Hochschule
Bonn-Rhein-Sieg

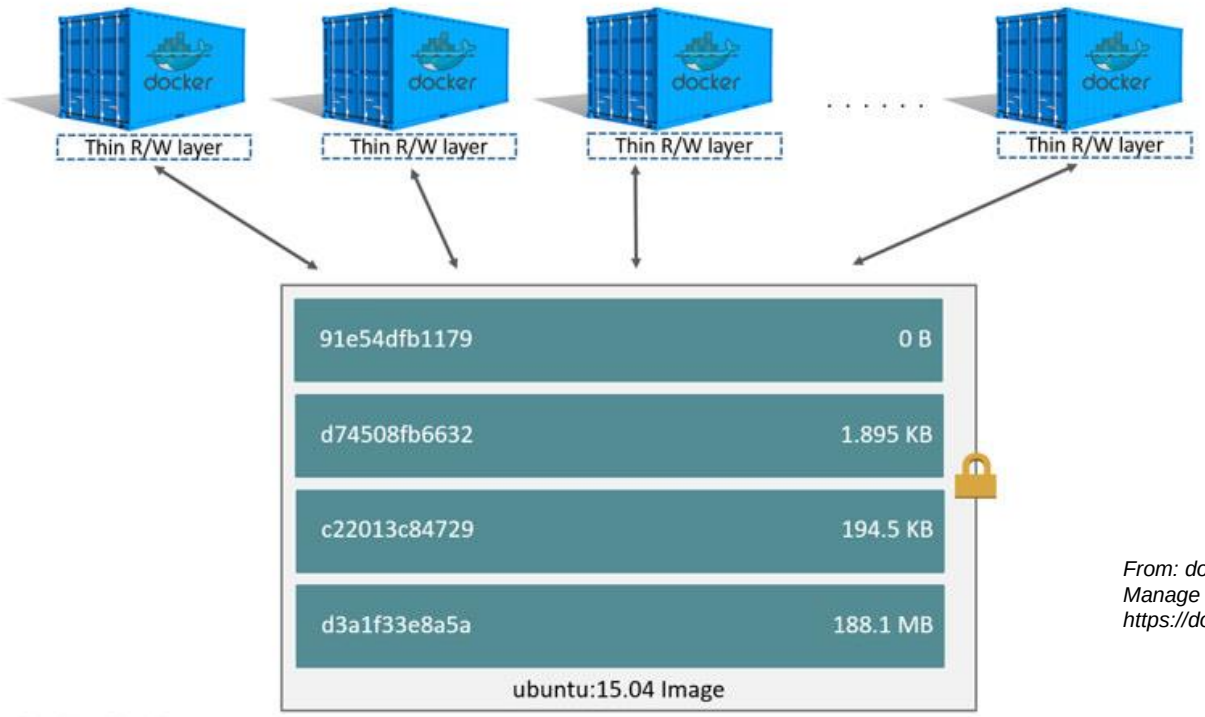
Prof. Dr. Martin Leischner
Networked Systems

Docker and Containers
Module 6: Managing Data

1. Files created inside a container
2. Volumes
3. Bind mounts
4. tmpfs mounts
5. Support of cloud storage providers

Files created inside a container

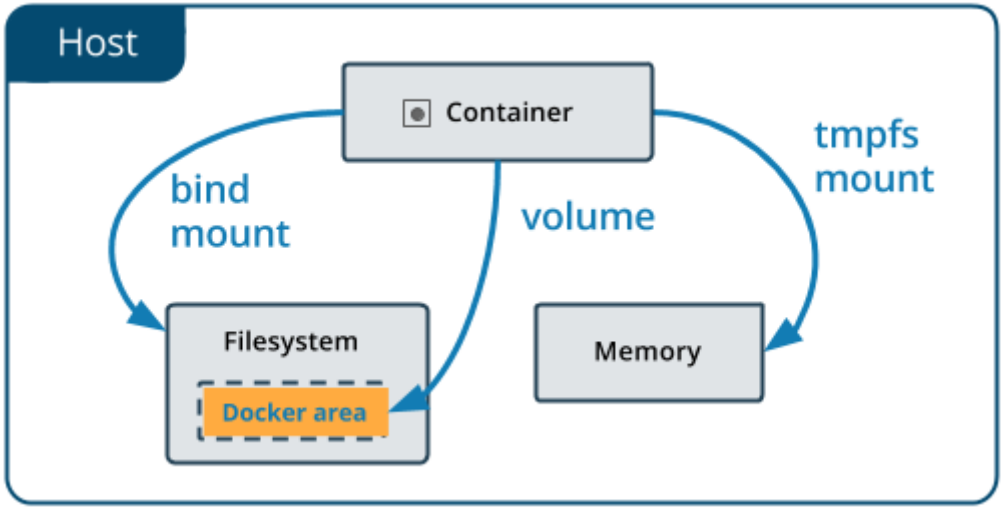
- Files created inside a container are stored on a writable container layer.
- The data doesn't persist when that container no longer exists.
- Writing into a container's writable layer requires a storage driver to manage the filesystem. This reduces performance.



From: docker docs:
Manage data in Docker,
https://docs.docker.com/storage/

Docker Volumes

- Volumes are stored in a part of the host filesystem which is managed by Docker (/var/lib/docker/volumes/ on Linux).
- Volumes are the best way to persist data in Docker.



From: docker docs:
Manage data in Docker,
https://docs.docker.com/storage/

Docker Volumes

- Volumes are relatively easy to back up.
- You can manage volumes using Docker CLI commands or the Docker API.
- The volumes are isolated from the core functionality of the host machine.
- When no running container is using a volume, the volume is still available to Docker and is not removed automatically.
- Volumes work on both Linux and Windows containers.
- Volumes can be named.
- 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.

From: docker docs: Manage data in Docker, https://docs.docker.com/storage/

Docker Volumes: CLI + Docker Compose

Command Line:

- **Create volume:** `$ docker volume create --name date-volume`
- **as run option:** `... -v data-volume:/my_data:ro ...`
- **or more explicitely and generally:**

```
... –mount 'type=volume,src=data-volume,\
                          dst=/var/my_data,rw' ...
```

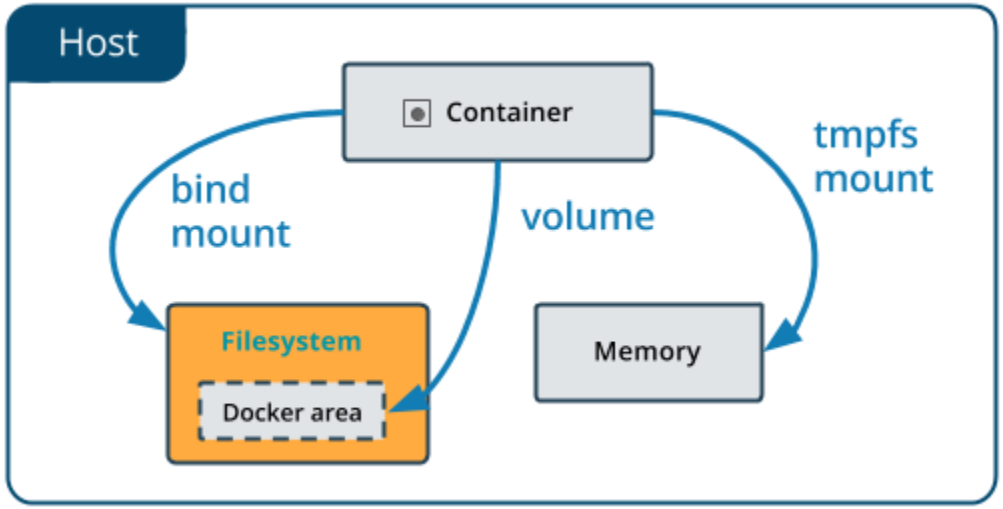
Docker Compose:

```
volumes:
   - type: volume
     source: data-volume
     target: /var/my_data
     readonly: false
   - data-volume2:/var/my_data2
   - /var/my_data3
```

From: docker docs: Manage data in Docker, https://docs.docker.com/storage/

Bind mounts

- By a bind mount a file or directory on the host machine is mounted into a container. If not exists it is created.
- If you bind-mount into a non-empty directory on the container, the directory's existing contents are obscured
- Docker does not manage that directory's contents on the host.
- Bind mounts are very performant.



From: docker docs:
Manage data in Docker,
https://docs.docker.com/storage/

Bind mounts: CLI + Docker Compose

Command Line:

- **as run option:** `... -v "$(pwd)"/host-volume:/var/my_data:ro ...`
- **or more explicitely and generally:**

```
... –mount 'type=volume,src= "$(pwd)"/host-volume,\
                          dst=/var/my_data,rw' ...
```

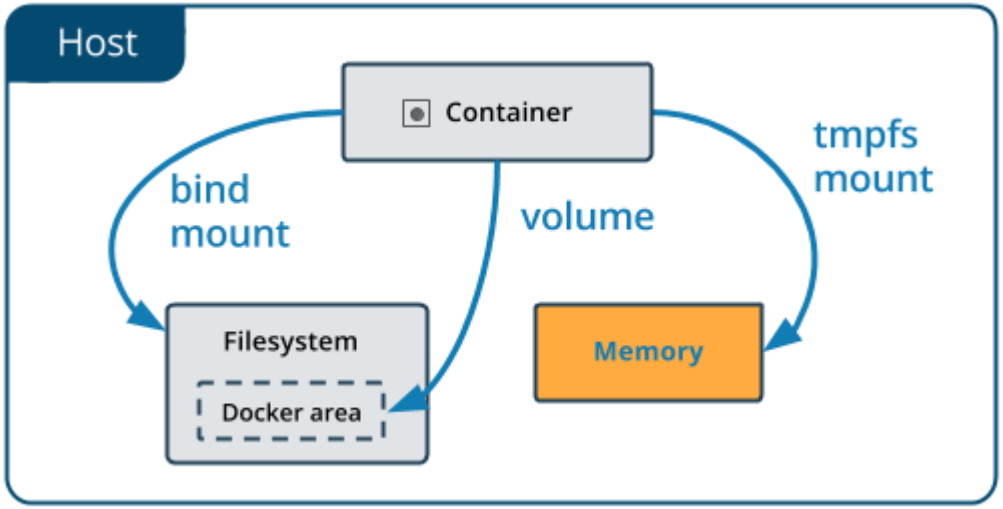
Docker Compose:

```
volumes:
   - type: bind
     source: "$(pwd)"/host-volume
     target: /var/my_data
   - "$(pwd)/host-volume2:/var/my_data2"
```

From: docker docs: Manage data in Docker, https://docs.docker.com/storage/

tmpfs mounts

- A tmpfs mount is temporary, and only persisted in the host memory.
- A tmpfs mount is useful for sensitive data that you don't want to persist in either the host or the container writable layer.
  For example to o mount secrets into a service's containers.
- A tmpfs mount is only available for Docker on Linux.



From: docker docs:
Manage data in Docker,
https://docs.docker.com/storage/

tmpfs mounts: CLI + Docker Compose

Command Line:

- **as run option:** `... --tmpfs /var/my_data ...`
- **or more explicitely and generally:**
  `... –mount 'type= tmpfs,dst=/var/my_data' ...`

Docker Compose:

```
volumes:
    - type: tmpfs
      target: /var/my_data
      tmpfs:
          size: 4096
```

From: docker docs: Manage data in Docker, https://docs.docker.com/storage/

Good use cases

Volumes:

- **Sharing data among multiple running containers.**
- **Decouple the configuration of the Docker host from the container runtime.**
- **When you want to store your container's data on a remote host or a cloud provider, rather than locally.**

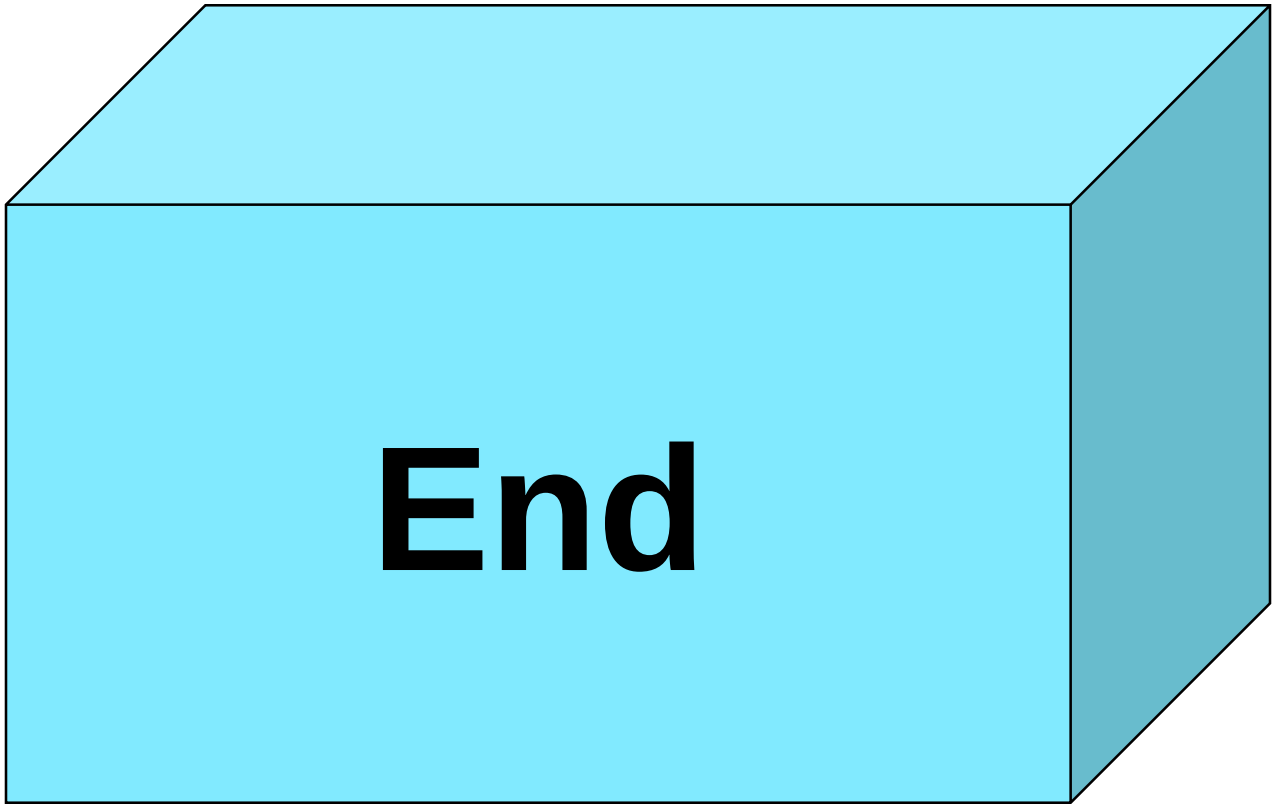
Bind mounts:

- **Sharing configuration files from the host machine to containers.**
- **Sharing source code between a development environment on the Docker host and a container.**
- **When the file or directory structure of the Docker host is guaranteed to be consistent with the bind mounts the containers require.**

tmpfs mounts:

- **Security reasons or to protect the performance of the container.**

From: docker docs: Manage data in Docker, https://docs.docker.com/storage/


End