Docker

Christian Külker

2024-02-19

Contents

1 Introduction

Docker is an open source software mechanism for building, shipping, and running applications. Docker allows you to separate your software applications from your software infrastructure through an abstraction layer so you can deploy software in a generalized way. With Docker, you can manage your servers the same way you manage your applications. By taking advantage of Docker's methods for rapidly shipping, testing, and deploying code, Docker promises to significantly reduce the time between writing code and running it on a server in a production environment.

Docker Engine can be installed in different ways:

- Setup own Docker repository and install from them for the ease of installation and upgrade tasks. This is the recommended approach, except for Raspbian.
- Download DEB packages and install them manually and manage upgrades completely manually. This gives full control on systems not connected to the internet.
- In testing and development environments scripts can install Docker in one shot. This is currently the most used approach for Raspbian.

=> https://docs.docker.com/engine/install/debian/

Membership in the docker group is more dangerous than `sudo`, according to the Debian Wiki. This is due to the `setUID` root of the docker daemon and easy access to root. Access to docker commands therefore effectively grants root access. Therefore, do not add users to the docker group and use docker only with `sudo`.

2   Editions

Docker comes in editions.

- ce: community edition
  - edge: monthly release
  - stable: quarterly release
- ee: enterprise edition
  - quarterly release

3   Debian 11 on ARM

The `docker.io` package contains the daemon and client. From the package description:

> "Using docker.io on non-amd64 hosts is not supported at this time. Please be careful when using it on anything besides amd64."

4   Docker Repository

Docker is used through what is called a Docker repository. That is, you create a Docker repository before you install the Docker engine. After installation, this repository can be used to update Docker. **Warning:** Rasbian (or Debian for Arm?) users cannot use this method. Instead, they use a so-called "convenience script", which can be obtained via a link. This method has not been tested.

5   Docker on Raspberry Pi 4 Debian 11 Bullseye

This works for Debian on `x86_64` / `amd64` , `armhf` , `arm64` , and `Raspbian` and is a hello world example.

```
aptitude purge docker-ce docker-ce-cli containerd.io
aptitude install docker docker.io containerd runc
 # test docker installation
docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
93288797bd35: Pull complete
Digest:
↪   sha256:cc15c5b292d8525effc0f89cb299f1804f3a725c8d05e158653a563f15e4f68
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.
    (arm64v8)
 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/
```

As root

```
export U=userid
aptitude install docker.io
root@w2:~# touch /var/run/docker.sock
```

```
root@w2:~# chown $U.$U /var/run/docker.so
```

As User

```
docker build --tag dh-venv-builder .
[...] (some minutes later)
Successfully built fba3ab967839
Successfully tagged dh-venv-builder:latest


docker run --rm dh-venv-builder tar -C /dpkg -c . | tar -C dist -xv
./
./dh-virtualenv_1.2.2-1~buster_all.deb
./dh-virtualenv_1.2.2-1~buster_amd64.buildinfo
./dh-virtualenv_1.2.2-1~buster_amd64.changes
```

5.1 Virtual Environment

```
aptitude install dh-virtualenv
```

6 Simple Docker Usage

List all containers

```
docker container ls
CONTAINER ID    IMAGE ...
419f0ee92d82    ...
```

Enter a container

```
docker exec -i -t 419f0ee92d82 bash
```

Leave a container with a bash shell

```
exit
```

7 History

Version	Date	Notes
0.1.2	2024-02-19	+Docker Repository, on Raspberry Pi4, Usage

Version	Date	Notes
0.1.1	2023-03-28	Improve writing, change cite format
0.1.0	2020-11-05	Initial release

8   Disclaimer of Warranty

THERE IS NO WARRANTY FOR THIS INFORMATION, DOCUMENTS AND PROGRAMS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE INFORMATION, DOCUMENT OR THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE INFORMATION, DOCUMENTS AND PROGRAMS IS WITH YOU. SHOULD THE INFORMATION, DOCUMENTS OR PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

9   Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE INFORMATION, DOCUMENTS OR PROGRAMS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE INFORMATION, DOCUMENTS OR PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE INFORMATION, DOCUMENTS OR PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.