

# MySQL Server

Installation und Konfiguration eines MySQL Servers.

- Installation
- Log für Abfragen

# Installation

Im folgenden wird die Installation eines MySQL Servers mit Docker gezeigt.

Für einen SQL Server mit persistentem Speicher ist folgende Docker Compose Datei geeignet.

```
services:
  mysql:
    image: mysql
    environment:
      - MYSQL_ROOT_PASSWORD=Ein-langes-Passwort-mit-619-Zahlen
    volumes:
      - /pfad/zu/mysql/data:/var/lib/mysql
    ports:
      - 3306:3306
    restart: unless-stopped
```

Mit `MYSQL_ROOT_PASSWORD` wird ein Passwort für den wichtigsten und zur Zeit einzigen Benutzer `root` gesetzt. Hierdurch kann eine Verbindung mit den Logindaten `root` und `Ein-langes-Passwort-mit-619-Zahlen` mit dem Server aufgebaut werden z. B. über die MySQL Workbench.

Unter `volumes` wird der Ordner, in welchem MySQL die Daten speichert, auf den Pfad `/pfad/zu/mysql/data` umgeleitet. Damit werden die Daten des MySQL Servers nicht gelöscht, wenn der Container gelöscht oder neu erstellt wird. Der Pfad ist an die eigene Umgebung anzupassen.

Die Freigabe des Containers unter dem Port 3306 wird unter `ports` konfiguriert. Somit kann der Container auch von anderen Servern erreicht werden, die Zugriff auf den Docker Host haben. Dies ist nur erforderlich, wenn nicht ausschließlich Dienst vom Docker Host auf die Datenbank zugreifen sollen.

Um die Docker Compose Konfiguration auszuführen, kann am besten in das Verzeichnis der YAML Datei gewechselt werden. Danach wird je nach gewählter Installation `sudo docker-compose up` oder `sudo docker compose up -d` (keine Bindestrich zwischen docker und compose) eingegeben, um die Standard Konfiguration `docker-compose.yml` zu starten. Compose erstellt dann die gewünschten Container mit den angegebenen Optionen. Sollten die Container bereits mit dieser Compose Konfiguration erstellt worden sein, so werden die Container in dieser neu erstellt, dessen Konfiguration geändert wurde.

# Log für Abfragen

Für das Logging von SQL-Abfragen können auf einem MySQL-Server verschiedene Optionen verwendet werden. Zum einen gibt es ein allgemeines Log von allen Abfragen und eines für langsame Abfragen.

Standardmäßig ist das Log von Abfragen deaktiviert. Das Log der Abfragen kostet Performance, wenn also ein Server sowieso schon stark ausgelastet ist, sollte das Log nur bei Bedarf aktiviert werden.

## Alle Abfragen

Um alle Abfragen zu erfassen, können die folgenden SQL Befehle verwendet werden.

```
SET GLOBAL general_log = 'OFF';  
SET GLOBAL general_log_file = '/pfad/zum/log/dateiname.log';  
SET GLOBAL general_log = 'ON';
```

Wichtig ist hierbei, dass der Pfad der Logdatei nur angepasst werden kann, wenn das Log deaktiviert wird. Somit wird pauschal das Log erstmal deaktiviert, der Pfad geändert und dann aktiviert.

Die Änderungen werden ohne Neustart des Servers direkt angewendet.

Um das Log wieder zu deaktivieren, einfach nur die 1. Zeile ausführen.

Anschließend kann das Log über die angegebene Log-Datei eingesehen werden.

## Langsame Abfragen

Um nur langsame Abfragen zu erfassen, können die folgenden SQL Befehle verwendet werden. Der Vorteil hierbei ist, dass der Performance Verlust und der Speicherplatzverbrauch geringer ist. Auch die Übersicht wird verbessert, da nur wirklich relevante Abfragen protokolliert werden, aus Sicht der Performancesteigerung.

```
SET GLOBAL slow_query_log = 'OFF';  
SET GLOBAL slow_query_log_file = '/pfad/zum/log/dateiname.log';  
SET GLOBAL slow_query_time = 5;
```

```
SET GLOBAL slow_query_log = 'ON';
```

3 Zeilen sollten von den Befehlen für das allgemeine Log bekannt sein, sie sind nur etwas anders benannt (slow\_query). Neu ist der Parameter für die Zeit, ab wann ein Befehl als langsam gilt. Je nach Umgebung können einige Sekunden ggf. normal sein, in anderen wäre alles über 0.5s langsam. Die Zahl gibt die Zeit in Sekunden an, also wäre 1 gleich 1 Sekunde, auch Bruchteile sind zulässig, wie z. B. 0.25 für eine Viertelsekunde.