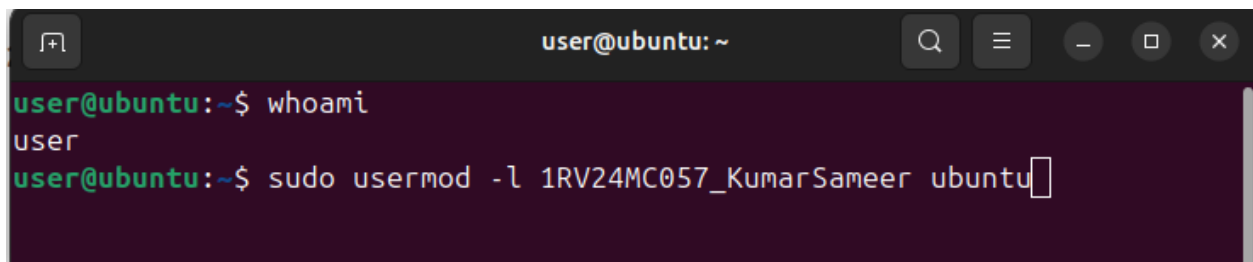# Changing username in Linux-

1. Check the present username using "whoami" command in the terminal.
2. Use the command "sudo usermod -l 1RV24MC057KumarSameer ubuntu" to change the username.
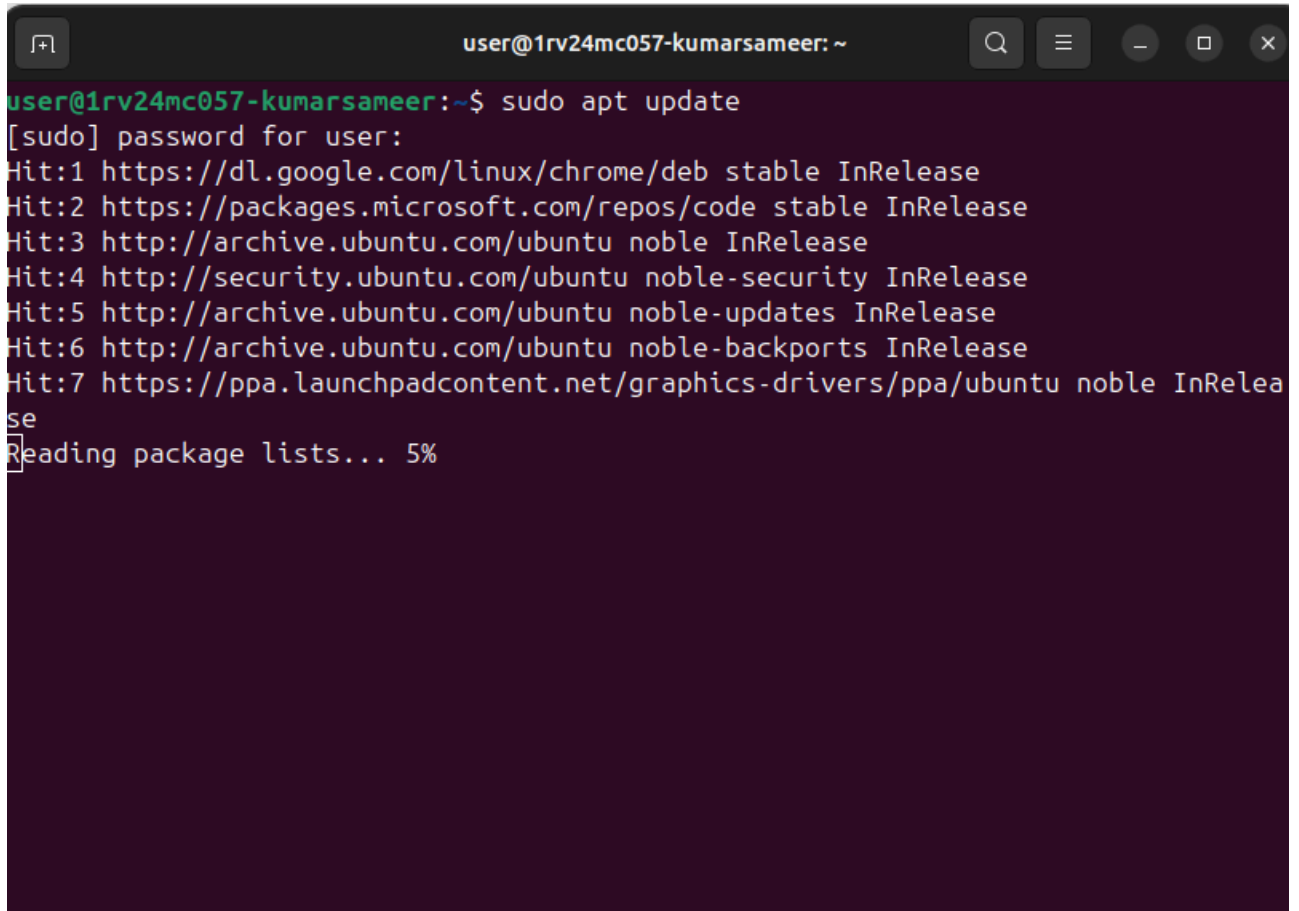
   The format is "sudo usermod -l NewUsername OldUsername" to change the username."

```
user@ubuntu: ~

user@ubuntu:~$ whoami
user
user@ubuntu:~$ sudo usermod -l 1RV24MC057_KumarSameer ubuntu
```

Installing Docker-

1. **Update the existing packages using "sudo apt update"command**

Install prerequisite packages using "sudo apt install apt-transport-https ca-certificates curl software-properties-common"

```
user@1rv24mc057-kumarsameer:~$ sudo apt install apt-transport-https ca-certifica
tes curl software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apt-transport-https is already the newest version (2.8.3).
ca-certificates is already the newest version (20240203).
curl is already the newest version (8.5.0-2ubuntu10.6).
software-properties-common is already the newest version (0.99.49.3).
The following packages were automatically installed and are no longer required:
  cpu-checker docker-ce-rootless-extras ipxe-qemu
  ipxe-qemu-256k-compat-efi-roms libaio1t64 libboost-iostreams1.83.0
  libboost-thread1.83.0 libcacard0 libdaxctl1 libfdt1 libiscsi7
  libjack-jackd2-0 libndctl6 libpmem1 libpmemobj1 librados2 librbd1
  librdmacm1t64 libsdl2-2.0-0 libslirp0 libspice-server1 libsubid4 liburing2
  libusbredirparser1t64 libvirglrenderer1 msr-tools ovmf pass pigz
  qemu-block-extra qemu-system-common qemu-system-data qemu-system-gui
  qemu-system-modules-opengl qemu-system-modules-spice qemu-system-x86
  qemu-utils qrencode seabios slirp4netns tree uidmap wl-clipboard xclip
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
user@1rv24mc057-kumarsameer:~$
```

Add Docker's official GPG key using "curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg"

```
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
user@1rv24mc057-kumarsameer:~$ curl -fsSL https://download.docker.com/linux/ubun
tu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
user@1rv24mc057-kumarsameer:~$
```
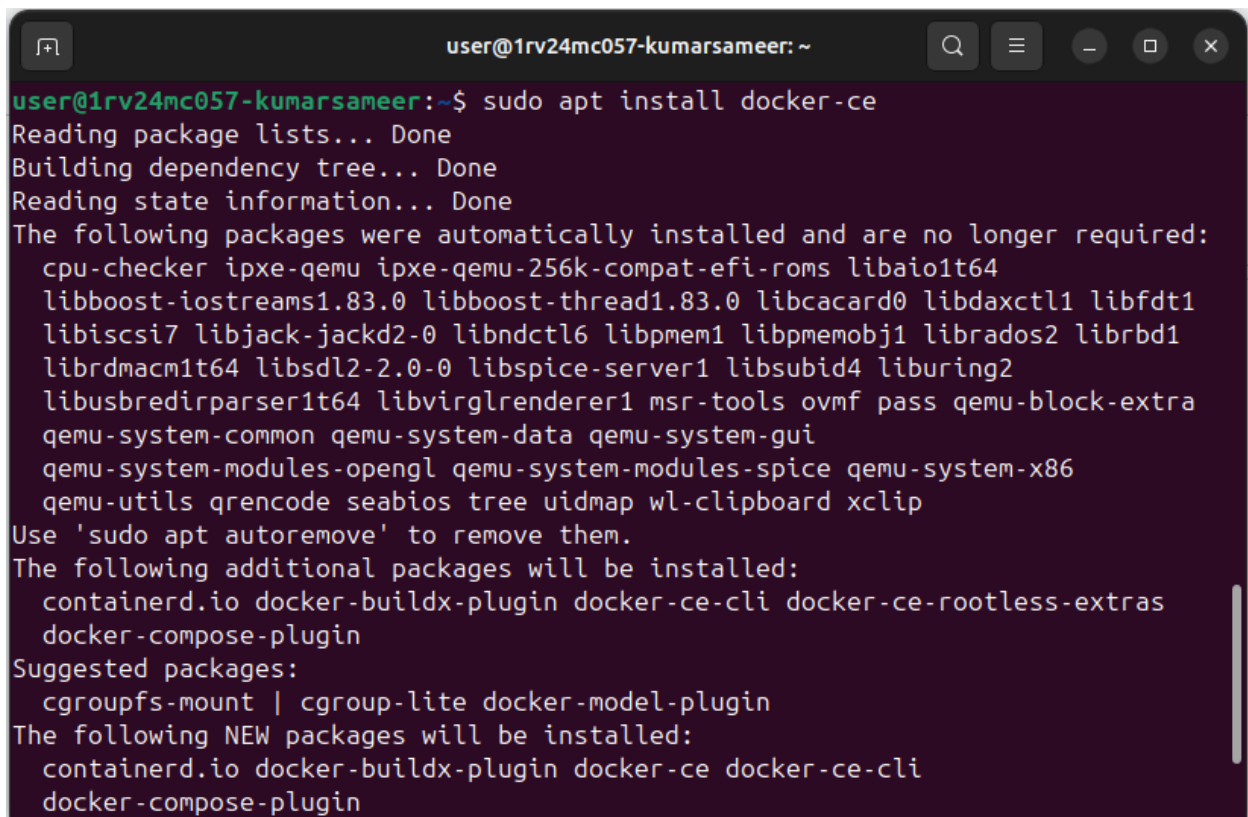
Add the Docker repository to APT sources using "echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null"

```
user@1rv24mc057-kumarsameer:~$ echo "deb [arch=$(dpkg --print-architecture) sign
ed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.co
m/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/doc
ker.list > /dev/null
user@1rv24mc057-kumarsameer:~$ []
```

Update the package database with Docker packages from the newly added repo using "sudo apt update"
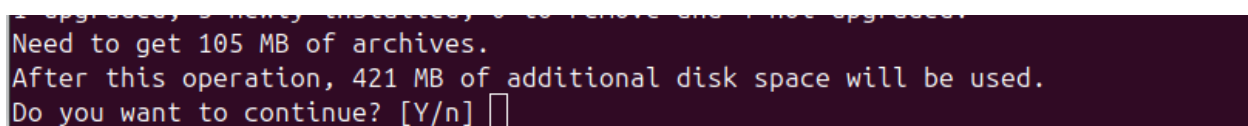
```
user@1rv24mc057-kumarsameer:~$ sudo apt update
Get:1 https://download.docker.com/linux/ubuntu noble InRelease [48.5 kB]
Hit:2 https://packages.microsoft.com/repos/code stable InRelease
Hit:3 https://dl.google.com/linux/chrome/deb stable InRelease
Get:4 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [33.0
 kB]
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:6 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:7 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:8 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:9 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu noble InRelea
se
Fetched 81.5 kB in 4s (20.1 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
5 packages can be upgraded. Run 'apt list --upgradable' to see them.
user@1rv24mc057-kumarsameer:~$ []
```

Install Docker using "sudo apt install docker-ce"


```
user@1rv24mc057-kumarsameer:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  cpu-checker ipxe-qemu ipxe-qemu-256k-compat-efi-roms libaio1t64
  libboost-iostreams1.83.0 libboost-thread1.83.0 libcacard0 libdaxctl1 libfdt1
  libiscsi7 libjack-jackd2-0 libndctl6 libpmem1 libpmemobj1 librados2 librbd1
  librdmacm1t64 libsdl2-2.0-0 libspice-server1 libsubid4 liburing2
  libusbredirparser1t64 libvirglrenderer1 msr-tools ovmf pass qemu-block-extra
  qemu-system-common qemu-system-data qemu-system-gui
  qemu-system-modules-opengl qemu-system-modules-spice qemu-system-x86
  qemu-utils qrencode seabios tree uidmap wl-clipboard xclip
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras
  docker-compose-plugin
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-compose-plugin
```

You'll be prompted for more archives


```
Need to get 105 MB of archives.
After this operation, 421 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```
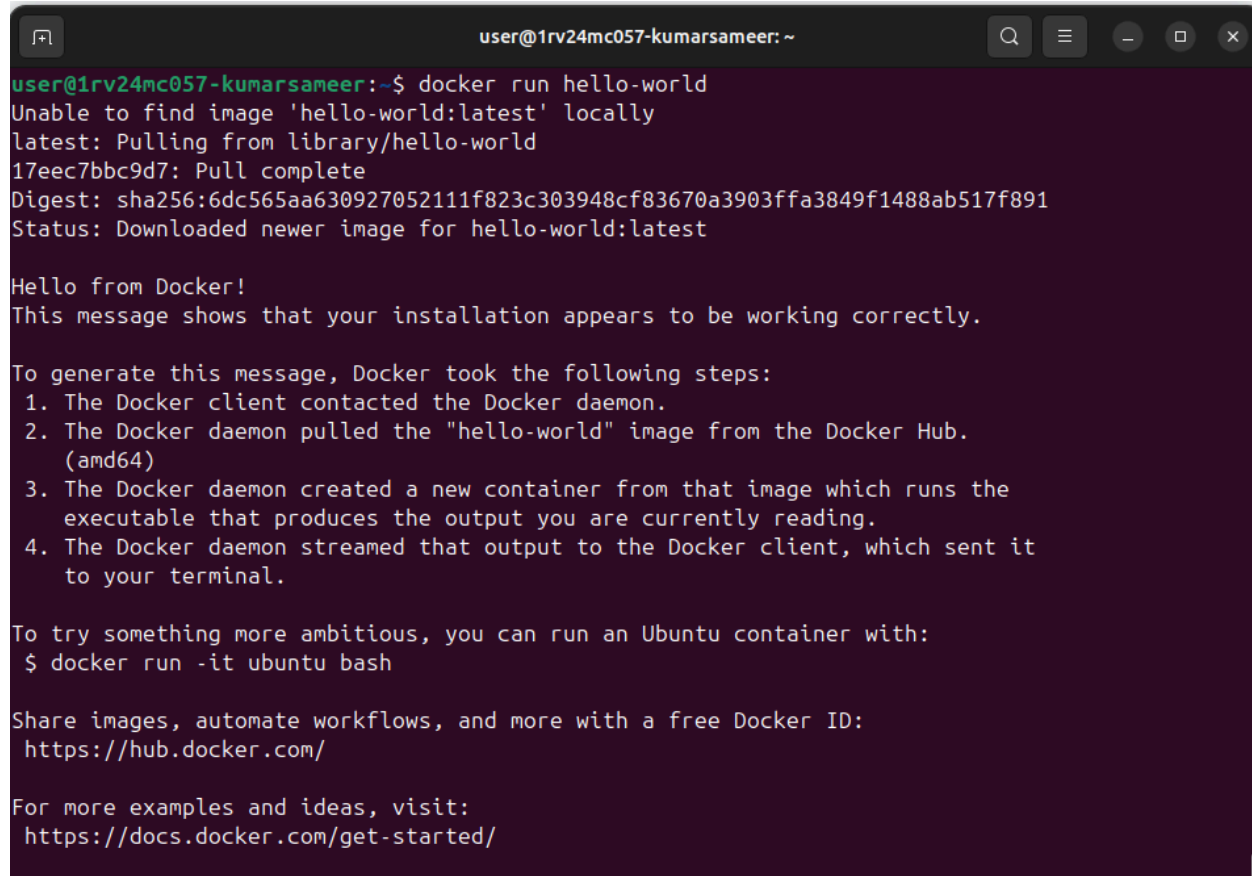
Press Y

Wait for a few minutes to get the installation complete depending on the internet speed

Verify Docker is installed and running using "sudo systemctl status docker"

Pull the image hello-world from the docker repository using "docker run hello-world"

If the image is present in the local system it will fetch that otherwise it will pull from the library and download it

```
user@1rv24mc057-kumarsameer:~$ 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/
```