

What is a CodeReadyContainer ??

In simple words, it's an OpenShift Cluster on your Linux machine

CodeReady Containers is the quickest way to get started building OpenShift clusters. It is designed to run on a local computer to simplify setup and testing and emulate the cloud development environment locally with all of the tools needed to develop container-based applications.

Let's start with setting up the cluster.

1. You need to have an account at <https://cloud.redhat.com/>
2. If you don't have an account you need to create it.

Hardware requirements

- 4 physical CPU cores
- 9 GB of free memory
- 35 GB of storage space

- Ubuntu 18.04 LTS or later and Debian 10 or later are not supported and may require manual set up of the host machine.

Required software packages for Linux

```
sudo apt install qemu-kvm libvirt-daemon libvirt-daemon-system  
network-manager
```

3. Keep this link open in a new tab

<https://console.redhat.com/openshift/create/local>

4. On the above link, you can see the download option called

“Download CodeReadyContainer”

1

Download what you need to get started

CodeReady Containers

Download and extract the CodeReady Containers archive for your operating system and place the binary in your `$PATH`.

Linux ▼

x86_64 ▼

[Download CodeReady Containers](#)

Pull secret

Download or copy your pull secret. You'll be prompted for this information during installation.

[Download pull secret](#)

 [Copy pull secret](#)

5. Let's Extract the contents of the archive.

```
$ tar xvf crc-linux-amd64.tar.xz
```

6. Let's Create the `~/bin` directory if it does not exist and copy the `crc` executable to it.

```
$ mkdir -p ~/bin
```

```
$ cp ~/Downloads/crc-linux-*-amd64/crc ~/bin
```

7. Now, Add the `~/bin` directory to your `$PATH`:

```
$ export PATH=$PATH:$HOME/bin
```

```
$ echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
```

8. Now, the next step is to copy the **secret and store** it in a file so you will need it while setting up the OpenShift Cluster.

1

Download what you need to get started

CodeReady Containers

Download and extract the CodeReady Containers archive for your operating system and place the binary in your **\$PATH**.

Linux ▾


x86_64 ▾

Download CodeReady Containers

Pull secret

Download or copy your pull secret. You'll be prompted for this information during installation.

Download pull secret

 Copy pull secret

Note: You can see the pull secret above in the image, similarly you'll see it on your console.

9. Now run the below command to start the **CRC** setup.

```
$ crc setup
```

Note: After running this command there will be lots of things going but at one step you'll be asked to paste the **pull secret**. So paste the pull secret which we stored above step no 8.

10. After the CRC setup is done successfully, you can start the cluster using.

```
$ crc start
```

You will see an output like this

```
> crc start
INFO Checking if running as non-root
INFO Checking if running inside WSL2
INFO Checking if crc-admin-helper executable is cached
INFO Checking for obsolete admin-helper executable
INFO Checking if running on a supported CPU architecture
INFO Checking minimum RAM requirements
INFO Checking if crc executable symlink exists
INFO Checking if Virtualization is enabled
INFO Checking if KVM is enabled
INFO Checking if libvirt is installed
INFO Checking if user is part of libvirt group
INFO Checking if active user/process is currently part of the libvirt group
INFO Checking if libvirt daemon is running
INFO Checking if a supported libvirt version is installed
INFO Checking if crc-driver-libvirt is installed
```

```
The server is accessible via web console at:
  https://console-openshift-console.apps-crc.testing

Log in as administrator:
  Username: kubeadmin
  Password: yQ8tf-Kkvea-nUHd6-Km8hR

Log in as user:
  Username: developer
  Password: developer

Use the 'oc' command line interface:
  $ eval $(crc oc-env)
  $ oc login -u developer https://api.crc.testing:6443
```

11. To use the **oc** command

```
$ eval $(crc oc-env)
```

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

12. To get the console credentials.

```
$ crc console --credentials
```

To login as a regular user, run 'oc login -u developer -p developer https://api.crc.testing:6443'.

To login as an admin, run 'oc login -u kubeadmin -p yQ8tf-Kkvea-nUHd6-Km8hR https://api.crc.testing:6443'

13. You can log in through the console now.

