How To Install Docker and Docker-Compose On Raspberry Pi.

Published on February 07, 2024

#raspberrypi #docker

Written by Victor Cobos

Table of Contents

In this article, we'll explore how to enhance your Raspberry Pi by installing Docker and Docker Compose, two powerful tools that significantly expand its capabilities. Docker brings the world of containerization to your fingertips, allowing for the deployment of applications in isolated environments. This means you can run multiple applications on your Raspberry Pi without them interfering with each other, optimizing both performance and resource utilization. Docker Compose further simplifies the management of multi-container Docker applications, making it easier to define and share complex setups. By the end of this guide, you'll have a Raspberry Pi equipped to handle a diverse range of projects with greater ease and efficiency, demonstrating why Docker is a valuable addition to your microcomputing toolkit.

Prerequisites

- Raspberry Pi with a running Raspbian OS
- SSH connection enabled

To do this you can check Raspberry Pi Setup

1. Update and Upgrade

First of all make sure that the system runs the latest version of the software.
Run the command:

```
sudo apt-get update && sudo apt-get upgrade
```

2. Install Docker

Now is time to install Docker! Fortunately, Docker provides a handy install script for that, just run:

```
curl -fsSL test.docker.com -o get-docker.sh && sh get-docker.sh
```

3. Add a Non-Root User to the Docker Group

By default, only users who have administrative privileges (root users) can run containers. If you are not logged in as the root, one option is to use the sudo prefix. However, you could also add your non-root user to the Docker group which will allow it to execute docker commands.

The syntax for adding users to the Docker group is:

```
sudo usermod -aG docker [user_name]
```

To add the permissions to the current user run:

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

Check it is running:

```
groups ${USER}
```

Reboot the Raspberry Pi to let the changes take effect.

4. Install Docker-Compose

Docker-Compose usually gets installed using pip3. For that, we need to have python3 and pip3 installed. If you don't have it installed, you can run the following commands:

```
sudo apt-get install libffi-dev libssl-dev
sudo apt install python3-dev
sudo apt-get install -y python3 python3-pip
```

Once python3 and pip3 are installed, we can install Docker-Compose using the following command:

```
sudo pip3 install docker-compose
```

5. Enable the Docker system service to start your containers on boot

This is a very nice and important addition. With the following command you can configure your Raspberry Pi to automatically run the Docker system service, whenever it boots up.

```
sudo systemctl enable docker
```

With this in place, containers with a restart policy set to always or unless-stopped will be re-started automatically after a reboot.

6. Run Hello World Container

The best way to test whether Docker has been set up correctly is to run the Hello World container.
To do so, type in the following command:

```
docker run hello-world
```

Once it goes through all the steps, the output should inform you that your installation appears to be working correctly.

7. A sample Docker Compose file

This section shows a quick sample of a Docker-Compose file, which starts three containers that once started will automatically come up, if the Raspberry Pi get fully power cycled. To learn more about the sample project, visit Docker Speed Test project on GitHub.

```
version: '3'
services:
  # Tests the current internet connection speed
  # once per hour and writes the results into an
  # influxDB instance
  speedtest:
    image: robinmanuelthiel/speedtest:0.1.1
    restart: always
    depends_on:
      - influxdb
    environment:
      - LOOP=true
      - LOOP_DELAY=3600 # Once per hour
      - DB_SAVE=true
      - DB_HOST=http://influxdb:8086
      - DB_NAME=speedtest
      - DB_USERNAME=admin
      - DB_PASSWORD=<MY_PASSWORD>

  # Creates an InfluxDB instance to store the
  # speed test results
  influxdb:
    image: influxdb
    restart: always
    volumes:
      - influxdb:/var/lib/influxdb
    ports:
      - "8083:8083"
      - "8086:8086"
    environment:
      - INFLUXDB_ADMIN_USER=admin
      - INFLUXDB_ADMIN_PASSWORD=<MY_PASSWORD>
      - INFLUXDB_DB=speedtest

  # Displays the results in a Grafana dashboard
  grafana:
    image: grafana/grafana:latest
    restart: always
    depends_on:
      - influxdb
    ports:
      - 3000:3000
    volumes:
      - grafana:/var/lib/grafana

volumes:
  grafana:
  influxdb:
```

To start the containers using Docker-Compose, run the following command:

```
docker-compose -f docker-compose.yaml up -d
```

Find Raspberry Pi Docker Images

Raspberry Pi is based on ARM architecture. Hence, not all Docker images will work on your Raspberry Pi.
Remember that when searching for images to pull from Docker Hub. Apply the Architectures filter to search for supported apps.

How to Upgrade Docker on Raspberry Pi?

Upgrade Docker using the package manager with the command:

```
sudo apt-get upgrade
```

How to Uninstall Docker on Raspberry Pi?

You can simply remove docker using the package manager:

```
sudo apt-get purge docker-ce
```

Note: Depending on the version of software, you may need to use an additional command to completely remove Docker:

```
sudo apt-get purge docker-ce-cli
```

To delete leftover images, containers, volumes and other related data, run the following command:

```
sudo rm -rf /var/lib/docker
```

Edited configuration files must be deleted manually.

Conclusion

Congratulations! Your Raspberry Pi is now fully equipped with Docker and Docker Compose, marking the beginning of a new chapter in your journey with this versatile device. You're set to develop isolated and lightweight applications using containers, a method that not only maximizes efficiency but also revolutionizes the way you manage and deploy projects. With Docker's power at your fingertips, your Raspberry Pi transforms into an even more powerful tool, ready to handle diverse applications with grace and agility. Explore, experiment, and enjoy the seamless integration of development processes Docker enables. Happy containerizing 🐳!

References

- How To Install Docker On Raspberry Pi
- Setup your Raspberry Pi for Docker and Docker-Compose