



AARHUS UNIVERSITET

Microservices and DevOps

DevOps and Container Technology

Docker Volumes

Henrik Bærbak Christensen



ReCap: Persistence

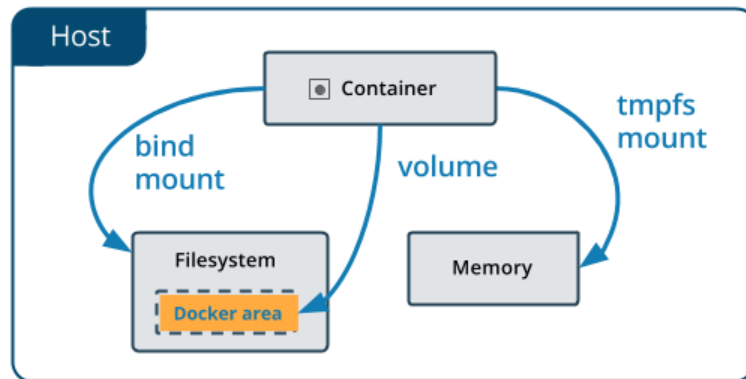
- Of course, we never install MongoDB in this course...
- We
 - `docker run -d -p 27017:27017 mongo:3.4`
- And can now contact our mongo db using
 - `localhost:27017`
- But... What happens to the data we write to it???
- Exercise: Any ideas?

You probably guessed right!

- Docker container = onion type file system
 - MongoDB writes in files in /data/db
 - ... which are in the container's file system
- ... which is simply destroyed when the container is removed...
- [Uhum, this is actually not correct...]

The Old Way

- The old way of handling this is *bind mounts*
 - Docker run ... `-v ~/my-mongo-data:/data/db` mongo:3.4
- This mounts a local folder *on host* as though it was a folder within the container
 - Net result:
 - All data is persisted on the host!



The New Way

- New way: *Named Volumes*

Volumes are the preferred mechanism for persisting data generated by and used by Docker containers. While **bind mounts** are dependent on the directory structure of the host machine, volumes are completely managed by Docker. Volumes have several advantages over bind mounts:

- Volumes are easier to back up or migrate than bind mounts.
- You can manage volumes using Docker CLI commands or the Docker API.
- Volumes work on both Linux and Windows containers.
- Volumes can be more safely shared among multiple containers.
- Volume drivers let you store volumes on remote hosts or cloud providers, to encrypt the contents of volumes, or to add other functionality.
- New volumes can have their content pre-populated by a container.

Example

- Still the folder in the container, but now just a 'volume' as source...

```
docker network create oknet
docker run --name mariadb -p 3306:3306 --network=oknet \
-e MYSQL_ROOT_PASSWORD=secret \
-e MYSQL_DATABASE=okdb \
-e MYSQL_USER=okuser \
-e MYSQL_PASSWORD=secret \
--mount 'source=mariadb,target=/var/lib/mysql' \
-d mariadb:10.4.6-bionic
```

- Now this folder is 'rewired' to some persistent area, managed by Docker

```
csdev@m51:~/proj/ok-case-study$ docker volume
DRIVER          VOLUME NAME
local           3a4766eca89b9a9c34a2471b
local           854ee7fb19ba2296aeld6f98
local           c00c0af39a76fc3618f849f0
local           e0e04af199941e775829667e
local           mariadb
```

```
csdev@m51:~/proj/ok-case-study$ docker volume inspect mariadb
[
  {
    "CreatedAt": "2019-12-13T13:55:13+01:00",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/mariadb/_data",
    "Name": "mariadb",
    "Options": null,
    "Scope": "local"
  }
]
```