Volumes

https://docs.docker.com/engine/tutorials/dockervolumes/

```
Data volumes are designed to persist data, independent of the container’s
life cycle. Docker therefore never automatically delete volumes when you
remove a container, nor will it “garbage collect”     volumes that are no
longer referenced by a container.
A Docker data volume persists after a container is deleted.
```

Volumes types:

- Anonymous Volume: any volume without a source, docker will create this as a local volume with a long unique id, and it behaves as a named volume
- Named Volume - local storage (original image data located in specified directory are copied to volume during creation)
    - **local** driver is using `/var/lib/docker/volumes` for storage
    - original image data **is copied** if volume is empty.
    - All UID/GIDs are correctly set
    - user has to take care about free space at `/var/lib/docker/volumes`
- Host volume: bind-mounted host
    - original image data are **not copied**
    - fastest - no volume driver is used
    - possible problem with UID/GID permisions
    - volume can be created on any location (different discs, etc)
- volume plugins

Move volume between discs:

```
rsync -aqxP jenkins1_jenkins_homeSNAP/ /mnt/NVMe/@jenkins1_jenkins_home
```

BTRFS Volume plugin for Docker

https://github.com/anybox/buttervolume

Volume destination inside container must be a absolute path.

Run shell with mounted volume from another docker:

```
docker run --rm -i --volumes-from dbdata busybox ash
```

```
docker run --rm -i --volumes-from dbdata debian:jessie-slim /bin/bash
```

Single file can be mounted as volume:

```
docker run --rm -it -v ~/.bash_history:/root/.bash_history debian:jessie-
slim bash /bin/bash
```

Create named volume and share it between multiple containers:

```
docker run -d -P -v my-named-volume:/opt --name test1 debian:jessie-slim bash
docker run -d -P -v my-named-volume:/opt --name test2 debian:jessie-slim bash
docker run -d -P -v my-named-volume:/opt --name test3 debian:jessie-slim bash
```

To protect data from being deleted with volume use `local-persist` plugin: https://github.com/CWSpear/local-persist

Find orphaned volumes

```
docker volume ls -f dangling=true
docker volume rm <volume name>
```

Transfer volume to another host https://www.guidodiepen.nl/2016/05/transfer-docker-data-volume-to-another-host/

https://github.com/gdiepen/docker-convenience-scripts/blob/master/docker_get_data_volume_info.sh

data persistence in swarm

http://mysqlrelease.com/2016/08/trying-out-mysql-in-docker-swarm-mode/
https://forums.docker.com/t/data-base-persistence-in-docker-swarm-mode/20665/7

From:
https://niziak.spox.org/wiki/ - **niziak.spox.org**

Permanent link:
https://niziak.spox.org/wiki/linux:docker:volumes

Last update: **2019/04/04 09:45**