

To change the user name in Linux:

- 1)check your user name >whoami
- 2)Create a temporary user >sudo adduser tempadmin
- 3)Add the new user to sudoers >sudo usermod -aG sudo tempadmin
- 4)Logout from the current user via GUI and login to tempadmin
- 5)In the terminal of tempadmin, >sudo usermod -l
1rv24mc051_karthikb
- 6)Now type the below command in original user after logging again via GUI: >whoami
- 7)Remove the temp user, delete it >sudo deluser tempadmin >sudo
rm -r /home/tempadmin

```
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~$ whoami
1rv24mc051_karthikb
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~$ sudo deluser tempadmin
[sudo] password for 1rv24mc051_karthikb:
info: Removing crontab ...
info: Removing user 'tempadmin' ...
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~$ sudo rm -r /home/tempadmin
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~$ whoami
1rv24mc051_karthikb
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~$ █
```

TO INSTALL DOCKER USING APT REPO

- 1)Set up Docker's apt repository.
- 2)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
- 3)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

The screenshot shows the Docker Docs website in a web browser. The browser's address bar displays the URL: `docs.docker.com/engine/install/ubuntu/#install-using-the-repository`. The page title is "Install Docker Engine on Ubuntu". The sidebar on the left contains a list of navigation links: "OPEN SOURCE", "Docker Engine", "Install", "Ubuntu" (selected), "Debian", "RHEL", "Fedora", "Raspberry Pi OS (32-bit / armhf)", "CentOS", "SLES (s390x)", "Binaries", "Post-installation steps", "Storage", "Networking", "Containers", "CLI", and "Daemon". The main content area features a heading "Install Docker Engine on Ubuntu" and a subheading "Prerequisites". A warning box states: "Before you install Docker, make sure you consider the following security implications and firewall incompatibilities." The table of contents on the right lists: "Prerequisites", "Firewall limitations", "OS requirements", "Uninstall old versions", "Installation methods", "Install using the apt repository", "Upgrade Docker Engine", "Install from a package", "Upgrade Docker Engine", "Install using the convenience script", "Install pre-releases", "Upgrade Docker after using the conveni...", and "Uninstall Docker Engine". The bottom of the page has a cookie consent banner with buttons for "Cookies Settings", "Reject All", and "Accept All Cookies".

```
11v24mc051_karthik@karthik-Lenovo-G50-80:~$ # Add Docker's official GPG key:
1rv24mc051_karthik@karthik-Lenovo-G50-80:~$ sudo apt-get update
1rv24mc051_karthik@karthik-Lenovo-G50-80:~$ sudo apt-get install ca-certificates curl
1rv24mc051_karthik@karthik-Lenovo-G50-80:~$ sudo install -m 0755 -d /etc/apt/keyrings
1rv24mc051_karthik@karthik-Lenovo-G50-80:~$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
1rv24mc051_karthik@karthik-Lenovo-G50-80:~$ 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
  $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
Hit:1 http://repo.mysql.com/apt/ubuntu noble InRelease
Get:2 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Get:3 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:4 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,209 B]
Hit:5 http://in.archive.ubuntu.com/ubuntu noble InRelease
Get:6 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:8 https://mirrors.ubuntu.com/mirrors.txt noble InRelease [51.8 kB]
Fetched 1,145 kB in 1s (1,145 kB/s)
Reading package lists... Done
```

```
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin dock
er-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras libslirp0 pigz slirp4netns
Suggested packages:
  cgroupfs-mount | cgroup-lite docker-model-plugin
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libslirp0 pigz
  slirp4netns
0 upgraded, 9 newly installed, 0 to remove and 182 not upgraded.
Need to get 105 MB of archives.
After this operation, 436 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://download.docker.com/linux/ubuntu noble/stable amd64 containerd.io amd64 1.7.28-1-ubuntu.24.04-noble [31.9 MB]
Get:2 http://in.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu noble/universe amd64 slirp4netns amd64 1.2.1-1build2 [34.9 kB]
Get:5 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-cli amd64 5:28.5.1-1-ubuntu.24.04-noble [16.5 MB]
Get:6 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce amd64 5:28.5.1-1-ubuntu.24.04-noble [19.7 MB]
Get:7 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-buildx-plugin amd64 0.29.1-1-ubuntu.24.04-noble [15.9 MB]
Get:8 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-ce-rootless-extras amd64 5:28.5.1-1-ubuntu.24.04-noble [
```

5)Check docker status sudo systemctl status docker

```
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-10-17 09:53:29 IST; 41s ago
   TriggeredBy: ● docker.service
   Docs: https://docs.docker.com
   Main PID: 15040 (dockerd)
   Tasks: 10
   Memory: 21.5M (peak: 22.7M)
   CPU: 591ms
   CGroup: /system.slice/docker.service
           └─15040 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Oct 17 09:53:28 karthik-Lenovo-G50-80 dockerd[15040]: time="2025-10-17T09:53:28.206543682+05:30" level=info msg="detected 127.0.0.1"
Oct 17 09:53:28 karthik-Lenovo-G50-80 dockerd[15040]: time="2025-10-17T09:53:28.256710742+05:30" level=info msg="Creating a container"
Oct 17 09:53:28 karthik-Lenovo-G50-80 dockerd[15040]: time="2025-10-17T09:53:28.467254260+05:30" level=info msg="Loading container"
Oct 17 09:53:29 karthik-Lenovo-G50-80 dockerd[15040]: time="2025-10-17T09:53:29.342073983+05:30" level=info msg="Loading container"
Oct 17 09:53:29 karthik-Lenovo-G50-80 dockerd[15040]: time="2025-10-17T09:53:29.440694532+05:30" level=info msg="Docker daemon"
Oct 17 09:53:29 karthik-Lenovo-G50-80 dockerd[15040]: time="2025-10-17T09:53:29.440870481+05:30" level=info msg="Initializing buildkit"
Oct 17 09:53:29 karthik-Lenovo-G50-80 dockerd[15040]: time="2025-10-17T09:53:29.522669889+05:30" level=info msg="Completed buildkit"
Oct 17 09:53:29 karthik-Lenovo-G50-80 dockerd[15040]: time="2025-10-17T09:53:29.544123956+05:30" level=info msg="Daemon has completed initialization"
Oct 17 09:53:29 karthik-Lenovo-G50-80 dockerd[15040]: time="2025-10-17T09:53:29.544123956+05:30" level=info msg="Daemon is ready"
```

6)Run a sample container sudo docker run hello-world

```
1rv24mc051_karthikb@karthik-Lenovo-G50-80:~$ sudo docker run hello-world
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

<https://hub.docker.com/>

For more examples and ideas, visit:

I **L** **E** **T** **A** **R** **E** **S** **E** **N** **C** **O** **N** **D** **E**