

Install Docker Engine on Ubuntu

Estimated reading time: 12 minutes

❗ Docker Desktop for Linux

Docker Desktop helps you build, share, and run containers easily on Mac and Windows as you do on Linux. Docker handles the complex setup and allows you to focus on writing the code. Thanks to the positive support we received on the subscription updates (<https://www.docker.com/blog/updating-product-subscriptions/>), we've started working on Docker Desktop for Linux (<https://www.docker.com/blog/accelerating-new-features-in-docker-desktop/>) which is the second-most popular feature request in our public roadmap. If you are interested in early access, sign up for our Developer Preview program (<https://www.docker.com/community/get-involved/developer-preview>).

To get started with Docker Engine on Ubuntu, make sure you meet the prerequisites, then install Docker.

Prerequisites

OS requirements

To install Docker Engine, you need the 64-bit version of one of these Ubuntu versions:

- Ubuntu Hirsute 21.04
- Ubuntu Focal 20.04 (LTS)
- Ubuntu Bionic 18.04 (LTS)

Docker Engine is supported on `x86_64` (or `amd64`), `armhf`, `arm64`, and `s390x` architectures.

❗ Ubuntu 16.04 LTS "Xenial Xerus" end-of-life

Ubuntu Linux 16.04 LTS reached the end of its five-year LTS window on April 30th 2021 and is no longer supported. Docker no longer releases packages for this distribution (including patch- and security releases). Users running Docker on Ubuntu 16.04 are recommended to update their system to a currently supported LTS version of Ubuntu.

Uninstall old versions

Older versions of Docker were called `docker` , `docker.io` , or `docker-engine` . If these are installed, uninstall them:

```
$ sudo apt-get remove docker docker-engine docker.io containerd runc
```

It's OK if `apt-get` reports that none of these packages are installed.

The contents of `/var/lib/docker/` , including images, containers, volumes, and networks, are preserved. If you do not need to save your existing data, and want to start with a clean installation, refer to the uninstall Docker Engine section at the bottom of this page.

Supported storage drivers

Docker Engine on Ubuntu supports `overlay2` , `aufs` and `btrfs` storage drivers.

Docker Engine uses the `overlay2` storage driver by default. If you need to use `aufs` instead, you need to configure it manually. See use the AUFS storage driver (`/storage/storagedriver/aufs-driver/`)

Installation methods

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

- Most users set up Docker's repositories and install from them, for ease of installation and upgrade tasks. This is the recommended approach.
- Some users download the DEB 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, some users choose to use automated convenience scripts to install Docker.

Install using the 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

1. Update the `apt` package index and install packages to allow `apt` to use a repository over HTTPS:

```
$ sudo apt-get update

$ sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
```

2. Add Docker's official GPG key:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/sh
```

3. Use the following command to set up the **stable** repository. To add the **nightly** or **test** repository, add the word `nightly` or `test` (or both) after the word `stable` in the commands below. Learn about **nightly** and **test** channels (/engine/install/).

Note: The `lsb_release -cs` sub-command below returns the name of your Ubuntu distribution, such as `xenial`. Sometimes, in a distribution like Linux Mint, you might need to change `$(lsb_release -cs)` to your parent Ubuntu distribution. For example, if you are using `Linux Mint Tessa`, you could use `bionic`. Docker does not offer any guarantees on untested and unsupported Ubuntu distributions.

```
$ echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-ke
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Install Docker Engine

1. Update the `apt` package index, and install the *latest version* of Docker Engine and containerd, or go to the next step to install a specific version:

```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

✔ Got multiple Docker repositories?

If you have multiple Docker repositories enabled, installing or updating without specifying a version in the `apt-get install` or `apt-get update` command always installs the highest possible version, which may not be appropriate for your stability needs.

2. To install a *specific version* of Docker Engine, list the available versions in the repo, then select and install:

a. List the versions available in your repo:

```
$ apt-cache madison docker-ce
```

```
docker-ce | 5:18.09.1~3-0~ubuntu-xenial | https://download.docker.com/linux/ubuntu xen
docker-ce | 5:18.09.0~3-0~ubuntu-xenial | https://download.docker.com/linux/ubuntu xen
docker-ce | 18.06.1~ce~3-0~ubuntu       | https://download.docker.com/linux/ubuntu xen
docker-ce | 18.06.0~ce~3-0~ubuntu               | https://download.docker.com/linux/ubuntu xen
```

b. Install a specific version using the version string from the second column, for example,

```
5:18.09.1~3-0~ubuntu-xenial .
```

```
$ sudo apt-get install docker-ce=<VERSION_STRING> docker-ce-cli=<VERSION_STRING> containerd.io
```

3. Verify that Docker Engine is installed correctly 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 message and exits.

Docker Engine is installed and running. The `docker` group is created but no users are added to it. You need to use `sudo` to run Docker commands. Continue to 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, first run `sudo apt-get update`, then follow the installation instructions, choosing the new version you want to install.

Install from a package

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

1. Go to <https://download.docker.com/linux/ubuntu/dists/>

(<https://download.docker.com/linux/ubuntu/dists/>), choose your Ubuntu version, then browse to [pool/stable/](#), choose `amd64`, `armhf`, `arm64`, or `s390x`, and download the `.deb` file for the Docker Engine version you want to install.

✔ Note

To install a **nightly** or **test** (pre-release) package, change the word `stable` in the above URL to `nightly` or `test`. Learn about **nightly** and **test** channels (/engine/install/).

2. Install Docker Engine, changing the path below to the path where you downloaded the Docker package.

```
$ sudo dpkg -i /path/to/package.deb
```

The Docker daemon starts automatically.

3. Verify that Docker Engine is installed correctly 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 message and exits.

Docker Engine is installed and running. The `docker` group is created but no users are added to it. You need to use `sudo` to run Docker commands. Continue to Post-installation steps for Linux (/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 file and repeat the installation procedure, pointing to the new file.

Install using the convenience script

Docker provides a convenience script at get.docker.com (<https://get.docker.com/>) to install Docker into development environments quickly and non-interactively. The convenience script is not recommended for production environments, but can be used as an example to create a provisioning script that is 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 can be found 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, and does not 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 (major) upgrades in a test environment before deploying to your production systems.
- The script is not 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, causing outdated versions to be used.

✔ **Tip: preview script steps before running**

You can run the script with the `DRY_RUN=1` option to learn what steps the script will execute during installation:

```
$ curl -fsSL https://get.docker.com -o get-docker.sh
$ DRY_RUN=1 sh ./get-docker.sh
```

This example downloads the script from 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
<...>
```

Docker is installed. 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 cannot run Docker commands by default.

🔒 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 (`/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user`). Docker can also be installed 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) (`/engine/security/rootless/`).

Install pre-releases

Docker also provides a convenience script at `test.docker.com` (`https://test.docker.com/`) to install pre-releases of Docker on Linux. This script is equivalent to the script at `get.docker.com`, but configures your package manager to enable the “test” channel from our package repository, which 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 are 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 is no advantage to re-running the convenience script, and it can cause issues if it attempts to re-add repositories which have already been added to the host machine.

Uninstall Docker Engine

1. Uninstall the Docker Engine, CLI, and Containerd packages:

```
$ sudo apt-get purge docker-ce docker-ce-cli containerd.io
```

2. Images, containers, volumes, or customized configuration files on your host are not automatically removed. To delete all images, containers, and volumes:

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

You must 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.

[requirements \(/search/?q=requirements\)](/search/?q=requirements), [apt \(/search/?q=apt\)](/search/?q=apt), [installation \(/search/?q=installation\)](/search/?q=installation), [ubuntu \(/search/?q=ubuntu\)](/search/?q=ubuntu), [install \(/search/?q=install\)](/search/?q=install), [uninstall \(/search/?q=uninstall\)](/search/?q=uninstall), [upgrade \(/search/?q=upgrade\)](/search/?q=upgrade), [update \(/search/?q=update\)](/search/?q=update)