

# Docker Volumes



CLARUSWAY©  
WAY TO REINVENT YOURSELF

## Table of Contents

- ▶ Review of Docker Architecture
- ▶ Container Layers
- ▶ Manage data in Docker
- ▶ Declaration of volumes
- ▶ Docker Volume Behaviours
- ▶ docker volume Commands

CLARUSWAY©  
WAY TO REINVENT YOURSELF

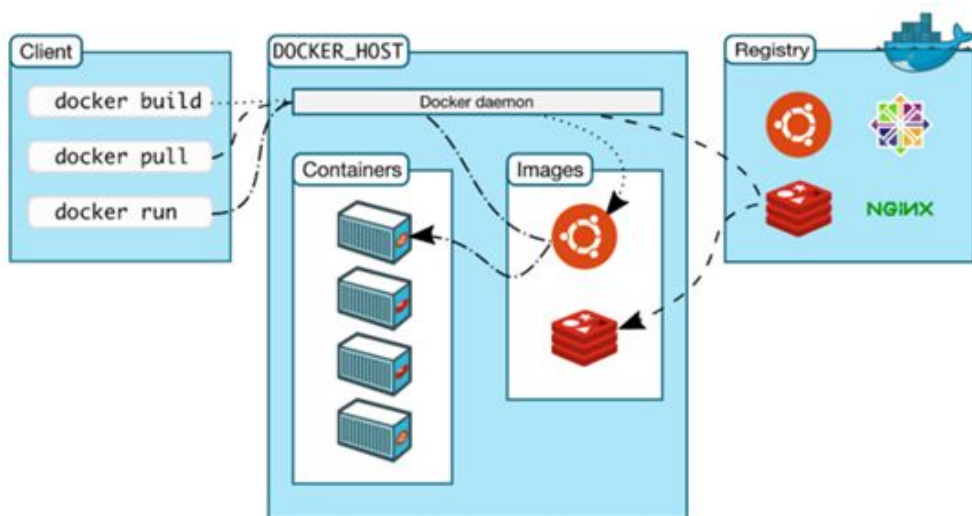


1

# Docker Architecture



## Docker Architecture



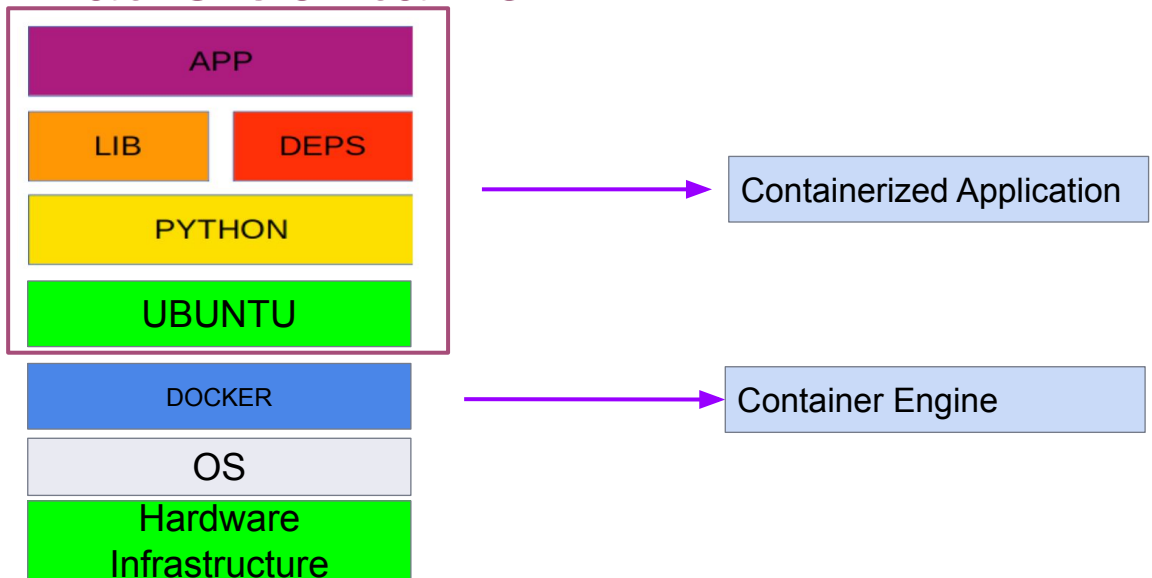


2

## Container Layers



### What is Container?





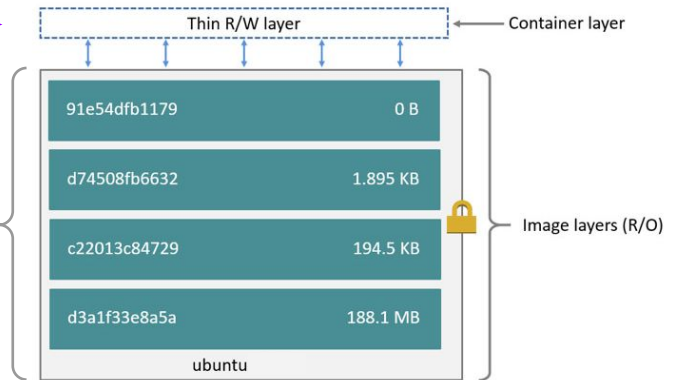
# Container Layers

- A Docker image is built up from a series of layers. Each layer represents an instruction in the image's Dockerfile. Each layer except the very last one is read-only.

```
docker run imageName
```

```
FROM ubuntu
COPY . /app
RUN apt-get install python3.6
CMD python3 /app/index.py
```

Dockerfile



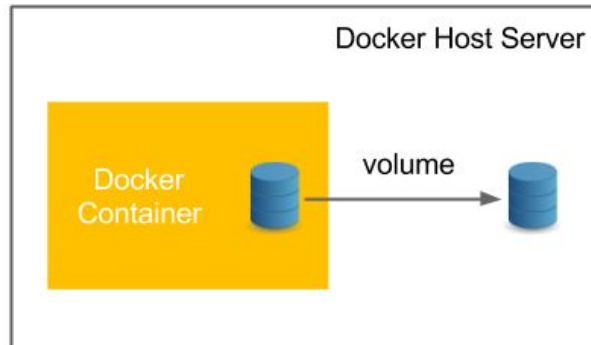
## 3

## Manage data in Docker



# Manage data in Docker

- By default, all files created inside a container are stored on a **writable container layer**. This means that the data doesn't persist when that container no longer exists.
- **Docker volumes**, which are special directories in a container, store files in the host machine so that the files are **persisted** even after the container stops.



# Manage data in Docker

Volumes are created and managed by Docker. We can create a volume explicitly using the `docker volume create` command.

```
$ docker volume create firstvolume
```



# Manage data in Docker

- When we create a volume, it is stored within a directory on the Docker host. When we mount the volume into a container, this directory is what is mounted into the container.

```
$ docker volume inspect firstvolume
[
  {
    "CreatedAt": "2020-07-12T13:19:27Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/firstvolume/_data",
    "Name": "firstvolume",
    "Options": {},
    "Scope": "local"
  }
]
```



## 4

## Declaration of volumes



## Declaration of volumes

- Volumes can be declared on the command-line, with the `--volume` or `-v` flag for docker run.
- `v` or `--volume`: Consists of three fields, separated by colon characters (:). The fields must be in the correct order.

```
--volume <volume_name>:<path>:<list of options>
```



## Declaration of volumes

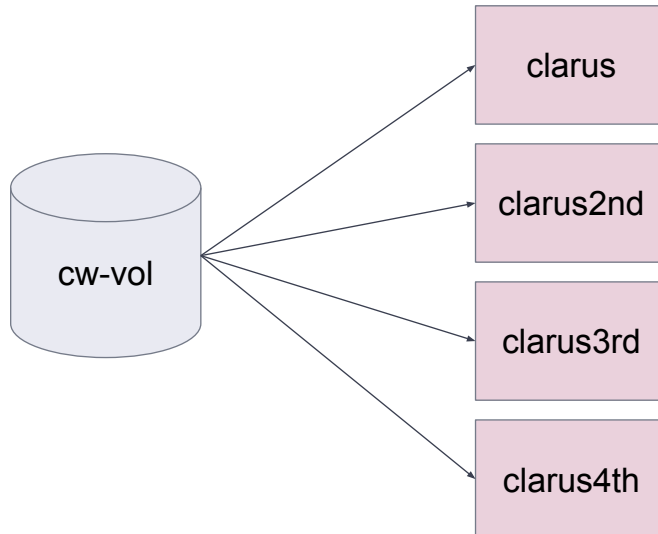
```
--volume <volume_name>:<path>:<list of options>
```

- The first field is the name of the volume, and is unique on a given host machine.
- The second field is the path where the file or directory are mounted in the container.
- The third field is optional, and is a comma-separated list of options, such as `ro` (read only).



# Declaration of volumes

We can use the same Volume with different Containers.



5

## Docker Volume Behaviours





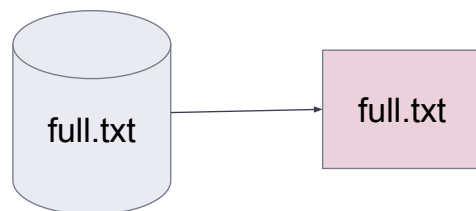
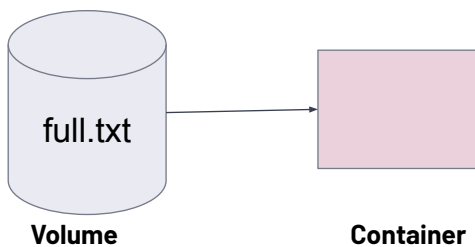
# Docker Volume Behaviours

No	Situation	Behaviour
1	If there is no target directory.	The target directory is created and files inside volume are copied to this directory.
2	If there is a target directory, but it is empty.	The files in the volume are copied to the target directory.
3	If there is a target directory and it is not empty, but volume is empty.	The files in the target directory are copied to volumes.
4	If the volume is not empty.	There will be just the files inside volume regardless of the target directory is full or empty.



# Docker Volume Behaviours

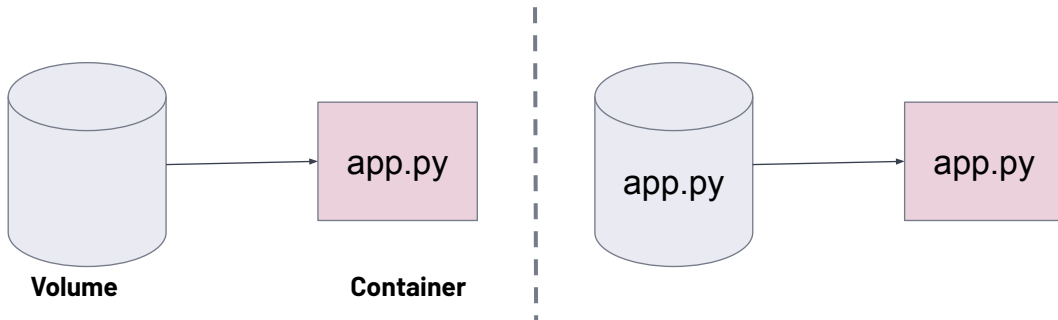
No	Situation	Behaviour
1	If there is no target directory.	The target directory is created and files inside volume are copied to this directory.
2	If there is a target directory, but it is empty.	The files in the volume are copied to the target directory.





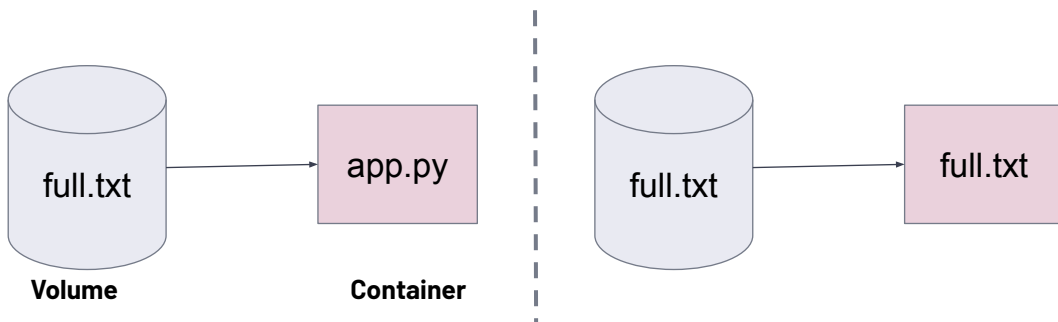
# Docker Volume Behaviours

No	Situation	Behaviour
3	If there is a target directory and it is not empty, but volume is empty.	The files in the target directory are copied to volumes.



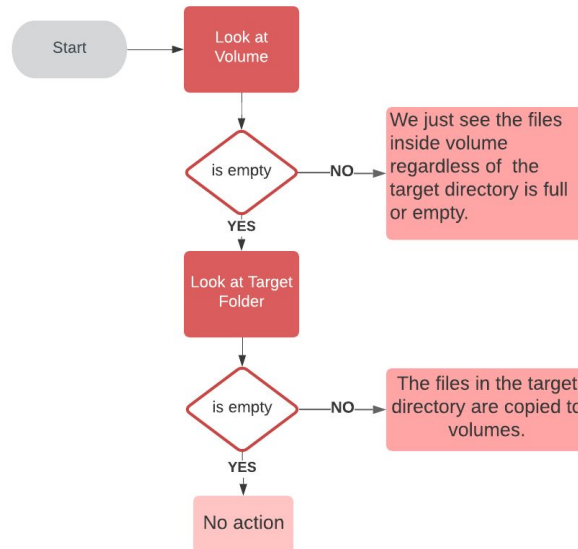
# Docker Volume Behaviours

No	Situation	Behaviour
4	If the volume is not empty.	There will be just the files inside volume regardless of the target directory is full or empty.





# Docker Volume Behaviours



## 6

# docker volume Commands



# ▶ docker volume Commands

Command	Description
<a href="#">docker volume create</a>	Create a volume
<a href="#">docker volume inspect</a>	Display detailed information on one or more volumes
<a href="#">docker volume ls</a>	List volumes
<a href="#">docker volume prune</a>	Remove all unused local volumes
<a href="#">docker volume rm</a>	Remove one or more volumes



# THANKS!

## Any questions?

You can find me at:

- ▶ alex.d@clarusway.com

