How to install Docker CE on RHEL 8 / CentOS 8

8 May 2020 by Egidio Docile

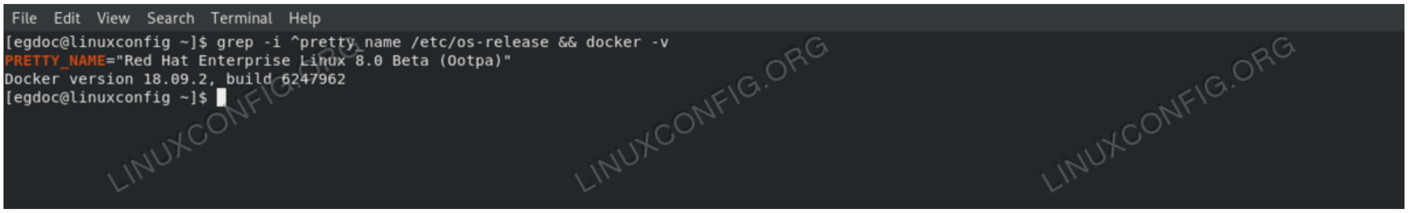
The latest release of the RHEL 8 / CentOS 8. Red Hat has built its own tools, `buildah` and `podman` , which aim to be compatible with existing docker images and work without relying on a daemon, allowing the creation of containers as normal users, without the need of special permissions (with some limitations: e.g. at the moment of writing, it's still not possible to map host ports to the container without privileges).

Some specific tools, however, are still missing: an equivalent of `docker-compose` , for example does not exists yet. In this tutorial we will see how to install and run the original Docker CE on Rhel8 by using the official Docker repository for CentOS7.

In this tutorial you will learn:

- How to enable the docker-ce repository on RHEL 8 / CentOS 8

- How to install docker and docker-compose on RHEL 8 / CentOS 8



```
[egdoc@linuxconfig ~]$ grep -i ^pretty_name /etc/os-release && docker -v
PRETTY_NAME="Red Hat Enterprise Linux 8.0 Beta (Ootpa)"
Docker version 18.09.2, build 6247962
[egdoc@linuxconfig ~]$
```

Docker installed on RHEL 8 / CentOS 8

Software Requirements and Conventions Used

Software Requirements and Linux Command Line Conventions

Category	Requirements, Conventions or Software Version Used
System	RHEL 8 / CentOS 8
Software	Docker version 18.09.2
Other	Permission to run command with root privileges.
Conventions	**#** – requires given linux commands to be executed with root privileges either directly as a root user or by use of `sudo` command **$** – requires given linux commands to be executed as a regular non-privileged user

What is Docker?

Docker is an open source project which allows the creation and distribution of applications inside `containers` , which are standardized environments that can be easily replicated, independently from the host system. While in Red Hat Enterprise Linux 7 Docker was officially supported, on the new release of this open source operating system, it has been replaced by a series of other tools developed by Red Hat itself: `buildah` and `podman` .

By the use of an external repository, however, it's still possible to install Docker CE (Community Edition). In this tutorial we will see how to install this repository; notice however, that it was originally meant for `CentOS 7` (a RHEL clone), and the community version of Docker has no official support for Red Hat Enterprise Linux. Because of this, issues exist – we discuss them below.

Adding the external repository

Since Docker is not available on RHEL 8 / CentOS 8, we need to add an external repository to obtain the software. In this case we will use the official Docker CE CentOS repository: this is, at the moment of writing, the only way to install Docker CE on RHEL 8 / CentOS 8.

The `dnf config-manager` utility let us, among the other things, easily enable or disable a repository in our distribution. By default, only the `appstream` and `baseos` repositories are enabled on Rhel8; we need to add and enable also the `docker-ce` repo. All we need to do to accomplish this task, is to run the following command:

```
$ sudo dnf config-manager --add-repo=https://download.docker.com/linux/cent
```

We can verify that the repository has been enabled, by looking at the output of the following command:

```
$ sudo dnf repolist -v
```

The command above will return detailed information about all the enabled repositories. This is what you should see at this point:

```
Repo-id      : docker-ce-stable
Repo-name    : Docker CE Stable - x86_64
Repo-revision: 1549905809
Repo-updated : Mon 11 Feb 2019 06:23:29 PM CET
Repo-pkgs    : 30
Repo-size    : 618 M
Repo-baseurl : https://download.docker.com/linux/centos/7/x86_64/stable
Repo-expire  : 172,800 second(s) (last: Mon 18 Feb 2019 10:23:54 AM CET)
Repo-filename: /etc/yum.repos.d/docker-ce.repo

Repo-id      : rhel-8-for-x86_64-appstream-rpms
Repo-name    : Red Hat Enterprise Linux 8 for x86_64 - AppStream Beta (RPMs
Repo-revision: 1542158694
Repo-updated : Wed 14 Nov 2018 02:24:54 AM CET
Repo-pkgs    : 4,594
Repo-size    : 4.9 G
Repo-baseurl : https://cdn.redhat.com/content/beta/rhel8/8/x86_64/appstrea
Repo-expire  : 86,400 second(s) (last: Mon 18 Feb 2019 10:23:55 AM CET)
Repo-filename: /etc/yum.repos.d/redhat.repo

Repo-id      : rhel-8-for-x86_64-baseos-rpms
Repo-name    : Red Hat Enterprise Linux 8 for x86_64 - BaseOS Beta (RPMs)
Repo-revision: 1542158719
Repo-updated : Wed 14 Nov 2018 02:25:19 AM CET
Repo-pkgs    : 1,686
Repo-size    : 925 M
Repo-baseurl : https://cdn.redhat.com/content/beta/rhel8/8/x86_64/baseos/o
Repo-expire  : 86,400 second(s) (last: Mon 18 Feb 2019 10:23:56 AM CET)
Repo-filename: /etc/yum.repos.d/redhat.repo
Total packages: 6,310
```

Installing docker-ce

The `docker-ce-stable` repository is now enabled on our system. The repository contains several versions of the `docker-ce` package, to display all of them, we can run:

```
$ dnf list docker-ce --showduplicates | sort -r
docker-ce.x86_64            3:19.03.2-3.el7                     docker-ce-s
docker-ce.x86_64            3:19.03.1-3.el7                     docker-ce-s
docker-ce.x86_64            3:19.03.0-3.el7                     docker-ce-s
docker-ce.x86_64            3:18.09.9-3.el7                     docker-ce-s
docker-ce.x86_64            3:18.09.8-3.el7                     docker-ce-s
docker-ce.x86_64            3:18.09.7-3.el7                     docker-ce-s
docker-ce.x86_64            3:18.09.6-3.el7                     docker-ce-s
docker-ce.x86_64            3:18.09.5-3.el7                     docker-ce-s
docker-ce.x86_64            3:18.09.4-3.el7                     docker-ce-s
docker-ce.x86_64            3:18.09.3-3.el7                     docker-ce-s
docker-ce.x86_64            3:18.09.2-3.el7                     docker-ce-s
docker-ce.x86_64            3:18.09.1-3.el7                     docker-ce-s
docker-ce.x86_64            3:18.09.0-3.el7                     docker-ce-s
docker-ce.x86_64            18.06.3.ce-3.el7                    docker-ce-s
docker-ce.x86_64            18.06.2.ce-3.el7                    docker-ce-s
docker-ce.x86_64            18.06.1.ce-3.el7                    docker-ce-s
docker-ce.x86_64            18.06.0.ce-3.el7                    docker-ce-s
docker-ce.x86_64            18.03.1.ce-1.el7.centos             docker-ce-s
docker-ce.x86_64            18.03.0.ce-1.el7.centos             docker-ce-s
docker-ce.x86_64            17.12.1.ce-1.el7.centos             docker-ce-s
docker-ce.x86_64            17.12.0.ce-1.el7.centos             docker-ce-s
docker-ce.x86_64            17.09.1.ce-1.el7.centos             docker-ce-s
docker-ce.x86_64            17.09.0.ce-1.el7.centos             docker-ce-s
docker-ce.x86_64            17.06.2.ce-1.el7.centos             docker-ce-s
docker-ce.x86_64            17.06.1.ce-1.el7.centos             docker-ce-s
docker-ce.x86_64            17.06.0.ce-1.el7.centos             docker-ce-s
docker-ce.x86_64            17.03.3.ce-1.el7                    docker-ce-s
docker-ce.x86_64            17.03.2.ce-1.el7.centos             docker-ce-s
docker-ce.x86_64            17.03.1.ce-1.el7.centos             docker-ce-s
docker-ce.x86_64            17.03.0.ce-1.el7.centos             docker-ce-s
```

What version to install? Well, Red Hat seems to have somehow blocked the installation of `containerd.io` > `1.2.0-3.el7`, which is a dependency of `docker-ce`. Because of this, simply running the `sudo dnf install docker-ce` command, won't work. As we will see in a minute, it's still possibile to workaround this problem; once `docker-ce` is installed, however, another problem becomes evident: as long as `firewalld`, the system firewall manager is enabled, `DNS resolution` inside docker containers does not work.

This is, of course a critical problem. However, if you still want to proceed with the installation, here are the possible methods that can be used to avoid the dependencies issues:

- Install a specific version of `docker-ce` which requires an installable version of the `containerd.io` package;
- Force the installation providing the `--nobest` option
- Install the latest available `containerd.io` rpm manually;

Install a specific version of docker-ce

At the moment of writing the versions of `docker-ce` that are installed without problems are:

- docker-ce-3:18.09.1-3.el7
- docker-ce-18.06.3.ce-3.el7;
- docker-ce-17.12.1.ce-1.el7.centos

To install a specific version, all we have to do is to provide the fully qualified package name, for example:

```
$ sudo dnf install docker-ce-3:18.09.1-3.el7
```

Force the installation of docker-ce with the –nobest option

Normally, when installing a package, the best available candidate is selected from a repository. In this case, for example, the installation of the latest version of `docker-ce` is attempted (and fails). By using the `--nobest` option, we can change this behavior so that the first version of `docker-ce` with satisfiable dependencies is selected as "fallback", in this case `3:18.09.1-3.el7`.

```
$ sudo dnf install --nobest docker-ce
Dependencies resolved.

Problem: package docker-ce-3:19.03.2-3.el7.x86_64 requires containerd.io >=
  - cannot install the best candidate for the job
  - package containerd.io-1.2.2-3.3.el7.x86_64 is excluded
  - package containerd.io-1.2.2-3.el7.x86_64 is excluded
  - package containerd.io-1.2.4-3.1.el7.x86_64 is excluded
  - package containerd.io-1.2.5-3.1.el7.x86_64 is excluded
  - package containerd.io-1.2.6-3.3.el7.x86_64 is excluded
```

```
================================================================================
 Package                            Arch         Version
================================================================================
Installing:
 docker-ce                          x86_64       3:18.09.1-3.el7
Installing dependencies:
 containerd.io                      x86_64       1.2.0-3.el7
 docker-ce-cli                      x86_64       1:19.03.2-3.el7
 container-selinux                  noarch       2:2.94-1.git1e99f1d.modul
 tar                                x86_64       2:1.30-4.el8
 libcgroup                          x86_64       0.41-19.el8
 python3-policycoreutils            noarch       2.8-16.1.el8
 python3-libsemanage                x86_64       2.8-5.el8
 python3-setools                    x86_64       4.2.0-2.el8
 checkpolicy                        x86_64       2.8-2.el8
 python3-audit                      x86_64       3.0-0.10.20180831git0047a
 policycoreutils-python-utils       noarch       2.8-16.1.el8
Skipping packages with broken dependencies:
 docker-ce                          x86_64       3:19.03.2-3.el7

Transaction Summary
================================================================================
Install  12 Packages
Skip      1 Package

Total download size: 85 M
Installed size: 351 M
Is this ok [y/N]:
```

Install the latest available containerd.io package manually

If we stricly need to install the latest version of `docker-ce` , we can install the required version of `containerd.io` manually, by running:

```
$ sudo dnf install https://download.docker.com/linux/centos/7/x86_64/stable
```

After the package is installed, we can simply install the latest `docker-ce` :

```
$ sudo dnf install docker-ce
Dependencies resolved.
================================================================================
  Package                          Arch                     Version
================================================================================
Installing:
  docker-ce                        x86_64                   3:19.03.2-3.e
Installing dependencies:
  docker-ce-cli                      x86_64                    1:19.03.2-3.e
  tar                                x86_64                    2:1.30-4.el8
  libcgroup                          x86_64                    0.41-19.el8

Transaction Summary
===========================================================================
Install  4 Packages

Total download size: 65 M
Installed size: 275 M
Is this ok [y/N]:
```

This option is less convenient since the `containerd.io` package is not installed as a dependency of `docker-ce`, therefore it will not be removed automatically when the latter is uninstalled from the system.

Whatever method we use to install `docker-ce`, as said before, in order to make `DNS resolution` work inside Docker containers, we must disable firewalld (a system reboot may be also needed):

```
$ sudo systemctl disable firewalld
```

Start and enable the docker daemon

Once `docker-ce` is installed, we must start and enable the docker daemon, so that it will be also launched automatically at boot. The command we need to run is the following:

```
$ sudo systemctl enable --now docker
```

At this point, we can confirm that the daemon is active by running:

```
$ systemctl is-active docker
active
```

Similarly, we can check that it is enabled at boot, by running:

```
$ systemctl is-enabled docker
enabled
```

Installing docker-compose

Docker compose is a very useful package which let us manage multi-container applications, like for example those based on the LAMP stack, where each part of the environment (PHP, Apache, MariaDB) is provided by a dedicated container (if you are interested in the subject, take a look at our tutorial about creating a docker-based lamp stack). The package is not available on Rhel8, nor an equivalent exists to be used with the Rhel tools. It's, however, possible to install it in many ways: just keep on reading and decide what suits you best.

Global installation

The way we should install `docker-compose` varies depending on whether we want to install it globally or just for a single user. At the moment of writing, the only way to install it globally is to download the binary from the github page of the project:

```
om/docker/compose/releases/download/1.23.2/docker-compose-$(uname -s)-$(uname
```

Once the binary is downloaded, we move it into `/usr/local/bin` and we make it executable:

```
$ sudo mv docker-compose /usr/local/bin && sudo chmod +x /usr/local/bin/doc
```

The `/usr/local` hierarchy is not chosen randomly. This directory structure is made to be used for files installed by the local administrator manually (for software compiled from source, for example), in order to ensure separation from the software installed with the system package manager.

Although it's possible for a normal user to run docker-related commands if he is part of the `docker` group (the group is automatically created when we install docker-ce), by default they must be executed with root privileges for security reasons. When we need to do the latter, since the `/usr/local/bin` directory is not in the root user's `PATH`, we either need to call the binary specifying its location or add `/usr/local/bin` to the `PATH` itself. The first option is the one which I recommend in this case.

Per-user installation

If our user is part of the `docker` group, and thus it is allowed to run docker commands, and since `docker-compose` is available as a python package, we can also install it using `pip`, the python package manager. First, make sure pip itself is installed:

```
$ sudo dnf install python3-pip
```

To obtain docker-compose we run:

```
$ pip3.6 install docker-compose --user
```

Ad
KodeKloud
Prepare For Your Kubernetes Certification
Become a DevOps expert now!

Please notice that even if would be possible to run pip as root to install a package globally, this is not recommended and highly discouraged.

Testing docker

We installed docker and docker-compose, now to check that everything works as expected, we can try to build an image and run a container: in this case we will use the official `httpd` one. All we have to do is to launch the following command:

```
sudo docker run --rm --name=linuxconfig-test -p 80:80 httpd
```

Since the `httpd` image does not exists locally it will be automatically fetched and built. Finally, a container based on it will be launched in the foreground (it will be automatically removed when stopped). We should be able to see the `It works!` message when we reach our machine ip via browser.

Conclusions

Red Hat Enterprise Linux 8 does not support Docker: on this distribution it has been replaced by Red Hat own tools like `buildah` and `podman` , which are compatible with Docker but don't need a server/client architecture to run. Using native tools, where possible, is always the recommended way to go, but for a reason or another you may still want to install the original Docker. In this tutorial, we saw how it is possible to install `Docker CE` on Rhel8, by using the official Docker repository for CentOS7, which is a 100% compatible clone.

This is not an ideal solution, and as we saw, at the moment, some workarounds are needed to make Docker work on RHEL8. If some new issues arises, or better solutions to the problems mentioned above are found, this article will be updated accordingly. Stay tuned.

Related Linux Tutorials:

- Install Docker on AlmaLinux
- How to connect to Docker container via ssh
- Manjaro Linux Docker installation
- How to install and configure Docker-ce/Moby Engine on Fedora...
- How to install Debian on an existing LUKS container
- How to export repositories with the git-daemon
- How to Install Kubernetes on All Linux Distros
- How to use a file as a LUKS device key
- How to migrate from CentOS to AlmaLinux
- CentOS vs CentOS Stream

Redhat / CentOS / AlmaLinux

administration, centos8, development, docker, rhel8

‹ How to add a user to a group on RHEL 8 / CentOS 8

› How To Install RHEL 8 Step by Step with Screenshots

NEWSLETTER

Subscribe to Linux Career Newsletter to receive latest news, jobs, career advice and featured configuration tutorials.

SUBSCRIBE

WRITE FOR US

LinuxConfig is looking for a technical writer(s) geared towards GNU/Linux and FLOSS technologies. Your articles will feature various GNU/Linux configuration tutorials and FLOSS technologies used in combination with GNU/Linux operating system.

When writing your articles you will be expected to be able to keep up with a technological advancement regarding the above mentioned technical area of expertise. You will work independently and be able to produce at minimum 2 technical articles a month.

APPLY NOW

TAGS

18.04 administration apache applications backup bash beginner browser centos centos8 commands database debian desktop development docker fedora filesystem firewall gaming gnome Hardware installation java kali manjaro multimedia networking nvidia programming python redhat rhel8 scripting security server ssh storage terminal ubuntu ubuntu 20.04 video virtualization webapp webserver

ABOUT US

FEATURED TUTORIALS

VIM tutorial for beginners

How to install the NVIDIA drivers on Ubuntu 20.04 Focal Fossa Linux

Bash Scripting Tutorial for Beginners

How to check CentOS version

How to find my IP address on Ubuntu 20.04 Focal Fossa Linux

Ubuntu 20.04 Remote Desktop Access from Windows 10

Howto mount USB drive in Linux

How to install missing ifconfig command on Debian Linux

AMD Radeon Ubuntu 20.04 Driver Installation

Ubuntu Static IP configuration

How to use bash array in a shell script

Linux IP forwarding – How to Disable/Enable

How to install Tweak Tool on Ubuntu 20.04 LTS Focal Fossa Linux

How to enable/disable firewall on Ubuntu 18.04 Bionic Beaver Linux

Netplan static IP on Ubuntu configuration

How to change from default to alternative Python version on Debian Linux

Set Kali root password and enable root login

How to Install Adobe Acrobat Reader on Ubuntu 20.04 Focal Fossa Linux

How to install the NVIDIA drivers on Ubuntu 18.04 Bionic Beaver Linux

How to check NVIDIA driver version on your Linux system

Nvidia RTX 3080 Ethereum Hashrate and Mining Overclock settings on HiveOS Linux

LATEST TUTORIALS

Introduction to pandas

Kubernetes and Linux: Is It a Good Combo?

Kubernetes vs. Docker Swarm: A Beginner's Comparison

How to Install Kubernetes on Linux Mint

How to Restart Kubernetes on Linux

Linux System Requirements for Kubernetes

How to Install Kubernetes on Rocky Linux

Kubernetes Basics: Understanding Pods, Services, and Deployments

How to Install Kubernetes on Ubuntu Linux

How to join line on Linux

How to run JAR file on Linux

How to query NTP server

How to format USB with exFAT on Linux

How to Check Kubernetes Version on Linux

How to Disable Swap in Linux for Kubernetes

How to check what shell I'm using on Linux

How to open file explorer from terminal on Linux

How to quit telnet

How to paste into terminal

How to pipe output to a file on Linux