Red Hat CodeReady Containers 1.6

Getting Started Guide

Quick-start guide to using and developing with CodeReady Containers

Last Updated: 2020-03-26

Red Hat CodeReady Containers 1.6 Getting Started Guide

Quick-start guide to using and developing with CodeReady Containers

Kevin Owen
kowen@redhat.com

Legal Notice

Abstract

This guide shows how to get up to speed using CodeReady Containers. Included instructions and examples guide through first steps developing containerized applications using Red Hat OpenShift Container Platform 4 from a host workstation (Microsoft Windows, macOS, or Red Hat Enterprise Linux).

Table of Contents

CHAPTER 1. INTRODUCING RED HAT CODEREADY CONTAINERS

1.1. UNDERSTANDING CODEREADY CONTAINERS

Red Hat CodeReady Containers brings a minimal OpenShift 4 cluster to your local computer. This cluster provides a minimal environment for development and testing purposes. It's mainly targetted at running on developers' desktops. For other use cases, such as headless, multi-developer or team-based setups, use of the full-fledged OpenShift installer is recommended.

You can refer to the OpenShift documentation for a more in-depth introduction to OpenShift.

CodeReady Containers includes the **crc** command-line interface (CLI) to interact with the CodeReady Containers virtual machine running the OpenShift cluster.

1.2. DIFFERENCES FROM A PRODUCTION OPENSHIFT INSTALL

Red Hat CodeReady Containers is a regular OpenShift installation with the following notable differences:

- It uses a single node which behaves both as a master and as a worker node.

- The **machine-config** and **monitoring** Operators are disabled by default.

- These disabled Operators will cause the corresponding parts of the web console to be non functional.

- For the same reason, there is currently no upgrade path to newer OpenShift versions.

- Due to technical limitations, the CodeReady Containers cluster is ephemeral and will need to be recreated from scratch once a month using a newer release.

- The OpenShift instance is running in a virtual machine, which could cause some other differences, in particular in relation with external networking.

CHAPTER 2. INSTALLATION

2.1. MINIMUM SYSTEM REQUIREMENTS

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

2.1.1. Hardware requirements

CodeReady Containers requires the following system resources:

- 4 virtual CPUs (vCPUs)

- 8 GB of memory

- 35 GB of storage space

NOTE

These requirements must be met in order to run OpenShift in the CodeReady Containers virtual machine. Depending on the workload, more resources need to be assigned.

2.1.2. Operating system requirements

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

2.1.2.1. Microsoft Windows

- On Microsoft Windows, CodeReady Containers requires the Windows 10 Pro Fall Creators Update (version 1709). CodeReady Containers does not work on earlier versions or editions of Microsoft Windows.

2.1.2.2. macOS

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

2.1.2.3. Linux

- On Linux, CodeReady Containers is only supported on Red Hat Enterprise Linux/CentOS 7.5 or newer (including 8.x versions) and on the latest two stable Fedora releases.

- When using Red Hat Enterprise Linux, the machine running CodeReady Containers must be registered with 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 Section 2.2, "Required software packages" to install the required packages for your Linux distribution.

2.2. REQUIRED SOFTWARE PACKAGES

CodeReady Containers requires the **libvirt** and **NetworkManager** packages. Consult the following table to determine the command used to install these packages for your Linux distribution.

Table 2.1. Package installation commands by distribution

Linux Distribution	Installation command
Fedora	**sudo dnf install NetworkManager**
Red Hat Enterprise Linux/CentOS	**su -c 'yum install NetworkManager'**
Debian/Ubuntu	**sudo apt install qemu-kvm libvirt-daemon libvirt-daemon-system network-manager**

2.3. INSTALLING CODEREADY CONTAINERS

Prerequisites

- Your host machine must meet the minimum system requirements. For more information, see Section 2.1, "Minimum system requirements".

Procedure

1. Download the latest release of CodeReady Containers for your platform and extract the contents of the archive to a location in your *PATH*.

2.4. UPGRADING CODEREADY CONTAINERS

Newer versions of the CodeReady Containers binary require manual set up to prevent potential incompatilibities with previous versions.

Procedure

- Download the latest release of CodeReady Containers .

- Delete the existing CodeReady Containers virtual machine:

      $ crc delete

> ⚠️ **WARNING**
>
> The **crc delete** command will result in the loss of data stored in the CodeReady Containers virtual machine. Save any desired instance information before running this command.

- Replace the previous **crc** binary with the binary of the latest release. Verify that the new **crc** binary is in use by checking its version:

```
$ crc version
```

- Set up the new CodeReady Containers release:

```
$ crc setup
```

- Start the new CodeReady Containers virtual machine:

```
$ crc start
```

CHAPTER 3. USING CODEREADY CONTAINERS

3.1. SETTING UP CODEREADY CONTAINERS

The **crc setup** command performs operations to set up the environment of your host machine for the CodeReady Containers virtual machine.

This procedure will create the **~/.crc** directory if it does not already exist.

Prerequisites

- Your user account must have permission to use the **sudo** command.

> **NOTE**
>
> - The **crc** binary should not be run as **root** (or Administrator). The **crc** binary should always be run with your user account.
>
> - If you are setting up a new version, capture any changes made to the instance before setting up a new CodeReady Containers release.

Procedure

1. Set up your host machine for CodeReady Containers:

   ```
   $ crc setup
   ```

3.2. STARTING THE VIRTUAL MACHINE

The **crc start** command starts the CodeReady Containers virtual machine and OpenShift cluster.

Prerequisites

- The host machine has been set up using the **crc setup** command. For more information, see Section 3.1, "Setting up CodeReady Containers".

- A valid OpenShift user pull secret. The pull secret can be copied or downloaded from the Pull Secret section of the Install on Laptop: Red Hat CodeReady Containers page on cloud.redhat.com.

> **NOTE**
>
> A Red Hat account is required in order to access the user pull secret.

Procedure

- Start the CodeReady Containers virtual machine:

  - For native hypervisors:

    ```
    $ crc start
    ```

- When prompted, supply your user pull secret.

> **NOTE**
>
> - The cluster takes a minimum of four minutes to start the necessary containers and operators before serving a request.
>
> - If errors are encountered during **crc start**, check the Chapter 6, *Troubleshooting Red Hat CodeReady Containers* section for potential solutions.

3.3. ACCESSING THE OPENSHIFT CLUSTER

The OpenShift cluster running in the CodeReady Containers virtual machine can be accessed via the OpenShift web console or the OpenShift client binary (**oc**).

3.3.1. Accessing the OpenShift web console

Prerequisites

- A running CodeReady Containers virtual machine. For more information, see Section 3.2, "Starting the virtual machine".

Procedure

To access the OpenShift web console, follow these steps:

1. Run **crc console**. This will open your web browser and direct it to the web console.

2. Log in to the OpenShift web console as the **developer** user with the password printed in the output of the **crc start** command.

> **NOTE**
>
> - You can also view the password for the **developer** user by running **crc console --credentials**.
>
> - The cluster can initially be accessed through either the **kubeadmin** or **developer** user. For creating projects or OpenShift applications, and for application deployment, the **developer** user should be used. The **kubeadmin** user should only be used for administrative tasks such as creating new users, setting roles, and so on.

See Chapter 6, *Troubleshooting Red Hat CodeReady Containers* if you cannot access the CodeReady Containers OpenShift cluster.

Additional resources

- The OpenShift documentation covers the creation of projects and applications.

3.3.2. Accessing the OpenShift cluster with oc

Prerequisites

- A running CodeReady Containers virtual machine. For more information, see Section 3.2, "Starting the virtual machine".

Procedure

To access the OpenShift cluster via the **oc** command, follow these steps:

1. Run the **crc oc-env** command to print the command needed to add the cached **oc** binary to your *PATH*:

   ```
   $ crc oc-env
   ```

2. Run the printed command.

3. Log in as the **developer** user:

   ```
   $ oc login -u developer https://api.crc.testing:6443
   ```

   > **NOTE**
   >
   > The password for the **developer** user was printed in the output of the **crc start** command. You can also view it by running the **crc console --credentials** command.

4. You can now use **oc** to interact with your OpenShift cluster. For example, to verify that the OpenShift cluster operators are available:

   ```
   $ oc get co
   ```

   > **NOTE**
   >
   > - The **machine-config** cluster operator is expected to report **False** availability.
   >
   > - The **monitoring** cluster operator is expected to report **Unknown** availability.

See Chapter 6, *Troubleshooting Red Hat CodeReady Containers* if you cannot access the CodeReady Containers OpenShift cluster.

Additional resources

- The OpenShift documentation covers the creation of projects and applications.

- OpenShift Do (**odo**) can also be used to create OpenShift projects and applications from the command line.

3.4. STOPPING THE VIRTUAL MACHINE

The **crc stop** command stops the running CodeReady Containers virtual machine and OpenShift cluster. The stopping process will take a few minutes while the cluster shuts down.

Procedure

- Stop the CodeReady Containers virtual machine and OpenShift cluster:

```
$ crc stop
```

3.5. DELETING THE VIRTUAL MACHINE

The **crc delete** command deletes an existing CodeReady Containers virtual machine.

Procedure

- Delete the CodeReady Containers virtual machine:

```
$ crc delete
```

> **WARNING**
>
> The **crc delete** command will result in the loss of data stored in the CodeReady Containers virtual machine. Save any desired instance information before running this command.

CHAPTER 4. NETWORKING

4.1. DNS CONFIGURATION DETAILS

4.1.1. Linux

- CodeReady Containers expects NetworkManager to be used to manage networking.

- NetworkManager is set up to use **dnsmasq** through the **/etc/NetworkManager/conf.d/crc-nm-dnsmasq.conf** configuration file.

- The configuration file for this **dnsmasq** instance is **/etc/NetworkManager/dnsmasq.d/crc.conf**:

  ```
  server=/crc.testing/192.168.130.11
  server=/apps-crc.testing/192.168.130.11
  ```

  - The NetworkManager **dnsmasq** instance forwards requests for the **crc.testing** and **apps-crc.testing** domains to the **192.168.130.11** DNS server. This DNS server is a **dnsmasq** instance running inside of the virtual machine.

4.1.2. macOS

- CodeReady Containers creates a **/etc/resolver/testing** file which instructs macOS to forward all DNS requests for the **testing** domain to the CodeReady Containers virtual machine.

- CodeReady Containers also adds an **api.crc.testing** entry to **/etc/hosts** pointing at the VM IP address. This is needed by the **oc** binary. See OpenShift issue #23266 for more information.

4.2. STARTING CODEREADY CONTAINERS BEHIND A PROXY

Prerequisites

- To use an existing **oc** binary on your host machine, export the **.testing** domain as part of the **no_proxy** environment variable.

- No manual settings are required to use the embedded **oc** binary. For more information on using the embedded **oc** binary, see Section 3.3.2, "Accessing the OpenShift cluster with **oc**".

Procedure

- To start CodeReady Containers behind a proxy:

  1. Define a proxy using the **http_proxy** and **https_proxy** environment variables or using the **crc config set** command as follows:

     ```
     $ crc config set http-proxy http://example.proxy.com:<port>
     $ crc config set https-proxy http://example.proxy.com:<port>
     $ crc config set no-proxy <comma-separated-no-proxy-entries>
     ```

     The **crc** binary will be able to use the defined proxy once set via environment variables or CodeReady Containers configuration.

NOTE

- Proxy-related values set in the configuration for CodeReady Containers have priority over values set via environment variables.

- SOCKS proxies are not supported by OpenShift Container Platform.

CHAPTER 5. ADMINISTRATIVE TASKS

5.1. STARTING MONITORING, ALERTING, AND TELEMETRY

In order to make sure CodeReady Containers can run on a typical laptop, some resource–heavy services get disabled by default. One of these is Prometheus and all the related monitoring, alerting, and telemetry functionality, with the latter being responsible for your cluster being listed in the Red Hat OpenShift Cluster Manager.

Prerequisites

- A running CodeReady Containers virtual machine and a working **oc** command. For more information, see Section 3.3.2, "Accessing the OpenShift cluster with **oc**".

Procedure

1. Start monitoring, alerting, and telemetry services:

   ```
   $ oc scale --replicas=1 statefulset --all -n openshift-monitoring; oc scale --replicas=1
   deployment --all -n openshift-monitoring
   ```

CHAPTER 6. TROUBLESHOOTING RED HAT CODEREADY CONTAINERS

NOTE

The goal of Red Hat CodeReady Containers is to deliver an OpenShift environment for development and testing purposes. Issues occurring during installation or usage of specific OpenShift applications are outside of the scope of CodeReady Containers and should be reported to the relevant project. For example, OpenShift issues are tracked on GitHub.

6.1. BASIC TROUBLESHOOTING

The majority of issues can be resolved by stopping a running CodeReady Containers virtual machine, deleting the virtual machine, and starting a new instance of the virtual machine.

Prerequisites

- The host machine has been set up using the **crc setup** command. For more information, see Section 3.1, "Setting up CodeReady Containers".

- The virtual machine has been started using the **crc start** command. For more information, see Section 3.2, "Starting the virtual machine".

- You are using the latest CodeReady Containers release. The generated certificates for the embedded system image bundle expire after 30 days. Using an older system image bundle may result in errors related to expired x509 certificates. For more information, see Section 6.3, "Troubleshooting expired certificates".

Procedure

To troubleshoot CodeReady Containers, perform the following steps:

1. Stop the CodeReady Containers virtual machine:

   ```
   $ crc stop
   ```

2. Delete the CodeReady Containers virtual machine:

   ```
   $ crc delete
   ```

3. Start the CodeReady Containers virtual machine:

   ```
   $ crc start
   ```

   NOTE

   The cluster takes a minimum of four minutes to start the necessary containers and operators before serving a request.

If your issue has not been resolved by this procedure, perform the following steps:

1. Search open issues for the issue that you are encountering.

2. If no existing issue addresses the encountered issue, create an issue and attach the ~/.crc/crc.log file to the created issue. The ~/.crc/crc.log file contains detailed debugging and troubleshooting information which can help diagnose the problem that you are experiencing.

6.2. GETTING SHELL ACCESS TO THE OPENSHIFT CLUSTER

Direct access to the OpenShift cluster is not needed for regular use and is strongly discouraged. If direct access to the cluster is needed for troubleshooting or debugging purposes, follow this procedure.

Prerequisites

- **oc** access to the cluster is enabled and you are logged in as the **kubeadmin** user. For detailed steps, see Section 3.3.2, "Accessing the OpenShift cluster with **oc**".

Procedure

1. Run **oc get nodes**. The output will be similar to this:

   ```
   $ oc get nodes
   NAME              STATUS   ROLES         AGE    VERSION
   crc-shdl4-master-0   Ready    master,worker   7d7h   v1.14.6+7e13ab9a7
   ```

2. Run **oc debug nodes/<node>** where **<node>** is the name of the node which was printed in the previous step.

6.3. TROUBLESHOOTING EXPIRED CERTIFICATES

Prior to Red Hat CodeReady Containers 1.2.0, the system bundle included in each released **crc** binary expired 30 days after the release due to certificates embedded in the OpenShift cluster. Using an older **crc** binary or system bundle will result in an expired certificates error.

Starting from CodeReady Containers 1.2.0, the embedded certificates can be automatically renewed by **crc**. The certificate renewal process is triggered when needed when **crc start** is run. When certificates need to be renewed, this can add up to five minutes to the start time of the cluster.

Procedure

With CodeReady Containers releases older than 1.2.0, to resolve expired certificate errors:

1. Download the latest CodeReady Containers release and place the **crc** binary in your **$PATH**.

2. Remove the cluster with certificate errors using the **crc delete** command:

   ```
   $ crc delete
   ```

> **⚠ WARNING**
>
> The **crc delete** command will result in the loss of data stored in the CodeReady Containers virtual machine. Save any desired instance information before running this command.

3. Set up the new release:

```
$ crc setup
```

4. Start the new cluster:

```
$ crc start
```

6.4. TROUBLESHOOTING BUNDLE VERSION MISMATCH

CodeReady Containers instances are created with bundle information and instance data. When a new CodeReady Containers release is set up, the bundle information and instance data is not updated due to customization in the previous instance data. This will lead to errors when running the **crc start** command:

```
$ crc start
...
FATA Bundle 'crc_hyperkit_4.2.8.crcbundle' was requested, but the existing VM is using
'crc_hyperkit_4.2.2.crcbundle'
```

Procedure

1. Issue the **crc delete** command before attempting to start the instance:

```
$ crc delete
```

> **⚠ WARNING**
>
> The **crc delete** command will result in the loss of data stored in the CodeReady Containers virtual machine. Save any desired instance information before running this command.