

# Install Docker Engine on Ubuntu

To get started with Docker Engine on Ubuntu, make sure you [meet the prerequisites](#), and then follow the [installation steps](#).

## Prerequisites

### “ Note

If you use ufw or firewalld to manage firewall settings, be aware that when you expose container ports using Docker, these ports bypass your firewall rules. For more information, refer to [Docker and ufw](#).

## OS requirements

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

- Ubuntu Lunar 23.04
- Ubuntu Kinetic 22.10
- Ubuntu Jammy 22.04 (LTS)
- Ubuntu Focal 20.04 (LTS)

Docker Engine for Ubuntu is compatible with x86\_64 (or amd64), armhf, arm64, s390x, and ppc64le (ppc64el) architectures.

## Uninstall old versions

Before you can install Docker Engine, you must first make sure that any conflicting packages are uninstalled.

Distro maintainers provide an unofficial distributions of Docker packages in APT. You must uninstall these packages before you can install the official version of Docker Engine.

The unofficial packages to uninstall are:

- `docker.io`
- `docker-compose`
- `docker-doc`
- `podman-docker`

Moreover, Docker Engine depends on `containerd` and `runc`. Docker Engine bundles these dependencies as one bundle: `containerd.io`. If you have installed the `containerd` or `runc` previously, uninstall them to avoid conflicts with the versions bundled with Docker Engine.

Run the following command to uninstall all conflicting packages:

```
$ for pkg in docker.io docker-doc docker-compose podman-docker containerd runc; do sudo apt-get remove $pkg; done
```

`apt-get` 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. If you want to start with a clean installation, and prefer to clean up any existing data, read the [uninstall Docker Engine](#) section.

## Installation methods

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

- Docker Engine comes bundled with [Docker Desktop for Linux](#). This is the easiest and quickest way to get started.
- Set up and install Docker Engine from [Docker's `apt` repository](#).
- [Install it manually](#) and manage upgrades manually.
- Use a [convenience script](#). Only recommended for testing and development environments.

## Install using the apt 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
```

2. Add Docker's official GPG key:

```
$ sudo install -m 0755 -d /etc/apt/keyrings
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
$ sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

3. Use the following command to set up the repository:

```
$ echo \
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

#### “ **Note**

If you use an Ubuntu derivative distro, such as Linux Mint, you may need to use `UBUNTU_CODENAME` instead of `VERSION_CODENAME`.

4. Update the `apt` package index:

```
$ sudo apt-get update
```

## Install Docker Engine

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

`Latest` `Specific version`

To install the latest version, run:

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-
compose-plugin
```

2. 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](#) 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 `apt` 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 Engine.

1. Go to <https://download.docker.com/linux/ubuntu/dists/> [open in new](#).
2. Select your Ubuntu version in the list.
3. Go to `pool/stable/` and select the applicable architecture (`amd64`, `armhf`, `arm64`, or `s390x`).
4. Download the following `deb` files for the Docker Engine, CLI, containerd, and Docker Compose packages:
  - `containerd.io_<version>_<arch>.deb`
  - `docker-ce_<version>_<arch>.deb`
  - `docker-ce-cli_<version>_<arch>.deb`
  - `docker-buildx-plugin_<version>_<arch>.deb`
  - `docker-compose-plugin_<version>_<arch>.deb`

5. Install the `.deb` packages. Update the paths in the following example to where you downloaded the Docker packages.

```
$ sudo dpkg -i ./containerd.io_<version>_<arch>.deb \  
./docker-ce_<version>_<arch>.deb \  
./docker-ce-cli_<version>_<arch>.deb \  
./docker-buildx-plugin_<version>_<arch>.deb \  
./docker-compose-plugin_<version>_<arch>.deb
```

The Docker daemon starts automatically.

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

```
$ sudo service docker start  
$ 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](#) 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](#), pointing to the new files.

## Install using the convenience script

Docker provides a convenience script at [https://get.docker.com/open\\_in\\_new](https://get.docker.com/open_in_new) 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](#)[open in new](#).

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.

#### 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/open\\_in\\_new](https://get.docker.com/open_in_new) 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.

### 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](#). 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 \(rootless mode\)](#).

## Install pre-releases

Docker also provides a convenience script at [https://test.docker.com/open\\_in\\_new](https://test.docker.com/open_in_new) 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 apt-get purge docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin docker-ce-rootless-extras
```

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.

---

Revision #1

Created 25 August 2023 21:45:45

Updated 19 January 2024 18:19:58