

Azure Synapse Analytics

April 2022 Camilla Gaardsted

SuperUsers A/S

Relational Databases in Azure (SQL databases)

MS SQL 4 options in Azure:

- **Azure SQL Database**
- VM with MS SQL Server 2017+ standard/enterprise
- Managed instance
- **Azure Synapse Analytics (Data warehouse)**

Alternatives:

- Azure database for MySQL, PostgreSQL, etc
- VM with Oracle, MySQL, etc

MS Relationelle database muligheder i 2021

Azure SQL Database

Platform as a Service (PaaS) =>

Altid **nyeste** SQL version

Azure AD

INGEN Jobagent

Elastic pool

INGEN classic backup/restore



Azure SQL VM

OS: Windows Server/Linux

SQL Server 2017+

Windows AD

Jobagent

SQL Server IaaS extension

Always On availability



Managed Instance (MI)

Platform as a Service (PaaS) =>

Altid **nyeste** SQL version

Azure AD

Jobagent

Classic backup/restore supported

Backup kan restores på SQL Server 2022!!!



On premise løsning

OS: Windows Server/Linux

SQL Server 2017+

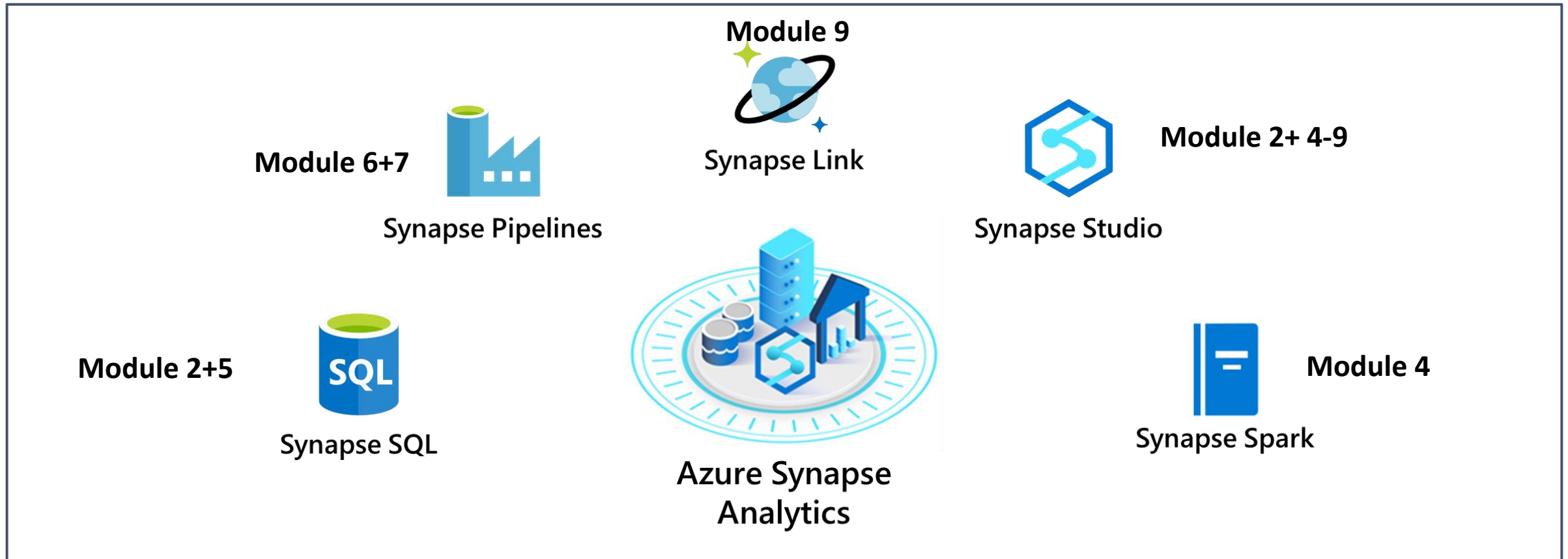
Windows AD

Jobagent

Always On availability



Explore Azure Synapse Analytics



Parallel DW timeline

- 2010 Parallel DW on premise
 - 2015 Parallel DW in Azure (on logical sql server)
 - 2019 Synapse Analytics (formerly Parallel DW)
 - 2020 Synapse Analytics
-
- NB Dedicated pool still exists as a standalone on a logical sql server



Dedicated SQL pool
(formerly SQL DW)



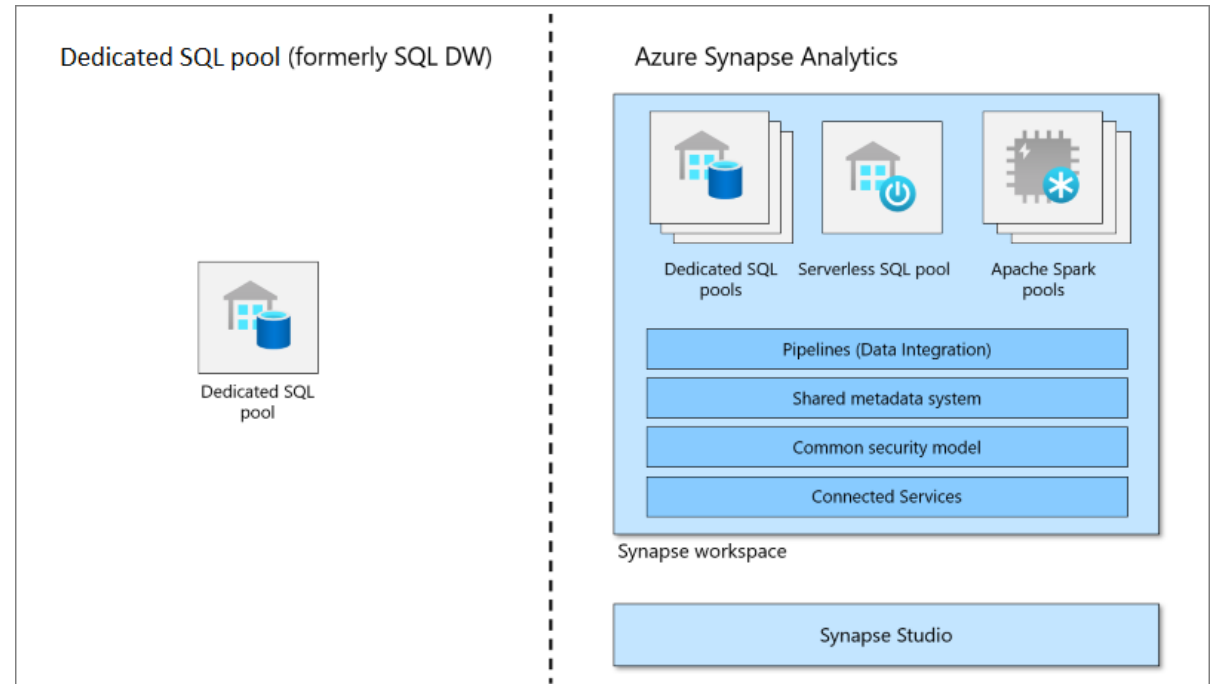
Azure Synapse Analytics

Azure Synapse Analytics

Formerly known as Azure SQL **Data Warehouse**

A portfolio with:

- Azure Synapse SQL
- Azure Synapse Pipelines
- Azure Synapse Link
- Apache Spark pool
- Synapse Studio



Synapse Analytics – Managed resource group

Resources

Recommendations

Filter for any field...

Type == all X


Location == all X

+ Add filter


Showing 1 to 2 of 2 records. ☒ Show hidden types ⓘ

☐ Name ↑↓

Type ↑↓

☐  master (synapse20220510/master)

SQL database

☐  synapse20220510

SQL server

Azure Synapse Workspace

Workspace name must be unique to form address:

- <workspace name>.sql.azuresynapse.net

Requires a primary storage account (Spark needs this)

- Data Lake (Gen 2) with a container

Filesystem is the container in the storage account

Storage blob data contributor role for workspace

- Optional for current user

Firewall rules at workspace level

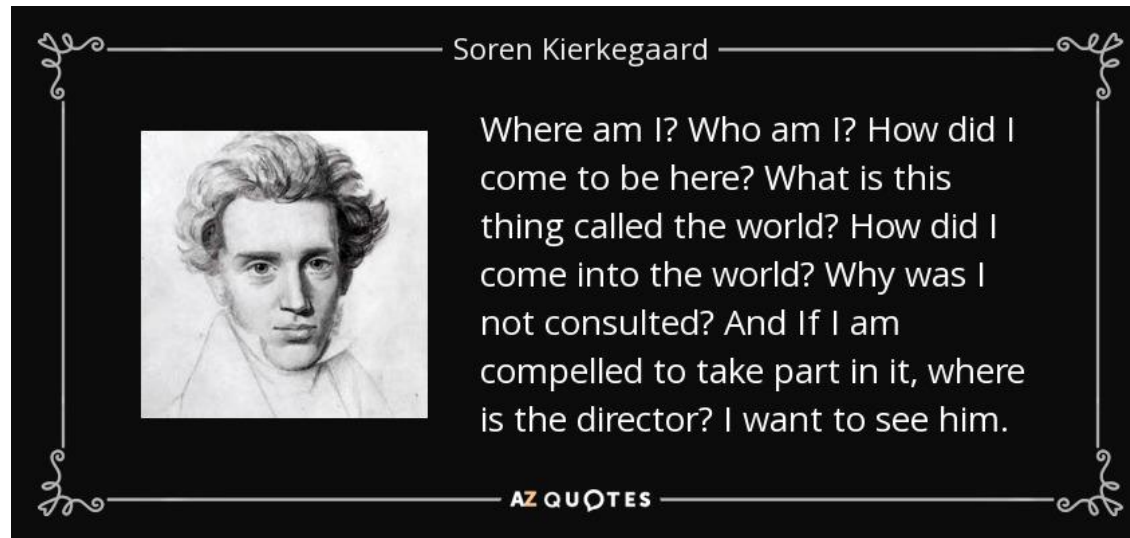
- **Allow all (default!)**

Rule name	Start IP	End IP
allowAll	0.0.0.0	255.255.255.255

Azure Synapse Workspace

SQL Admin username

SQL Active Directory admin (auto created when AD user creates it)

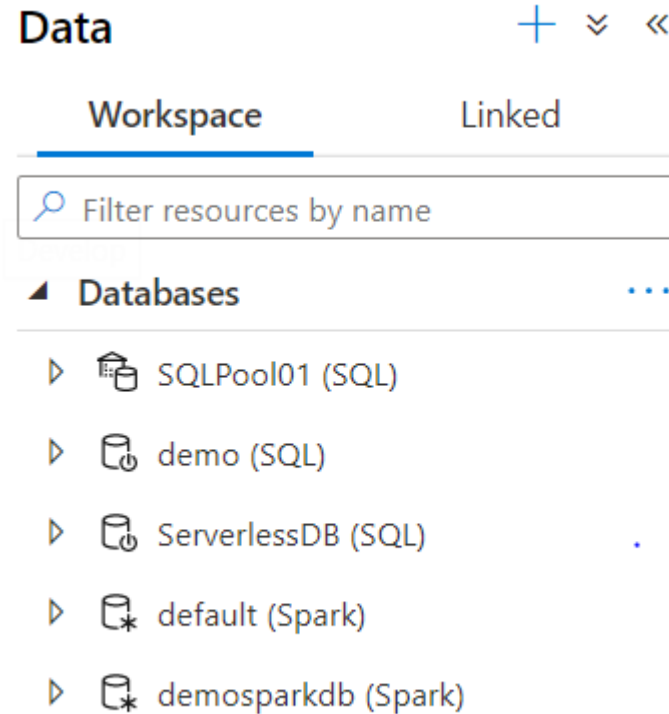


Kilde: <https://www.azquotes.com/>

Azure Synapse Analytics - Pools

- Serverless SQL pool
- Dedicated SQL pool
- Apache Spark pool

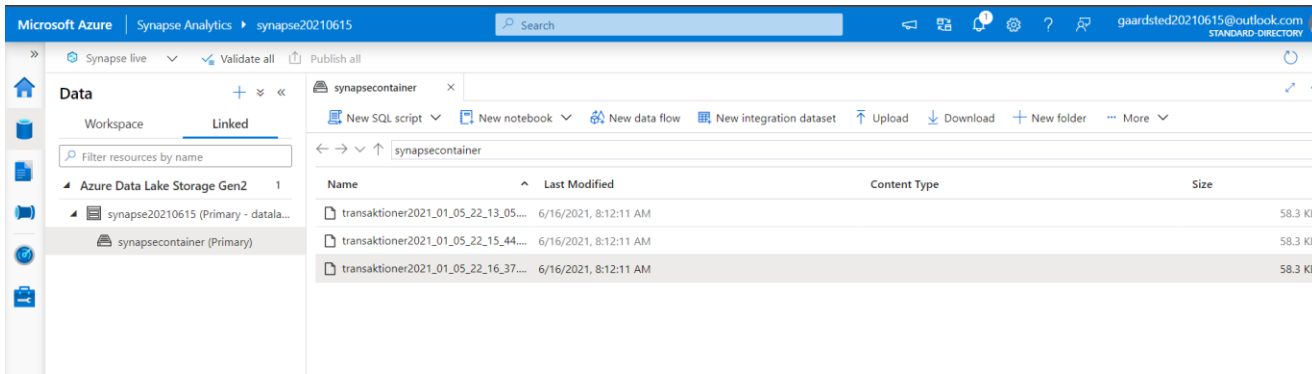
All 3 can contain databases



Azure Synapse Studio

Via url from portal or <https://web.azuresynapse.net>

Generate OPENROWSET sql for datalake files



Azure Synapse Analytics - Develop

The screenshot displays the Azure Synapse Analytics 'Develop' environment. The left sidebar contains a 'Develop' section with a search bar and a list of resources: Power Query (Preview) with 2 items, SQL scripts with 5 items, Notebooks with 3 items, Apache Spark job definitions with 1 item, and Power BI. The main workspace shows a notebook titled 'Notebook 5' with a code cell. The code cell contains the following Python code:

```
1 %%pyspark
2 blob_account_name = "gaardsted20210105std"
3 blob_container_name = "corona"
4 from pyspark.sql import SparkSession
5
6 sc = SparkSession.builder.getOrCreate()
7 token_library = sc._jvm.com.microsoft.azure.synapse.tokenlibrary.TokenLibrary
8 blob_sas_token = token_library.getConnectionString("AzureBlobStorage1")
9
10 spark.conf.set(
11     'fs.azure.sas.%s.%s.blob.core.windows.net' % (blob_container_name, blob_account_name),
12     blob_sas_token)
13 df = spark.read.load('wasbs://corona@gaardsted20210105std.blob.core.windows.net/Test_pos_over_tir
14 ## If header exists uncomment line below
15 ##, header=True
16 )
17 display(df.limit(10))
```

Below the code cell, a message states: 'Failed to create session for user cg after trying for 424ms'. The top of the interface shows the Microsoft Azure logo, the Synapse Analytics workspace name 'cgsynapse', and a search bar. The top right corner displays the URL: <https://web.azuresynapse.net/en-us/authoring/analyze/notebooks/Notebook%205?workspace=%2Fsubscriptions%2F7de49188-ca51-4...>

Serverless Pool

Serverless SQL pool

A workspace has a Built-in serverless pool

- <workspace name>-**ondemand**.sql.azuresynapse.net

Query data **directly** in the primary **data lake** without loading it

- OPENROWSET (e.g. script in Synapse Studio)

Contains **ONLY Metadata objects** e.g views and external tables

Storage account authentication via Azure AD or (database) credentials unless they are public

Pay per use model for queries you run

Tables in Spark databases are automatically visible, and they can be queried by **serverless** SQL pool.

Serverless pool - Databases

Master database with logins (SQL/Azure AD authentication)

No user defined views and external tables here

User defined databases with **meta data**:

CREATE TABLE <table_name> is not supported

CREATE EXTERNAL TABLE is supported and Views

USE - switches database context

CROSS database queries are allowed

Max 20 databases pr workspace

Max query duration 30 minutes

The screenshot shows the Azure SQL Serverless pool interface. At the top, there are two dropdown menus: 'Connect to' with a green checkmark and 'Built-in' selected, and 'Use database' with 'master' selected. Below the 'Use database' dropdown, a list of databases is visible: 'master', 'serverlessDB', and 'ServerLessDB2'. Below this, there are two tabs: 'Dedicated SQL pool' and 'Serverless SQL pool', with the latter being selected. At the bottom, there is a SQL query editor with the text 'syntaxsql' and a SQL command: 'CREATE DATABASE database_name [COLLATE collation_name] [;]'.

Serverless SQL pool

Read data data from files stored on Azure Storage or CosmosDB
via

- OPENROWSET function
- External TABLE in user defined database

OPENROWSET function

OPENROWSET reads **content of file(s)** in a remote data source and returns the content as a set of rows

The datasource is an **Azure Storage account container**

Is referenced in the **FROM** clause as a table

Storage path supports **wildcards**

Use directly with url OR use a DATA_SOURCE

Serverless pool

OPENROWSET to read from

- csv
- Parquet
- Json
- Delta Lake
- Cosmosdb container analytical store

Mapping between parquet datatypes and sql datatypes

Serverless Pool - OPENROWSET

```
OPENROWSET BULK 'unstructured_data_path'  
, [DATA_SOURCE = <data source name>]  
, FORMAT= 'PARQUET' | 'DELTA' | 'CSV'
```

Read data from blob storage/datalake file or folder with wildcards
Data source definition may contain a database scoped credential

Serverless pool - security

Permission levels to read external data from Data Lake

- Storage: Permission to read from Data Lake (Azure)
- SQL:Permission to OPENROWSET (SQL serverless pool)

Azure permissions to a storage account via

- AD principal
- SAS
- Managed Identity for the Synapse workspace
- Anonymous access

Permission is given by a SQL (database) credential

Parquet data storage format

- Column oriented data storage format
- Efficient data compression

Column names and data types are automatically read from Parquet files in Synapse.

Type mappings

Synapse – filename and filepath functions

```
-- filepath function
-- filename function
-- filepath 1 - det er første wildcard
select TOP 100 T.filename()           AS csvFileName
|      |      |      |      ,T.filepath()           AS FullFilePath
|      |      |      |      ,T.filepath(1)          AS YearFromFileName
FROM
|      |      |      |      OPENROWSET(
|      |      |      |      BULK 'https://datalakesu20220406.dfs.core.windows.net/raspberry500/sensor=1984/year=2022/month=04/data\*\_\*\_\*\_\*\_\*\_.csv',
|      |      |      |      FORMAT = 'CSV',
|      |      |      |      PARSER_VERSION = '2.0',
|      |      |      |      HEADER_ROW = TRUE
|      |      |      |      ) AS T
WHERE T.filepath(1) = 2022
```

Serverless pool – filepath function

Filepath function

- When called without a parameter, it returns the full file path that the row originates from. When DATA_SOURCE is used in OPENROWSET, it returns path relative to DATA_SOURCE.
- When called with a parameter, it returns part of the path that matches the wildcard on the position specified in the parameter. For example, parameter value 1 would return part of the path that matches the first wildcard

Spark Pool

Azure Synapse - Spark pool

Node size

- Small (4 vCores/32 GB)
- Medium (8 vCores/64 GB)
- Large (16 vCores / 128 GB)
- Xlarge (32 vCores / 256 GB)
- XXLarge (64 vCores / 512 GB)

Number of nodes (3 to 200)

- Fixed number
- Autoscale (between min max)

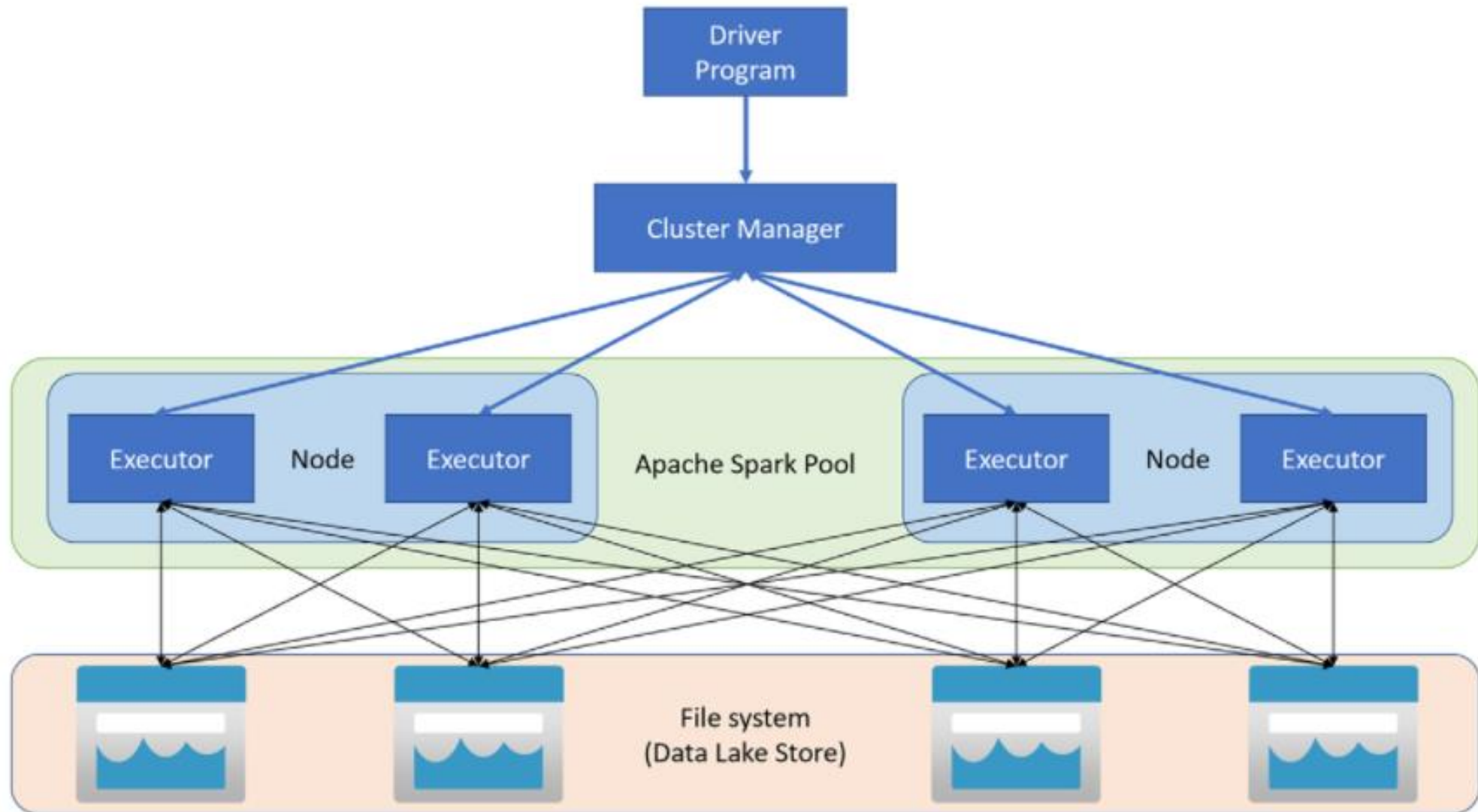
Auto-pause after #Idle minutes

Apache Spark version

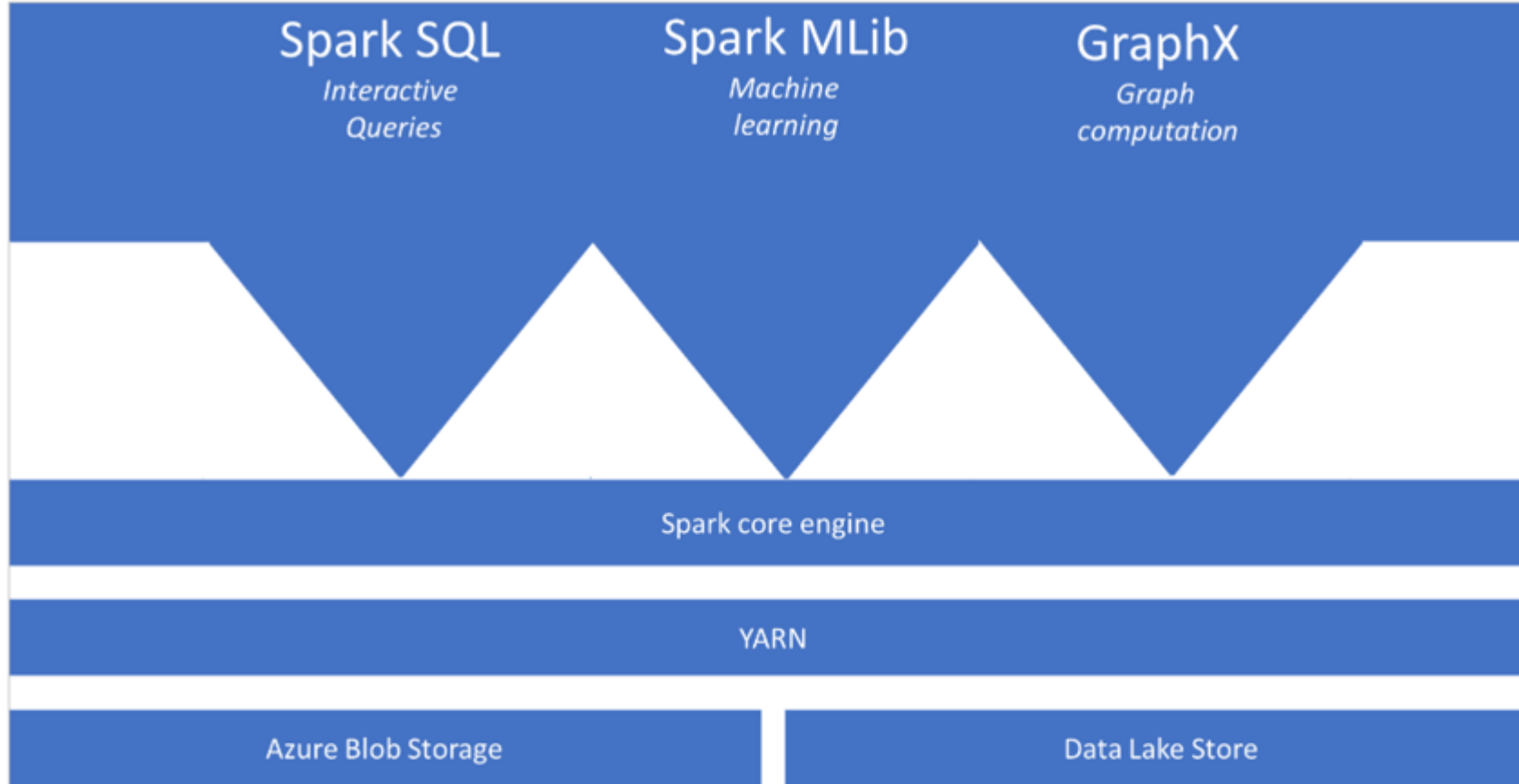
Environment packages

Spark configuration

How do Apache Spark pools work in Azure Synapse Analytics



Introduction to big data engineering with Apache Spark in Azure Synapse Analytics



Spark pool

- Run scripts in notebooks
- You can mix multiple languages in a notebook
- Use temporary table to reference data using different language
- Read data from Data Lake/dedicated pool
- Write data to Spark database/dedicated pool
- Process/load/transform data
- REST API for handling jobs remotely
- Spark session can be configured

Language	Magic command
Python	%%pyspark
Scala	%%spark
SparkSQL	%%sql
Spark C#	%%csharp

Spark Pool – Databaser og tabeller

Databases are called **Lake databases**

A default database exists and is called **default**

User defined databases can be created

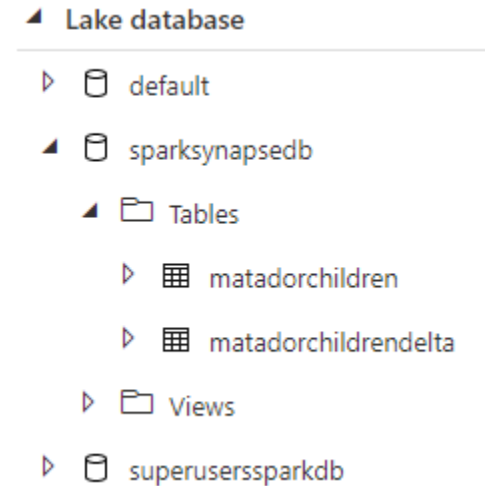
CREATE Schema gives a database...

```
CREATE {DATABASE | SCHEMA} [ IF NOT EXISTS ] database_name  
[ COMMENT database_comment ]  
[ LOCATION database_directory ]
```

Database tables are stored in the Synapse storage account container

Metadata for Spark databases is synchronized async to **serverless pool** databases with the same names

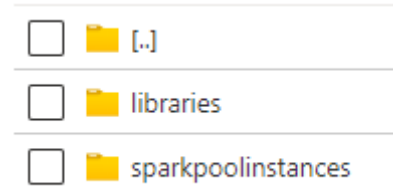
- **csv** and **parquet** external tables are available
- Spark pool can be shut down and you can still query the tables
- DML Operations are not supported with external tables from Serverless Pool



Synapse – ADLS Gen2 storage account

An Azure Synapse workspace uses a default storage container for

- Storing the backing data files for Spark tables
- Execution logs for Spark jobs
- Managing libraries that you choose to install




Spark Pool – RBAC Roles

RBAC role

+ storage account permissions

Synapse Contributor ⓘ

☐  BI Group

Add role assignment

Grant others access to this workspace by assigning roles to users, groups, and/or service principals.
[Learn more](#) ⓘ

Scope * ⓘ

☒ Workspace ☐ Workspace item

Role * ⓘ

Select a role ▾

Filter...

- Synapse Administrator ⓘ
- Synapse SQL Administrator ⓘ
- Synapse Apache Spark Administrator ⓘ
- Synapse Contributor ⓘ
- Synapse Artifact Publisher ⓘ
- Synapse Artifact User ⓘ
- Synapse Compute Operator ⓘ
- Synapse Monitoring Operator ⓘ

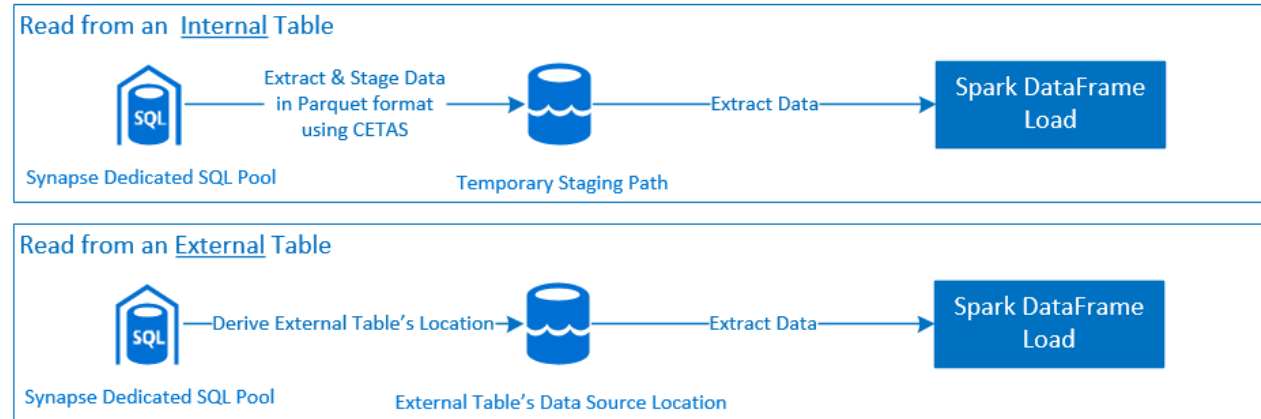
Azure Synapse dedicated pool Connector

Parallel connection between Spark pool and Dedicated pool to read/write large datasets

Implemented in Scala only (python requires a workaround)

USE Azure AD or SQL Server authentication for Dedicated pool

Azure Synapse dedicated pool Connector - Read

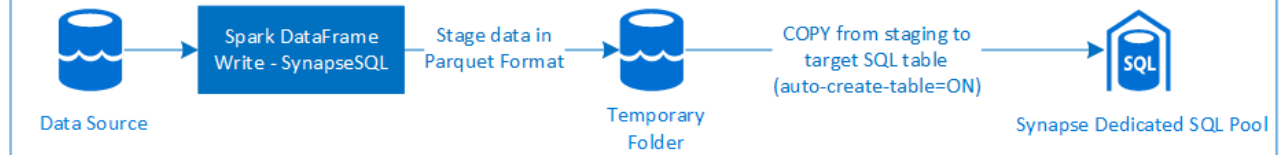


```
EXEC sp_addrolemember 'db_exporter', [<your_domain_user>@<your_domain_name>.com];
```

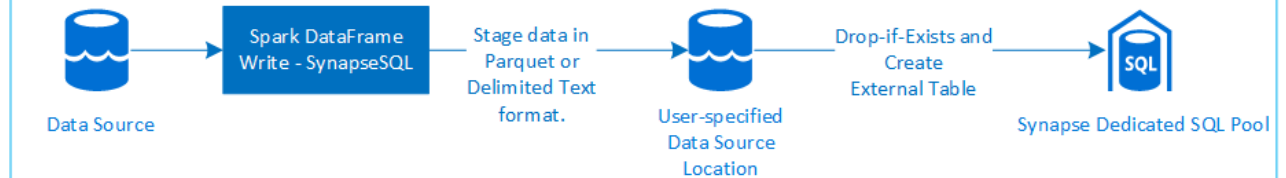
- db_exporter role in Dedicated pool

Azure Synapse dedicated pool Connector - Write

Write to an Internal Table Type



Write to an External Table Type

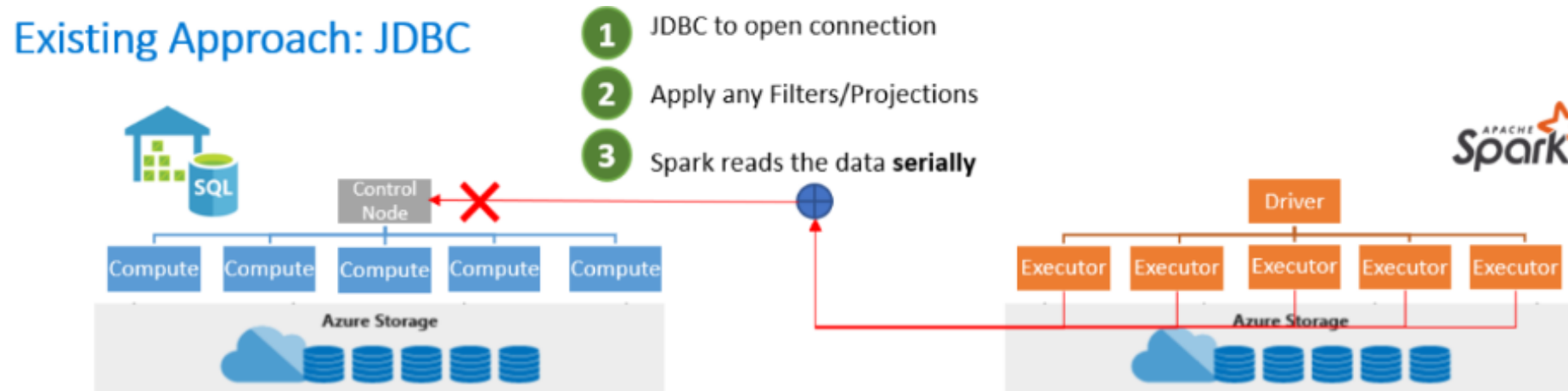


```
--Make sure your user has the permissions to CREATE tables in the [dbo] schema
GRANT CREATE TABLE TO [<your_domain_user>@<your_domain_name>.com];
GRANT ALTER ON SCHEMA::<target_database_schema_name> TO [<your_domain_user>@<your_domain_name>.com];

--Make sure your user has ADMINISTER DATABASE BULK OPERATIONS permissions
GRANT ADMINISTER DATABASE BULK OPERATIONS TO [<your_domain_user>@<your_domain_name>.com];

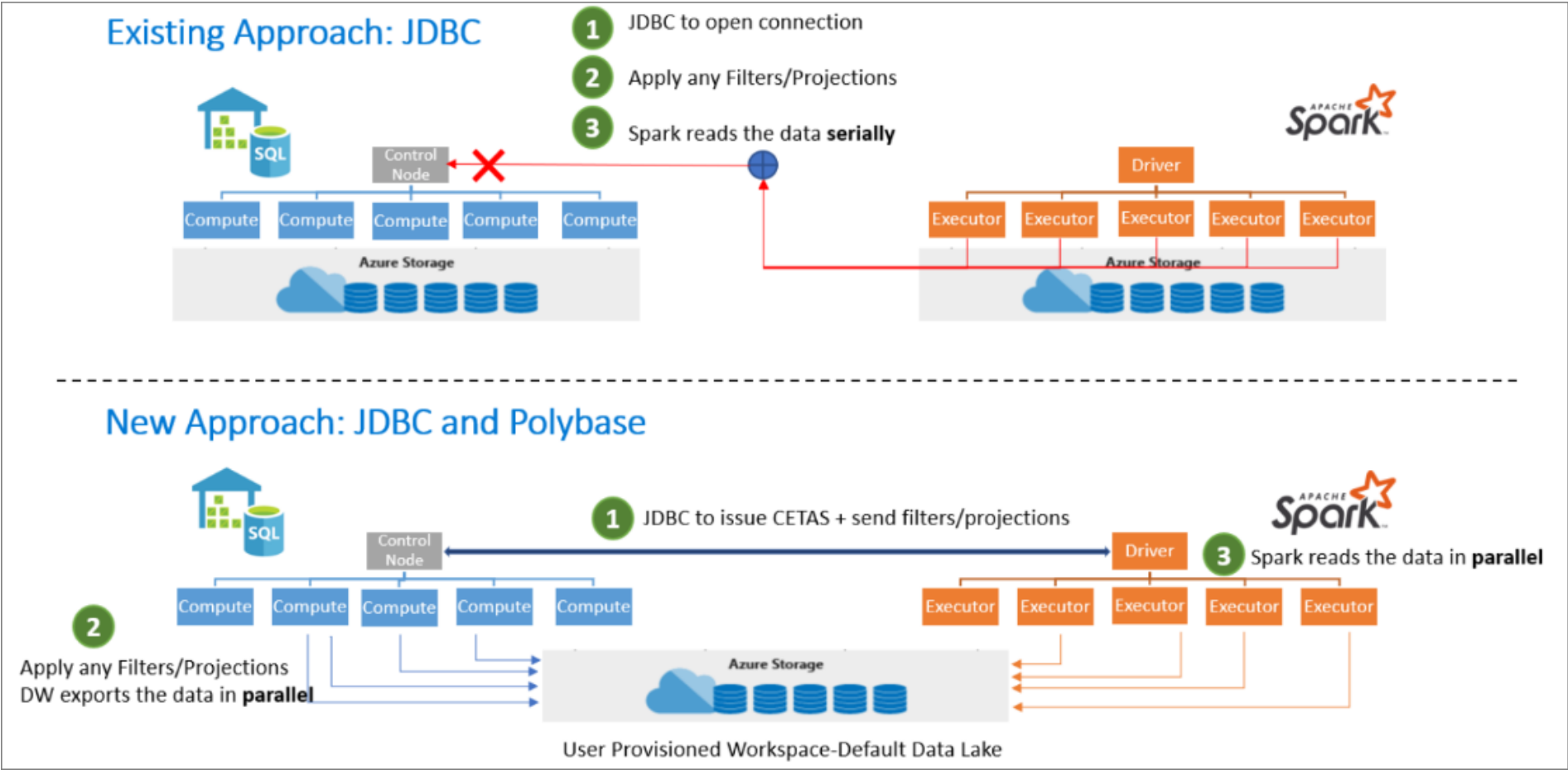
--Make sure your user has INSERT permissions on the target table
GRANT INSERT ON <your_table> TO [<your_domain_user>@<your_domain_name>.com]
```

Integrate SQL and Apache Spark pools in Azure Synapse Analytics

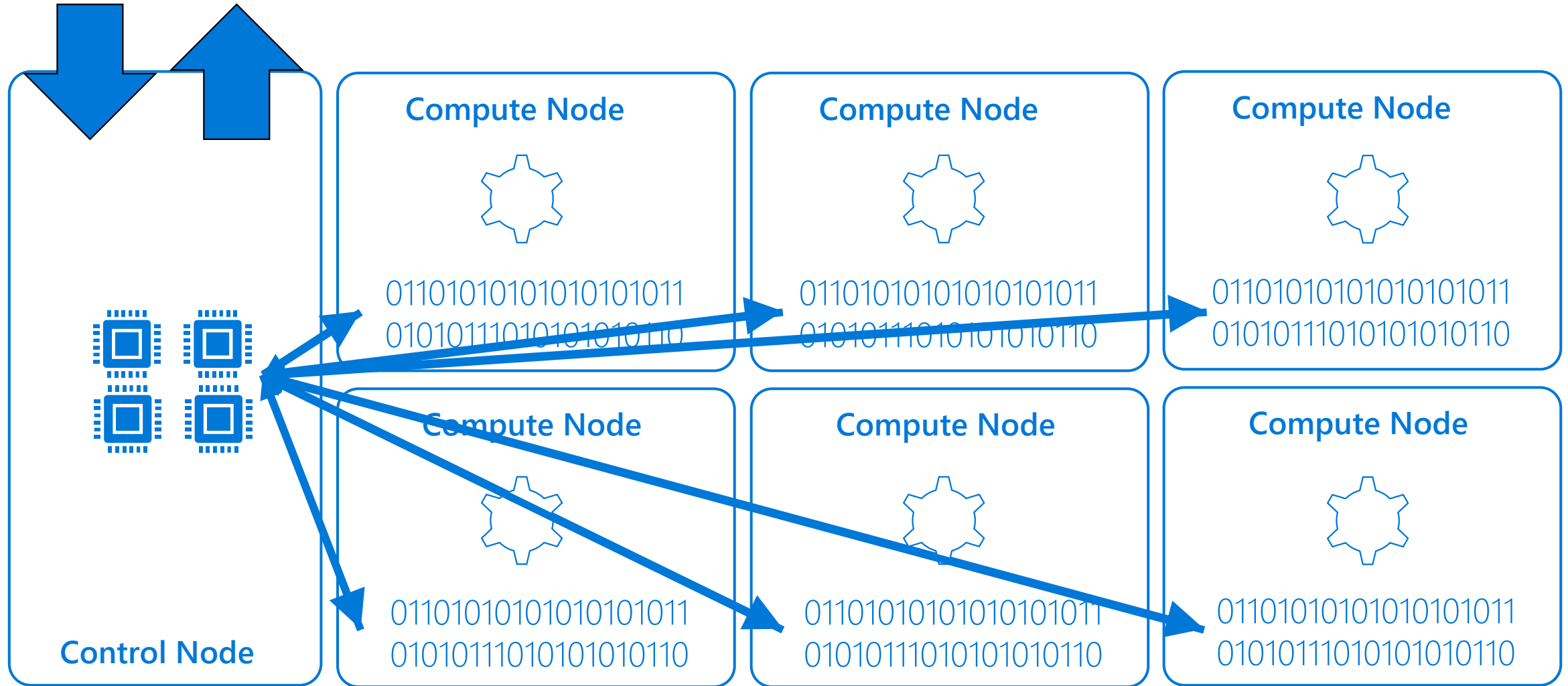


The Azure Synapse Apache Spark to Synapse SQL connector is designed to efficiently transfer data in parallel

Azure Synapse Apache Spark to Synapse SQL connector (Scala only)



Dedicated SQL Pool architecture revision



MS Relationelle database muligheder i 2022

Azure SQL Database

Platform as a Service (PaaS) =>

Altid **nyeste** SQL version

Azure AD

INGEN Jobagent

Elastic pool

INGEN classic backup/restore



Azure SQL VM

OS: Windows Server/Linux

SQL Server 2017+

Windows AD

Jobagent

SQL Server IaaS extension

Always On availability



Managed Instance (MI)

Platform as a Service (PaaS) =>

Altid **nyeste** SQL version

Azure AD

Jobagent

Classic backup/restore supported

Backup kan restores på SQL Server 2022!!!



On premise løsning

OS: Windows Server/Linux

SQL Server 2017+

Windows AD

Jobagent

Always On availability



Dedicated Pool

Parallel DW timeline

- 2010 Parallel DW on premise
 - 2015 Parallel DW in Azure (on logical sql server)
 - 2019 Synapse Analytics (formerly Parallel DW)
 - 2020 Synapse Analytics
-
- NB Dedicated pool still exists as a standalone on a logical sql server



Dedicated SQL pool
(formerly SQL DW)



Azure Synapse Analytics

Dedicated pool – Service levels: cDWUs

Service Levels

The service levels range from DW100c to DW30000c.

Performance level	Compute nodes	Distributions per Compute node	Memory per data warehouse (GB)
DW100c	1	60	60
DW200c	1	60	120
DW300c	1	60	180
DW400c	1	60	240
DW500c	1	60	300
DW1000c	2	30	600
DW1500c	3	20	900
DW2000c	4	15	1200
DW2500c	5	12	1500
DW3000c	6	10	1800
DW5000c	10	6	3000
DW6000c	12	5	3600
DW7500c	15	4	4500
DW10000c	20	3	6000
DW15000c	30	2	9000
DW30000c	60	1	18000

Azure Synapse Pool

Pool name

Performance level: Minimum DW100c

Scale up/down when needed

Pause to save money (**Very important!!!**)

Dedicated pool – Workload group

- System defined workload groups and roles
 - Dynamic groups
 - Static groups
- **Smalrc** is the **default workload group** for all queries
 - Performance gets worse when scaling up!!!

To solve this either:

- Assign a user/role to a system defined workload group
- Create a custom workload group and assign a user/role to it

Dedicated pool – Static resource classes

Service Level	Maximum concurrent queries	Concurrency slots available	Slots used by staticrc10	Slots used by staticrc20	Slots used by staticrc30	Slots used by staticrc40	Slots used by staticrc50	Slots used by staticrc60	Slots used by staticrc70	Slots used by staticrc80
DW100c	4	4	1	2	4	4	4	4	4	4
DW200c	8	8	1	2	4	8	8	8	8	8
DW300c	12	12	1	2	4	8	8	8	8	8
DW400c	16	16	1	2	4	8	16	16	16	16
DW500c	20	20	1	2	4	8	16	16	16	16
DW1000c	32	40	1	2	4	8	16	32	32	32
DW1500c	32	60	1	2	4	8	16	32	32	32
DW2000c	48	80	1	2	4	8	16	32	64	64
DW2500c	48	100	1	2	4	8	16	32	64	64
DW3000c	64	120	1	2	4	8	16	32	64	64
DW5000c	64	200	1	2	4	8	16	32	64	128
DW6000c	128	240	1	2	4	8	16	32	64	128
DW7500c	128	300	1	2	4	8	16	32	64	128
DW10000c	128	400	1	2	4	8	16	32	64	128
DW15000c	128	600	1	2	4	8	16	32	64	128
DW30000c	128	1200	1	2	4	8	16	32	64	128

Dynamic resource classes – memory allocation

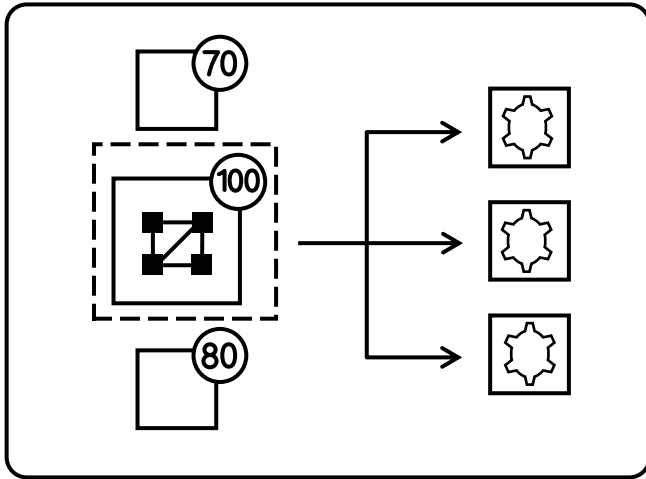
Service Level	smallrc	mediumrc	largerc	xlargerc
DW100c	25%	25%	25%	70%
DW200c	12.5%	12.5%	22%	70%
DW300c	8%	10%	22%	70%
DW400c	6.25%	10%	22%	70%
DW500c	5%	10%	22%	70%
DW1000c to DW30000c	3%	10%	22%	70%

Dynamic resource classes

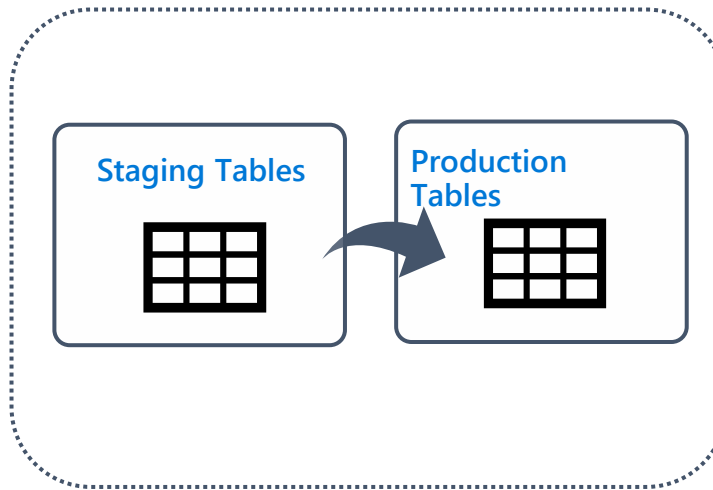
Service Level	Maximum concurrent queries	Concurrency slots available	Slots used by smallrc	Slots used by mediumrc	Slots used by largerc	Slots used by xlargerc
DW100c	4	4	1	1	1	2
DW200c	8	8	1	1	1	5
DW300c	12	12	1	1	2	8
DW400c	16	16	1	1	3	11
DW500c	20	20	1	2	4	14
DW1000c	32	40	1	4	8	28
DW1500c	32	60	1	6	13	42
DW2000c	32	80	2	8	17	56
DW2500c	32	100	3	10	22	70
DW3000c	32	120	3	12	26	84
DW5000c	32	200	6	20	44	140
DW6000c	32	240	7	24	52	168
DW7500c	32	300	9	30	66	210
DW10000c	32	400	12	40	88	280
DW15000c	32	600	18	60	132	420
DW30000c	32	1200	36	120	264	840

Use PolyBase, the Copy command or the Copy Activity

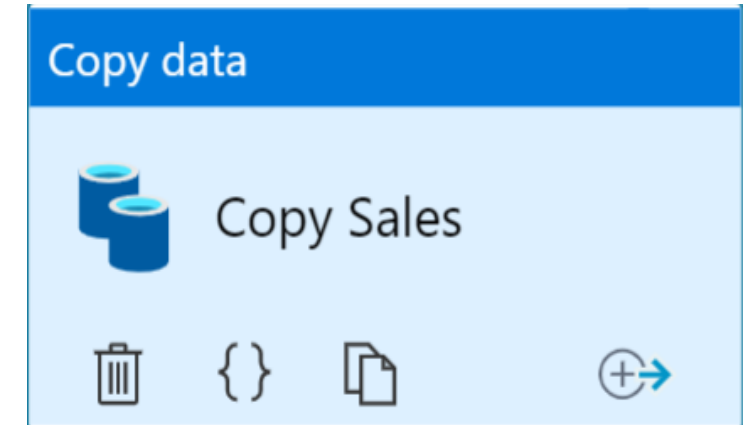
PolyBase



Copy command



Copy data activity



Azure Synapse Analytics Load of data

Storage account and failures

Once or via ADF each night/month etc

BCP / SQL Bulk copy

COPY INTO or

Polybase fastest and scalable way to load data

- Extract the source data into text files.
- Load the data into Azure Blob storage, Hadoop, or Azure Data Lake Store.
- Import the data into SQL Data Warehouse staging tables using PolyBase.
- Transform the data (optional).
- Insert the data into production tables.

Azure Synapse Analytics - Polybase

ELT – Extract Load Transform

Formats: CSV, ORC, Parquet, Gzip, Snappy

Polybase with T-SQL

External table has a table schema

- Like a view it points to data
- Data is stored outside SQL pool

CETAS – Export a resultset

CREATE EXTERNAL TABLE AS SELECT (CETAS)

For dedicated SQL pool or serverless SQL pool

Azure Synapse Analytics - Backup

- A *data warehouse snapshot* creates a restore point you can leverage to recover or copy your data warehouse to a previous state
- A *data warehouse restore* is a **new data warehouse** that is created from a restore point of an existing or deleted data warehouse
- Snapshots of your data warehouse are taken throughout the day creating restore points that are available for seven days.
- Snapshots are not taken when a dedicated SQL pool is paused.
- Create manual snapshots before pausing.
- Dedicated SQL pool supports an **eight-hour** recovery point objective (RPO).
- You can restore your data warehouse in the primary region from any one of the snapshots taken in the past seven days.
- A geo-backup is created once per day to a paired data center (can be disabled to save cost)

Access to Synapse Workspace via RBAC

Access control is found under Manage in the Synapse Workspace
Azure AD users and groups (RBAC – role based access control)

Dedicated pool - Tables

Table distribution

- Round-robin (default)
- Hash (distribution column)
- Replicated

Statistics

- Created automatically
- Not updated automatically (!)

Azure Synapse Analytics - Tables

Table features **not supported** in dedicated SQL pool:

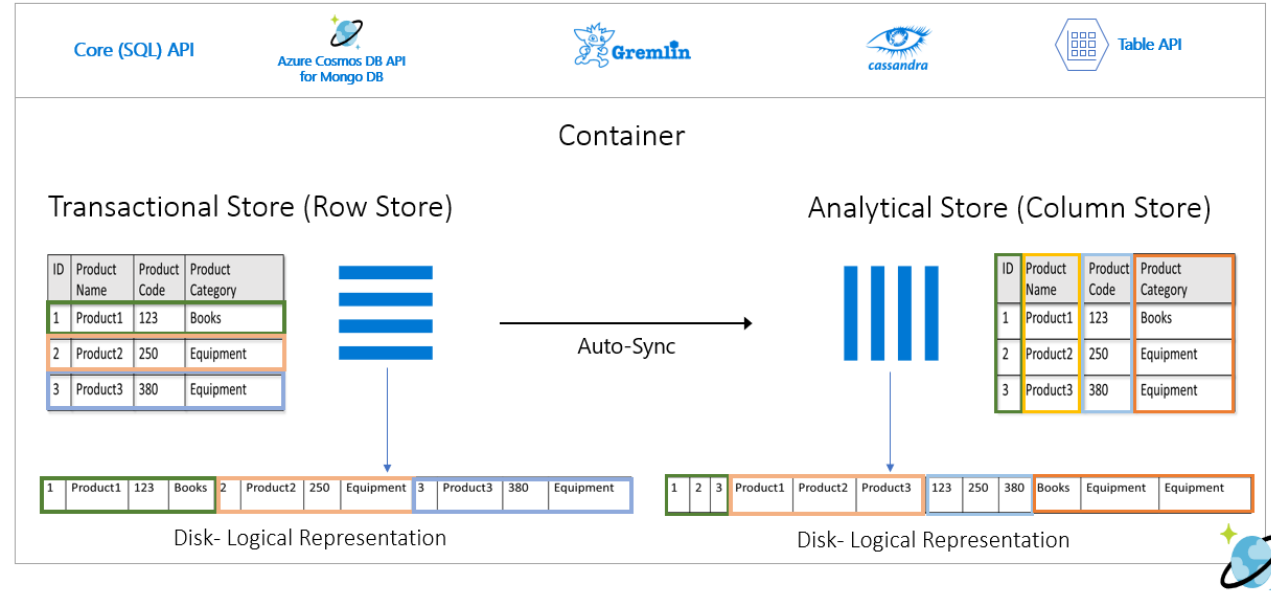
- Foreign key, Check Table Constraints
- Computed Columns
- Indexed Views
- Sequence
- Sparse Columns
- Surrogate Keys. Implement with Identity.
- Synonyms
- Triggers
- Unique Indexes
- User-Defined Types

Azure Cosmos DB analytical store

Only supported for **SQL API** and **MongoDB**

Enable at container level for a **new** container

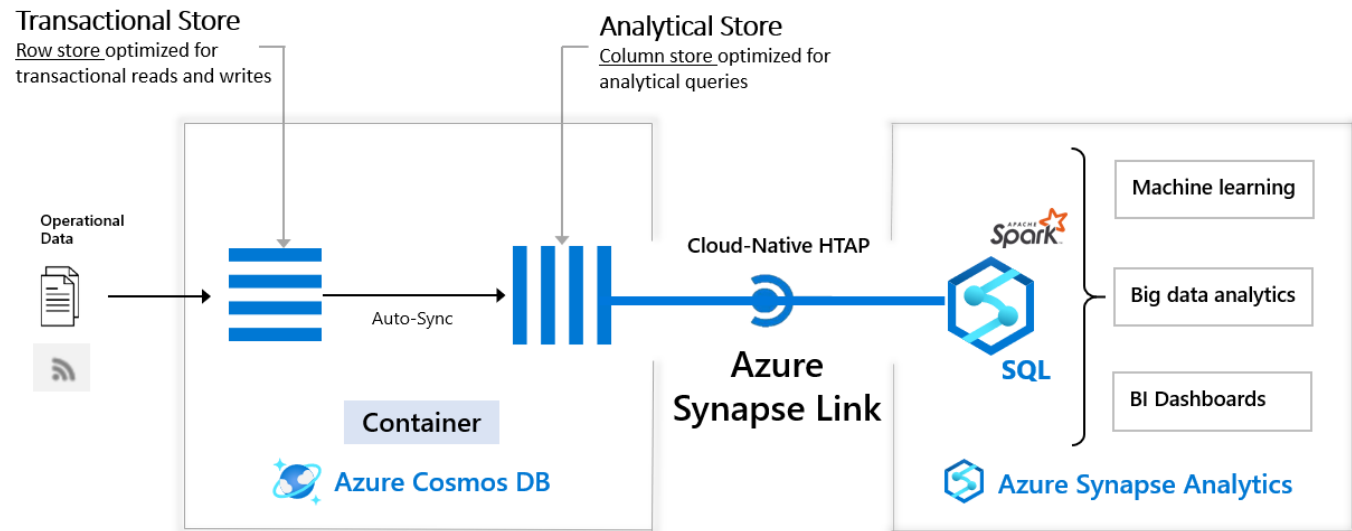
Data is synced and saved twice!



Azure Synapse Link for Cosmos DB

Run near real-time **analytics** over operational data in Azure Cosmos DB

- Spark pool
- Serverless pool



TTL for records in Analytical store

TTL – time to live

Synapse Spark Pool - Read data from
analytical store

Synapse – Managed virtual network

Synapse – Managed private endpoints

Requires a Managed virtual network

Azure AD – conditional access

Synapse Analytics - TDE

Encrypting data at rest (datafiles, translog + backup)

Transparent Data Encryption (TDE)

Default OFF (must be turned on for Azure SQL Database)

Service managed key / BYOK in Azure key vault

Synapse – Spark - TokenLibrary

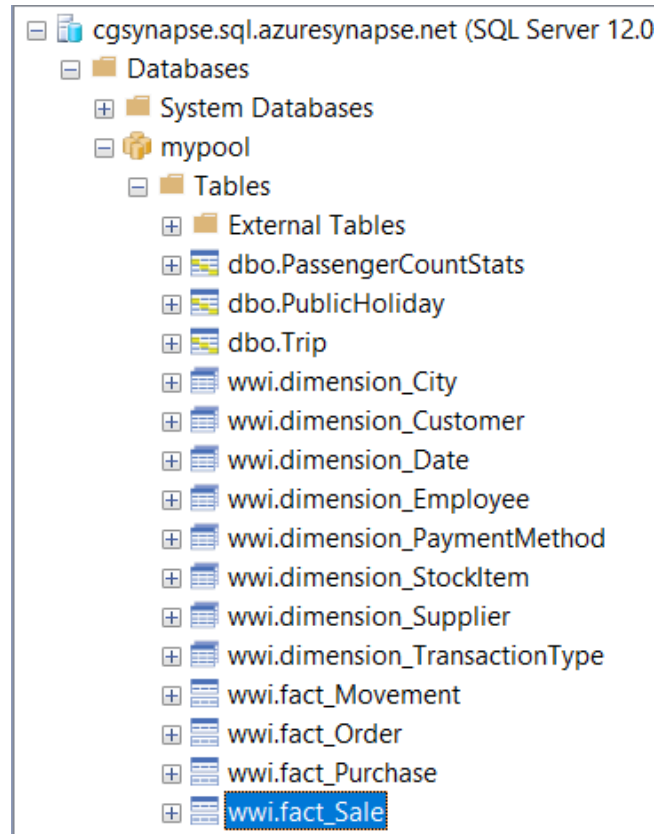
- Apache Spark can reference the linked services from Synapse via the TokenLibrary
- TokenLibrary can also fetch secrets from Azure Key Vault
 - Specify key vault as a connection string
 - Synapse workspace managed identity needs Get secrets permission on vault

Slowly changing dimensions

- Type 1 SCD
- Type 2 SCD

Azure Synapse – Table distribution

- Round robin
- Replicated
- Hash



Synapse - External tables

Synapse – Development endpoint

Used by the workspace web UI as well as DevOps to execute and publish artifacts like SQL scripts, notebook.