To change the user name in Linux:
1.check your user name
>whoami

```
abhi@abhi-GF63-Thin-9SCSR:~$ whoami
abhi
```

2. Create a temporary user
>sudo adduser tempuser

```
abhi@abhi-GF63-Thin-9SCSR:~$ sudo adduser tempuser
[sudo] password for abhi:
info: Adding user `tempuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `tempuser' (1001) ...
info: Adding new user `tempuser' (1001) with group `tempuser (1001)' ...
warn: The home directory `/home/tempuser' already exists.  Not touching this directory.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
Changing the user information for tempuser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
info: Adding new user `tempuser' to supplemental / extra groups `users' ...
```

3. Add the new user to sudoers
> sudo usermod -aG sudo tempuser

4. Logout from the current user via GUI and login to tempuser

5. In the terminal of tempuser
>sudo usermod -l 1RV24MC018_ananyah abhi

6. Now type the below command in original user after logging again via GUI:
>whoami

```
1RV24MC018_ananyah@abhi-GF63-Thin-9SCSR:~$ whoami
1RV24MC018_ananyah
```

7. Remove the temp user,delete it
>sudo deluser tempuser

```
1RV24MC018_ananyah@abhi-GF63-Thin-9SCSR:~$ sudo deluser tempuser
[sudo] password for 1RV24MC018_ananyah:
info: Removing crontab ...
info: Removing user `tempuser' ...
```

>sudo rm -r /home/tempuser
```
1RV24MC018_ananyah@abhi-GF63-Thin-9SCSR:~$ sudo rm -r /home/tempuser
1RV24MC018_ananyah@abhi-GF63-Thin-9SCSR:~$ █
```

Docker Installation

Step 1: Set up Docker's apt repository.

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

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 \

  $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \

  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

Step 2: Install the Docker packages.

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

Step 3: Verify that the installation is successful by running the hello-world image:

```
E: Release file for http://archive.ubuntu.com/ubuntu/dists/noble-updates/InRelease is not valid yet (invalid for another 9h 1min 39s). Updates for this repository will not be applied.
E: Release file for http://archive.ubuntu.com/ubuntu/dists/noble-backports/InRelease is not valid yet (invalid for another 6h 49min 26s). Updates for this repository will not be applie
d.
1RV24MC018_ananyah@abhi-GF63-Thin-9SCSR:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-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 85 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://archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble/main amd64 libslirp0 amd64 4.7.0-1ubuntu3 [63.8 kB]
Get:4 http://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 [6,481 kB]
Get:9 https://download.docker.com/linux/ubuntu noble/stable amd64 docker-compose-plugin amd64 2.40.0-1-ubuntu.24.04~noble [14.2 MB]
Fetched 105 MB in 1min 31s (1,147 kB/s)
Selecting previously unselected package containerd.io.
(Reading database ... 251780 files and directories currently installed.)
Preparing to unpack .../0-containerd.io_1.7.28-1-ubuntu.24.04~noble_amd64.deb ...
Unpacking containerd.io (1.7.28-1-ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../1-docker-ce-cli_5%3a28.5.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce-cli (5:28.5.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../2-docker-ce_5%3a28.5.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-ce (5:28.5.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package pigz.
Preparing to unpack .../3-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../4-docker-buildx-plugin_0.29.1-1~ubuntu.24.04~noble_amd64.deb ...
Unpacking docker-buildx-plugin (0.29.1-1~ubuntu.24.04~noble) ...
Selecting previously unselected package docker-ce-rootless-extras.
```

```
        CPU: 762ms
     CGroup: /system.slice/docker.service
             └─8793 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Oct 17 00:08:47 abhi-GF63-Thin-9SCSR dockerd[8793]: time="2025-10-17T00:08:47.204371158+05:30" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, so using reso
Oct 17 00:08:47 abhi-GF63-Thin-9SCSR dockerd[8793]: time="2025-10-17T00:08:47.230464802+05:30" level=info msg="Creating a containerd client" address=/run/containerd/containerd.sock ti
Oct 17 00:08:47 abhi-GF63-Thin-9SCSR dockerd[8793]: time="2025-10-17T00:08:47.270035482+05:30" level=info msg="Loading containers: start."
Oct 17 00:08:47 abhi-GF63-Thin-9SCSR dockerd[8793]: time="2025-10-17T00:08:47.867844000+05:30" level=info msg="Loading containers: done."
Oct 17 00:08:47 abhi-GF63-Thin-9SCSR dockerd[8793]: time="2025-10-17T00:08:47.886875362+05:30" level=info msg="Docker daemon" commit=f8215cc containerd-snapshotter=false storage-drive
Oct 17 00:08:47 abhi-GF63-Thin-9SCSR dockerd[8793]: time="2025-10-17T00:08:47.887051177+05:30" level=info msg="Initializing buildkit"
Oct 17 00:08:47 abhi-GF63-Thin-9SCSR dockerd[8793]: time="2025-10-17T00:08:47.913353796+05:30" level=info msg="Completed buildkit initialization"
Oct 17 00:08:47 abhi-GF63-Thin-9SCSR dockerd[8793]: time="2025-10-17T00:08:47.923812693+05:30" level=info msg="Daemon has completed initialization"
Oct 17 00:08:47 abhi-GF63-Thin-9SCSR dockerd[8793]: time="2025-10-17T00:08:47.924006438+05:30" level=info msg="API listen on /run/docker.sock"
Oct 17 00:08:47 abhi-GF63-Thin-9SCSR systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-22/22 (END)
1RV24MC018_ananyah@abhi-GF63-Thin-9SCSR:~$ sudo systemctl start docker
1RV24MC018_ananyah@abhi-GF63-Thin-9SCSR:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:6dc565aa630927052111f823c303948cf83670a3903ffa3849f1488ab517f891
Status: Downloaded newer image for hello-world:latest

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:
 https://docs.docker.com/get-started/

1RV24MC018_ananyah@abhi-GF63-Thin-9SCSR:~$
```