SQL Server Docker mit Windows Server

Kategorie
SQL Server

Anna
Mussenbrock

Mit diesem Artikel erklären wir Ihnen in ausführlichen Schritten, wie Sie die SQL Server Developer Edition in einem Windows-basierten Docker-Container verwenden können.

Zur Automatisierung von Datenbankeinstellungen ist ein wichtiger Bestandteil das Ausführen der SQL Server Developer Edition. Ist die Developer Edition installiert, wird SQL Server in der Regel lokal ausgeführt. Dabei muss jedoch der SQL Server Windows-Dienst permanent ausgeführt werden und verbraucht damit wichtige Ressourcen. Das Installationsprogramm fügt automatisch eine Reihe zusätzlicher Anwendungen hinzu, für die der User verantwortlich ist, diese zu aktualisieren.

Die Frage die wir uns stellen ist, ob es Möglichkeiten gibt, SQL Server lokal auszuführen ohne dabei permanent den SQL Server Windows Dienst ausführen zu müssen. Gibt es eventuell einfachere Upgrademöglichkeiten, die sich simpel einrichten lässt? Da SQL Server seit längerer Zeit ein Docker-Image ist, sollte es in der Lage sein, unser Anliegen zu lösen. Gehen wir der Sache einmal auf den Grund.

Zur Vorbereitung sollten Sie sich mit Docker vertraut machen und sicher gehen, dass Sie über die Kernfunktionen Bescheid wissen. Zusätzlich sind weitere Vorraussetzungen zu beachten, auf die wir folgend im Detail eingehen werden:

- Aktivieren Sie Ihre CPU Virtualisierung
- Docker für Windows installieren
- Ordner einrichten zur Freigabe für Docker Container
- Anti-Virus Konfiguration

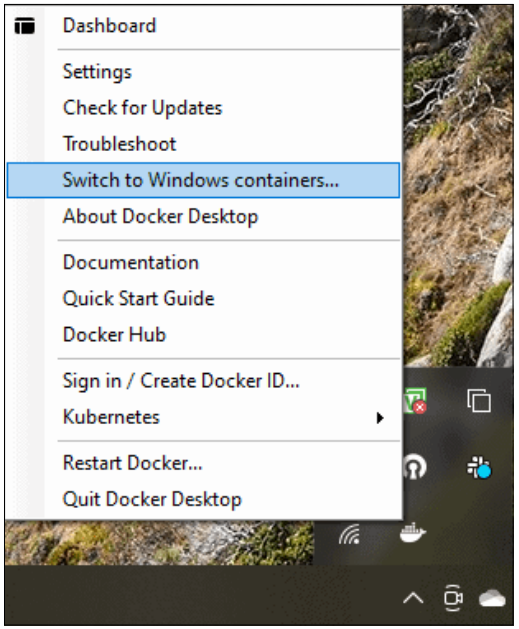
Aktivieren der CPU Virtualisierung

Bei Docker handelt es sich um einen Virtualisierungshost, daher ist es notwendig, dass die eigene CPU die Virtualisierung auch unterstützt. Diese Funktion aktivieren Sie im Regelfall über die BIOS Ihres Computers. Achten Sie darauf, dass es unterschiedliche Virtualisierungstechnologien gibt. Wir empfehlen Ihnen daher im Internet zu recherchieren, welche Technologie Ihren Computer betrifft und den Aktivierungsanweisungen entsprechend zu folgen.

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Docker für Windows installieren

Nachdem wir unsere CPU Virtualisierung eingestellt haben, installieren wir nun die **Docker Desktop Version**. Darin enthalten sind **Docker Compose** und **Docker-CLI**. Sie werden merken, wenn Sie bisher die Hyper-V Anwendung noch nicht aktiviert haben, wird das Installationsprogramm das für Sie übernehmen und einen Neustart des Computers verlangen.



Wie bereits am Anfang des Artikels gesagt, werden wir heute Windowsbasierte Container verwenden. Entsprechend müssen wir auf den Windows-Container umsteigen. Um die Einstellung vorzunehmen, gehen wir mit einem Rechtsklick auf das Docker Desktop Symbol und wählen ‚switch to Windows Containers' aus.

Einrichten von Ordnern zur Freigabe für Docker-Container

Standardmäßig behandelt alle Container als zustandslos. Das heißt im Detail, dass alle am Container vorgenommenen Änderungen, beispielsweise eine neu erstellte Datenbank, zerstört werden können. Da wir das nicht möchten, verwenden wir Volumes, die wir auf unserer Festplatte im neu angelegten Ordner C:\Docker\Volumes speichern.

Anti-Virus-Konfiguration

Ein Szenario, welches beim Ausführen des Windows-Containers aufkommen könnte, ist dass die aktivierte Antivirensoftware das Herunterladen blockieren könnte. Hintergrund ist, dass Docker Bilder im Windows Dateisystem speichert und einen anderen Ordner namens ‚Windows' an einem scheinbar zufälligen Ort anzeigt. Das löst die Alarmglocken der Antivirensoftware aus. Stellen Sie also sicher, dass sie die neuste Version der Antivirensoftware benutzen. Gegebenenfalls müssten Sie beim Scan der Antivirensoftware C:\ProgrammData\Docker ausschließen.

Nachdem wir nun alle Vorbereitungen getroffen haben, starten wir mit der **Konfiguration des SQL Server Developer Container**.

Zur Konfiguration müssen wir folgende Schritte vornehmen:

1. Starten des Containers ohne zusätzliche Konfiguration
2. Verbindung über SSMS herstellen
3. Im Container erstelle Datenbanken beibehalten
4. Speichern der Image-Konfiguration für andere Benutzer

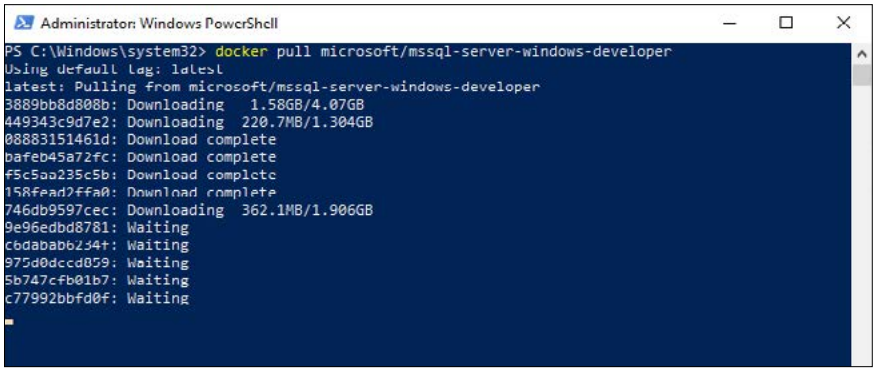
Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

SQL Server Developer Container zum ersten Mal ausführen

Wir erklären nun Schrittweise, wie Sie den SQL Server Developer Container ausführen können. Dafür führen wir zunächst den folgenden Befehl aus, um SQL Server Windows Developer-Image vom Docker Hub abzurufen:

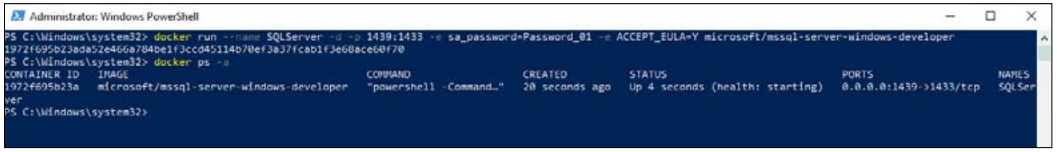
```
docker pull microsoft/mssql-server-windows-developer
```

Der Vorgang kann eine Weile dauern, da nicht nur das Image heruntergeladen wird, sondern auch alle dazugehörigen Abhängigkeiten.



Nachdem das Image heruntergeladen wurde, gehen wir weiter mit der Ausführung einiger SQL-Skripte. Wir empfehlen, sich den Parameter **–name** zu speichern, da er in späteren Schritten immer wieder verwendet wird. Danach legen wir den Standard SQL Server Port **1433** fest und führen folgenden Befehl zum Start aus:

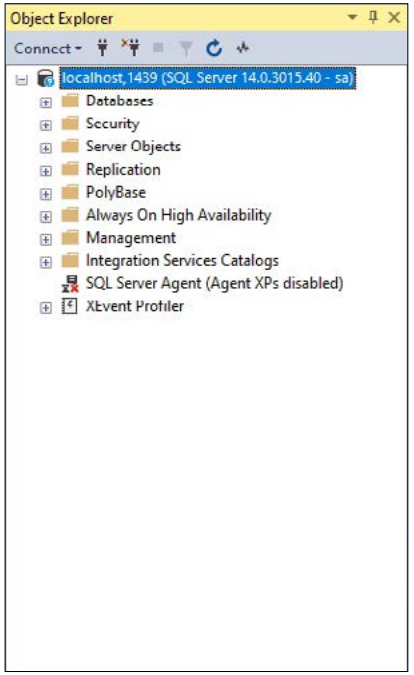
```
docker run --name SQLServer -d -p 1433:1433 -e sa_password=Password_01 -e ACCEPT_
EULA=Y microsoft/mssql-server-windows-developer
```

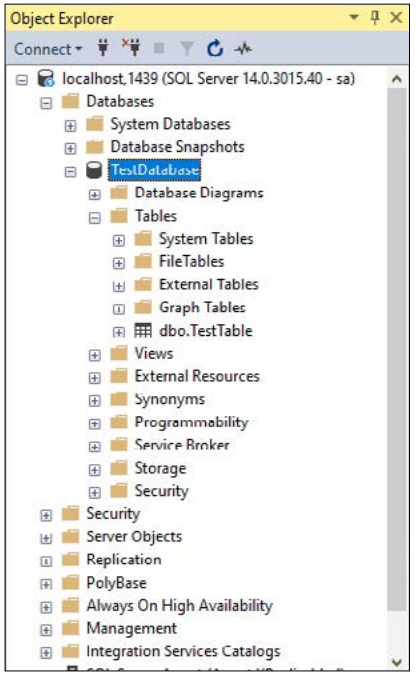


Herstellen der Verbindung zum Container über SSMS

Nun haben wir den SQL Server Container zum Laufen gebracht. Jedoch möchten wir jetzt eine Verbindung vom Host über SSMS herstellen. Im eben ausgeführten ‚run' Befehl haben wir **-p** (‚publish') verwendet. Damit haben wir im Wesentlichen nur den Port 1433 für den Host veröffentlicht, sodass wir über **localhost** darauf zugreifen können.

Um das SSMS mit dem Docker SQL Server verbinden zu können, geben wir localhost zusammen mit dem **-sa** definierten Benutzernamen & Passwort an. Ganz genau wie bei einem normalen SQL Server können wir nun ohne Probleme Datenbanken und Tabellen erstellen.





Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Was passiert, wenn der Container neu gestartet werden muss?

Wir erstellen dafür eine persistente Datenbank innerhalb unseres Containers. Eine persistente Datenbank ist eine Datenbank, in der sich Dateien (Mdf, ldf, ndf) in einem Ordner außerhalb des Containers befinden. Dadurch bleiben Änderungen an der Datenbank weiter bestehen, nachdem der SQL-Container neu gestartet oder neu erstellt werden musste. Um eine persistente Datenbank zu verknüpfen benötigen wir zwei Befehle. **-v** ordnet die Ordner der Datenbankdatei vom Host zu einem Container zu. **-attach_dbs** gibt den Datenbanknamen und die Datenbankdateien an, die dem Container angehängt werden sollen.

Container benennen

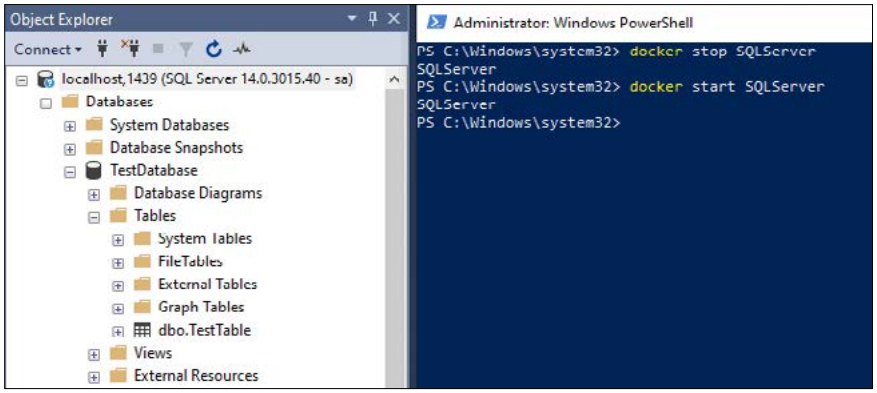
Wenn Docker einen neuen Container erstellen soll, weist er ihm eine eindeutige ID zu, mit der der Container beim Starten und Stoppen authentifiziert wird. Diese IDs bestehen in der Regel aus zufälligen Zahlen und Buchstaben. Um ähnlich benannte Container auseinander zu halten, ist es möglich einen optionalen Attributnamen zu vergeben. Dafür geben wir folgende konsolidierte Syntax ein:

Docker run <Attribute> <Repository- / Image-Name>
-d (getrennter Modus)
-it (interaktiver Modus)
-p (Portzuordnung) -Syntax: <Hostport>: <Container-Port>
-v (Verzeichniszuordnung)
Hinweis: Laufwerksbuchstaben müssen für den Hostcomputer gültig sein.
-Name (optionaler Containername)
Hinweis: Muss mit 2 Bindestrichen versehen werden.
-e (Umgebungsvariablen)

× SA_PASSWORD = <Kennwort> *** Erforderlich ***
× ACCEPT_EULA = Y *** Erforderlich ***
× attach_dbs = "[{'dbName': 'PersistentDB', 'dbFiles': ['c: \ data \ PersistentDB.mdf', 'c: \ log \ PersistentDB_log.ldf']}]"

Zum Starten führen wir folgenden Befehl aus:

```
docker start SQLServer
```



Die Datenbanken und Tabellen sind in unserem Beispiel nach dem Neustart noch vorhanden.

Was ist jedoch, wenn der Container neu erstellt werden muss? Das Problem könnte auftreten, wenn neue Containerkonfigurationen vorgenommen, oder eine neue Version veröffentlicht wurde. Daher ergänzen wir zum **-Stop** Befehl, den Befehl **-rm** mit dem der Container entfernt wird.

Wie Sie anhand unseres Beispiels sehen, sind nun alle Datenbanken gelöscht worden. Wir suchen allerdings nun eine Möglichkeit, die Daten beizubehalten, auch wenn ein Neustart des Containers oder auch eine Neuerstellung eines Containers notwendig wird.

Dafür kommen wir nun auf unsere erstellten Volumes **C:\Docker\Volumes\SQLServer** zurück. Wir wollen nun ein Volume zum Container hinzufügen. Wichtig ist jedoch, bevor das Volume hinzugefügt wird, muss der bereits ausgeführte Container zerstört werden. Das wird mit folgendem Befehl ausgeführt:

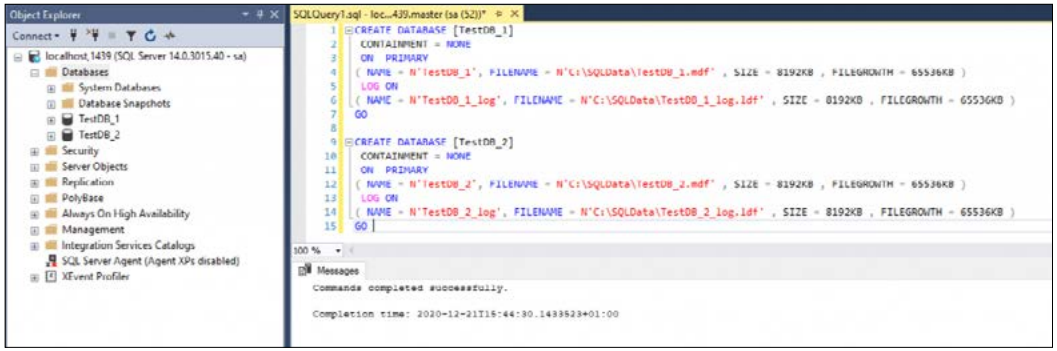
Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

```
docker stop SQLServer
docker rm SQLServer
docker run --name SQLServer -d -p 1433:1433 --volume c:\Docker\Volumes\
SQLServer:c:\SQLData -e sa_password=Password_01 -e ACCEPT_EULA=Y microsoft/mssql-
server-windows-developer
```

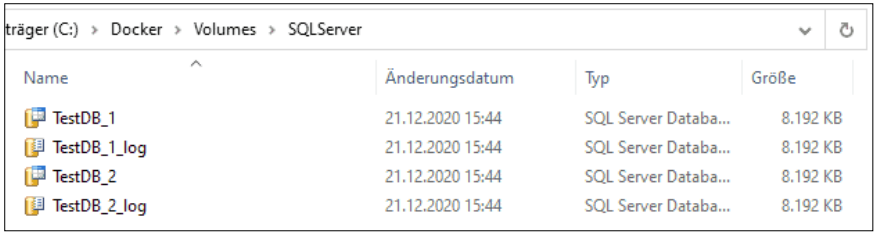
Alle Befehle zur Erstellung einer Datenbank sollten als Datenverzeichnis C:\SQLData\ angeben. Soll der SQL Server Container nun unsere Datenbanken hosten, müssen wir folgenden Befehl ausführen:

```
CREATE DATABASE [TestDB_1]
 CONTAINMENT = NONE
 ON  PRIMARY
( NAME = N'TestDB_1', FILENAME = N'C:\SQLData\TestDB_1.mdf' , SIZE = 8192KB ,
FILEGROWTH = 65536KB )
 LOG ON
( NAME = N'TestDB_1_log', FILENAME = N'C:\SQLData\TestDB_1_log.ldf' , SIZE =
8192KB , FILEGROWTH = 65536KB )
GO

CREATE DATABASE [TestDB_2]
 CONTAINMENT = NONE
 ON  PRIMARY
( NAME = N'TestDB_2', FILENAME = N'C:\SQLData\TestDB_2.mdf' , SIZE = 8192KB ,
FILEGROWTH = 65536KB )
 LOG ON
( NAME = N'TestDB_2_log', FILENAME = N'C:\SQLData\TestDB_2_log.ldf' , SIZE =
8192KB , FILEGROWTH = 65536KB )
GO
```



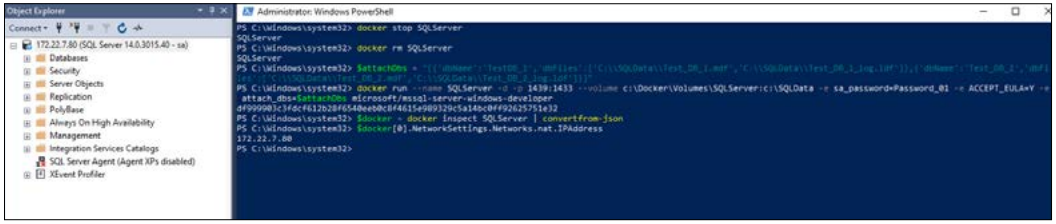
Unser Befehl war erfolgreich und wird uns nun im Verzeichnis des Hostsystems angezeigt.

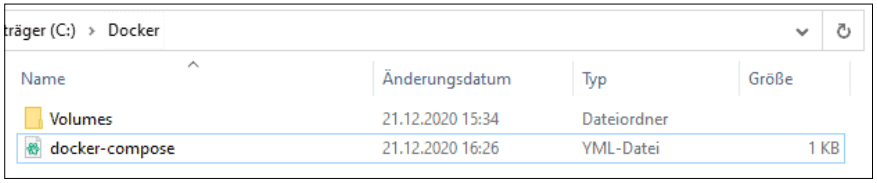


Die Datenbanknamen und -pfade können wir mithilfe der Übungsvariable **-attach_dbs** an den Container übergeben. Mit folgendem Befehl werden die Datenbanken bereitgestellt, sofern ein neuer Container erstellt wird:

```
docker stop SQLServer
docker rm SQLServer
$attachDbs = "[{"dbName":"TestDB_1","dbFiles":["C:\\SQLData\\TestDB_1.mdf","C:\\
SQLData\\TestDB_1_log.ldf"]},{"dbName":"TestDB_2","dbFiles":["C:\\SQLData\\
TestDB_2.mdf","C:\\SQLData\\TestDB_2_log.ldf"]}]"
docker run --name SQLServer -d -p 1433:1433 --volume c:\Docker\Volumes\
SQLServer:c:\SQLData -e sa_password=Password_01 -e ACCEPT_EULA=Y -e attach_
dbs=$attachDbs microsoft/mssql-server-windows-developer
```


Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612





Speichern der Konfiguration in Docker Compose

Wir haben bisher viele Befehle über die Kommandozeile ausgeführt. Wir sparen uns Zeit und Aufwand, indem wir die Docker-Container Konfigurationen in Docker Compose speichern. Somit speichern wir die Infos in einer YAML-Datei, anstelle einer Skriptdatei. Wir speichern unsere Docker Compose Datei auf **C:\Docker** auf unserer Festplatte.

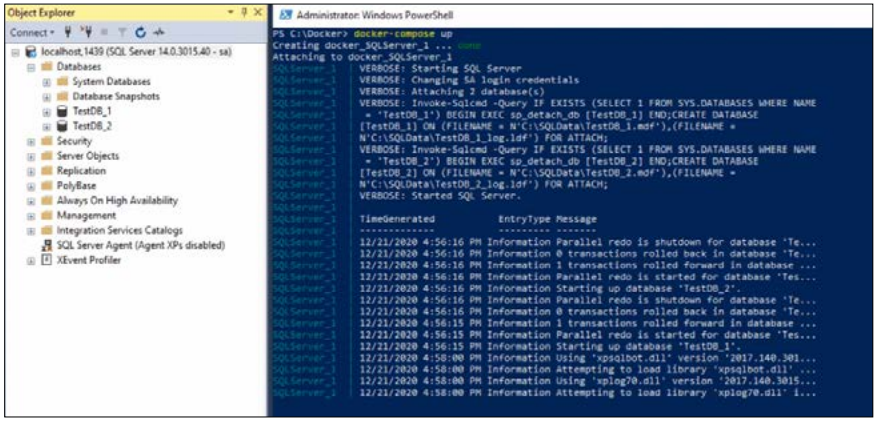
```
version: "3.7"

services:
    SQLServer:
      image: microsoft/mssql-server-windows-developer
      environment:
        - ACCEPT_EULA=Y
        - SA_PASSWORD=Password_01
        - attach_dbs=[{"dbName":"TestDB_1","dbFiles":["C:\\
          SQLData\\TestDB_1.mdf","C:\\SQLData\\TestDB_1_log.
          ldf"]},{"dbName":"TestDB_2","dbFiles":["C:\\SQLData\\TestDB_2.
          mdf","C:\\SQLData\\TestDB_2_log.ldf"]}]
      ports:
        - "1439:1433"
      volumes:
        - c:\Docker\Volumes\SQLServer:c:\SQLData
```

Zum Ausführen geben anschließend unseren Befehl in der PowerShell ein:

```
Set-Location C:\Docker
docker-compose up -d
```

Wie wir sehen können, erhalten wir die gleichen Ergebnisse und sparen uns damit viel Zeit.



Fazit

Wir haben Ihnen nun in einfachen Schritten erklärt, wie Sie SQL Server in Docker ausführen können. Sie können mit dem Speichern der Konfiguration in Docker Compose wichtige Zeit sparen und erzielen die gleichen Ergebnisse wie wenn Sie einzelne Befehle über die Kommandozeile ausführen. Damit geben wir Ihnen eine Möglichkeit an die Hand, Docker auf Ihrem Developercomputer zu hosten, ohne SQL Server Developer installieren zu müssen.

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612