

Intermediate Docker Commands

INTERMEDIATE DOCKER



Mike Metzger

Data Engineering Consultant

Docker refresher

- Docker is a container runtime
- Is designed to run and manage various containerized applications on Windows, Mac, and Linux
- Can run containers using pre-built images, or create our own
- Dockerfiles are text files used to Docker container images
- Containers are instances of a given Docker image

The DataCamp [Introduction to Docker course](#) is a pre-requisite to this course



INTERACTIVE COURSE

Introduction to Docker

[Continue Course](#) [Bookmark](#)

●● Intermediate ⌚ 4 hours ▶ 15 videos <> 56 exercises

Docker commands

- `docker run`
- `docker stop`
- `docker build`
- `docker --help`
 - Provides a list of potential Docker commands
- `docker COMMAND --help`
 - `docker run --help`
 - Provides options for the `docker run` command

Temporary containers

- Docker containers are usually created with `docker run`
- Containers remain even after stopping / exiting
- Often want to run a container instance and remove it immediately upon exit
 - Development
 - Testing
 - Scripts
- `docker run --rm`
 - `docker run --rm alpine:latest`
 - `/bin/sh`
- Referenced as 'clean-up' or 'remove'

docker ps

- Used for determining name, id, status, and other attributes of containers on a given machine running Docker
- Use the `-a` flag to get more information about existing containers
 - `docker ps -a`
- Will cover how to get extremely detailed information about containers later in the course

```
$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
588bb3a3134e	nginx	"/docker-entrypoint..."	49 seconds ago	Up 47 seconds
8a0578a9b33c	ubuntu	"/bin/bash"	About a minute ago	Exited (0) About a minute ago

Let's practice!
INTERMEDIATE DOCKER

Mounting the host filesystem

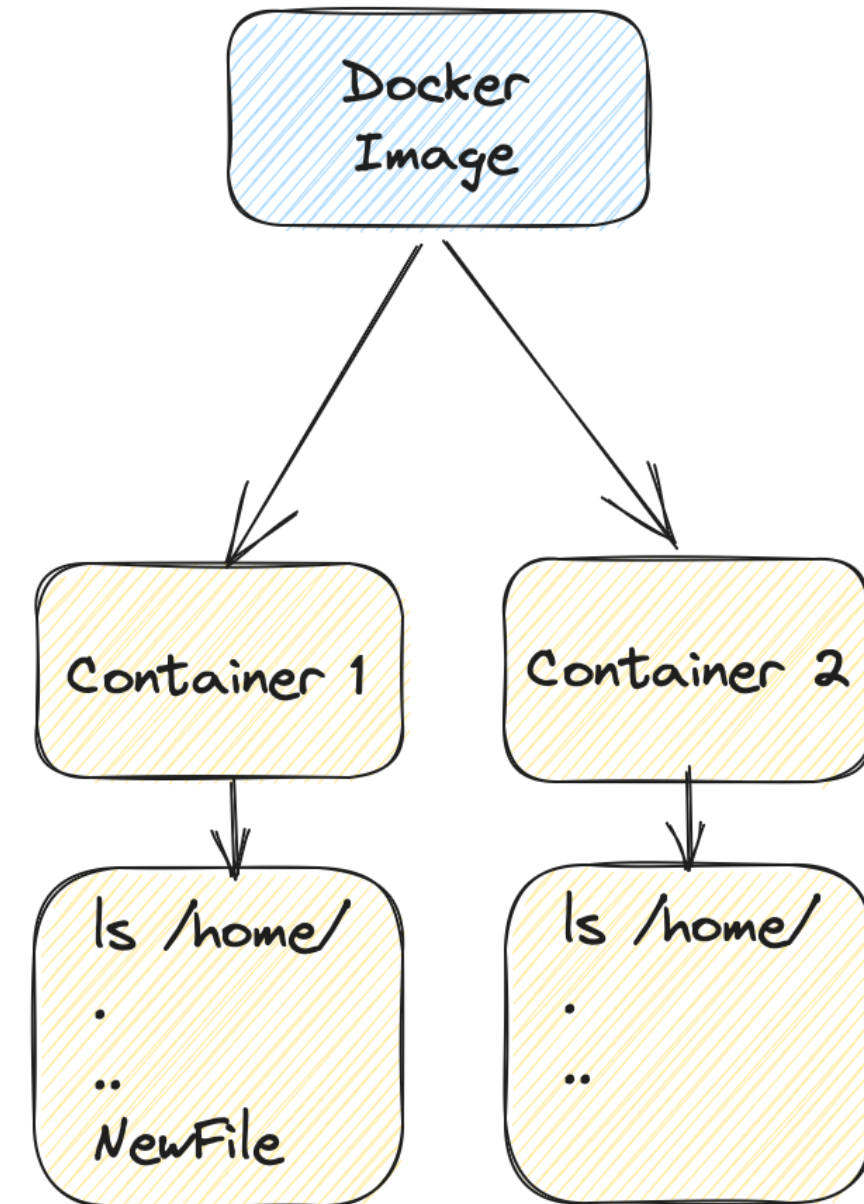
INTERMEDIATE DOCKER



Mike Metzger
Data Engineer

Container filesystems

- Container instances each have their own filesystem
 - Based off the image the container was created with
- Any changes are tied to that specific container instance
- Any changes are maintained across restarts
 - For that instance **only**
- New containers only have the data in the image, not instance specific changes



Sharing files or directories

- Can attach specific files or directories to containers
- Allows for persistence of data, without maintaining a specific container
- Can upgrade container to new version but safely keep data / changes
- Known as **bind-mount**
- Can be read-only or read/write
- Note: When files or directories are attached to a container, they are not accessible to the host until the container is shutdown

Using the -v option

- bind-mounts most often use the `-v` flag
- `-v <source>:<destination>`
- Multiple `-v` commands permitted
- Can also use the `--mount` option
- Note: bind-mount **hides** any content already present in the destination directory

```
docker run -v ~/html:/var/www/html \  
    nginx
```

```
docker run \  
    -v ~/pgdata:/opt/data \  
    -v ~/pg.conf:/etc/pg.conf \  
    postgresql
```

Let's practice!
INTERMEDIATE DOCKER

Persistent volumes

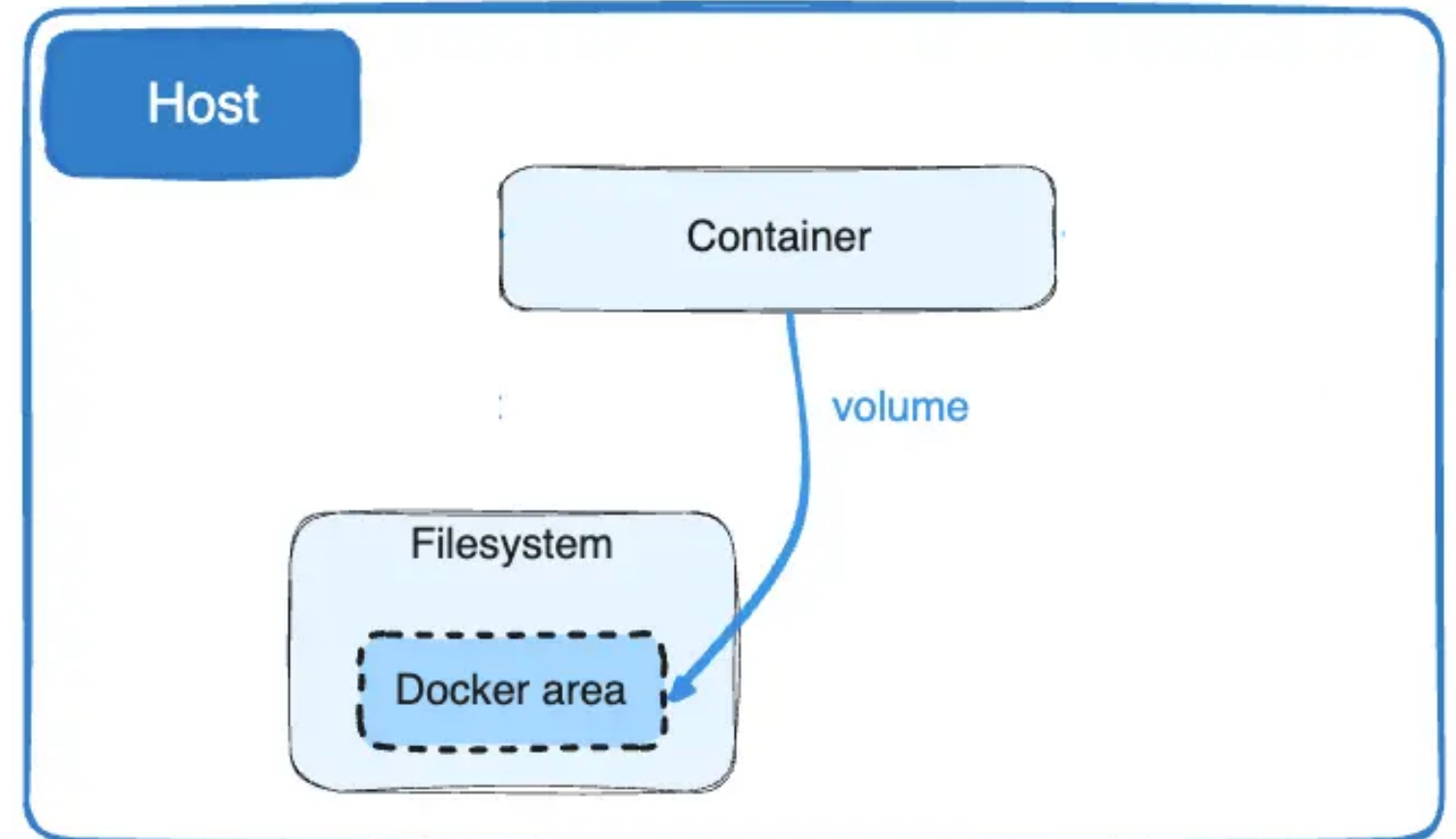
INTERMEDIATE DOCKER



Mike Metzger
Data Engineering Consultant

What is a volume?

- **Volumes** are an option to store data in Docker, unrelated to the container image or host filesystem
- Are managed from the command line (or API)
- Can share with multiple containers
- Higher performance than file share / bind mounts
- Exist until removed



¹ Image modified from <https://docs.docker.com>

Managing volumes

- `docker volume`
- `docker volume create <volumename>`
- `docker volume ls` or
`docker volume list`
- `docker volume inspect`
 - Provides assorted metadata about the volume, including Name, Mountpoint, Options, and so forth
- `docker volume rm`

Volume creation example

```
bash> docker volume create sqldata
```

```
sqldata
```

```
bash> docker volume ls
```

DRIVER	VOLUME NAME
local	2f2b7f710551e004dcdd9edf4cad31c37826b428de12f1c04ca02305d216ab00
local	14da7ff0c6eb29f644e6f9f9d59bbcf56b3699c04881dd7cbcaa9ecd6bef239c
local	150aa3c5c7aee30ffd1ec7ecf39f03989bf561536a9413ebed96ffbaa537d103
local	sqldata
...	

Volume inspect example

```
bash> volume inspect sqldata
[
  {
    "CreatedAt": "2024-01-27T04:27:51Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/sqldata/_data",
    "Name": "sqldata",
    "Options": null,
    "Scope": "local"
  }
]
```


Attaching volumes

- Uses the `-v` command
 - `docker run -v <volumename>:
<destination path>:<options>`
 - Volume name is name of existing volume
 - Destination path is the location the volume will be mounted (such as `/data`)
 - Options are optional comma-separated list of values such as `ro` for read-only.
- `--mount` exists as with bind-mounts

```
$ docker run -v sqldata:/data postgres
```

Drivers

- Methods of storing Docker volumes
- Can include:
 - Local filesystem (default)
 - NFS (Unix filesharing)
 - SMB / CIFS (Windows filesharing)

Let's practice!
INTERMEDIATE DOCKER