

How to Install and Use Docker on Ubuntu 18.04

First, update your existing list of packages:

- `sudo apt update`

Next, install a few prerequisite packages which let apt 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 APT sources:

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

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

- `sudo apt update`

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

- `apt-cache policy docker-ce`

You'll see output like this, although the version number for Docker may be different:

Output of `apt-cache policy docker-ce`

`docker-ce:`

`Installed: (none)`

`Candidate: 18.03.1~ce~3-0~ubuntu`

How to Install and Use Docker on Ubuntu 18.04

Version table:

```
18.03.1~ce~3-0~ubuntu 500
```

```
500 https://download.docker.com/linux/ubuntu bionic/stable amd64
```

Packages

Notice that `docker-ce` is not installed, but the candidate for installation is from the Docker repository for Ubuntu 18.04 (bionic).

Finally, install Docker:

- `sudo apt install docker-ce`

Docker should now be installed, the daemon started, and the process enabled to start on boot. Check that it's running:

- `sudo systemctl status docker`

The output should be similar to the following, showing that the service is active and running:

Output

- `docker.service - Docker Application Container Engine`

```
Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor  
preset: enabled)
```

```
Active: active (running) since Thu 2018-07-05 15:08:39 UTC; 2min 55s  
ago
```

```
Docs: https://docs.docker.com
```

```
Main PID: 10096 (dockerd)
```

```
Tasks: 16
```

```
CGroup: /system.slice/docker.service
```

```
└─10096 /usr/bin/dockerd -H fd://
```

```
└─10113 docker-containerd --config
```

```
/var/run/docker/containerd/containerd.toml
```

How to Install and Use Docker on Ubuntu 18.04

Installing Docker now gives you not just the Docker service (daemon) but also the docker command line utility, or the Docker client. We'll explore how to use the docker command later in this tutorial.

```
## List Docker CLI commands
```

```
docker
```

```
docker container --help
```

```
## Display Docker version and info
```

```
docker --version
```

```
docker version
```

```
docker info
```

```
## Execute Docker image
```

```
docker run hello-world
```

```
## List Docker images
```

```
docker image ls
```

```
## List Docker containers (running, all, all in quiet mode)
```

```
docker container ls
```

```
docker container ls --all
```

```
docker container ls -aq
```

Command docker

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:

- config string Location of client config files (default "/home/sk/.docker")
- D, --debug Enable debug mode
- H, --host list Daemon socket(s) 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

How to Install and Use Docker on Ubuntu 18.04

```
"/home/sk/.docker/ca.pem")
--tlscert string Path to TLS certificate file (default
"/home/sk/.docker/cert.pem")
--tlskey string Path to TLS key file (default
"/home/sk/.docker/key.pem")
--tlsverify      Use TLS and verify the remote
-v, --version    Print version information and quit
```

Management Commands:

config	Manage Docker configs
container	Manage containers
image	Manage images
network	Manage networks
node	Manage Swarm nodes
plugin	Manage plugins
secret	Manage Docker secrets
service	Manage services
stack	Manage Docker stacks
swarm	Manage Swarm
system	Manage Docker
trust	Manage trust on Docker images
volume	Manage volumes

Commands:

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

How to Install and Use Docker on Ubuntu 18.04

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

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

```
docker build -t friendlyhello . # Create image using this directory's
Dockerfile
docker run -p 4000:80 friendlyhello # Run "friendlyname" mapping port 4000 to
80
docker run -d -p 4000:80 friendlyhello # Same thing, but in detached
mode
docker container ls # List all running
containers
docker container ls -a # List all containers, even those not
running
docker container stop <hash> # Gracefully stop the specified
container
docker container kill <hash> # Force shutdown of the specified
container
```

How to Install and Use Docker on Ubuntu 18.04

```
docker container rm <hash>          # Remove specified container from this
machine
docker container rm $(docker container ls -a -q)    # Remove all
containers
docker image ls -a                   # List all images on this
machine
docker image rm <image id>           # Remove specified image from this
machine
docker image rm $(docker image ls -a -q)           # Remove all images from this
machine
docker login                         # Log in this CLI session using your Docker
credentials
docker tag <image> username/repository:tag         # Tag <image> for upload to
registry
docker push username/repository:tag                # Upload tagged image to
registry
docker run username/repository:tag                 # Run image from a
registry
```