

SQL Server & Docker: Volume Backup und Wiederherstellung

Kategorie
Docker, SQL Server

In diesem Artikel möchten wir das Erstellen und Wiederherstellen des Backups eines Docker Volumens das von einer SQL Server Installation genutzt wird, betrachten. Stellen wir uns zunächst folgende Ausgangssituation vor:

Auf einem Docker Container läuft ein MS SQL Server 2017, der auf einem Volumen benötigte Daten gespeichert hat. Dieses Volumen, oder viel mehr die Daten auf diesem Volumen, sind persistent. Das bedeutet, dass auch nach einem Neustart des Containers die gespeicherten Informationen noch vorhanden sind.

Tipp: Sollen die gespeicherten Daten nach einem Neustart verschwinden, kann auf folgende Weise die "-v" Flagge genutzt werden: **docker-compose down -v**.

Aaron
Priesterroth

Hier die zur Situation zugehörige **docker-compose.yml** Datei:

```
services:
  sql:
    image: mcr.microsoft.com/mssql/server:2017-GA-ubuntu
    volumes:
      - sqldata:/var/opt/mssql
      ...
volumes:
  sqldata:
    driver: local
    name: sqldata
```

Bei den auf dem Volumen gespeicherten Daten handelt es sich in unserem Fall um die Datenbank-Dateien des SQL Servers. Diese können wir sichern und ein Backup erstellen, indem wir SQL Server zur Hilfe nehmen: mit den folgenden Befehlen können wir ein Backup der Datenbank innerhalb des Containers erzeugen, eine lokale Kopie erstellen und dieses wiederherstellen:

```
# Create backup of database and copy .bak file to local
docker exec sql rm -rf /var/opt/mssql/backup/MyDatabase.bak
sqlcmd -S 127.0.0.1,1433 -U sa -P !!SecureP4ssw0rd!! -H 127.0.0.1,1433 -Q "BACKUP
DATABASE [MyDatabase] TO DISK='/var/opt/mssql/backup/MyDatabase.bak'"
docker cp sql:/var/opt/mssql/backup/MyDatabase.bak .\MyDatabase.bak
```

```
# Restore a backup from container volume
docker exec -it sql mkdir /var/opt/mssql/backup
docker cp .\MyDatabase.bak sql:/var/opt/mssql/backup
sqlcmd -S 127.0.0.1,1433 -U sa -P !!SecureP4ssw0rd!! -H 127.0.0.1,1433 -Q
"RESTORE DATABASE [MyDatabase] FROM DISK='/var/opt/mssql/backup/MyDatabase.bak'
WITH REPLACE"
```

Es gibt aber auch die Möglichkeit ein Backup von dem vom Container genutzten Volumen zu erstellen. Damit haben wir nicht nur die auf dem Volumen befindliche Datenbank gesichert, sondern zusätzlich auch alle anderen Informationen, die auf dem Volumen gespeichert sind.

Hinweis: Das Sichern eines Volumens ist für viele verschiedene Arten von Containern nützlich – nicht nur für SQL Server Container, die Datenbanken auf dem Volumen speichern.

Mit den folgenden Befehlen wird ein Backup des Volumens selbst erstellt:

```
# Backup the container volume
docker run --rm \
  -v sqldata:/sqldata \
  -v $pwd\:/backup \
  ubuntu tar cvf /backup/backup.tar /sqldata
```

Mit der resultierenden ".tar"-Datei können wir später das Volumen wiederherstellen. Um dies zu testen muss das Volumen zunächst entfernt werden:

```
# Remove sqldata volume
docker volume rm sqldata
```

Nachdem das Volumen entfernt wurde, können wir es mithilfe der zuvor erstellen ".tar"-Datei wiederherstellen:

```
# Restore the container volume
docker run --rm \
  -v sqldata:/sqldata \
  -v $pwd\:/backup \
  ubuntu tar xvf /backup/backup.tar -C sqldata --strip 1
```

Um zu überprüfen, ob das Volumen wiederhergestellt wurde, kann der Befehl `docker volume ls` zum Auflisten aller bekannter Volumens genutzt werden. Ist das Volumen aufgelistet, haben wir erfolgreich ein Backup eines Container Volumens erstellt!