

Step 1 — Installing Docker

First, update your current package list:

```
$ sudo apt update
```

```
user@psi:~$ sudo apt update
Hit:1 http://id.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://id.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
user@psi:~$
```

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
```

```
user@psi:~$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20210119~20.04.2).
ca-certificates set to manually installed.
curl is already the newest version (7.68.0-1ubuntu2.7).
software-properties-common is already the newest version (0.99.9.8).
software-properties-common set to manually installed.
apt-transport-https is already the newest version (2.0.6).
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
user@psi:~$
```

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 -
```

```
user@psi:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
```

Add the Docker repository to the APT source:

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

```
user@psi:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"
Hit:1 http://id.archive.ubuntu.com/ubuntu focal InRelease
Get:2 https://download.docker.com/linux/ubuntu focal InRelease [57,7 kB]
Hit:3 http://id.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:5 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [15,5 kB]
Hit:6 http://security.ubuntu.com/ubuntu focal-security InRelease
Fetched 73,1 kB in 1s (93,2 kB/s)
Reading package lists... Done
```

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

```
$ sudo apt update
```

```

user@psi:~$ sudo apt update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 http://id.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://id.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.

```

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:

```

user@psi:~$ apt-cache policy docker-ce
docker-ce:
  Installed: (none)
  Candidate: 5:20.10.14-3-0~ubuntu-focal
  Version table:
   5:20.10.14-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.13-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.12-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.11-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.10-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.9-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.8-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.7-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.6-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.5-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.4-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.3-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.2-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.1-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:20.10.0-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:19.03.15-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:19.03.14-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:19.03.13-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:19.03.12-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:19.03.11-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:19.03.10-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:19.03.9-3-0~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
user@psi:~$

```

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
```



```

user@psi:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  containerd.io docker-ce-cli docker-ce-rootless-extras docker-scan-plugin git git-man liberror-perl pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs gi
t-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-scan-plugin git git-man liberror-perl pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 102 MB of archives.
After this operation, 443 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://download.docker.com/linux/ubuntu focal/stable amd64 containerd.io amd64 1.5.11-1 [22,9 MB]
Get:2 http://id.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57,4 kB]
Get:3 http://id.archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.17029-1 [26,5 kB]
Get:4 http://id.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all 1:2.25.1-1ubuntu3.2 [884 kB]
Get:5 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-cli amd64 5:20.10.14-3-0-ubuntu-focal [41,0 MB]
Get:6 http://id.archive.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-1ubuntu3.2 [4.554 kB]
Get:7 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce amd64 5:20.10.14-3-0-ubuntu-focal [20,9 MB]
Get:8 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-rootless-extras amd64 5:20.10.14-3-0-ubuntu-focal [7.932 k
B]
Get:9 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-scan-plugin amd64 0.17.0-ubuntu-focal [3.521 kB]
Get:10 http://id.archive.ubuntu.com/ubuntu focal/universe amd64 slirp4netns amd64 0.4.3-1 [74,3 kB]
Fetched 102 MB in 17s (5.860 kB/s)

Selecting previously unselected package pigz.
(Reading database ... 177730 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../1-containerd.io_1.5.11-1_amd64.deb ...
Unpacking containerd.io (1.5.11-1) ...
Selecting previously unselected package docker-ce-cli.

```

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:

```

user@psi:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-04-07 11:03:19 WIB; 4min 11s ago
 TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
  Main PID: 24167 (dockerd)
    Tasks: 9
   Memory: 28.6M
   CGroup: /system.slice/docker.service
           └─24167 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Apr 07 11:03:17 psi dockerd[24167]: time="2022-04-07T11:03:17.897830909+07:00" level=warning msg="Your kernel does not support CPU re
Apr 07 11:03:17 psi dockerd[24167]: time="2022-04-07T11:03:17.897876137+07:00" level=warning msg="Your kernel does not support cgroup
Apr 07 11:03:17 psi dockerd[24167]: time="2022-04-07T11:03:17.897887581+07:00" level=warning msg="Your kernel does not support cgroup
Apr 07 11:03:17 psi dockerd[24167]: time="2022-04-07T11:03:17.898144980+07:00" level=info msg="Loading containers: start."
Apr 07 11:03:18 psi dockerd[24167]: time="2022-04-07T11:03:18.572580962+07:00" level=info msg="Default bridge (docker0) is assigned with IP address 172.17.0.1/24"
Apr 07 11:03:18 psi dockerd[24167]: time="2022-04-07T11:03:18.804398752+07:00" level=info msg="Loading containers: done."
Apr 07 11:03:18 psi dockerd[24167]: time="2022-04-07T11:03:18.929206395+07:00" level=info msg="Docker daemon" commit=87a90dc graphdriver=overlay2
Apr 07 11:03:18 psi dockerd[24167]: time="2022-04-07T11:03:18.929743283+07:00" level=info msg="Daemon has completed initialization"
Apr 07 11:03:19 psi systemd[1]: Started Docker Application Container Engine.
Apr 07 11:03:19 psi dockerd[24167]: time="2022-04-07T11:03:19.159797223+07:00" level=info msg="API listen on /run/docker.sock"
lines 1-21/21 (END)

```

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}
```

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
```