



Red Hat CodeReady Workspaces 2.1

Release Notes and Known Issues

Release Notes and Known Issues for Red Hat CodeReady Workspaces 2.1

Red Hat CodeReady Workspaces 2.1 Release Notes and Known Issues

Release Notes and Known Issues for Red Hat CodeReady Workspaces 2.1

Robert Kratky
rkratky@redhat.com

Michal Maléř
mmaler@redhat.com

Fabrice Flore-Thébault
ffloreth@redhat.com

Yana Hontyk
yhontyk@redhat.com

Legal Notice

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Information about new and noteworthy features as well as known issues in Red Hat CodeReady Workspaces 2.1.

Table of Contents

CHAPTER 1. RELEASE NOTES	3
1.1. ABOUT RED HAT CODEREADY WORKSPACES	3
1.2. NOTABLE ENHANCEMENTS	4
1.2.1. Support of self-signed certificates and SSH key pairs, Git flow improvements	4
1.2.2. Devfile	5
1.2.3. Other enhancements	5
1.2.3.1. Termination of a running task from the IDE	5
1.2.3.2. Access to the workspace container logs is now persistent	5
1.2.3.3. The URL in the preview panel is now editable	5
1.2.3.4. Monitoring and Tracing for multiple Threads pools of CodeReady Workspaces server	6
1.2.3.5. Opening terminal in a specific container	6
1.2.3.6. Offline devfile creation	6
1.2.3.7. Operators use a digest for containerImage reference instead of a tag	6
1.2.3.8. Plug-in broker refactoring	6
1.2.3.9. Ability to add a plug-in via a URL that is not in a plug-in registry	6
1.2.3.10. CodeReady Workspaces command line tool (crwctl) improvements	6
1.2.3.11. Added codeready-workspaces command syntax auto-completion to the Task editor	6
1.2.3.12. Workspaces are created in a namespace based on the username value	6
1.2.3.13. Internal editor improvements	7
1.3. SUPPORTED PLATFORMS	7
1.3.1. Supported platforms and installation methods	7
1.3.2. Installing and deploying CodeReady Workspaces	7
1.3.3. Support policy	8
1.4. DIFFERENCE BETWEEN ECLIPSE CHE AND RED HAT CODEREADY WORKSPACES	8
CHAPTER 2. KNOWN ISSUES	9
2.1. WORKSPACES BASED ON JAVA-EAP-MAVED CREATED IN CODEREADY WORKSPACES 2.0 FAIL TO START AFTER THE UPDATE TO 2.1.1 ON OPENSIFT 3.11	9
2.2. OPENSIFT CONNECTOR PLUG-IN REQUIRES MANUAL CONNECTING TO THE TARGET CLUSTER	9
2.3. MISSING DEBUG CONFIGURATION FOR THE NODE JS CONFIG MAP DEVFILE WORKSPACES	9
2.4. CODEREADY WORKSPACES IS FAILING TO SHUT DOWN AFTER EXECUTING CRWCTL SERVER:STOP	9
2.5. THE CRWCTL WORKSPACE:INJECT COMMAND DOES NOT WORK IN WORKSPACES WITH OPENSIFT OAUTH SUPPORT	10
2.6. OPENSIFT PROJECTS REMAIN PRESENT AFTER A NAMESPACE DELETION CODEREADY WORKSPACES 2.1.1	10
2.7. THE CRWCTL SERVER:DELETE UNINSTALLATION COMMAND DOES NOT REMOVE THE OPENSIFT PROJECT	10
2.8. UNINSTALLATION COMMAND CRWCTL SERVER:DELETE MIGHT MAKE OPENSIFT INSTANCE UNUSABLE	11
2.9. AN EMBEDDED APPLICATION OF THE "JAVA EAP MAVEN" STACK TENDS TO FAIL AT LAUNCH IN THE DEBUG MODE	11
2.10. ENTERING A WORKSPACE FAILS AFTER RESTARTING IT	11
2.11. WORKSPACE CAP AND WORKSPACE RAM CAP ORGANIZATION RESTRICTIONS DO NOT WORK	11
2.12. THE TERMINAL TAB FOR A WORKSPACE IN SOME CASES DOES NOT OPEN	11
2.13. WORKSPACES ARE NOT STOPPED BY THE IDLING TIMEOUT	12
2.14. ERROR HIGHLIGHTING AND CODE COMPLETION DO NOT WORK IN A GO DEVFILE	12
2.15. DEBUGGING UTILITY DOES NOT WORK CORRECTLY ON ITS FIRST RUN IN THE GO DEVFILE WORKSPACE	12
CHAPTER 3. FAQ	13

CHAPTER 1. RELEASE NOTES

Red Hat CodeReady Workspaces is a 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 Red Hat CodeReady Workspaces. For the list of CodeReady Workspaces 2.1 release issues, see the [Chapter 2, *Known issues*](#) section.

CodeReady Workspaces 2.1.1 with several bugfixes has been released. To upgrade, follow the instructions in the [Upgrading CodeReady Workspaces](#) chapter of the Installation Guide.

TLS (https) now enabled by default; adjust old CodeReady Workspaces instance manually

- In CodeReady Workspaces 2.0, the **tlsSupport** parameter was set to **false** by default, allowing the use of insecure HTTP. In CodeReady Workspaces 2.1, secure HTTPS is required to use the most recent Theia IDE, and is therefore enabled by default. Existing workspaces from 2.0 need to be adjusted to use **tlsSupport: true**:

```
$ oc patch -n <codereadyNamespace> checluster/codeready-workspaces \
--patch "{\"spec\":{\"server\":{\"tlsSupport\": true}}}" --type=merge
```



NOTE

This issue is fixed by updating to CodeReady Workspaces version 2.1.1.

1.1. ABOUT RED HAT CODEREADY WORKSPACES

Red Hat CodeReady Workspaces 2.1 provides an enterprise-level cloud developer workspace server and browser-based integrated development environment (IDE). CodeReady Workspaces includes ready-to-use developer stacks for most of the popular programming languages, frameworks, and Red Hat technologies.

This minor release of Red Hat CodeReady Workspaces is based on Eclipse Che 7.9 and offers a number of enhancements and new features, including:

- **Support for OpenShift Dedicated 4.3**
Using the CodeReady Workspaces Operator and crwctl, CodeReady Workspaces can be installed on OpenShift Dedicated versions 3.11 and 4.3. Users can then benefit from all Red Hat-managed container application platform features, including:
 - A flexible application environment.
 - The ability to connect and extend local services.
 - An isolated platform, which improves security and reduces downtime.
See [Supported platforms for deploying Red Hat CodeReady Workspaces](#) for a table of supported platforms and installation methods.
- **New onboarding flow from the dashboard**
The **Getting Started with CodeReady Workspaces** page lets users start workspaces by clicking a single button and without having to configure anything. Users see the **Getting Started** page by default when there is no workspace created.
- **Temporary storage options for workspace creation**

The dashboard propose options to enable or disable the temporary storage and displays the kubernetes namespace used when creating a workspace.

- **Quarkus implementation**

Quarkus support for workspaces is now available. By default, the create, read, update, and delete (CRUD) services are provided. The workspace contains ready-to-use commands to start developer mode, including debugging capabilities.

The workspace also contains the Red Hat VS Code Quarkus plug-in to generate a new project or provide snippets using the Command palette.

- **CodeReady Workspaces Air Gap with OpenShift 4.3**

On OpenShift 4.3, it is now possible to follow [Configuring OperatorHub for restricted networks using OpenShift Container Platform](#) and configure the CodeReady Workspaces Operator to be used this way.

- **Languages updates**

Provided versions:

- .NET 3.1 - Update from version 2.1
- Java 11 - Extending the current use of Java 8

- **Support added**

- Apache Camel K, a lightweight integration framework for serverless and microservice architectures.
- Gradle, a build automation tool, often used for JVM languages such as Java, Groovy, or Scala.

- **New chapters in the documentation**

- Using artifact repositories in a restricted environment
- Backup and Disaster Recovery
- Configuring system properties for CodeReady Workspaces
- OpenShift Connector, the VS Code extension

CodeReady Workspaces 2.1 is available in the [Red Hat Container Catalog](#). Install it on OpenShift Container Platform, starting at version 3.11, by following the instructions in the [Installing CodeReady Workspaces](#) chapter of the Installation Guide.

From OpenShift 4.1, CodeReady Workspaces 2.1 is available from the OperatorHub. Based on a new Operator that uses the Operator Lifecycle Manager, the installation flow is simpler and can be handled without leaving the OpenShift Console.

For OpenShift 4.3, get CodeReady Workspaces from the OperatorHub and follow the [Installing CodeReady Workspaces on OpenShift 4 from OperatorHub](#) chapter of the Installation Guide.

1.2. NOTABLE ENHANCEMENTS

1.2.1. Support of self-signed certificates and SSH key pairs, Git flow improvements

- **git clone** command is now supported for repositories with self-signed SSL certificates, allowing customers behind a firewall (air gap) to connect to company internal Git repositories and clone from them.
- SSH key upload in CodeReady Workspaces using the command palette is now allowed. Users are now able to configure Git credentials while doing a commit without those parameters already set.
- The user's SSH keys are mounted automatically on workspace start into a single Kubernetes secret.
- Users can list their SSH keys using the **SSH: view public key** command and check if the public part is uploaded to the Git server by performing a private remote Git operation, such as **git clone** or **git push**.

1.2.2. Devfile

- Monaco is now the main devfile editor, replacing Codemirror. This provides YAML highlighting and validation, which makes it easier to customize devfiles manually. The Che devfile editor now supports auto-completion, validation, and hover.
- Added the ability to create a workspace from the dashboard with a devfile.
- A preview URL is now offered when a task is started, and support for such URLs is added to devfiles.
- Environment variables for plug-ins, editors, and other OpenShift or Kubernetes components are supported. This allows plug-in developers to set default values that are usable in most cases and can be overwritten later in a devfile if needed.
- Adds the ability for CodeReady Workspaces factories to override devfile properties.
- Adds the ability for the devfile to be set to the current project opened in CodeReady Workspaces by default, if not specified in the **project** section of the devfile. Also, launching a workspace using the factory stored within a Git repository or a Git branch leads to project creation. The project name then matches the name of the repository or the branch.
- Adds the ability for the CodeReady Workspaces factory loader to read a devfile from a GitHub repository URL that ends in **.git**.

1.2.3. Other enhancements

1.2.3.1. Termination of a running task from the IDE

Adds the ability to stop a running task by sending **Ctrl+c** to the corresponding terminal or by using the **Command palette → Terminate Task** action.

1.2.3.2. Access to the workspace container logs is now persistent

All of the workspace lifetime logs are now persistent and accessible to the end-user using the **crwctl** and dashboard.

1.2.3.3. The URL in the preview panel is now editable

The preview panel text-field that displays the currently opened URL is now editable and allows the user to navigate to the different components of an application, such as endpoints and pages.

1.2.3.4. Monitoring and Tracing for multiple Threads pools of CodeReady Workspaces server

Users can now monitor and trace multiple thread pools of CodeReady Workspaces server, the proper adjustment of which can lead to better handling of workspace start load.

1.2.3.5. Opening terminal in a specific container

The command pallet offers actions related to specific containers.

1.2.3.6. Offline devfile creation

Air gap mode allows offline devfile registry to include sample projects and images.

1.2.3.7. Operators use a digest for containerImage reference instead of a tag

In CodeReady Workspaces, registries and Operator metadata now use specific SHA256 image digests instead of mutable image tags like **:2.0**. This provides a number of benefits, including better security and support for the new approach to restricted environments in Openshift 4.3. A complete list of images that the Operator could use, including all the stack and sidecar images, is now included in the **relatedImages** section of the Operator metadata **clusterserviceversion** (CSV) YAML file.

1.2.3.8. Plug-in broker refactoring

The plug-in installation process was split into separate phases to reduce the time needed to process plug-ins and to enable plug-ins to be cached between workspace launches.

1.2.3.9. Ability to add a plug-in via a URL that is not in a plug-in registry

Improvements in the plug-in broker and plug-in resolution code allow adding a set of default plug-ins, without including them in the plug-in registry. Users can configure the CodeReady Workspaces server with a list of URLs to plug-in **meta.yaml** files.

1.2.3.10. CodeReady Workspaces command line tool (crwctl) improvements

- User can use the watch utility to monitor a newly created Pod.
- The **crwctl inject** command is now supported in OpenShift command line interface (CLI).

1.2.3.11. Added codeready-workspaces command syntax auto-completion to the Task editor

Added support for CRW specific autocompletion and hinting when editing tasks in a CRW workspace. The editor now includes CRW-specific fields. For example, which container to use, and assisting the user while adding a task to Che-Theia.

1.2.3.12. Workspaces are created in a namespace based on the username value

In CRW 2.0, workspaces were created in a dedicated namespace named with the workspace ID by default.

Starting with CRW 2.1, by default, all workspaces of a given user are created in a single namespace named `<username>-crw`.

1.2.3.13. Internal editor improvements

- File tree empty-space reduction
- VS Code API compatibility extension
- More plug-ins supported

1.3. SUPPORTED PLATFORMS

1.3.1. Supported platforms and installation methods

The following section provides information about the availability of CodeReady Workspaces 2.1 on OpenShift Container Platform, OpenShift Dedicated, and about their supported installation methods.

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

Table 1.1. Availability of CodeReady Workspaces 2.1 on OpenShift Container Platform and OpenShift Dedicated

	3.11	4.3	4.4	4.5
OpenShift Container Platform	✓	✓	✓	Technical Preview
OpenShift Dedicated	✓	✓	Technical Preview	Technical Preview

Table 1.2. Supported installation method for CodeReady Workspaces 2.1 on OpenShift Container Platform and OpenShift Dedicated

	3.11	4.3	4.4
OpenShift Container Platform	crwctl	OperatorHub	OperatorHub
OpenShift Dedicated	crwctl	OperatorHub	N/A

It is possible to use the `crwctl` utility script for deploying CodeReady Workspaces 2.1 on OpenShift Container Platform versions 4.3, 4.4, and OpenShift Dedicated version 4.3. This method is considered unofficial and serves as a backup installation method for situations where the installation method using OperatorHub is not available.

1.3.2. Installing and deploying CodeReady Workspaces

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

For OpenShift 4.4, see the [Installing CodeReady Workspaces from Operator Hub](#) chapter of the Installation Guide.

1.3.3. Support policy

For Red Hat CodeReady Workspaces 2.1, Red Hat will provide support for deployment, configuration, and use of the product.

CodeReady Workspaces 2.1 has been tested on Chrome version 83.0.4103.97 (Official Build) (64-bit).

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

1.4. DIFFERENCE BETWEEN ECLIPSE CHE AND RED HAT CODEREADY WORKSPACES

The main difference between CodeReady Workspaces and Eclipse Che is that CodeReady Workspaces is supported by Red Hat. There is no difference in the technologies these two products use. Nevertheless, CodeReady Workspaces runs the plug-ins and devfiles from supported images. Licensing, packaging, and support are also provided by Red Hat.

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

CodeReady Workspaces	Eclipse Che
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

CHAPTER 2. KNOWN ISSUES

This section lists known issues with Red Hat CodeReady Workspaces 2.1. Where available, workaround suggestions are provided.

2.1. WORKSPACES BASED ON JAVA-EAP-MAVED CREATED IN CODEREADY WORKSPACES 2.0 FAIL TO START AFTER THE UPDATE TO 2.1.1 ON OPENSIFT 3.11

Users with the cluster-admin role who use OpenShift 3.11 may experience the 2.0 version **java-eap-maved**-based workspaces with TLS and OpenShift OAuth support fail to start after the product server is updated to version 2.1.1. Users are then not permitted to access the old Persistent Volume.

To work around this issue, create a new 2.1.1 workspace using the same devfile that was used to create the workspace in the previous version.

2.2. OPENSIFT CONNECTOR PLUG-IN REQUIRES MANUAL CONNECTING TO THE TARGET CLUSTER

By default, the Openshift Connector plug-in logs into the cluster as **inClusterUser**, which may not have the manage project permission. This causes an error message to be displayed when a new project is being created using Openshift Application Explorer:

```
Failed to create Project with error 'Error: Command failed: "/tmp/vscode-unpacked/redhat.vscode-openshift -connector.latest.qvkozqtkba.openshift-connector-0.1.4-523.vsix/extension/out/tools/linux/odo" project create test-project X projectrequests.project.openshift.io is forbidden
```

To work around this issue, log out from the local cluster and relog in to OpenShift cluster using the OpenShift user's credentials.

2.3. MISSING DEBUG CONFIGURATION FOR THE NODE JS CONFIG MAP DEVFILE WORKSPACES

Absence of the debug configuration, situated in the debug panel of a created workspace, when the **Node JS Config Map** devfile is used for a new workspace creation.

2.4. CODEREADY WORKSPACES IS FAILING TO SHUT DOWN AFTER EXECUTING **CRWCTL SERVER:STOP**

The dedicated crwctl command **crwctl server:stop** is unable to shut down the CodeReady Workspaces server and instead fails with a timeout and displays the following error message:

```
> Error: E_SHUTDOWN_CHE_SERVER_FAIL - Failed to shutdown CodeReady Workspaces server. E_CHE_API_NO_RESPONSE - Endpoint: http://codeready-ndp-test.apps.crw.codereadyqe.com/api/system/stop?shutdown=true - Error message: timeout of > 3000ms exceeded
```

To work around the issue, execute **crwctl server:stop** again.

2.5. THE `CRWCTL WORKSPACE:INJECT` COMMAND DOES NOT WORK IN WORKSPACES WITH OPENSIFT OAUTH SUPPORT

The `crwctl workspace:inject` command causes the error during the workspace Pod localization in workspaces created with OpenShift OAuth support.

```
$ crwctl workspace:inject -n codeready-tls-oauth -k
✓ Verify if namespace codeready-tls-oauth exists
✗ Verify if the workspaces is running
  → No workspace pod is found
  Injecting configurations
  › Error: No workspace pod is found
```

To work around the issue, use `oc login` command inside the affected container instead.

2.6. OPENSIFT PROJECTS REMAIN PRESENT AFTER A NAMESPACE DELETION CODEREADY WORKSPACES 2.1.1

When a workspace has been created in a dedicated namespace that is later entirely deleted, the corresponding OpenShift project needs to be removed manually to complete the deletion process.

To work around the issue: delete the remaining empty project from the OpenShift console:

```
$ oc delete project <projectname>
```

2.7. THE `CRWCTL SERVER:DELETE` UNINSTALLATION COMMAND DOES NOT REMOVE THE OPENSIFT PROJECT

After using the `crwctl server:delete` command, the OpenShift project that used to host the CodeReady Workspaces instance remains. This makes it impossible to install a new CodeReady Workspaces instance into the default namespace, which still exists. To uninstall CodeReady Workspaces completely, manually remove the namespace.

To work around the issue:

1. Stop the Red Hat CodeReady Workspaces Server:

```
$ crwctl server:stop
```

2. Obtain the name of the CodeReady Workspaces namespace:

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

3. Remove CodeReady Workspaces from the cluster:

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

This removes all CodeReady Workspaces installations from the cluster.

4. Delete the `checluster` object and the `codeready-workspaces` resource:

```
$ oc delete checluster codeready-workspaces --namespace=<openshift_namespace>
```

<openshift_namespace> is the name of the OpenShift project where CodeReady Workspaces is deployed.

5. Delete the OpenShift namespace:

```
$ oc delete project <openshift_namespace>
```

2.8. UNINSTALLATION COMMAND `CRWCTL SERVER:DELETE` MIGHT MAKE OPENSHIFT INSTANCE UNUSABLE

Creation of the CodeReady Workspaces cluster, **codeready-workspaces**, in a namespace can be affected by previous use of the **crwctl server:delete** uninstallation command.

To work around the issue:

1. Patch the Custom Resource Definition:

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

2.9. AN EMBEDDED APPLICATION OF THE "JAVA EAP MAVEN" STACK TENDS TO FAIL AT LAUNCH IN THE DEBUG MODE

An embedded application of the Java EAP Maven stack in the debug mode tends to fail. The dialog window with application URL is already displayed, but the application, according to the terminal output, is still starting. The use of the URL link displayed leads to an error.

2.10. ENTERING A WORKSPACE FAILS AFTER RESTARTING IT

Attempting to restart a workspace and re-enter it fails, and an error message is displayed instead. To work around this issue, restart the workspace again.

2.11. WORKSPACE CAP AND WORKSPACE RAM CAP ORGANIZATION RESTRICTIONS DO NOT WORK

The **Workspace Cap** and **Workspace RAM cap** functions, which control the maximum number of workspaces for an organization and the maximum RAM that organization workspaces can use, currently do not work.

2.12. THE TERMINAL TAB FOR A WORKSPACE IN SOME CASES DOES NOT OPEN

Devfiles contain a set of predefined commands that can be executed in workspaces started using devfiles from the devfile registry. However, when a command defined by a devfile is executed from the workspace, the terminal, in which the commands normally run, does not open.

The work around this issue, do not open the same workspace link in two different browsers.

2.13. WORKSPACES ARE NOT STOPPED BY THE IDLING TIMEOUT

Due to CodeReady Workspaces and OpenShift OAuth integration, workspaces are not stopped by the idling timeout, when the workspace is located in a user's Pod.

To work around this issue, disable the stopping of idling workspace by timeout feature for the workspaces with OpenShift OAuth integration:

```
$ oc patch checluster/eclipse-che --patch '{"spec":{"server":{"customCheProperties":{"CHE_LIMITS_WORKSPACE_IDLE_TIMEOUT": "-1"}}}}' --type=merge -n che
```

2.14. ERROR HIGHLIGHTING AND CODE COMPLETION DO NOT WORK IN A GO DEVFILE

To work around this issue, update the Go language server plug-in to the latest version.

2.15. DEBUGGING UTILITY DOES NOT WORK CORRECTLY ON ITS FIRST RUN IN THE GO DEVFILE WORKSPACE

In workspaces based on the Go devfile, an error notification is displayed during the start of the **Debug current file** configuration.

To work around this issue, execute the predefined **Run current file** command first, then repeat the debugging procedure.

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 *Installation 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 *Installation Guide*.

3. Can I run multiple workspaces simultaneously?

Yes. The following two conditions must be met to run multiple workspaces simultaneously:

- CodeReady Workspaces must use the **per-workspace** Persistent Volume Claim (PVC) strategy (default), and
- Persistent volumes (PVs) must use **ReadWriteMany** (RWX) access mode.
Thus to run multiple workspaces simultaneously, ensure the following configuration is set:
- Set **ReadWriteMany** (RWX) access mode for PVs.
- Use the **per-workspace** PVC strategy (default in CodeReady Workspaces), or optionally, the **unique** strategy.