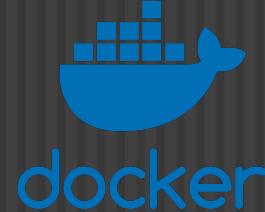




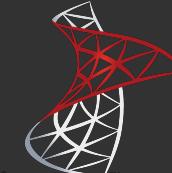
SETTING UP AND DEVELOPING SQL SERVER ON NON-WINDOWS OS

SQLPort – 24 September 2019



ABOUT ME

- Software developer
 - .NET Core and EF Core;
- Getting to know SQL Server better;
- DIY and IoT enthusiast;
- RaspberryPi fan;



WHAT WILL BE COVERED

- Azure Data Studio (ADS)
 - Interface
 - SQL Server Profiler
 - Notebooks
 - Backup/Restore features
- Docker
 - Create and launch containers with different SQL Server versions
 - Mounted volumes
- dbatools – Backup/Restore

LET'S UNDERSTAND THIS COMMAND

```
docker run \
```

Call the docker program to run according to the given settings.

```
--name 'sql2019_test' \
-e 'ACCEPT_EULA=Y'
-e 'MSSQL_SA_PASSWORD=SqlPort2019' \
-p 1443:1433 \
-v sqlPortdata:/var/opt/mssql \
-v /Users/daniel/Documents/Projects/Presentations/SqlPort/Backups:/backup \
-d mcr.microsoft.com/mssql/server:2019-latest
```

LET'S UNDERSTAND THIS COMMAND

```
docker run \  
  --name 'sql2019_test' \  
  -e 'ACCEPT_EULA=Y'  
  -e 'MSSQL_SA_PASSWORD=SqlPort2019' \  
  -p 1443:1433 \  
  -v sqlPortdata:/var/opt/mssql \  
  -v /Users/daniel/Documents/Projects/Presentations/SqlPort/Backups:/backup \  
  -d mcr.microsoft.com/mssql/server:2019-latest
```

--name 'sql2019_test' → Specify the name of the container.\

LET'S UNDERSTAND THIS COMMAND

```
docker run \
  --name 'sql2019_test' \
  -e 'ACCEPT_EULA=Y'
  -e 'MSSQL_SA_PASSWORD=SqlPort2019' \
  -p 1443:1433 \
  -v sqlPortdata:/var/opt/mssql \
  -v /Users/daniel/Documents/Projects/Presentations/SqlPort/Backups:/backup \
  -d mcr.microsoft.com/mssql/server:2019-latest
```

-e 'ACCEPT_EULA=Y'
-e 'MSSQL_SA_PASSWORD=SqlPort2019' → Environment variables. Used to accept the EULA and specify the SA password.

LET'S UNDERSTAND THIS COMMAND

```
docker run \
--name 'sql2019_test' \
-e 'ACCEPT_EULA=Y'
-e 'MSSQL_SA_PASSWORD=SqlPort2019' \


-p 1443:1433 \


-v sqlPortdata:/var/opt/mssql \
-v /Users/daniel/Documents/Projects/Presentations/SqlPort/Backups:/backup \
-d mcr.microsoft.com/mssql/server:2019-latest
```

The port 1433 on the container (right side) will be mapped to port 1443 on the host (left side).

LET'S UNDERSTAND THIS COMMAND

```
docker run \
  --name 'sql2019_test' \
  -e 'ACCEPT_EULA=Y'
  -e 'MSSQL_SA_PASSWORD=SqlPort2019' \
  -p 1443:1433 \
  -v sqlPortdata:/var/opt/mssql \
  -v /Users/daniel/Documents/Projects/Presentations/SqlPort/Backups:/backup \
  -d mcr.microsoft.com/mssql/server:2019-latest
```

Mounts the folder `/var/opt/mssql` on the volume `sqlPortdata`.
Because this is not a path that exists on the host, it is mapped to a path on the docker VM.

LET'S UNDERSTAND THIS COMMAND

```
docker run \
  --name 'sql2019_test' \
  -e 'ACCEPT_EULA=Y'
  -e 'MSSQL_SA_PASSWORD=SqlPort2019' \
  -p 1443:1433 \
  -v sqlPortdata:/var/opt/mssql \
  -v /Users/daniel/Documents/Projects/Presentations/SqlPort/Backups:/backup \
  -d mcr.microsoft.com/mssql/server:2019-latest
```

Maps the **Backups** folder to the **/backup** folder on the container. This means that whatever is on the **Backups**, can be accessed on the container by going to the **/backup**

LET'S UNDERSTAND THIS COMMAND

```
docker run \
--name 'sql2019_test' \
-e 'ACCEPT_EULA=Y'
-e 'MSSQL_SA_PASSWORD=SqlPort2019' \
-p 1443:1433 \
-v sqlPortdata:/var/opt/mssql \
-v /Users/daniel/Documents/Projects/Presentations/SqlPort/Backups:/backup \
-d mcr.microsoft.com/mssql/server:2019-latest
```

The command above is a Docker run command used to create a container named 'sql2019_test'. It specifies several environment variables: 'ACCEPT_EULA=Y' and 'MSSQL_SA_PASSWORD=SqlPort2019'. It maps port 1443 to 1433. It mounts two volumes: 'sqlPortdata' to '/var/opt/mssql' and a local directory containing backups to '/backup'. Finally, it pulls the Microsoft SQL Server 2019 image from the Docker Hub.

-d mcr.microsoft.com/mssql/server:2019-latest

Launches the container in detached mode.

LET'S UNDERSTAND THIS COMMAND

```
docker run \
--name 'sql2019_test' \
-e 'ACCEPT_EULA=Y'
-e 'MSSQL_SA_PASSWORD=SqlPort2019' \
-p 1443:1433 \
-v sqlPortdata:/var/opt/mssql \
-v /Users/daniel/Documents/Projects/Presentations/SqlPort/Backups:/backup \
-d mcr.microsoft.com/mssql/server:2019-latest
```

Specifies which image is to run. If the image does not exist locally, it is pulled from the repository.

DOCKER COMMANDS CHEAT SHEET

- Docker run - Launches the specified container with the specified settings. If the image does not exist locally, it will fetch it
- Docker pull – Fetch the image from the repository
- Docker ps – Lists the containers. Specify the –a switch to list all (by default only shows the ones running)
- Docker stop – Stops the container
- Docker rm – Removes the container
- Docker exec – Run a command inside the container
- Docker volume ls – Lists all volumes created
- Docker rm volume – Removes the specified volume

REFERENCES

- Azure Data Studio Notebooks by Rob Sewell -
<https://sqldbawithabeard.com/2019/07/17/powershell-in-sql-notebooks-in-azure-data-studio/>
- Persisting SQL Server Data in Docker Containers by Anthony Nocentino -
<http://www.centinosystems.com/blog/sql/persisting-sql-server-data-in-docker-containers-part-1/>
- SQLServer docker repository - https://hub.docker.com/_/microsoft-mssql-server
- Tags available - <https://mcr.microsoft.com/v2/mssql/server/tags/list>

QUESTIONS?



THANK YOU!