

Install Docker Engine on CentOS

To get started with Docker Engine on CentOS, make sure you meet the prerequisites, and then follow the installation steps.

Prerequisites

OS requirements

To install Docker Engine, you need a maintained version of one of the following CentOS versions:

- CentOS 7
- CentOS 8 (stream)
- CentOS 9 (stream)

The `centos-extras` repository must be enabled. This repository is enabled by default, but if you have disabled it, you need to re-enable it (<https://wiki.centos.org/AdditionalResources/Repositories>).

Uninstall old versions

Older versions of Docker went by the names of `docker` or `docker-engine`. Uninstall any such older versions before attempting to install a new version, along with associated dependencies.

```
$ sudo yum remove docker \
    docker-client \
    docker-client-latest \
    docker-common \
    docker-latest \
    docker-latest-logrotate \
    docker-logrotate \
    docker-engine
```



`yum` might report that you have none of these packages installed.

Images, containers, volumes, and networks stored in `/var/lib/docker/` aren't automatically removed when you uninstall Docker.

Installation methods

You can install Docker Engine in different ways, depending on your needs:

- You can set up Docker's repositories and install from them, for ease of installation and upgrade tasks. This is the recommended approach.
- You can download the RPM package and install it manually and manage upgrades completely manually. This is useful in situations such as installing Docker on air-gapped systems with no access to the internet.
- In testing and development environments, you can use automated convenience scripts to install Docker.

Install using the rpm repository

Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.

Set up the repository

Install the `yum-utils` package (which provides the `yum-config-manager` utility) and set up the repository.

```
yum install -y yum-utils  
yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```



Install Docker Engine

1. Install Docker Engine, containerd, and Docker Compose:

Latest

Specific version

To install the latest version, run:

```
docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```



If prompted to accept the GPG key, verify that the fingerprint matches `060A 61C5 1B55 8A7F 742B 77AA C52F EB6B 621E 9F35`, and if so, accept it.

This command installs Docker, but it doesn't start Docker. It also creates a `docker` group, however, it doesn't add any users to the group by default.

2. Start Docker.

```
$ sudo systemctl start docker
```



3. Verify that the Docker Engine installation is successful by running the `hello-world` image.

```
$ sudo docker run hello-world
```



This command downloads a test image and runs it in a container. When the container runs, it prints a confirmation message and exits.

You have now successfully installed and started Docker Engine.

Tip

Receiving errors when trying to run without root?

The `docker` user group exists but contains no users, which is why you're required to use `sudo` to run Docker commands. Continue to [Linux postinstall \(/engine/install/linux-postinstall\)](/engine/install/linux-postinstall) to allow non-privileged users to run Docker commands and for other optional configuration steps.

Upgrade Docker Engine

To upgrade Docker Engine, follow the installation instructions, choosing the new version you want to install.

Install from a package

If you can't use Docker's `rpm` repository to install Docker Engine, you can download the `.rpm` file for your release and install it manually. You need to download a new file each time you want to upgrade Docker Engine.

1. Go to <https://download.docker.com/linux/centos/> (<https://download.docker.com/linux/centos/>) and choose your version of CentOS. Then browse to `x86_64/stable/Packages/` and download the `.rpm` file for the Docker version you want to install.
2. Install Docker Engine, changing the path below to the path where you downloaded the Docker package.

```
$ sudo yum install /path/to/package.rpm
```



Docker is installed but not started. The `docker` group is created, but no users are added to the group.

3. Start Docker.

```
$ sudo systemctl start docker
```



4. Verify that the Docker Engine installation is successful by running the `hello-world` image.

```
$ sudo docker run hello-world
```



This command downloads a test image and runs it in a container. When the container runs, it prints a confirmation message and exits.

You have now successfully installed and started Docker Engine.

Tip

Receiving errors when trying to run without root?

The `docker` user group exists but contains no users, which is why you're required to use `sudo` to run Docker commands. Continue to [Linux postinstall \(/engine/install/linux-postinstall\)](/engine/install/linux-postinstall) to allow non-privileged users to run Docker commands and for other optional configuration steps.

Upgrade Docker Engine

To upgrade Docker Engine, download the newer package files and repeat the installation procedure, using `yum -y upgrade` instead of `yum -y install`, and point to the new files.

Install using the convenience script

Docker provides a convenience script at <https://get.docker.com/> (<https://get.docker.com/>) to install Docker into development environments non-interactively. The convenience script isn't recommended for production environments, but it's useful for creating a provisioning script tailored to your needs. Also refer to the install using the repository steps to learn about installation steps to install using the package repository. The source code for the script is open source, and you can find it in the `docker-install` repository on GitHub (<https://github.com/docker/docker-install>).

Always examine scripts downloaded from the internet before running them locally. Before installing, make yourself familiar with potential risks and limitations of the convenience script:

- The script requires `root` or `sudo` privileges to run.
- The script attempts to detect your Linux distribution and version and configure your package management system for you.

- The script doesn't allow you to customize most installation parameters.
- The script installs dependencies and recommendations without asking for confirmation. This may install a large number of packages, depending on the current configuration of your host machine.
- By default, the script installs the latest stable release of Docker, containerd, and runc. When using this script to provision a machine, this may result in unexpected major version upgrades of Docker. Always test upgrades in a test environment before deploying to your production systems.
- The script isn't designed to upgrade an existing Docker installation. When using the script to update an existing installation, dependencies may not be updated to the expected version, resulting in outdated versions.

i Tip: preview script steps before running

You can run the script with the `--dry-run` option to learn what steps the script will run when invoked:

```
$ curl -fsSL https://get.docker.com -o get-docker.sh
$ sudo sh ./get-docker.sh --dry-run
```



This example downloads the script from <https://get.docker.com/> (<https://get.docker.com/>) and runs it to install the latest stable release of Docker on Linux:

```
$ curl -fsSL https://get.docker.com -o get-docker.sh
$ sudo sh get-docker.sh
Executing docker install script, commit: 7cae5f8b0decc17d6571f9f52eb840fbc13b2737
<...>
```



You have now successfully installed and started Docker Engine. The `docker` service starts automatically on Debian based distributions. On `RPM` based distributions, such as CentOS, Fedora, RHEL or SLES, you need to start it manually using the appropriate `systemctl` or `service` command. As the message indicates, non-root users can't run Docker commands by default.

i Use Docker as a non-privileged user, or install in rootless mode?

The installation script requires `root` or `sudo` privileges to install and use Docker. If you want to grant non-root users access to Docker, refer to the [post-installation steps for Linux](https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user) ([/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user](https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user)). You can also install Docker without `root` privileges, or configured to run in rootless mode. For instructions on running Docker in rootless mode, refer to [run the Docker daemon as a non-root user \(rootless mode\)](https://docs.docker.com/engine/security/rootless/) ([/engine/security/rootless/](https://docs.docker.com/engine/security/rootless/)).

Install pre-releases

Docker also provides a convenience script at <https://test.docker.com/> (<https://test.docker.com/>) to install pre-releases of Docker on Linux. This script is equal to the script at get.docker.com, but configures your package manager to use the test channel of the Docker package repository. The test channel includes both stable and pre-releases (beta versions, release-candidates) of Docker. Use this script to get early access to new releases, and to evaluate them in a testing environment before they're released as stable.

To install the latest version of Docker on Linux from the test channel, run:

```
$ curl -fsSL https://test.docker.com -o test-docker.sh
$ sudo sh test-docker.sh
```



Upgrade Docker after using the convenience script

If you installed Docker using the convenience script, you should upgrade Docker using your package manager directly. There's no advantage to re-running the convenience script. Re-running it can cause issues if it attempts to re-install repositories which already exist on the host machine.

Uninstall Docker Engine

1. Uninstall the Docker Engine, CLI, containerd, and Docker Compose packages:

```
$ sudo yum remove docker-ce docker-ce-cli containerd.io docker-buildx-plugin doc
```



2. Images, containers, volumes, or custom configuration files on your host aren't automatically removed.

To delete all images, containers, and volumes:

```
$ sudo rm -rf /var/lib/docker
$ sudo rm -rf /var/lib/containerd
```



You have to delete any edited configuration files manually.

Next steps

- Continue to Post-installation steps for Linux (</engine/install/linux-postinstall/>).
- Review the topics in Develop with Docker (</develop/>) to learn how to build new applications using Docker.