

CHEAPER EACH MONTH - LIMITED DEAL [LEARN MORE](#)

Home > Blog

How to Install Docker on Ubuntu 20.04? | Step-by-step Tutorial

by Rimvydas Misevičius

October 13th, 2021

DOCKER



We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

[Accept Cookies](#)

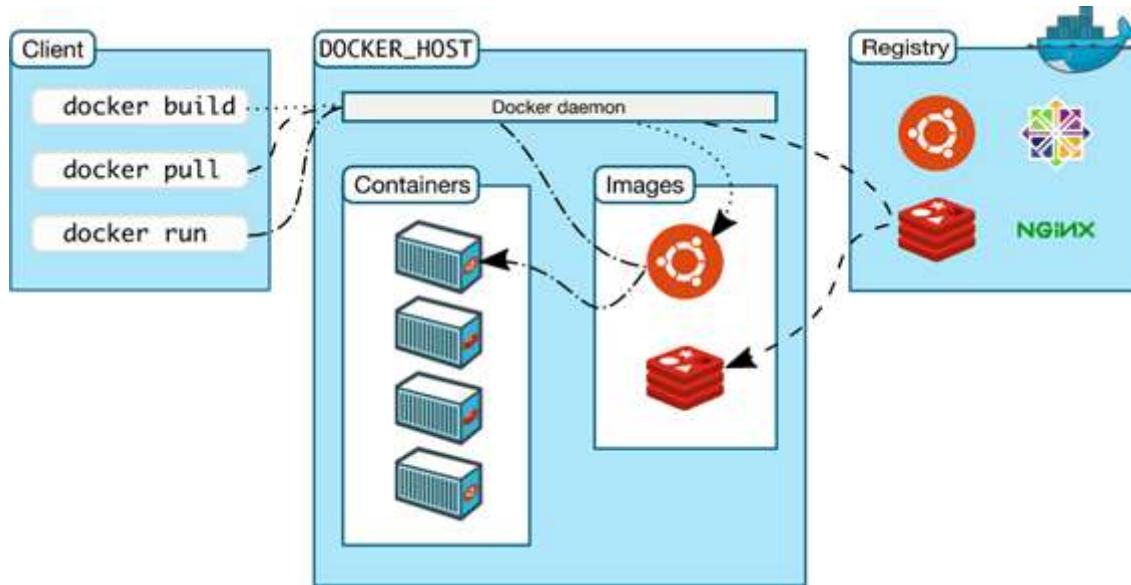
Docker is an open-source platform that allows you to package and run an application in a loosely isolated environment called containers. This security and isolation enable you to execute many containers simultaneously on a given host.

What is Docker container?

Containers are self-sufficient software units with their own application code, configuration, and dependencies. They provide you with a stable running environment that is easily portable to different hosts.

Docker uses a client-server architecture. The client talks to the Docker daemon, which does most of the work of building, running, and distributing containers.

The client and daemon may run on the same host, or you can connect a Docker client to a remote daemon. Client and daemon communication may be accomplished using REST API, UNIX sockets or a network interface.



With Docker you may manage the full lifecycle of your application development:

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

Now that we know what Docker is and what it is used for, let's dive into how to install Docker on Ubuntu 20.04 and how to start using it on Ubuntu in 6 steps to get you started.

Step 1: Set-up prerequisite packages

First, install a few Docker prerequisites that will allow you to use SSL data transfer (`apt-transport-https`), add commonly used certificates (`ca-certificates`), manage PPA's (`software-properties-common`), as well as download data from the Internet (`curl`).

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

Next, download the official Docker repository GPG key and add it to the apt-key trusted key manager:

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

And add the repository of Docker stable version for Ubuntu 20.04 (focal) to APT sources list:

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

You may want to double check the installation source priority for `docker-ce` package to make sure it is going to be installed from your recently added software repository.

```
apt-cache policy docker-ce
```

```
docker@t3st:~$ apt-cache policy docker-ce
docker-ce:
  Installed: (none)
  Candidate: 5:20.10.8-3~ubuntu-focal
  Version table:
    5:20.10.8-3~ubuntu-focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
    5:20.10.7-3~ubuntu-focal 500
```

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

Before installing any software, it is recommended to update the existing package list. Whether you are working locally, get the newest installation package requirements.

```
sudo apt update
```

It's now time to install Docker Community Edition. It includes the Docker platform with full core features, but no commercial support, nor enterprise functionality which is most often not necessary for small and medium businesses.

```
sudo apt install docker-ce
```

After installation Docker service should be up and running. You may double-check this by using `systemctl status` command, which is used to see whether a particular systemd process is active.

```
sudo systemctl status docker
```

If you successfully installed Docker, you would see a similar output with information about Docker being active.

```
docker@t3st:~$ sudo systemctl status docker
sudo: unable to resolve host t3st: Name or service not known
● docker.service - Docker Application Container Engine
    Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
    Active: active (running) since Thu 2021-09-30 10:49:34 EEST; 1min 7s ago
      TriggeredBy: ● docker.socket
        Docs: https://docs.docker.com
   Main PID: 14503 (dockerd)
     Tasks: 13
    Memory: 32.9M
      CGroup: /system.slice/docker.service
              └─14503 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/con...
```

Step 3: Start Using Docker

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

When working with Docker it is very handy to have system-wide information at hand. With docker info command you can always refer to this information to check for the number of containers (running/paused/stopped), types of plugins (volumes, networking, logs), and many more.

Let's now run a demo container from the default testing image hello-world.

```
docker run hello-world
```

When you execute this command for the first time, docker run will look for the Docker image (a read-only template with specified instructions) named *hello-world* locally. If it is unable to find the image, it will look it up in the Docker Hub image repository. If the image is found, Docker will automatically download and run it in a container (a runnable instance of an image) for you.

A detailed explanation of what is happening under the hood is then displayed in your terminal.

```
docker@t3st:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:9ade9cc2e26189a19c2e8854b9c8f1e14829b51c55a630ee675a5a9540ef6ccf
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/
```

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

```
docker@t3st:~$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
hello-world    latest    feb5d9fea6a5  10 days ago  13.3kB
```

You also have one stopped container that was created from the hello-world image. You may list all container logs with `-a` attribute (default only shows running containers).

```
docker container ls -a
```

```
docker@t3st:~$ docker container ls -a
CONTAINER ID  IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
c82a5dc6c4d1  hello-world  "/hello"  3 minutes ago  Exited (0) 3 minutes ago
                                                              
                                                               exciting_taussig
```

We can see the newly created container named `exciting_taussig` that stopped 3 minutes ago with an exit code 0.

Let's now remove this unused container and double check if it has been removed successfully.

```
docker rm exciting_taussig
```

```
docker container ls -a
```

```
docker@t3st:~$ docker rm exciting_taussig
exciting_taussig
docker@t3st:~$ docker container ls -a
CONTAINER ID  IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
```

Only after removing the associated container you are allowed to remove the unused hello-world image. Do so, and double check if it has been removed successfully.

```
docker rmi hello-world
```

```
docker images
```

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

Step 4: Run containers from images

Docker has an extensive image library called the Docker Hub where you may find thousands of Docker images that were created by official software vendors or 3rd parties.

For instance, let's build an isolated Debian base environment on our Ubuntu host. First off, search Docker Hub for a *debian* image.

```
docker search debian
```

```
docker@t3st:~$ docker search debian
NAME                               DESCRIPTION                                     STARS   OFFICIAL   AUTOMATED
ubuntu                            Ubuntu is a Debian-based Linux operating sys... 12897   [OK]
debian                            Debian is a Linux distribution that's compos... 4828    [OK]
arm32v7/debian                   Debian is a Linux distribution that's compos... 73      [OK]
```

You can see that *debian* image is available, has 4020 stars on Docker Hub and comes from an official source.

Download this Docker image without automatically running a container from it. The latest version will be pulled by default.

```
docker pull debian
```

```
docker@t3st:~$ docker pull debian
Using default tag: latest
latest: Pulling from library/debian
df5590a8898b: Pull complete
Digest: sha256:86ddd82ddd445aea3d2ea26af46cebd727bf2f47ed810fa1450a0d79722d55
Status: Downloaded newer image for debian:latest
docker.io/library/debian:latest
```

You can now run your Docker image and create a new container from it. Let's create a detached (`-d`) container in interactive (`-i`) mode using a pseudo-terminal (`-t`). We will also bridge (`--network="bridge"`) the container network with the host network, so we could access the Internet.

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

Feel free to double-check that your new container is up and running.

```
docker ps
```

```
docker@t3st:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
acf707828617 debian "bash" 4 seconds ago Up 4 seconds
interesting_wescoff
```

We can see a container named *interesting_wescoff* created 4 seconds ago that is based on a debian image with a bash command passed to it by default and no mapped system ports.

Let's stop the currently running container for now and check for active containers again.

```
docker container stop interesting_wescoff
```

```
docker ps
```

```
docker@t3st:~$ docker container stop interesting_wescoff
interesting_wescoff
docker@t3st:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
```

There are no active containers at this moment.

Step 5: Manipulate active containers

If we were to use docker run at this point, it would automatically create a second container from the debian image. To start an already existing container we should use a different command.

```
docker container start interesting_wescoff
```

```
docker@t3st:~$ docker container start interesting_wescoff
interesting_wescoff
```

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

```
docker exec -it interesting_wescoff /bin/bash
```

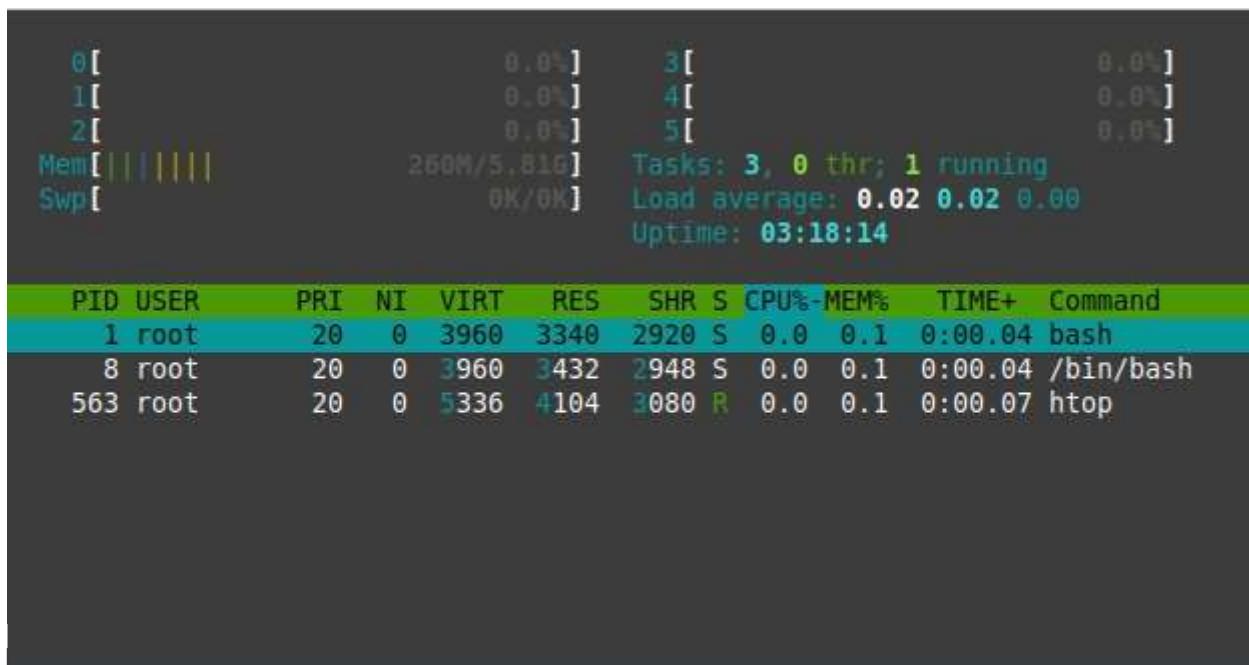
```
docker@t3st:~$ docker exec -it interesting_wescoff /bin/bash
root@acf707828617:/# htop
bash: htop: command not found
```

You can see from the terminal window that your hostname `t3st` has just been changed to a container identifier `acf707828617` which means that you have successfully initiated an interactive session with the `bash` utility.

Unfortunately, there is no `htop` utility, so the later command had failed. Nevertheless, we have previously specified the `-network="bridge"` parameter, so we should have Internet connectivity in our Debian container and can install the `htop` utility ourselves.

```
apt update && apt install htop
```

After installation is complete, we can run the `htop` utility inside the `debian` container.



We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

all installation prompts, if any.

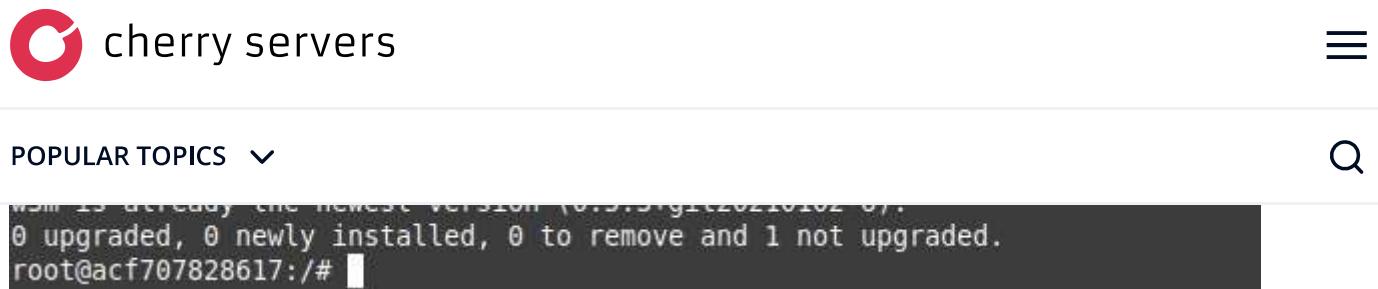
```
docker exec -d interesting_wescoff apt install -y w3m mc
```

After running the command, we get no output from the container whatsoever.

However, after starting a command line shell on the container in an interactive mode and running `apt install` again, we can see that `w3m` and `mc` packages have been successfully installed.

[Privacy Policy](#) [Terms of Service](#) [Knowledgebase](#)

[LOG IN](#) [USD](#) ▾



The screenshot shows a terminal window on the Cherry Servers website. The output of the command `apt upgrade` is displayed, showing that 0 packages were upgraded, 0 were newly installed, 0 were to be removed, and 1 was not upgraded. The terminal prompt is `root@acf707828617:~#`.

```
w3m is already the newest version (0.5.5-9git20210102-0).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
root@acf707828617:~#
```

And are running properly.

| Left | File | Command | Options | Right | | | |
|--------|------|---------|--------------|--------|------|------|--------------|
| .n | Name | Size | Modify time | .n | Name | Size | Modify time |
| <- / | | .[>] | <- / | ----- | .[>] | | |
| /bin | | 4096 | Oct 4 12:53 | /bin | | 4096 | Oct 4 12:53 |
| /boot | | 4096 | Apr 10 20:15 | /boot | | 4096 | Apr 10 20:15 |
| /dev | | 360 | Oct 4 12:15 | /dev | | 360 | Oct 4 12:15 |
| /etc | | 4096 | Oct 4 12:53 | /etc | | 4096 | Oct 4 12:53 |
| /home | | 4096 | Apr 10 20:15 | /home | | 4096 | Apr 10 20:15 |
| /lib | | 4096 | Sep 27 00:00 | /lib | | 4096 | Sep 27 00:00 |
| /lib64 | | 4096 | Sep 27 00:00 | /lib64 | | 4096 | Sep 27 00:00 |
| /media | | 4096 | Sep 27 00:00 | /media | | 4096 | Sep 27 00:00 |
| /mnt | | 4096 | Sep 27 00:00 | /mnt | | 4096 | Sep 27 00:00 |
| /opt | | 4096 | Sep 27 00:00 | /opt | | 4096 | Sep 27 00:00 |
| /proc | | 0 | Oct 4 12:15 | /proc | | 0 | Oct 4 12:15 |
| /root | | 4096 | Oct 4 13:03 | /root | | 4096 | Oct 4 13:03 |
| /run | | 4096 | Sep 27 00:00 | /run | | 4096 | Sep 27 00:00 |
| /sbin | | 4096 | Sep 27 00:00 | /sbin | | 4096 | Sep 27 00:00 |

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

But what if you have frequently used software, and don't want to install it anew every time you initialize a new docker container? What if you need to transfer your configuration to another machine to share your work with a friend or colleague?

As you may remember, Docker image is a read-only base template. Nevertheless, there's a chance to make a new local image by using the docker commit command.

To commit all the changes that we've made to our container, we first need to get the container ID.

```
docker container ls -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAMES |
|--------------|--------|---------|--------------|-------------|-------|---------------------|
| acf707828617 | debian | "bash" | 18 hours ago | Up 17 hours | | interesting_wescoff |

We can see that our container ID is *acf707828617*.

Now we can commit our container changes and make a local image by using the following pattern:

```
docker commit [container_id] [repository_name]/[image_name] .
```

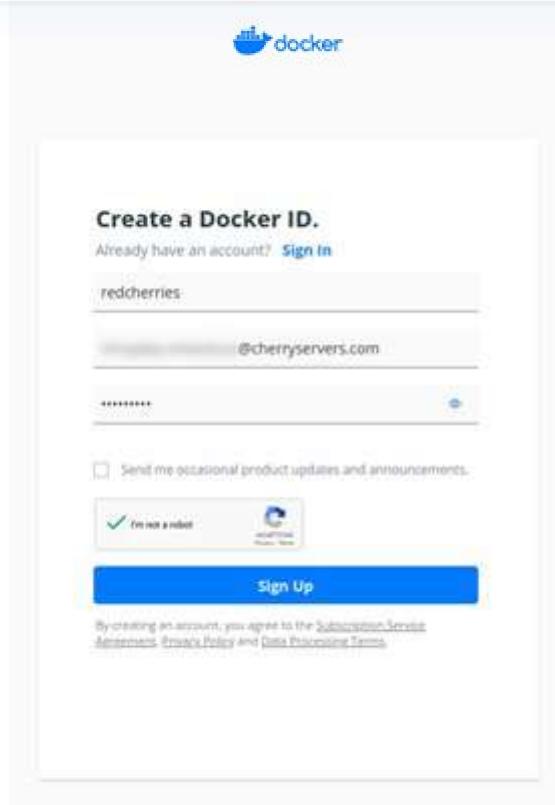
We can also use **-m** operand to leave a comment and **-a** operand to specify the name of the author.

```
docker commit -m "Installed htop, w3m and mc" -a "redcherries" acf707828617
redcherries/debian_htop_mc_w3m
```

| | | | | |
|--------------------------------|--------|--------------|---------------|-------|
| REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
| redcherries/debian_htop_mc_w3m | latest | b39811bbdd11 | 8 seconds ago | 269MB |

The `docker commit` made a local image with our implemented changes and assigned a unique SHA256 hash value to it.

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).



After creating an account, you should log-in to Docker Hub through your Docker client using your newly created username and password.

```
docker login
```

```
docker@t3st:~$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't
have a Docker ID, head over to https://hub.docker.com to create one.
Username: redcherries
Password:
WARNING! Your password will be stored unencrypted in /home/docker/.docker/config
.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
```

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

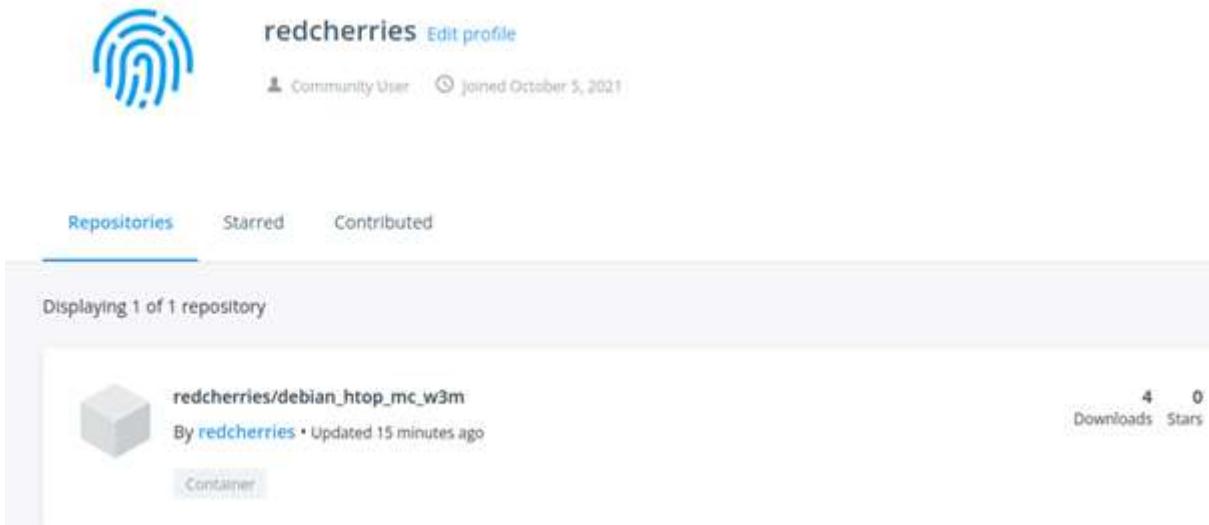
```
docker@t3st:~$ docker push redcherries/debian_htop_mc_w3m
Using default tag: latest
The push refers to repository [docker.io/redcherries/debian_htop_mc_w3m]
d4d2e706b741: Pushed
5e6a409f30b6: Mounted from library/debian
latest: digest: sha256:90bdfc7264438eaa7ba03ccff914574d4ed0cc345bc19168d12d20f58
edcccd46 size: 741
```

To check if our push was successful, search the Docker Hub for your repository.

```
docker search redcherries
```

```
docker@t3st:~$ docker search redcherries
NAME          DESCRIPTION     STARS     OFFICIAL     AUTOMATED
redcherries/debian_htop_mc_w3m          0
```

You may also use the web interface to check your Docker Hub account:



redcherries [Edit profile](#)

Community User Joined October 5, 2021

Repositories Starred Contributed

Displaying 1 of 1 repository

redcherries/debian_htop_mc_w3m
By [redcherries](#) • Updated 15 minutes ago
Container

4 0
Downloads Stars

Let's now clear the host and use `docker pull` command to download the image from Docker Hub.

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

```
docker@t3st:~$ docker container ls -a
CONTAINER ID  IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES
docker@t3st:~$ docker images
REPOSITORY  TAG      IMAGE ID   CREATED    SIZE
docker@t3st:~$ docker search redcherries
NAME          DESCRIPTION   STARS      OFFICIAL   AUTOMATED
redcherries/debian_htop_mc_w3m          0
docker@t3st:~$ docker pull redcherries/debian_htop_mc_w3m
Using default tag: latest
latest: Pulling from redcherries/debian_htop_mc_w3m
df5590a8898b: Pull complete
b4dad05b2b15: Pull complete
Digest: sha256:90bdfc7264438eaa7ba03ccff914574d4ed0cc345bc19168d12d20f58edcc46
Status: Downloaded newer image for redcherries/debian_htop_mc_w3m:latest
docker.io/redcherries/debian_htop_mc_w3m:latest
```

You may now run the downloaded container with docker run and test whether w3m is already installed.

```
docker@t3st:~$ docker run -it redcherries/debian_htop_mc_w3m
root@15f1c37f8c62:/# w3m cherryservers.com
```

Explore services [Watch video](#)

Start your cloud journey

[I have read and agree with privacy policy](#) and [terms of service](#).

[Create Account](#)

or sign up with

[Google](#) [Facebook](#)

Maximize Your Infrastructure Efficiency

Adjust your cloud to your workload, not the other way around

Excellent 4.3 [Cherry Servers Trustpilot rating](#) [6127ed2]

[For easy start and quick scale](#)

Top Quality Virtual Servers

Shared or dedicated virtual servers for small & medium workloads. Deployed in 3 minutes.

[Learn more](#)

[For high-end workloads](#)

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

knowledge and sharpen your skills.

Please refer to <https://hub.docker.com> in case you need to find out about a specific Docker image.

Rimvydas Misevičius

Network Administration

I develop Cherry Servers network infrastructure, and write articles to help other engineers make sense of tricky Network Engineering topics.

Share this article



Twitter



Facebook



LinkedIn

Cloud VPS - Cheaper Each Month

Start with \$9.99 and pay \$0.5 less until your price reaches \$6 / month.

[Check Available Servers](#)

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

[DOCKER](#) [KUBERNETES](#)

Docker Swarm vs. Kubernetes – What are the Subtle Differences?

Apr 14, 2020 by Marius Rimkus

Kubernetes and Docker Swarms are essential tools used to deploy containers inside a cluster. Learn how they differ and when to use them

[DOCKER](#)

How to Keep Docker Secrets When Working with Sensitive Data?

Aug 24, 2022 by Mantas Levinas

Docker secret is a BLOB of data that can be safely passed

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

DOCKER

How To Install Docker Compose on Ubuntu 20.04 | Step-by-Step

Nov 15, 2022 by Mantas Levinas

Learn how to use Docker Compose - Docker's official tool for defining and running multi-container applications on a single host.

Ready to get started?

[Create an Account](#)

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

Sales: sales@cherryservers.com

Support: support@cherryservers.com

GET THE LATEST NEWS FIRST

Subscribe to our newsletter

Subscribe

I agree to receive marketing communication as per privacy policy

PRICING

Dedicated Servers

Virtual Servers

Custom Server Builder

PRODUCTS

Dedicated Servers

Virtual Servers

Spot Servers

Cloud Platform

RESOURCES

API & Integrations

Case Studies

Knowledgebase

COMPANY

Contacts

About us

Our Team

Core Values

EU Projects

LEGAL

Terms of Service

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).

We use cookies to ensure seamless user experience for our website. Required cookies - technical, functional and analytical - are set automatically. Please accept the use of targeted cookies to ensure the best marketing experience for your user journey. You may revoke your consent at any time through our [Cookie Policy](#).