## Step-by-Step Tutorial: Installing Docker and Docker Compose on Ubuntu

Tomer Klein

·

Follow

4 min read
·
Oct 22, 2023

--

1

Listen

Share

## Introduction

Welcome to our straightforward guide on installing Docker and Docker Compose on Ubuntu. Docker is a powerful tool for containerization, and Docker Compose simplifies multi-container application management. In this step-by-step tutorial, we will walk you through the installation process, making it easy for both beginners and experienced users to get started with containerization on your Ubuntu system. Let’s dive in!

## Step 1 — Prepare the system for installation

Before we can start installing docker & docker-compose we need to make sure that the system is up to date. You can do it by running the following commands:

```
sudo apt update
sudo apt upgrade -y
```

## Step 2 — Download and Install the docker repository

Docker is using an installation repository. to install the repository and use it we need to install the following packages:

```
sudo apt install -y ca-certificates curl gnupg lsb-release
```

After the package installation, we need to add Docker’s GPG Key to our system by running the following commands:

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

Now we can install the docker repository by running the following commands:

```
sudo echo  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/u
sudo apt update
```

## Step 3 — Installing Docker

You can proceed to install Docker by executing the provided command below. This straightforward command will enable you to initiate the Docker installation process on your system.

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

## Step 4 — Verify the installation

Once the installation is complete, you’ll have the opportunity to employ the following command to launch a “Hello World” Docker container. This will serve as a practical test to verify the successful installation of Docker on your system.

```
sudo docker run hello-world
```

If all components were configured correctly and the setup proceeded without issues, you should observe the following output:

## Step 5 — Run Docker as a non-root user

In order to execute Docker commands, root privileges are required, which is why you must prefix your commands with "sudo." However, if you prefer to run Docker as a user without root privileges, you'll need to establish a Docker group. This group will allow specified users to interact with Docker without the need for elevated permissions.

Create a group named "docker" that you can assign users to with the following command:

```
sudo groupadd docker
```

By utilizing a straightforward command, you can easily include users in the Docker group you've established. This action grants them the ability to execute Docker commands without requiring root privileges:

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

It's important to keep in mind that "$USER" is merely a placeholder and should be substituted with the actual username of the user you intend to add to the Docker group. In order for these changes to become effective, you'll need to log out of your session and then log back in. Following this procedure, you'll be able to utilize Docker without the necessity of utilizing "sudo" for each command.

## Step 6 — Install docker-compose

Docker Compose is a tool that enables you to define and manage multi-container Docker applications through a simple, declarative configuration file. With Docker Compose, you can specify the services, networks, and volumes required for your application in a single YAML file. It allows you to describe the components of your application and their relationships, making it easier to set up and manage complex applications with multiple containers.

Some of the key benefits of Docker Compose include:

1. **Simplified Orchestration**: Docker Compose simplifies the process of orchestrating containers, allowing you to define how they interact, start, and stop together.
2. **Easy Replication**: You can replicate your development environment in a consistent and repeatable manner by using the same Docker Compose configuration across different environments.
3. **Efficient Collaboration**: Teams can collaborate more effectively as they share a common configuration file that defines the entire application stack.
4. **Portability**: Docker Compose configurations are portable, which means you can use them on different systems and platforms as long as Docker and Docker Compose are installed.
5. **Scalability**: While Docker Compose is primarily used for development and testing, you can also use it for smaller production deployments, but for larger-scale production setups, Docker Swarm or Kubernetes are more commonly employed.

Docker Compose simplifies the process of defining, running, and managing multi-container applications, making it a valuable tool for both developers and system administrators.

To install docker-compose tun the following commands:

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

To verify that your Docker Compose installation was successful, you can use the following command:

```
docker-compose --version
```

Running this command will display the version of Docker Compose that is currently installed on your system. If you see the version information, it means the installation was successful, and Docker Compose is ready for use.