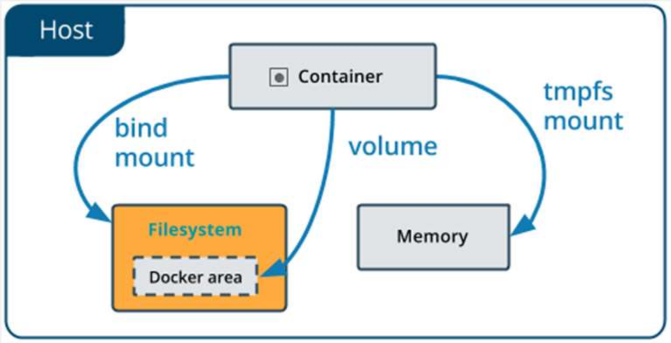
CLOUD
COMPUTING
APPLICATIONS

Docker Swarm

Prof. Reza Farivar

Data Volumes

- Recall that Docker containers are based on Unionfs
  - Multiple immutable (read-only) base layers
  - One read-write container-specific layer
- When a container is removed, the top layer is also removed
- To persist changes, and to access data outside the container, we need to mount an external storage location
- Three types of host to container mapping
  - Bind mount
  - Volume
  - tmpfs

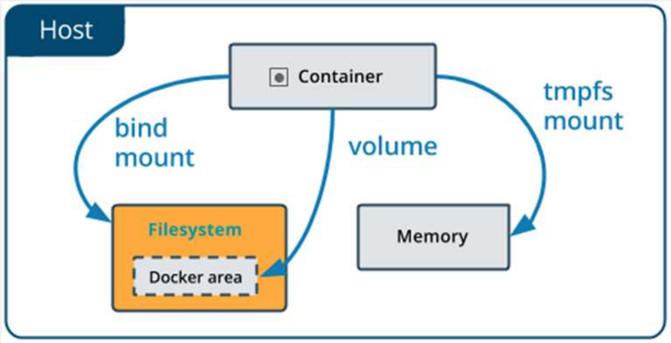


tmpfs

- tmpfs mounts are best used for cases when you do not want the data to persist either on the host machine or within the container
  - for security reasons
  - to protect the performance of the container when your application needs to write a large volume of non-persistent state data

Persistent Data Storage: Bind Mount

- When you use a bind mount, a file or directory on the *host machine* is mounted into a container
- The file or directory is referenced by its absolute path on the host machine



Docker Volume

- Persistent storage ***abstraction***
- Managed by Docker
- Will last after the container is removed
- Different drivers
  - Volume drivers let you store volumes on remote hosts or cloud providers, to encrypt the contents of volumes, or to add other functionality.
  - For local deployments, usually `local` driver
- For distributed applications (swarm)
  - Use an NFS and use `local` driver
  - Some drivers support writing files to an external storage system like NFS or Amazon S3
    - REX-Ray, CloudStor
    - `vieux/sshfs` volume drive

Mount Volume examples

- `$docker volume create my-vol`
- `$docker volume ls`
- `$docker volume inspect my-vol`
- `$docker volume rm my-vol`

-
  ```
  $ docker run -d \
      --name devtest \
      --mount source=my-vol,target=/app \
      nginx:latest
  ```