

Intro to Serverless SQL in Azure Synapse



Agenda

- Overview of the Serverless SQL pool
- Create the Synapse Workspace in Azure
- Walkthrough of the Synapse Workspace
- Analyse data using the Serverless SQL Pool
 - CSV, Parquet, JSON
 - Tips and tricks
- Next steps to try yourself



Brian Bønk Rueløkke

Principal & Enterprise arkitekt, Data & Analytics

Fellowmind



<https://linkedin.com/in/brianbonk>



<https://brianbonk.dk>



Microsoft

FastTrack Recognized
Solution Architect
Power BI
2022 >>



Microsoft

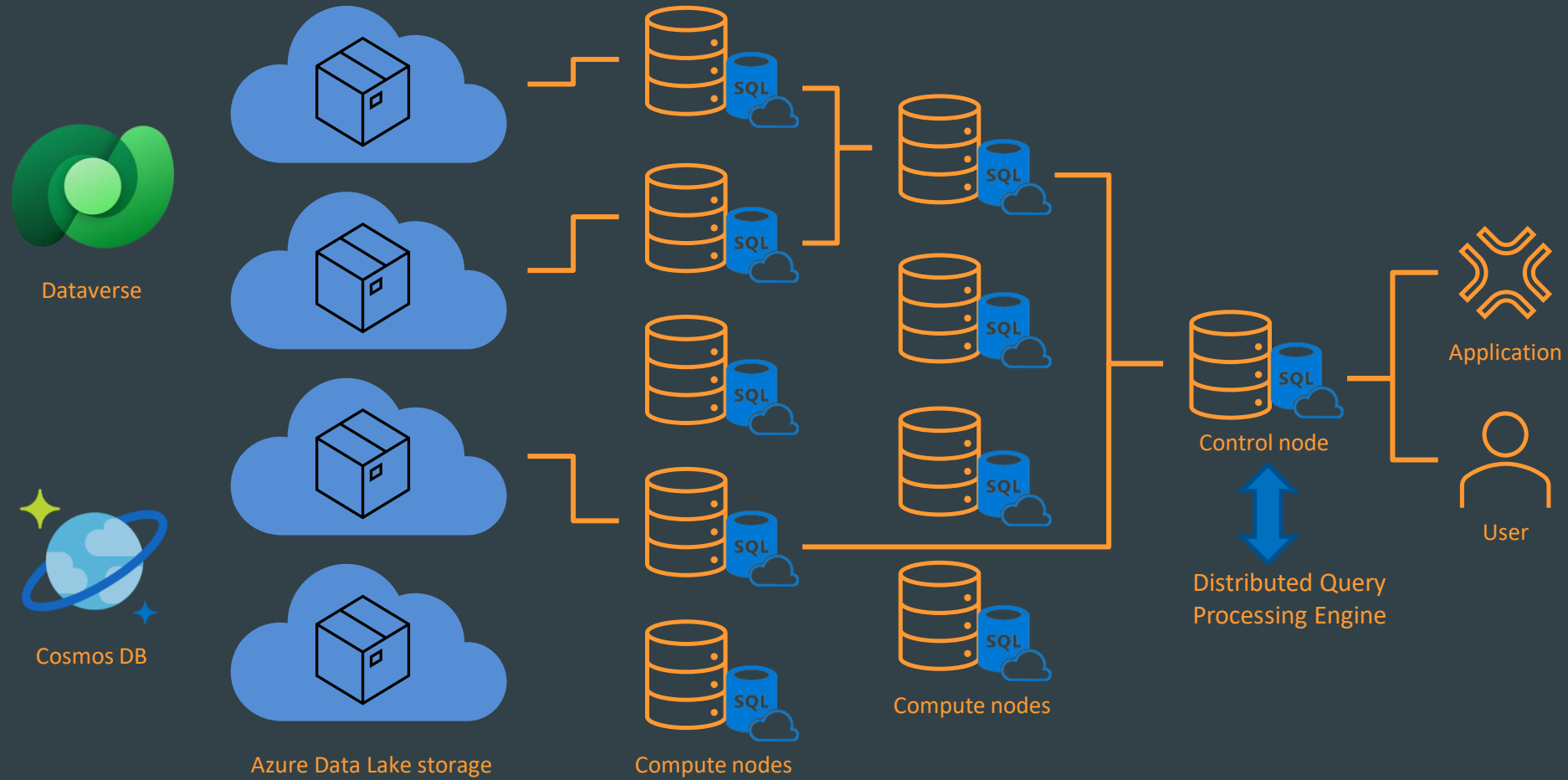
Certified Trainer
Data Platform

2018 >>

The Serverless SQL Pool

- A part of every Synapse Analytics Workspace
- Enables you to query data from Azure Data Lake, Cosmos DB and Dataverse
- Using familiar T-SQL syntax to query data in place without the need to move or copy the data
- What about the cost?
- Pay-per-use setup – approx. 5\$ pr TB processed data

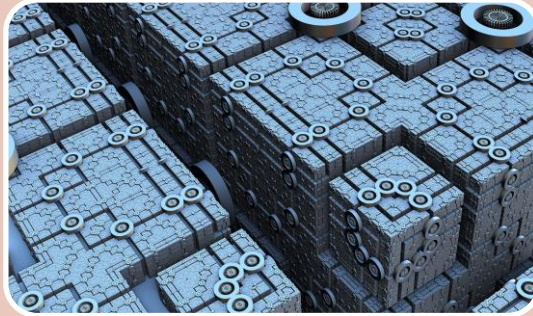
The Serverless SQL Pool - architecture



Benefits



Basic discovery
and exploration



Logical Data
Warehouse



Simple data
transformations

Benefits



Data Engineers can explore the lake, transform and prepare data using this service, and simplify their data transformation pipelines.



Data Scientists can quickly reason about the contents and structure of the data in the lake, thanks to features such as OPENROWSET and automatic schema inference.



Data Analysts can explore data and Spark external tables created by Data Scientists or Data Engineers using familiar T-SQL language or their favourite tools, which can connect to serverless SQL pool.



BI Professionals can quickly create Power BI reports on top of data in the lake and Spark tables.

T-SQL support

- DDL: Databases, Schemas, Views and Stored Procedures -> No Tables, Materialized Views and Functions!
- Security: Logins and users, Credentials, Grant/Deny on object level, AAD integration
- SELECT: (almost) full SELECT support – a minor subset of features are not (yet) available



DEMO

Create Synapse Workspace

The initial setup



**Azure Synapse
Analytics**



**Azure Blob
Storage**



**Serverless SQL
endpoint**



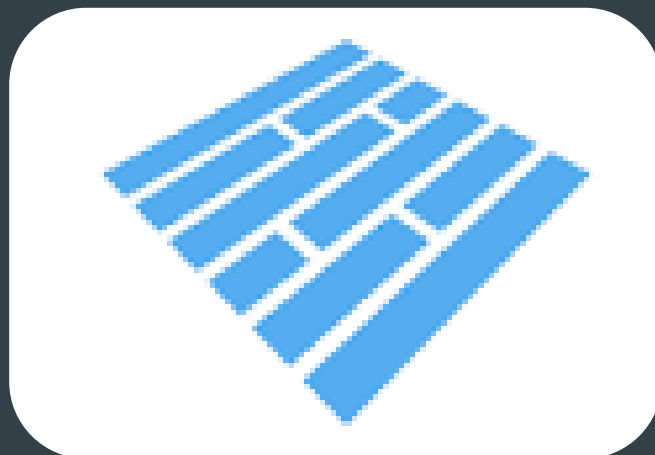
DEMO

A quick walkthrough of the Synapse workspace

Query data using Synapse SQL Serverless

```
created_at,entry_id,field1,field2,field3,  
2019-07-26 12:50:13,102771,27.3,29.8,45.0,  
2019-07-26 13:00:13,102772,26.9,28.3,44.0,  
2019-07-26 13:10:13,102773,28.0,28.3,41.0,  
2019-07-26 13:20:13,102774,27.8,28.3,39.0,  
2019-07-26 13:30:13,102775,27.0,28.3,40.0,  
2019-07-26 13:40:13,102776,26.8,28.3,42.0,  
2019-07-26 13:50:13,102777,27.0,28.3,42.0,  
2019-07-26 14:00:14,102778,26.8,27.2,42.0,  
2019-07-26 14:10:13,102779,27.0,27.2,42.0,  
2019-07-26 14:20:13,102780,26.8,27.2,43.0,  
2019-07-26 14:30:13,102781,26.4,27.2,44.0,  
2019-07-26 14:40:13,102782,27.1,27.2,42.0,  
2019-07-26 14:50:13,102783,26.2,27.2,43.0,  
2019-07-26 15:00:14,102784,25.6,26.6,44.0
```

CSV



Parquet

```
{  
  "id": 12635853,  
  "name": "json-viewer",  
  "full_name": "tulios/json-viewer",  
  "owner": {  
    "login": "tulios",  
    "id": 33231,  
    "avatar_url": "https://avatars.githubusercontent.com/u/33231?v=4",  
    "gravatar_id": "",  
    "url": "https://api.github.com/users/tulios",  
    "html_url": "https://github.com/tulios",  
    "followers_url": "https://api.github.com/users/tulios/followers",  
    "following_url": "https://api.github.com/users/tulios/following",  
    "gists_url": "https://api.github.com/users/tulios/gists"  
```

JSON



DEMO

Query CSV, JSON and Parquet data

CSV PARSER VERSIONS

CSV parser version 1.0 specifics:

- Following options aren't supported: HEADER_ROW.
- Default terminators are \r\n, \n and \r.
- If you specify \n (newline) as the row terminator, it will be automatically prefixed with a \r (carriage return) character, which results in a row terminator of \r\n.

CSV parser version 2.0 specifics:

- Not all data types are supported.
- Maximum character column length is 8000.
- Maximum row size limit is 8 MB.
- Following options aren't supported: DATA_COMPRESSION.
- Quoted empty string ("") is interpreted as empty string.
- DATEFORMAT SET option is not honored.
- Supported format for DATE data type: YYYY-MM-DD
- Supported format for TIME data type: HH:MM:SS[.fractional seconds]
- Supported format for DATETIME2 data type: YYYY-MM-DD HH:MM:SS[.fractional seconds]
- Default terminators are \r\n and \n.

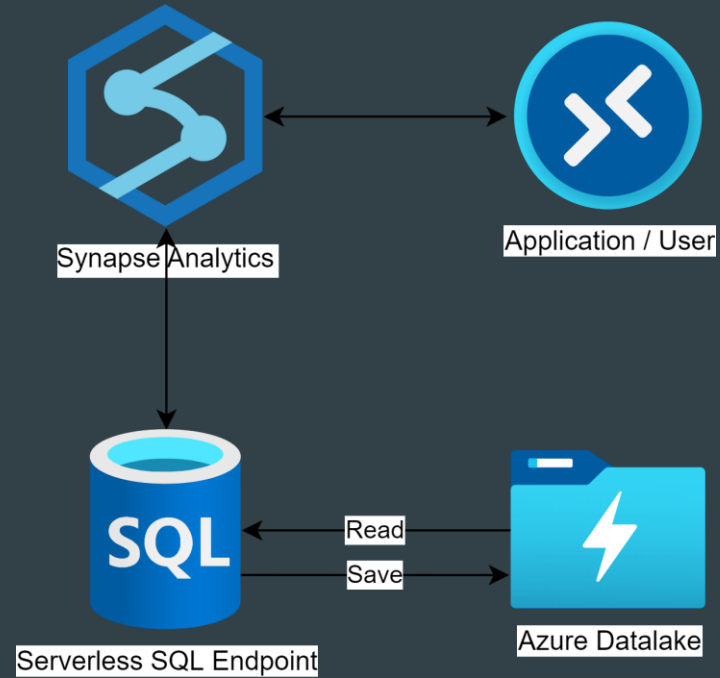
CETAS

Synapse SQL also provides the ability to save a coded view from the DataLake to a physical file.

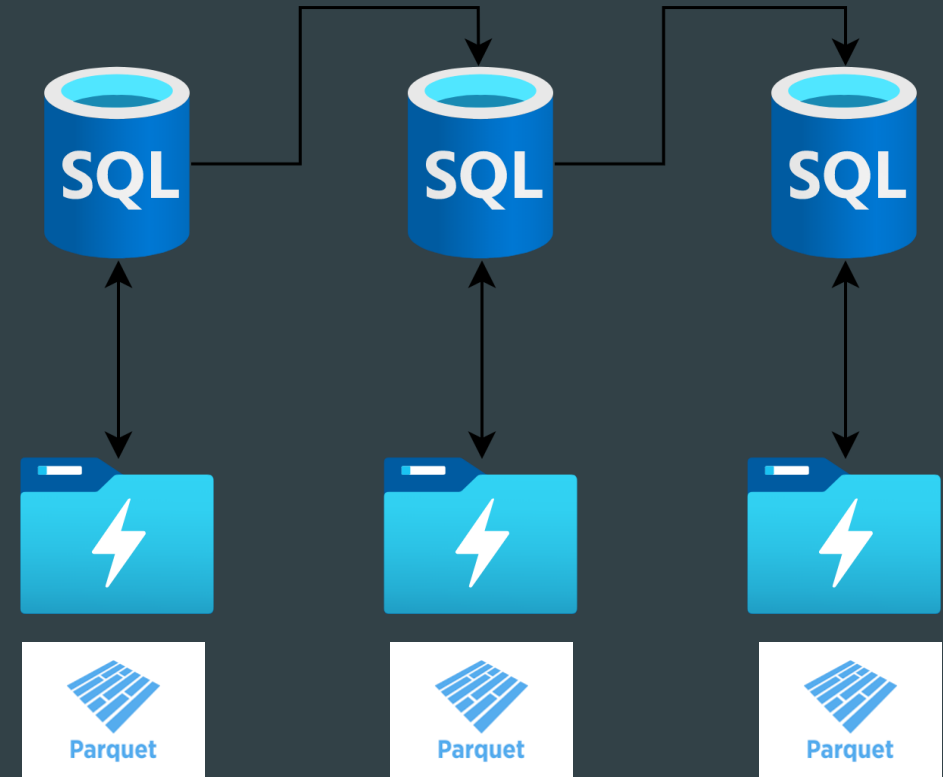
This process is called a CETAS – Create External Table AS.
It saves a Parquet file in the DataLake in a specified folder.

The usage of this could be to provide precalculated dataset to an application or end user with very little wait time. The precalculation happens in the creation of the CETAS.

CETAS



Virtual Data Warehouse



Tips & Tricks

- If your CSV file is UTF-8 formatted, then use a UTF-8 collation when creating the database – this helps and removes a lot of frustration
- Try to use PARSER VERSION 2 when reading CSV – it is much faster, but demands specific formatting in the CSV file – read more in the addendum slides.

Try it out for yourself

NEXT STEP

<https://github.com/brianbonk/public/tree/master/Speaks/2023>

Demo data and scripts

- All demo data and scripts will be provided.
- You need to change the “address” to the storage account, to point to your own account.
- Data is free to use and comes from the public Taxi data endpoint from [NY Taxi website](#) (Parquet and CSV) and my own blog (JSON).

Want more?

Then these two videos could be the next step into the Synapse world

Supercharge Power BI with Azure Synapse Analytics
(with Mathias Halkjær) – Havens Consulting



<https://www.youtube.com/live/CgSuTHKJH60>

Azure Synapse Serverless SQL Pools - CETAS vs
Views vs Parquet – Data Toboggan



<https://www.youtube.com/watch?v=f3uQCbJO16U>