



# Red Hat CodeReady Workspaces 1.2

## Release Notes and Known Issues

Release Notes and Known Issues for Red Hat CodeReady Workspaces 1.2



# Red Hat CodeReady Workspaces 1.2 Release Notes and Known Issues

---

Release Notes and Known Issues for Red Hat CodeReady Workspaces 1.2

Robert Kratky

[rkratky@redhat.com](mailto:rkratky@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 about new and noteworthy features as well as known issues in Red Hat CodeReady Workspaces 1.2.

---

## Table of Contents

<b>CHAPTER 1. RELEASE NOTES</b> .....	<b>3</b>
1.1. ABOUT RED HAT CODEREADY WORKSPACES	3
1.1.1. Notable enhancements	3
1.1.1.1. Compatibility with OpenShift 4.x	3
1.1.1.2. New UBI 8-based developer stacks	3
1.1.1.3. Documentation improvements	4
1.1.1.4. Supported environments	4
1.1.2. Installing and deploying CodeReady Workspaces	4
1.1.3. Support policy	4
1.2. DIFFERENCE BETWEEN ECLIPSE CHE AND RED HAT CODEREADY WORKSPACES	4
<b>CHAPTER 2. KNOWN ISSUES</b> .....	<b>6</b>
2.1. GENERAL KNOWN ISSUES	6
2.1.1. Multiple workspaces fail to run at the same time	6
2.1.2. Undefined error when starting a workspace	6
2.1.3. "Authorization token is missed" error at the login page	6
2.1.4. Occasional NullPointerException when opening a workspace from the "Java 1.8" stack with an imported Java project	6
2.1.5. Delay when launching commands using the command palette	6
2.1.6. New terminal windows freeze after a minute of inactivity on macOS X	7
2.1.7. Workspaces fail to start with certain configurations of StorageClass	7
2.2. KNOWN ISSUES ON OPENSIFT CONTAINER PLATFORM 4.1	7
2.2.1. Logging in using OpenShift OAuth does not work	7
2.2.2. Error marker fails to appear in Java-stack workspaces	7
2.2.3. CodeReady Workspaces scalability workaround on OCP 4.1/AWS	7
<b>CHAPTER 3. FAQ</b> .....	<b>8</b>



# CHAPTER 1. RELEASE NOTES

Red Hat CodeReady Workspaces is a next-generation web-based integrated development environment (IDE). CodeReady Workspaces runs in OpenShift and is well-suited for container-based development.

This section documents the most important features and bug fixes in the Red Hat CodeReady Workspaces product.

## 1.1. ABOUT RED HAT CODEREADY WORKSPACES

Red Hat CodeReady Workspaces 1.2 provides a cloud developer workspace server and browser-based IDE built for teams and organizations. CodeReady Workspaces includes ready-to-use developer stacks for most of the popular programming languages, frameworks, and Red Hat technologies.

This version of CodeReady Workspaces introduces:

- Compatibility with OpenShift 4.1
- New UBI 8 based developer stacks
- Containers built against newer base images to include CVE fixes
- Improved deploy.sh script

CodeReady Workspaces 1.2 is available in the [Red Hat Container Catalog](#). Install it on OpenShift Container Platform or OpenShift Dedicated, starting at version 3.11, by following the instructions in the [Installing CodeReady Workspaces](#) chapter of the Administration Guide.

For OpenShift 4.1, CodeReady Workspaces 1.2 is available from the OperatorHub. Based on a new operator that leverages the Operator Lifecycle Manager, the installation flow is getting simpler and can be handled without leaving the OpenShift Console. If you already have OpenShift 4.1, get CodeReady Workspaces from the OperatorHub and follow the [Installing CodeReady Workspaces on Openshift 4 with OperatorHub](#) chapter of the Administration Guide.

### 1.1.1. Notable enhancements

#### 1.1.1.1. Compatibility with OpenShift 4.x

CodeReady Workspaces 1.2 is compatible with OpenShift 4.1 and gets a dedicated operator leveraging the Operator Lifecycle Manager.

#### 1.1.1.2. New UBI 8-based developer stacks

CodeReady Workspaces now provides default developer stacks for different technologies based on [Red Hat Universal Base Image](#) (UBI). All existing stacks are now based on UBI 8 so you can take advantage of the latest and greatest official Red Hat container images.

Figure 1.1. CodeReady Workspaces stacks updated to UBI 8

Stacks

A stack is environment configuration for workspace defined by its runtime recipe. Create workspaces from stacks that define projects, runtimes and commands. [Learn more.](#)

[Add Stack](#)

NAME	DESCRIPTION	COMPONENTS	ACTIONS
.NET	.NET 2.0 stack with .NET Core SDK and Runtime	RHEL, .NET Runtime, .NET SDK	
Cpp	C and C++ Developer Tools stack	RHEL, make, cmake, gcc, g++, llvm-toolset	
Go	Golang stack	RHEL, Golang	
Java 1.8	RHEL 8 java stack with EAP 7.2, OpenJDK 1.8 and Maven 3.5	RHEL, OpenJDK, Maven, EAP	
jBoss EAP	RHEL 8 java stack with EAP 7.2, OpenJDK 1.8 and Maven 3.5	RHEL, OpenJDK, Maven, EAP	
Node	NodeJS 8x stack on RHEL 7	RHEL, NodeJS, NPM	
Node10	NodeJS 10x stack	RHEL, NodeJS, NPM	
PHP	PHP 7.2 stack, including most popular extensions	RHEL, PHP, Zend Debugger, Composer	
Python	Python 3.6 stack	RHEL, Python, pip	
Red Hat Fuse	Red Hat Fuse stack	RHEL, OpenJDK, Maven, EAP	
Spring Boot	Spring Boot stack	RHEL, OpenJDK, Maven, EAP	
Thorntail	Thorntail stack	RHEL, OpenJDK, Maven, EAP	
Vert.x	Vert.x stack	RHEL, OpenJDK, Maven, EAP	

In addition to a new Node.js 10 stack based on UBI 8, the Node 8 stack has been updated to the latest version of Node.js 8 on UBI 7.

### 1.1.1.3. Documentation improvements

This release also includes documentation for:

- Installing CodeReady Workspaces with specific CustomStorageClass
- Creating custom images for custom developer stacks

### 1.1.1.4. Supported environments

Red Hat CodeReady Workspaces for OpenShift Container Platform can be installed on OpenShift Container Platform or OpenShift Dedicated starting at version 3.11.

## 1.1.2. Installing and deploying CodeReady Workspaces

For OpenShift 3.11, see the [Installing CodeReady Workspaces](#) chapter of the Administration Guide.

For OpenShift 4.1, see the [Installing CodeReady Workspaces from Operator Hub](#) chapter of the Administration Guide.

## 1.1.3. Support policy

For Red Hat CodeReady Workspaces 1.2, Red Hat will provide support for deployment, configuration, and use of the product.

CodeReady Workspaces 1.2 has been tested on Chrome version 71.0.3578.98 (Official Build) (64-bit).

For more information, see [CodeReady Workspaces life-cycle and support policy](#).

## 1.2. DIFFERENCE BETWEEN ECLIPSE CHE AND RED HAT CODEREADY WORKSPACES



The following table lists the differences between Eclipse Che and Red Hat CodeReady Workspaces:

CodeReady Workspaces	Eclipse Che
Includes the Bayesian plug-in for analytics	Does not include the Bayesian plug-in
The CodeReady Workspaces stacks are based on Red Hat Enterprise Linux. The CodeReady Workspaces stacks list includes several stack images based on Red Hat Enterprise Application Platform, such as Vert.x, Springboot, etc.	The Eclipse Che stacks are based on CentOS and other free operating systems
The Java debugger runs on Red Hat Enterprise Application Platform	The Java debugger runs on Tomcat
The PHP debugger uses httpd (Apache HyperText Transfer Protocol [HTTP] server)	The PHP debugger uses the Apache web server

## CHAPTER 2. KNOWN ISSUES

This section lists known issues with Red Hat CodeReady Workspaces 1.2. Where available, workaround suggestions are provided.

### 2.1. GENERAL KNOWN ISSUES

#### 2.1.1. Multiple workspaces fail to run at the same time

The following two conditions prevent the user from running multiple workspaces simultaneously:

- CodeReady Workspaces uses the **common** Persistent Volume Claim (PVC) strategy
- Persistent volumes (PVs) use **ReadWriteOnce** (RWO) access mode

To work around this limitation to run multiple workspaces simultaneously, use one of the following measures:

- set **ReadWriteMany** (RWX) access mode for PVs
- use the **unique** PVC strategy
- use the **per-workspace** strategy

#### 2.1.2. Undefined error when starting a workspace

When starting workspace a broken websocket connection can cause an **Undefined error**. The error does not impact the workspace start process, and refreshing the browser page fixes it.

#### 2.1.3. "Authorization token is missed" error at the login page

After installing CodeReady Workspaces on OpenShift Container Platform with TLS support and a self-signed certificate, it is not possible to enter the login page, and the **Authorization token is missed**. **Click here to reload page** error message is displayed.

To work around this limitation, add the self-signed TLS certificates to the browser for each route exposed in CodeReady Workspaces. This includes the **codeready-workspaces** deployment, the **keycloak** (Red Hat Single Sign-On) deployment, the **plugin-registry** and **devfile-registry** deployments, and any other ones you might create thereafter. Repeat this for all browsers you use to access CodeReady Workspaces.

For instructions on adding self-signed certificates, follow the documentation for your browser.

#### 2.1.4. Occasional NullPointerException when opening a workspace from the "Java 1.8" stack with an imported Java project

When opening a workspace (Java 1.8 stack) with an imported Java project, an error (**NullPointerException**) is encountered. The project does not open. To work around this issue, open the project again.

#### 2.1.5. Delay when launching commands using the command palette

When launching a command using the command palette, a delay of approximately five seconds occurs before the real execution of the command.

To work around this issue, activate the JDT.LS (Java Language Server Protocol) installer in workspace configuration in **Dashboard > Workspace details** page.

### 2.1.6. New terminal windows freeze after a minute of inactivity on macOS X

When a new terminal is opened and the user stops typing, the terminal becomes unresponsive after some time of inactivity. The user must close the terminal and open a new one to continue.

### 2.1.7. Workspaces fail to start with certain configurations of StorageClass

When the *StorageClass* resource object in OpenShift is configured with **volumeBindingMode=WaitForFirstConsumer**, the workspaces in CodeReady Workspaces fail to start.

To work around the issue, take the following steps:

1. In the OpenShift web console, expand **Workloads** and click **Config Maps > custom**.
2. Click the **YAML** tab.
3. In the **data** section, add the following environment variable and set it to **false**:

```
CHE_INFRA_KUBERNETES_PVC_WAIT__BOUND: 'false'
```

The **CHE\_INFRA\_KUBERNETES\_PVC\_WAIT\_\_BOUND** environment variable is used to disable or enable the wait time for the PVCs to bind.

4. Save the **YAML** file.
5. Run the following commands to manually scale down and then scale up the deployment to ensure that the wait time for PVCs to bind is disabled:

```
$ oc scale deployment/codeready --replicas=0  
$ oc scale deployment/codeready --replicas=1
```

## 2.2. KNOWN ISSUES ON OPENSIFT CONTAINER PLATFORM 4.1

### 2.2.1. Logging in using OpenShift OAuth does not work

On OpenShift Container Platform 4.1, authentication with Red Hat SSO that is configured to use OpenShift OAuth does not work. There is no work around for this issue other than not using OpenShift OAuth.

### 2.2.2. Error marker fails to appear in Java-stack workspaces

On OpenShift Container Platform 4.1, when editing Java source code in a CodeReady Workspaces workspace based on the Java stack, error markers do not appear when expected. At times, an error marker appears when there should be none. Reloading the page solves the issue only temporarily.

### 2.2.3. CodeReady Workspaces scalability workaround on OCP 4.1/AWS

For details, see [CodeReady Workspaces scalability workaround on OCP 4.1/AWS](#) .

## CHAPTER 3. FAQ

1. Can I install CodeReady Workspaces **offline** (that is, disconnected from the internet)?  
Yes, you can. For detailed instructions, see [Installing CodeReady Workspaces in restricted environments](#) chapter of the Administration Guide.
2. Can I use non-default certificates with CodeReady Workspaces?  
Yes, you can use self-signed or public certificates. See [Installing CodeReady Workspaces on OpenShift v3](#) chapter of the Administration Guide.
3. Can I run multiple workspaces simultaneously?  
The following two conditions prevent the user from running multiple workspaces simultaneously:
  - CodeReady Workspaces uses the **common** Persistent Volume Claim (PVC) strategy
  - Persistent volumes (PVs) use **ReadWriteOnce** (RWO) access mode  
To work around this limitation to run multiple workspaces simultaneously, use one of the following measures:
    - set **ReadWriteMany** (RWX) access mode for PVs
    - use the **unique** PVC strategy
    - use the **per-workspace** strategy