

# Install Docker GCP VM



how to install docker and docker-compose in a gcp vm



To install Docker and Docker Compose on a Google Cloud Platform (GCP) virtual machine (VM), follow these steps:

## 1. Connect to Your GCP VM

- Open the **Google Cloud Console**.
- Go to **Compute Engine** and select your VM.
- Click on the **SSH** button to connect to the VM.

## 2. Install Docker

### 1. Update the package list:

```
bash

sudo apt-get update
```

### 2. Install dependencies:

```
bash

sudo apt-get install apt-transport-https ca-certificates curl software-properties-commo
n
```

### 3. Add Docker's official GPG key:

```
bash

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

### 4. Add the Docker repository:

```
bash

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(ls
b_release -cs) stable"
```

## 5. Update the package list again:

```
bash

sudo apt-get update
```

## 6. Install Docker CE (Community Edition):

```
bash

sudo apt-get install docker-ce
```

## 7. Check if Docker is installed correctly:

```
bash

sudo docker --version
```

## 8. Start and enable Docker to run on boot:

```
bash

sudo systemctl start docker
sudo systemctl enable docker
```

## 9. (Optional) Add your user to the `docker` group to run Docker commands without `sudo`:

```
bash

sudo usermod -aG docker $USER
```

After running the above command, log out and log back in or run `newgrp docker` to apply the group changes.

# 3. Install Docker Compose

## 1. Download the latest version of Docker Compose:

```
bash

sudo curl -L "https://github.com/docker/compose/releases/download/$(curl -s https://api.github.com/repos/docker/compose/releases/latest | jq -r .tag_name)/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

## 2. Apply executable permissions to the Docker Compose binary:

```
bash

sudo chmod +x /usr/local/bin/docker-compose
```

### 3. Verify the Docker Compose installation:

```
bash

docker-compose --version
```

## 4. Test Docker and Docker Compose

- To verify Docker is working, run:

```
bash

sudo docker run hello-world
```

- To verify Docker Compose is working, create a simple `docker-compose.yml` file and run:

```
bash

docker-compose up
```

That's it! Docker and Docker Compose should now be installed and running on your GCP VM.



```
sudo apt-get update
Get:1 file:/etc/apt/mirrors/debian.list Mirrorlist [30 B]
Get:2 file:/etc/apt/mirrors/debian-security.list Mirrorlist [39 B]
Hit:7 https://packages.cloud.google.com/apt google-compute-engine-bookworm-stable InRelease
Hit:3 https://deb.debian.org/debian bookworm InRelease
Hit:8 https://packages.cloud.google.com/apt cloud-sdk-bookworm InRelease
Hit:4 https://deb.debian.org/debian bookworm-updates InRelease
Hit:5 https://deb.debian.org/debian bookworm-backports InRelease
Hit:6 https://deb.debian.org/debian-security bookworm-security InRelease
Ign:9 https://download.docker.com/linux/ubuntu bookworm InRelease
Err:10 https://download.docker.com/linux/ubuntu bookworm Release
      404 Not Found [IP: 13.226.22.125 443]
Hit:11 http://apt.postgresql.org/pub/repos/apt bookworm-pgdg InRelease
Reading package lists... Done
E: The repository 'https://download.docker.com/linux/ubuntu bookworm Release' does not have a
Release file.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
```

W: http://apt.postgresql.org/pub/repos/apt/dists/bookworm-pgdg/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.



The error occurs because you are trying to use the Docker repository for **Ubuntu**, but your system is running **Debian Bookworm**, which has a different repository structure. Docker doesn't provide official packages for every Debian release immediately.

Here's how to fix the issue:

## 1. Remove the incorrect Docker repository

Remove the Ubuntu-specific Docker repository added earlier:

```
bash

sudo add-apt-repository --remove "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
```

## 2. Add the correct Docker repository for Debian

Instead of the Ubuntu repository, use the repository for **Debian**:

1. Add Docker's official GPG key:

```
bash

curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/sh
are/keyrings/docker-archive-keyring.gpg
```

2. Add the Docker repository:

```
bash

echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] http
s://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee /etc/apt/sou
rces.list.d/docker.list > /dev/null
```

## 3. Update and install Docker

1. Update the package list:

```
bash

sudo apt-get update
```

## 2. Install Docker:

```
bash

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

## 3. Verify Docker installation:

```
bash

sudo docker --version
```

## 4. (Optional) Handle the PostgreSQL Warning

The warning about the PostgreSQL key being in a legacy trusted keyring can be resolved by migrating to the new GPG key storage method:

### 1. Download and store the key in a modern location:

```
bash

curl https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo gpg --dearmor -o /usr/sh
are/keyrings/postgresql.gpg
```

### 2. Update the PostgreSQL repository to use the new key:

```
bash

echo "deb [signed-by=/usr/share/keyrings/postgresql.gpg] http://apt.postgresql.org/pub/
repos/apt/ $(lsb_release -cs)-pgdg main" | sudo tee /etc/apt/sources.list.d/pgdg.list >
/dev/null
```

### 3. Update the package list:

```
bash

sudo apt-get update
```

After these steps, both Docker and PostgreSQL repositories should work seamlessly on your Debian Bookworm system.



```
sudo add-apt-repository --remove "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
sudo: add-apt-repository: command not found
```



The `add-apt-repository` command is not available because it's part of the `software-properties-common` package, which may not be installed by default. You can either install it or manually remove the incorrect repository entry. Here's how:

## Option 1: Install `add-apt-repository` and Remove the Repository

1. Install the required package:

```
bash

sudo apt-get update
sudo apt-get install software-properties-common
```

2. Run the removal command again:

```
bash

sudo add-apt-repository --remove "deb [arch=amd64] https://download.docker.com/linux/ub
untu $(lsb_release -cs) stable"
```

## Option 2: Manually Edit the Repository List

1. Open the Docker repository file in an editor (e.g., `nano`):

```
bash

sudo nano /etc/apt/sources.list.d/docker.list
```

2. Remove or comment out the line referring to `https://download.docker.com/linux/ubuntu`.
3. Save and exit the file (`Ctrl+O`, `Enter`, `Ctrl+X` in Nano).

## Continue with Correct Setup

Once the incorrect repository is removed, proceed with adding the correct Docker repository for Debian as described earlier.



# lets restart the process of installing docker and docker-compose, this is what i get when i run sudo apt-get update:

```
sudo apt-get update
Get:1 file:/etc/apt/mirrors/debian.list Mirrorlist [30 B]
Get:2 file:/etc/apt/mirrors/debian-security.list Mirrorlist [39 B]
Hit:7 https://packages.cloud.google.com/apt google-compute-engine-bookworm-stable InRelease
Hit:8 https://packages.cloud.google.com/apt cloud-sdk-bookworm InRelease
Hit:3 https://deb.debian.org/debian bookworm InRelease
Hit:4 https://deb.debian.org/debian bookworm-updates InRelease
Hit:5 https://deb.debian.org/debian bookworm-backports InRelease
Hit:6 https://deb.debian.org/debian-security bookworm-security InRelease
Hit:9 http://apt.postgresql.org/pub/repos/apt bookworm-pgdg InRelease
Reading package lists... Done
W: http://apt.postgresql.org/pub/repos/apt/dists/bookworm-pgdg/InRelease: Key is stored in legacy
trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
```



From your output, the Docker repository issue has been resolved, but there is a **warning** regarding the PostgreSQL repository key being stored in the legacy `trusted.gpg` keyring. This does not block updates or installations but indicates a move to the new GPG keyring format for security and compatibility.

Let's proceed with installing Docker and Docker Compose on your **Debian Bookworm VM**:

## Step 1: Install Docker

### 1. Install prerequisites:

```
bash

sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-co
mmon gnupg
```

### 2. Add Docker's official GPG key and repository:

```
bash

curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/sh
are/keyrings/docker-archive-keyring.gpg
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] http
s://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee /etc/apt/sou
rces.list.d/docker.list > /dev/null
```

### 3. Update the package list and install Docker:

```
bash
```

```
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin dock
er-compose-plugin
```

#### 4. Verify Docker installation:

```
bash
```

```
sudo docker --version
```

## Step 2: Install Docker Compose (Standalone Version)

If you need the standalone Docker Compose binary, follow these steps:

### 1. Download the latest version of Docker Compose:

```
bash
```

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

### 2. Make it executable:

```
bash
```

```
sudo chmod +x /usr/local/bin/docker-compose
```

### 3. Verify Docker Compose installation:

```
bash
```

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

## Step 3: Handle PostgreSQL Warning

To remove the warning about the legacy trusted keyring, update PostgreSQL's GPG key management:

### 1. Add the new PostgreSQL GPG key:

```
bash
```

```
curl https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo gpg --dearmor -o /usr/sh
```



```
are/keyrings/postgresql.gpg
```

## 2. Update the PostgreSQL repository to use the new key:

```
bash
```

```
echo "deb [signed-by=/usr/share/keyrings/postgresql.gpg] http://apt.postgresql.org/pub/
repos/apt/ $(lsb_release -cs)-pgdg main" | sudo tee /etc/apt/sources.list.d/pgdg.list >
/dev/null
```

## 3. Update the package list:

```
bash
```

```
sudo apt-get update
```

This resolves the warning about the legacy GPG key.

## Step 4: Test Docker and Docker Compose

- Run a test container to check Docker:

```
bash
```

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

- Test Docker Compose by running a simple configuration:

1. Create a file called `docker-compose.yml`:

```
yaml
```

```
version: '3'
services:
  hello:
    image: hello-world
```

2. Run:

```
bash
```

```
docker-compose up
```

You should now have Docker and Docker Compose installed and fully operational on your GCP VM!



how to add docker to group to avoid using sudo



To allow your user to run Docker commands without `sudo`, you need to add the user to the `docker` group. Here's how to do it:

---

## Step 1: Create the `docker` Group (if not already created)

Run the following command to ensure the `docker` group exists:

```
bash

sudo groupadd docker
```

If the group already exists, you will see a message saying so.

---

## Step 2: Add Your User to the `docker` Group

Replace `<username>` with your actual username, or use `$USER` for the current user:

```
bash

sudo usermod -aG docker $USER
```

## Step 3: Apply the Group Changes

For the group changes to take effect, you need to restart your session. You can either:

1. **Log out and log back in, or**
2. **Run this command to apply the group change without logging out:**

```
bash

newgrp docker
```

## Step 4: Verify the Setup

Run a Docker command without `sudo` to check if the permissions are applied correctly:

```
bash

docker run hello-world
```

If the container runs without errors, you've successfully configured Docker to work without `sudo`.