Used to obtain OpenShift OAuth token.

Table 5.4. Internal

Environment Variable Name	Default value	Description
<b>SCHEDULE_CORE_POOL_SIZE</b>	<b>10</b>	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.
<b>ORG_EVERREST_ASYNCRONOUS</b>	<b>false</b>	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.
<b>ORG_EVERREST_ASYNCRONOUS_POOL_SIZE</b>	<b>20</b>	Quantity of asynchronous requests which may be processed at the same time
<b>ORG_EVERREST_ASYNCRONOUS_QUEUE_SIZE</b>	<b>500</b>	Size of queue. If asynchronous request can't be processed after consuming it will be added in queue.
<b>ORG_EVERREST_ASYNCRONOUS_JOB_TIMEOUT</b>	<b>10</b>	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.
<b>ORG_EVERREST_ASYNCRONOUS_CACHE_SIZE</b>	<b>1024</b>	Size of cache for waiting, running and ended request.
<b>ORG_EVERREST_ASYNCRONOUS_SERVICE_PATH</b>	<b>/async/</b>	Path to asynchronous service
<b>DB_SCHEMA_FLYWAY_BASELINE_ENABLED</b>	<b>true</b>	DB initialization and migration configuration
<b>DB_SCHEMA_FLYWAY_BASELINE_VERSION</b>	<b>5.0.0.8.1</b>	DB initialization and migration configuration
<b>DB_SCHEMA_FLYWAY_SCRIPTS_PREFIX</b>		DB initialization and migration configuration

Environment Variable Name	Default value	Description
<b>DB_SCHEMA_FLYWAY_SCRIPTS_SUFFIX</b>	<b>.sql</b>	DBinitializationandmigration configuration
<b>DB_SCHEMA_FLYWAY_SCRIPTS_VERSION_SEPARATOR</b>	<b>---</b>	DBinitializationandmigrationconfiguration
<b>DB_SCHEMA_FLYWAY_SCRIPTS_LOCATIONS</b>	<b>classpath:che-schema</b>	DBinitializationandmigrationconfiguration

Table 5.5. Kubernetes Infra parameters

Environment Variable Name	Default value	Description
<b>CHE_INFRA_KUBERNETES_MASTER_URL</b>		Configuration of Kubernetes client that Infra will use
<b>CHE_INFRA_KUBERNETES_TRUST_CERTS</b>		Configuration of Kubernetes client that Infra will use
<b>CHE_INFRA_KUBERNETES_SERVER_STRATEGY</b>	<b>default-host</b>	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
<b>CHE_INFRA_KUBERNETES_SINGLE_HOST_WORKSPACE_EXPOSURE</b>	<b>native</b>	Defines the way in which the workspace plugins and editors are exposed in the single-host mode. Supported exposures: - 'native': Exposes servers using OpenShift Ingresses. Works only on Kubernetes.
<b>CHE_INFRA_KUBERNETES_INGRESS_DOMAIN</b>		Used to generate domain for a server in a workspace in case property <b>che.infra.kubernetes.server_strategy</b> is set to <b>multi-host</b>



Environment Variable Name	Default value	Description
<b>CHE_INFRA_KUBERNETES_NAMESPACE</b>		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 <b>che.infra.openshift.project</b> 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.
<b>CHE_INFRA_KUBERNETES_NAMESPACE_DEFAULT</b>	<b>&lt;username&gt;-che</b>	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
<b>CHE_INFRA_KUBERNETES_NAMESPACE_ALLOW_USER_DEFINED</b>	<b>false</b>	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
<b>CHE_INFRA_KUBERNETES_SERVICE_ACCOUNT_NAME</b>	<b>NULL</b>	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 <b>che.infra.openshift.project</b> 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
<b>CHE_INFRA_KUBERNETES_WORKSPACE_SA_CLUSTER_ROLES</b>	<b>NULL</b>	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'.
<b>CHE_INFRA_KUBERNETES_WORKSPACE_START_TIMEOUT_MIN</b>	<b>8</b>	Defines time frame that limits the Kubernetes workspace start time
<b>CHE_INFRA_KUBERNETES_INGRESS_START_TIMEOUT_MIN</b>	<b>5</b>	Defines the timeout in minutes that limits the period for which Kubernetes Ingress become ready

Environment Variable Name	Default value	Description
<b>CHE_INFRA_KUBERNETES_WORKSPACE__UNRECOVERABLE__EVENTS</b>	<b>FailedMount,FailedScheduling,MountVolume.SetUpfailed,Failed to pull image,FailedCreate</b>	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.
<b>CHE_INFRA_KUBERNETES_PVC_ENABLED</b>	<b>true</b>	Defines whether use the Persistent Volume Claim for the workspace needs e.g backup projects, logs etc or disable it.
<b>CHE_INFRA_KUBERNETES_PVC_STRATEGY</b>	<b>common</b>	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
<b>CHE_INFRA_KUBERNETES_PVC_PRECREATE__SUBPATHS</b>	<b>true</b>	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: <a href="https://github.com/kubernetes/kubernetes/issues/41638">https://github.com/kubernetes/kubernetes/issues/41638</a> Note that this property has effect only if the 'common' PVC strategy used.
<b>CHE_INFRA_KUBERNETES_PVC_NAME</b>	<b>claim-che-workspace</b>	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
<b>CHE_INFRA_KUBERNETES_PVC_STORAGE__CLASS__NAME</b>		Defines the storage class of Persistent Volume Claim for the workspaces. Empty strings means 'use default'.
<b>CHE_INFRA_KUBERNETES_PVC_QUANTITY</b>	<b>10Gi</b>	Defines the size of Persistent Volume Claim of che workspace. Format described here: <a href="https://docs.openshift.com/container-platform/4.4/storage/understanding-persistent-storage.html">https://docs.openshift.com/container-platform/4.4/storage/understanding-persistent-storage.html</a>
<b>CHE_INFRA_KUBERNETES_PVC_JOBS_IMAGE</b>	<b>centos:centos7</b>	Pod that is launched when performing persistent volume claim maintenance jobs on OpenShift

Environment Variable Name	Default value	Description
<b>CHE_INFRA_KUBERNETES_PVC_JOBS_IMAGE_PULL_POLICY</b>	<b>IfNotPresent</b>	Image pull policy of container that used for the maintenance jobs on Kubernetes/OpenShift cluster
<b>CHE_INFRA_KUBERNETES_PVC_JOBS_MEMORYLIMIT</b>	<b>250Mi</b>	Defines pod memory limit for persistent volume claim maintenance jobs
<b>CHE_INFRA_KUBERNETES_PVC_ACCESS_MODE</b>	<b>ReadWriteOnce</b>	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: <a href="https://docs.openshift.com/container-platform/4.4/storage/understanding-persistent-storage.html">https://docs.openshift.com/container-platform/4.4/storage/understanding-persistent-storage.html</a>
<b>CHE_INFRA_KUBERNETES_PVC_WAIT_BOUND</b>	<b>true</b>	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 <b>false</b> in case if <b>volumeBindingMode</b> is configured to <b>WaitForFirstConsumer</b> otherwise workspace starts will hangs up on phase of waiting PVCs. Default value is true (means that PVCs should be waited to be bound)
<b>CHE_INFRA_KUBERNETES_INSTALLER_SERVER_MIN_PORT</b>	<b>10000</b>	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
<b>CHE_INFRA_KUBERNETES_INSTALLER_SERVER_MAX_PORT</b>	<b>20000</b>	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.
<b>CHE_INFRA_KUBERNETES_INGRESS_ANNOTATIONS_JSON</b>	<b>NULL</b>	<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><b>{'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'}</b></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</p> <p>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>
<b>CHE_INFRA_KUBERNETES_INGRESS_PATH_TRANSFORM</b>	<b>NULL</b>	<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>
<b>CHE_INFRA_KUBERNETES_POD_SECURITY_CONTEXT_RUN_AS_USER</b>	<b>NULL</b>	<p>Defines security context for pods that will be created by Kubernetes Infra This is ignored by OpenShift infra</p>
<b>CHE_INFRA_KUBERNETES_POD_SECURITY_CONTEXT_FS_GROUP</b>	<b>NULL</b>	<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
<b>CHE_INFRA_KUBERNETES_POD_TERMINATION_GRACE_PERIOD_SEC</b>	<b>0</b>	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 <b>terminationGracePeriodSeconds</b> have been explicitly set in Kubernetes / OpenShift recipe it will not be overridden.
<b>CHE_INFRA_KUBERNETES_CLIENT_HTTP_ASYNC_REQUESTS_MAX</b>	<b>1000</b>	Number of maximum concurrent async web requests (http requests or ongoing web socket calls) supported in the underlying shared http client of the <b>KubernetesClient</b> 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)
<b>CHE_INFRA_KUBERNETES_CLIENT_HTTP_ASYNC_REQUESTS_MAX_PER_HOST</b>	<b>1000</b>	Number of maximum concurrent async web requests (http requests or ongoing web socket calls) supported in the underlying shared http client of the <b>KubernetesClient</b> instances. Default values are 64, and 5 per-host, which doesn't seem correct for multi-user scenarios knowing that <b>{prod-short}</b> keeps a number of connections opened (e.g. for command or ws-agent logs)
<b>CHE_INFRA_KUBERNETES_CLIENT_HTTP_CONNECTION_POOL_MAX_IDLE</b>	<b>5</b>	Max number of idle connections in the connection pool of the Kubernetes-client shared http client



Environment Variable Name	Default value	Description
<b>CHE_INFRA_KUBERNETES_CLIENT_HTTP_CONNECTION_POOL_KEEP_ALIVE_MIN</b>	<b>5</b>	Keep-alive timeout of the connection pool of the Kubernetes-client shared http client in minutes
<b>CHE_INFRA_KUBERNETES_TLS_ENABLED</b>	<b>false</b>	Creates Ingresses with Transport Layer Security (TLS) enabled In OpenShift infrastructure, Routes will be TLS-enabled
<b>CHE_INFRA_KUBERNETES_TLS_SECRET</b>		Name of a secret that should be used when creating workspace ingresses with TLS Ignored by OpenShift infrastructure
<b>CHE_INFRA_KUBERNETES_TLS_KEY</b>	<b>NULL</b>	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
<b>CHE_INFRA_KUBERNETES_TLS_CERT</b>	<b>NULL</b>	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
<b>CHE_INFRA_KUBERNETES_RUNTIMES_CONSISTENCY_CHECK_PERIOD_MIN</b>	<b>-1</b>	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 <b>-1</b> , where <b>-1</b> 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
<b>CHE_INFRA_OPENSIFT_PROJECT</b>		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.
<b>CHE_INFRA_OPENSIFT_TRUSTED_CA_BUNDLES_CONFIG_MAP</b>	<b>ca-certs</b>	Configures name of the trust-store config map where the CA bundles are stored in Openshift 4. This map is supposed to be initially created by CodeReady Workspaces installer (operator or etc) with basically any name, and CodeReady Workspaces server finds it by specific label (see below) during workspace startup and then creates and mounts same map in the namespace of the workspace. The property defines name of the map in workspace namespace.
<b>CHE_INFRA_OPENSIFT_TRUSTED_CA_BUNDLES_CONFIG_MAP_LABELS</b>	<b>config.openshift.io/inject-trusted-cabundle=true</b>	Label name for config maps which are used for automatic certificate injection in Openshift 4.
<b>CHE_INFRA_OPENSIFT_TRUSTED_CA_BUNDLES_MOUNT_PATH</b>	<b>/public-certs</b>	Configures path on workspace containers where the CA bundles are mount.

Environment Variable Name	Default value	Description
<b>CHE_SINGLEPORT_WILDCARD__DOMAIN_HOST</b>	<b>NULL</b>	Single port mode wildcard domain host & port. nip.io is used by default
<b>CHE_SINGLEPORT_WILDCARD__DOMAIN_PORT</b>	<b>NULL</b>	Singleport mode wildcard domain host & port. nip.io is used by default
<b>CHE_SINGLEPORT_WILDCARD__DOMAIN_IPLESS</b>	<b>false</b>	Enable single port custom DNS without inserting the IP

Table 5.7. Experimental properties

Environment Variable Name	Default value	Description
<b>CHE_WORKSPACE_PLUGIN__BROKER_METADATA_IMAGE</b>	<b>quay.io/eclipse/che-plugin-metadata-broker:v3.3.0</b>	Docker image of CodeReady Workspaces plugin broker app that resolves workspace tooling configuration and copies plugins dependencies to a workspace
<b>CHE_WORKSPACE_PLUGIN__BROKER_ARTIFACTS_IMAGE</b>	<b>quay.io/eclipse/che-plugin-artifacts-broker:v3.3.0</b>	Docker image of <b>{prod-short}</b> plugin broker app that resolves workspace tooling configuration and copies plugins dependencies to a workspace
<b>CHE_WORKSPACE_PLUGIN__BROKER_PULL_POLICY</b>	<b>Always</b>	Docker image of CodeReady Workspaces plugin broker app that resolves workspace tooling configuration and copies plugins dependencies to a workspace
<b>CHE_WORKSPACE_PLUGIN__BROKER_WAIT_TIMEOUT_MIN</b>	<b>3</b>	Defines the timeout in minutes that limits the max period of result waiting for plugin broker.
<b>CHE_WORKSPACE_PLUGIN__REGISTRY_URL</b>	<b>https://che-plugin-registry.prod-preview.openshift.io/v3</b>	Workspace tooling plugins registry endpoint. Should be a valid HTTP URL. <b>Example: http://che-plugin-registry-eclipse-che.192.168.65.2.nip.io</b> In case CodeReady Workspaces plugins tooling is not needed value 'NULL' should be used

Environment Variable Name	Default value	Description
<b>CHE_WORKSPACE_DEVFILE_REGISTRY_URL</b>	<b>https://che-devfile-registry.prod-preview.openshift.io/</b>	Devfile Registry endpoint. Should be a valid HTTP URL. <b>Example: http://che-devfile-registry-eclipse-che.192.168.65.2.nip.io</b> In case CodeReady Workspaces plugins tooling is not needed value 'NULL' should be used
<b>CHE_WORKSPACE_STORAGE_AVAILABLE_TYPES</b>	<b>persistent,ephemeral,async</b>	The configuration property that defines available values for storage types that clients like Dashboard should propose for users during workspace creation/update. Available values: <ul style="list-style-type: none"> <li>- 'persistent': Persistent Storage slow I/O but persistent.</li> <li>- 'ephemeral': Ephemeral Storage allows for faster I/O but may have limited storage and is not persistent.</li> <li>- 'async': Experimental feature: Asynchronous storage is combination of Ephemeral and Persistent storage. Allows for faster I/O and keep your changes, will backup on stop and restore on start workspace. Will work only if: <ul style="list-style-type: none"> <li>- che.infra.kubernetes.pvc.strategy='common'</li> <li>- che.limits.user.workspaces.run.count=1</li> <li>- che.infra.kubernetes.namespace.allow_user_defined=false</li> <li>- che.infra.kubernetes.namespace.default contains &lt;username&gt; in other cases remove 'async' from the list.</li> </ul> </li> </ul>
<b>CHE_WORKSPACE_STORAGE_PREFERRED_TYPE</b>	<b>persistent</b>	The configuration property that defines a default value for storage type that clients like Dashboard should propose for users during workspace creation/update. The 'async' value not recommended as default type since it's experimental

Environment Variable Name	Default value	Description
<b>CHE_SERVER_SECURE_EX POSER</b>	<b>default</b>	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.
<b>CHE_SERVER_SECURE_EX POSER_JWTPROXY_TOKEN _ISSUER</b>	<b>wsmaster</b>	Jwtproxy issuer string, token lifetime and optional auth page path to route unsigned requests to.
<b>CHE_SERVER_SECURE_EX POSER_JWTPROXY_TOKEN _TTL</b>	<b>8800h</b>	Jwtproxyissuer string, token lifetime and optional auth page path to route unsigned requests to.
<b>CHE_SERVER_SECURE_EX POSER_JWTPROXY_AUTH_ LOADER_PATH</b>	<b>/_app/loader.html</b>	Jwtproxyissuerstring, token lifetime and optional auth page path to route unsigned requests to.
<b>CHE_SERVER_SECURE_EX POSER_JWTPROXY_IMAGE</b>	<b>quay.io/eclipse/che- jwtproxy:0.10.0</b>	Jwtproxyissuerstring,token lifetime and optional auth page path to route unsigned requests to.
<b>CHE_SERVER_SECURE_EX POSER_JWTPROXY_MEMOR Y_LIMIT</b>	<b>128mb</b>	Jwtproxyissuerstring,tokenlifetim e and optional auth page path to route unsigned requests to.
<b>CHE_SERVER_SECURE_EX POSER_JWTPROXY_CPU_ LIMIT</b>	<b>0.5</b>	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
<b>CHE_CORE_JSONRPC_PRO CESSOR__MAX__POOL__SI ZE</b>	<b>50</b>	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
<b>CHE_CORE_JSONRPC_PROCESSOR_CORE_POOL_SIZE</b>	<b>5</b>	Initial json processing pool. Minimum number of threads that used to process major JSON RPC messages.
<b>CHE_CORE_JSONRPC_PROCESSOR_QUEUE_CAPACITY</b>	<b>100000</b>	Configuration of queue used to process Json RPC messages.

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

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

Table 5.10. CORS settings

Environment Variable Name	Default value	Description
<b>CHE_CORS_ALLOWED_ORIGINS</b>	<b>*</b>	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

Environment Variable Name	Default value	Description
<b>CHE_CORS_ALLOW_CREDENTIALS</b>	<b>false</b>	'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
<b>CHE_FACTORY_DEFAULT_EDITOR</b>	<b>eclipse/che-theia/next</b>	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 of .factory.json) Multiple plugins must be comma-separated, for example: pluginFooPublisher/pluginFooName/pluginFooVersion,pluginBarPublisher/pluginBarName/pluginBarVersion
<b>CHE_FACTORY_DEFAULT_PLUGINS</b>	<b>eclipse/che-machine-exec-plugin/nightly</b>	Editor and plugin which will be used for factories which are created from remote git repository which doesn't contain any <b>{prod-short}</b> -specific workspace descriptors (like .devfile of .factory.json) Multiple plugins must be comma-separated, for example: pluginFooPublisher/pluginFooName/pluginFooVersion,pluginBarPublisher/pluginBarName/pluginBarVersion
<b>CHE_FACTORY_DEFAULT_DEVFILE_FILENAMES</b>	<b>devfile.yaml,.devfile.yaml</b>	Devfile filenames to look on repository-based factories (like GitHub etc). Factory will try to locate those files in the order they enumerated in the property.

Table 5.12. Devfile defaults



Environment Variable Name	Default value	Description
<b>CHE_WORKSPACE_DEVFILE_DEFAULT_EDITOR</b>	<b>eclipse/che-theia/next</b>	Default Editor that should be provisioned into Devfile if there is no specified Editor Format is <b>editorPublisher/editorName/editorVersion</b> value. <b>NULL</b> or absence of value means that default editor should not be provisioned.
<b>CHE_WORKSPACE_DEVFILE_DEFAULT_EDITOR_PLUGINS</b>	<b>eclipse/che-machine-exec-plugin/nightly</b>	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 <b>pluginPublisher/pluginName/pluginVersion</b> 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.
<b>CHE_WORKSPACE_PROVISION_SECRET_LABELS</b>	<b>app.kubernetes.io/part-of=che.eclipse.org,app.kubernetes.io/component=workspace-secret</b>	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.
<b>CHE_WORKSPACE_DEVFILE_ASYNC_STORAGE_PLUGIN</b>	<b>eclipse/che-async-pv-plugin/nightly</b>	Plugin is added in case async storage feature will be enabled in workspace config and supported by environment
<b>CHE_INFRA_KUBERNETES_ASYNC_STORAGE_IMAGE</b>	<b>quay.io/eclipse/che-workspace-data-sync-storage:latest</b>	Docker image for the CodeReady Workspaces async storage

Environment Variable Name	Default value	Description
<b>CHE_WORKSPACE_POD_NODE_SELECTOR</b>	<b>NULL</b>	Optionally configures node selector for workspace pod. Format is comma-separated key=value pairs, e.g: disktype=ssd,cpu=xlarge,foo=bar
<b>CHE_INFRA_KUBERNETES_ASYNC_STORAGE_SHUTDOWN_TIMEOUT_MIN</b>	<b>120</b>	The timeout for the Asynchronous Storage Pod shutdown after stopping the last used workspace. Value less or equal to 0 interpreted as disabling shutdown ability.
<b>CHE_INFRA_KUBERNETES_ASYNC_STORAGE_SHUTDOWN_CHECK_PERIOD_MIN</b>	<b>30#</b>	Defines the period with which the Asynchronous Storage Pod stopping ability will be performed (once in 30 minutes by default)

Table 5.13. Che system

Environment Variable Name	Default value	Description
<b>CHE_SYSTEM_SUPER_PRIVILEGED_MODE</b>	<b>false</b>	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.
<b>CHE_SYSTEM_ADMIN_NAME</b>	<b>admin</b>	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
<b>CHE_LIMITS_WORKSPACE_ENV_RAM</b>	<b>16gb</b>	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.
<b>CHE_LIMITS_WORKSPACE_IDLE_TIMEOUT</b>	<b>1800000</b>	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.
<b>CHE_LIMITS_WORKSPACE_RUN_TIMEOUT</b>	<b>0</b>	The length of time in milliseconds that a workspace will run, regardless of activity, before the system will suspend it. Set this property if you want to automatically stop workspaces after a period of time. The default is zero, meaning that there is no run timeout.

Table 5.15. Users workspace limits

Environment Variable Name	Default value	Description
<b>CHE_LIMITS_USER_WORKSPACES_RAM</b>	<b>-1</b>	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
<b>CHE_LIMITS_USER_WORKSPACES_COUNT</b>	<b>-1</b>	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.
<b>CHE_LIMITS_USER_WORKSPACES_RUN_COUNT</b>	<b>1</b>	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
<b>CHE_LIMITS_ORGANIZATION_WORKSPACES_RAM</b>	<b>-1</b>	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.
<b>CHE_LIMITS_ORGANIZATION_WORKSPACES_COUNT</b>	<b>-1</b>	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
<b>CHE_LIMITS_ORGANIZATION_WORKSPACES_RUN_COUNT</b>	<b>-1</b>	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.
<b>CHE_MAIL_FROM_EMAIL_ADDRESS</b>	<b>che@noreply.com</b>	Address that will be used as from email for email notifications

Table 5.17. Organizations notifications settings

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

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

Table 5.18. Multi-user-specific OpenShift infrastructure configuration

Environment Variable Name	Default value	Description
<b>CHE_INFRA_OPENSHIFT_OAUTH_IDENTITY_PROVIDER</b>	<b>NULL</b>	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 <b>che.infra.openshift.project</b> is set to a non-empty value. For more information see the following documentation: <a href="https://www.keycloak.org/docs/latest/server_admin/index.html#openshift-4">https://www.keycloak.org/docs/latest/server_admin/index.html#openshift-4</a>

Table 5.19. Keycloak configuration

Environment Variable Name	Default value	Description
<b>CHE_KEYCLOAK_AUTH_SERVER_URL</b>	<b>http://\${CHE_HOST}:5050/auth</b>	Url to keycloak identity provider server Can be set to NULL only if <b>che.keycloak.oidcProvider</b> is used
<b>CHE_KEYCLOAK_REALM</b>	<b>che</b>	Keycloak realm is used to authenticate users Can be set to NULL only if <b>che.keycloak.oidcProvider</b> is used
<b>CHE_KEYCLOAK_CLIENT_ID</b>	<b>che-public</b>	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
<b>CHE_KEYCLOAK_OSO_END POINT</b>	<b>NULL</b>	URL to access OSO oauth tokens
<b>CHE_KEYCLOAK_GITHUB_E NDPOINT</b>	<b>NULL</b>	URL to access Github oauth tokens
<b>CHE_KEYCLOAK_ALLOWED __CLOCK__SKEW__SEC</b>	<b>3</b>	The number of seconds to tolerate for clock skew when verifying exp or nbf claims.
<b>CHE_KEYCLOAK_USE__NO NCE</b>	<b>true</b>	Use the OIDC optional <b>nonce</b> feature to increase security.
<b>CHE_KEYCLOAK_JS__ADAP TER__URL</b>	<b>NULL</b>	URL to the Keycloak Javascript adapter we want to use. if set to NULL, then the default used value is <b><code>#{che.keycloak.auth_server_url}/js/keycloak.js</code></b> , or <b><code>&lt;che-server&gt;/api/keycloak/OIDCKe ycloak.js</code></b> if an alternate <b>oidc_provider</b> is used
<b>CHE_KEYCLOAK_OIDC__PR OVIDER</b>	<b>NULL</b>	Base URL of an alternate OIDC provider that provides a discovery endpoint as detailed in the following specification <a href="https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderConfig">https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderConfig</a>
<b>CHE_KEYCLOAK_USE__FIX ED__REDIRECT__URLS</b>	<b>false</b>	Set to true when using an alternate OIDC provider that only supports fixed redirect Urls This property is ignored when <b>che.keycloak.oidc_provider</b> is NULL
<b>CHE_KEYCLOAK_USERNAM E__CLAIM</b>	<b>NULL</b>	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
<b>CHE_OAUTH_SERVICE_MODE</b>	<b>delegated</b>	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.

## 5.3. CONFIGURING PROJECT STRATEGIES

The project strategies are configured using the **CHE\_INFRA\_KUBERNETES\_NAMESPACE\_DEFAULT** environment variable.



### WARNING

**CHE\_INFRA\_KUBERNETES\_NAMESPACE** and **CHE\_INFRA\_OPENSIFT\_PROJECT** are legacy variables. Keep these variables unset for a new installations. Changing these variables during an update can lead to data loss.

### 5.3.1. One project per workspace strategy

The strategy creates a new project for each new workspace.

To use the strategy, the **CHE\_INFRA\_KUBERNETES\_NAMESPACE\_DEFAULT** variable value must contain the **<workspaceID>** identifier. It can be used alone or combined with other identifiers or any string.

#### Example 5.2. One project per workspace

To assign project names composed of a ``codeready-ws`` prefix and workspace id, set:

```
CHE_INFRA_KUBERNETES_NAMESPACE_DEFAULT=codeready-ws-<workspaceID>
```

### 5.3.2. One project for all workspaces strategy

The strategy uses one predefined project for all workspaces.



To use the strategy, the **CHE\_INFRA\_KUBERNETES\_NAMESPACE\_DEFAULT** variable value must be the name of the desired project to use.

### Example 5.3. One project for all workspaces

To have all workspaces created in ``codeready-ws`` project, set:

```
CHE_INFRA_KUBERNETES_NAMESPACE_DEFAULT=codeready-ws
```



### IMPORTANT

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

### 5.3.3. One project per user strategy

The strategy isolates each user in their own project.

To use the strategy, the **CHE\_INFRA\_KUBERNETES\_NAMESPACE\_DEFAULT** variable value must contain one or more user identifiers. Currently supported identifiers are **<username>** and **<userId>**.

### Example 5.4. One project per user

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

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



### IMPORTANT

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

To limit the number of concurrently running workspaces per user to one, set the **CHE\_LIMITS\_USER\_WORKSPACES\_RUN\_COUNT** environment variable to **1**.

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

- For Operator deployments: set the **spec.server.cheCustomProperties.CHE\_LIMITS\_USER\_WORKSPACE\_RUN\_COUNT** variable of the CheCluster Custom Resource (CR) to **1**.

### 5.3.4. Allowing user-defined workspace projects

CodeReady Workspaces server can be configured to honor the user selection of a project when a workspace is created. This feature is disabled by default. To allow user-defined workspace projects:

- For Operator deployments, set the following field in the CheCluster Custom Resource:

allowUserDefinedWorkspaceNamespaces

## 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, for the OpenShift Web Console method see [Section 6.1, “Uninstalling CodeReady Workspaces after OperatorHub installation using the OpenShift web console”](#).
- For CodeReady Workspaces installed using OperatorHub, for the CLI method see [Section 6.2, “Uninstalling CodeReady Workspaces after OperatorHub installation using OpenShift CLI”](#).
- For CodeReady Workspaces installed using `crwctl`, see [Section 6.3, “Uninstalling CodeReady Workspaces after `crwctl` installation”](#)

### 6.1. UNINSTALLING CODEREADY WORKSPACES AFTER OPERATORHUB INSTALLATION USING THE OPENSIFT 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

1. Navigate to the OpenShift web console and select the Administrator Perspective.
2. In the **Home > Projects** section, navigate to the project containing the CodeReady Workspaces instance.

#### TIP

The default project name is `<workspaces>`.

3. In the **Operators > Installed Operators** section, click **Red Hat CodeReady Workspaces** in the list of installed operators.
4. In the **Red Hat CodeReady Workspaces Cluster** tab, click the displayed Red Hat CodeReady Workspaces Cluster, and select the **Delete cluster** option in the **Actions** drop-down menu on the top right.

#### TIP

The default Red Hat CodeReady Workspaces Cluster name is `<red-hat-codeready-workspaces>`.

5. In the **Operators > Installed Operators** section, click **Red Hat CodeReady Workspaces** in the list of installed operators and select the **Uninstall Operator** option in the **Actions** drop-down menu on the top right.

6. In the **Home > Projects** section, navigate to the project containing the CodeReady Workspaces instance, and select the **Delete Project** option in the **Actions** drop-down menu on the top right.

## 6.2. UNINSTALLING CODEREADY WORKSPACES AFTER OPERATORHUB INSTALLATION USING OPENSIFT CLI

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.
- The **oc** tool is available.

### 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.v2.3**:

```
$ oc get csv
NAME                                DISPLAY          VERSION  REPLACES          PHASE
red-hat-codeready-workspaces.v2.3  Red Hat CodeReady Workspaces 2.3  red-hat-codeready-workspaces.v2.2  Succeeded
```

6. Delete the CodeReady Workspaces CSV:

```
$ oc delete csv red-hat-codeready-workspaces.v2.3
clusterserviceversion.operators.coreos.com "red-hat-codeready-workspaces.v2.3" deleted
```

## 6.3. 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.

### Prerequisites

- The **crwctl** tool is available.
- The **oc** tool is available.
- The **crwctl** tool installed the CodeReady Workspaces instance on OpenShift.

### Procedure

1. Sign in to the OpenShift cluster:

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

2. Obtain the name of the CodeReady Workspaces namespace:

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

3. Remove the CodeReady Workspaces instance from the *<namespace>* project:

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

### TIP

When the name of the project containing the CodeReady Workspaces instance is **workspaces**, the **-n** argument is not necessary.

4. Remove the *<namespace>* project:

```
$ oc delete projects <namespace>
```