

Install Docker Engine on Ubuntu

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 Kinetic 22.10
- Ubuntu Jammy 22.04 (LTS)
- Ubuntu Focal 20.04 (LTS)
- Ubuntu Bionic 18.04 (LTS)

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

Uninstall old versions

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

```
$ 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.

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, refer to 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 (`/desktop/install/linux-install/`). This is the easiest and quickest way to get started.
- You can also set up and install Docker Engine from Docker's `apt` repository.
- Install it manually and manage upgrades manually.
- Using a convenience scripts. Only recommended for testing and development environments.

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:

```
$ sudo mkdir -p /etc/apt/keyrings
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /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/
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Install Docker Engine

1. Update the `apt` package index:

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

i Receiving a GPG error when running `apt-get update` ?

Your default `umask` (<https://en.wikipedia.org/wiki/Umask>), may be incorrectly configured, preventing detection of the repository public key file. Try granting read permission for the Docker public key file before updating the package index:

```
$ sudo chmod a+r /etc/apt/keyrings/docker.gpg
$ sudo apt-get update
```

2. 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-compose-plugin
```



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. 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 post-install (/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 `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/> (<https://download.docker.com/linux/ubuntu/dists/>).
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-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-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 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. 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 post-install (/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 <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 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.
- 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/> (<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 \(/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user\)](/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 \(rootless mode\) \(/engine/security/rootless/\)](/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 apt-get purge docker-ce docker-ce-cli containerd.io 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 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.