

# Installation

## Docker Compose

Updated: ?? Docker Compose ????? Docker ???? ?

????

```
sudo apt-get install docker-compose-plugin
```

????

```
docker compose version
```

The latest download: <https://docs.docker.com/compose/install/>

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose

# If you can't run the command, creating a sybolic link as follows.
sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose

docker-compose --version
```

## ARM64 Linux

```
# Download: https://github.com/docker/compose/releases
sudo curl -L --fail https://github.com/docker/compose/releases/download/v2.2.3/docker-compose-linux-aarch64 -o /usr/local/bin/docker-compose
sudo chmod 0755 /usr/local/bin/docker-compose
```

## Docker

Latest: <https://docs.docker.com/engine/install/>

# Fedora 41

```
sudo dnf -y install dnf-plugins-core
sudo dnf-3 config-manager --add-repo https://download.docker.com/linux/fedora/docker-ce.repo

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

# Ubuntu 20.04/22.04

```
sudo apt-get update
sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-
archive-keyring.gpg

echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

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

# Debian 9/10

```
sudo apt-get update

sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl gnupg \
    lsb-release

# Add Docker's GPG key
```

```

curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o
/etc/apt/trusted.gpg.d/docker.gpg

# Set up the repository
echo \
"deb [arch=$(dpkg --print-architecture)] https://download.docker.com/linux/debian \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Install Docker Engine
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

# Verify that Docker is installed correctly.
sudo docker version
sudo docker run hello-world

```

## RedHat 8

“ RHEL 8 ?? IBM Z(s390x) ?????? Docker????? x86\_64 ????? CentOS ??????  
 ????? docker-ce ????????????????

???? (?????????? RHN)

```
dnf update
```

???? docker-ce ?????

?? /etc/yum.repos.d/docker-ce.repo

```

[docker-ce-stable]
name=Docker CE Stable - $basearch
baseurl=https://download.docker.com/linux/centos/$releasever/$basearch/stable
enabled=1
gpgcheck=1
gpgkey=https://download.docker.com/linux/centos/gpg

```

?????

```
dnf config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

```
dnf update
```

```
[root@mydocker ~]# docker version
Client: Docker Engine - Community
Version:      20.10.14
API version:  1.41
Go version:   go1.16.15
Git commit:   a224086
Built:        Thu Mar 24 01:47:44 2022
OS/Arch:      linux/amd64
Context:      default
Experimental:  true

Server: Docker Engine - Community
Engine:
Version:      20.10.14
```

```
API version:    1.41 (minimum version 1.12)
Go version:     go1.16.15
Git commit:     87a90dc
Built:          Thu Mar 24 01:46:10 2022
OS/Arch:        linux/amd64
Experimental:   false
containerd:
  Version:      1.5.11
  GitCommit:    3df54a852345ae127d1fa3092b95168e4a88e2f8
runc:
  Version:      1.0.3
  GitCommit:    v1.0.3-0-gf46b6ba
docker-init:
  Version:      0.19.0
  GitCommit:    de40ad0
```

[Optional] non-root ????

```
usermod -aG docker $USER
```

## Raspberry Pi OS

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
sudo usermod -aG docker Pi
docker version
docker info
docker run hello-world
```

## Rootless Mode

### Ubuntu 22.04

Stop system-wide Docker daemon

```
sudo systemctl disable --now docker.service docker.socket
```

Install

```
sudo apt-get install docker-ce-rootless-extras uidmap
dockerd-rootless-setuptool.sh install
docker info
```

## Optional: Configure the systemd

```
systemctl --user start docker
systemctl --user enable docker
sudo loginctl enable-linger $(whoami)
```

- The socket path is set to `$XDG_RUNTIME_DIR/docker.sock` by default. `$XDG_RUNTIME_DIR` is typically set to `/run/user/$UID`.
- The data dir is set to `~/.local/share/docker` by default. The data dir should not be on NFS.
- The daemon config dir is set to `~/.config/docker` by default. This directory is different from `~/.docker` that is used by the client.

## Optional: Client

```
# Optional #1: with socket path
export DOCKER_HOST=unix://$XDG_RUNTIME_DIR/docker.sock
docker run -d -p 8080:80 nginx
```

```
# Optional #2: with CLI context
docker context use rootless
docker run -d -p 8080:80 nginx
```