Ref link: https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-22-04

Step 1 - Installing Docker

sudo apt update

sudo apt install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

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

sudo apt update

apt-cache policy docker-ce (will get outpu for Docker repo instead of the default Ubuntu repo)

sudo apt install docker-ce

sudo systemctl status docker

Step 2 — Executing the Docker Command Without Sudo (Optional)

sudo usermod -aG docker ${USER} (Adding all users to group)

sudo usermod -aG docker username (Adding 1 user at a time)

su - ${USER} (To apply the membership)

Step 3 — Using the Docker Command

docker [option] [command] [arguments]

docker

Output

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:

run Create and run a new container from an image

exec Execute a command in a running container

ps List containers

build Build an image from a Dockerfile

pull Download an image from a registry

push Upload an image to a registry

images List images

login Authenticate to a registry

logout Log out from a registry

search Search Docker Hub for images

version Show the Docker version information

info Display system-wide information

Management Commands:

builder Manage builds

buildx\* Docker Buildx

compose\* Docker Compose

container Manage containers

context Manage contexts

image Manage images

manifest Manage Docker image manifests and manifest lists

network Manage networks

plugin Manage plugins

system Manage Docker

trust Manage trust on Docker images

volume Manage volumes

Swarm Commands:

swarm Manage Swarm

Commands:

attach Attach local standard input, output, and error streams to a running container

commit Create a new image from a container's changes

cp Copy files/folders between a container and the local filesystem

create Create a new container

diff Inspect changes to files or directories on a container's filesystem

events Get real time events from the server

export Export a container's filesystem as a tar archive

history Show the history of an image

import Import the contents from a tarball to create a filesystem image

inspect Return low-level information on Docker objects

kill Kill one or more running containers

load Load an image from a tar archive or STDIN

logs Fetch the logs of a container

pause Pause all processes within one or more containers

port List port mappings or a specific mapping for the container

rename Rename a container

restart Restart one or more containers

rm Remove one or more containers

rmi Remove one or more images

save Save one or more images to a tar archive (streamed to STDOUT by default)

start Start one or more stopped containers

stats Display a live stream of container(s) resource usage statistics

stop Stop one or more running containers

tag Create a tag TARGET\_IMAGE that refers to SOURCE\_IMAGE

top Display the running processes of a container

unpause Unpause all processes within one or more containers

update Update configuration of one or more containers

wait Block until one or more containers stop, then print their exit codes

Global Options:

--config string Location of client config files (default "/home/neosoft/.docker")

-c, --context string Name of the context to use to connect to the daemon (overrides DOCKER\_HOST env var and default context set with "docker context use")

-D, --debug Enable debug mode

-H, --host list Daemon socket to connect to

-l, --log-level string Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")

--tls Use TLS; implied by –tlsverify

--tlscacert string Trust certs signed only by this CA (default "/home/neosoft/.docker/ca.pem")

--tlscert string Path to TLS certificate file (default "/home/neosoft/.docker/cert.pem")

--tlskey string Path to TLS key file (default "/home/neosoft/.docker/key.pem")

--tlsverify Use TLS and verify the remote

-v, --version Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For more help on how to use Docker, head to https://docs.docker.com/go/guides/