

Manjaro Linux Docker installation

 linuxconfig.org/manjaro-linux-docker-installation

30 June 2020

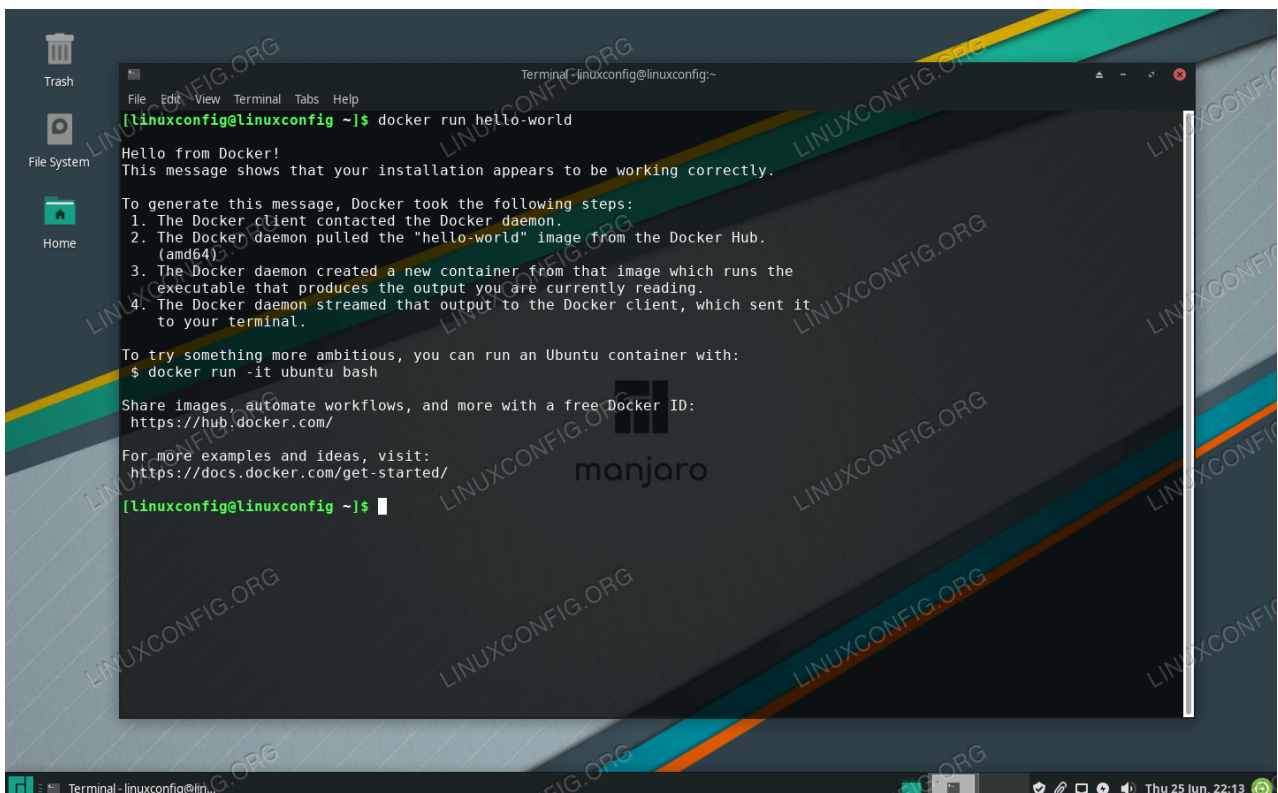
Docker is a tool that is used to run software in a container. It's a great way for developers and users to worry less about compatibility with an operating system and dependencies because the contained software should run identically on any system.

Docker is available for download and installation on Manjaro as well as most other distributions of Linux. After Docker is installed, you can use it to install software packages much the same way you would use your distro's package manager to download an app. The difference of using Docker is that everything is more automated, with compatibility and dependencies no longer being potential issues.

In this guide, we'll show you how to install Docker on Manjaro Linux and get started with installing containerized software.

In this tutorial you will learn:

- How to install Docker
- How to run Docker without root
- How to search for a Docker image
- How to install a Docker image
- How to run a Docker image
- How to monitor Docker with various commands



```
Terminal - linuxconfig@linuxconfig:~  
File Edit View Terminal Tabs Help  
[linuxconfig@linuxconfig ~]$ docker run hello-world  
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/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/  
[linuxconfig@linuxconfig ~]$
```

Docker running a container image

Category	Requirements, Conventions or Software Version Used
System	<u>Manjaro Linux</u>
Software	Docker
Other	Privileged access to your Linux system as root or via the <code>sudo</code> command.
Conventions	<code>#</code> – requires given <u>linux commands</u> to be executed with root privileges either directly as a root user or by use of <code>sudo</code> command <code>\$</code> – requires given <u>linux commands</u> to be executed as a regular non-privileged user

Software Requirements and Linux Command Line Conventions

Install Docker

To get started installing Docker, the first thing you should do is open a terminal and make sure that Manjaro is up to date.

```
$ sudo pacman -Syu
```

Next, execute the following command to install Docker:

```
$ sudo pacman -S docker
```

DID YOU KNOW?

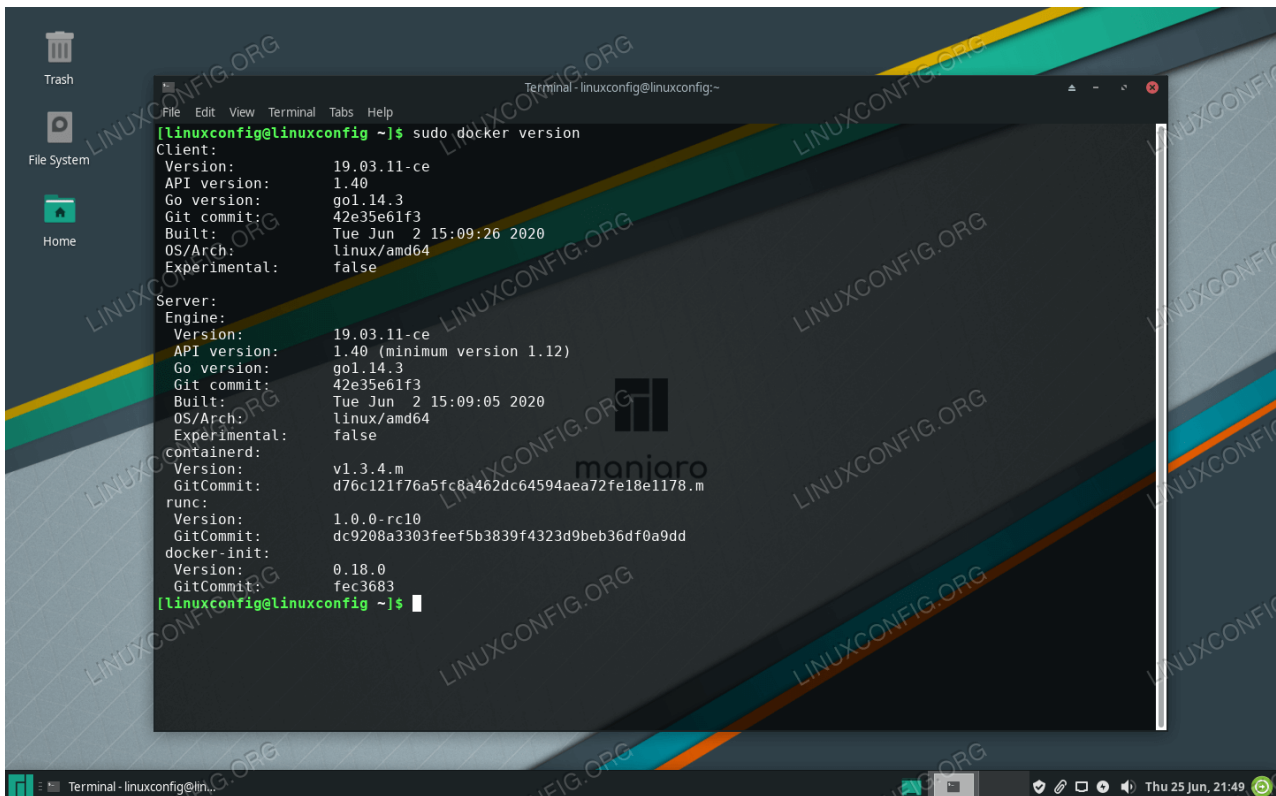
You can also install the development version of Docker from the AUR, but it's not recommended to use that release in a production environment.

Once installation is completed, start the Docker service and, optionally, enable it to run whenever the system is rebooted:

```
$ sudo systemctl start docker.service  
$ sudo systemctl enable docker.service
```

You can verify that Docker is installed and gather some information about the current version by entering this command:

```
$ sudo docker version
```



```
linuxconfig@linuxconfig ~]$ sudo docker version
Client:
Version:           19.03.11-ce
API version:       1.40
Go version:        go1.14.3
Git commit:        42e35e61f3
Built:             Tue Jun 2 15:09:26 2020
OS/Arch:           linux/amd64
Experimental:      false

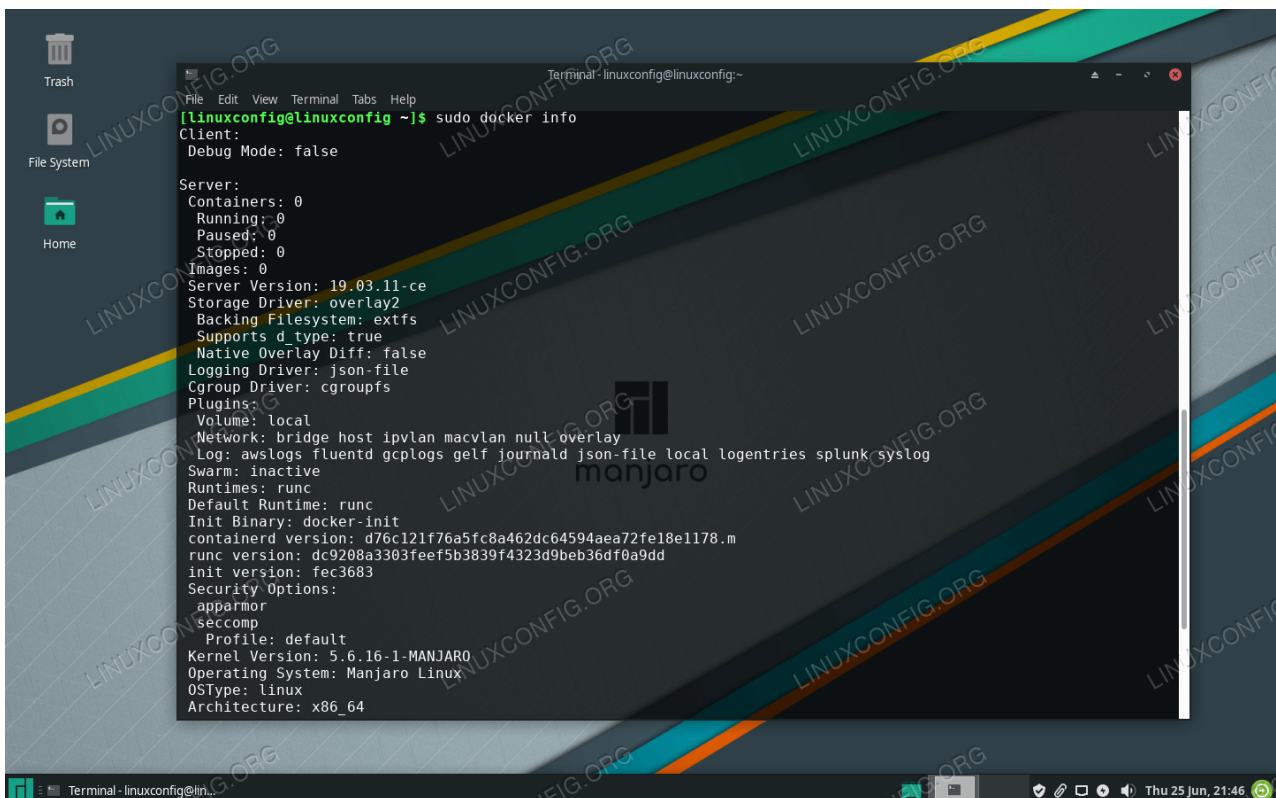
Server:
Engine:
Version:           19.03.11-ce
API version:       1.40 (minimum version 1.12)
Go version:        go1.14.3
Git commit:        42e35e61f3
Built:             Tue Jun 2 15:09:05 2020
OS/Arch:           linux/amd64
Experimental:      false

containerd:
Version:           v1.3.4-m
GitCommit:         d76c121f76a5fc8a462dc64594aea72fe18e1178.m
runc:
Version:           1.0.0-rc10
GitCommit:         dc9208a3303feef5b3839f4323d9beb36df0a9dd
docker-init:
Version:           0.18.0
GitCommit:         fec3683
```

Output of docker version command

There is also a quick and easy way to see how many Docker containers are currently running and see some of Docker's configured options by entering:

```
$ sudo docker info
```



```
linuxconfig@linuxconfig ~]$ sudo docker info
Client:
 Debug Mode: false

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 19.03.11-ce
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
 Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: d76c121f76a5fc8a462dc64594aea72fe18e1178.m
 runc version: dc9208a3303feef5b3839f4323d9beb36df0a9dd
 init version: fec3683
 Security Options:
  apparmor
  seccomp
   Profile: default
 Kernel Version: 5.6.16-1-MANJARO
 Operating System: Manjaro Linux
 OSType: linux
 Architecture: x86_64
```

Output of docker info command showing its configuration

Run Docker without root

By default, you'll have to use `sudo` or login to root anytime you want to run a Docker command. This next step is optional, but if you'd prefer the ability to run Docker as your current user, add your account to the `docker` group with this command:

```
$ sudo usermod -aG docker $USER
```

You'll need to reboot your system for those changes to take effect.

```
$ reboot
```

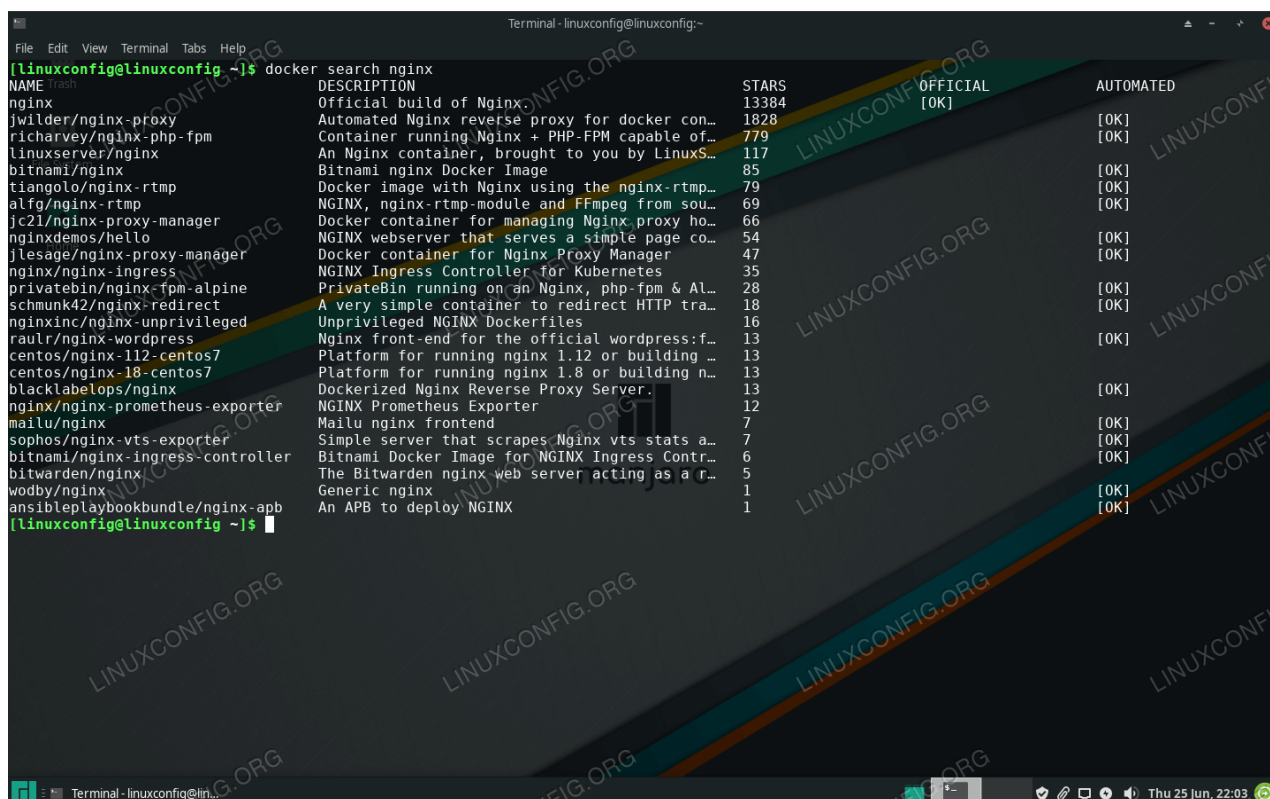
Searching for a Docker image

Now you're ready to install images with Docker. If you already know the name of an image that you'd like to install, you can move on to the next section. If you need to search through Docker for the desired software, you can use the following command syntax:

```
$ docker search [name]
```

For example, let's try searching for `nginx`, which is popular web server software.

```
$ docker search nginx
```



```
Terminal - linuxconfig@linuxconfig:~$ docker search nginx
NAME                DESCRIPTION                STARS     OFFICIAL   AUTOMATED
nginx               Official build of Nginx.   13384     [OK]
jwilder/nginx-proxy Automated Nginx reverse proxy for docker con... 1828
richarvey/nginx-php-fpm Container running Nginx + PHP-FPM capable of... 779
linuxserver/nginx   An Nginx container, brought to you by LinuxS... 117
bitnami/nginx       Bitnami nginx Docker Image 85
tiangolo/nginx-rtmp Docker image with Nginx using the nginx-rtmp... 79
alfg/nginx-rtmp     NGINX, nginx-rtmp-module and FFmpeg from sou... 69
jc21/nginx-proxy-manager Docker container for managing Nginx proxy ho... 66
nginxdemos/hello    NGINX webserver that serves a simple page co... 54
jlesage/nginx-proxy-manager Docker container for Nginx Proxy Manager 47
nginx/nginx-ingress NGINX Ingress Controller for Kubernetes 35
privatebin/nginx-fpm-alpine PrivateBin running on an Nginx, php-fpm & AL... 28
schmunk42/nginx-redirect A very simple container to redirect HTTP tra... 18
nginxinc/nginx-unprivileged Unprivileged NGINX Dockerfiles 16
raulr/nginx-wordpress Nginx front-end for the official wordpress:f... 13
centos/nginx-112-centos7 Platform for running nginx 1.12 or building ... 13
centos/nginx-18-centos7 Platform for running nginx 1.8 or building n... 13
blacklabelops/nginx Dockerized Nginx Reverse Proxy Server. 13
nginx/nginx-prometheus-exporter NGINX Prometheus Exporter 12
mailu/nginx         Mailu nginx frontend 7
sophos/nginx-vts-exporter Simple server that scrapes Nginx vts stats a... 7
bitnami/nginx-ingress-controller Bitnami Docker Image for NGINX Ingress Contr... 6
bitwarden/nginx     The Bitwarden nginx web server acting as a r... 5
wodby/nginx         Generic nginx 1
ansibleplaybookbundle/nginx-apb An APB to deploy NGINX 1
```

Docker can search for any available container images

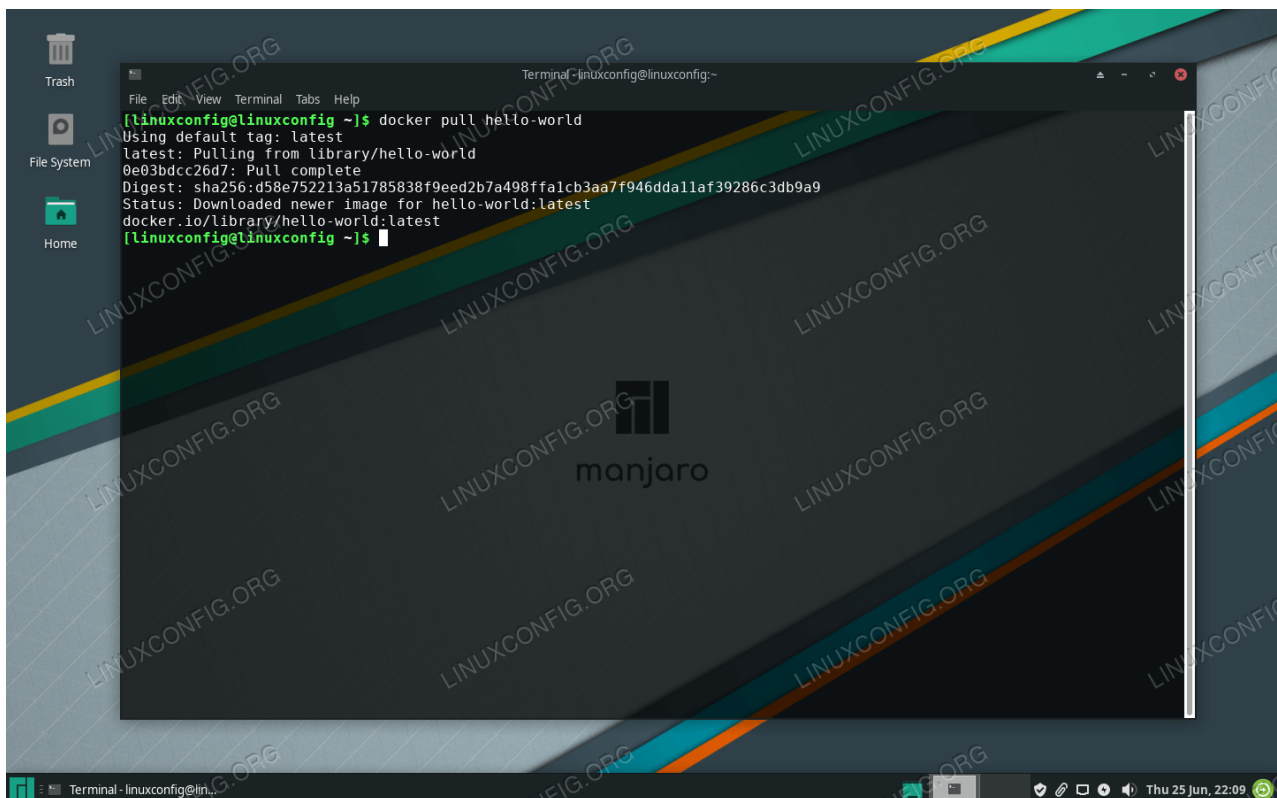
As you can see, there is one official image for nginx (indicated by the `OFFICIAL` column) simply called `nginx`. There are also other releases available, and you would have to read their descriptions to see what they do differently from the official image.

Install a Docker image

AD

Once you know which image you'd like to install, you can use the following command to instruct Docker to download the desired software. Just as an example, we'll install the `hello-world` package which can be used to make sure that Docker is able to download and run images successfully.

```
$ docker pull hello-world
```



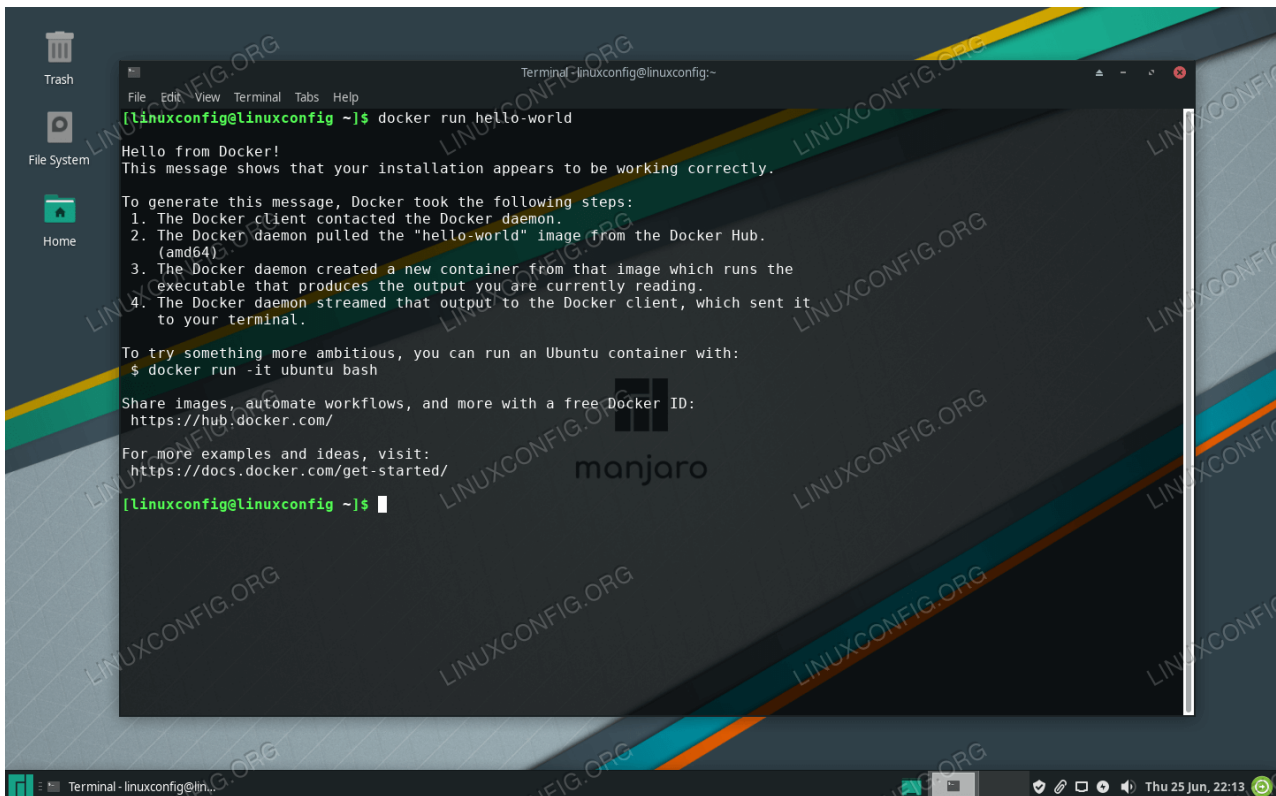
Docker downloading a container image

The output in the screenshot above indicates that Docker was able to find and download the image we specified.

Running a Docker image

Now that the image is downloaded, run it with the following command:

```
$ docker run hello-world
```

A screenshot of a Linux desktop environment with a Manjaro logo in the background. A terminal window is open, displaying the output of the command 'docker run hello-world'. The output includes a 'Hello from Docker!' message, a confirmation that the installation is working, a list of four steps Docker took to run the container, and links to Docker documentation and the Docker Hub. The terminal prompt is '[linuxconfig@linuxconfig ~]\$'.

Docker was able to run the hello-world image successfully

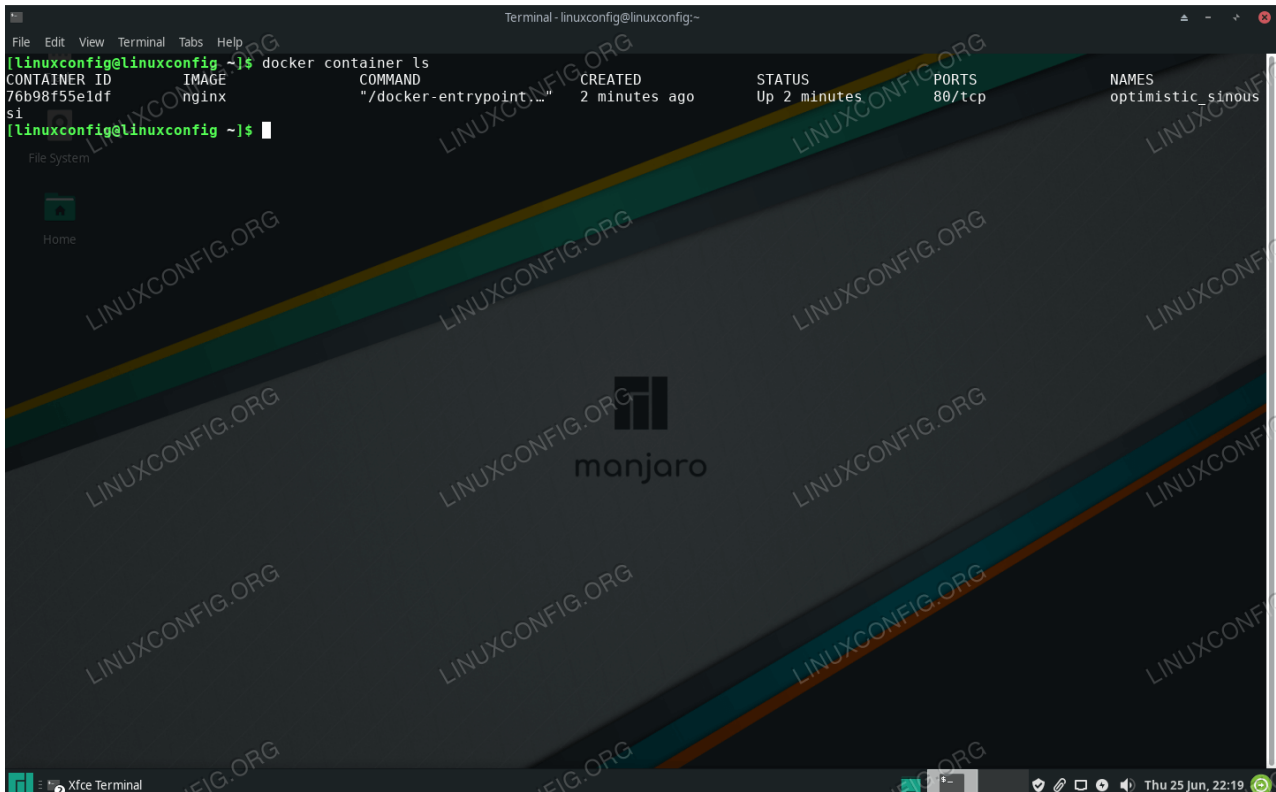
Monitoring Docker

Docker gives us a lot of options to see what's going on with our images, how many system resources they're using, etc. The following commands come in handy for monitoring Docker and our installed images.

To see which Docker containers are running and check their current status, type:

```
$ docker container ls
```

```
Terminal - linuxconfig@linuxconfig:~  
File Edit View Terminal Tabs Help  
[linuxconfig@linuxconfig ~]$ docker container ls  
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS      NAMES  
76b98f55e1df   nginx     "/docker-entrypoint..." 2 minutes ago  Up 2 minutes  80/tcp     optimistic_sinous  
si  
[linuxconfig@linuxconfig ~]$
```

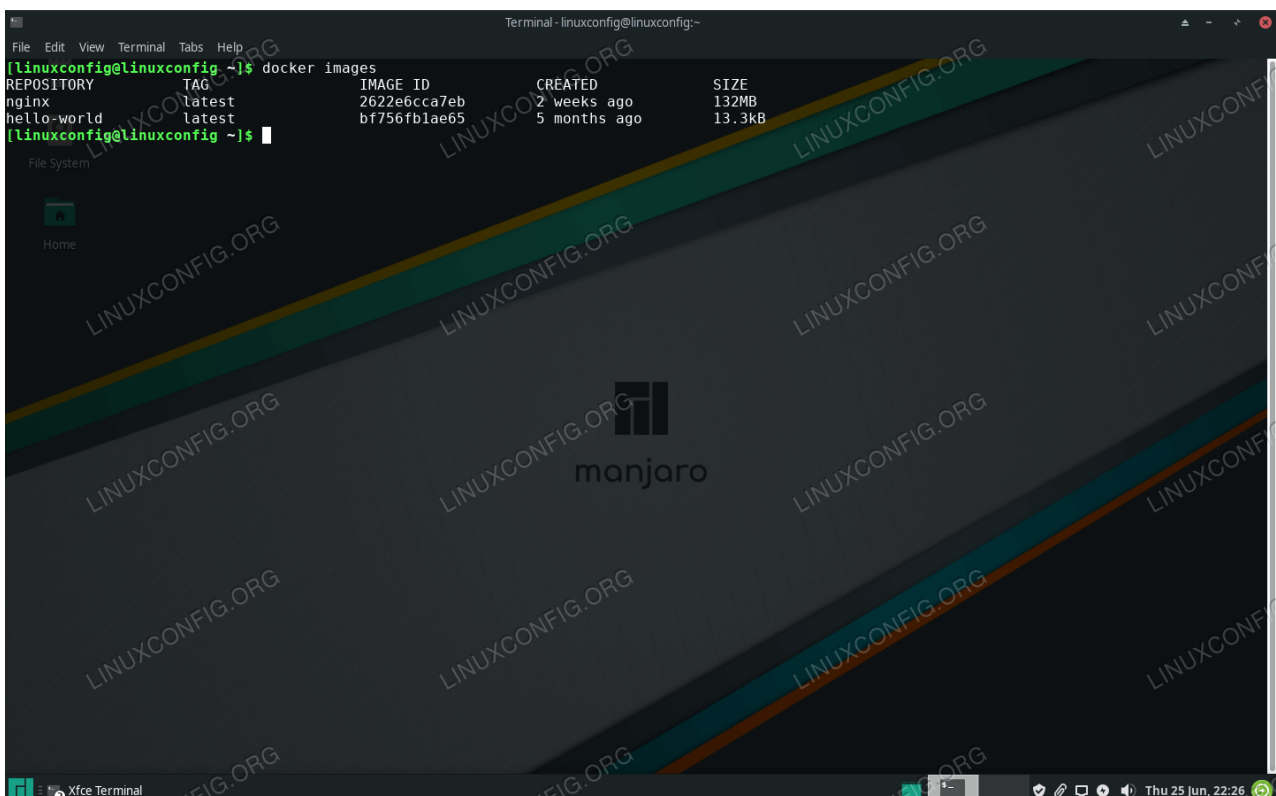


The docker container ls command shows currently running containers
To see a list of all the Docker images installed, type:

```
$ docker images
```

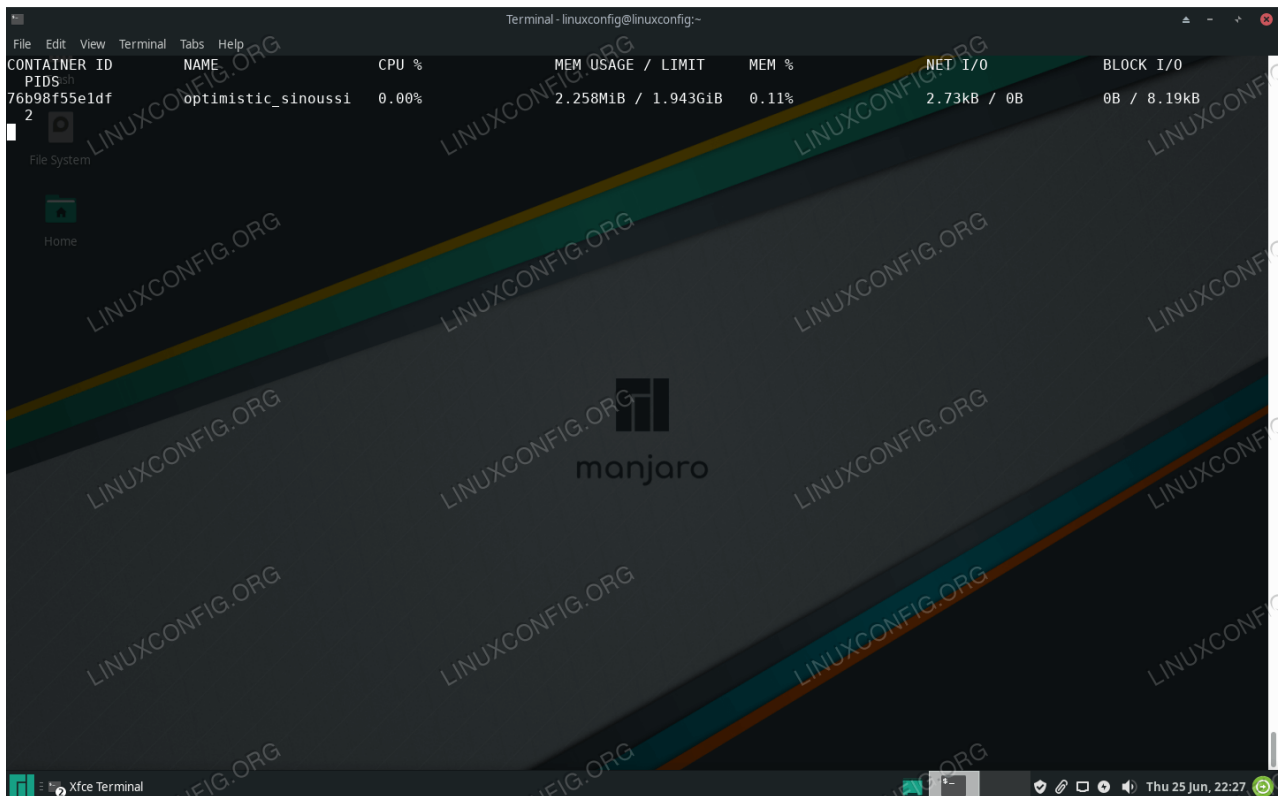
AD

```
Terminal - linuxconfig@linuxconfig:~  
File Edit View Terminal Tabs Help  
[linuxconfig@linuxconfig ~]$ docker images  
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE  
nginx         latest   2622e6cca7eb   2 weeks ago   132MB  
hello-world   latest   bf756fb1ae65   5 months ago  13.3kB  
[linuxconfig@linuxconfig ~]$
```



The docker images command shows which images are currently installed in Docker
To see the current CPU, RAM, and network usage of running images, type:

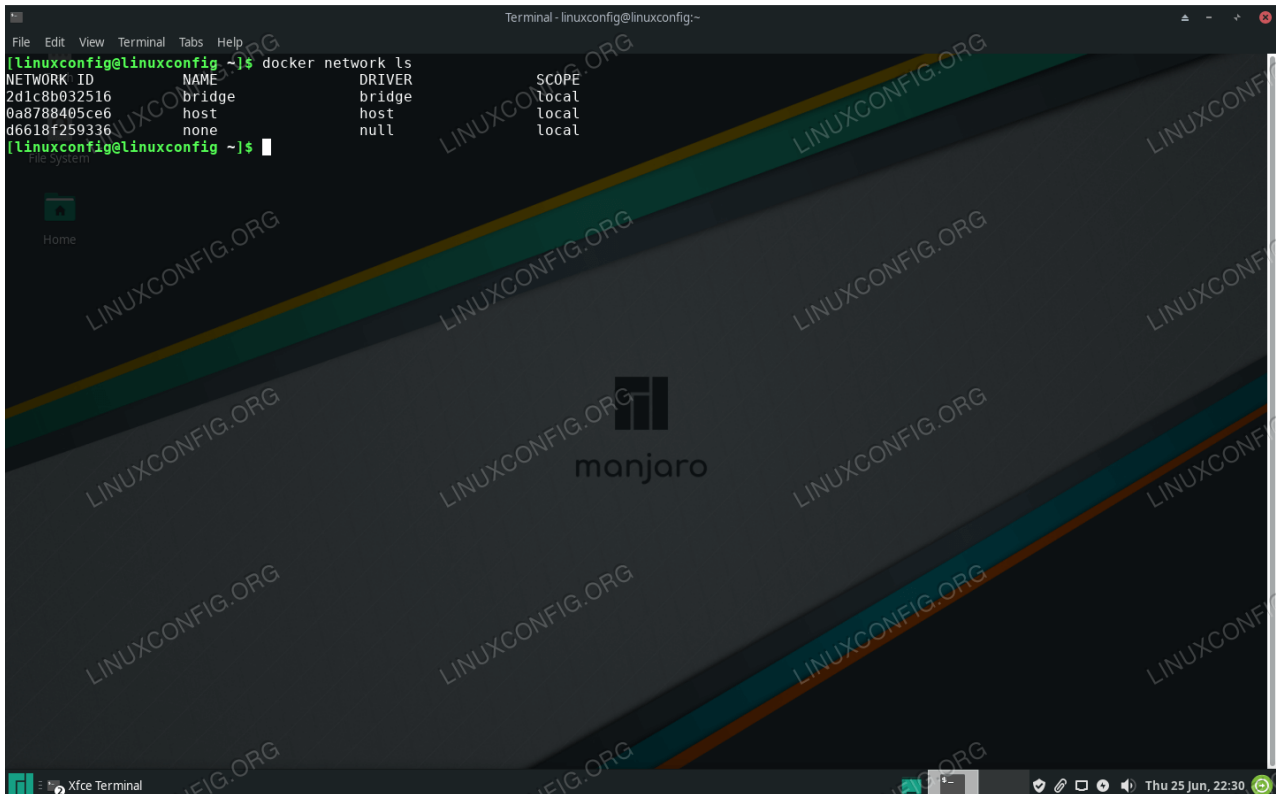
```
$ docker stats
```



See the current system usage of Docker images with docker stats command
To see Docker's network configuration, type:

AD

```
$ docker network ls
```


A screenshot of a Linux desktop environment with a Manjaro logo in the center. A terminal window is open in the top-left corner, displaying the command 'docker network ls' and its output. The output is a table with four columns: NETWORK ID, NAME, DRIVER, and SCOPE. It lists three networks: 'bridge' (local), 'host' (local), and 'null' (local). The desktop background is dark with a diagonal pattern and the Manjaro logo. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The bottom status bar shows 'Xfce Terminal' and the date/time 'Thu 25 Jun, 22:30'.

```
Terminal - linuxconfig@linuxconfig:~  
File Edit View Terminal Tabs Help  
[linuxconfig@linuxconfig ~]$ docker network ls  
NETWORK ID          NAME                DRIVER              SCOPE  
2d1c8b032516        bridge             bridge             local  
0a8788405ce6        host              host              local  
d6618f259336        none              null              local  
[linuxconfig@linuxconfig ~]$
```

See currently configured networks in Docker with `docker network ls`

Closing Thoughts

Docker is pretty easy to use on Manjaro. Like most new things, you just need to learn the command syntax that comes with it, which is very simple anyway. Now that Docker is up and running on your Manjaro system, you'll be able to search for and install new container images as you please. Chances are that you'll find this a lot more convenient than manually installing or building certain packages on your own.