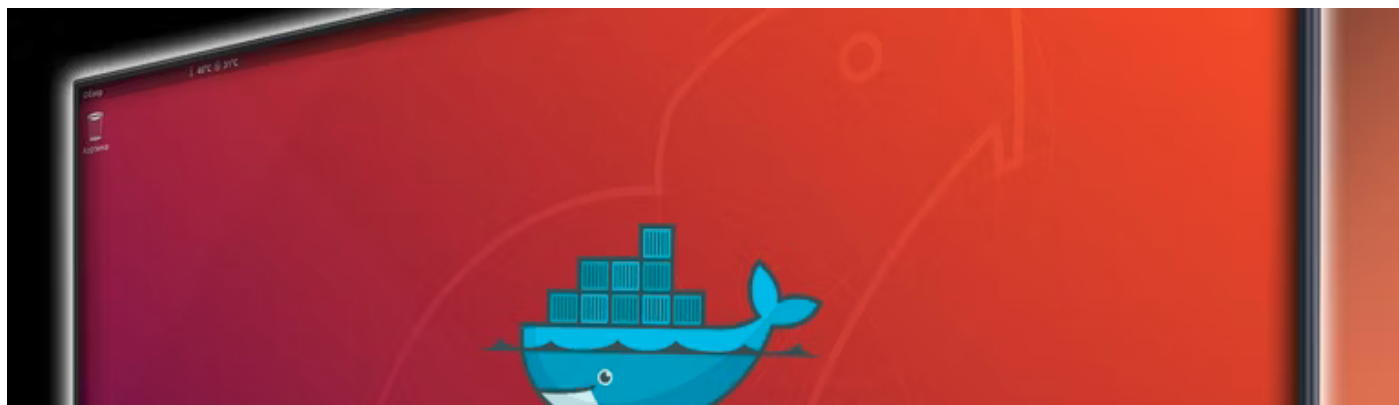




How to Run Ubuntu as a Docker Container



By Mwiza Kumwenda — Jul 5, 2023



Ubuntu



Follow



Share

Docker is the most-loved programming tool according to Stack Overflow's 2022 and 2023 developer survey. It is widely used in IT and has revolutionized the way we deploy applications.

Docker containers are a bit similar to virtual machines, but they are more lightweight and come packed with only the basic required elements of an operating system. To appreciate how minimal Docker containers can be, let's see how you can run Ubuntu in Docker.

Step 1: Getting the Ubuntu Docker Image

If you don't have Docker installed, follow our guide on [installing Docker on Ubuntu](#). And of course, you can also [install Docker on macOS](#) and [Windows](#).

Docker Hub is the recommended repository to get publicly verified and official Docker images. You can also use Docker Hub to manage and create private Docker images.

Cloud providers such as Azure and AWS also provide Docker repositories where you can host and manage your custom Docker images.

You can download the latest Ubuntu Docker image using the following command:

```
sudo docker pull ubuntu
```

Docker will download the latest image to your PC if you don't already have it stored locally.

If you are interested in a specific version, simply look at the available tags of the image in Docker Hub and then download it using that specific tag. For example, to download Ubuntu 20.04, run:

```
sudo docker pull ubuntu:20.04
```

You can list all the Docker images on your PC using the **sudo docker images** command.

```
mwiku@shire:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mongo	4.4	dba92ac6108e	4 weeks ago	438MB
archlinux	latest	fff7556904e9	5 weeks ago	410MB
mcr.microsoft.com/mssql/server	2022-latest	010a534cdc79	6 weeks ago	1.6GB
ubuntu	22.04	2dc39ba059dc	2 months ago	77.8MB
almalinux	9	9ea1821e2d87	2 months ago	189MB
nodered/node-red	3.0.2	a15fc0f4e930	3 months ago	475MB
nodered/node-red	latest	a15fc0f4e930	3 months ago	475MB
fedora	latest	98ffdbffd207	5 months ago	163MB

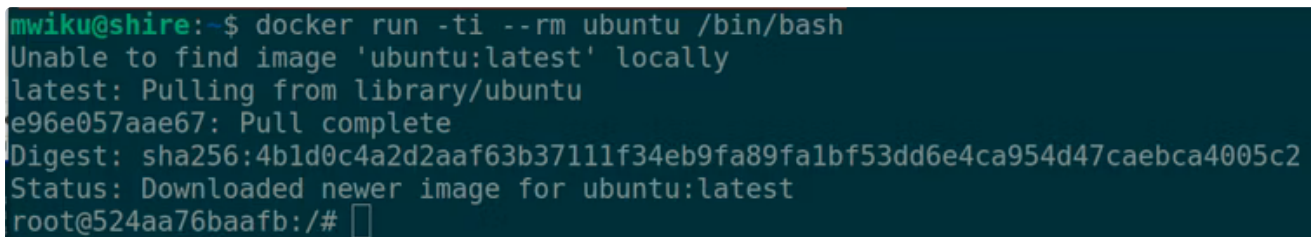
Step 2: Running the Ubuntu Docker Image

A Docker image is simply a blueprint of instructions for building a container. A container is a running instance of a Docker image. To bring the Docker Ubuntu image you've just downloaded to life, run the following command:

```
sudo docker run -ti --rm ubuntu /bin/bash
```

This command tells Docker to run the Docker Ubuntu container in an interactive terminal mode (**-ti**). The **/bin/bash** argument is a way of telling the container to run the Bash shell terminal. Finally, the **--rm** flag instructs Docker to automatically remove the Ubuntu Docker container after we stop it.

The shell starts as a root user and the terminal is similar to what you get on a typical Linux system. By default, the container gets a random hostname.

A terminal window with a dark background. The prompt is 'mwiku@shire:~\$'. The command entered is 'docker run -ti --rm ubuntu /bin/bash'. The output shows: 'Unable to find image 'ubuntu:latest' locally', 'latest: Pulling from library/ubuntu', 'e96e057aae67: Pull complete', 'Digest: sha256:4b1d0c4a2d2aaf63b37111f34eb9fa89fa1bf53dd6e4ca954d47caebca4005c2', 'Status: Downloaded newer image for ubuntu:latest', and the final prompt 'root@524aa76baafb:/#' with a cursor.

```
mwiku@shire:~$ docker run -ti --rm ubuntu /bin/bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
e96e057aae67: Pull complete
Digest: sha256:4b1d0c4a2d2aaf63b37111f34eb9fa89fa1bf53dd6e4ca954d47caebca4005c2
Status: Downloaded newer image for ubuntu:latest
root@524aa76baafb:/#
```

You'll also notice that the image does not have a GUI or many other standard command-line tools that come with Ubuntu. The Ubuntu Docker image is only about 78MB, which is super light.

Running Linux Commands on Ubuntu Docker

Normally, you check OS information on Linux using the **lsb_release -a** command, but in this case, you'll find that this command is not available because the Docker Ubuntu image doesn't have the **lsb-core** package installed by default. Remember, Docker images only come with the core elements and nothing else.

Fortunately, we can check the OS information using the cat command, which comes as part of the Bash shell.

```
cat /etc/os-release
```

```
root@524aa76baafb:/# cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.1 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.1 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
```

As you can see from the output, this is the LTS version of Ubuntu released in 2022. Thanks to the power of Docker, we can run Ubuntu in Docker using very minimal system resources.

And since this is simply a lightweight OS, we can install the applications we need using familiar Ubuntu commands. First, update the package sources and install **lsb-core** by typing:

```
apt update && apt install lsb-core
```

Note

You don't need to use the **sudo** command as you've logged in to the Docker shell as the root user.

Once installed, you can run the **lsb_release -a** command and this time around you'll get some output.

Saving the Docker Container State

When you stop the Docker container at this stage, you'll lose all the changes you made, including software updates and installed tools. That is how Docker containers are designed; they're easy to replace, stop, and manage.

As you might know by now, Docker is a versatile tool; it allows you to save the state of containers if you wish to. First, check the container ID of the Docker Ubuntu Image using the following command:

```
sudo docker ps
```

```
mwiku@shire:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
524aa76baafb	ubuntu	"/bin/bash"	50 minutes ago
1c3442888f25	mcr.microsoft.com/mssql/server:2022-latest	"/opt/mssql/bin/perm..."	3 weeks ago
:::1433->1433/tcp			
58f392746470	mongo:4.4	"docker-entrypoint.s..."	3 weeks ago

The preceding command lists all Docker containers on your PC. The Ubuntu image in the output above has the ID **524aa76baafb**, yours will be a different one.

Save the state of the container by running the following command:

```
docker commit -p container_id new_container_name
```

Remember to replace **container_id** in the following command with the correct one. Also, Docker image names can only be lowercase.

```
sudo docker commit -p 524aa76baafb myubuntu
```

The preceding command will pause the container before saving it and will

create a new Docker image named **myubuntu**. The new Docker image will contain all changes that you've made to it. And with that, you've just created a custom Ubuntu Docker image.

List your Docker images using the **sudo docker images** command; your new custom image should be listed along.

Persisting Data on the Ubuntu Docker Container

Another powerful feature of Docker is the ability to persist or share data with the host machine. There are two main options: using mounted volumes or Docker volumes. Docker advocates for the latter because it is better in comparison to mounted volumes.

You can create a Docker volume anywhere on your PC. Let's create it in the home directory and name it `Docker_Share`:

```
sudo mkdir -p Docker_Share
```

Next, stop the Ubuntu container using the following command, substituting **container_id** with the actual ID of the Ubuntu Docker container:

```
sudo docker stop container_id
```

Finally, we can run the Ubuntu Docker image to persist data using the **Docker_Share** directory using the command below. Alternatively, you can [create a docker-compose file](#) to easily fire up your Docker images.

```
sudo docker run -ti --rm -v ~/Docker_Share:/data ubuntu /bin/bash
```

The command will start the Ubuntu image and create the **/data** directory within the Docker container. The **/data** directory is mapped to the **Docker_Share** folder you created earlier.

You can access any created or modified files on the **/data** directory of the container using the **Docker_Share** directory. The reverse is also true; Docker will replicate any file modifications in the **Docker_Share** directory in the **/data** directory of the container.

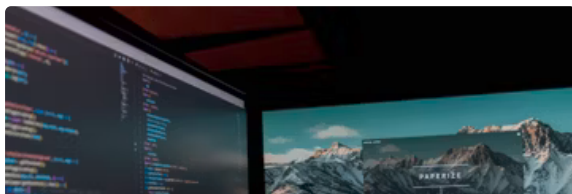
Docker Is a Great Replacement for Virtual Machines

Docker is a very powerful technology that allows you to deploy and run applications in a safe and secure environment. For instance, you can use it for a wide range of tasks, and in most cases, you'll be fine running a lightweight Ubuntu Docker container instead of running a virtual machine.

[Linux](#)[Ubuntu](#)[Docker](#)

Readers like you help support MakeUseOf. When you make a purchase using links on our site, we may earn an affiliate commission.
[Read More](#)

RECOMMENDED

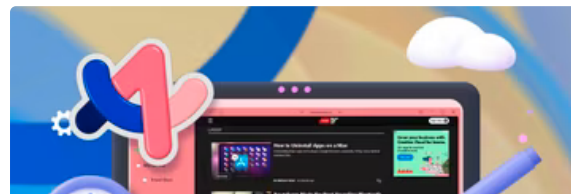


Virtual Machine

Why Should You Pick Docker Over a Virtual Machine?

Docker is becoming more popular, but why should you use it instead of a virtual machine?

Jul 13, 2021

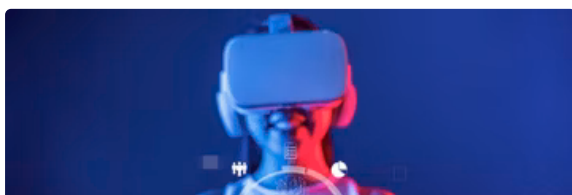


Browser

This New Browser Is a Productivity Miracle

Fed up of countless tabs and memory-hogging browsers? Check out Arc.

Jul 13, 2024



Artificial Intelligence

These 7 Signs Show We've Already Reached Peak AI

How much is too much AI?

1 day ago



Cable Management

I Sorted Through My Random Box of Cables, and These Are...

Everyone has a random box of cables lurking somewhere.

23 hours ago



Password Manager

Link copied to clipboard

I've Started Using NordPass: Why You Should Give This...

If you're wondering about using a new password manager, NordPass is the one for you.

3 hours ago



Smartphone Privacy

This Popular App Company Was Sold, and Now Its Androi...

It's one of Android's most beloved app suites, but many users are now looking for alternatives.

1 day ago



Join Our Team

Our Audience

About Us

Press & Events

Contact Us

Follow Us



Advertising

Careers

Terms

Privacy

Policies

MUO is part of the **Valnet Publishing Group**

Copyright © 2024 Valnet Inc.