

1. Ziele

- Ein erstes funktionierendes docker-compose Beispiel realisieren

2. Aufgaben

1. Überführen Sie folgendes docker run Kommando für einen mariadb-Container in eine Docker compose Anwendung:

```
docker run -d --name mariadb-container \  
-e MYSQL_ROOT_PASSWORD=geheim \  
-p 3306:3306 \  
-v mydata:/var/lib/mysql \  
mariadb:latest
```

Wechseln Sie dazu in ein Arbeitsverzeichnis und erstellen Sie darin eine Datei mit Namen docker-compose.yml Verwenden Sie die folgende Vorlage und passen Sie diese gemäss dem docker run an:

```
version: "3.8"

services:
  mariadb:
    image:
    container_name:
    volumes:

    environment:

    ports:

volumes:
```

version: "3.8"
services:
 mariadb:
 image: mariadb:latest
 container_name: mariadb-container
 environment:
 MYSQL_ROOT_PASSWORD: geheim
 ports:
 - "3306:3306"
 volumes:
 - mydata:/var/lib/mysql

2. Testen Sie die Anwendung mit

```
docker compose up
```

Beachten Sie den Output. Wenn keine Fehler sichtbar werden, können Sie den container mit ctrl + C wieder stoppen. Der Container ist nach wie vor vorhanden wie ein

```
docker ps -a
```

zeigt. Mit

```
docker compose down
```

wird der Container gelöscht und auch das automatisch erstellte Netzwerk entfernt.

3. Starten Sie nun die Anwendung erneut mit:

```
docker compose up -d
```

Der Output ist nun nicht sichtbar, da der Container im Hintergrund gestartet wird. Der Output kann jedoch mit

```
docker logs mariadb-container
```

untersucht werden. Verbinden Sie sich mit der Datenbank mit:

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

4. Beenden Sie die Anwendung wieder mit

```
docker compose down
```

3. Hilfsmittel

<https://gbssg.gitlab.io/m347/>

Internet

4. Erwartete Resultate

Mit Screenshots dokumentiertes und kommentiertes Vorgehen (Word) Zeit: 30 Minuten