Install Docker on Ubuntu Linux

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 Mantic 23.10
- 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 need to uninstall any conflicting packages.

Distro maintainers provide 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-compose-v2`
- `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 docker-compose-v2 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.

1. Set up Docker's `apt` repository.

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

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

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

2. Install the Docker packages.

To install the latest version, run:

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

> **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 step 2 of 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/`.
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-install`

`root` `sudo`

> **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
> ```

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

`docker`

`RPM`

`systemctl` `service`

> **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).

get.docker.com

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

Revision #1
Created 2024-04-02 04:03:19 UTC by Miles Menninga
Updated 2024-04-02 04:04:00 UTC by Miles Menninga