Install Docker Engine on Debian

- Prerequisites
  - Firewall limitations
  - OS requirements
  - Uninstall old versions
- Installation methods
  - Install using the apt repository {#install-using-the-repository}
    - Upgrade Docker Engine
  - Install from a package
    - Upgrade Docker Engine
- Uninstall Docker Engine
- Next steps

To get started with Docker Engine on Debian, make sure you meet the prerequisites, and then follow the installation steps.

Prerequisites

Firewall limitations

> Before you install Docker, make sure you consider the following security implications and firewall incompatibilities.

- 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.

- Docker is only compatible with `iptables-nft` and `iptables-legacy`. Firewall rules created with `nft` are not supported on a system with Docker installed. Make sure that any firewall rulesets you use are created with `iptables` or `ip6tables`, and that you add them to the `DOCKER-USER` chain, see Packet filtering and firewalls.

OS requirements

To install Docker Engine, you need one of these Debian versions:

- Debian Trixie 13 (stable)
- Debian Bookworm 12 (oldstable)

Docker Engine for Debian is compatible with x86_64 (or amd64), armhf (arm/v7), arm64, and ppc64le (ppc64el) architectures.

Uninstall old versions

Before you can install Docker Engine, you need to uninstall any conflicting packages.

Your Linux distribution may provide unofficial Docker packages, which may conflict with the official packages provided by Docker. You must uninstall these packages before you install the official version of Docker Engine.

The unofficial packages to uninstall are:

- `docker.io`
- `docker-compose`
- `docker-doc`
- `docker-buildx`
- `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:

```
$ sudo apt remove $(dpkg --get-selections docker.io docker-compose docker-doc docker-buildx
podman-docker containerd runc | cut -f1)
```

`apt` 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.

{{% include "engine-license.md" %}}

Install using the `apt` repository {#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 `apt` 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 update
sudo apt install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL {{% param "download-url-base" %}}/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
sudo tee /etc/apt/sources.list.d/docker.sources <<EOF
Types: deb
URIs: {{% param "download-url-base" %}}
Suites: $(. /etc/os-release && echo "$VERSION_CODENAME")
Components: stable
```

```
Architectures: $(dpkg --print-architecture)
Signed-By: /etc/apt/keyrings/docker.asc
EOF

sudo apt update
```

> If you use Debian testing or a derivative distribution such as Kali Linux, you may need to substitute the part of this command that's expected to print the version codename:
>
> `$(. /etc/os-release && echo "$VERSION_CODENAME")`
>
> Replace this part with the codename of the corresponding Debian release, such as `trixie`.

2. Install the Docker packages.
{{< tabs >}} {{< tab name="Latest" >}}
To install the latest version, run:

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

{{< /tab >}} {{< tab name="Specific version" >}}
To install a specific version of Docker Engine, start by listing the available versions in the repository:

```
$ apt list --all-versions docker-ce

docker-ce/bookworm 5:{{% param "docker_ce_version" %}}-1~debian.12~bookworm <arch>
docker-ce/bookworm 5:{{% param "docker_ce_version_prev" %}}-1~debian.12~bookworm
<arch>
...
```

Select the desired version and install:

```
$ VERSION_STRING=5:{{% param "docker_ce_version" %}}-1~debian.12~bookworm
$ sudo apt install docker-ce=$VERSION_STRING docker-ce-cli=$VERSION_STRING
containerd.io docker-buildx-plugin docker-compose-plugin
```

{{< /tab >}} {{< /tabs >}}

> After installation, verify that Docker is running:
>
> `$ sudo systemctl status docker`
>
> If Docker is not running, start it manually:

```
$ sudo systemctl start docker
```

3. Verify that the 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.

{{% include "root-errors.md" %}}

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 [`{{% param "download-url-base" %}}/dists/`]({{% param "download-url-base" %}}/dists/).
2. Select your Debian 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
```

> After installation, verify that Docker is running:
>
> ```
> $ sudo systemctl status docker
> ```
>
> If Docker is not running, start it manually:
>
> ```
> $ sudo systemctl start docker
> ```

6. Verify that the 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.

{{% include "root-errors.md" %}}

Upgrade Docker Engine

To upgrade Docker Engine, download the newer package files and repeat the installation procedure, pointing to the new files.

{{% include "install-script.md" %}}

Uninstall Docker Engine

1. Uninstall the Docker Engine, CLI, containerd, and Docker Compose packages:

   ```
   $ sudo apt 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
   ```

3. Remove source list and keyrings

```
$ sudo rm /etc/apt/sources.list.d/docker.sources
$ sudo rm /etc/apt/keyrings/docker.asc
```

You have to delete any edited configuration files manually.

Next steps

- Continue to Post-installation steps for Linux.

□□□ #2
□ □□□ □□□ 2026-10-31 19:23:33 UTC
□ □□□ □□□ 2026-10-31 19:23:34 UTC