DATASTAX®
DataStax Monday Learning

DataStax Monday Learning

Upgrade yourself, unlock new skills

- Every Week
- Best Instructors
- Most Important Topics
- From Engineers to Engineers
- Absolutely Free

DATASTAX®

DataStax Monday Learning

Docker Containers

From Basics to Best Practices

4-weeks Learning Path: 28.09-23.10.2020

Speakers:

- Aleks Volochnev
- Developer Advocates of DataStax

Schedule:

- **Week I 28.09.2020** Docker Fundamentals I
- **Week II 05.10.2020** Docker Fundamentals II
- **Week III 12.10.2020** Application Development with Docker
- **Week IV 19.10.2020** Best Practices + Final Assignment

DataStax Monday Learning

Docker Containers

From Basics to Best Practices

- Over 1 thousand of registrations
- More than 3,5K views on Youtube
- People from 25 countries
- 1,000 HOURS overall watch time

Thank you!

DATASTAX
DataStax Monday Learning

Docker Containers
From Basics to Best Practices

Week II

Docker Fundamentals II

3 Questions to know you better

DATA

Bind Mounts

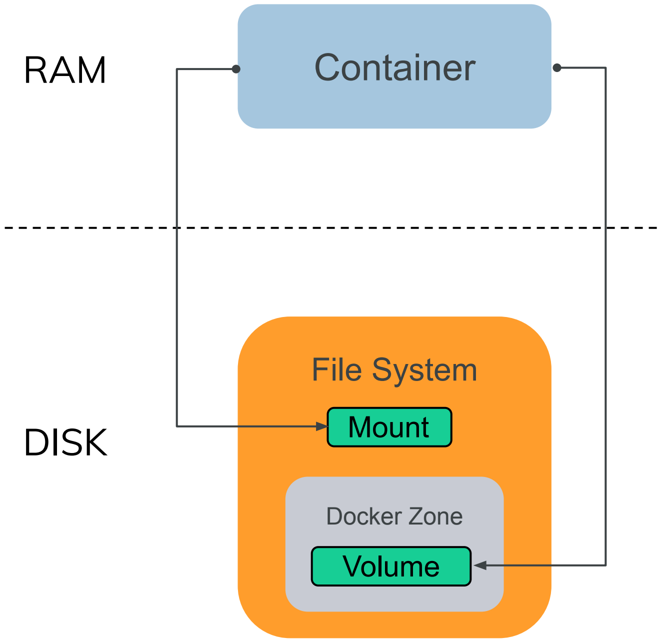
The simple and "old-school" way to mount a local folder or a file into the container file system. Have limited functionality but usually enough for most of the use-cases. Allows direct access to the files from both host and container, very often used for development purposes. Default access mode is RW (read-write) but can be configured to be RO (read-only). **Managed by user,** so you can't use Docker CLI commands to directly manage bind mounts. **Host mount precedes container mount.** Prefer for the cases when data comes from host.

```
docker run --volume LOCAL-PATH:CONTAINER-PATH
```

```
docker run -v /home/anna/project:/app:ro
```

```
docker run -v $(pwd):/opt/project
```

```
docker run --mount type=bind,source="$(pwd)",target=/opt/project
```



Volumes

Another way to handle persistent data. The main idea is still the same: mount a local folder into the container file system, but this time is less direct and more "docker-native" way. Direct access to the files from host is a bit complicated. Very often used for cases when data is created by a container. **Managed by docker,** so you can use Docker CLI commands to manage volumes. **Container Data used to fill volume on creation.** Prefer for the cases when data comes from a container.

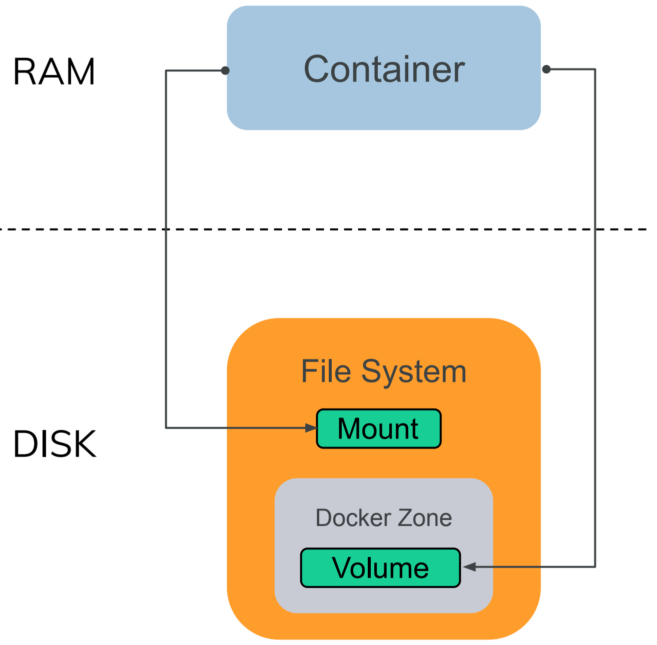
```
docker run --volume VOLUME-NAME:CONTAINER-PATH
```

```
docker run -v project-data:/app
```

```
docker run --mount source=mysql-data,target=/var/lib/mysql
```

Docker commands:

- docker volume create NAME
- docker volume ls
- docker volume inspect NAME
- docker volume rm NAME



Live Demo I: Persistent Data

We cover three cases:

● mysql, no mount, no volume	**data lost**
● mysql, mount	**data kept, local folder**
● mysql, volume	**data kept, docker volume**

```
docker run -d -e MYSQL_ALLOW_EMPTY_PASSWORD=true mysql
```

```
docker run -d -e MYSQL_ALLOW_EMPTY_PASSWORD=true -v $(pwd)/mysql-data:/var/lib/mysql mysql
```

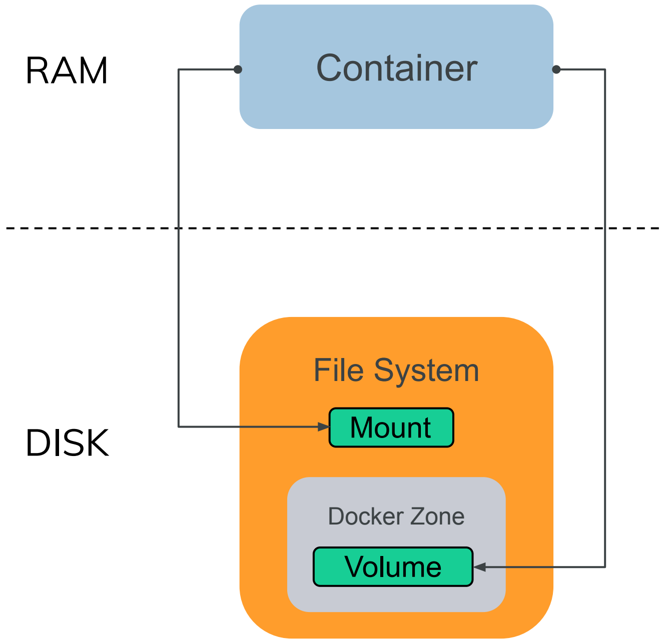
```
docker run -d -e MYSQL_ALLOW_EMPTY_PASSWORD=true -v mysql-data:/var/lib/mysql mysql
```

Attach to the container, create a database, list databases.

```
docker exec -it CONTAINER_ID mysql
```

```
create database DB_NAME; show databases; exit
```

Delete the container.



Multiple Containers

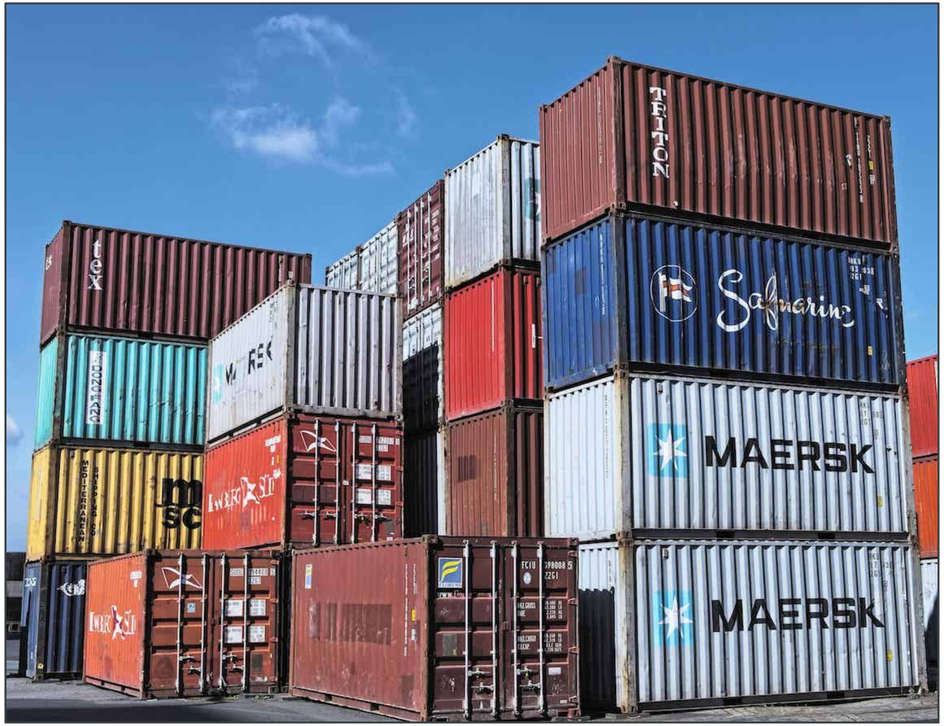
Multiple Containers

Containers may collaborate directly **as long as they are in the same network.** Let's cover some network types first:

- Bridge [default, single host, DNS]
- Host [only linux, direct attach]
- Overlay [multiple hosts]

Working with networks:

- docker network ls
- docker network create NAME
- docker network inspect NAME
- docker network rm NAME
- docker network connect NET CONT



Live Demo II: Multiple Containers

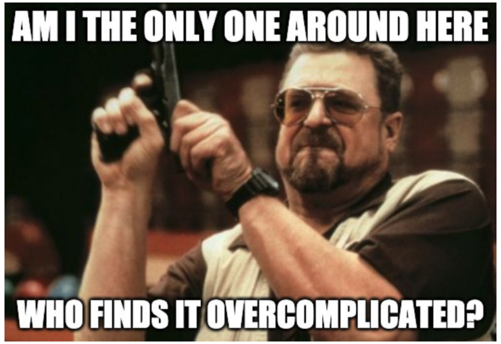
Two things for us to investigate:

- Play with networks
- Connect two containers

```
docker network create wp --driver bridge
```

```
docker run --detach -v mysql_data:/var/lib/mysql --network wp --name database
-e MYSQL_ROOT_PASSWORD=secretpassword -e MYSQL_DATABASE=wordpress -e MYSQL_USER=wordpress -e MYSQL_PASSWORD=wordpress
mysql:5
```

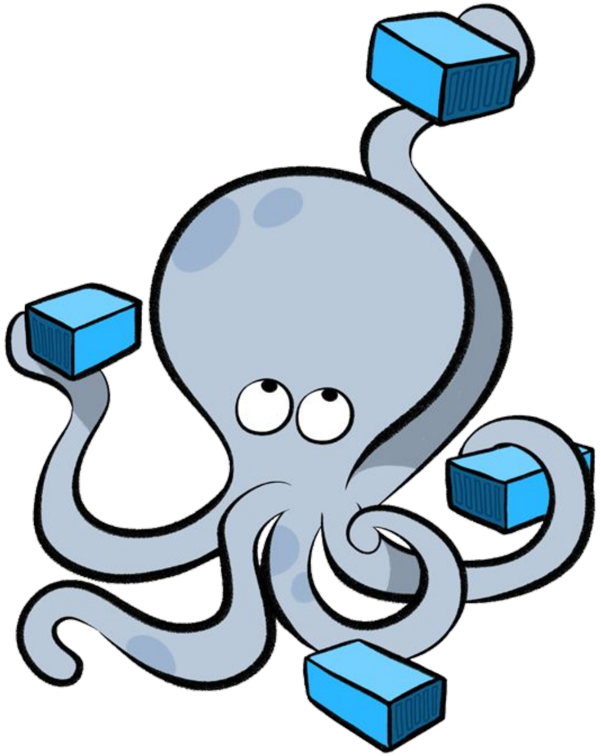
```
docker run -p 80:80 -v wp-data:/var/www/html --network wp --detach
-e WORDPRESS_DB_HOST=database:3306 -e WORDPRESS_DB_USER=wordpress -e WORDPRESS_DB_PASSWORD=wordpress -e WORDPRESS_DB_NAME=wordpress
wordpress:latest
```



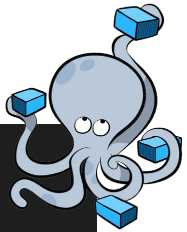
I’m too lazy to type all the commands…

Docker-Compose WAN LUV ❤️

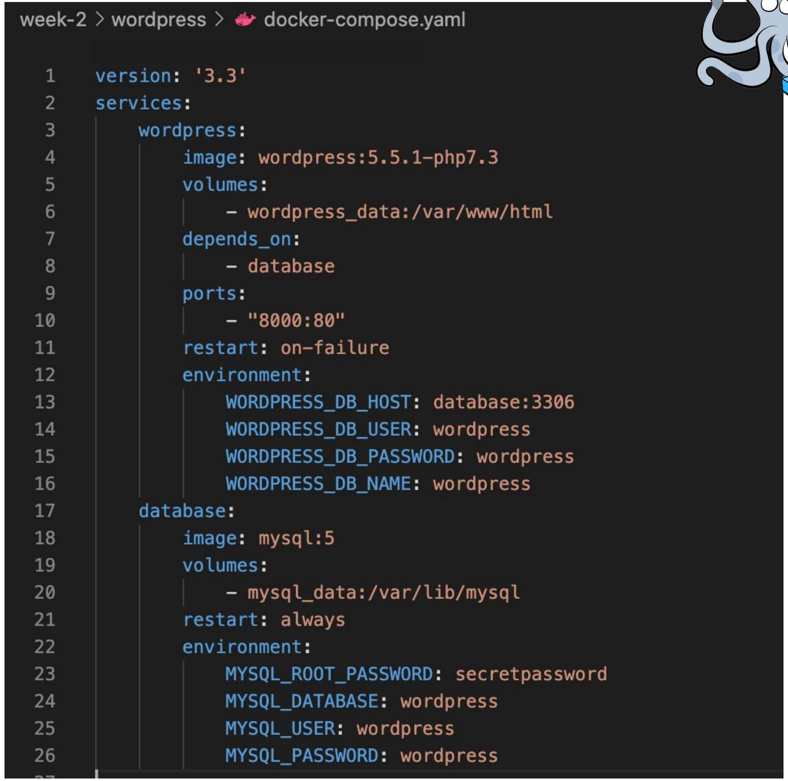
Docker Compose is a powerful tool behind the simple idea: Infrastructure as a Code. Instead of typing commands all day, describe required setup in a configuration file and let docker-compose do the work for you. Technically speaking, it’s just a python-based wrapper which converts yaml config file into docker console commands.



Docker-Compose Configuration



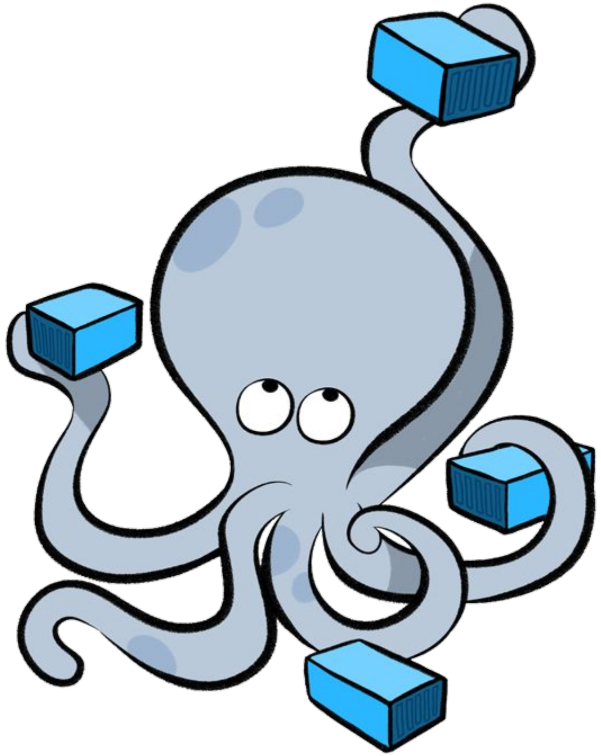
It uses configuration options exactly like those you use in `docker run`: publish, environment, volumes, and does some more job like takes care about networks. Now you don't have to copy-paste commands from your projects docs but keep it all in docker-compose.yaml file.



week-2 > wordpress > docker-compose.yaml

```yaml
version: '3.3'
services:
    wordpress:
        image: wordpress:5.5.1-php7.3
        volumes:
            - wordpress_data:/var/www/html
        depends_on:
            - database
        ports:
            - "8000:80"
        restart: on-failure
        environment:
            WORDPRESS_DB_HOST: database:3306
            WORDPRESS_DB_USER: wordpress
            WORDPRESS_DB_PASSWORD: wordpress
            WORDPRESS_DB_NAME: wordpress
    database:
        image: mysql:5
        volumes:
            - mysql_data:/var/lib/mysql
        restart: always
        environment:
            MYSQL_ROOT_PASSWORD: secretpassword
            MYSQL_DATABASE: wordpress
            MYSQL_USER: wordpress
            MYSQL_PASSWORD: wordpress
```

Live Demo III: Docker-Compose



Let’s run the application from Live Demo II using docker-compose.

```
docker-compose up -d
```

Isn’t that incredible?!

LIVE QUIZ!

Week II Assignment

Week II Assignment

1. This time you have to do a more complex setup using docker-compose. It should include at least one predefined image from hub.docker.com (like a database) and one custom image you build on your own. It should use bind mounts or volumes so `docker-compose down` will not wipe out data. Please use our examples of wordpress and voting application for the inspiration.
2. If possible, publish your code from p.1 on github by creating a new issue at github.com/datastaxdevs/docker-learning-path/issues. It may not be an option if you containerised a proprietary project, but please proceed to step III anyway.
3. Open the issues list from p.2, pick one not taken project, write a comment that you have "taken" it. Review the project and think on how would you improve it. Write down your suggestions in the issue. Feel free to review multiple projects, also feel free to review a taken one - more opinions is better! **Stay polite!**
4. If you want us to review your assignment publicly, send an issue link to me as well! We will pick some projects to discuss during week III. We cover both mistakes and good decisions. :)
5. Add me on linkedin. We spend together over 4 hours already so let's celebrate it! linkedin.com/in/volochnev/

Resources:

- https://github.com/datastaxdevs/docker-learning-path
- https://discord.gg/va4vnsm

DATASTAX®

Thank You!

You are awesome!