Docker Engine
Overview
Install
Installation per distro
Install on CentOS
Install on Debian
Install on Fedora
Install on Ubuntu
Install binaries
Optional post-installation steps
Release notes
Previous versions
Deprecated features
Docker App
Docker Buildx
Docker Context
Docker Scan
Docker Compose
Docker Desktop
Docker Hub
Open-source projects
Release notes
Superseded products and tools

Install Docker Engine on Ubuntu

Estimated reading time: 10 minutes

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 Groovy 20.10
- Ubuntu Focal 20.04 (LTS)
- Ubuntu Bionic 18.04 (LTS)
- Ubuntu Xenial 16.04 (LTS)

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

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

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 \
    apt-transport-https \
    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/share.
```

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.

> **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.

x86_64 / amd64   armhf   arm64

Edit this page
Request docs changes

On this page:

```
$ echo \
  "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://downloa
  $(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  xenial
      docker-ce | 5:18.09.0~3-0~ubuntu-xenial | https://download.docker.com/linux/ubuntu  xenial
      docker-ce | 18.06.1~ce~3-0~ubuntu       | https://download.docker.com/linux/ubuntu  xenial
      docker-ce | 18.06.0~ce~3-0~ubuntu       | https://download.docker.com/linux/ubuntu  xenial
    ```

    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.i
    ```

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 an informational 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 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/`, choose your Ubuntu version, then browse to `pool/stable/`, choose `amd64`, `armhf`, or `arm64`, 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.

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 an informational 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 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 convenience scripts at get.docker.com and test.docker.com for installing edge and testing versions of Docker Engine - Community into development environments quickly and non-interactively. The source code for the scripts is in the `docker-install` repository. **Using these scripts is not recommended for production environments**, and you should understand the potential risks before you use them:

- The scripts require `root` or `sudo` privileges to run. Therefore, you should carefully examine and audit the scripts before running them.
- The scripts attempt to detect your Linux distribution and version and configure your package management system for you. In addition, the scripts do not allow you to customize any installation parameters. This may lead to an unsupported configuration, either from Docker's point of view or from your own organization's guidelines and standards.
- The scripts install all dependencies and recommendations of the package manager without asking for confirmation. This may install a large number of packages, depending on the current configuration of your host machine.
- The script does not provide options to specify which version of Docker to install, and installs the latest version that is released in the "edge" channel.
- Do not use the convenience script if Docker has already been installed on the host machine using another mechanism.

This example uses the script at get.docker.com to install the latest release of Docker Engine - Community on Linux. To install the latest testing version, use test.docker.com instead. In each of the commands below, replace each occurrence of `get` with `test`.

> **Warning:**
>
> Always examine scripts downloaded from the internet before running them locally.

```
$ curl -fsSL https://get.docker.com -o get-docker.sh
$ sudo sh get-docker.sh
<...>
```

If you would like to use Docker as a non-root user, you should now consider adding your user to the "docker" group with something like:

```
$ sudo usermod -aG docker <your-user>
```

Remember to log out and back in for this to take effect!

> **Warning:**
>
> Adding a user to the "docker" group grants them the ability to run containers which can be used to obtain root privileges on the Docker host. Refer to Docker Daemon Attack Surface for more information.

Docker Engine - Community is installed. It starts automatically on `DEB`-based distributions. On `RPM`-based distributions, 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.

> **Note:**
>
> To install Docker without root privileges, see Run the Docker daemon as a non-root user (Rootless mode).

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.
- Review the topics in Develop with Docker to learn how to build new applications using Docker.

requirements, apt, installation, ubuntu, install, uninstall, upgrade, update

Rate this page:

👍 858 👎 243