Red Hat CodeReady Containers 1.19

Release Notes and Known Issues

Highlighted features and identified problems in CodeReady Containers 1.19

Last Updated: 2020-12-04

Red Hat CodeReady Containers 1.19 Release Notes and Known Issues

Highlighted features and identified problems in CodeReady Containers 1.19

Kevin Owen
kowen@redhat.com

Legal Notice

Abstract

This document lists and briefly describes new and improved features of CodeReady Containers 1.19. It also contains information about potential problems you may encounter while using the software. Where possible, workarounds are described for identified issues.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

PART I. RELEASE NOTES

This section documents the most important features and bug fixes in the CodeReady Containers 1.19 product.

CHAPTER 1. COMPONENT VERSIONS

CodeReady Containers 1.19 is shipped with the following versions of the main components:

Table 1.1. CodeReady Containers, Component versions

Component	Version
OpenShift Container Platform	4.6.3
OpenShift client binary (**oc**)	v4.6.3

CHAPTER 2. MINIMUM SYSTEM REQUIREMENTS

CodeReady Containers has the following minimum hardware and operating system requirements.

2.1. HARDWARE REQUIREMENTS

CodeReady Containers requires the following system resources:

- 4 virtual CPUs (vCPUs)
- 9 GB of free memory
- 35 GB of storage space

NOTE

The OpenShift cluster requires these minimum resources to run in the CodeReady Containers virtual machine. Some workloads may require more resources. To assign more resources to the CodeReady Containers virtual machine, see Configuring the virtual machine.

2.2. OPERATING SYSTEM REQUIREMENTS

CodeReady Containers requires the following minimum version of a supported operating system:

2.2.1. Microsoft Windows

- On Microsoft Windows, CodeReady Containers requires the Windows 10 Fall Creators Update (version 1709) or newer. CodeReady Containers does not work on earlier versions of Microsoft Windows. Microsoft Windows 10 Home Edition is not supported.

2.2.2. macOS

- On macOS, CodeReady Containers requires macOS 10.12 Sierra or newer. CodeReady Containers does not work on earlier versions of macOS.

2.2.3. Linux

- On Linux, CodeReady Containers is only supported on Red Hat Enterprise Linux/CentOS 8 and on the latest two stable Fedora releases.
- When using Red Hat Enterprise Linux, the machine running CodeReady Containers must be registered with the Red Hat Customer Portal .
- Ubuntu 18.04 LTS or newer and Debian 10 or newer are not officially supported and may require manual set up of the host machine.
- See Required software packages to install the required packages for your Linux distribution.

CHAPTER 3. CHANGES AND IMPROVEMENTS

This section highlights some of the notable changes introduced in CodeReady Containers 1.19.

3.1. NEW FEATURES

- CodeReady Containers brings a minimal, preconfigured OpenShift Container Platform 4 cluster to your local laptop or desktop computer for development and testing purposes. CodeReady Containers is delivered as a Red Hat Enterprise Linux virtual machine that supports native hypervisors for Linux, macOS, and Microsoft Windows 10.
    - CodeReady Containers is designed for local development and testing on an OpenShift 4 cluster. To run an OpenShift 3 cluster locally, see Red Hat Container Development Kit.
- CodeReady Containers 1.19 is a maintenance release.

3.1.1. Technology Previews

Support for these features falls under the Technology Preview Features Support Scope .

- An experimental system tray is included in this release of CodeReady Containers for macOS and Microsoft Windows. To enable the system tray, run the **crc setup** command with the **--enable-experimental-features** flag:

```
$ crc setup --enable-experimental-features
```

On Microsoft Windows, a prompt will appear requiring you to enter your login credentials.

To start CodeReady Containers from the system tray, you must configure the **pull-secret-file** property:

```
$ crc config set pull-secret-file path-to-pull-secret-file
```

The pull secret file can be downloaded from the Pull Secret section of the Install on Laptop: Red Hat CodeReady Containers page on cloud.redhat.com.

To remove the system tray, run the **crc cleanup** command:

```
$ crc cleanup
```

3.1.2. Usability improvements

- Disk resizing now works the first time the cluster is started through the **crc start** command.

3.2. NOTABLE CHANGES

- CodeReady Containers 1.19 provides OpenShift Container Platform 4.6.3 as the embedded OpenShift version.

PART II. KNOWN ISSUES

This section describes issues that users of CodeReady Containers 1.19 may encounter, as well as possible workarounds for these issues.

CHAPTER 4. GENERAL ISSUES

Issues affecting all supported platforms.

4.1. EMBEDDED CERTIFICATES EXPIRE AFTER 30 DAYS

Each released **crc** binary includes an embedded system bundle that expires 30 days after the release due to certificates embedded in the OpenShift cluster.

The certificate renewal process is not working as intended as of the CodeReady Containers 1.10 release. Install new versions of CodeReady Containers as they become available to ensure your cluster remains operational. For more information about upgrading CodeReady Containers, see Upgrading CodeReady Containers in the Red Hat CodeReady Containers Getting Started Guide .

4.2. METRICS ARE DISABLED BY DEFAULT

To ensure CodeReady Containers can run on a typical laptop, some resource-heavy services are disabled by default. One of these services is Prometheus and all of the related monitoring, alerting, and telemetry functionality.

Enabling these features will require more resources than the CodeReady Containers virtual machine uses by default.

4.3. ENABLING MULTIPLE OPERATORS REQUIRES MORE MEMORY THAN THE DEFAULT

The **crc start** command assigns 9 GiB of memory to the CodeReady Containers virtual machine by default. Enabling multiple Operators may increase memory requirements.

See Configuring the virtual machine in the Red Hat CodeReady Containers Getting Started Guide to assign additional memory.

CHAPTER 5. ISSUES ON LINUX

This section describes CodeReady Containers issues that affect users on a Linux host.

5.1. CODEREADY CONTAINERS 1.19 IS NOT SUPPORTED ON RED HAT ENTERPRISE LINUX 7

The version of **qemu-kvm** provided in Red Hat Enterprise Linux 7 does not support the **q35** machine type used for the CodeReady Containers virtual machine in CodeReady Containers 1.19. As a result, the CodeReady Containers virtual machine cannot be created on Red Hat Enterprise Linux 7.

This issue only exists in CodeReady Containers 1.19 and will be resolved in the next release of CodeReady Containers. Upgrade to the latest CodeReady Containers release once available to resolve this issue.

5.2. FEDORA 33 USES SYSTEMD-RESOLVED INSTEAD OF DNSMASQ

By default, Fedora 33 uses **systemd-resolved** for DNS resolution whereas the **crc setup** command configures DNS through **dnsmasq**. To configure Fedora 33 for split DNS resolution for the CodeReady Containers cluster, follow these steps:

1. Clean up after any previous attempts to start CodeReady Containers:

```
$ crc delete
$ crc cleanup
```

2. Disable the startup checks for **dnsmasq**:

```
$ crc config set skip-check-network-manager-config true
$ crc config set skip-check-crc-dnsmasq-file true
```

3. Create a **/etc/NetworkManager/dispatcher.d/99-crc** file with the following contents:

```
#!/bin/sh
# This is a NetworkManager dispatcher script to configure split DNS for
# the 'crc' libvirt network.
# The corresponding bridge is recreated each time the system reboots, so
# it cannot be configured permanently through NetworkManager.
# Changing DNS settings through nmcli requires the connection to go down/up,
# so we directly do the change through resolvectl

export LC_ALL=C

if [ "$1" = crc ]; then
        resolvectl domain crc ~testing
        resolvectl dns crc 192.168.130.11
        resolvectl default-route crc false
fi

exit 0
```

4. Make the file executable:

```
$ sudo chmod 755 /etc/NetworkManager/dispatcher.d/99-crc
```

5. Start CodeReady Containers:

```
$ crc setup
$ crc start
```

CHAPTER 6. ISSUES ON MACOS

This section describes CodeReady Containers issues that affect users on a macOS host.

6.1. THE SYSTEM TRAY DOES NOT START THE VIRTUAL MACHINE

The experimental macOS system tray cannot start the CodeReady Containers virtual machine. To work around this issue, start the CodeReady Containers virtual machine from the command line using the **crc start** command:

```
$ crc start
```

For more information, see Starting the virtual machine in the Red Hat CodeReady Containers Getting Started Guide.

6.2. THE CRC CLEANUP COMMAND ABORTS WITH AN ERROR

The **crc cleanup** command on macOS returns an error and stops the cleanup process from proceeding.

To resolve this issue, run the following command:

```
$ launchctl stop crc.daemon
```

6.3. HIBERNATION CAUSES VM TIME TO DESYNCHRONIZE

Time in the CodeReady Containers virtual machine can become desynchronized with the time on your host machine. This issue occurs if the CodeReady Containers virtual machine is running when the host machine enters hibernation. To resolve this issue, stop the CodeReady Containers virtual machine and restart it:

```
$ crc stop
$ crc start
```

CHAPTER 7. ISSUES ON MICROSOFT WINDOWS

This section describes CodeReady Containers issues that affect users on a Microsoft Windows host.

7.1. UNEXPECTED BEHAVIOR WHEN RUN OUTSIDE OF %WINDRIVE%

The Hyper-V driver will fail when the **crc** binary is executed from a network drive. The **crc** binary must be placed in a location on **%WINDRIVE%**. **%WINDRIVE%** is normally set to **C:**\.

7.2. CODEREADY CONTAINERS EXPECTS FULLLANGUAGE SUPPORT IN POWERSHELL

The **ConstrainedLanguage** PowerShell mode is supported with exceptions determined by your system administrator.

7.3. THE CRC OC-ENV COMMAND DOES NOT WORK WITH SPECIAL CHARACTERS IN %PATH%

On Microsoft Windows, PowerShell and Command Prompt do not use the UTF-8 encoding. As a result, running the **crc oc-env** command with special characters present in the **%PATH%** will not accurately encode UTF-8 characters. There is no known workaround for this issue.

ADDITIONAL RESOURCES

- See the Red Hat CodeReady Containers Getting Started Guide for an overview of CodeReady Containers features and an introduction to the use of **OpenShift Container Platform**.
- Report issues with CodeReady Containers or request new features using the **OpenShift Container Platform** product with the **crc** component on Red Hat BugZilla.