

1. Ziele

- Verständnis von Volumes (benannt und gemountet)

2. Aufgaben

Der mariadb-Server verwendet standardmässig den Port 3306. Seine Datenbanken speichert er in /var/lib/mysql. Dies ist natürlich auch beim offiziellen Image der Fall.

Nun soll folgendes Szenario aufgebaut werden:

- Starten Sie zwei Container aus dem mariadb-Image und verbinden Sie die Containerports mit den zwei Hostports 3306 und 3307. Das Rootpassword kann mit dem Parameter -e angegeben werden:
-e MYSQL_ROOT_PASSWORD=geheim

```
docker run -d --name mariaImage1 -p 3307:3306 -e MYSQL_ROOT_PASSWORD=geheim -v myVolume1:/var/lib/mysql mariadb:latest
```

- Verwenden Sie im ersten Container ein benanntes Volumen

siehe oben -v

- Verwenden Sie im zweiten Container ein Volumen, welches auf dem Hostrechner in /home/vmadmin/mysql liegt.

```
docker run -d --name mariaImage2 -p 3306:3306 -e MYSQL_ROOT_PASSWORD=geheim -v /home/vmadmin/mysql:/var/lib/mysql mariadb:latest
```

- Lassen Sie sich in beiden Fällen den Inhalt des Volumes mit `ls -l` anzeigen

```
docker inspect -f '{{.Mounts}}' mariaImage1
```

- Installieren Sie auf dem Host den mysql-client mit `sudo apt install mariadb-client` (evtl. zuerst `sudo apt update`)

-

- Verbinden Sie sich bei beiden Containern mit
`mysql -u root -p -P 3306 -h 127.0.0.1`
`mysql -u root -p -P 3307 -h 127.0.0.1`
und legen Sie je eine neue Datenbank an. (-h 127.0.0.1 ist nötig um TCP zu erzwingen)

```
create database {name}  
show databases
```

[illegible]

- Stoppen und **löschen** Sie nun die Container (Simulation eines Upgrades vom mariadb)

```
docker stop marImage1
```

```
docker rm marialimage1
```

- Erstellen Sie zwei neue Container mit den selben Volumes und Ports wie vorher.

```
docker run -d --name mariaImage1 -p 3307:3306 -e MYSQL_ROOT_PASSWORD=geheim -v myVolume1:/var/lib/mysql mariadb:latest
```

- Überprüfen Sie ob die zuvor erstellten Datenbanken noch vorhanden sind.

```
mysql -u root -p -P 3306 -h 127.0.0.1
```

```
show databases;
```

- Löschen Sie alle Container, Images **und** Volumes

```
docker stopp ...
```

docker rm

```
docker rmi ...
```

```
docker volume rm (name)
```

3. Hilfsmittel

<https://gbssg.gitlab.io/m347/docker-volumes/>

4. Erwartete Resultate

Mit Screenshots dokumentiertes und kommentiertes Vorgehen

Zeit: 45 Minuten