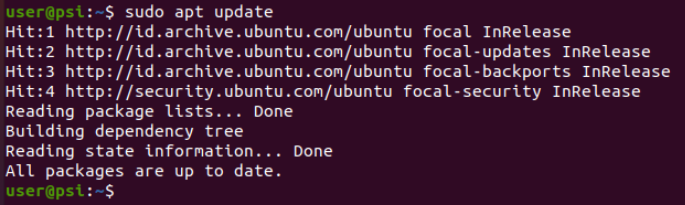
**Step 1 — Installing Docker**

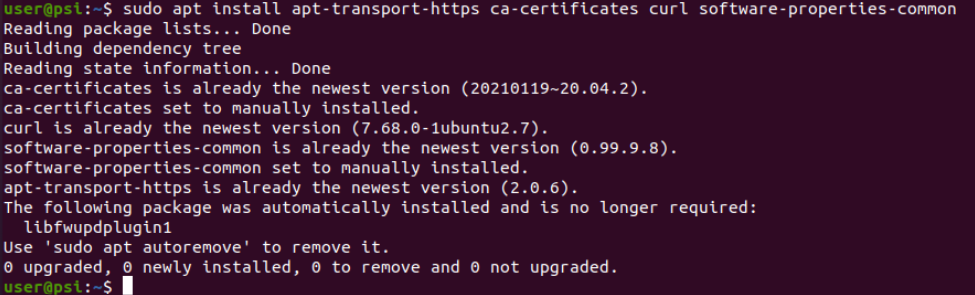
First, update your current package list:

$ sudo apt update



Next, install some prerequisite packages that allow apt to use packages over HTTPS:

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



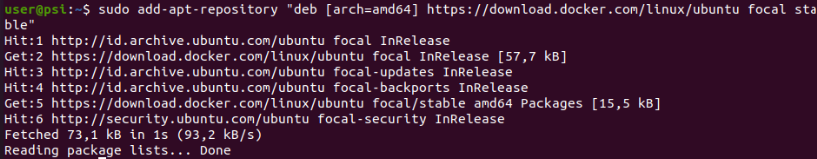
Then add the GPG key for the official Docker repository to your system:

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



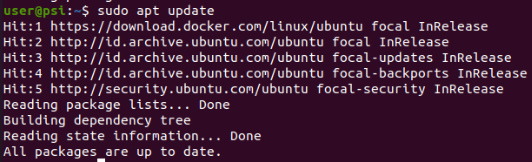
Add the Docker repository to the APT source:

$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"



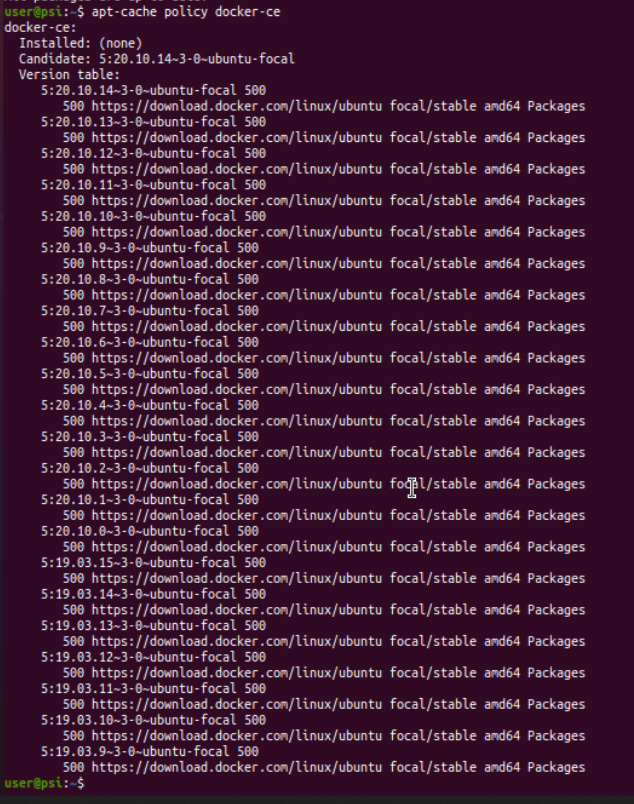
Next, update the packages database with the Docker packages from the newly added repo:

$ sudo apt update



Make sure you are going to install from the Docker repo instead of the default Ubuntu repo:

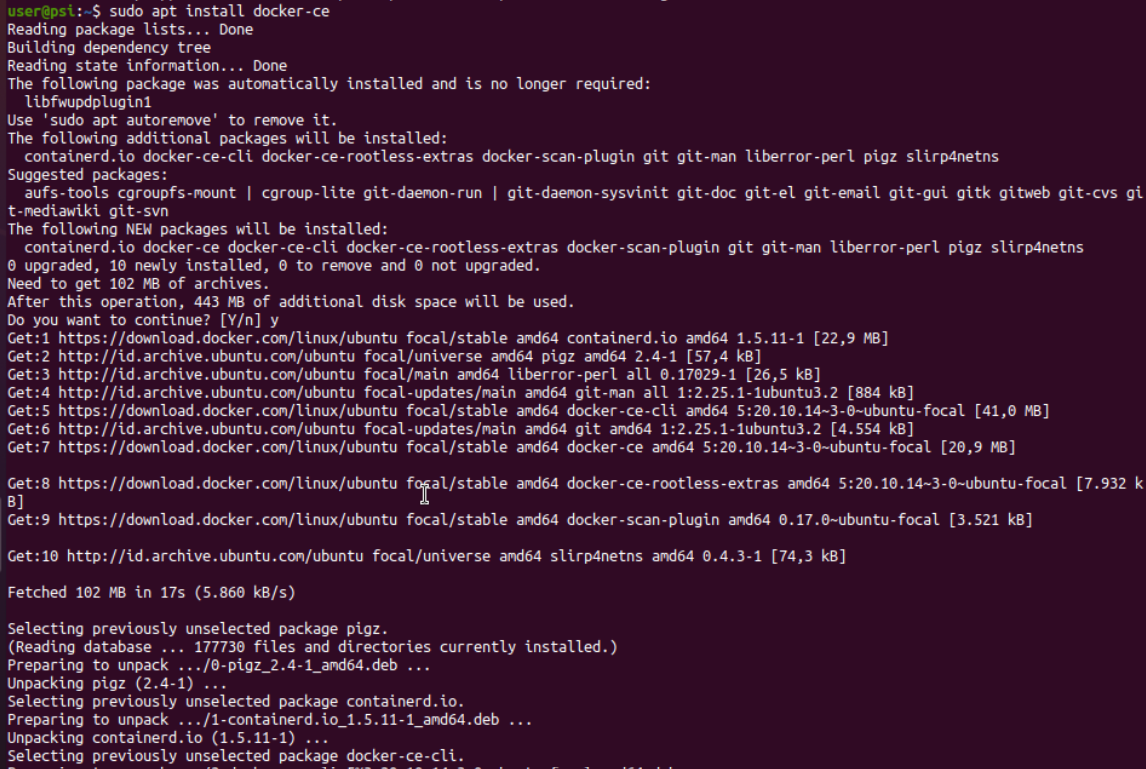
$ apt-cache policy docker-ce

You should see output like this, although the version numbers for Docker may be different: 

Note that docker-ce is not installed yet, but a candidate for installation is from the Docker repository for Ubuntu 20.04 (focal).

Finally, install Docker:

$ sudo apt install docker-ce



.

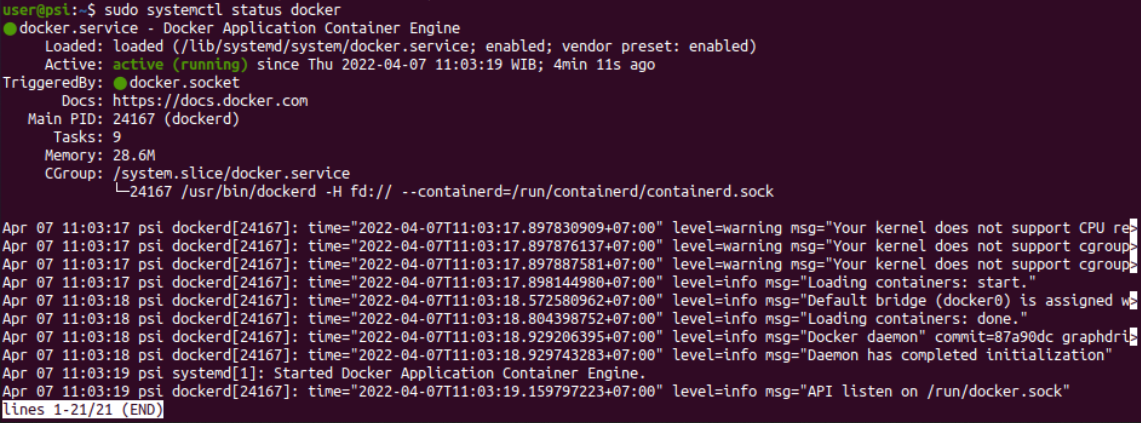
.

.

Docker should now be installed, the daemon started, and the process should now be able to run on startup at boot. Check that it's running:

$ sudo systemctl status docker

The output should be similar to the following, indicating that the service is up and running:



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

By default, the docker command can only be run the **root** user or by a user in the **docker** group, which is automatically created during Docker’s installation process. If you attempt to run the docker command without prefixing it with sudo or without being in the **docker** group, you’ll get an output like this:

Output

docker: Cannot connect to the Docker daemon. Is the docker daemon running on this host?.

See 'docker run --help'.

If you want to avoid typing sudo whenever you run the docker command, add your username to the docker group:

$ sudo usermod -aG docker ${USER} && newgrp docker

To apply the new group membership, log out of the server and back in, or type the following:

$ su - ${USER}

You will be prompted to enter your user’s password to continue.

Confirm that your user is now added to the **docker** group by typing:

$ groups

If you need to add a user to the docker group that you’re not logged in as, declare that username explicitly using:

$ sudo usermod -aG docker username

The rest of this article assumes you are running the docker command as a user in the **docker** group. If you choose not to, please prepend the commands with sudo.

**Using the Docker Command**

Using docker consists of passing it a chain of options and commands followed by arguments. The syntax takes this form:

$ docker [option] [command] [arguments]

To view all available subcommands, type:

$ docker

As of Docker 19, the complete list of available subcommands includes:

Output

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

build Build an image from a Dockerfile

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

exec Run a command in a running container

export Export a container's filesystem as a tar archive

history Show the history of an image

images List images

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

info Display system-wide information

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

login Log in to a Docker registry

logout Log out from a Docker registry

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

ps List containers

pull Pull an image or a repository from a registry

push Push an image or a repository to a registry

rename Rename a container

restart Restart one or more containers

rm Remove one or more containers

rmi Remove one or more images

run Run a command in a new container

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

search Search the Docker Hub for images

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

version Show the Docker version information

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

To view the options available to a specific command, type:

$ docker docker-subcommand --help

To view system-wide information about Docker, use:

$ docker info