**How to Install Docker on CentOS 8**

Posted on [March 23, 2020](https://www.liquidweb.com/kb/how-to-install-docker-on-centos-8/) by [Ellen Sletton](https://www.liquidweb.com/kb/author/esletton/) | Updated: June 12, 2020  
Category: [Tutorials](https://www.liquidweb.com/kb/category/tutorials/) | Tags: [Automation](https://www.liquidweb.com/kb/tag/automation/), [Cent OS](https://www.liquidweb.com/kb/tag/cent-os/), [CentOS 8](https://www.liquidweb.com/kb/tag/centos-8/), [Container](https://www.liquidweb.com/kb/tag/container/), [Containerization](https://www.liquidweb.com/kb/tag/containerization/), [Developer](https://www.liquidweb.com/kb/tag/developer/), [Development](https://www.liquidweb.com/kb/tag/development/), [DevOps](https://www.liquidweb.com/kb/tag/devops/), [Docker](https://www.liquidweb.com/kb/tag/docker/), [Environment](https://www.liquidweb.com/kb/tag/environment/), [Git](https://www.liquidweb.com/kb/tag/git/), [Hypervisor](https://www.liquidweb.com/kb/tag/hypervisor/), [Image](https://www.liquidweb.com/kb/tag/image/), [Install](https://www.liquidweb.com/kb/tag/install/), [Installation](https://www.liquidweb.com/kb/tag/installation/), [Kubectl](https://www.liquidweb.com/kb/tag/kubectl/), [Kubernetes](https://www.liquidweb.com/kb/tag/kubernetes/), [KVM](https://www.liquidweb.com/kb/tag/kvm/), [Linux](https://www.liquidweb.com/kb/tag/linux/), [MiniKube](https://www.liquidweb.com/kb/tag/minikube/), [Platform](https://www.liquidweb.com/kb/tag/platform/), [Server Administration](https://www.liquidweb.com/kb/tag/server-administration/), [Tools](https://www.liquidweb.com/kb/tag/tools/), [Virtual Machines](https://www.liquidweb.com/kb/tag/virtual-machines/), [VirtualBox](https://www.liquidweb.com/kb/tag/virtualbox/), [Virtualization](https://www.liquidweb.com/kb/tag/virtualization/)

Reading Time: 6 minutes

What is Docker?

Before we begin, let’s describe what Docker is. Docker is a set of virtualization tools that allows us to create, test, and deploy containerized applications quickly and easily on a [dedicated server](https://www.liquidweb.com/products/dedicated/). It has become very popular and used almost everywhere in our daily lives. Thanks to containerization, we can quickly launch applications on different [cloud platforms](https://www.liquidweb.com/products/private-cloud/) utilizing small bundles which contain all the needed packages, libraries and configuration file to run an application. These docker packages communicate via established network channels.

Docker Silhouette

Docker has two main versions:

* Docker CE (Community Edition)
* Docker EE (Enterprise Edition)

We will not be discussing many of the new features of CentOS 8 that function with Docker, as RedHat has decided not to provide official support for Docker. But, Redhat does offer several built-in tools for [creating and managing container images within a server cluster](https://www.liquidweb.com/products/server-clusters/), more specifically Buildah and Podman. These tools are docker compatible, but they require a server/client architecture. Many prefer to use Docker containerization, and in this tutorial, we want to demonstrate how to install Docker on CentOS using an external repository. So let’s begin!

Install Docker CE

To get started, we should remove all existing docker applications (that might currently be installed) in order to make sure we are using a clean installation. You may not have some of these components installed.

[root@host ~]# yum remove docker docker-client docker-client-latest docker-common docker-latest docker-latest-logrotate docker-logrotate docker-engine

Modular dependency problems:

Problem 1: conflicting requests

- nothing provides module(perl:5.26) needed by module perl-DBD-SQLite:1.58:8010020191114033549:073fa5fe-0.x86\_64

Problem 2: conflicting requests

- nothing provides module(perl:5.26) needed by module perl-DBI:1.641:8010020191113222731:16b3ab4d-0.x86\_64

No match for argument: docker

No match for argument: docker-client

No match for argument: docker-client-latest

No match for argument: docker-common

No match for argument: docker-latest

No match for argument: docker-latest-logrotate

No match for argument: docker-logrotate

No match for argument: docker-engine

No packages marked for removal.

Dependencies resolved.

Nothing to do.

Complete!

[root@host ~]#

Next, we want to add an external repository that will help us obtain the Docker software.

[root@host ~]# dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo

Adding repo from: https://download.docker.com/linux/centos/docker-ce.repo

[root@host ~]#

Now, let’s check what versions of docker we have in the repository.

[root@host ~]# dnf list docker-ce

CentOS-8 - AppStream 7.7 kB/s | 4.3 kB 00:00

CentOS-8 - Base 32 kB/s | 3.8 kB 00:00

CentOS-8 - Extras 4.0 kB/s | 1.5 kB 00:00

Docker CE Stable - x86\_64 55 kB/s | 22 kB 00:00

Available Packages

docker-ce.x86\_64 3:19.03.7-3.el7 docker-ce-stable

[root@host ~]#

Unfortunately, the latest version of Docker is not always in the available in all repositories. To resolve this and obtain the most suitable, and stable release version, we will use the “*nobest*” flag. We will use this option because it will allow CentOS 8 to install the most appropriate version for our needs. If we need to [install the latest version of docker](https://www.liquidweb.com/kb/install-docker-ubuntu-16-04/), this option would not the right choice for us. We will, however, continue to install the version that best matches our CentOS server version.

[root@host ~]# dnf install docker-ce --nobest

CentOS-8 - AppStream 16 kB/s | 4.3 kB 00:00

CentOS-8 - Base 10 kB/s | 3.8 kB 00:00

CentOS-8 - Extras 12 kB/s | 1.5 kB 00:00

Docker CE Stable - x86\_64 15 kB/s | 3.5 kB 00:00

Dependencies resolved.

Problem: package docker-ce-3:19.03.7-3.el7.x86\_64 requires containerd.io >= 1.2.2-3, but none of the providers can be installed

- cannot install the best candidate for the job

- package containerd.io-1.2.10-3.2.el7.x86\_64 is excluded

- package containerd.io-1.2.13-3.1.el7.x86\_64 is excluded

- 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 Repository Size

============================================================================

Installing:

docker-ce x86\_64 3:18.09.1-3.el7 docker-ce-stable 19 M

Installing dependencies:

libcgroup x86\_64 0.41-19.el8 BaseOS 70 k

containerd.io x86\_64 1.2.0-3.el7 docker-ce-stable 22 M

docker-ce-cli x86\_64 1:19.03.7-3.el7 docker-ce-stable 40 M

Skipping packages with broken dependencies:

docker-ce x86\_64 3:19.03.7-3.el7 docker-ce-stable 25 M

Transaction Summary

============================================================================

Install 4 Packages

Skip 1 Package

Total download size: 80 M

Installed size: 339 M

Is this ok [y/N]: y

Downloading Packages:

(1/4): libcgroup-0.41-19.el8.x86\_64.rpm 2.0 MB/s | 70 kB 00:00

(2/4): docker-ce-18.09.1-3.el7.x86\_64.rpm 2.9 MB/s | 19 MB 00:06

(3/4): containerd.io-1.2.0-3.el7.x86\_64.rpm 2.9 MB/s | 22 MB 00:07

(4/4): docker-ce-cli-19.03.7-3.el7.x86\_64.rpm 4.9 MB/s | 40 MB 00:08

----------------------------------------------------------------------------

Total 9.7 MB/s | 80 MB 00:08

warning: /var/cache/dnf/docker-ce-stable-091d8a9c23201250/packages/containerd.io-1.2.0-3.el7.x86\_64.rpm: Header V4 RSA/SHA512 Signature, key ID 621e9f35: NOKEY

Docker CE Stable - x86\_64 8.4 kB/s | 1.6 kB 00:00

Importing GPG key 0x621E9F35:

Userid : "Docker Release (CE rpm) <docker@docker.com>"

Fingerprint: 060A 61C5 1B55 8A7F 742B 77AA C52F EB6B 621E 9F35

From : https://download.docker.com/linux/centos/gpg

Is this ok [y/N]: y

Key imported successfully

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

Preparing : 1/1

Installing : docker-ce-cli-1:19.03.7-3.el7.x86\_64 1/4

Running scriptlet: docker-ce-cli-1:19.03.7-3.el7.x86\_64 1/4

The fast memory caches was not invalidated by NSS responder.

Installing : containerd.io-1.2.0-3.el7.x86\_64 2/4

Running scriptlet: containerd.io-1.2.0-3.el7.x86\_64 2/4

Running scriptlet: libcgroup-0.41-19.el8.x86\_64 3/4

Installing : libcgroup-0.41-19.el8.x86\_64 3/4

Running scriptlet: libcgroup-0.41-19.el8.x86\_64 3/4

Running scriptlet: docker-ce-3:18.09.1-3.el7.x86\_64 4/4

Installing : docker-ce-3:18.09.1-3.el7.x86\_64 4/4

Running scriptlet: docker-ce-3:18.09.1-3.el7.x86\_64 4/4

Verifying : libcgroup-0.41-19.el8.x86\_64 1/4

Verifying : containerd.io-1.2.0-3.el7.x86\_64 2/4

Verifying : docker-ce-3:18.09.1-3.el7.x86\_64 3/4

Verifying : docker-ce-cli-1:19.03.7-3.el7.x86\_64 4/4

Installed:

docker-ce-3:18.09.1-3.el7.x86\_64 libcgroup-0.41-19.el8.x86\_64

containerd.io-1.2.0-3.el7.x86\_64 docker-ce-cli-1:19.03.7-3.el7.x86\_64

Skipped:

docker-ce-3:19.03.7-3.el7.x86\_64

Complete!

[root@host ~]#

Verify Docker Version



Now, we can double-check if the installation was successful by using the version command.

[root@host ~]# docker -v

Docker version 19.03.7, build 7141c199a2

[root@host ~]#

Launch Docker

Next, lets start the Docker daemon with the systemctl command.

[root@host ~]# systemctl enable --now docker

Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.

[root@host ~]#

Confirm Install

Next, we can finally confirm that [Docker](https://www.liquidweb.com/kb/how-to-install-docker-on-centos-7/) has started and is working with the following command.

[root@host ~]# systemctl status docker

● docker.service - Docker Application Container Engine

Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor pres>

Active: active (running) since Thu 2020-03-05 06:30:02 EST; 2min 30s ago

Docs: https://docs.docker.com

Main PID: 2798 (dockerd)

Tasks: 21

Memory: 140.7M

CGroup: /system.slice/docker.service

├─2798 /usr/bin/dockerd -H fd://

└─2805 containerd --config /var/run/docker/containerd/containerd.tom>

Mar 05 06:29:52 localhost.localdomain dockerd[2798]: time="2020-03-05T06:29:52.>

Mar 05 06:29:52 localhost.localdomain dockerd[2798]: time="2020-03-05T06:29:52.>

Mar 05 06:29:52 localhost.localdomain dockerd[2798]: time="2020-03-05T06:29:52.>

Mar 05 06:29:52 localhost.localdomain dockerd[2798]: time="2020-03-05T06:29:52.>

Mar 05 06:29:57 localhost.localdomain dockerd[2798]: time="2020-03-05T06:29:57.>

Mar 05 06:29:59 localhost.localdomain dockerd[2798]: time="2020-03-05T06:29:58.>

Mar 05 06:30:01 localhost.localdomain dockerd[2798]: time="2020-03-05T06:30:01.>

Mar 05 06:30:01 localhost.localdomain dockerd[2798]: time="2020-03-05T06:30:01.>

Mar 05 06:30:02 localhost.localdomain dockerd[2798]: time="2020-03-05T06:30:02.>

Mar 05 06:30:02 localhost.localdomain systemd[1]: Started Docker Application Co>

[root@host ~]#

Add User

The next step is to add a user to the Docker user group employing the “*usermod*” command.

Additionally, when we use the “*id root*” command after adding the user, this will show us whether the user is now included within the docker group.

[root@host ~]# usermod -aG docker root

[root@host ~]# id root

uid=0(root) gid=0(root) groups=0(root),976(docker)

[root@host ~]#

Disable Firewall

Next, we are going to temporarily turn off firewalld to allow Docker containers to work internally.

[root@host ~]# systemctl disable firewalld

Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.

Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.

[root@host ~]#

**Caution:**

We recommend you perform a reboot after this step.

Deploy Test App

In our last step, we will install a test application called “*hello-world*” to ensure our installation is working as expected. We will deploy the program using the following “*docker*” command.

[root@host ~]# docker run hello-world

Unable to find image 'hello-world:latest' locally

latest: Pulling from library/hello-world

1b930d010525: Pull complete

Digest: sha256:fc6a51919cfeb2e6763f62b6d9e8815acbf7cd2e476ea353743570610737b752

Status: Downloaded newer image for hello-world:latest

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.

2. The Docker daemon pulled the "hello-world" image from the Docker Hub.

(amd64)

3. The Docker daemon created a new container from that image which runs the

executable that produces the output you are currently reading.

4. The Docker daemon streamed that output to the Docker client, which sent it

to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:

https://hub.docker.com/

For more examples and ideas, visit:

https://docs.docker.com/get-started/

[root@host ~]#

Install Docker Compose

As an additional step, we will install Docker Compose. Docker Compose is a very useful application which allows us to connect several containers together using minimal commands, or when we need to run several containers together which depend on each other.

We will start the installation using the following “*dnf*“command.

[root@host ~]# dnf install curl -y

Last metadata expiration check: 0:42:26 ago on Thu 05 Mar 2020 06:08:47 AM EST.

Package curl-7.61.1-11.el8.x86\_64 is already installed.

Dependencies resolved.

Nothing to do.

Complete!

[root@host ~]#

Obtaining the Docker Compose Package

We will obtain the Docker Compose package from GitHub.

[root@host ~]# curl -L "https://github.com/docker/compose/releases/download/1.25.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 617 100 617 0 0 1672 0 --:--:-- --:--:-- --:--:-- 1672

100 16.2M 100 16.2M 0 0 3914k 0 0:00:04 0:00:04 --:--:-- 4990k

[root@host ~]

Next, we will adjust the permissions for the file using the “*chmod*” command to make it executable.

[root@host ~]# chmod +x /usr/local/bin/docker-compose

[root@host ~]#

To check which version we have installed, run the following command.

[root@host ~]# docker-compose -v

docker-compose version 1.25.0, build 0a186604

[root@host ~]#

Conclusion  
In this article, we have discussed the basic concepts of Docker and how to install it in CentOS 8. Additionally, we installed Docker compose. Docker is one of the most useful tools when a modern containerization platform is needed.

If you would like to know more about how best to utilize this new technology, give us a call at 800.580.4985, or open a [chat](https://www.liquidweb.com/kb/how-to-install-docker-on-centos-8/) or [ticket](mailto:support@liquidweb.com) with us to speak with one of our knowledgeable Solutions or Experienced Hosting advisors!