

Lab: Het kopiëren van data met Azure Data Factory

Doel

Het "Data Movement" stuk van Azure Data Factory wordt in de meest eenvoudige (én voordelige!) vorm vertegenwoordigd door de **Copy Data** activity.

In dit lab gaan we deze activity gebruiken om data te kopiëren:

- vanuit de tabel **Customer** binnen de **awlt** database
- naar het Data Lake in een **Parquet** bestand

Waarom Parquet?

Parquet is niet zo'n heel bekend bestandsformaat wanneer je het vergelijkt met bijvoorbeeld CSV of Excel. In Data Lakes wordt het echter veelvuldig gebruikt. Onderaan dit lab staat een video met een korte verdieping waarom Parquet-bestanden zo goed werken voor een Data Lake.

Stappenplan

Het aanmaken van zowel pipelines als datasets gebeurt in de sectie **Author** van Azure Data Factory.

In het vorige lab hebben we een *linked service* aangemaakt. Hier staat alle informatie in over hoe we een bepaalde *service* kunnen benaderen. Maar we hebben ook gezien dat deze services allerlei vormen kunnen aannemen: van een Storage Account tot aan SQL databases, maar ook wachtwoordbeheer met Azure Key Vault bijvoorbeeld. En zelfs binnen een SQL database kan het een stored procedure zijn die we een opdracht geven (orchestratie), of juist een tabel die we uitlezen (data movement).

Wanneer we data gaan lezen en schrijven, heeft Data Factory daarom nog wat extra informatie nodig: een **Dataset**. De Dataset bevat alle informatie over de structuur van de data.

Aanmaken van datasets

Maak een dataset **ds_awlt_Customer**

- De dataset staat op een Azure SQL Database
- Gebruik als Linked service **ls_sql_awlt**
- Selecteer de table **SalesLT.Customer**
- Laat het vinkje **Import Schema** staan op **From connection/store**

Bekijk de dataset die je zojuist gemaakt hebt. Onder het tabje **Schema** zie je alle informatie die Azure Data Factory zojuist opgehaald heeft uit deze tabel.

Maak nu een tweede dataset met de naam **ds_adls_awlt_Customer**

- Deze dataset moet landen op een Azure Blob Storage
- Kies als format **Parquet**
- Gebruik als Linked Service **ls_adls**
- Onder **File path** vul je in het eerste vakje (*container*) de waarde **stg** in.

- In dit geval hebben we nog geen data beschikbaar voor het schema. Kies daarom onder **Import schema** voor **None**

Bekijk opnieuw de dataset die je zojuist gemaakt hebt. Merk op:

- Er staat vastgelegd dat het een Parquet-bestand is
- Er wordt een bepaalde compressie toegepast ("Snappy", whatever that may be)
- Onder het tabje **Schema** is geen informatie over het schema in het Parquet-bestand.

Daarnaast zie je een grijze bal bij de namen van je datasets (zowel bovenin je scherm als aan de linkerkzijde). Dit betekent simpelweg dat ze nog niet gepubliceerd zijn. Zodra je op **Publish all** klikt, verdwijnt de grijze bal en staan je datasets "live".

 Grijze bal betekent niet gepubliceerd


Aanmaken van de pipeline

Zoals besproken is een pipeline bij ADF niet iets waar data "doorheen stroomt", maar een verzameling van activiteiten. Je zou het ook een "orchestratie" kunnen noemen. (Een technische term die bijvoorbeeld in een vergelijkbaar product als Apache Airflow wordt gebruikt is een *DAG - Directed Acyclic Graph*)

Maak een pipeline met de naam `pl_alwt_adls_Customer`

- Plaats een **Copy data** activity op het canvas
 - Configureer als bron (*source*) de dataset `ds_awlt_Customer`
 - Configureer als doel (*sink*) de dataset `ds_adls_awlt_Customer`

Debug de pipeline

- Start nu de pipeline met de knop **Debug** (bovenin het scherm) 
- Wanneer de uitvoer goed is gegaan, controleer dan in je Data Lake of er ook daadwerkelijk een bestand verschijnt.
- Publiceer alle gemaakte resources.

En nu zelf!

Maak nu de data-oplossing af door ook de andere tabellen binnen de `awlt` database via een pipeline in te laden naar je Data Lake.

Verdieping: Waarom Parquet?

Parquet is niet zo'n heel bekend bestandsformaat wanneer je het vergelijkt met bijvoorbeeld CSV of Excel. In Data Lakes wordt het echter veelvuldig gebruikt.

In onderstaande video legt één van onze trainers uit waarom Parquet zo interessant is voor opslag op je Data Lake. Het is een eerste video in een serie van drie, waarin meer uitgelegd wordt over het direct kunnen uitvoeren van SQL-queries op je Data Lake.

