

Azure Synapse Analytics

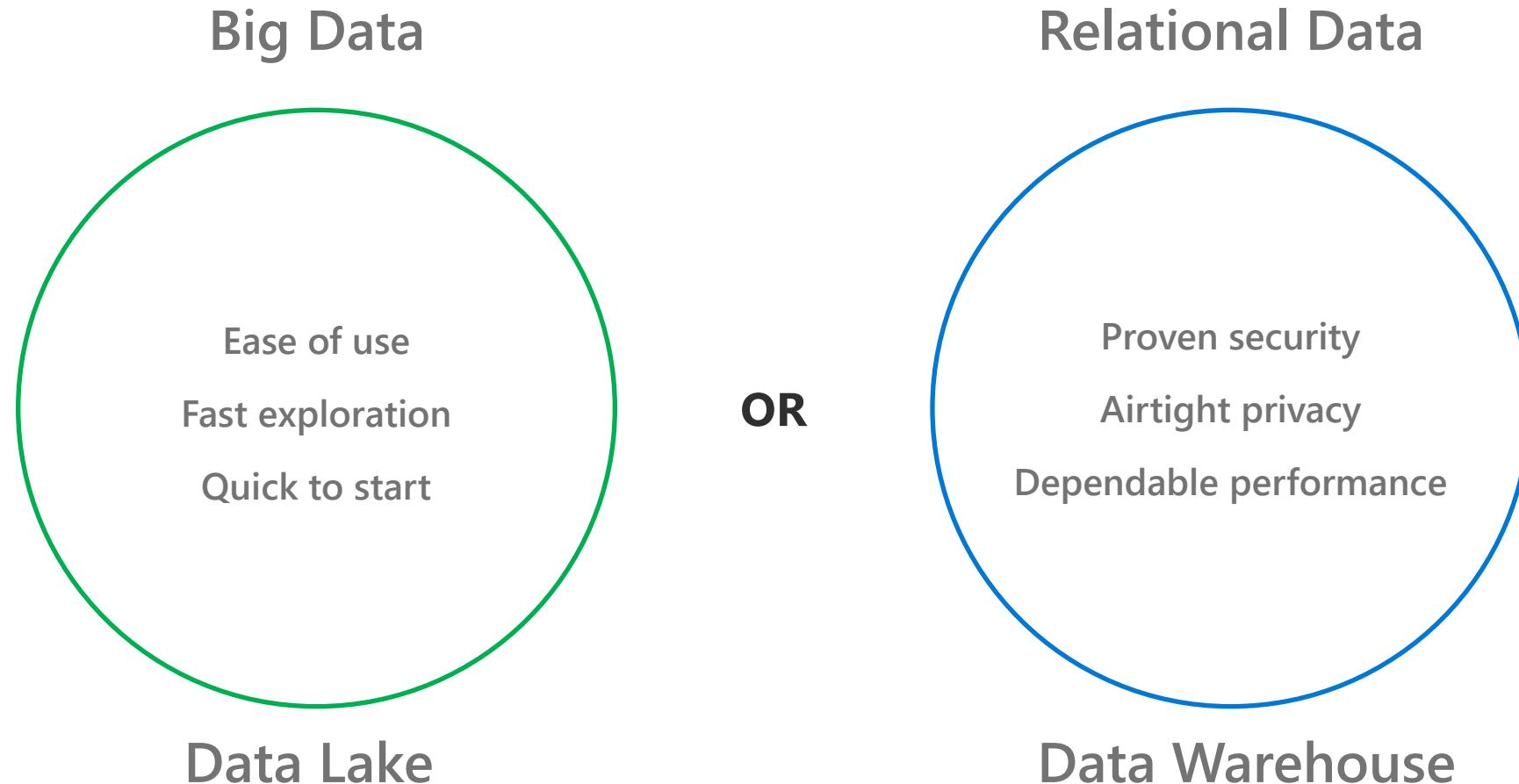


Azure Synapse Analytics

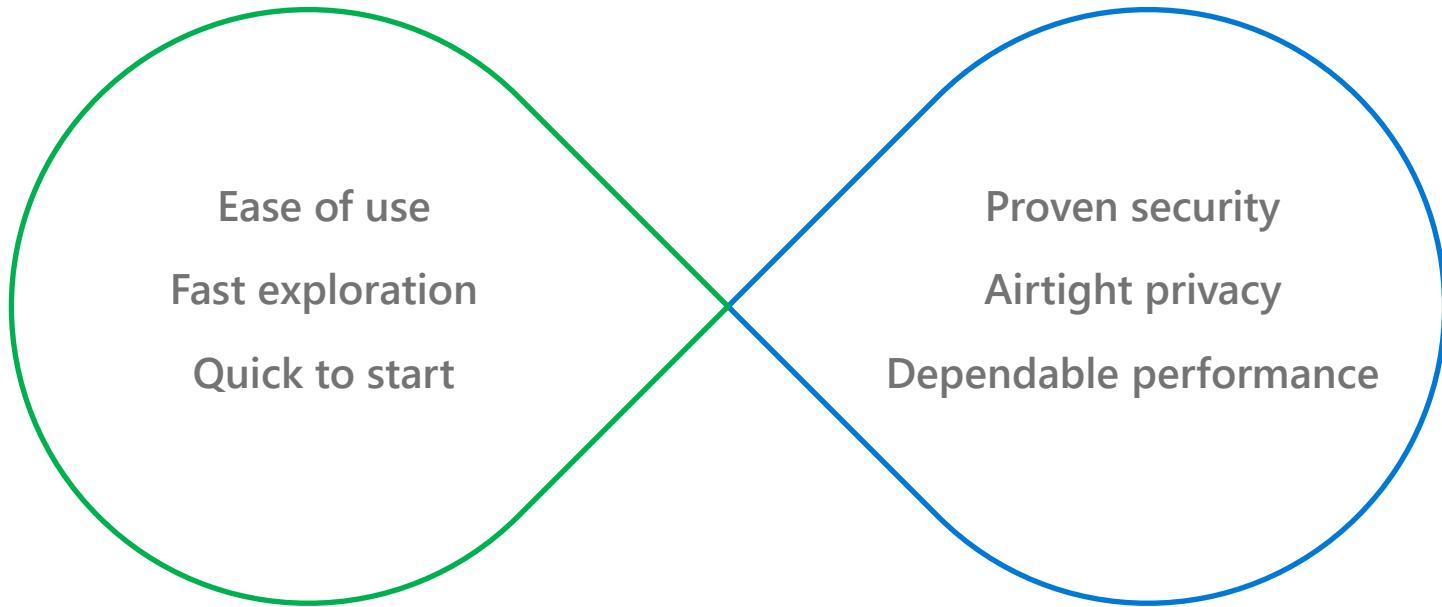
Limitless analytics service with unmatched time to insight

Azure Synapse is Azure SQL Data Warehouse evolved—blending big data, data warehousing, and data integration into a single service for end-to-end analytics at cloud scale.

This is a result of businesses being forced to maintain two critical, yet independent analytics systems



Azure brings these two worlds together, in a single service



Welcome to Azure Synapse Analytics

Azure Synapse Analytics – Unified Platform



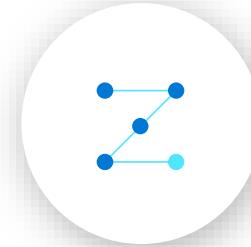
Azure Synapse Analytics



Limitless Scale



Powerful Insights



Unified Experience

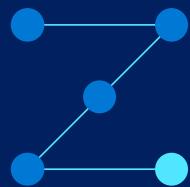


Unmatched Security

Why Microsoft for Analytics



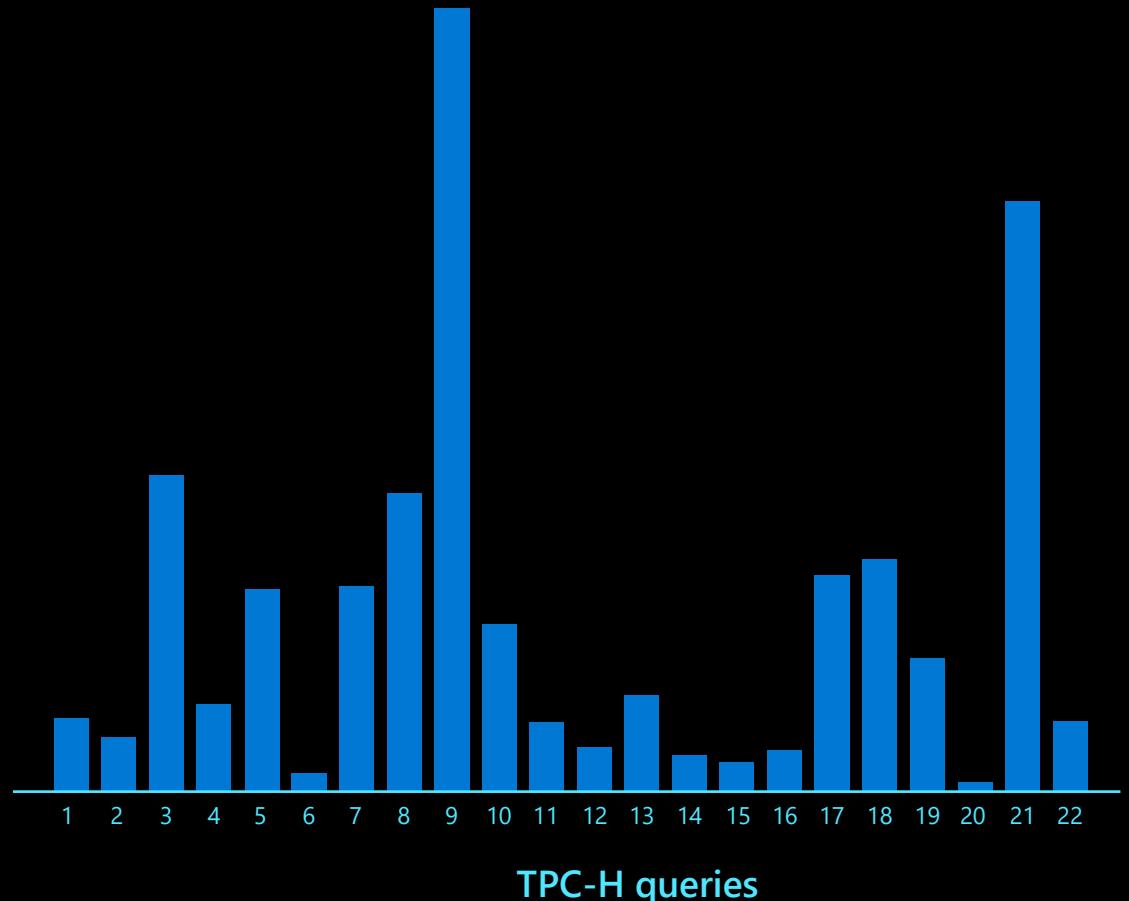
Price-performance



Simplicity

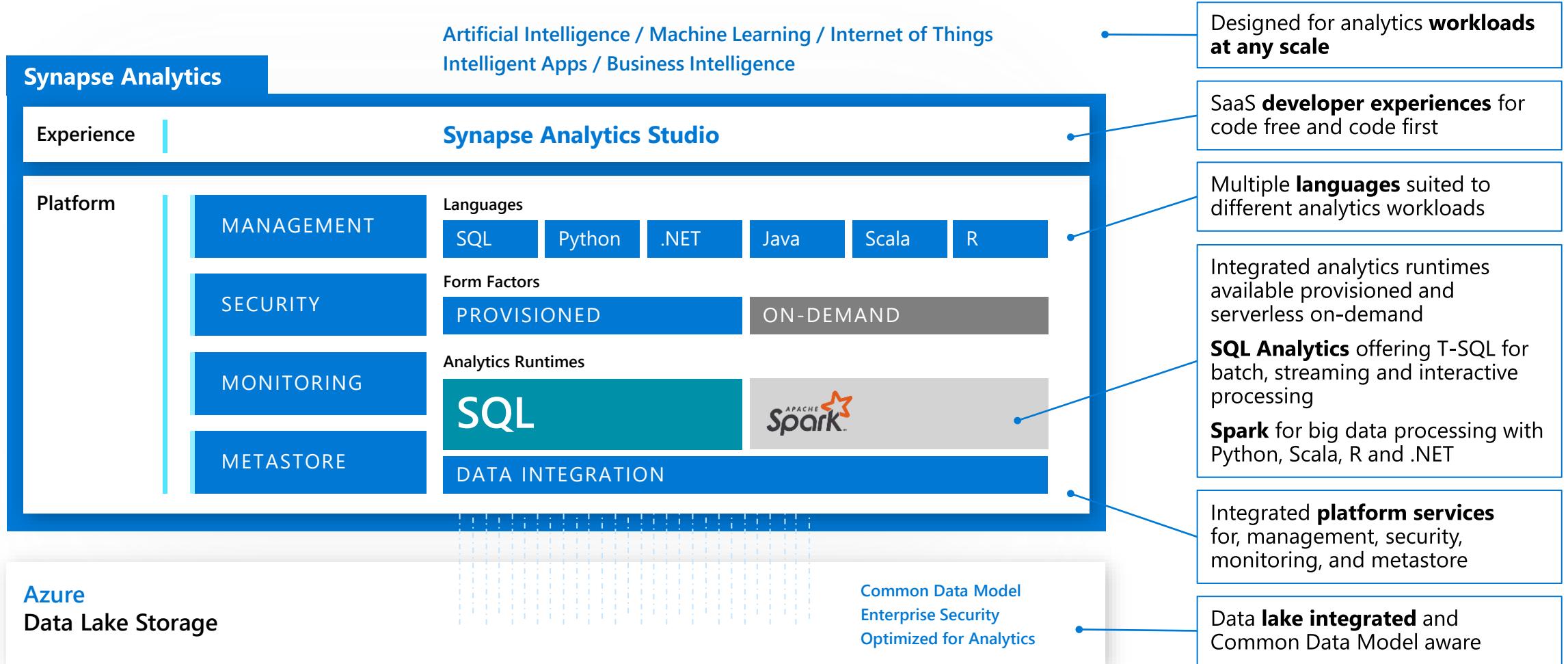
Azure Synapse is the first and only analytics system to have run all TPC-H queries at 1 petabyte-scale

TPC-H 1 Petabyte Query Execution



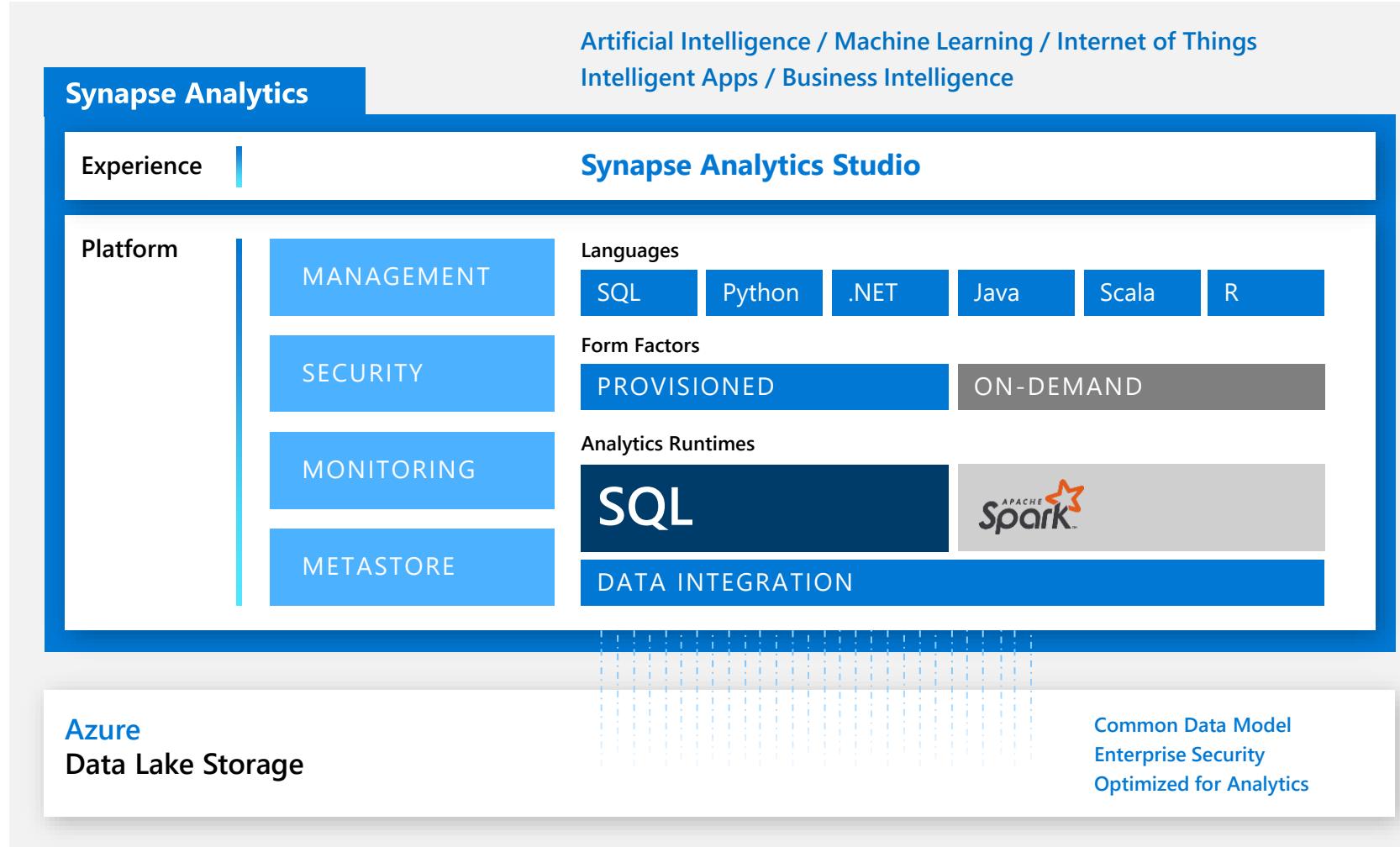
Azure Synapse Analytics

Integrated data platform for BI, AI and continuous intelligence



Azure Synapse Analytics

Integrated data platform for BI, AI and continuous intelligence



Connected Services

- Azure Data Catalog
- Azure Data Lake Storage
- Azure Data Share
- Azure Databricks
- Azure HDInsight
- Azure Machine Learning
- Power BI

3rd Party Integration

Studio

Microsoft Azure | Synapse Analytics | [New](#) | [Logout](#)

Overview Data Develop Orchestrate Monitor Manage

Synapse workspace [New](#)

Ingest Use the copy data tool to import data once or on a schedule.

Explore Learn how to navigate and interact with your data.

Analyze Learn how to use SQL or Spark to get insights from your data.

Visualize Build interactive reports with integrated Power BI capabilities.

Resources

Recent Pinned

No recent resources

Your recently opened resources will show up here.

Useful links

[Synapse Analytics overview](#) Discover the capabilities offered by Synapse and learn how to make the most of them.

[Pricing](#) Learn about pricing details for Synapse capabilities.

[Documentation](#) Visit the documentation center for quickstarts, how-to guides, and references for PowerShell, APIs, etc.

[Give feedback](#) Share your comments or suggestions with us to improve Synapse.

Select another Workspace



Azure Synapse Analytics

Data Integration

Data Hub – Storage accounts

Browse Azure Data Lake Storage Gen2 accounts and filesystems – navigate through folders to see data

ADLS Gen2 Account

Container (filesystem)

The screenshot shows the Microsoft Azure Data Hub interface for managing storage accounts. On the left, a sidebar lists navigation options: Overview, Data (selected), Develop, Orchestrate, Monitor, and Manage. Below this, account details are shown: Name: prlangadws2, Region: West US 2, Resource group: prlangadrg, Subscription ID: 58f8824d-32b0-4825-9825-02fa6a801546. The main area is titled 'Data' and shows a 'Storage accounts' section. A red box highlights the 'prlangaddemosa (Primary)' storage account. Another red box highlights the 'nyctlc' container within it. The 'Filepath' bar shows the path 'nyctlc > yellow'. The right side displays a file browser with a list of folders named 'puYear=2001' through 'puYear=2026', each with a timestamp and 'Folder' type.

NAME	LAST MODIFIED	CONTENT TYPE	SIZE
puYear=2001	10/25/2019, 2:25:03 PM	Folder	
puYear=2002	10/25/2019, 2:25:21 PM	Folder	
puYear=2003	10/25/2019, 2:25:03 PM	Folder	
puYear=2008	10/25/2019, 2:20:38 PM	Folder	
puYear=2009	10/25/2019, 2:19:33 PM	Folder	
puYear=2010	10/25/2019, 2:19:24 PM	Folder	
puYear=2011	10/25/2019, 2:23:56 PM	Folder	
puYear=2012	10/25/2019, 2:20:01 PM	Folder	
puYear=2013	10/25/2019, 2:19:52 PM	Folder	
puYear=2014	10/25/2019, 2:24:06 PM	Folder	
puYear=2015	10/25/2019, 2:20:12 PM	Folder	
puYear=2016	10/25/2019, 2:19:21 PM	Folder	
puYear=2017	10/25/2019, 2:20:28 PM	Folder	
puYear=2018	10/25/2019, 2:24:38 PM	Folder	
puYear=2019	10/25/2019, 2:20:33 PM	Folder	
puYear=2020	10/25/2019, 2:24:47 PM	Folder	
puYear=2021	10/25/2019, 2:28:34 PM	Folder	
puYear=2026	10/25/2019, 2:20:20 PM	Folder	

Develop Hub

Overview

It provides development experience to query, analyze, model data

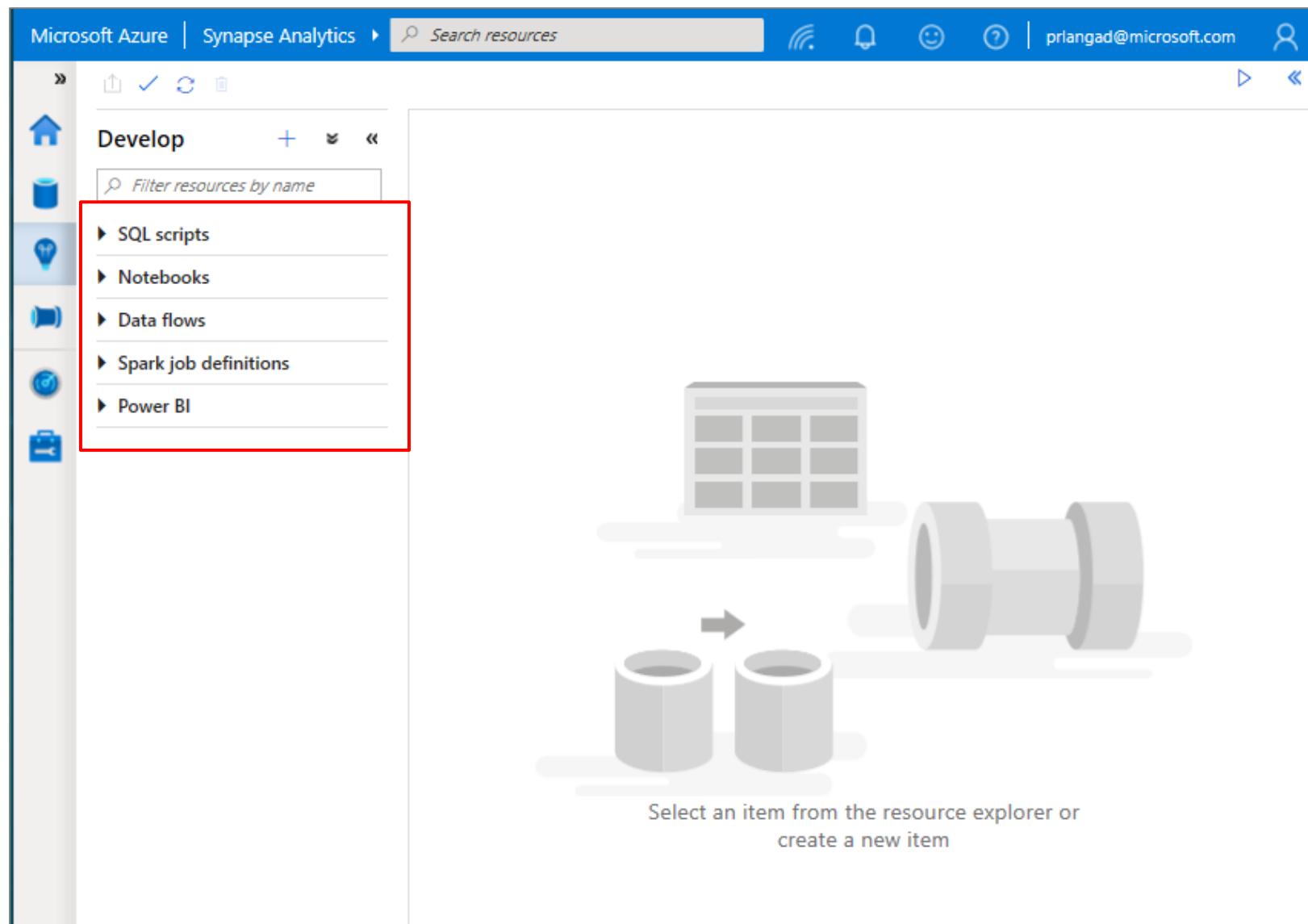
Benefits

Multiple languages to analyze data under one umbrella

Switch over notebooks and scripts without loosing content

Code intellisense offers reliable code development

Create insightful visualizations



Develop Hub – Power BI

Edit reports in Synapse workspace

The screenshot shows the Azure Synapse Analytics Develop Hub interface for Power BI. On the left, the sidebar lists various resources: Notebooks (e.g., AMLAutoMLPredict, AutoML, Data Download_Weather), Data flows (e.g., PrepareCabDataFlow), and Power BI (e.g., SynapseNYTaxiInsights, Power BI Datasets, Power BI Reports). The 'Power BI Reports' section is expanded, showing 'SynapseNYIgnite2019' and 'SynapseNYIgnite2019 (1)'. The main area displays a report titled 'SynapseNYIgnite2019' containing two visualizations: a line chart showing 'Triphives, Greenrides and Yellowrides by DatePickup' from Jan 2015 to Jul 2017, and a bar chart showing 'Trips by holidayName' for various holidays like Veterans Day, Thanksgiving, Martin Luther King Jr. Day, Columbus Day, and Memorial Day. The right side of the screen features the Power BI Visualizations pane, which is currently open for the first visualization. This pane includes sections for Filters, Visualizations, Fields, Axis, Legend, Value, Tooltips, DRILLTHROUGH, and Cross-report. The 'Value' section is set to 'numTrips'. The 'Axis' section is set to 'holidayName'. The 'Fields' pane on the far right lists various data fields such as dimHoliday, dimNYCLocations, Fhv, GreenCab, PredictedValues, vwFhvMarketShare, vwGrnCabMarketS..., vwMarketShareBy..., vwPredictedValues, vwYelCabMarketSh..., weather, YellowCab, and YellowCabTripsHoli... . The 'DRILLTHROUGH' section at the bottom of the Visualizations pane includes checkboxes for 'date', 'holidayName', 'Σ numTrips', and 'year'.

Orchestrator Hub

Overview

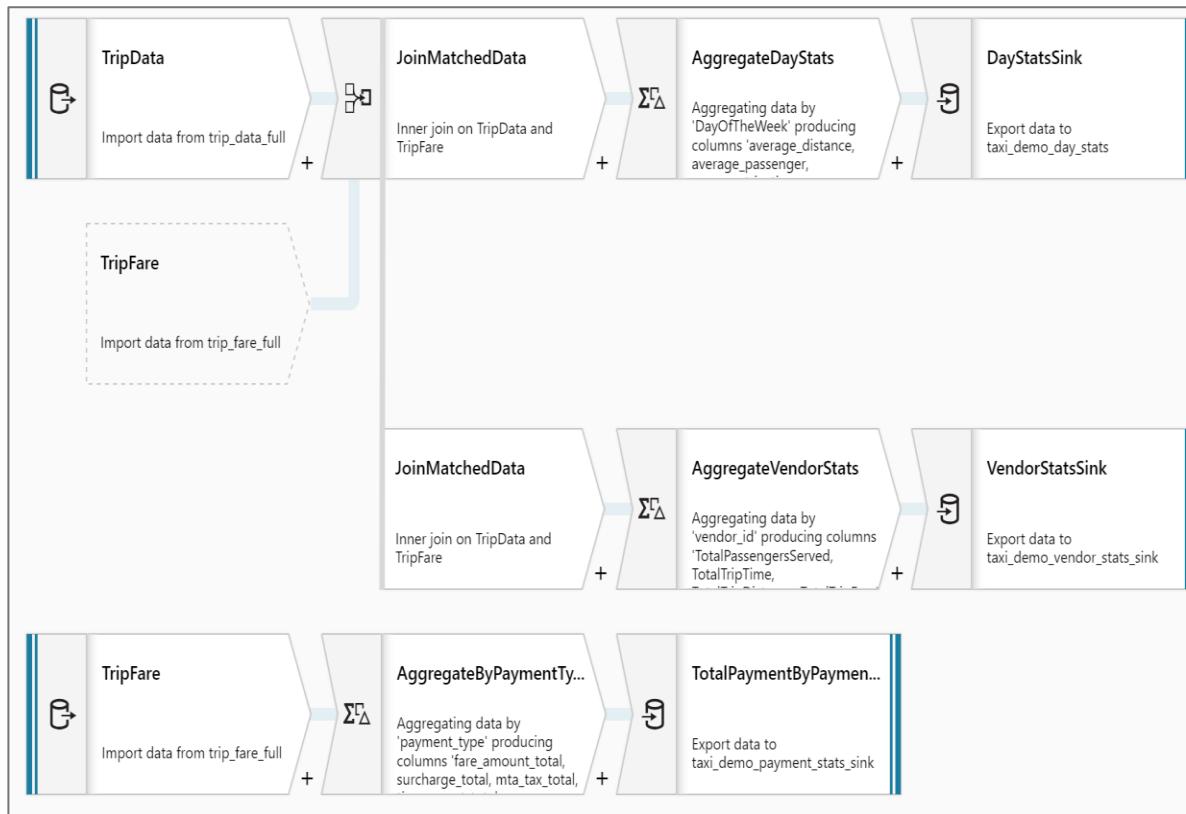
It provides ability to create pipelines to ingest, transform and load data with 80+ inbuilt connectors

The screenshot shows the Microsoft Azure Synapse Analytics Studio Orchestrate Hub. The left sidebar lists 'Orchestrator' resources: 'Pipelines' (Copy Open Dataset, Pipeline 1, Load Data to SQLDW), 'Activities' (Move & Transform, Azure Data Explorer, Azure Function, Batch Service, Data Lake Analytics, Databricks, General, HDInsight, Iteration & Conditionals, Machine Learning, Synapse), and 'SQL script i *'. The main area displays a pipeline named 'Load Data to SQLDW' with a single activity step 'Copy data' which has a sub-step 'WeatherData'. The pipeline details pane at the bottom shows fields for Name (Load Data to SQLDW), Description, Concurrency, and Annotations.

Prep, Transform Data, & Load Data

Mapping Dataflow

Code free data transformation @scale



Wrangling Dataflow

Code free data preparation @scale

CustId	FirstName	LastName	City	ZIP	Email	State	BasePay
1	"Harry"	"Potter"	"Bellevue"	"98004"	"harryk@fabrikam.com"	"WA"	90000
2	"Harry"	"Potter"	"Bellevue"	"98004"	"harryk@fabrikam.com"	"WA"	90000
3	"Hermione"	"Granger"	"Wilmington"	"19801"	"hermione@fabrikam.com"	"DE"	100000
4	"Hermione"	"Granger"	"Wilmington"	"19801"	"gamalloy@fabrikam.com"	"DE"	100000
5	"Lord"	"Voldemort"	"Billings"	"59115"	"lordc@fabrikam.com"	"MT"	110000
6	"Albus"	"Dumbledore"	"Newyork"	"12345"	"albusd@fabrikam.com"	"NY"	120000
7	"Severus"	"Snape"	"Columbus"	"56789"	"severus@fabrikam.com"	"OH"	130000
8	"Draco"	"Malfoy"	"Houston"	"91019"	"dracoh@fabrikam.com"	"TX"	140000
9	"Dobby"	"Elf"	"Salt Lake C...	"11128"	"dobbyz@fabrikam.com"	"UT"	150000
10	"Ron"	"Weasley"	"Las Vegas"	"51527"	"ronag@fabrikam.com"	"NV"	160000
11	"Sirius"	"Black"	"Providence"	"61623"	"hblack@fabrikam.com"	"RI"	170000
12	"Luna"	"Lovegood"	"Kansas City"	"68692"	"lunal@fabrikam.com"	"MO"	180000
13	"Rubeus"	"Hagrid"	"Boston"	"98052"	"gammaloy@fabrikam.com"	"Malfoy"	190000
14	"Bellatrix"	"Lestrange"	"Los Angeles"	"78965"	"mlestrange@fabrikam.com"	"CA"	200000
15	"Ginny"	"Weasley"	"Redmond"	"98052"	"ginnyw@fabrikam.com"	"WA"	210000
16	"Neville"	"Longbottom"	"Bothell"	"98053"	"nevilea@fabrikam.com"	"WA"	220000
17	"Alastor"	"Moody"	"Renton"	"98054"	"albusd@fabrikam.com"	"WA"	230000
18	"Lucius"	"Malfoy"	"Bellevue"	"98004"	"luciusmalfoy@fabrikam.co...	"WA"	240000
19	"Cedric"	"Diggory"	"Seattle"	"98899"	"cedricp@fabrikam.com"	"WA"	250000
20	"Argus"	"Filch"	"Salt Lake C...	"11128"	"argusm@fabrikam.com"	"UT"	260000
21	"Lord"	"Voldemort"	"Billings"	"59115"	"lordc@fabrikam.com"	"MT"	110000
22	"Albus"	"Dumbledore"	"Newyork"	"12345"	"albusd@fabrikam.com"	"NY"	120000
23	"Severus"	"Snape"	"Columbus"	"56789"	"severus@fabrikam.com"	"OH"	130000
24	"Luna"	"Lovegood"	"Bellmore"	"98004"	"lunal@fabrikam.com"	"MA"	00000

Orchestrate @ Scale

Overview

It offers trigger types as - schedule, event, tumbling window. Monitor pipeline runs, control trigger execution.

Benefits

Flexible invocation

Control execution flows, conditional logic

Ability to monitor execution

New trigger

Choose a name for your trigger. This name can be updated at any time until it is published.

Name * Trigger 1

Description

Type * Schedule Tumbling window Event

Start Date (UTC) * 10/30/2019 11:20 PM

Recurrence * Every 1 Minute(s)

End * No End On Date

Annotations + New

Activated * Yes No

OK

Microsoft Azure | Synapse Analytics

External connections
Linked services
Orchestration
Triggers
Integration runtimes
Security
Access control

+ New

NAME ↑	TYPE ↑	STATUS ↑
* CopyParquetDataTrigger	Schedule	Started
* Trigger 1	Schedule	Stopped

Manage – Integration runtimes

Overview

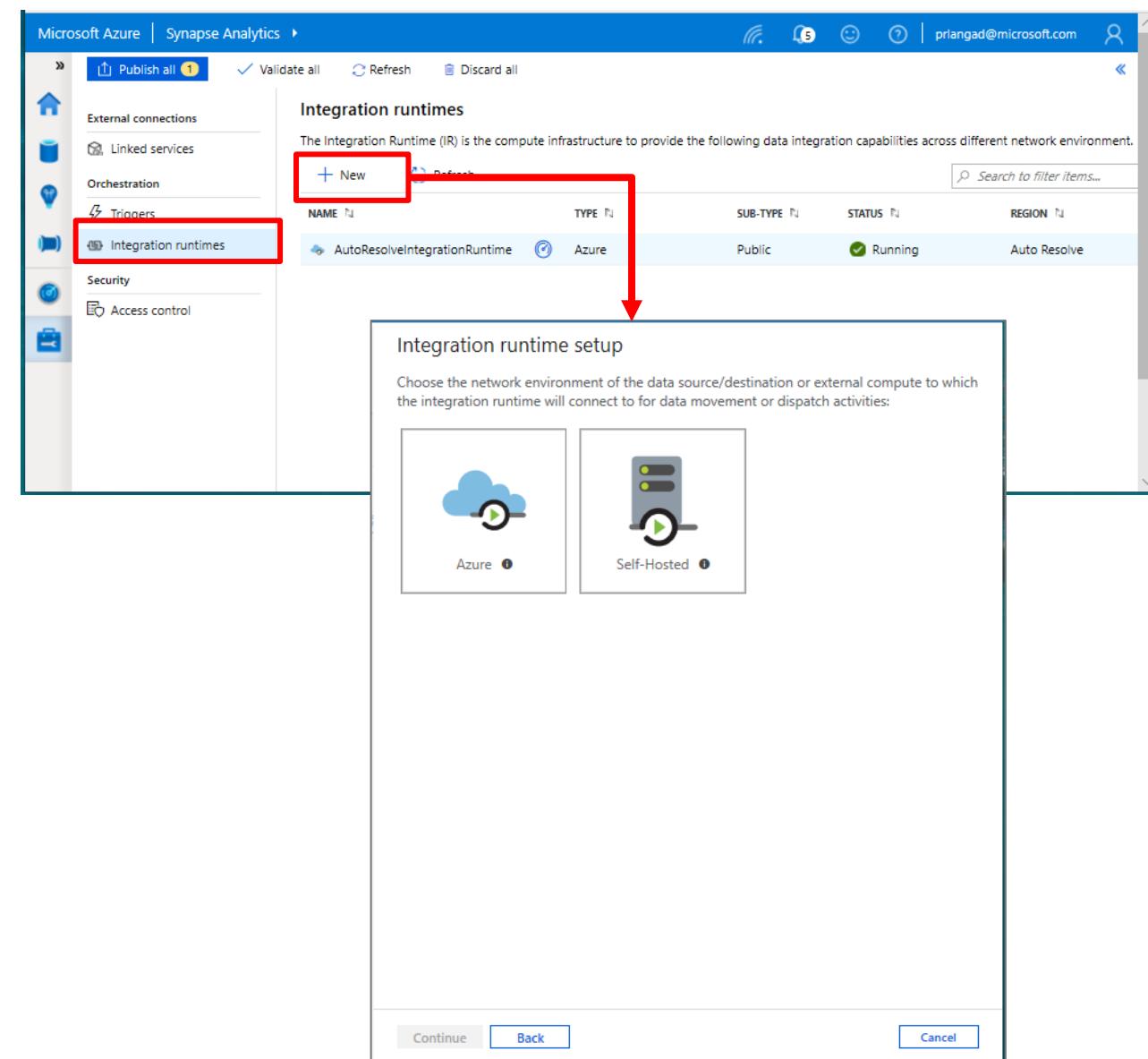
Integration runtimes are the compute infrastructure used by Pipelines to provide the data integration capabilities across different network environments. An integration runtime provides the bridge between the activity and linked services.

Benefits

Offers Azure Integration Runtime or Self-Hosted Integration Runtime

Azure Integration Runtime – provides fully managed, serverless compute in Azure

Self-Hosted Integration Runtime – use compute resources in on-premises machine or a VM inside private network





Azure Synapse Analytics

SQL Analytics

Comprehensive SQL language



Performance optimizations

- Table partitions
- Distributed tables
- Isolation modes
- Materialized Views
- Indexing
- Columnstore index
- Result-set caching

T-SQL Querying

- Windowing aggregates
- Approximate execution (Hyperloglog)
- GROUP BY ROLLUP

Complete SQL object model

- Tables
- Views
- Stored procedures
- Functions
- JSON data support
- and more

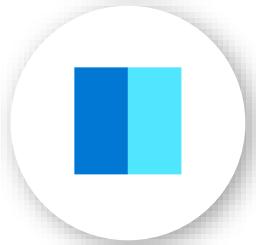
Workload Management



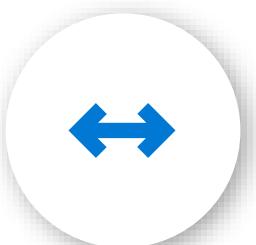
Workload Classification



Workload Importance



Intra-Cluster Isolation



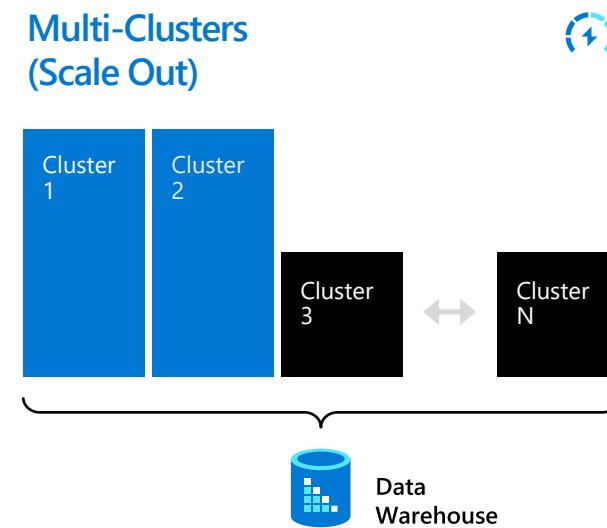
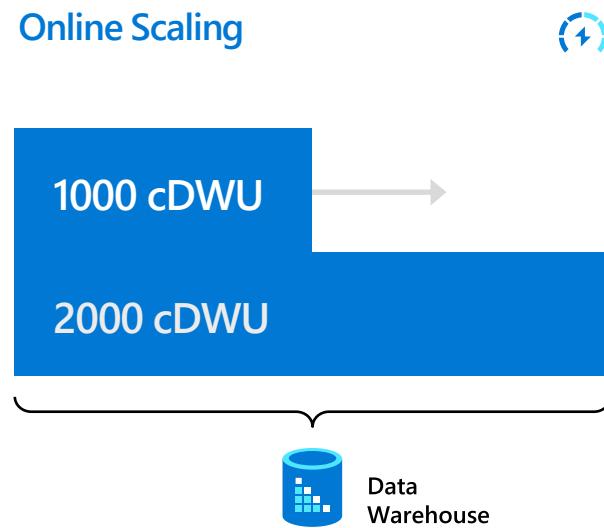
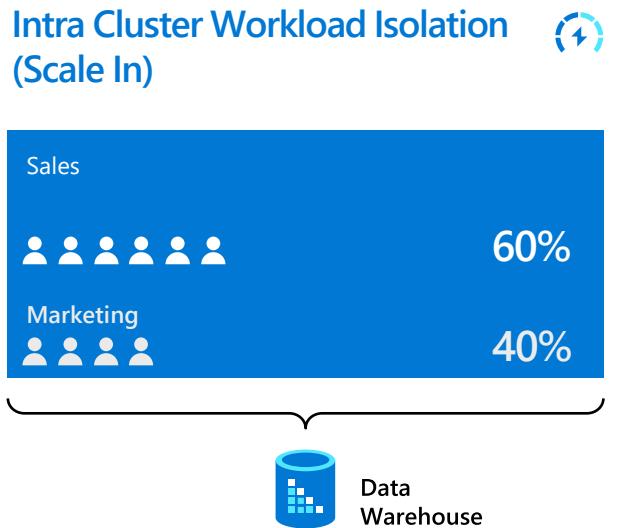
Elasticity



Multi-Cluster Isolation

WORKLOAD MANAGEMENT

Elasticity and Flexibility



Benefits:

- Most predictable cost
- Most efficient for unpredictable workloads
- No cache eviction for scaling (no performance cliff)
- Workload isolation
- Single endpoint (auto isolation with classification)

Benefits:

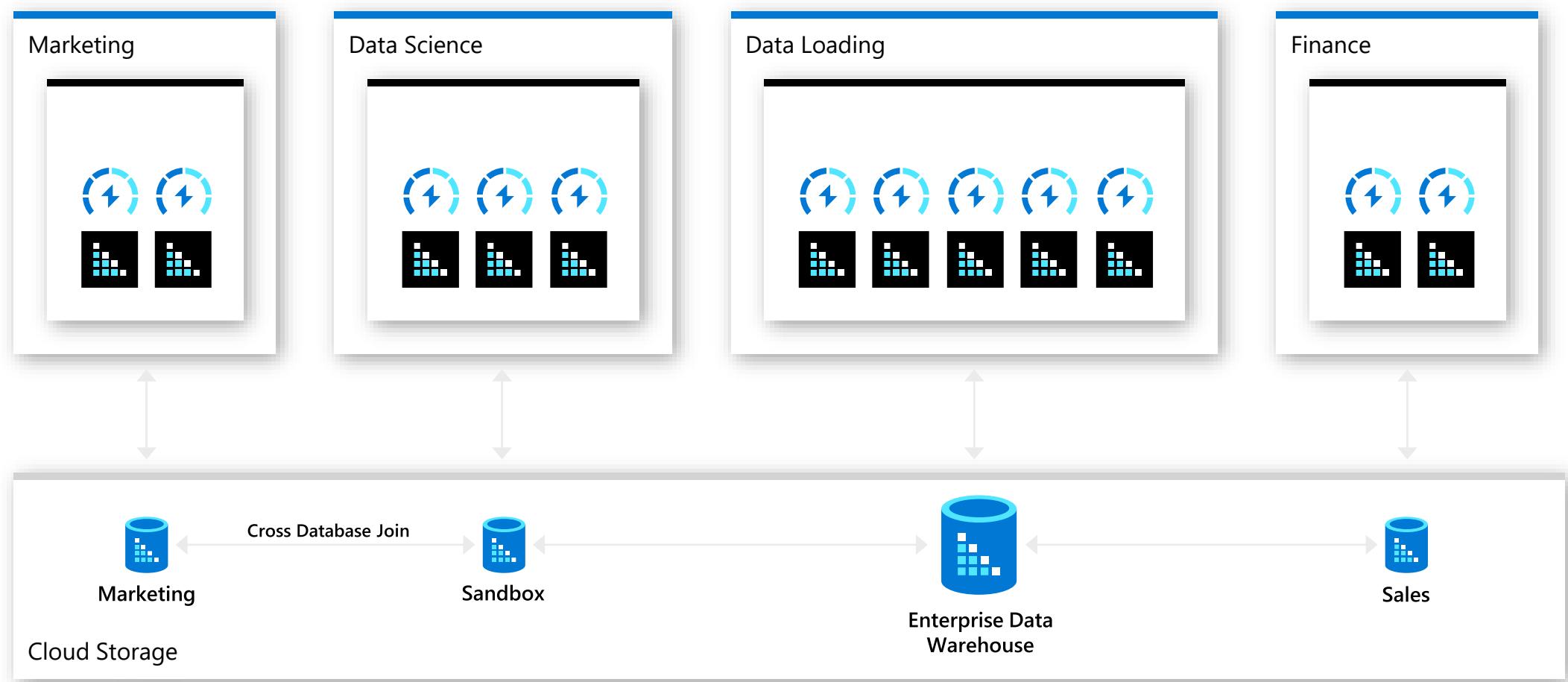
- Incremental add compute
- Increase large query performance
- Single cache for heterogeneous workloads
- Single endpoint

Benefits:

- Most performance
- Physical workload isolation
- Chargeback
- Highest concurrency

Scope: **Private Preview Coming**

MULTI-CLUSTER WORKLOAD MANAGEMENT



Interactive Query

Optimized Storage

Less Data Scanned

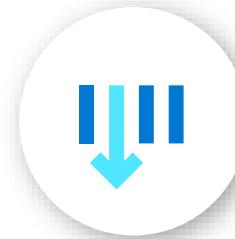
Smaller Cache Required

Smaller Clusters

Faster Queries



Columnar Storage



Columnar Ordering

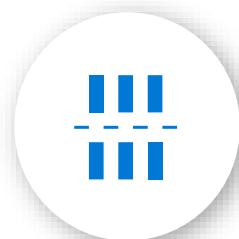
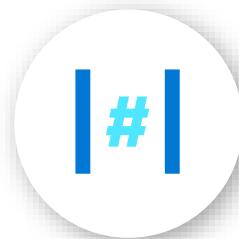


Table Partitioning



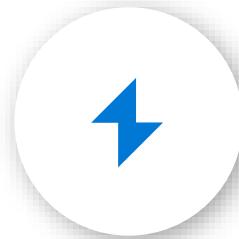
Nonclustered Indexes



Hash Distribution



Materialized Views



Resultset Cache

Interactive Query

JSON Support

Store JSON in original format

Query JSON documents with SQL language

Join to regular data types

ISJSON	Verify if text is valid JSON
JSON_VALUE	Extract a scalar value
JSON_QUERY	Extract a JSON object or array
JSON_MODIFY	Modifies a value in a JSON string
OPENJSON	Convert JSON collection to a set of rows and columns

```
-- Extract values from JSON string
SELECT CustomerId,
       Country,
       JSON_VALUE(OrderDetails, '$.StoreId') AS StoreId,
       JSON_QUERY(OrderDetails, '$.Item') AS ItemDetails
FROM   CustomerOrders;
```

CustomerId	Country	StoreId	ItemDetails
101	Bahrain	AW73565	{ "Price":2024.40, "Quantity":1 }

```

CREATE TABLE FactTransactions(
    [OrderKey],
    [OrderDateKey],
    [ProductKey],
    [CustomerKey],
    [DiscountAmt],
    [SalesAmt]
    WITH (DISTRIBUTION = HASH(ProductKey, CustomerKey))

```

Interactive Query

Multi-Column Hash Distribution

Compatible with Teradata data models

Simplifies and accelerates migration

No schema redesign required

Hash Distribution

OrderKey	OrderDateKey	ProductKey	CustomerKey	DiscountAmt	SalesAmt
7483	20190801	123	78543	12.26	423.43
7523	20190802	123	58483	13.75	812.45
7499	20190901	345	23456	32.64	123.43

OrderKey	OrderDateKey	ProductKey	CustomerKey	DiscountAmt	SalesAmt
7501	20190801	123	62304	12.26	423.43
7523	20190802	123	62304	13.75	812.45
7499	20190901	893	11256	32.64	123.43

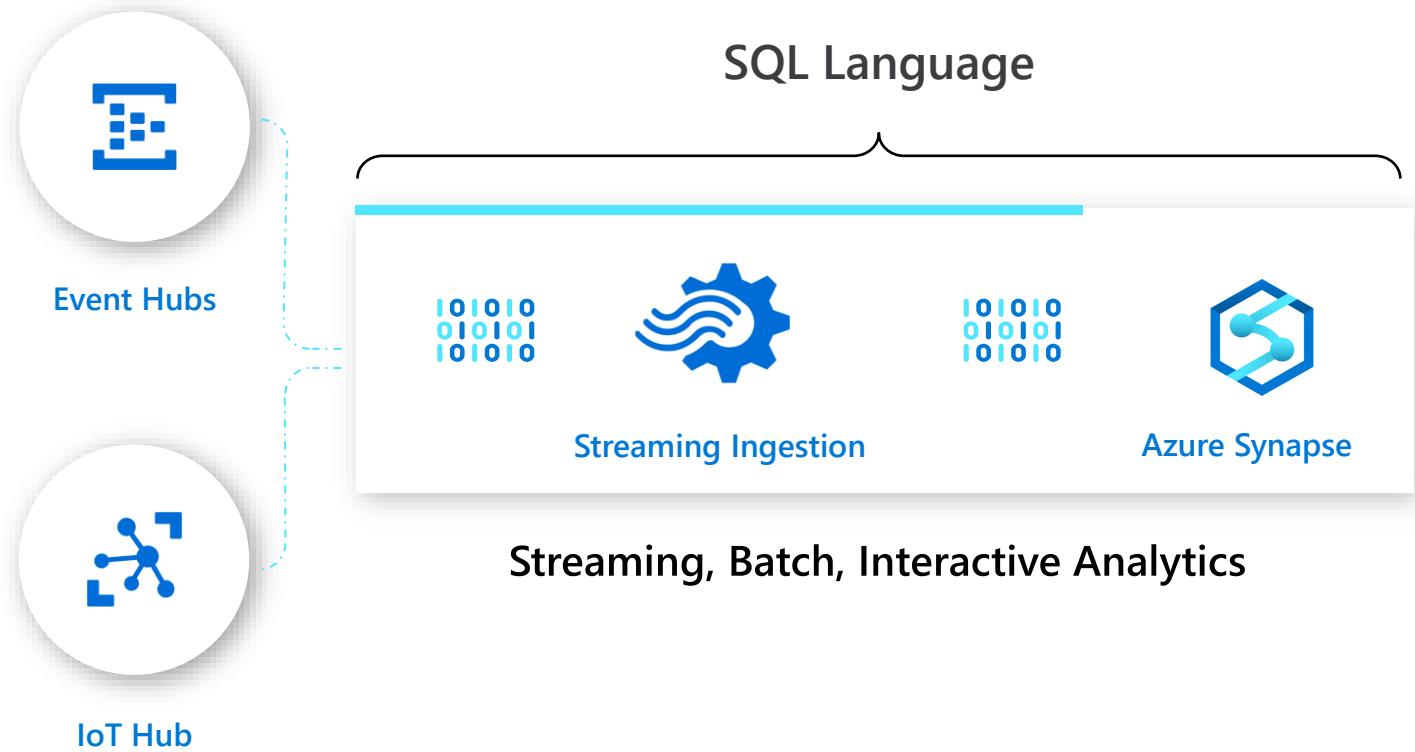
Data Preparation and Ingestion

Native SQL Streaming

High Throughput to Data Warehouse (200MB/sec)

Guaranteed Repeatability

Guaranteed Once-Only Delivery



Data Preparation and Ingestion

Auto-schema Discovery

Detect schema on load

Automatically create target table

Id,FirstName,LastName,Country,BirthDate,Distance

7483,William,Horn,England,27/5/1950,124.20
324,Fiona,Smith,Australia,15/10/1972,30.13
32981,Jennifer,Taylor,USA,2/4/1990,60.0

File

Detect Schema (CSV, Datatypes)

Column Name	Data Type
Id	int
FirstName	varchar(50)
LastName	varchar(50)
Country	varchar(50)
BirthDate	date
Distance	decimal(5,2)

Create Table

Create Table / Load Data

Id	FirstName	LastName	Country	BirthDate	Distance
7483	William	Horn	England	1950-05-27	124.20
324	Fiona	Smith	Australia	1972-10-15	30.13
32981	Jennifer	Taylor	USA	1990-04-02	60.0

Table

Data Preparation and Ingestion

COPY Command

Single SQL statement for loading data

No External Table required

~2x faster than PolyBase loading

Support Azure Blob Storage and Data Lake Storage

```
--Create destination table in SQL DW
CREATE TABLE [dbo].[weatherTable]
(
    [ObservationTypeCode] [nvarchar](5) NOT NULL,
    [ObservationTypeName] [nvarchar](100) NOT NULL,
    [ObservationUnits] [nvarchar](5) NULL
)
WITH (DISTRIBUTION = ROUND_ROBIN, HEAP);

--Copy files in parallel directly into data warehouse table
COPY INTO [dbo].[weatherTable]
FROM 'abfss://<storageaccount>.blob.core.windows.net/<filepath>'
WITH (
    FILE_FORMAT = 'DELIMITEDTEXT',
    SECRET = CredentialObject);
```

SQL On-Demand
SQL On Demand

SQL On-Demand

Overview

An interactive query service that provides T-SQL queries over high scale data in Azure Storage.

Benefits

Serverless

No infrastructure

Pay only for query execution

No ETL

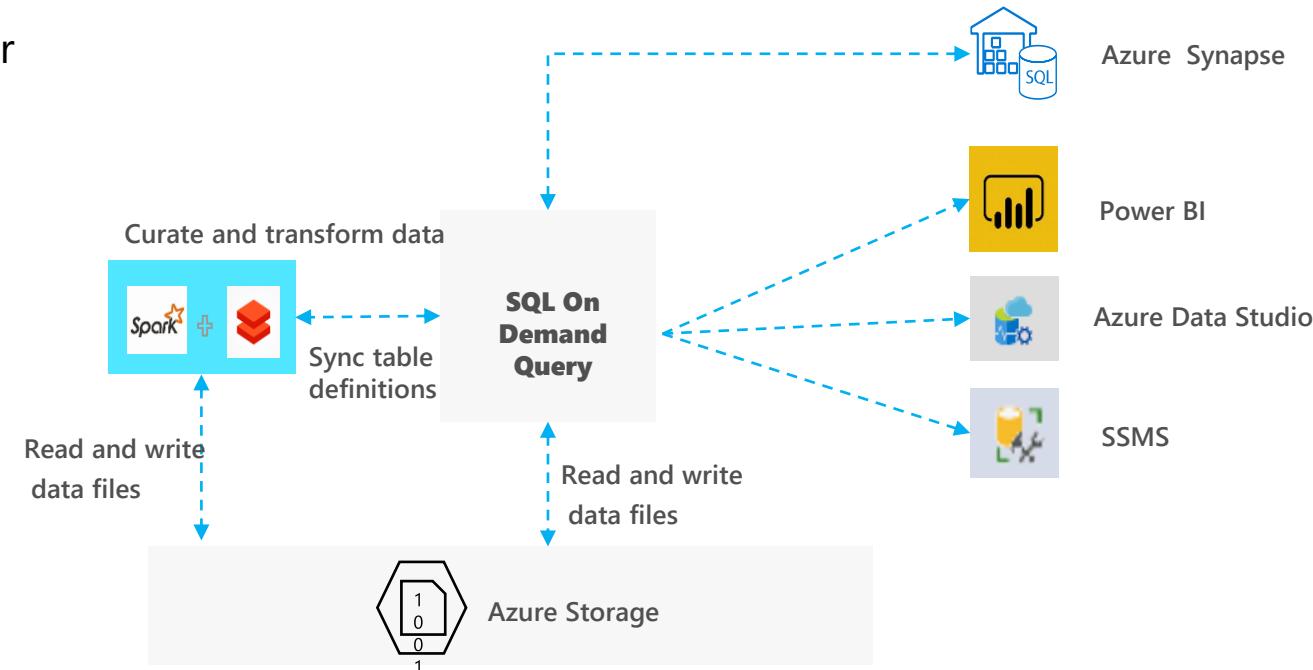
Offers security

Data integration with Databricks, HDInsight

T-SQL syntax to query data

Supports data in various formats (Parquet, CSV, JSON)

Support for BI ecosystem



SQL On Demand – Querying on storage

The screenshot illustrates the workflow for querying data stored in Azure Data Lake Storage using SQL On Demand:

- Left Panel (Storage Accounts):** Shows the storage account structure under "prlangaddemo (Primary)". The "nytlc" container is selected.
- Middle Panel (File Explorer):** Displays the contents of the "nytlc" container, specifically a file named "part-00133-19836543-aea5b543-5e83-4a7d-8d31-69f72c50b05d-15253-1.c000.snappy.parquet". A context menu is open over this file, with the "New SQL script" option highlighted.
- Right Panel (Query Editor):** The query editor shows the SQL code generated by the "New SQL script" option. The "Connect to" dropdown is set to "SQL Analytics on-demand", which is highlighted with a red box.
- Results View:** The results pane displays the query results, showing trip data from New York City (NYC) with columns like VENDORID, TPEPICKUPDATETIME, TRIPDISTANCE, and STARTLAT.

```

1 SELECT
2   TOP 100 *
3   FROM
4     OPENROWSET(
5       BULK 'https://prlangaddemo.dfs.core.windows.net/nytlc/yellow/puYear=2015/puMonth=3/part-00133-tid-210938564719836543-aea5b543-5e83-4a7d-8d31-69f72c50b05d-15253-1.c000.snappy.parquet'
6       FORMAT='PARQUET'
7     ) AS nyc;
8
  
```

00:01:00 Query executed successfully.

SQL On Demand – Querying CSV File

Overview

Uses OPENROWSET function to access data

Benefits

Ability to read CSV File with

- no header row, Windows style new line
- no header row, Unix-style new line
- header row, Unix-style new line
- header row, Unix-style new line, quoted
- header row, Unix-style new line, escape
- header row, Unix-style new line, tab-delimited
- without specifying all columns

```
SELECT *
FROM OPENROWSET(
    BULK 'https://XXX.blob.core.windows.net/csv/population/population.csv',
    FORMAT = 'CSV',
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n'
)
WITH (
    [country_code] VARCHAR (5) COLLATE Latin1_General_BIN2,
    [country_name] VARCHAR (100) COLLATE Latin1_General_BIN2,
    [year] smallint,
    [population] bigint
) AS [r]
WHERE
    country_name = 'Luxembourg'
    AND year = 2017
```

	country_code	country_name	year	population
1	LU	Luxembourg	2017	594130

SQL On Demand – Querying folders

Overview

Uses OPENROWSET function to access data from multiple files or folders

Benefits

Offers reading multiple files/folders through usage of wildcards

Offers reading specific file/folder

Supports use of multiple wildcards

```

SELECT YEAR(pickup_datetime) AS [year], SUM(passenger_count) AS passengers_total,
COUNT(*) AS [rides_total]
FROM OPENROWSET(
BULK 'https://XXX.blob.core.windows.net/csv/taxi/*.csv',
FORMAT = 'CSV'
, FIRSTROW = 2 )
WITH (
    vendor_id VARCHAR(100) COLLATE Latin1_General_BIN2,
    pickup_datetime DATETIME2,
    dropoff_datetime DATETIME2,
    passenger_count INT,
    trip_distance FLOAT,
    rate_code INT,
    store_and_fwd_flag VARCHAR(100) COLLATE Latin1_General_BIN2,
    pickup_location_id INT,
    dropoff_location_id INT,
    payment_type INT,
    fare_amount FLOAT,
    extra FLOAT, mta_tax FLOAT,
    tip_amount FLOAT,
    tolls_amount FLOAT,
    improvement_surcharge FLOAT,
    total_amount FLOAT
) AS nyc
GROUP BY YEAR(pickup_datetime)
ORDER BY YEAR(pickup_datetime)

```

	year	passengers_total	rides_total
1	2001	14	10
2	2002	29	16
3	2003	22	16
4	2008	378	188
5	2009	594	353
6	2016	102093687	61758523
7	2017	184464988	113496932
8	2018	86272771	53925040
9	2019	37	29
...	2020	6	6

SQL On Demand – Querying specific files

Overview

filename – Provides file name that originates row result

filepath – Provides full path when no parameter is passed or part of path when parameter is passed that originates result

Benefits

Provides source name/path of file/folder for row result set

Example of filename function

```

SELECT
    r.filename() AS [filename]
    ,COUNT_BIG(*) AS [rows]
FROM OPENROWSET(
    BULK 'https://XXX.blob.core.windows.net/csv/taxi/yellow_tripdata_201
7-1*.csv',
    FORMAT = 'CSV',
    FIRSTROW = 2
)
WITH (
    vendor_id INT,
    pickup_datetime DATETIME2,
    dropoff_datetime DATETIME2,
    passenger_count SMALLINT,
    trip_distance FLOAT,
    <...columns>
) AS [r]
GROUP BY r.filename()
ORDER BY [filename]

```

	filename	rows
1	yellow_tripdata_2017-10.csv	9768815
2	yellow_tripdata_2017-11.csv	9284803
3	yellow_tripdata_2017-12.csv	9508276

SQL On Demand – Querying Parquet files

Overview

Uses OPENROWSET function to access data

Benefits

Ability to specify column names of interest

Offers auto reading of column names and data types

Provides target specific partitions using filepath function

```

SELECT
    YEAR(pickup_datetime),
    passenger_count,
    COUNT(*) AS cnt
FROM
    OPENROWSET(
        BULK 'https://XXX.blob.core.windows.net/parquet/taxi/\*/\*/\*',
        FORMAT='PARQUET'
    ) WITH (
        pickup_datetime DATETIME2,
        passenger_count INT
    ) AS nyc
GROUP BY
    passenger_count,
    YEAR(pickup_datetime)
ORDER BY
    YEAR(pickup_datetime),
    passenger_count
  
```

	(No column name)	passenger_count	cnt
1	2016	0	2557
2	2016	1	43735845
3	2016	2	9056714
4	2016	3	2610541
5	2016	4	1309639
6	2016	5	3086097
7	2016	6	1956607

SQL On Demand – Creating views

The screenshot displays three Microsoft Azure Synapse Analytics windows:

- Top Window:** Shows a query script for creating a view named `yellow_2017`. The script includes a `BULK` command to load data from a Parquet file in Azure Data Lake Storage.
- Middle Window:** Shows the results of the query execution. A message indicates "00:00:17 Query executed successfully."
- Bottom Window:** Shows a chart of passenger count versus year. The X-axis represents the year (2017) and the passenger count (0 to 10). The Y-axis represents the count (0 to 100M). The chart shows a single data series named `cnt`, which peaks at year 1 with a value of approximately 80M.

The screenshot shows a Microsoft Azure Synapse Analytics window with the following details:

- Top Bar:** Shows the connection to `SQL Analytics on-demand` and the database `nycyellow`.
- Query Editor:** Displays a `SELECT` query to calculate the count of passengers per year from the `yellow_2017` table.
- Results:** Shows a table with the following data:

(NO COLUMN NAME)	PASSENGERCOUNT	CNT
2017	0	166086
2017	1	81034075
2017	2	16545571
2017	3	4748869
2017	4	2257813
2017	5	5407319

- Message:** "00:02:19 Query executed successfully."

SQL On Demand – Querying JSON files

Overview

Read JSON files and provides data in tabular format

Benefits

Supports OPENJSON, JSON_VALUE and JSON_QUERY functions

```
SELECT *
FROM
    OPENROWSET(
        BULK 'https://XXX.blob.core.windows.net/json/books/book
1.json',
        FORMAT='CSV',
        FIELDTERMINATOR = '0x0b',
        FIELDQUOTE = '0x0b',
        ROWTERMINATOR = '0x0b'
    )
    WITH (
        jsonContent varchar(8000)
    ) AS [r]
```

	jsonContent
1	{"_id": "kim95", "type": "Book", "title": "Modern Databas...

Create External Table As Select

Overview

Creates an external table and then exports results of the Select statement. These operations will import data into the database for the duration of the query

Steps:

1. Create Master Key
2. Create Credentials
3. Create External Data Source
4. Create External Data Format
5. Create External Table

```
-- Create a database master key if one does not already exist
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'S0me!nfo'
;

-- Create a database scoped credential with Azure storage account key
-- as the secret.
CREATE DATABASE SCOPED CREDENTIAL AzureStorageCredential
WITH
    IDENTITY      = '<my_account>'
,   SECRET        = '<azure_storage_account_key>'
;

-- Create an external data source with CREDENTIAL option.
CREATE EXTERNAL DATA SOURCE MyAzureStorage
WITH
(   LOCATION      = 'wasbs://daily@logs.blob.core.windows.net/'
,   CREDENTIAL    = AzureStorageCredential
,   TYPE          = HADOOP
)
;

-- Create an external file format
CREATE EXTERNAL FILE FORMAT MyAzureCSVFormat
WITH (FORMAT_TYPE = DELIMITEDTEXT,
      FORMAT_OPTIONS(
          FIELD_TERMINATOR = ',',
          FIRST_ROW = 2))
;

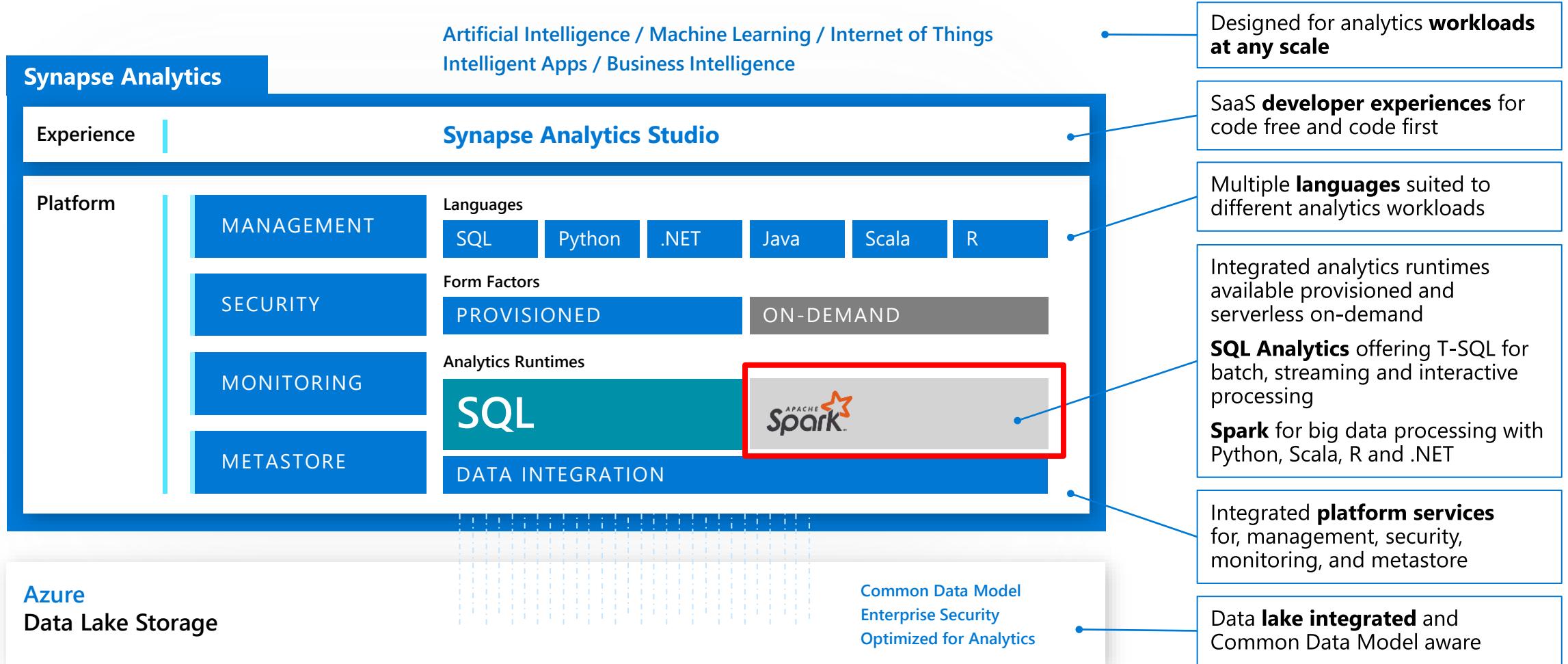
--Create an external table
CREATE EXTERNAL TABLE dbo.FactInternetSalesNew
WITH(
    LOCATION = '/files/Customer',
    DATA_SOURCE = MyAzureStorage,
    FILE_FORMAT = MyAzureCSVFormat
)
AS SELECT T1.* FROM dbo.FactInternetSales T1 JOIN dbo.DimCustomer T2
ON ( T1.CustomerKey = T2.CustomerKey )
OPTION ( HASH JOIN );
```



Azure Synapse Analytics Spark

Azure Synapse Analytics

Integrated data platform for BI, AI and continuous intelligence



Library Management - Python

Overview

Customers can add new python libraries at Spark pool level

Benefits

Input requirements.txt in simple pip freeze format

Add new libraries to your cluster

Update versions of existing libraries on your cluster

Libraries will get installed for your Spark pool during cluster creation

Ability to specify different requirements file for different pools within the same workspace

Constraints

The library version must exist on PyPI repository

Version downgrade of an existing library not allowed

In the Portal

Specify the new requirements while creating Spark Pool in Additional Settings blade

Microsoft Azure (Preview) Restore default configuration Report a bug Search resources, services, and data

Home > nushuklasynapsewestus2 > Create Apache Spark pool

Create Apache Spark pool

Enter required settings for this Apache Spark pool, including setting auto-pause and picking versions.

Auto-pause * Enabled Disabled
Number of minutes idle *

Component versions
Select the Apache Spark version for your Apache Spark pool.

Apache Spark *	<input type="text" value="2.4"/>
Python	3.6.1
Scala	2.11.12
Java	1.8.0_222
.NET Core	3.0
.NET for Apache Spark	0.6.0
Delta Lake	0.4.0

Packages
Upload environment configuration file ("PIP freeze" output).

File upload

Review + create < Previous Next: Tags >

Library Management - Python

The screenshot shows the Microsoft Azure Synapse Analytics interface, specifically the 'Develop' workspace. On the left, a sidebar lists resources under 'Develop': SQL scripts, Notebooks (Notebook 2, * Notebook 4, Notebook-SparkDotNet Exam...), Data flows, Spark job definitions, and Power BI. The 'Notebooks' section is expanded, and 'Notebook 4' is selected. The main area displays a Python notebook titled 'Notebook 4'. Cell 1 contains the following code:

```
1 import pprint
2 import pip
3 installed_packages = pip.get_installed_distributions()
4 installed_packages_list = sorted(["%s==%s" % (i.key, i.version)
5 | | for i in installed_packages])
6 pprint.pprint(installed_packages_list)
```

Below the code, a message indicates the command was executed: "Command executed in 1mins 58s 291ms by prlangad on 10-30-2019 13:08:49.447 -07:00". The output of the code is a long list of installed Python packages and their versions, starting with 'absl-py==0.8.1' and ending with 'azure-storage-common==2.1.0'. The bottom of the screen shows session status: 'Ready' (green checkmark), 'Stop session', 'Spark history server', and 'Configure session'.

Languages

Overview

Supports multiple languages to develop notebook

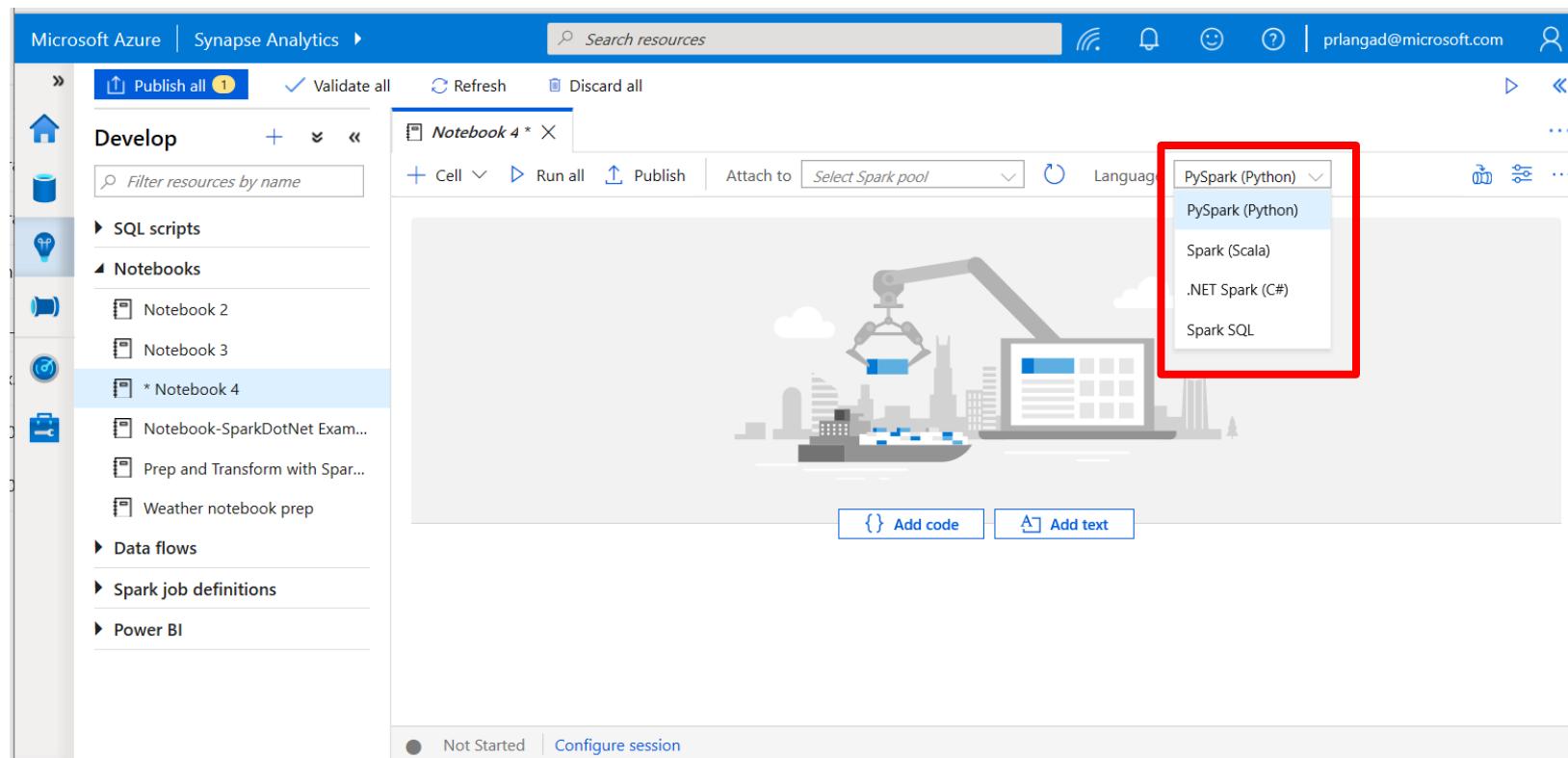
- PySpark (Python)
- Spark (Scala)
- .NET Spark (C#)
- Spark SQL

Benefits

Allows to write multiple languages in one notebook

%%<Name of language>

Offers use of temporary tables across languages



Languages – PySpark (Python)

Screenshot of the Azure Synapse Analytics Python notebook interface.

The left sidebar shows the "Develop" workspace with the following items:

- SQL scripts
- Notebooks
 - Notebook 2
 - Notebook 3
 - Notebook 4** (selected)
 - Notebook-SparkDotNet Exam...
 - Prep and Transform with Spar...
 - Weather notebook prep
- Data flows
- Spark job definitions
- Power BI

The main area displays "Notebook 4 * X" titled "Weather notebook". The toolbar includes: Publish all (1), Validate all, Refresh, Discard all, Cell (dropdown), Run all, Publish, Attach to (prlSpark), Language (PySpark (Python)), and a three-dot menu.

Cell 1 contains the following Python code:

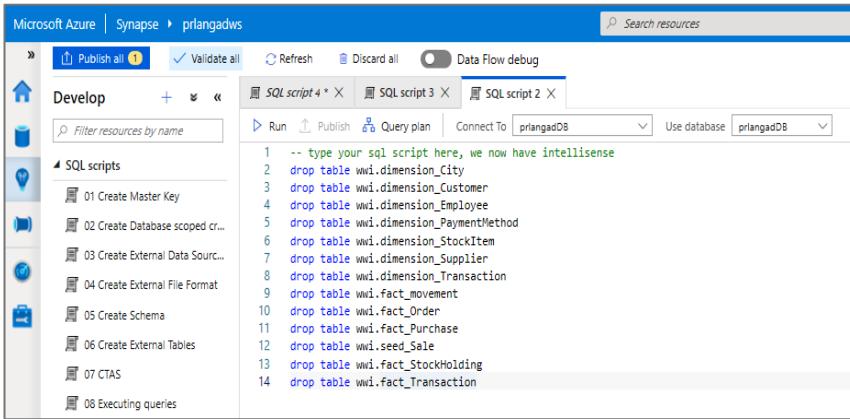
```
1 import numpy as np
2 from matplotlib import pyplot as plt
3
4 np.random.seed(3)
5 x = np.random.randn(250)
6 plt.hist(x)
7 plt.show()
```

The output of Cell 1 is a histogram showing a standard normal distribution (bell curve) centered at 0, with the x-axis ranging from -3 to 3 and the y-axis ranging from 0 to 60.

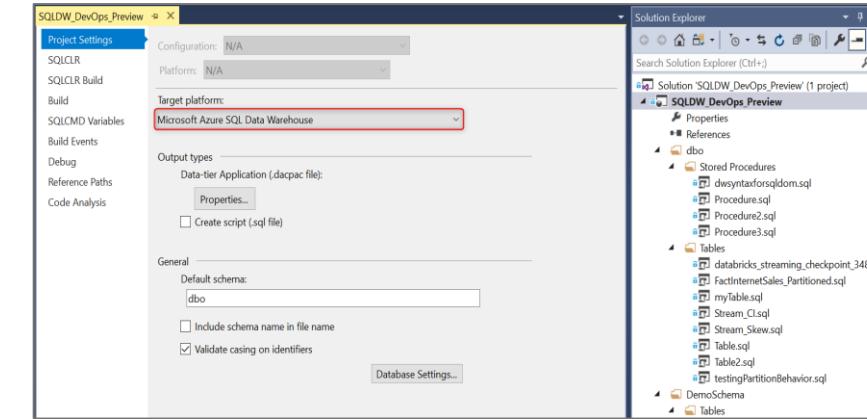
At the bottom, the status bar shows: Ready | Stop session | Spark history server | Configure session.

Developer Tools

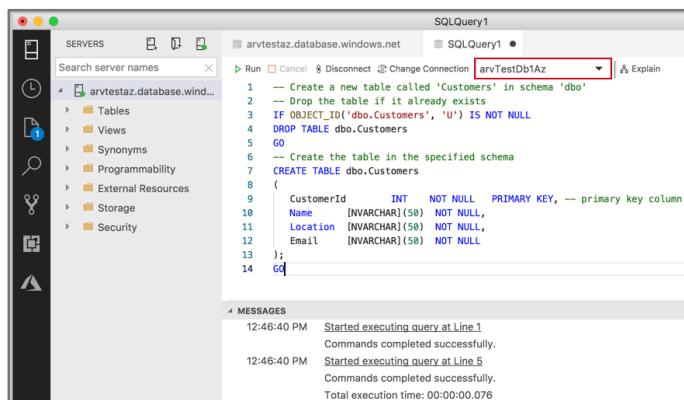
Azure Synapse Analytics



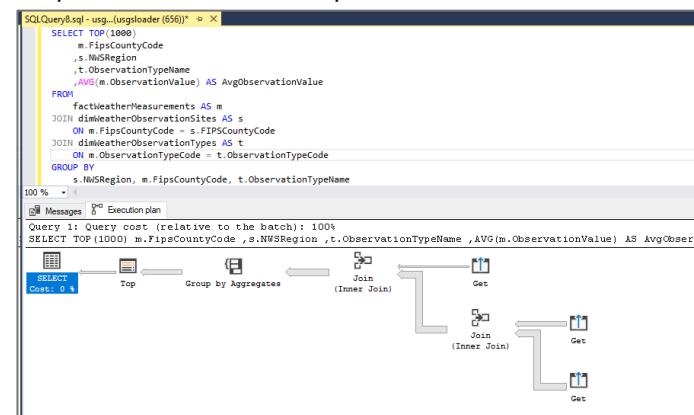
Visual Studio - SSDT database projects



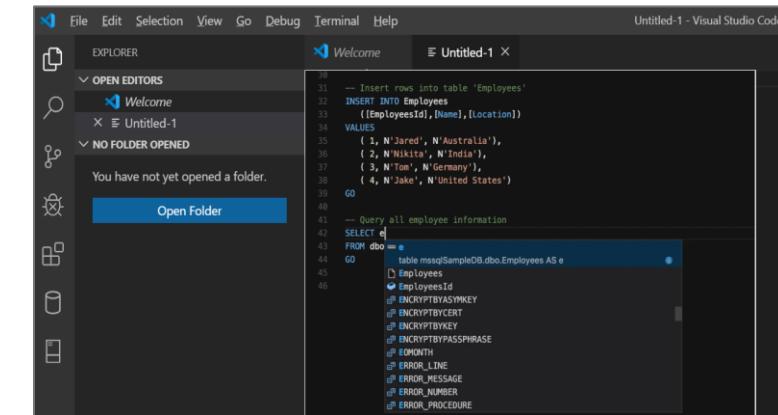
Azure Data Studio (queries, extensions etc.)



SQL Server Management Studio (queries, execution plans etc.)

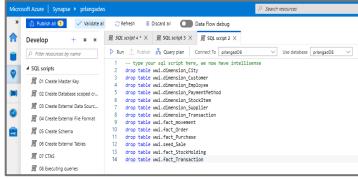


Visual Studio Code



Developer Tools

Azure Synapse Analytics

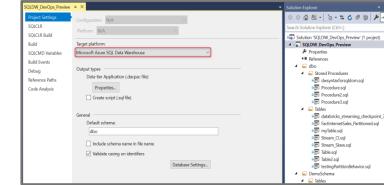


Azure Cloud Service

Offers end-to-end lifecycle for analytics

Connects to multiple services

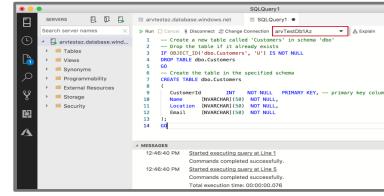
Visual Studio - SSDT database projects



Runs on Windows

Create, maintain database code, compile, code refactoring

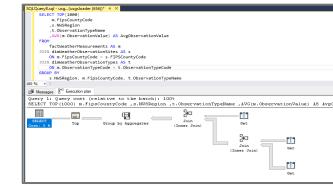
Azure Data Studio



Runs on Windows, Linux, macOS

Light weight editor, (queries and extensions)

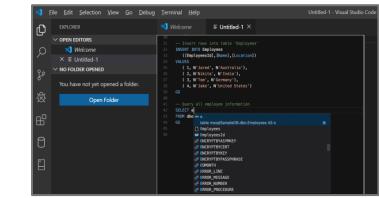
SQL Server Management Studio



Runs on Windows

Offers GUI support to query, design and manage

Visual Studio Code



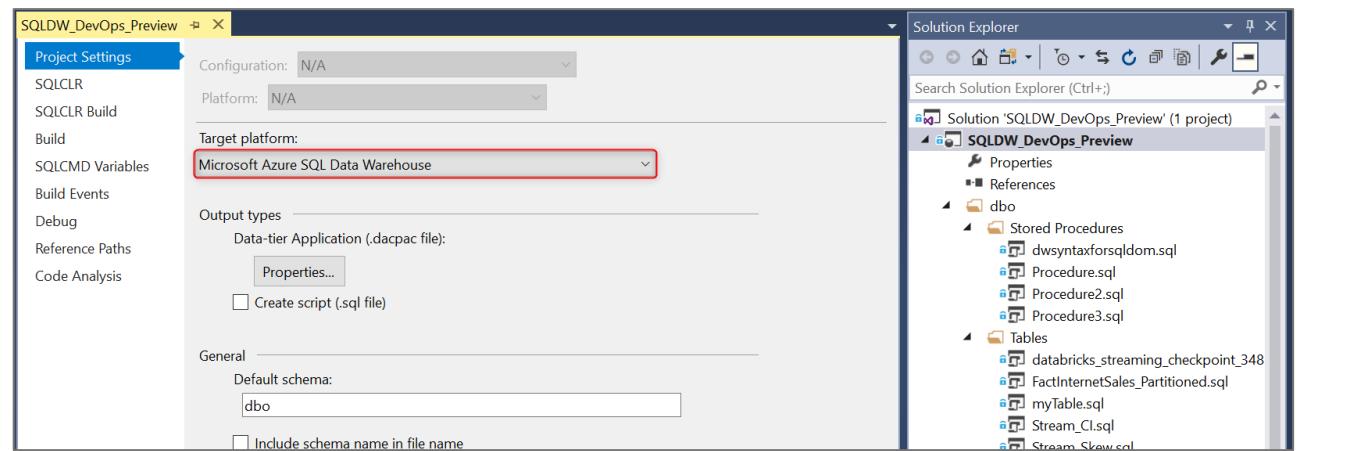
Runs on Windows, Linux, macOS

Offers development experience with light-weight code editor

Continuous integration and delivery (CI/CD)

Overview

Database project support in SQL Server Data Tools (SSDT) allows teams of developers to collaborate over a version-controlled data warehouse, and track, deploy and test schema changes.



The screenshot displays three main windows: 1) The 'Tables' page in Azure DevOps showing a list of database objects like 'bcprate14.sql', 'compressedRate.sql', and 'FactOnlineSales_1500MB.sql'. 2) The 'Builds' page in OneITVSO showing a build named 'Build 20161130.4' with a status of 'Build Started'. 3) The 'Console' window at the bottom showing the output of a build command, including NuGet package restoration and MSBuild logs.

Benefits

Database project support includes first-class integration with Azure DevOps. This adds support for:

- **Azure Pipelines** to run CI/CD workflows for any platform (Linux, macOS, and Windows)
- **Azure Repos** to store project files in source control
- **Azure Test Plans** to run automated check-



Azure Synapse Analytics Security

Comprehensive Security

Category	Feature	
Data Protection	Data in Transit	✓
	Data Encryption at Rest	✓
	Data Discovery and Classification	✓
Access Control	Object Level Security (Tables/Views)	✓
	Row Level Security	✓
	Column Level Security	✓
Authentication	Dynamic Data Masking	✓
	SQL Login	✓
	Azure Active Directory	✓
Network Security	Multi-Factor Authentication	✓
	Virtual Networks	✓
	Firewall	✓
Threat Protection	Azure ExpressRoute	✓
	Thread Detection	✓
	Auditing	✓
	Vulnerability Assessment	✓



Industry-leading compliance



ISO 27001



SOC 1 Type 2



SOC 2 Type 2



PCI DSS Level 1



Cloud Controls Matrix



ISO 27018



Content Delivery and Security Association



Shared Assessments



FedRAMP JAB P-ATO



HIPAA / HITECH



FIPS 140-2



21 CFR Part 11



FERPA



DISA Level 2



CJIS



IRS 1075 / ITAR-ready



European Union Model Clauses



EU Safe Harbor



United Kingdom G-Cloud



China Multi Layer Protection Scheme



China GB 18030



China CCCPPF



Singapore MTCS Level 3



Australian Signals Directorate



New Zealand GCIO



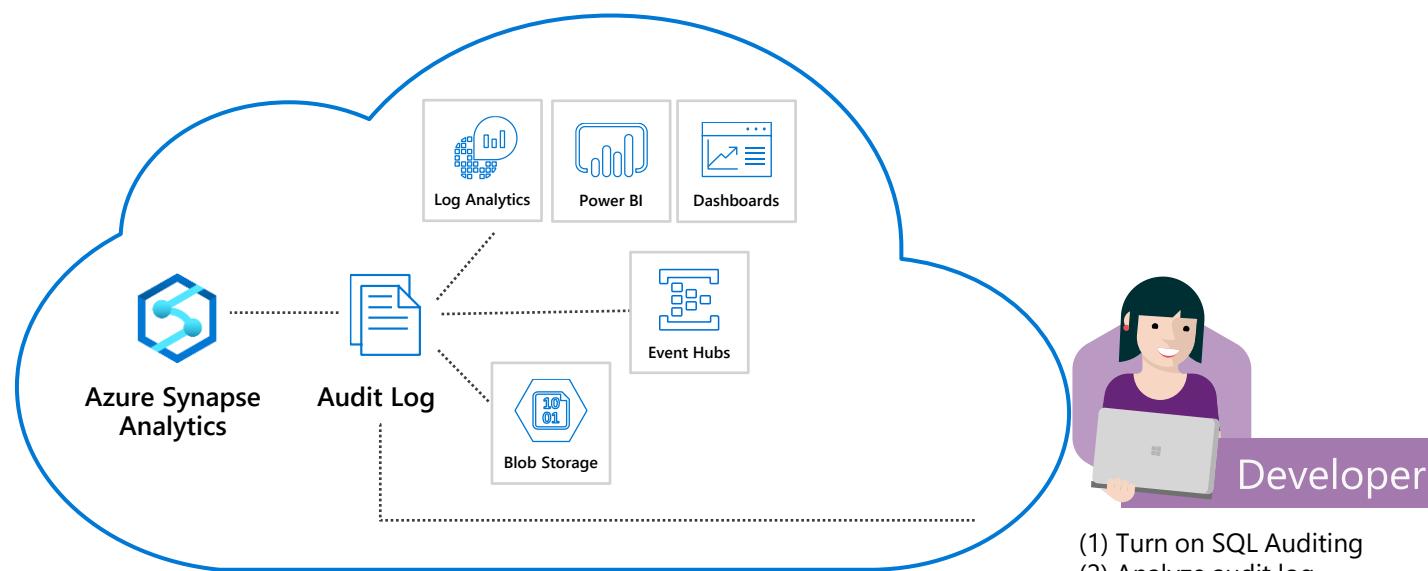
Japan Financial Services



ENISA IAF

SQL auditing in Azure Log Analytics and Event Hubs

Gain insight into database audit log

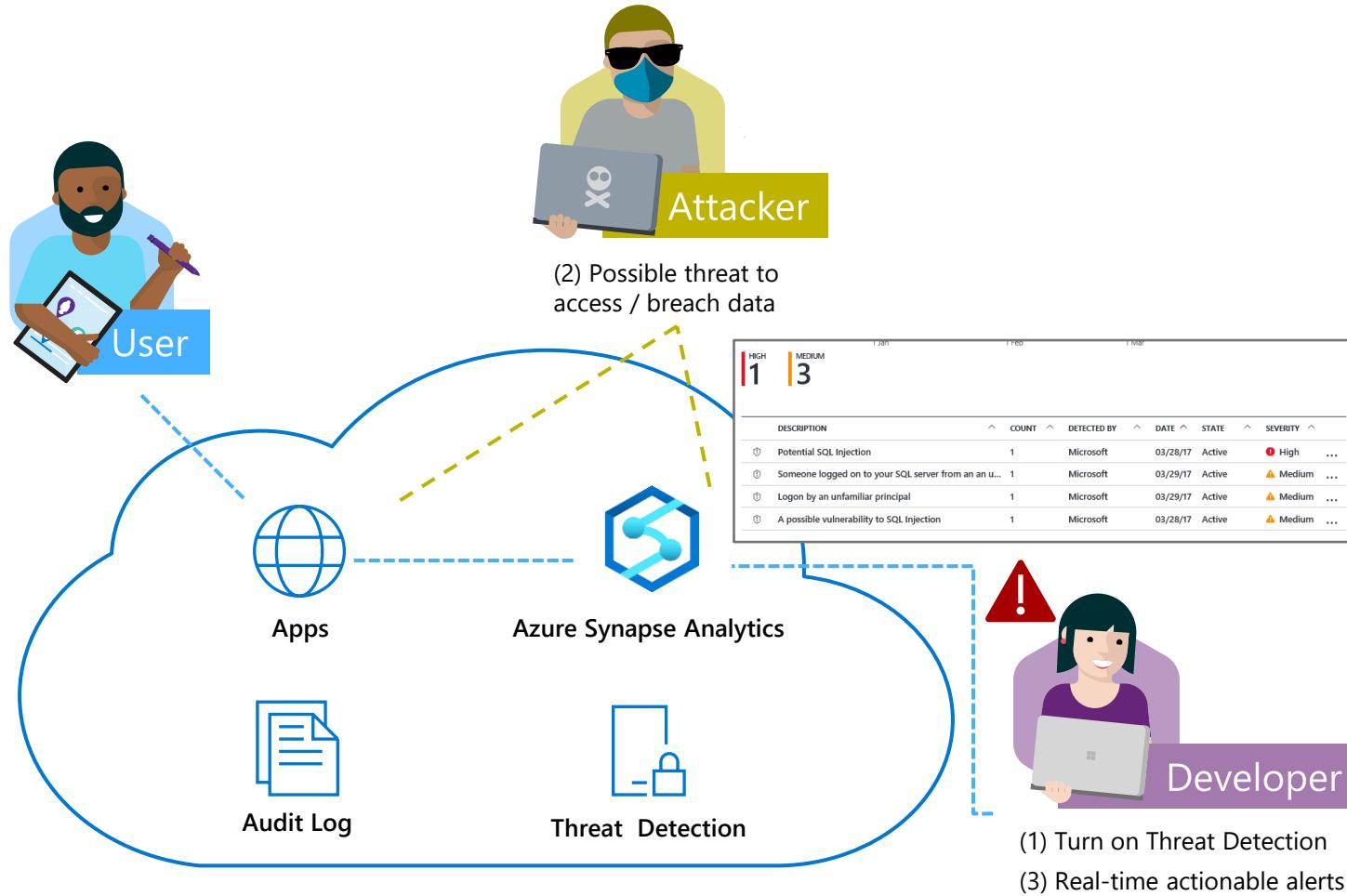


TimeGenerated	server_principal_name_s	statement_s	affected_rows_d	SeverityLevel
8/15/2018 12:00:22.521 AM	admin1	exec sp_executesql N'SELECT t1.name AS [Name], SCHEMA_NAME(t1...	0	
8/15/2018 12:00:22.521 AM	admin1	exec sp_executesql N'SELECT ISNULL(HAS_PERMS_BY_NAME('QUOTEEN...)	0	
8/15/2018 12:00:22.521 AM	admin1	DECLARE @edition sysname; SET @edition = cast(SERVERPROPERTY(N'...	4	
8/15/2018 12:00:22.521 AM	admin1	exec sp_executesql N'SELECT CAST(t1.is_enabled AS bit) AS [isEnabled]...	0	
8/15/2018 12:00:22.521 AM	admin1	IF OBJECT_ID ('[sys].[database_query_store_options]') IS NOT NULL BE...	2	

- ✓ Configurable via audit policy
- ✓ SQL audit logs can reside in
 - Azure Storage account
 - Azure Log Analytics
 - Azure Event Hubs
- ✓ Rich set of tools for
 - Investigating security alerts
 - Tracking access to sensitive data

SQL threat detection

Detect and investigate anomalous database activity



- ✓ Detects potential SQL injection attacks
- ✓ Detects unusual access & data exfiltration activities
- ✓ Actionable alerts to investigate & remediate
- ✓ View alerts for your entire Azure tenant using Azure Security Center

SQL Data Discovery & Classification

Discover, classify, protect and track access to sensitive data

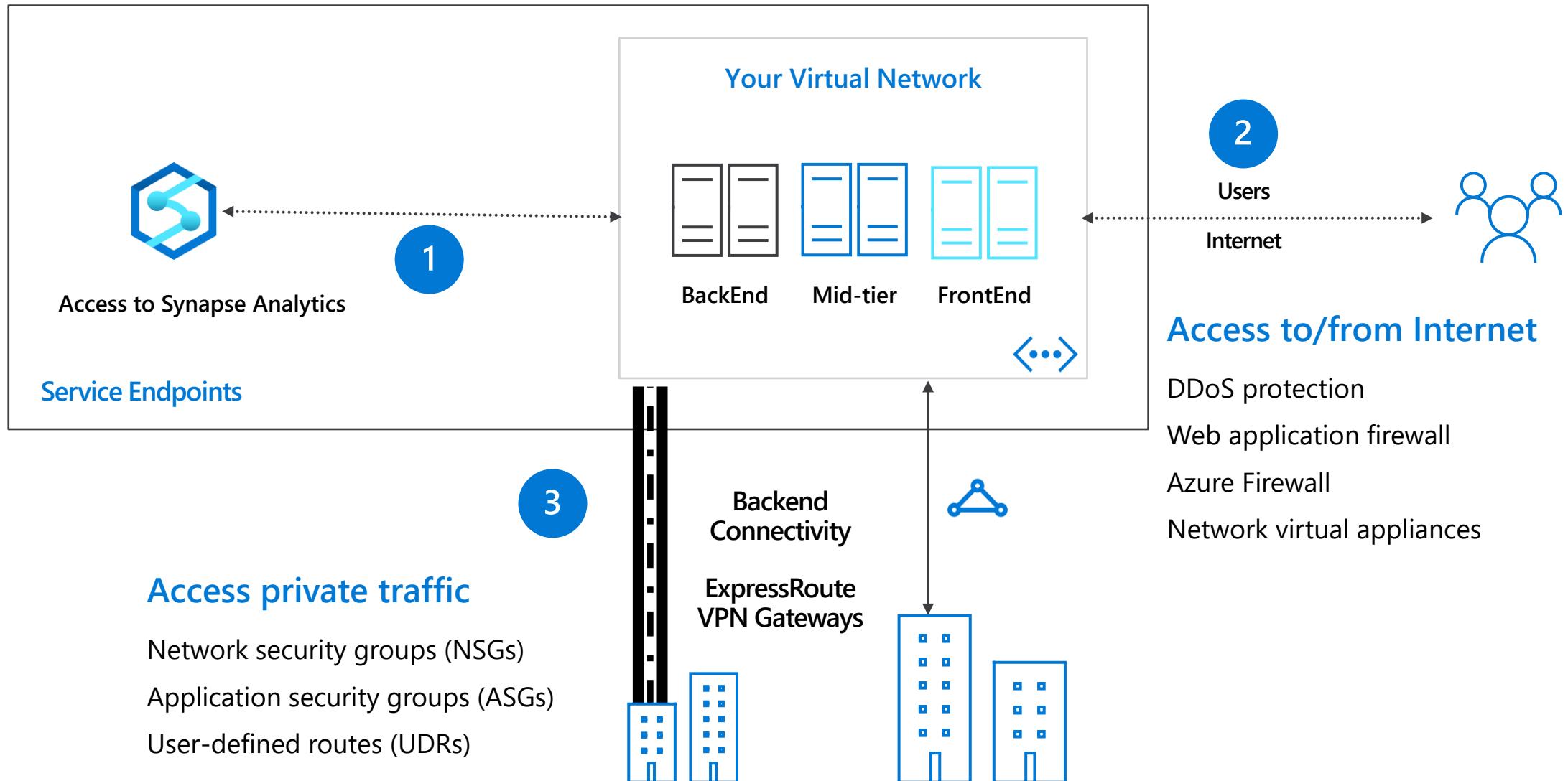
The screenshot shows the Azure portal interface for SQL Data Discovery & Classification. The main dashboard displays various metrics and donut charts. Below the dashboard, a table provides a detailed breakdown of columns by schema, table, column, information type, and sensitivity label. A separate window titled 'Settings - Information protection' shows a list of sensitivity labels with their descriptions and ordering options.

Schema	Table	Column	Information Type	Sensitivity Label
dbo	ErrorLog		UserName	Credentials
				Confidential

DISPLAY NAME	DESCRIPTION
Public	Business data that is specifically prepared and approved for public consumption
General	Business data that is not intended for public consumption. However, this can be shared with...
Confidential	Sensitive business data that could cause damage to the business if shared with unauthoriz...
Confidential - GDPR	Sensitive data containing personal information associated with an individual, that could b...
Highly confidential	Very sensitive business data that would cause damage to the business if it was shared wit...
Highly confidential - GDPR	Sensitive data containing personal information associated with an individual, that can cau...

- ✓ Automatic **discovery** of columns with sensitive data
- ✓ Add **persistent sensitive data labels**
- ✓ Audit and detect access to the sensitive data
- ✓ Manage labels for your entire Azure tenant using Azure Security Center

Azure networking: application-access patterns



Virtual network service endpoints

Overview

Extend VNET identity to the service.

Secure critical Azure resources to only your VNET.

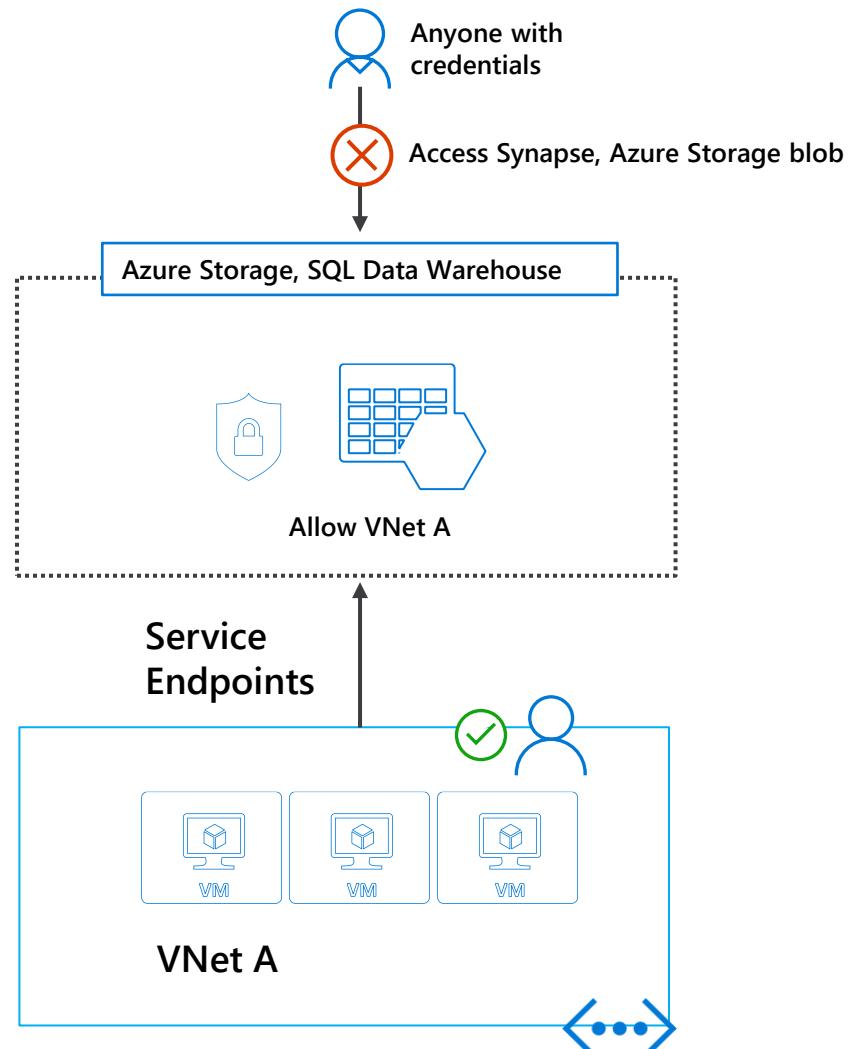
Traffic remains on the Azure backbone.

Virtual network Rules

Firewall security feature that allows communications from only specified subnets in virtual networks.

Finer granular security control than "Allow access to Azure Services."

Internet – Home location



Azure Active Directory authentication

Overview

Manage user identities in one location.

Enable access to Azure Synapse Analytics and other Microsoft services with Azure Active Directory user identities and groups.

Azure Synapse Analytics

Benefits

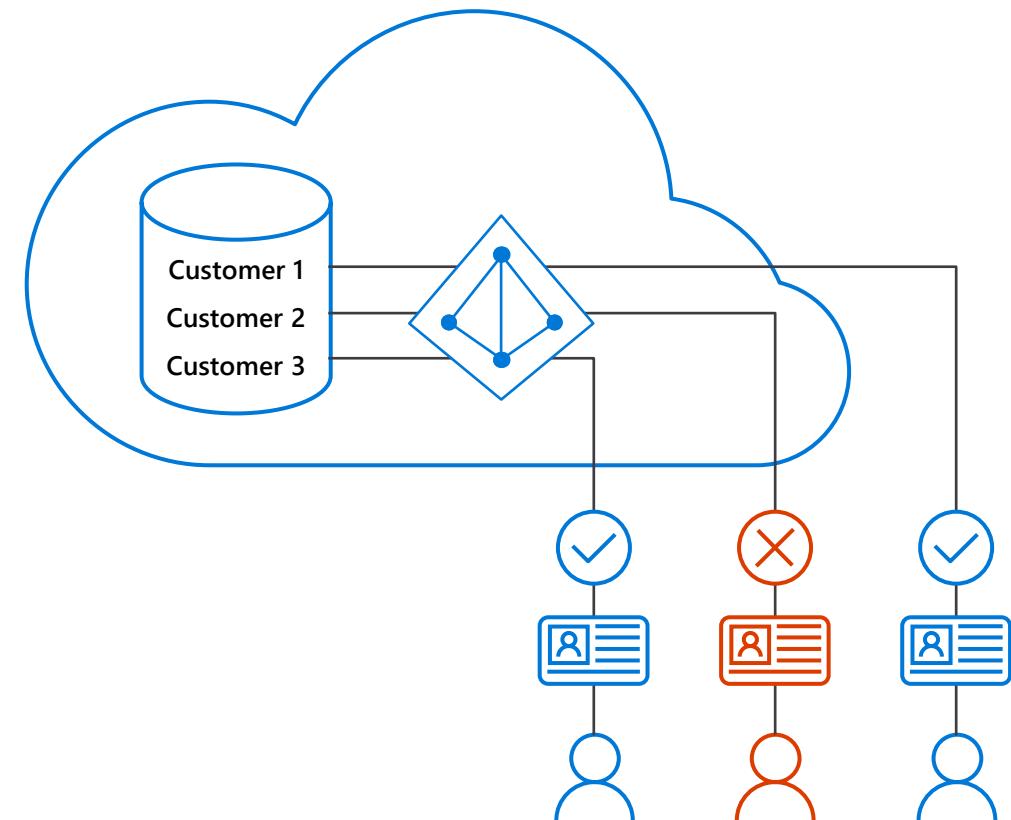
Alternative to SQL Server authentication

Limits proliferation of user identities across databases

Allows password rotation in a single place

Enables management of database permissions by using external Azure Active Directory groups

Eliminates the need to store passwords



Row-level security (RLS)

Overview

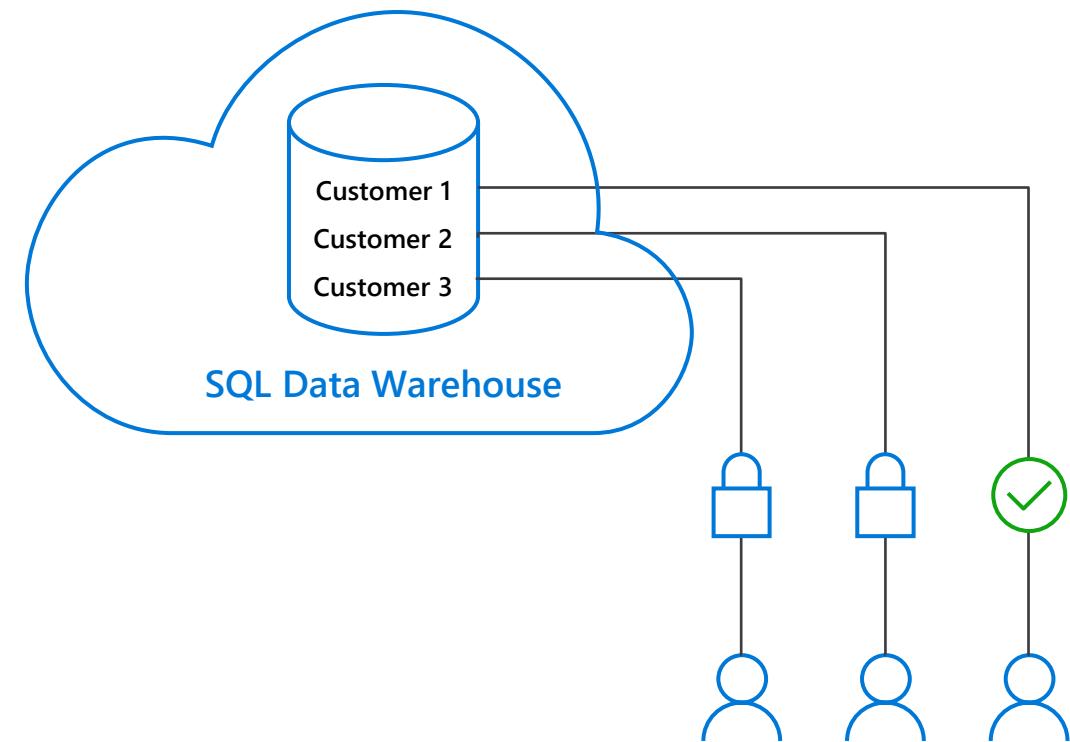
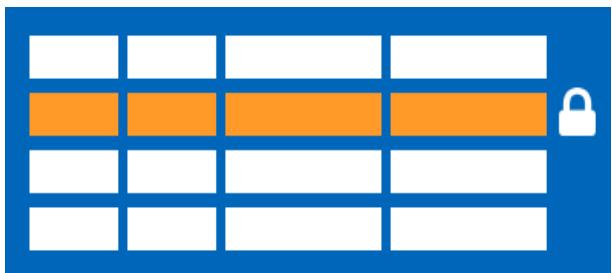
Fine grained access control of specific rows in a database table.

Help prevent unauthorized access when multiple users share the same tables.

Eliminates need to implement connection filtering in multi-tenant applications.

Administer via SQL Server Management Studio or SQL Server Data Tools.

Easily locate enforcement logic inside the database and schema bound to the table.



Column-level security

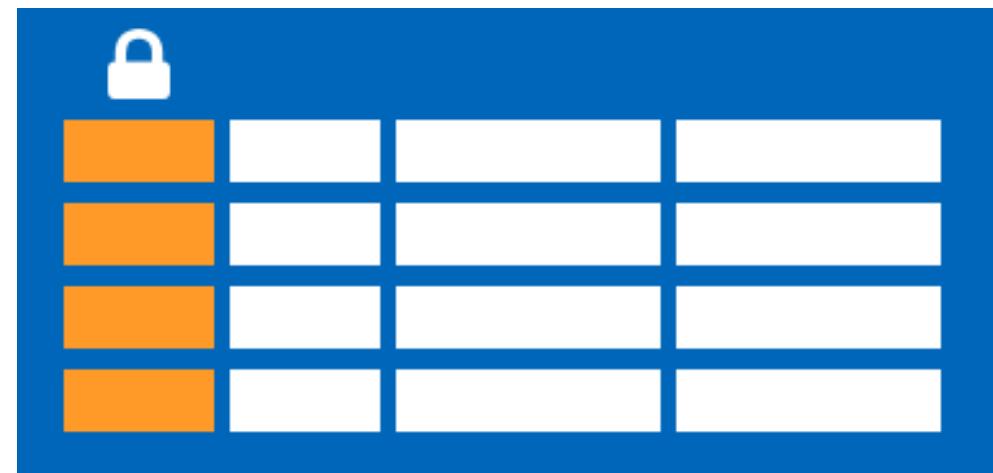
Overview

Control access of specific columns in a database table based on customer's group membership or execution context.

Simplifies the design and implementation of security by putting restriction logic in database tier as opposed to application tier.

Administer via GRANT T-SQL statement.

Both Azure Active Directory (AAD) and SQL authentication are supported.



Dynamic Data Masking

Overview

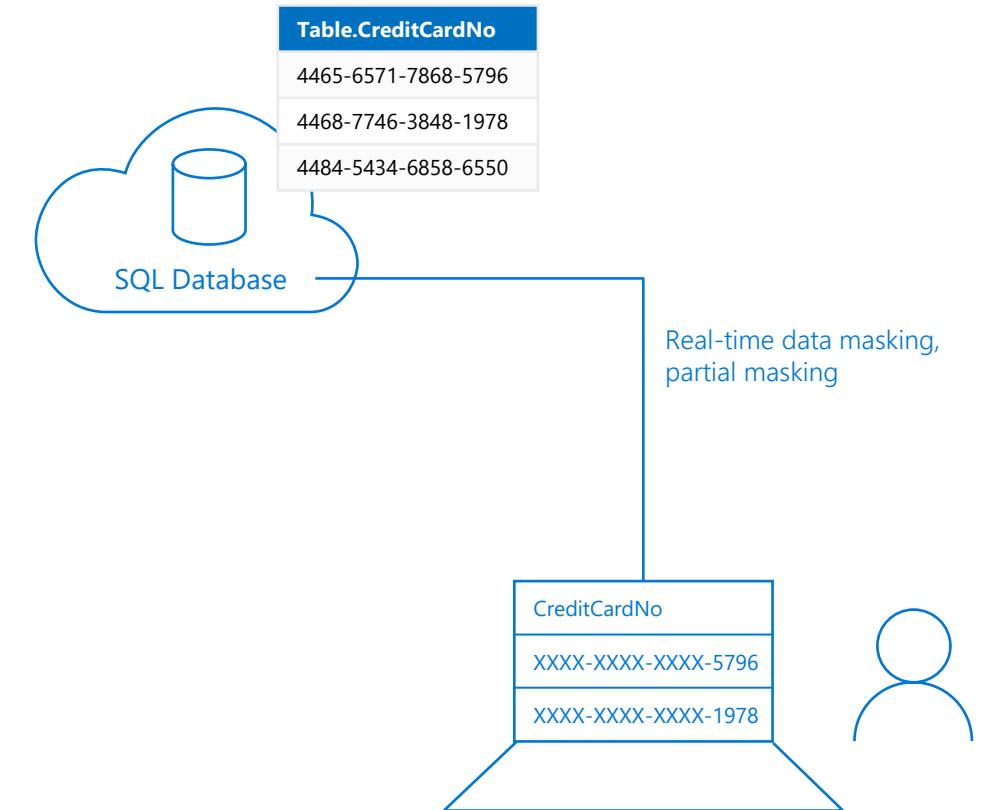
Prevent abuse of sensitive data by hiding it from users

Easy configuration in new Azure Portal

Policy-driven at table and column level, for a defined set of users

Data masking applied in real-time to query results based on policy

Multiple masking functions available, such as full or partial, for various sensitive data categories (credit card numbers, SSN, etc.)



Transparent data encryption (TDE)

Overview

All customer data encrypted at rest

TDE performs real-time I/O encryption and decryption of the data and log files.

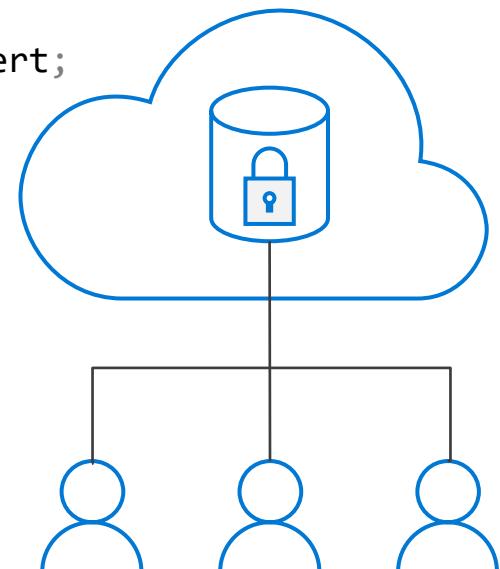
Service OR User managed keys.

Application changes kept to a minimum.

Transparent encryption/decryption of data in a TDE-enabled client driver.

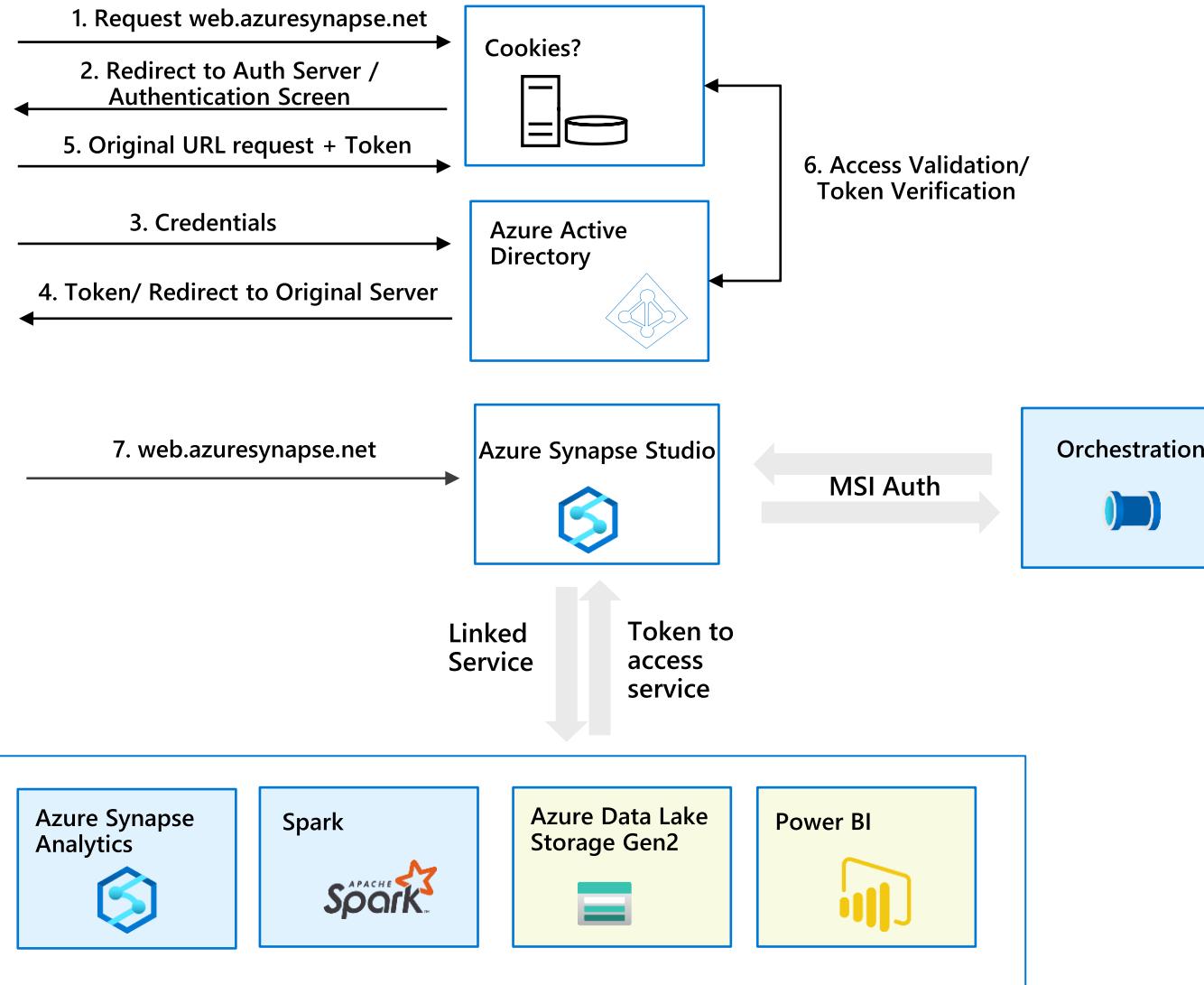
Compliant with many laws, regulations, and guidelines established across various industries.

```
USE master;
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = '<UseStrongPasswordHere>';
go
CREATE CERTIFICATE MyServerCert WITH SUBJECT = 'My DEK Certificate';
go
USE MyDatabase;
GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_128
ENCRYPTION BY SERVER CERTIFICATE MyServerCert;
GO
ALTER DATABASE MyDatabase
SET ENCRYPTION ON;
GO
```



Single Sign-On

Synapse Foundation Components
 Synapse Linked Services



Azure Synapse Analytics Customers



SQL Analytics

new features available

GA features:

- **Performance:** Resultset caching
- **Performance:** Materialized Views
- **Performance:** Ordered columnstore
- **Heterogeneous data:** JSON support
- **Trustworthy:** Dynamic Data Masking
- **Continuous integration & deployment:** SSDT support
- **Language:** Read committed snapshot isolation

Public preview features:

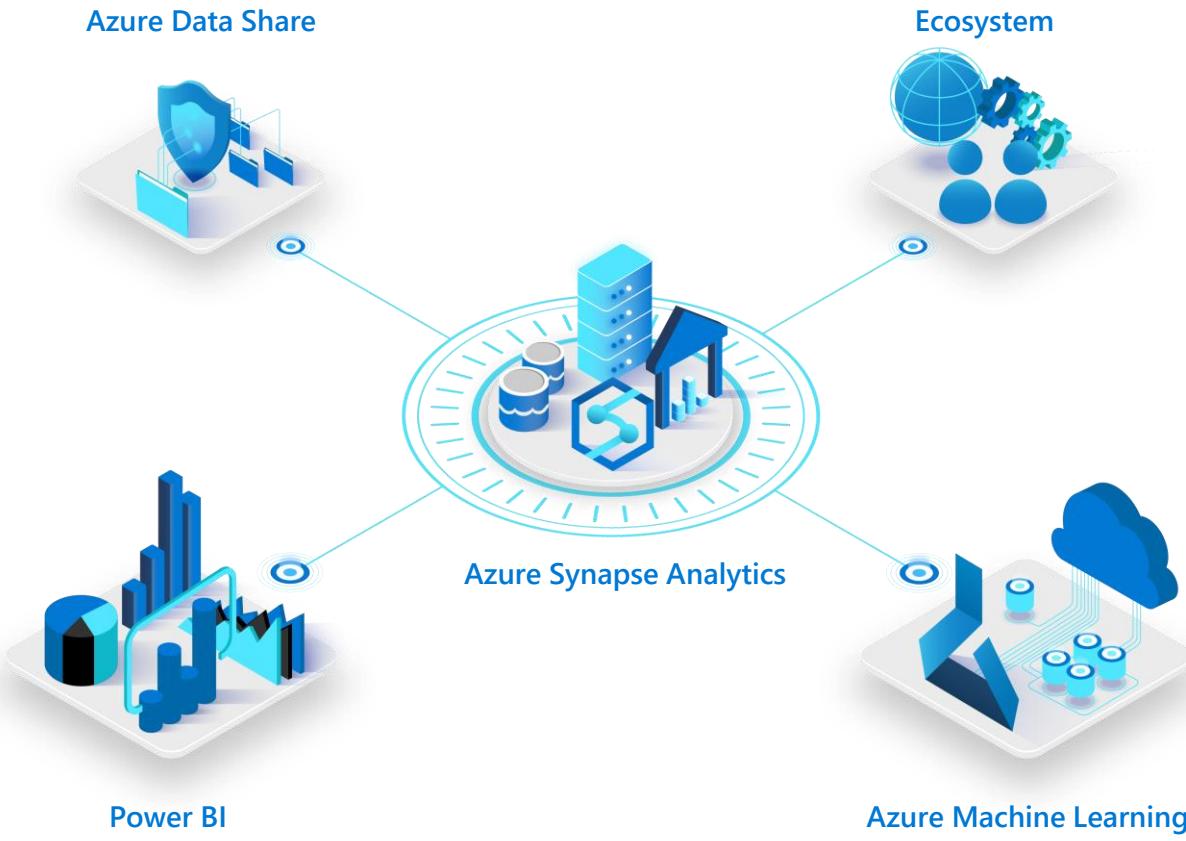
- **Workload management:** Workload Isolation
- **Data ingestion:** Simple ingestion with COPY
- **Data Sharing:** Share DW data with Azure Data Share
- **Trustworthy:** Private LINK support

Private preview features:

- **Data ingestion:** Streaming ingestion & analytics in DW
- **Built-in ML:** Native Prediction/Scoring
- **Data lake enabled:** Fast query over Parquet files
- **Language:** Updateable distribution column
- **Language:** FROM clause with joins
- **Language:** Multi-column distribution support

Note: private preview features require whitelisting

Leveraging ISV partners with Azure Synapse Analytics



+ many more

Full backward compatibility with Azure SQL Data Warehouse for data integration and orchestration

Additional analytics capabilities in Azure Synapse unlocks new ISV scenarios

Azure Synapse + ISV can bring data continuity with Azure Machine Learning and Power BI

Reduce migration effort by reusing existing partner platforms

Get started today

[Learn more ➔](#)

[Start building with a free trial ➔](#)

[Attend the Analytics in a Day workshop ➔](#)