



Azure **Synapse** Analytics

- **Adrián J. Fernández Zenteno**
- *Sr. Cloud Solution Architect - Azure Advanced Analytics*
- Email: adrian.fernandez@microsoft.com
- Twitter: @AdrianFZ10

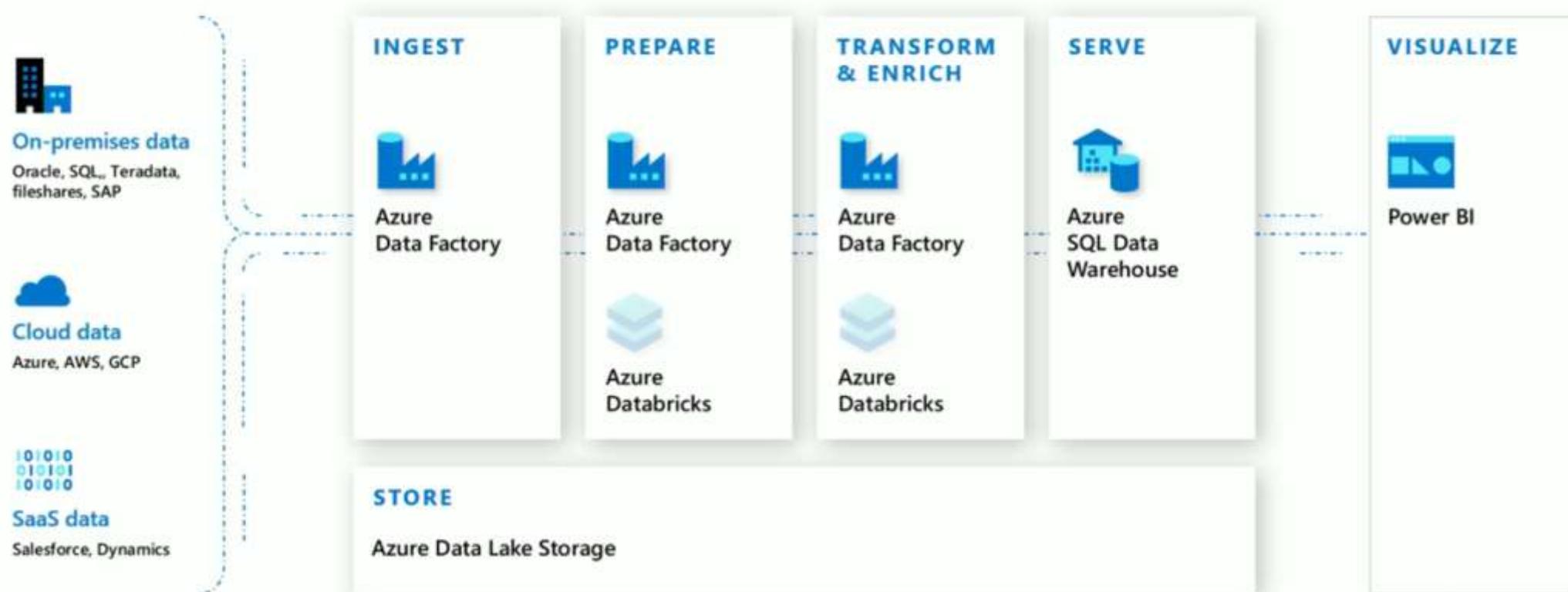




Azure Synapse Analytics

- Introduction
- Studio
- Data Integration
- Execution Pools:
 - SQL Analytics (*formerly SQL Datawarehouse*)
 - SQL On-Demand
 - Spark
- Foundation
- Connected Services

Modern Data Warehouse



Azure Synapse Analytics - *Data Lakehouse*



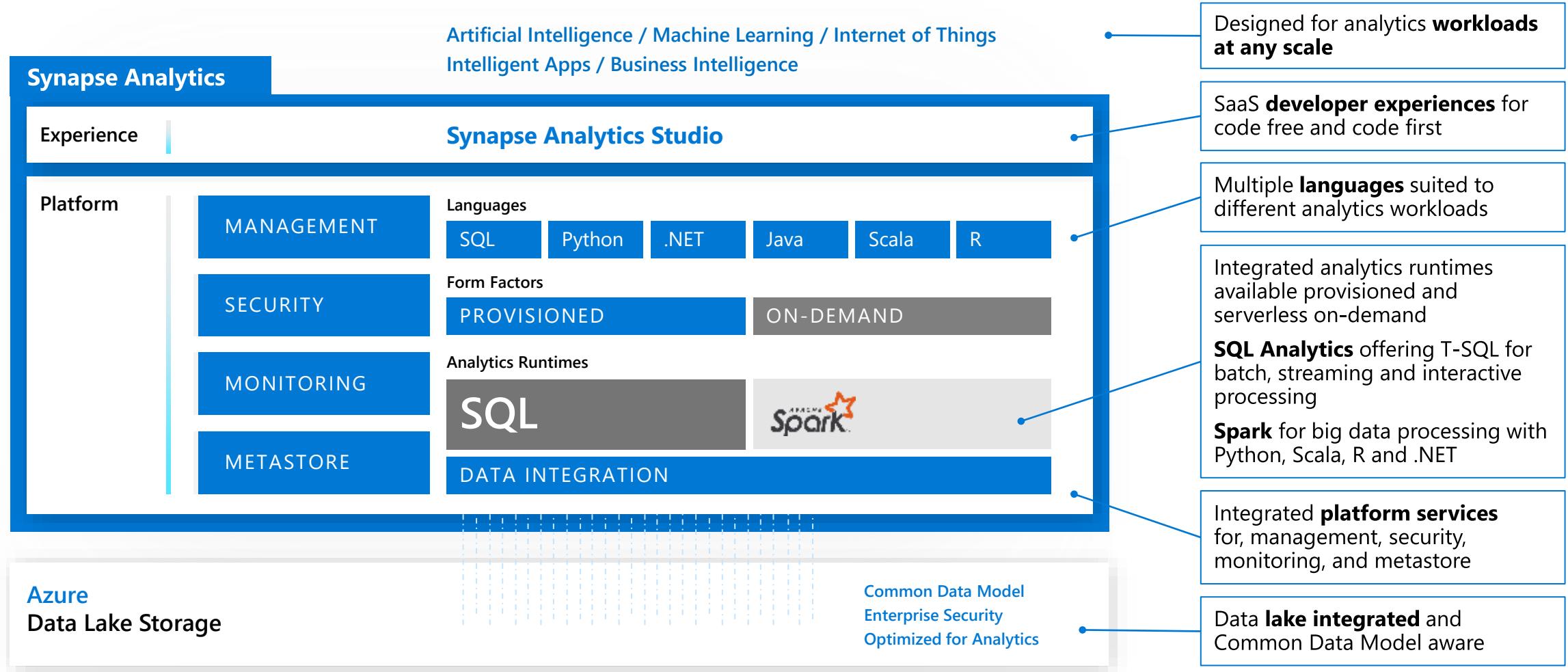


Azure Synapse Analytics

*Azure Synapse Analytics is a **limitless** analytics service, that brings **together enterprise data warehousing and Big Data analytics**. It gives you the freedom to query data on your terms, using either **serverless on-demand or provisioned resources, at scale**. Azure Synapse brings these two worlds together with a unified experience to ingest, prepare, manage, and serve data for immediate business intelligence and machine learning needs.*

Azure Synapse Analytics

Integrated data platform for BI, AI and continuous intelligence



Provisioning Synapse workspace

Providing Synapse is easy

Subscription

Resource Group

Workspace Name

Region

Data Lake Storage Account

Home > Synapse workspaces > Create Synapse workspace

Create Synapse workspace

Basics * Security + networking Tags Summary

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all of your resources.

Subscription * ⓘ

Resource group * ⓘ Select existing...

Workspace details

Name your workspace, select a location, and choose a primary Data Lake Storage Gen2 file system to serve as the default location for logs and job output.

Workspace name * Enter workspace name

Region *

Select Data Lake Storage Gen2 * ⓘ From subscription Manually via URL

Account Name *

File system name *

The managed identity of the workspace will be assigned the [Storage Blob Data Contributor](#) role on the selected Data Lake Storage Gen2 file system, granting it full data access.

Synapse workspace

 internal sandboxwe
Synapse workspace

Search (Ctrl + F)

+ New SQL pool + New Apache Spark pool Refresh Reset SQL admin password Delete Launch Synapse Studio

Resource group (change) : Arcadia-Private-Preview-BASE
Status : Succeeded
Location : West Europe
Subscription (change) : BigDataPMInternal
Subscription ID : 58f8824d-32b0-4825-9825-02fa6a801546
Managed Identity object... : 5eff8ac2-fd6f-4b09-84fd-760bab64802c

Firewalls : Show firewall settings
Primary ADLS Gen2 acc... : https://internalsandboxwe.dfs.core.windows.net
Primary ADLS Gen2 file ... : tempdata
SQL Active Directory ad... : acomet@microsoft.com
SQL endpoint : internalsandboxwe.sql.azuresynapse.net
SQL on-demand endpoint : internalsandboxwe-ondemand.sql.azuresynapse.net
Development endpoint : https://internalsandboxwe.dev.azuresynapse.net
Workspace web URL : https://web.azuresynapse.net?workspace=%2bsubscr

Tags (change) : pointOfContact : <unknown>

Available resources

Search to filter items...

| Name | Size | Type |
|--------------------|---------|-------------------|
| SQL pools | | |
| SQLPoolSandbox | DW1000c | SQL pool |
| Apache Spark pools | | |
| SparkSandbox | Medium | Apache Spark pool |

SQL pools

Apache Spark pools

Firewalls

Alerts

Metrics

Diagnostic settings

Logs

Advisor recommendations

New support request

SQL pools

+ New Refresh

Search to filter items...

| Name | Type | Status | Size |
|-----------------|-------------------------|----------|---------|
| SQL on-demand | SQL Analytics on-demand | N/A | N/A |
| SQLPoolSandbox | SQL Analytics pool | ✓ Online | DW1000c |
| SQLSandboxLarge | SQL Analytics pool | ✓ Online | DW2000c |
| SQLSandboxSmall | SQL Analytics pool | ✓ Online | DW100c |

Create SQL pool

Synapse

Basics * Additional settings * Tags Review + create

Create a SQL pool with your Preferred Configuration. Complete the basics tab then go to Review + create provision with smart defaults. [Learn more](#)

SQL pool Details

Name your SQL pool and choose its initial settings.

SQL pool Name *

Enter SQL pool Name

Performance level ⓘ



DW1000c

Basics *

Additional settings *

Tags

Review + create

Customize additional configuration parameters including collation & sample data.

Data source

Start with a blank SQL pool, restore from a backup or select sample data to populate your new SQL pool.

Use existing data *

None Backup

SQL pool collation

Collation defines the rules that sort and compare data, and cannot be changed after SQL pool creation. The default collation is SQL_Latin1_General_CI_AS. [Learn more](#)

Collation *

SQL_Latin1_General_CI_AS

SQL Analytics pool = SQL Data Warehouse

Apache Spark pools

+ New Refresh

Search to filter items...

Name

- SparkSandbox
- SparkSmall
- SparkLarge

Create Apache Spark pool

Basics * Additional settings * Tags Sun

Create a Synapse Analytics Apache Spark pool with create to provision with smart defaults, or visit each

Apache Spark pool details

Name your Apache Spark pool and choose its initial

Apache Spark pool name *

Node size family *

Node size *

Large (16 vCPU / 128 GB)

Number of nodes *

3 200

Autoscale * Enabled Disabled

Number of nodes *

3 40

Note: There are no on-demand pools for Spark

Scale Apache Spark pool: SparkEngine100

Configure the settings that best align with the workload on the Apache Spark pool.

Autoscale Enabled Disabled

Scale details

Node size family MemoryOptimized

Node size *

Large (16 vCPU / 128 GB)

Number of nodes *

3 200

Autoscale * Enabled Disabled

Number of nodes *

3 40

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 Select a file

Synapse workspace

supplychaindataanalytics

[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.



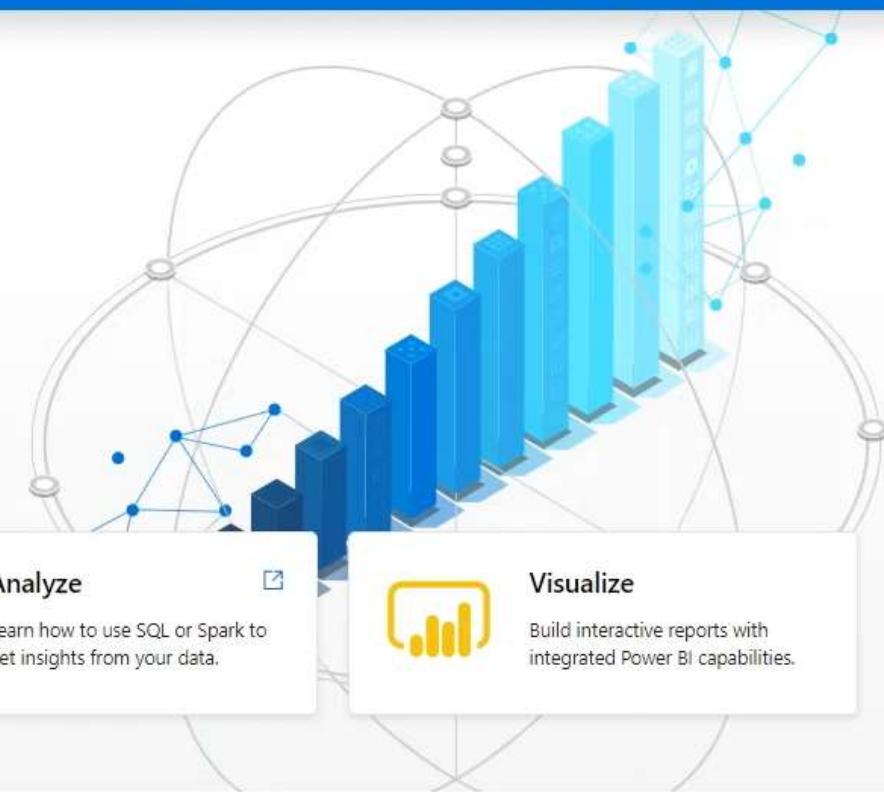
Analyze

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](#) ▾

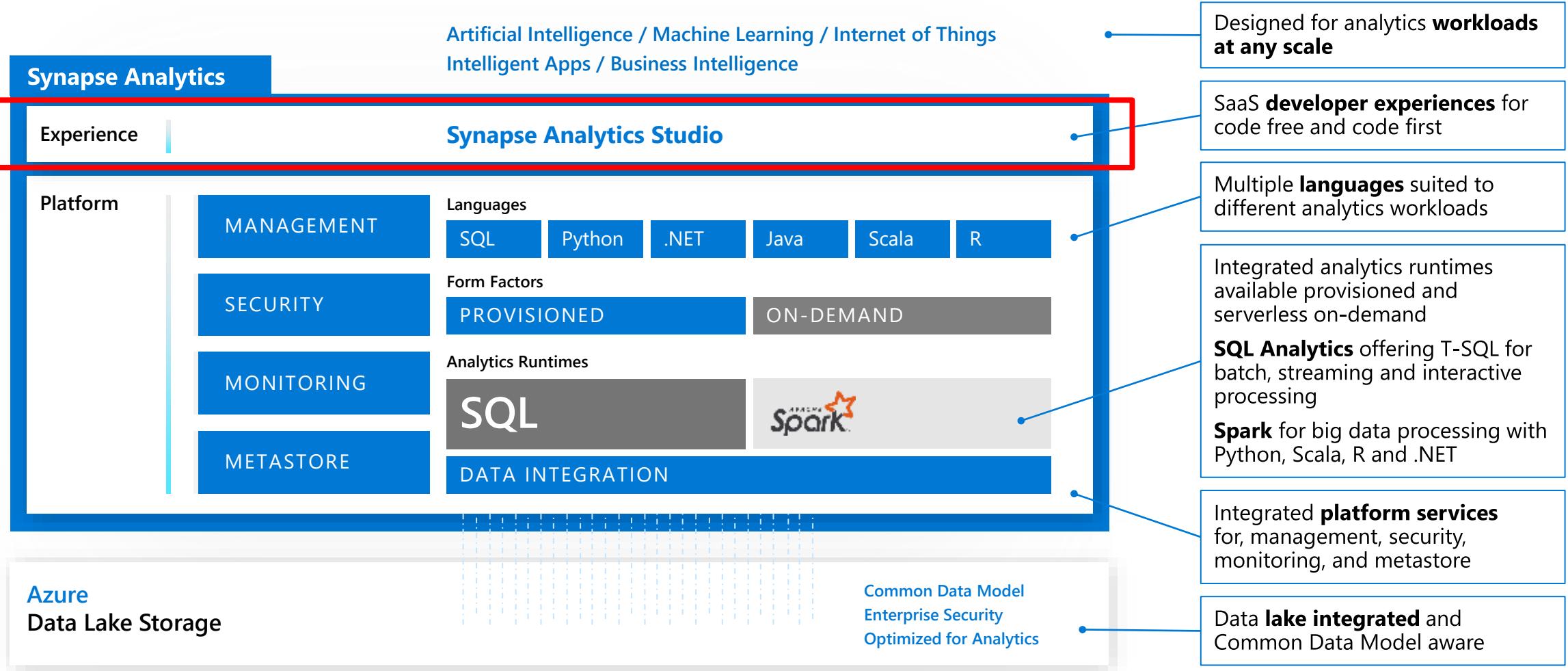
Share your comments or suggestions with us to improve Synapse.



Azure Synapse Analytics Studio

Azure Synapse Analytics

Integrated data platform for BI, AI and continuous intelligence



Studio

<https://web.azuresyanapse.net>

Microsoft Azure



Select workspace

Azure Synapse Analytics is a limitless cloud data warehouse with unmatched time-to-insight. [Learn more](#)

Azure Active Directory *

Microsoft

Subscription

Adrian Microsoft Azure Internal Consumption (80a07312-7e2c-49...)

Workspace name *

synapse101ws

Continue

Studio

<https://web.azuresyanapse.net>

Microsoft Azure | Synapse Analytics > synapse101ws

Home Data Develop Orchestrate Monitor Manage

Synapse workspace
synapse101ws

New <

Ingest Explore Analyze Visualize

Resources

Recent Pinned

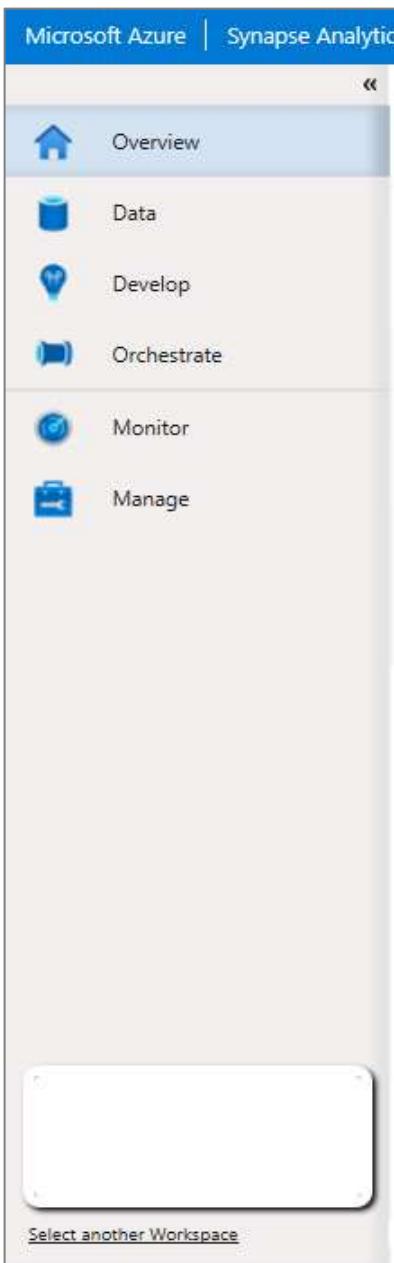
| NAME | LAST OPENED BY YOU |
|--|--------------------|
| Python installed packages | 26 minutes ago |
| 07 Data Exploration and ML Modeling - NYC taxi predict using Spark MLlib | 26 minutes ago |
| 02- Analyze NYC Taxi dataset | 26 minutes ago |

Useful links

- [Getting started](#)
Samples, guide and tour to get you started.
- [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.

Studio

<https://web.azuresynapse.net>



- **Data:** Shows the available data sources available to the workspace. These can exist internally in the workspace (such as a SQL compute database or a Spark database), or externally (such as a Data Lake Store Gen2, or Azure Blob Storage account)
- **Develop:** Shows the different objects used to query or operate with the data, such as SQL scripts, notebooks, data flows, Spark job definitions, Power BI, etc
- **Orchestrate:** Shows the objects used to automate analytics processes (such as pipelines, datasets, etc.)
- **Monitor:** Shows metrics for pipeline runs, trigger runs, integration runtimes, and spark applications
- **Manage:** Create linked services, pipeline triggers, integration runtimes, and manage access to Synapse

Data Hub

Overview

Access to storage accounts, databases, and datasets (for Data Integration)

The screenshot shows the Microsoft Azure Synapse Analytics Data Hub interface. At the top, there are buttons for 'Publish all' (with 5 pending changes) and 'Validate all'. Below this is a navigation bar with icons for Home, Data, Storage, Databases, and Datasets. The 'Data' icon is selected. A search bar says 'Filter resources by name'. The main pane displays a hierarchical list of resources:

- Storage accounts:**
 - internalsandboxwe (Primary)
 - bwalker
 - opendataset
 - tempdata- Databases:**
 - SQLPoolSandbox (SQL pool)
 - Tables
 - Views
 - Programmability
 - External resources
 - Security
 - Schemas
 - default (SQL on-demand)
 - External tables
 - Views
 - External resources
 - Security
 - Schemas
 - default (Spark)
 - Tables
- Datasets:**
 - Address

Develop Hub

Overview

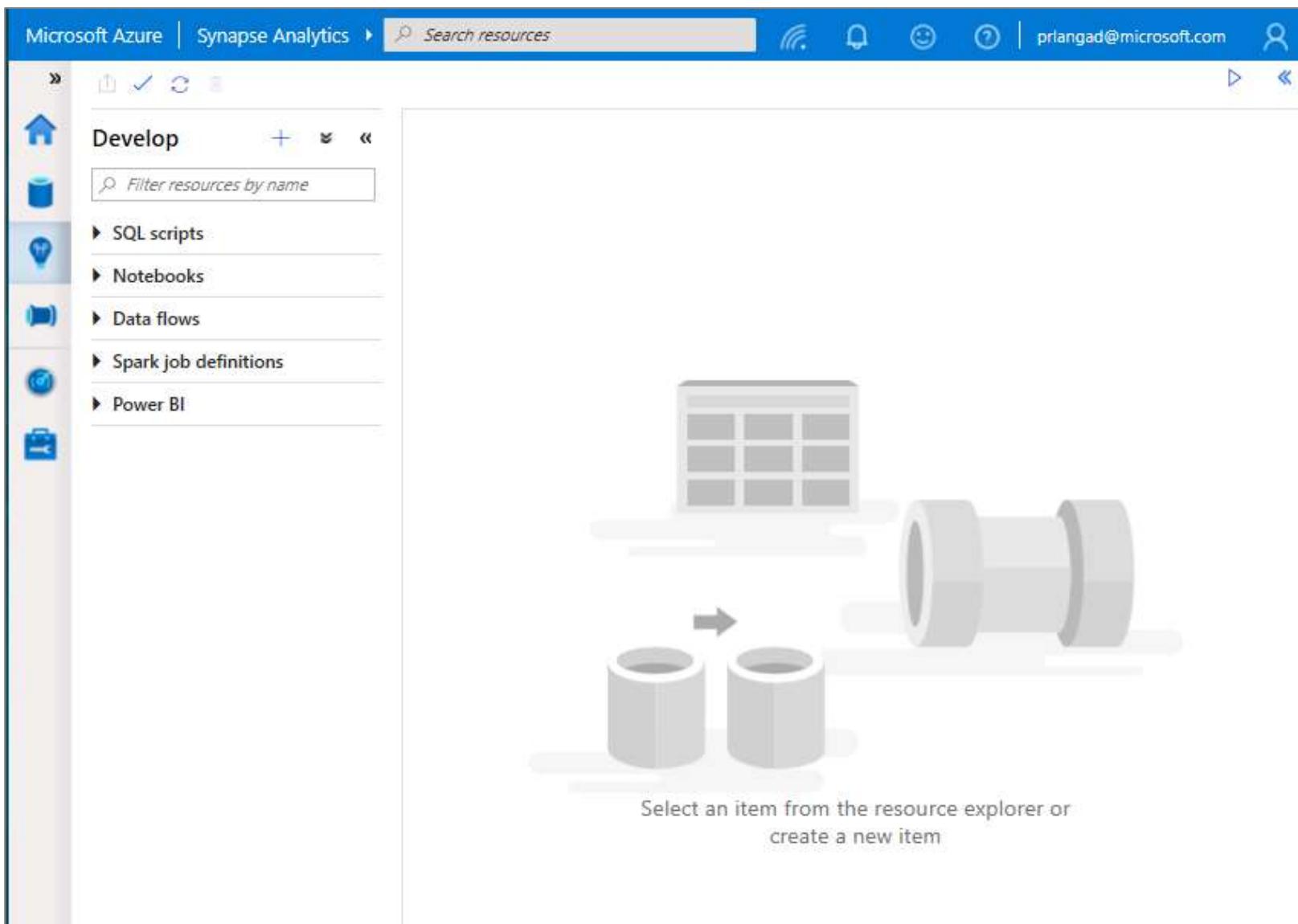
It provides development experience to query, analyze, model data

Benefits

Multiple languages to analyze data under one umbrella

Switch over notebooks and scripts without losing content

Code intellisense offers reliable code development



Develop Hub/SQL scripts

SQL Script

Authoring SQL Scripts

Execute SQL script on provisioned SQL Pool or SQL On-demand

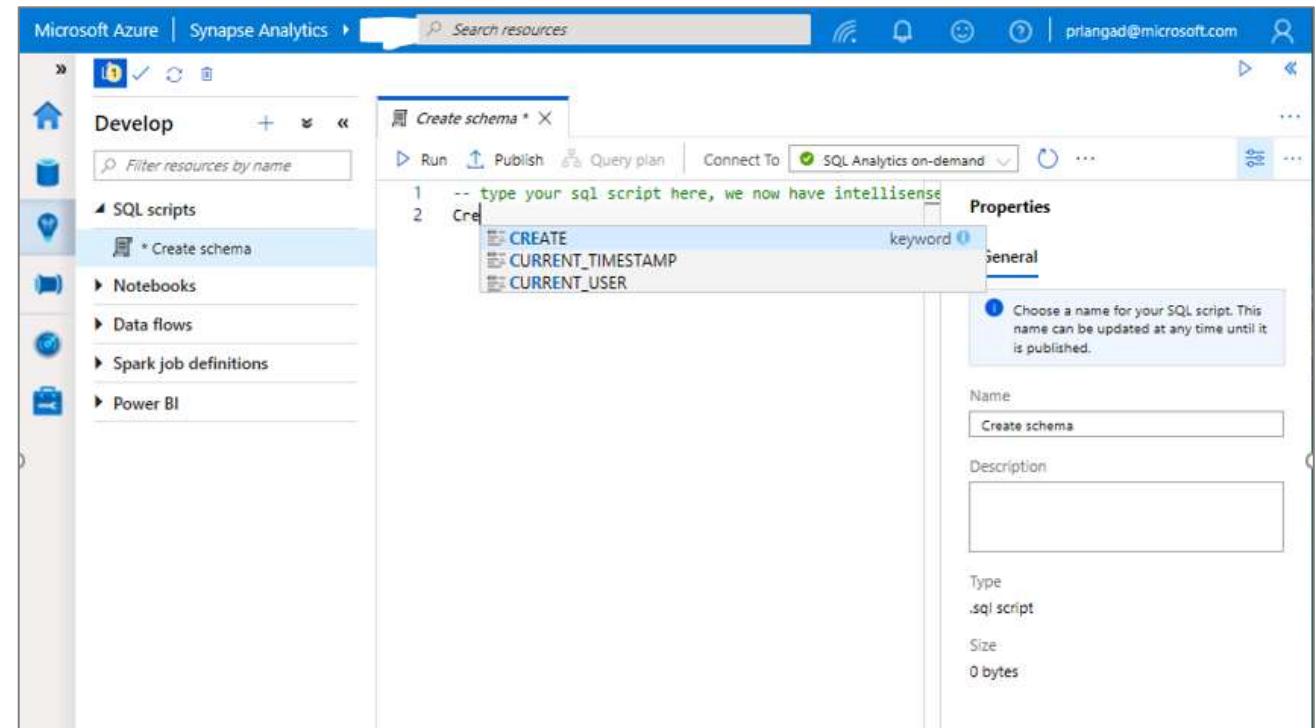
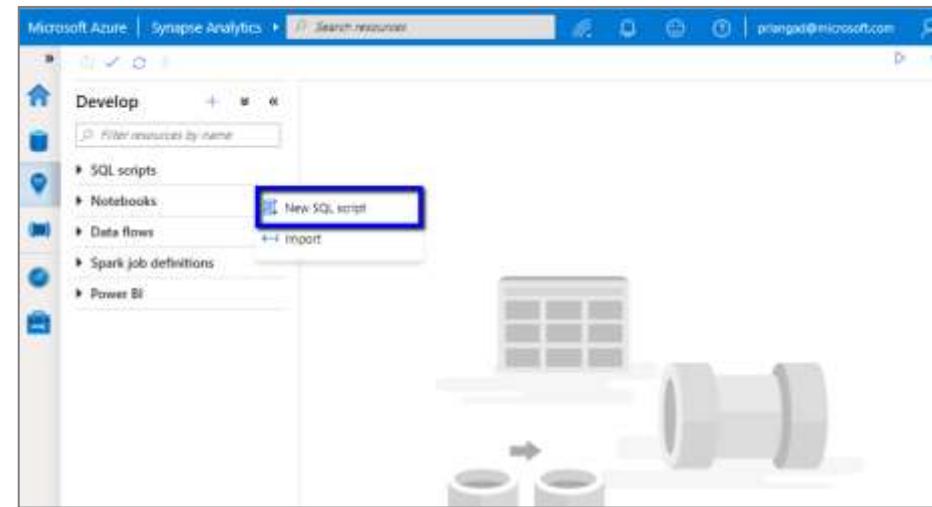
The key thing here is that a "pool" is a set of pre-provisioned resources for DW style workloads. In fact, it's SQLDW with net new enhancements

Publish individual SQL script or multiple SQL scripts through Publish all feature

Language support and intellisense

View result in tabular or chart format

Export result



Develop Hub/Notebooks

Notebooks

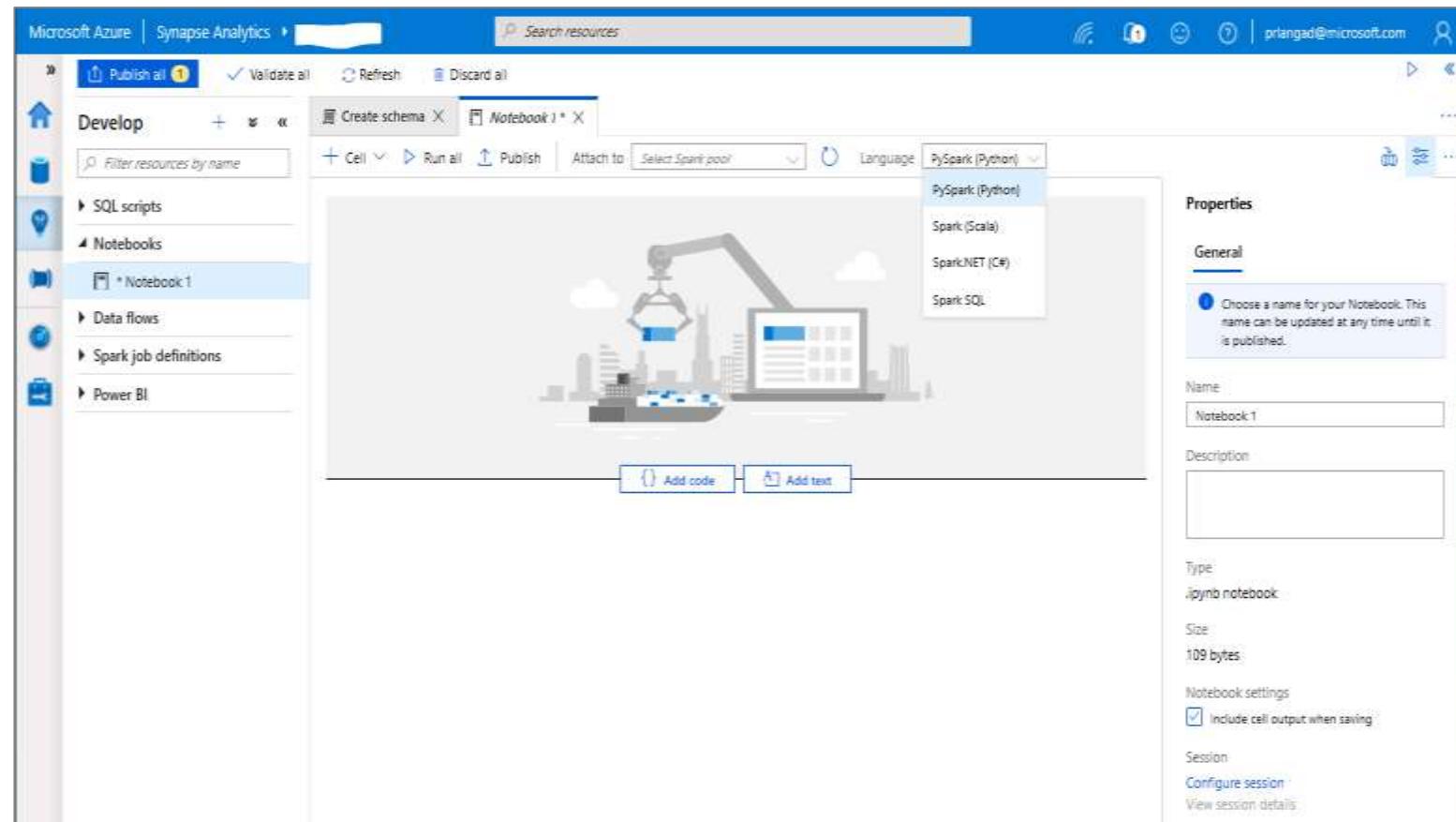
Allows to write multiple languages in one notebook

```
%%<Name of language>
```

Offers use of temporary tables across languages

Language support for Syntax highlight, syntax error, syntax code completion, smart indent, code folding

Export results



Orchestrate Hub

Overview

It provides ability to create pipelines to ingest, transform and load data with 90+ inbuilt connectors

The screenshot shows the Microsoft Azure Synapse Analytics Studio Orchestrate Hub. The left sidebar displays navigation icons for Home, Pipelines, Datasets, Triggers, and Jobs. The Pipelines section is currently selected, showing three existing pipelines: 'Copy Open Dataset', 'Pipeline 1', and 'Load Data to SQLDW'. The 'Load Data to SQLDW' pipeline is highlighted. The main workspace shows a 'Copy data' activity named 'WeatherData' being configured. The activity details pane at the bottom shows the following configuration:

| General | Parameters | Variables | Output |
|-------------|--------------------|-----------|--------|
| Name * | Load Data to SQLDW | | |
| Description | | | |
| Concurrency | | | |
| Annotations | + New | | |

Monitor Hub

Overview

This feature provides ability to monitor orchestration, activities and compute resources.

The screenshot shows the Microsoft Azure Synapse Analytics Monitor Hub interface. The left sidebar has icons for Home, Pipeline runs (selected), Trigger runs, Integration runtimes, Activities (Spark applications), Computers (SQL Pools), and Annotations. The main area is titled "Pipeline runs" and shows two completed runs:

| Pipeline Name | Run Start | Duration | Triggered By | Status | Annotations | Errors |
|--------------------|------------------------|----------|----------------|-----------|-------------|--------|
| Load Data to SQLDW | 10/25/2019, 3:49:42 PM | 00:10:55 | Manual trigger | Succeeded | | |
| Copy Open Dataset | 10/25/2019, 2:17:54 PM | 00:14:12 | Manual trigger | Succeeded | | |

Monitor Hub

Microsoft Azure | Synapse Analytics ▶ synapse101ws adrianjf@microsoft.com MICROSOFT

Copy data

1 Properties One time copy

2 Source Azure Data Lake Storage Gen2

Connection
Dataset

3 Destination Azure Data Lake Storage Gen2

Connection
Dataset

4 Settings

5 Summary

6 Deployment

Azure Data Lake Storage Gen2 → Azure Data Lake Storage Gen2

Deployment complete

- Validate copy runtime environment ✓
- Creating datasets ✓
- Creating pipelines ✓
- Running pipelines ✓

Datasets and pipelines have been created. You can now monitor and edit the copy pipelines or click finish to close the copy wizard.

Edit pipeline Monitor

Finish

Manage Hub

Overview

This feature provides ability to manage Linked Services (in Data Integration), Orchestration and Security.

The screenshot shows the Microsoft Azure Synapse Analytics Studio interface. The left sidebar has icons for Home, External connections, Linked services (which is selected and highlighted in blue), Orchestration, Triggers, Integration runtimes, Security, and Access control. The top navigation bar includes Publish all (with 1 update), Validate all, Refresh, Discard all, and user information (prlangad@microsoft.com). The main content area is titled "Linked services" and contains a sub-instruction: "Linked services are much like connection strings, which define the connection information needed for Arcadia to connect to external resources." Below this is a table with columns: NAME, TYPE, and ANNOTATIONS. The table lists seven linked services:

| NAME | TYPE | ANNOTATIONS |
|-----------------------------|------------------------------|-------------|
| ADLSG2OpenDataSetSink | Azure Data Lake Storage Gen2 | |
| AzureBlobStorage1 | Azure Blob Storage | |
| AzureDataLakeStorage1 | Azure Data Lake Storage Gen2 | |
| AzureDataLakeStorage2Source | Azure Data Lake Storage Gen2 | |
| AzureOpenDataset | Azure Blob Storage | |
| AzureOpenDataSet2 | Azure Blob Storage | |
| AzureSqlDW1 | Azure Synapse Analytics | |

There is also a "New" button and a search/filter bar at the top of the table.

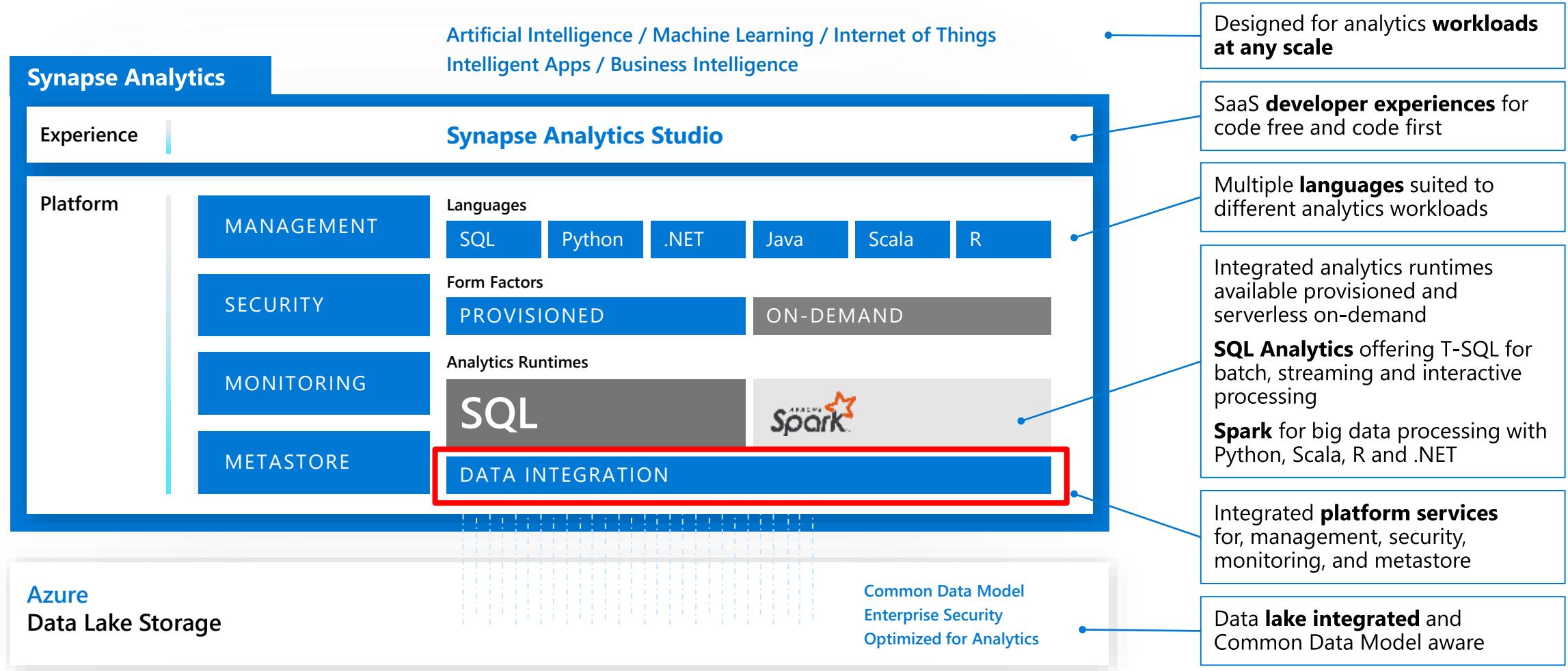


Azure Synapse Analytics

Data Integration

Azure Synapse Analytics

Integrated data platform for BI, AI and continuous intelligence



Data Integration = Separate version Azure Data Factory (ADF). Will have 1-click migration

Manage – Linked Services

Overview

It defines the connection information needed for Pipeline to connect to external resources.

Benefits

Offers pre-build 90+ connectors

Easy cross platform data migration

Represents data store or compute resources

The screenshot shows the Microsoft Azure Synapse Analytics interface for managing linked services. The left sidebar has 'External connections' selected, with 'Linked services' highlighted. A 'New' button is visible. The main area displays a table of existing linked services:

| NAME | TYPE | ANNOTATIONS |
|-----------------------------|------------------------------|-------------|
| ADLSG2OpenDataSetSink | Azure Data Lake Storage Gen2 | |
| AzureBlobStorage1 | Azure Blob Storage | |
| AzureDataLakeStorage1 | Azure Data Lake Storage Gen2 | |
| AzureDataLakeStorage2Source | | |
| AzureOpenDataset | | |
| AzureOpenDataSet2 | | |
| AzureSqlDW1 | | |

A large blue arrow points from the 'New' button to a modal dialog titled 'New linked service (Power BI)'. The dialog contains fields for 'Name' (PowerBIWorkspace1), 'Description', and 'Workspace name'. Below the dialog is a grid of connector icons:

| PayPal (Preview) | Phoenix | PostgreSQL |
|------------------------|------------------|----------------------|
| Power BI | Presto (Preview) | QuickBooks (Preview) |
| REST | SAP BW Open Hub | SAP BW via MDX |
| SAP Cloud for Customer | SAP ECC | SAP HANA |
| SAP | SAP | SAP |

At the bottom of the modal are 'Continue' and 'Cancel' buttons.

Manage – Integration runtimes

Overview

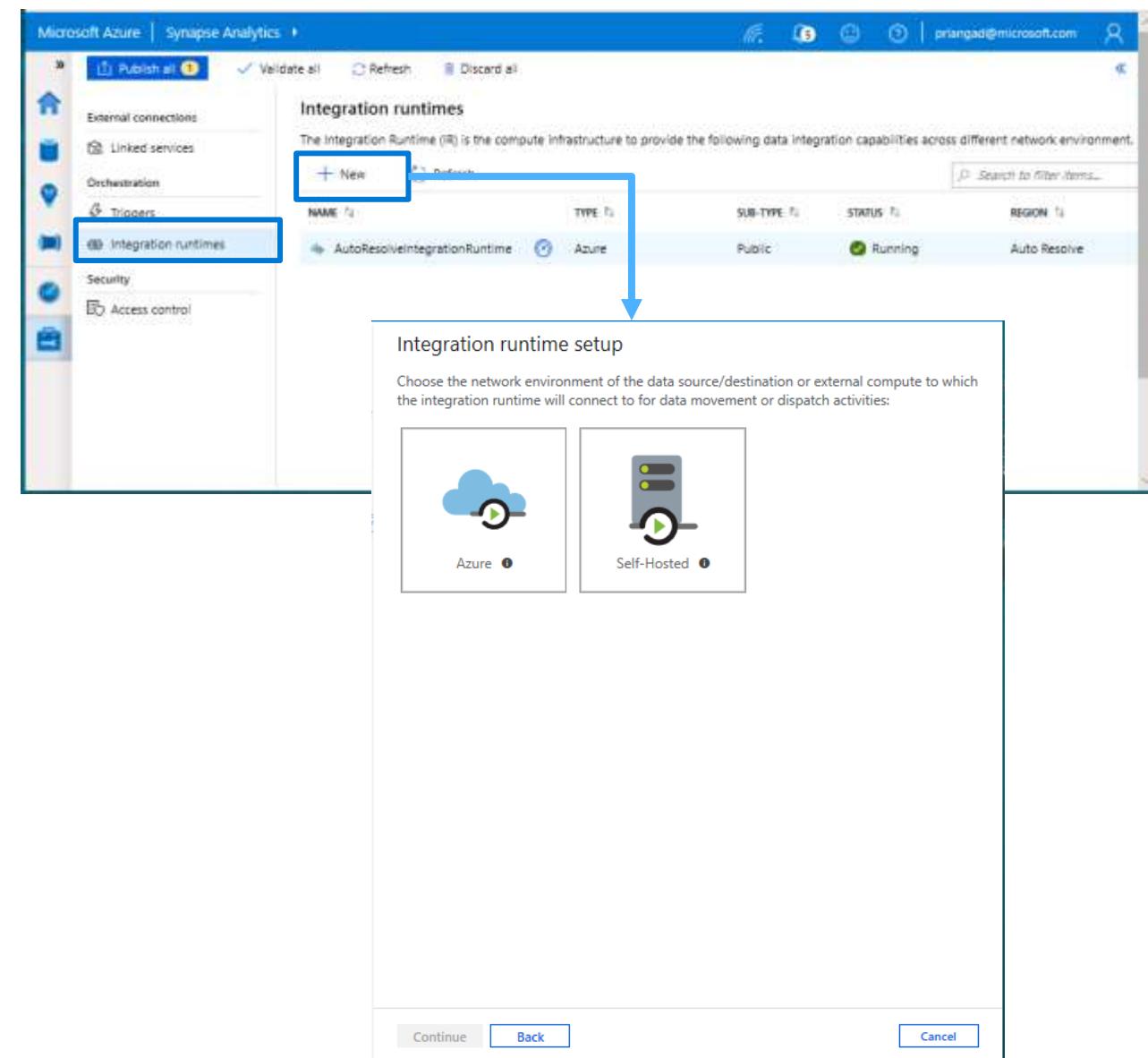
It is the compute infrastructure used by Pipelines to provide the data integration capabilities across different network environments. An integration runtime provides the bridge between the activity and linked Services.

Benefits

Offers Azure Integration Runtime or Self-Hosted Integration Runtime

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

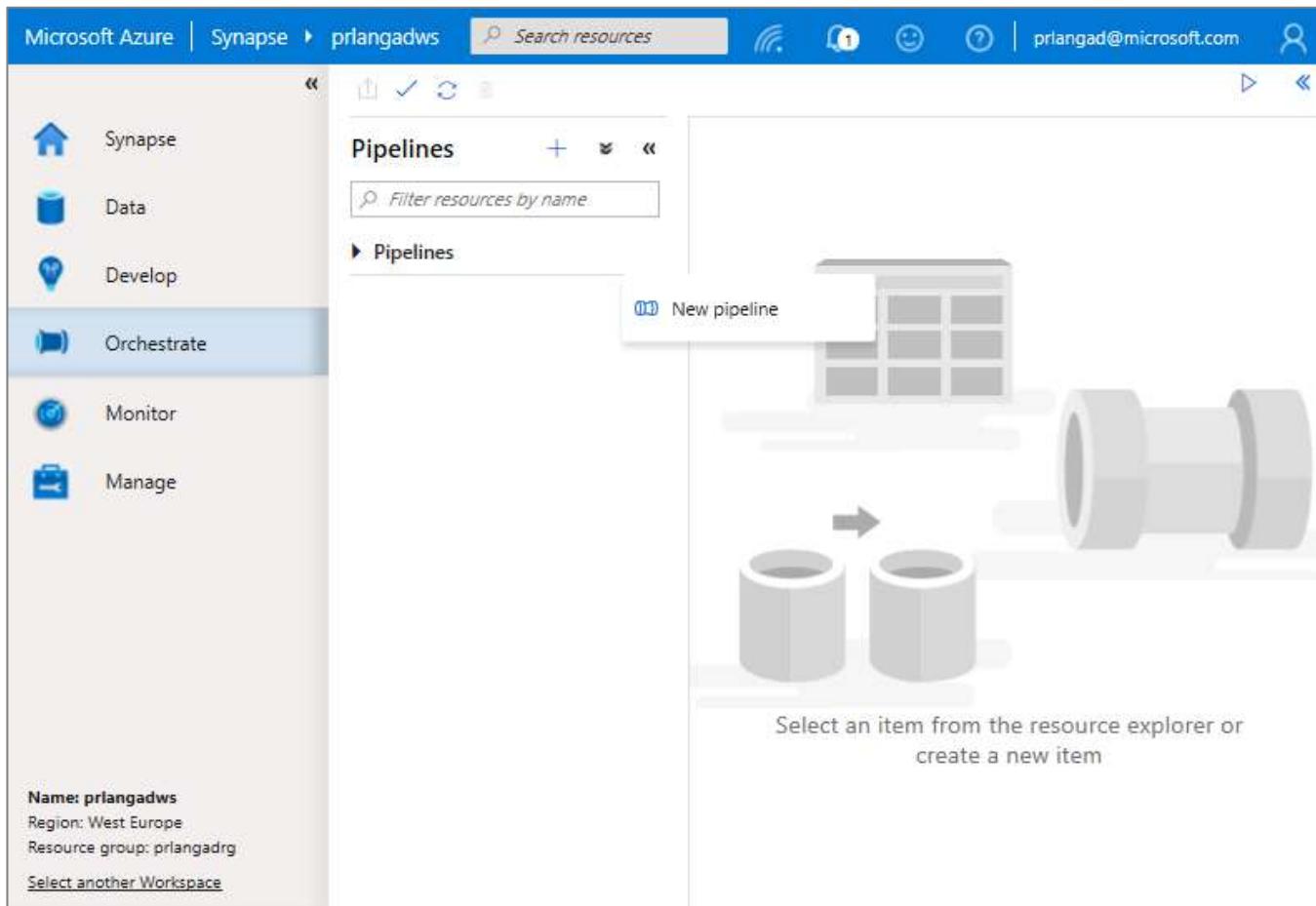
Self-Hosted Integration Runtime – use compute resources in on-premises machine or a VM inside private network



Pipelines

Overview

- Scalable – per job elasticity up to 4GB/s
- Visually author or via code (Python, Spark(Scala), Spark.NET,(C#) etc.)
- Serverless, no infrastructure to manage
- 24 points of presence worldwide
- Self-hostable Integration Runtime for hybrid movement
- Supported file formats: CSV, AVRO, ORC, Parquet, JSON



90+ Connectors out of the box

| Azure (15) | Database & DW (26) | | File Storage (6) | NoSQL (3) | Services and App (28) | | | Generic (4) |
|-------------------------|--------------------|-----------------|----------------------|-----------|------------------------------|----------------------------|--|---------------|
| Blob storage | Amazon Redshift | Oracle | Amazon S3 | Cassandra | Amazon MWS | Oracle Service Cloud | | Generic HTTP |
| Cosmos DB - SQL API | DB2 | Phoenix | File system | Couchbase | Common Data Service | PayPal | | Generic OData |
| Cosmos DB - MongoDB API | Drill | PostgreSQL | FTP | MongoDB | Concur | QuickBooks | | Generic ODBC |
| Data Explorer | Google BigQuery | Presto | Google Cloud Storage | | Dynamics 365 | Salesforce | | Generic REST |
| Data Lake Storage Gen1 | Greenplum | SAP BW Open Hub | HDFS | | Dynamics AX | Salesforce Service Cloud | | |
| Data Lake Storage Gen2 | HBase | SAP BW via MDX | SFTP | | Dynamics CRM | Salesforce Marketing Cloud | | |
| Database for MariaDB | Hive | SAP HANA | Google AdWords | | SAP Cloud for Customer (C4C) | | | |
| Database for MySQL | Apache Impala | SAP table | HubSpot | | SAP ECC | | | |
| Database for PostgreSQL | Informix | Spark | | | Jira | ServiceNow | | |
| File Storage | MariaDB | SQL Server | | | Magento | Shopify | | |
| SQL Database | Microsoft Access | Sybase | | | Marketo | Square | | |
| SQL Database MI | MySQL | Teradata | | | Office 365 | Web table | | |
| SQL Data Warehouse | Netezza | Vertica | | | Oracle Eloqua | Xero | | |
| Search index | | | | | Oracle Responsys | Zoho | | |
| Table storage | | | | | | | | |

Load Data

Overview

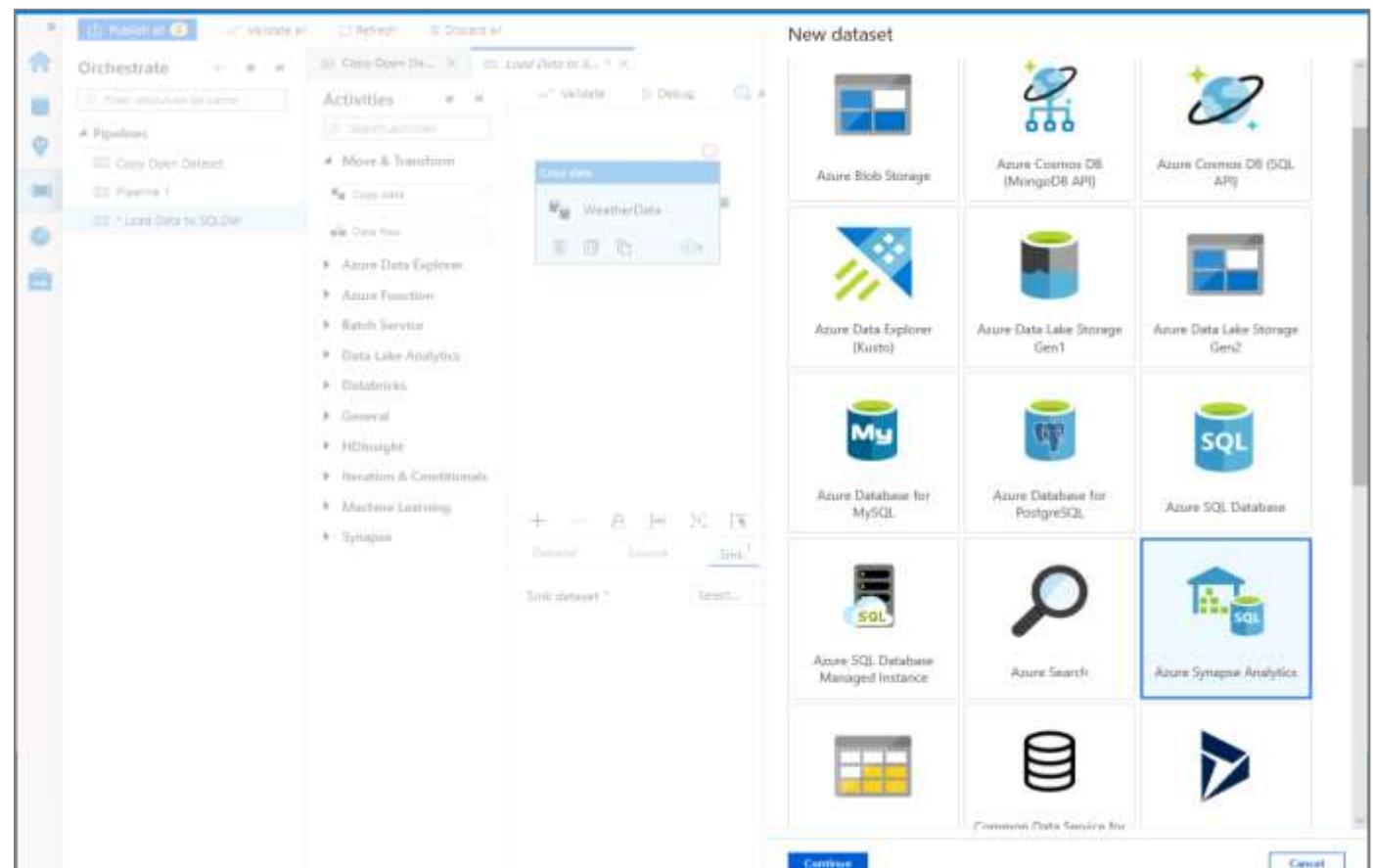
It provides the ability to load data from storage account to desired linked service. Load data by manual execution of a pipeline or by orchestration

Benefits

Supports common loading patterns

Polybase for data loading @ scale

Note there is a new COPY statement for loading data



Prep & Transform Data

Overview

It offers data cleansing, transformation, aggregation, conversion, etc

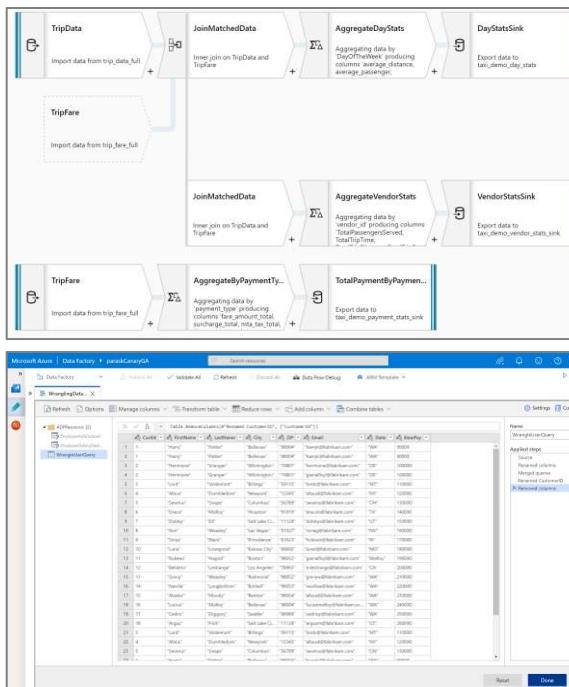
Benefits

Cloud scale via Spark execution

Guided experience to easily build resilient data flows

Flexibility to transform data per user's comfort

Monitor and manage dataflows from a single pane of glass

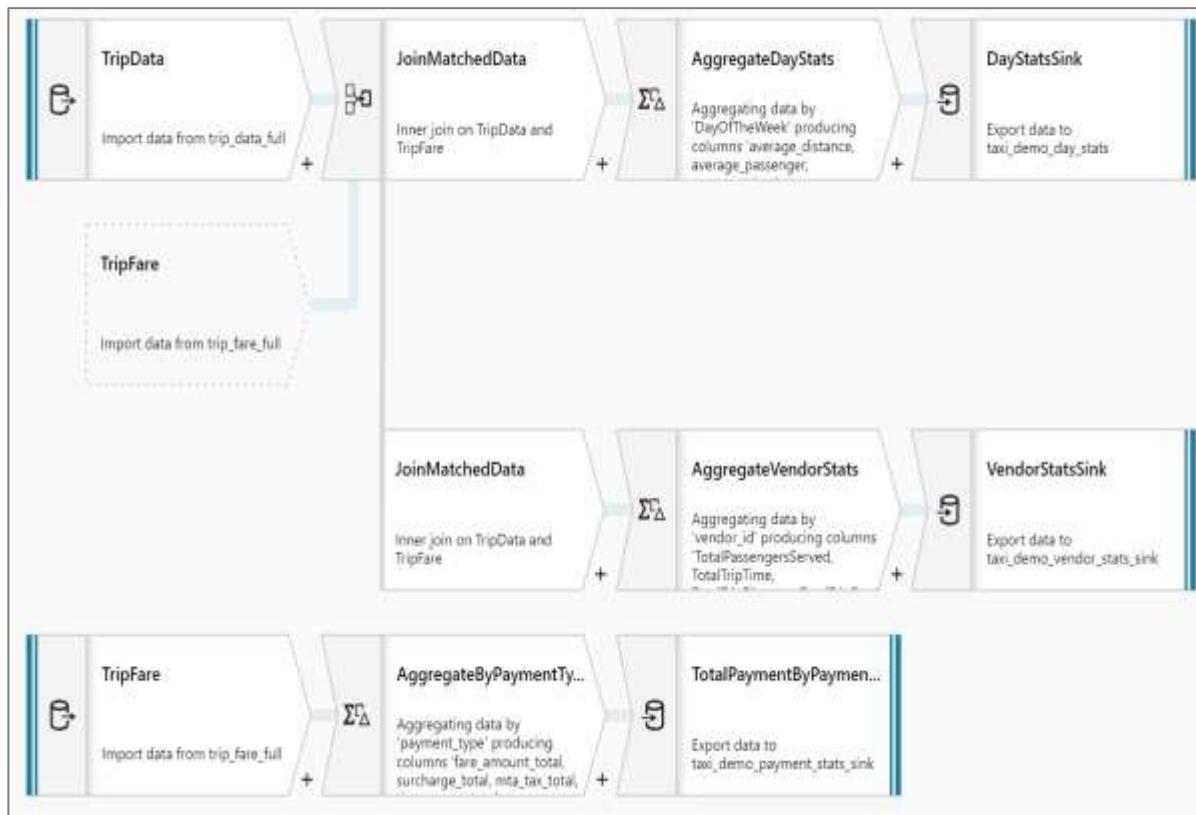


...
not

Prep & Transform Data

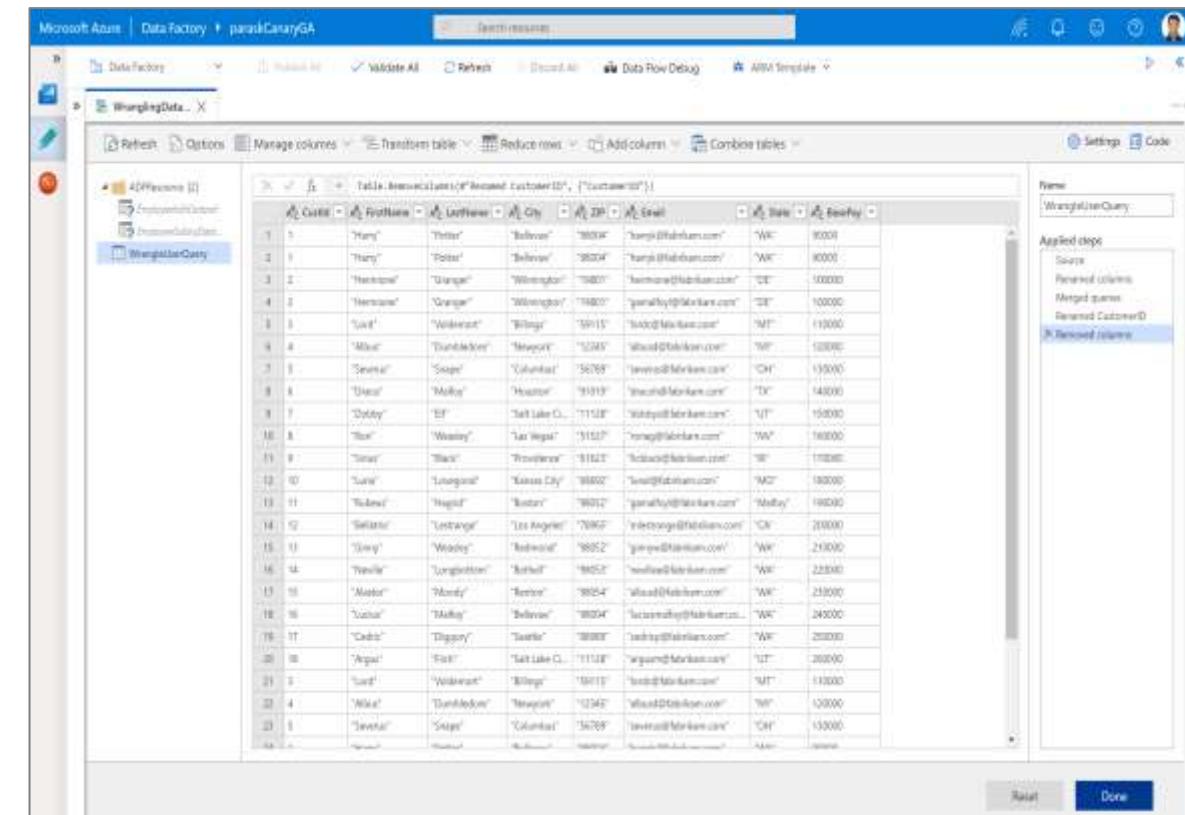
Mapping Dataflow

Code free data transformation @scale



Wrangling Dataflow

Code free data preparation @scale



Data Flows

Overview

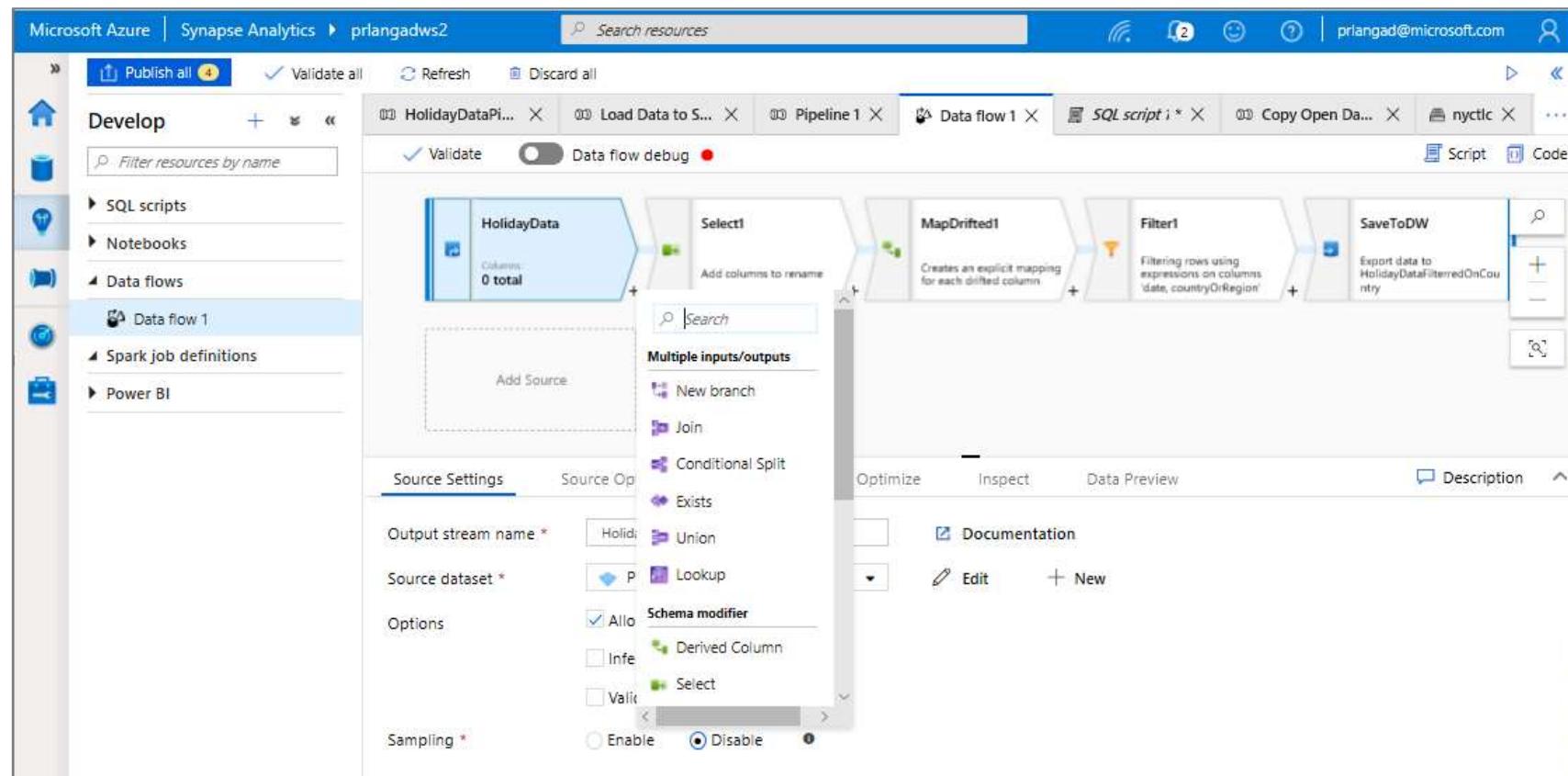
It provides ability to transform data at scale without any coding required.

Offers data manipulation operations such as schema modifier, row modifier, input/output – join, branch, union etc.

Benefits

Use data flows as part of pipeline to transform data.

Debug data flows offers data preview to verify applied transformation.



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

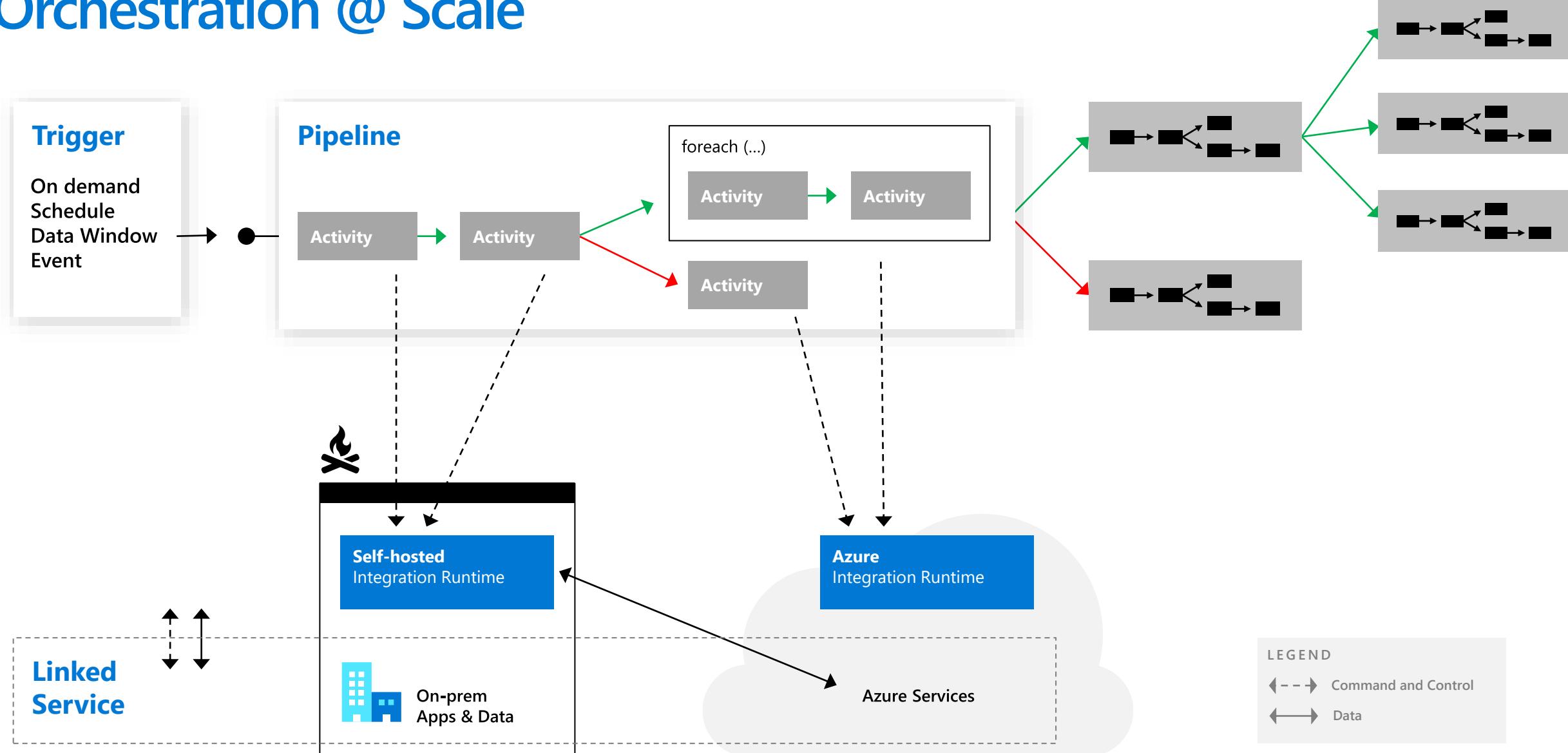
Microsoft Azure | Synapse Analytics

External connections Linked services Orchestration Triggers Integration runtimes Security Access control

+ New

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

Orchestration @ Scale



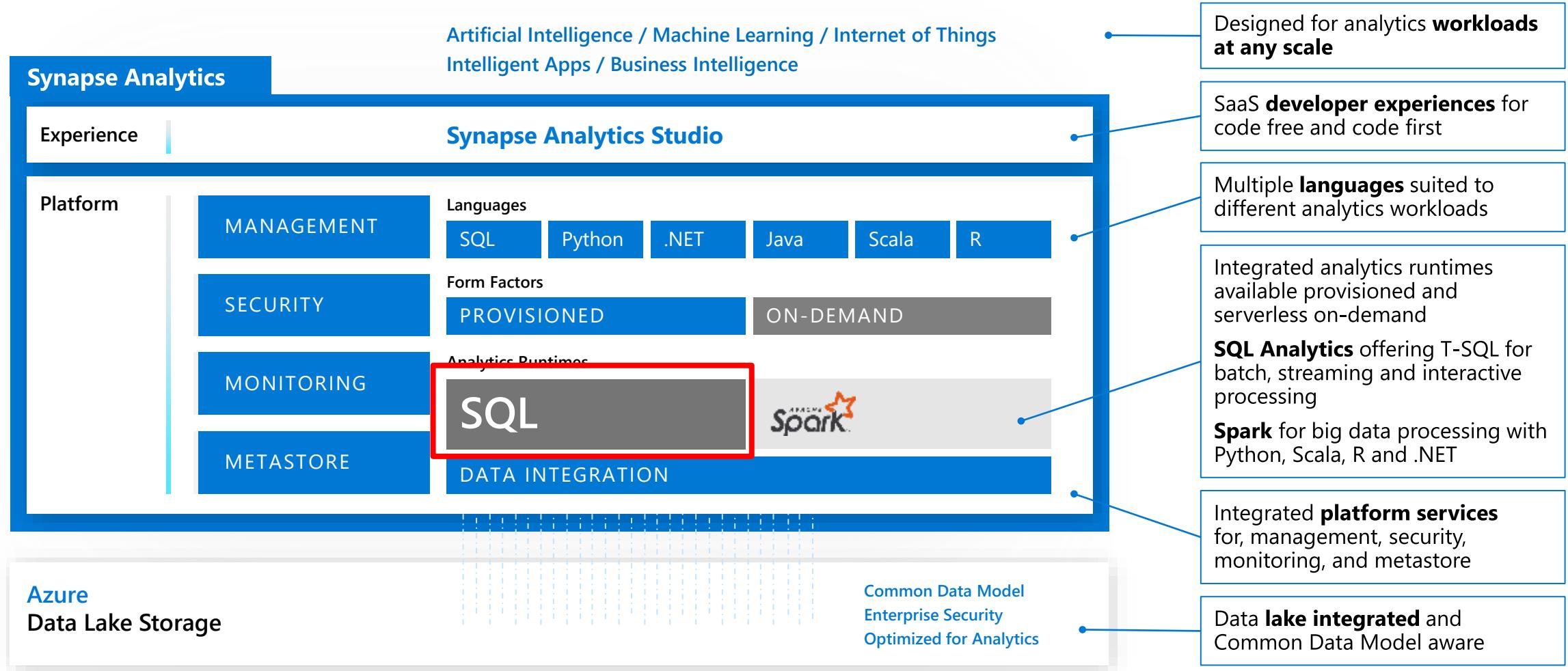


Azure Synapse Analytics

SQL Analytics *(formerly SQL Datawarehouse)*

Azure Synapse Analytics

Integrated data platform for BI, AI and continuous intelligence

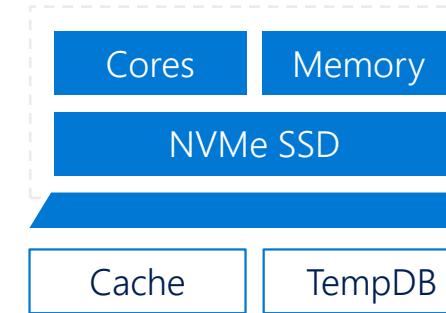
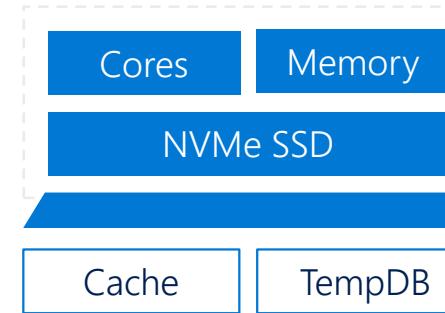
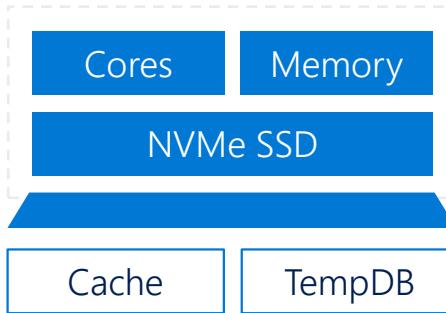


SQL DW Architecture (MPP)

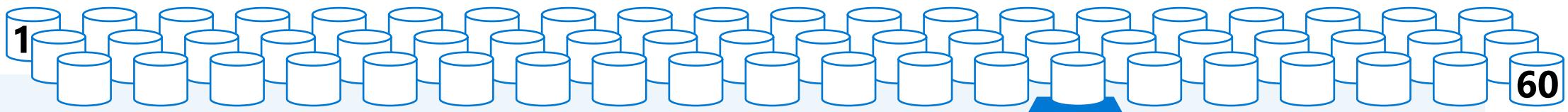
Control



Compute



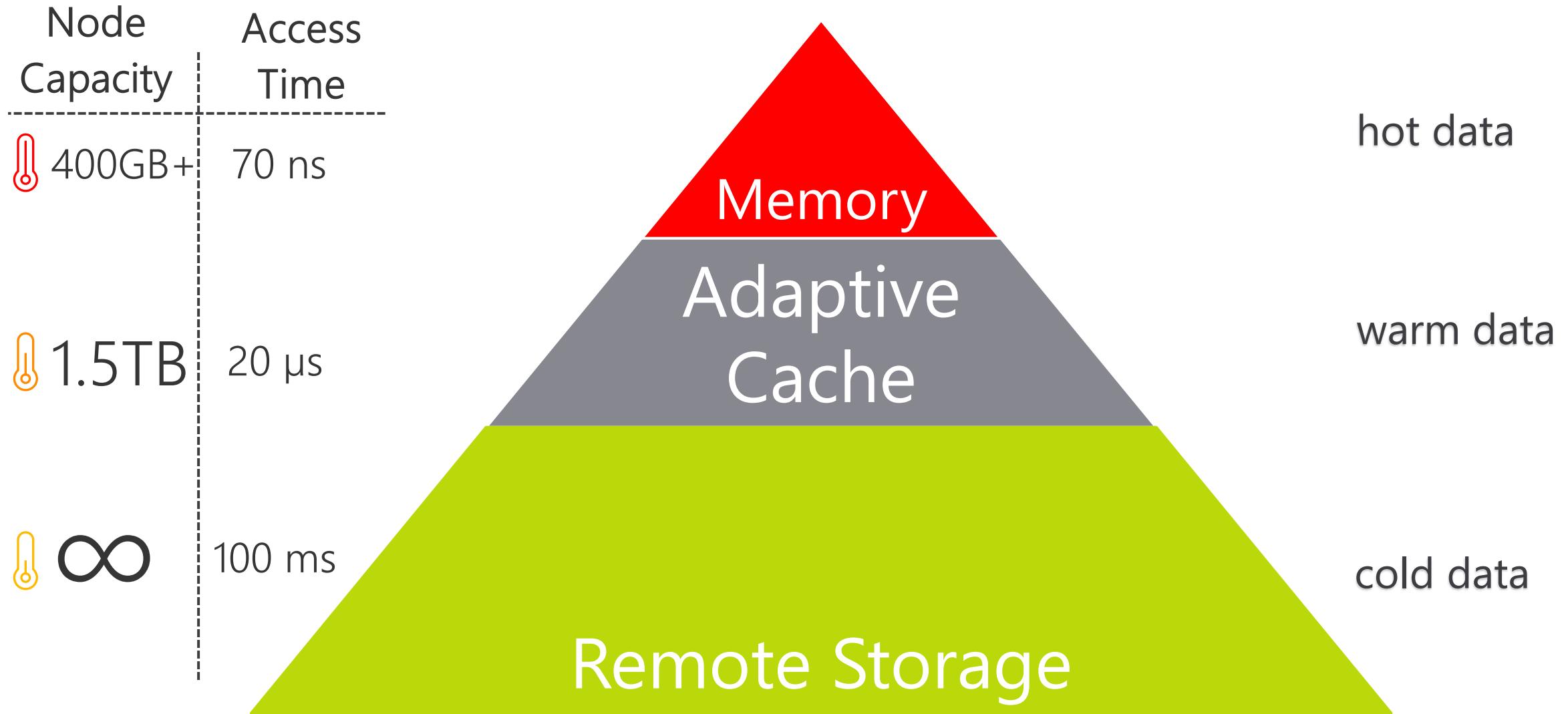
Remote storage



Snapshot backups



Automated Tiering Of Storage Layers



Data Warehouse Units (DWU)

Normalized amount of compute

Converts to billing units i.e. what you pay

CPU

RAM

I/O

| DWUc | |
|-------|--------------------|
| 100 | |
| 200 | |
| 300 | |
| 400 | |
| 500 | → 1 Compute Node |
| 1000 | → 2 Compute Nodes |
| 1500 | → 3 Compute Nodes |
| 2000 | |
| 2500 | |
| 3000 | |
| 5000 | |
| 6000 | |
| 7500 | |
| 10000 | |
| 30000 | → 60 Compute Nodes |

Reserved capacity pricing

Overview

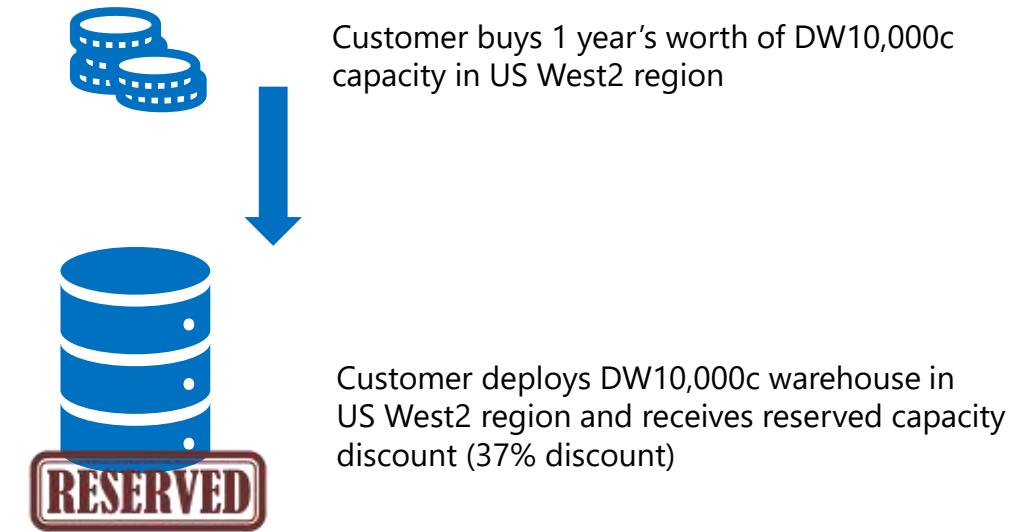
Provides customers with a significant discount (up to 69%) compared to on-demand instance pricing for signing up for an upfront monetary commitment in an Azure region.

Size flexibility: Reserved instance pricing applies to a purchased capacity amount, and is independent of instance size. So, a customer who reserves DW3000c can deploy 1x DW3000c, or 3x 1000c etc. within the same region.

Capacity returns: Paused DW's do not count towards a customer's reservation purchase. This capacity (and associated discount) can be re-used for additional data warehouses.

Pricing details: Reserved capacity discounts are based on full upfront payment and are determined as follows:

- **1 year: 37% discount**
- **3 years : 65% discount**



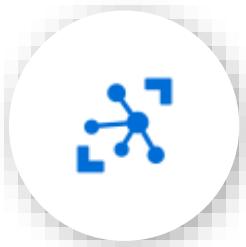
Heterogenous Data Preparation & Ingestion

Native SQL Streaming

- High throughput ingestion (up to 200MB/sec)
- Delivery latencies in seconds
- Ingestion throughput scales with compute scale
- Analytics capabilities (SQL-based queries for joins, aggregations, filters)



Event Hubs



IoT Hub

T-SQL Language

SQL Analytics



Streaming Ingestion



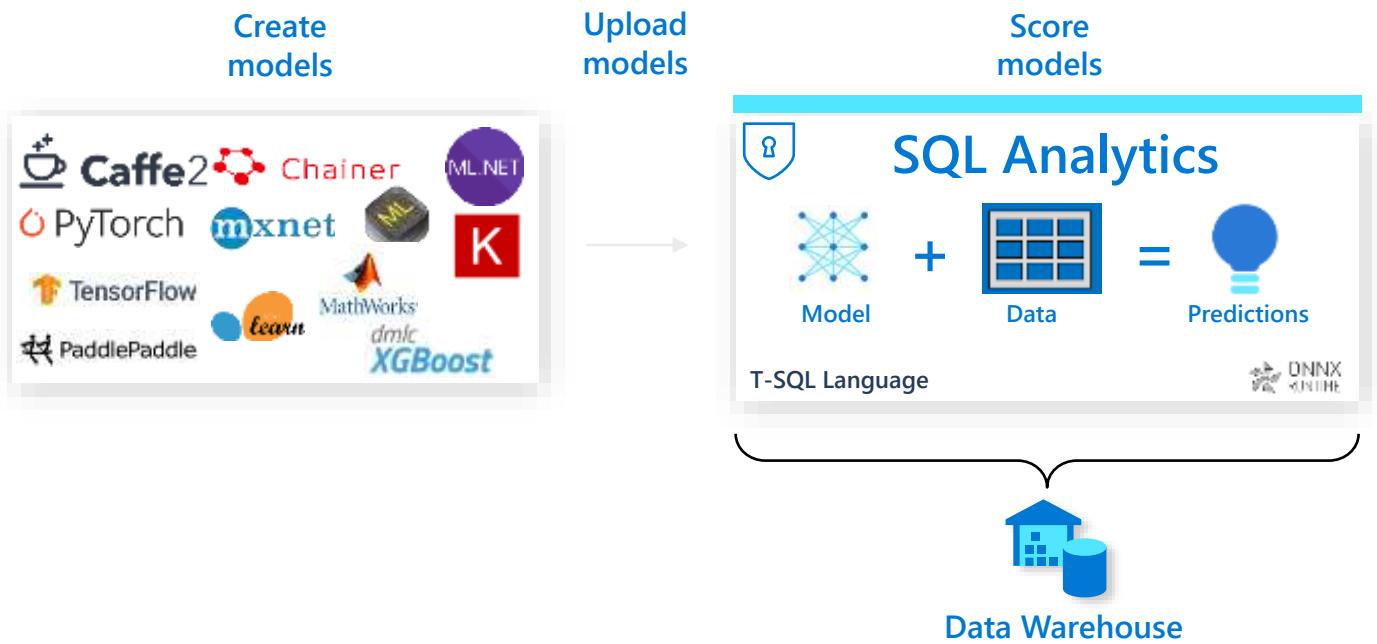
Data Warehouse

Built-in streaming ingestion & analytics

Machine Learning enabled DW

Native PREDICT-ion

- T-SQL based experience (interactive./batch scoring)
- Interoperability with other models built elsewhere
- Execute scoring where the data lives



```
--T-SQL syntax for scoring data in SQL DW
SELECT d.*, p.Score
FROM PREDICT(MODEL = @onnx_model, DATA = dbo.mytable AS d)
WITH (Score float) AS p;
```

Azure Data Share

Enterprise data sharing

- Share from DW to DW/DB/other systems
- Choose data format to receive data in (CSV, Parquet)
- One to many data sharing
- Share a single or multiple datasets

| Feature | Azure Data Share |
|---|------------------|
| Multiple Data Store Support Sharing from Azure Data Lake, Azure Storage, Azure SQL Data Warehouse, Azure SQL DB | Yes |
| Heterogenous Data Sharing Flexible sharing from/to heterogenous data stores | Yes |
| Single pane of glass Centrally managed data sharing experience | Yes |
| Governed data sharing Customer can specify terms of use | Yes |
| Snapshot based sharing Perform analytics on data for unrestricted computation & no compromise on performance | Yes |



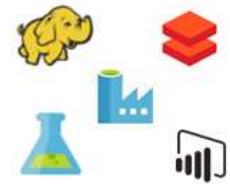
Any Azure Data Sources

Share data from any Azure regions and data stores



Single Pane of Glass

Manage and monitor data sharing with multiple organizations



Rich Analytics Tools

Use Azure analytics tools to prepare data and derive insights



Governance

Control data access governed by enterprise policies



Monetization

Charge for data or cost of data curation and access

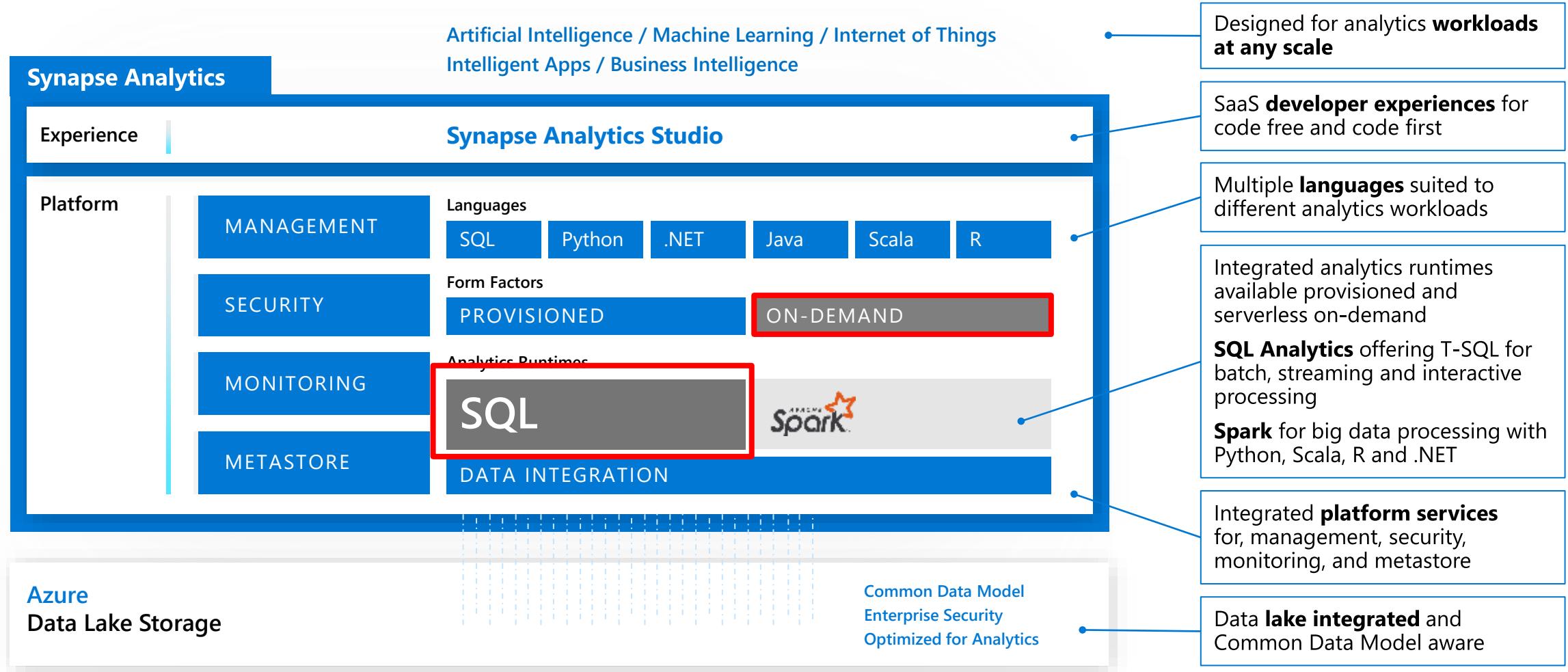


Azure Synapse Analytics

SQL On-Demand

Azure Synapse Analytics

Integrated data platform for BI, AI and continuous intelligence



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

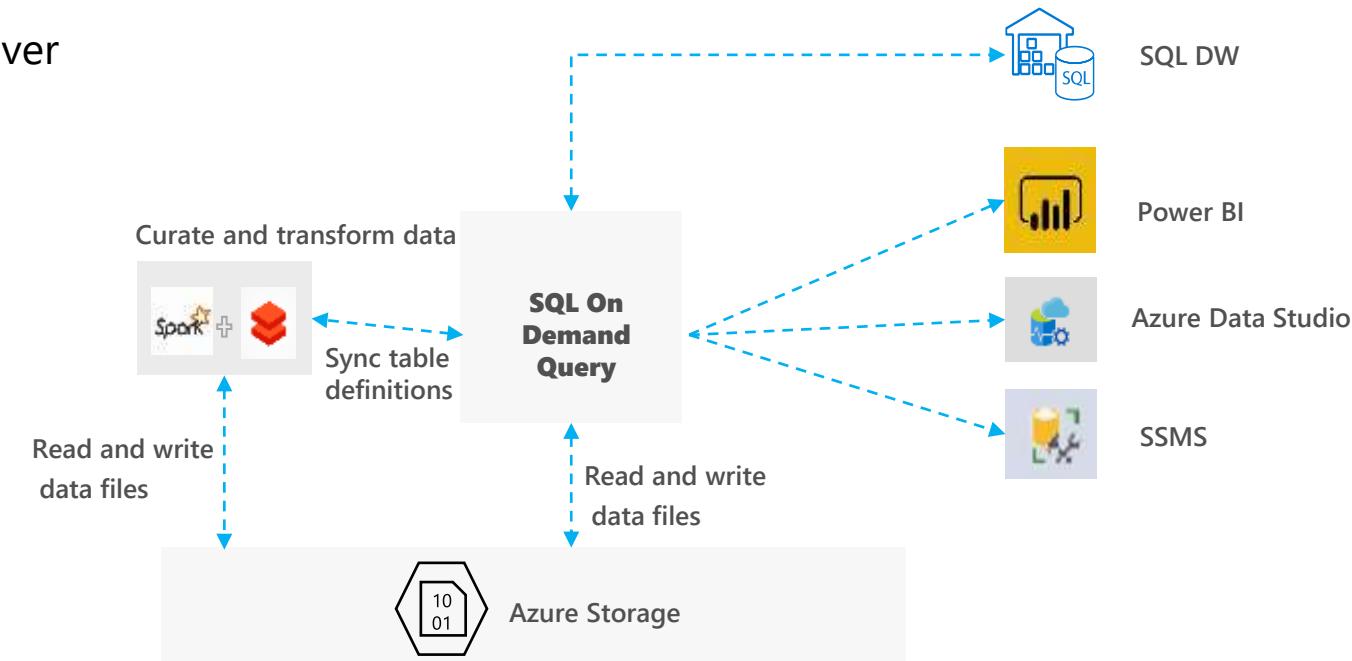
Offers security

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 shows two main windows from the Microsoft Azure portal:

- Left Window (Storage Accounts):** Shows the storage account structure for "prlangademos" (Primary). It includes:
 - Data:** Storage accounts (filesystem, holidaydatacontainer, isdweatherdatacontainer, nyctic, prlangaddemos, tmpcontainer, wwmporters).
 - Databases:** prlangadSQLDW (SQL pool), default (SQL on-demand), default (Spark).
 - Datasets:**
- Right Window (SQL Script Editor):** Displays a SQL query against a parquet file in the "nyctic" container.


```

1 SELECT
2   TOP 100 *
3   FROM
4   OPENROWSET(
5     RULE 'https://prlangademosa.blob.core.windows.net/nyctic/nyctic/part-00133-11d23892864719830543-aea5b543-5e83-4a7d-8d31-69f72c500050-15253-1.c000.snappy.parquet'
6     FORMAT='PARQUET'
7   ) AS nytic;
      
```

The results pane shows a table with columns: VENDORID, TRIPID, UPDATETIME, TRIPENDPOINTID, PASSENGERCOUNT, TRIPDISTANCE, PILOCATIONID, DOLOCATIONID, STATION, STARTLAT, and ENDLAT. The data includes several rows of trip information.

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 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... |

SQL On Demand – Querying JSON files

Example of JSON_VALUE function

```

SELECT
    JSON_VALUE(jsonContent, '$.title') AS title,
    JSON_VALUE(jsonContent, '$.publisher') AS publisher,
    jsonContent
FROM
    OPENROWSET(
        BULK 'https://XXX.blob.core.windows.net/json/books/*.json',
        FORMAT='CSV',
        FIELDTERMINATOR = '0x0b',
        FIELDQUOTE = '0x0b',
        ROWTERMINATOR = '0x0b'
    )
    WITH (
        jsonContent varchar(8000)
    ) AS [r]
WHERE
    JSON_VALUE(jsonContent, '$.title') = 'Probabilistic and Statistical Methods in Cryptology, An Introduction by Selected Topics'

```

| | title | publisher | jsonContent |
|---|---|-----------|---------------------|
| 1 | Probabilistic and Statistical Methods in Cryptology, An Introduction by Selected Topics | Springer | {"_id": "neuen...", |

Example of JSON_QUERY function

```

SELECT
    JSON_QUERY(jsonContent, '$.authors') AS authors,
    jsonContent
FROM
    OPENROWSET(
        BULK 'https://XXX.blob.core.windows.net/json/books/*.json',
        FORMAT='CSV',
        FIELDTERMINATOR = '0x0b',
        FIELDQUOTE = '0x0b',
        ROWTERMINATOR = '0x0b'
    )
    WITH (
        jsonContent varchar(8000)
    ) AS [r]
WHERE
    JSON_VALUE(jsonContent, '$.title') = 'Probabilistic and Statistical Methods in Cryptology, An Introduction by Selected Topics'

```

| | authors | jsonContent |
|---|---------------------------|---|
| 1 | ["Daniel Neuenschwander"] | {"_id": "neuenschwander04", "type": "Book", "title": "Probabi..." |



Azure Synapse Analytics Spark

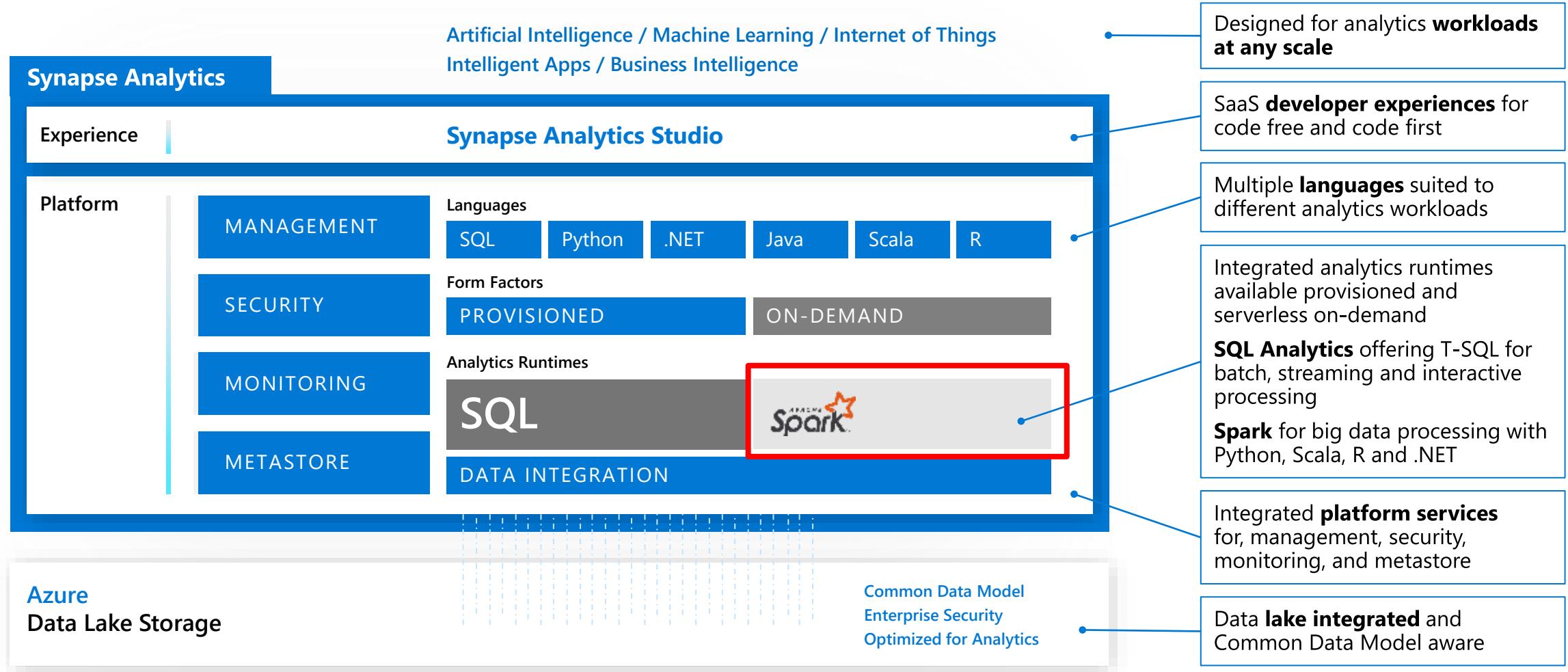
Azure Synapse Apache Spark - Summary



- **Apache Spark 2.4 derivation**
 - Linux Foundation Delta Lake 0.4 support
 - .Net Core 3.0 support
 - Python 3.6 + Anacondas support
- **Tightly coupled to other Azure Synapse services**
 - Integrated security and sign on
 - Integrated Metadata
 - Integrated and simplified provisioning
 - Integrated UX including Jupyter based notebooks
 - Fast load of SQL Analytics pools
- **Core scenarios**
 - Data Prep/Data Engineering/ETL
 - Machine Learning via Spark ML and Azure ML integration
 - Extensible through library management
- **Efficient resource utilization**
 - Fast Start
 - Auto scale (up and down)
 - Auto pause
 - Min cluster size of 3 nodes
- **Multi Language Support**
 - .Net (C#), PySpark, Scala, Spark SQL, Java

Azure Synapse Analytics

Integrated data platform for BI, AI and continuous intelligence



Languages

Overview

Supports multiple languages to develop notebook

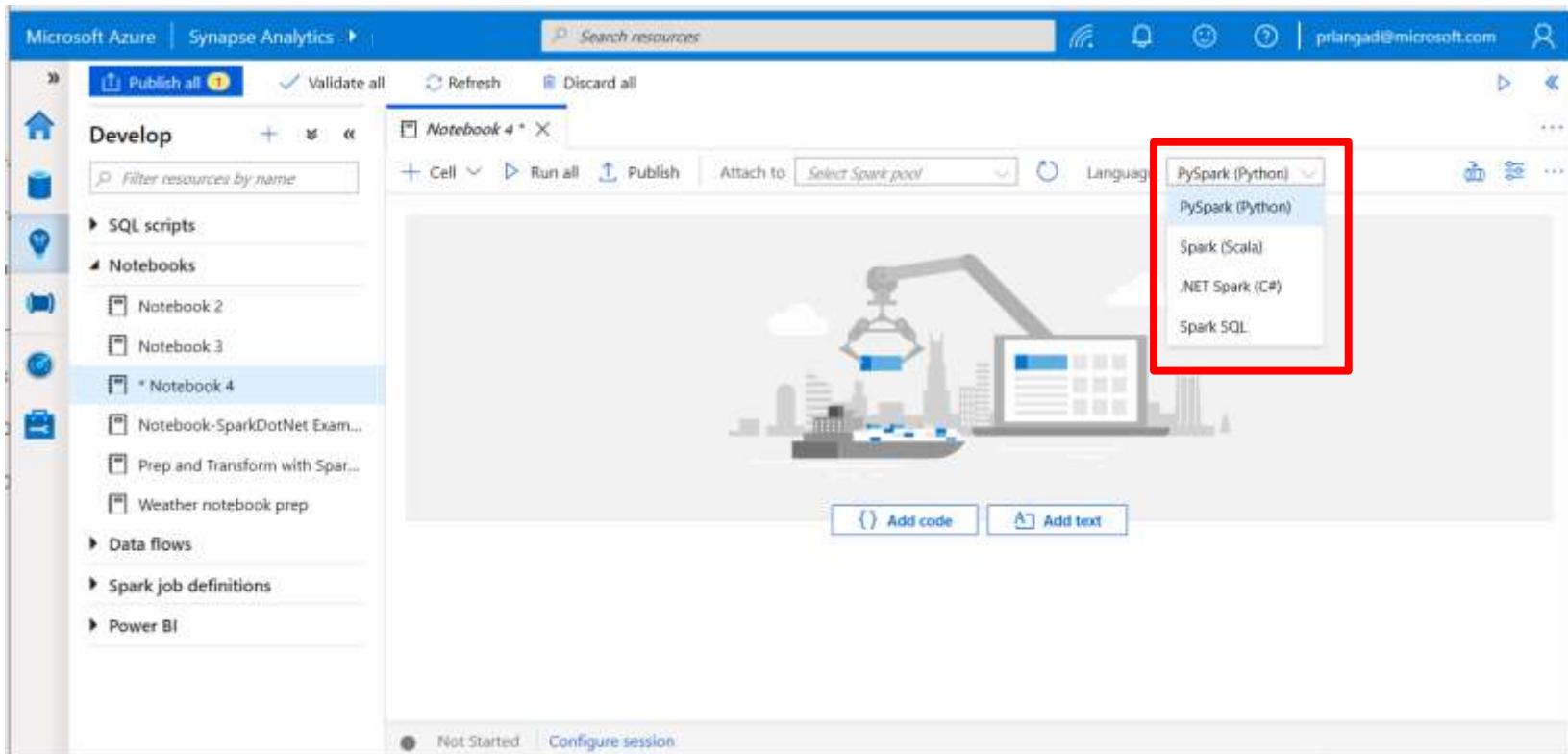
- PySpark (Python)
- Spark (Scala)
- .NET Spark (C#)
- Spark SQL

Benefits

Allows to write multiple languages in one notebook

%%<Name of language>

Offers use of temporary tables across languages



Create Notebook on files in storage

The screenshot displays two overlapping Microsoft Azure Synapse Analytics workspace windows. The top window shows a file browser interface with a sidebar for 'Storage accounts' and 'Databases'. The main area lists files in a directory structure: nyctic > yellow > puYear=2015 > puMonth=3. A context menu is open over a file named 'part-00133-tid-211'. The bottom window shows a notebook editor with a sidebar for 'Storage accounts' and 'Databases'. It contains a code cell using \$ pyspark and a table of job execution results.

File Browser (Top Window):

- Storage accounts:** prlangaddemosa (Primary) - filesystem, holidaydatacontainer, isdweatherdatacontainer, nyctic, priangaddemosa, tmpcontainer, wwimporters.
- Databases:** prlangadSQLDW (SQL pool), default (SQL on-demand), default (Spark).

File List:

| NAME | LAST MODIFIED | CONTENT TYPE | SIZE |
|--------------------|------------------------|--------------|------|
| part-00133-tid-211 | 10/25/2019, 2:20:23 PM | 324.2 MB | |

Context Menu (Over part-00133-tid-211):

- New SQL script
- New notebook (selected)
- Copy ABFS path
- Manage Access...

Notebook Editor (Bottom Window):

Code Cell:

```
$ pyspark  
data_path = spark.read.load('abfss://nyctic@prlangaddemosa.dfs.core.windows.net/yellow/puyear=2015/pumonth=3/part-00133-tid-211')  
data_path.show(10)
```

Job Execution Results:

| ID | DESCRIPTION | STATUS | STAGES | TASKS | SUBMISSION TIME | DURATION |
|-------|---|-----------|--------|-------|------------------------|----------|
| Job 0 | load at NativeMethodAccessImpl.java:0 | Succeeded | 1/1 | 1/1 | 11/14/2019, 9:56:49 AM | 7s |
| Job 1 | showString at NativeMethodAccessImpl.java:0 | Succeeded | 1/1 | 1/1 | 11/14/2019, 9:56:58 AM | 1s |
| Job 2 | showString at NativeMethodAccessImpl.java:0 | Succeeded | 1/1 | 1/1 | 11/14/2019, 9:56:59 AM | 1s |

Data Preview:

| VendorId | LocationId | RateCodeId | StoreAndStayFlag | PaymentType | FareAmount | ExtraFareAmount | Surcharge | TipAmount | TollAmount | TotalAmount | StartLat | StartLon | EndLat | EndLon | |
|---|------------|------------|------------------|-------------|-------------------|-------------------|-------------------|-------------------|------------|-------------------|-------------------|--------------------|--------------------|--------|--|
| 1 2015-03-28 23:53:18 2015-03-01 00:00:29 | 6 | 1.63 | null | null | 74.00004686279297 | 48.73869381713867 | -73.9841537475588 | 48.74470528019531 | | | | | | | |
| 1 2015-03-28 19:21:05 2015-03-28 19:28:31 | 3 | 8.5 | 0.0 | 0.5 | 2.2 | 2.2 | null | 0.0 | null | 73.9765358341797 | 48.73168705566406 | -73.95382471923828 | 48.78608311279287 | | |
| 1 2015-03-28 23:53:19 2015-03-01 00:12:08 | 3 | 14.5 | 0.5 | 0.5 | 2.23 | 2.23 | null | 0.0 | null | 73.96012879417969 | 48.76215744018955 | -73.9881881796875 | 40.728118896484375 | | |
| 1 2015-03-28 19:21:05 2015-03-28 19:37:02 | 2 | 1.63 | 0.0 | 0.5 | 4.74 | 4.74 | null | 0.0 | null | 73.98143005371004 | 48.7818855847168 | -74.00001552734175 | 40.7917723576172 | | |

Languages – PySpark (Python)

Screenshot of the Azure Synapse Analytics Python notebook interface showing a histogram generated by PySpark.

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 a notebook titled "Notebook 4 *". The cell content is:

```
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()
```

Below the code, a message indicates the command was executed in 1min 52s 837ms by prlangad on 11-02-2019 22:08:44.165 -07:00.

The output of the cell is a histogram plot showing a normal distribution of data points. The x-axis ranges from -3 to 3, and the y-axis ranges from 0 to 60. The distribution is centered at 0, with the highest frequency occurring near -0.5.

Bottom navigation bar:

- Ready
- Stop session
- Spark history server
- Configure session

Languages – Spark.NET (C#)

Microsoft Azure | Synapse Analytics | Search resources | 2

Develop | Publish all 2 | Validate all | Refresh | Discard all | Data flow 1 * | Notebook 1 *

Cell 1
[3] 1 using static Microsoft.Spark.Sql.Functions;
2 string StoragePath = "wasbs://sparkdotnet@stsuhsstorage.blob.core.windows.net/gtorrent/";
Command executed in 2s 888ms by prlangad on 10-28-2019 14:49:58.825 -07:00

Cell 2
[5] 1 "descriptor STRING, language STRING, created_at STRING, " + "forked_from INT, deleted STRING, updated_at STRING").Csv(StoragePath + "projects_sample.csv").Filter(Col("language") == "C#");
Command executed in 3s 192ms by prlangad on 10-28-2019 14:50:48.420 -07:00

Cell 3
[6] 1 DataFrame watchers = spark.Read().Option("header", "true").Schema("repo_id INT, user_id INT, created_at TIMESTAMP").Csv(StoragePath + "watchers_sample.csv");
Command executed in 3s 228ms by prlangad on 10-28-2019 14:51:12.932 -07:00

Cell 4
1 projects.Join(watchers, Col("id") == watchers["repo_id"]).GroupBy("name").Agg(Count("*").Alias("stars")).OrderBy(Desc("stars")).Show();
Command executed in 10s 983ms by prlangad on 10-28-2019 14:52:24.759 -07:00

Job execution Succeeded | Spark 2 executors 4 cores | [Spark history server](#)

| ID | DESCRIPTION | STATUS | STAGES | TASKS | SUBMISSION TIME | DURATION |
|-------|---|--------------------------|--------|---|------------------------|----------|
| Job 0 | run at ThreadPoolExecutor.java:1149 | ✓ Succeeded | 1/1 | <div style="width: 100%; background-color: #2e6b2e;"></div> | 10/28/2019, 2:52:16 PM | 3s |
| Job 1 | showString at NativeMethodAccessorImpl.java:0 | ✓ Succeeded | 2/2 | <div style="width: 100%; background-color: #2e6b2e;"></div> | 10/28/2019, 2:52:21 PM | 3s |

```
+-----+| name|stars|+-----+|corefx| 994|+-----+
```

Languages – Spark (Scala)

The screenshot shows the Azure Synapse Analytics interface for Spark Scala notebooks. On the left, a sidebar titled 'Develop' lists various resources: SQL scripts, Notebooks (Notebook 2, Notebook 3, Notebook-SparkScalaExample, Notebook-SparkDotNet Example), Prep and Transform with Spark, Weather notebook prep, Data flows, Spark job definitions, and Power BI.

The main area displays a Scala notebook with three cells:

- Cell 4:** Contains the command `holiday.createOrReplaceTempView("holidayview")`. It shows a success message: "Command executed in 3s 200ms by prialong on 11/04/2019 12:17:13.749 -08:00".
- Cell 5:** Contains two lines of Scala code: `val namesDF = spark.sql("SELECT holidayName FROM holidayview WHERE countryOrRegion = 'Canada' ")` and `namesDF.map(attributes => "holidayName: " + attributes(0)).show()`. It shows a success message: "Command executed in 4s 0ms by prialong on 11/04/2019 12:20:02.563 -08:00" and "Job execution Succeeded Spark 2 executors 4 cores". Below the code, the resulting DataFrame is displayed as a table:

| holidayName |
|-------------|
| New ... |
| Good... |
| Vict... |
| Dom... |
| Civi... |
| Labo... |
| Than... |
| Chri... |
| Boxi... |
| New... |
| Good... |
| Vict... |
| Dom... |
| Civi... |
| Labo... |
| Than... |
| Chri... |
| Boxi... |
| New ... |

Only showing top 20 rows.
- Cell 6:** Contains three lines of Scala code: `val namesRDD = namesDF.rdd`, `val outputpath = "/holiday_canada.txt"`, and `namesRDD.saveAsTextFile(adis_path + outputpath)`. It shows a success message: "Command executed in 5s 140ms by prialong on 11/04/2019 12:23:00.199 -08:00" and "Job execution Succeeded Spark 2 executors 4 cores". Below the code, the job history is shown in a table:

| ID | DESCRIPTION | STATUS | STAGES | TASKS | SUBMISSION TIME | DURATION |
|-------|--------------------------------------|-----------|--------|-------|------------------------|----------|
| Job 5 | runJob at SparkHadoopWriter.scala:78 | Succeeded | 1/1 | 1/1 | 11/4/2019, 12:22:56 PM | 1s |

Languages – Spark SQL

The screenshot shows the Azure Synapse Analytics Spark SQL interface. On the left, the 'Develop' sidebar is open, showing a list of notebooks, including 'Notebook-SparkScalaExample' which is currently selected. The main area displays a notebook cell with the following content:

```
1 %%sql
2 select * from holidayview where countryOrRegion = 'United States' and YEAR(date) = 2019
```

Below the code, it says "Command executed in 2s 91ms by prlangad on 11-04-2019 12:57:28.580 -08:00". A summary of the job execution is shown:

| ID | DESCRIPTION | STATUS | STAGES TASKS | SUBMISSION TIME | DURATION |
|--------|---------------------------------------|-----------|--------------|------------------------|----------|
| Job 12 | take at NativeMethodAccessImpl.java:0 | Succeeded | 1/1 | 11/4/2019, 12:57:27 PM | 0s |

The results of the query are displayed in a table:

| countryOrRegion | holidayName | normalizeHolidayName | isPaidTimeOff | countryRegionCode | date |
|-----------------|-----------------------------|-----------------------------|---------------|-------------------|----------------------|
| United States | New Year's Day | New Year's Day | | US | 2019-01-01T00:00:00Z |
| United States | Martin Luther King, Jr. Day | Martin Luther King, Jr. Day | | US | 2019-01-21T00:00:00Z |
| United States | Washington's Birthday | Washington's Birthday | | US | 2019-02-18T00:00:00Z |
| United States | Memorial Day | Memorial Day | | US | 2019-05-27T00:00:00Z |
| United States | Independence Day | Independence Day | | US | 2019-07-04T00:00:00Z |
| United States | Labor Day | Labor Day | | US | 2019-09-02T00:00:00Z |
| United States | Columbus Day | Columbus Day | | US | 2019-10-14T00:00:00Z |
| United States | Veterans Day | Veterans Day | | US | 2019-11-11T00:00:00Z |
| United States | Thanksgiving | Thanksgiving | | US | 2019-11-28T00:00:00Z |
| United States | Christmas Day | Christmas Day | | US | 2019-12-25T00:00:00Z |

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

The screenshot shows the 'Create Apache Spark pool' blade in the Microsoft Azure (Preview) portal. At the top, there are tabs for 'Home', 'nushuklasynapsewestus2', and 'Create Apache Spark pool'. Below the tabs, there's a heading 'Create Apache Spark pool' and a sub-instruction 'Enter required settings for this Apache Spark pool, including setting auto-pause and picking versions.' There are two radio buttons for 'Auto-pause': 'Enabled' (which is selected) and 'Disabled'. A 'Number of minutes idle' input field is set to '15'. The 'Component versions' section lists various components and their versions: Apache Spark (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), and Delta Lake (0.4.0). The 'Packages' section, which contains a 'File upload' input field with the value 'requirements.txt' and a 'Upload' button, is highlighted with a red border. At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next: Tags >'.

Library Management - Python

The screenshot shows the Microsoft Azure Synapse Analytics interface, specifically the Library Management - Python section. The left sidebar displays navigation options like Develop, SQL scripts, Notebooks, Data flows, Spark job definitions, and Power BI. The main area shows a list of notebooks: Notebook 2, * Notebook 4 (selected), Notebook-SparkDotNet Exam..., Prep and Transform with Spar..., Weather notebook prep. The top navigation bar includes a search bar, resource count (38), and user info (prlangad@microsoft.com). The central workspace 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)
```

The output of the command is a long list of package names and versions, starting with:

```
['absl-py==0.8.1',
 'adal==1.2.2',
 'alabaster==0.7.10',
 'altair==3.2.0',
 'applicationinsights==0.11.9',
 'asnincrypto==1.0.1',
 'astor==0.8.0',
 'astroid==1.4.9',
 'astropy==1.3.2',
 'attrbs==19.2.0',
 'azure-common==1.1.23',
 'azure-graphrbac==0.61.1',
 'azure-mgmt-authorization==0.60.0',
 'azure-mgmt-containerregistry==2.8.0',
 'azure-mgmt-keyvault==2.0.0',
 'azure-mgmt-resource==5.1.0',
 'azure-mgmt-storage==4.2.0',
 'azure-storage-blob==2.1.0',
 'azure-storage-common==2.1.0']
```

At the bottom, there are session controls: Ready (green checkmark), Stop session, Spark history server, and Configure session.



Azure Synapse Analytics Foundation

Manage – Access Control

Overview

It provides access control management to workspace resources and artifacts for admin and users

Benefits

Share workspace with the team

Increases productivity

Manage permissions on code artifacts and Spark pools

Add admin

An admin has full control over code artifacts, can attach to Spark pools, and can schedule pipelines. Permissions to Storage accounts and SQL pool databases are managed on the resources directly. [Learn more](#)

* Select user

Search by name or email address

Selected individual, groups or apps

No individual, groups, or apps selected.

Apply

Cancel

Spark Monitoring

Overview

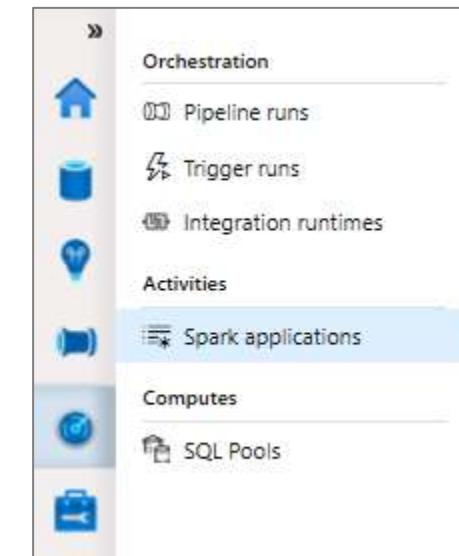
Monitor Spark pools, Spark applications for the progress and status of activities

Benefits

Monitor Spark pools for the status as paused, active, resume, scaling and upgrading

Build a dashboard to monitor performance

Track the usage of resources



| Spark applications | | | | | |
|--|-------------------------|--|--------------|---------------------|---------------|
| Submit time : 24 hours (default) (10/30/2019 9:52 AM - 10/31/2019 9:52 AM) | | Time zone : Pacific Time (US & Canada) (UT...) | | List | Chart |
| All types | Cancel | Refresh | Edit columns | | |
| APPLICATION NAME | SUBMITTER | SUBMIT TIME | STATUS | POOL | TYPE |
| Synapse_prlang-syntax... | priLangad@microsoft.com | 10/30/2019 1:21 PM | Cancelled | prlLang-syntaxcheck | Spark session |
| Synapse_prlSpark_1572... | priLangad@microsoft.com | 10/30/2019 1:06 PM | Cancelled | prlSpark | Spark session |

SQL Monitoring

Overview

Monitor SQL Pool in Azure Portal for overall usage and query activities.

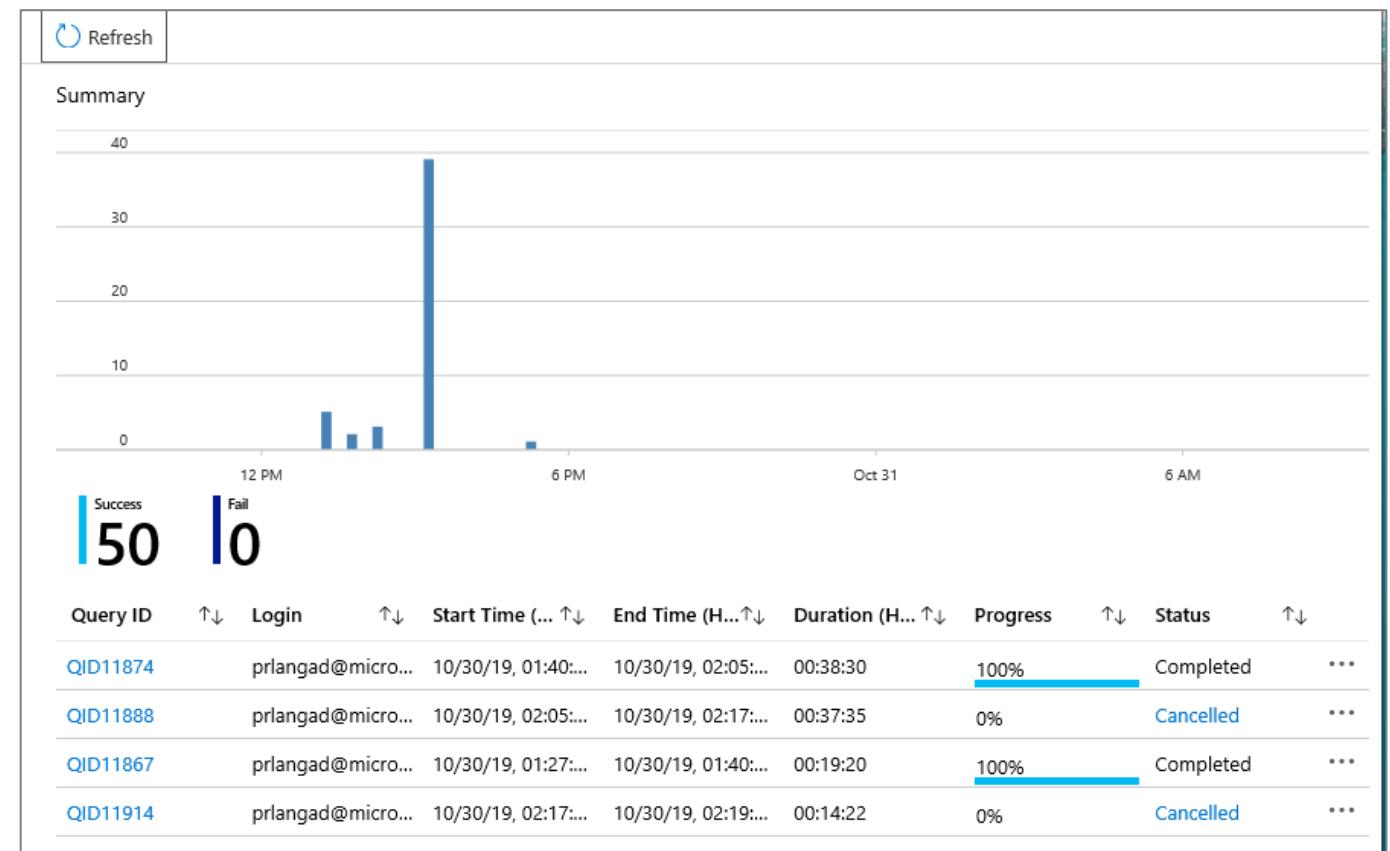
Benefits

Access SQL Audit Logs for my SQL computes

Monitor status and progress of all/specific activities

Dashboard view to monitor performance

Get to know scale of SQL compute resource

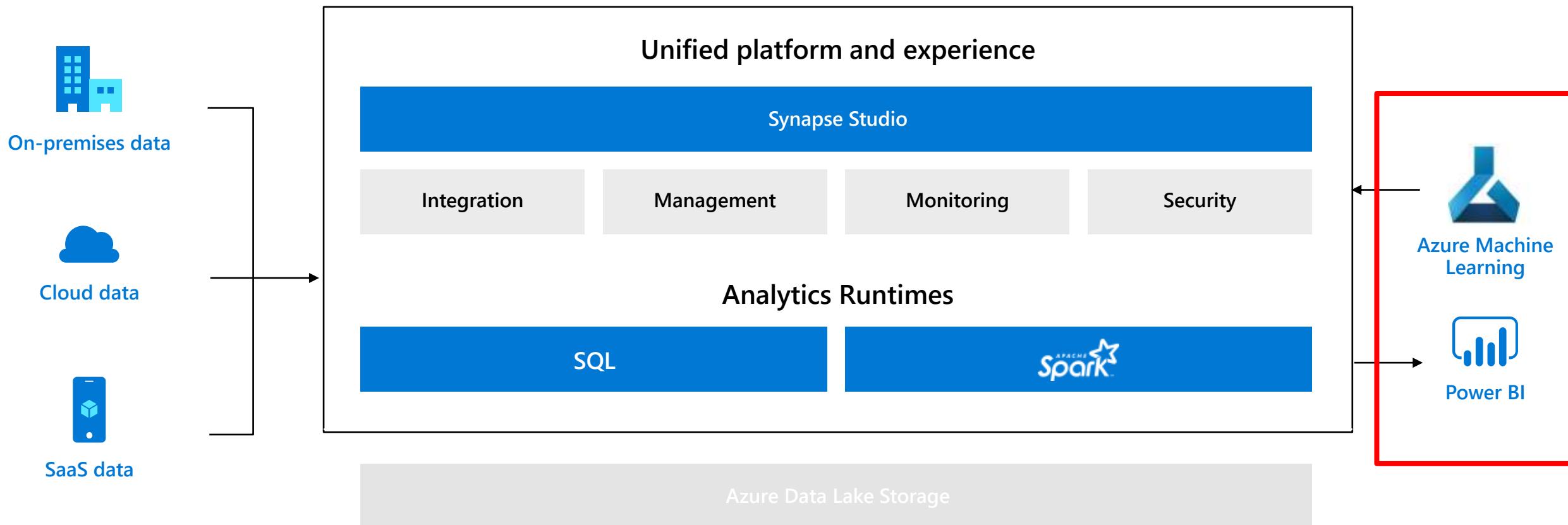




Azure Synapse Analytics Connected Services

Azure Synapse Analytics

Limitless analytics service with unmatched time to insight



Azure Machine Learning

Overview

Data Scientists can use Azure ML notebooks to do
(distributed) data preparation on Synapse Spark compute.

Benefits

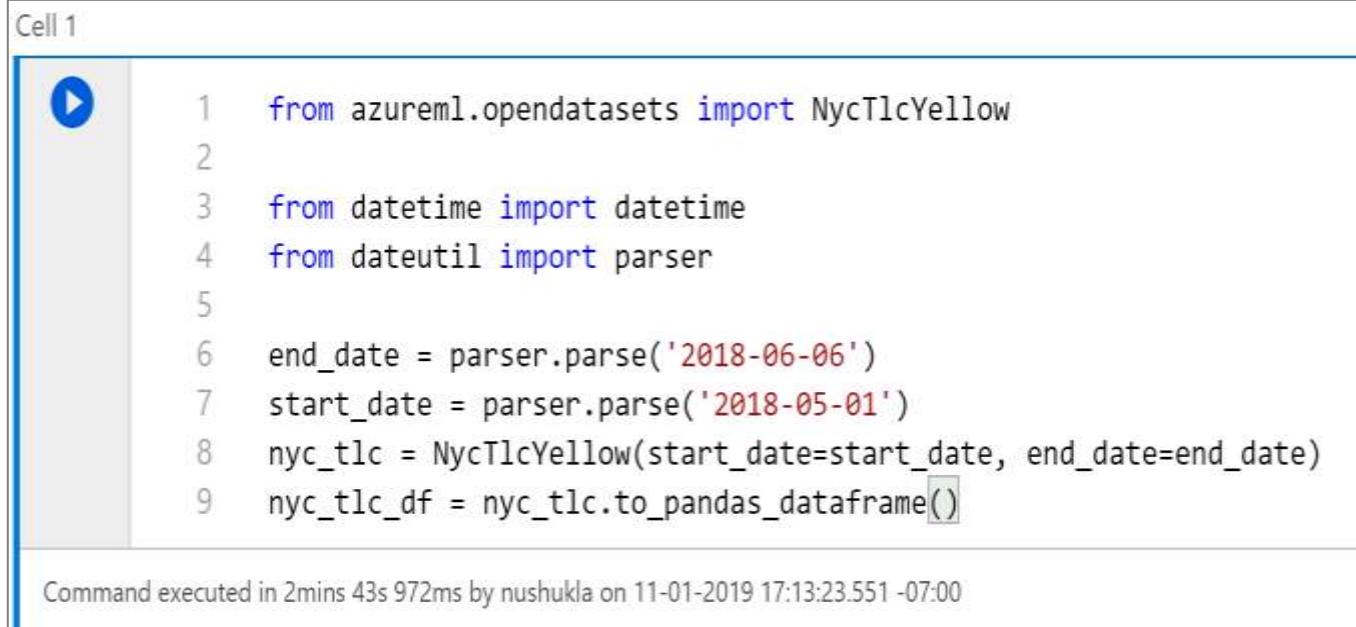
Connect to your existing Azure ML workspace and project

Use the AutoML Classifier for classification or regression
problem

Train the model

Access open datasets

Cell 1



The screenshot shows a Jupyter-style notebook cell titled "Cell 1". It contains the following Python code:

```
1 from azureml.opendatasets import NycTlcYellow
2
3 from datetime import datetime
4 from dateutil import parser
5
6 end_date = parser.parse('2018-06-06')
7 start_date = parser.parse('2018-05-01')
8 nyc_tlc = NycTlcYellow(start_date=start_date, end_date=end_date)
9 nyc_tlc_df = nyc_tlc.to_pandas_dataframe()
```

Below the code, a status message indicates: "Command executed in 2mins 43s 972ms by nushukla on 11-01-2019 17:13:23.551 -07:00".

Azure Machine Learning (continued)

Configure AutoML and Train the Models

Cell 9

```
1 l_config = AutoMLConfig(task = 'regression', debug_log = 'automl_errors.log',  
2                         primary_metric = 'normalized_root_mean_squared_error', iteration_timeout_minutes = 10,  
3                         iterations = 2, preprocess = True, n_cross_validations = 2, max_concurrent_iterations = 2,  
4                         verbosity = logging.INFO, spark_context=sc, enable_onnx_compatible_models=True, cache_store=True)
```

Cell 10

```
[ ] 1 local_run = experiment.submit(automl_config, show_output = True)
```

Best Model

Cell 12

```
[ ] 1 best_run, fitted_model = local_run.get_output(return_onnx_model=True)  
2 print(fitted_model)
```

Portal URL for Monitoring Runs

Cell 14

```
[ ] 1 more Insights of experiment  
2 displayHTML("<a href={} target='_blank'>Your experiment in Azure Portal: {}</a>".format(local_run.get_portal_url(), local_r
```



Power BI

Synapse

Power BI

Overview

Power BI is a business analytics service that delivers insights to enable fast, informed decisions

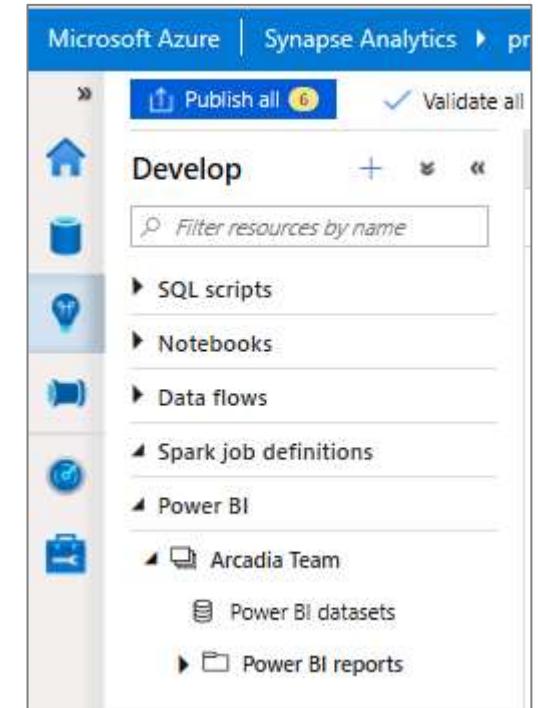
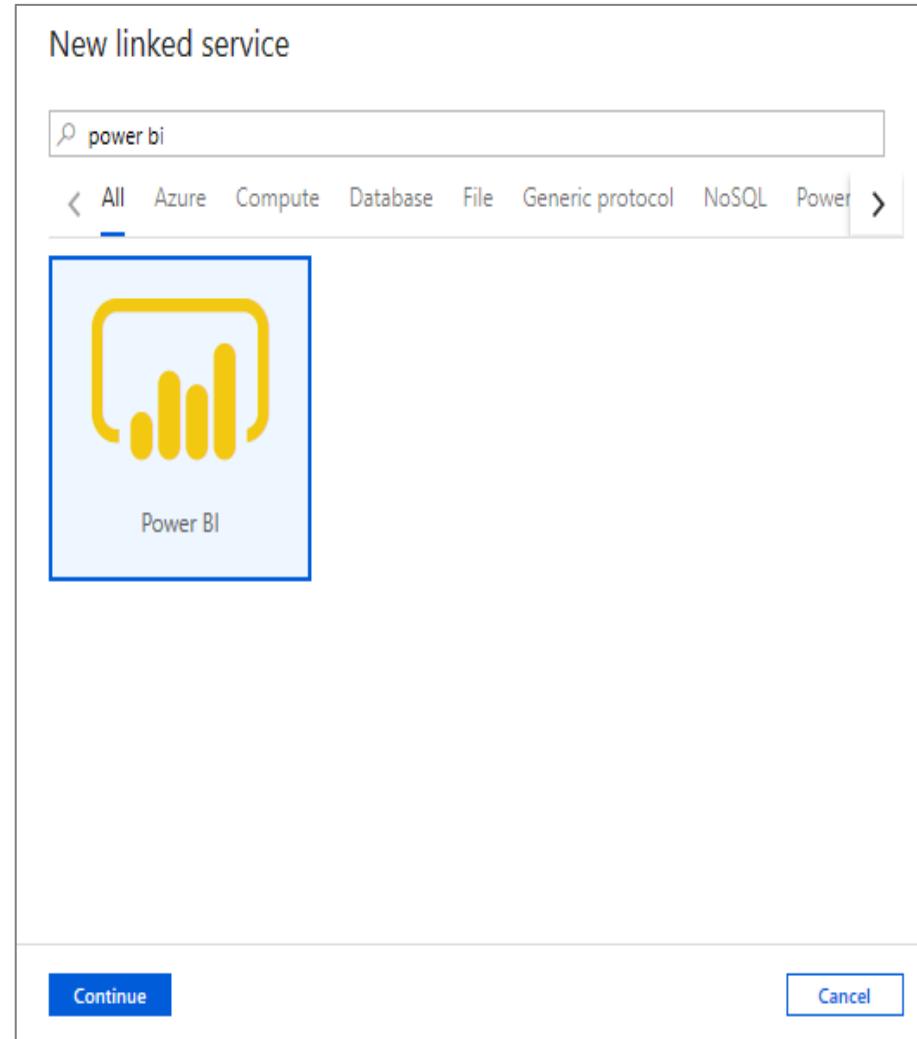
Benefits

Create Power BI reports in the workspace

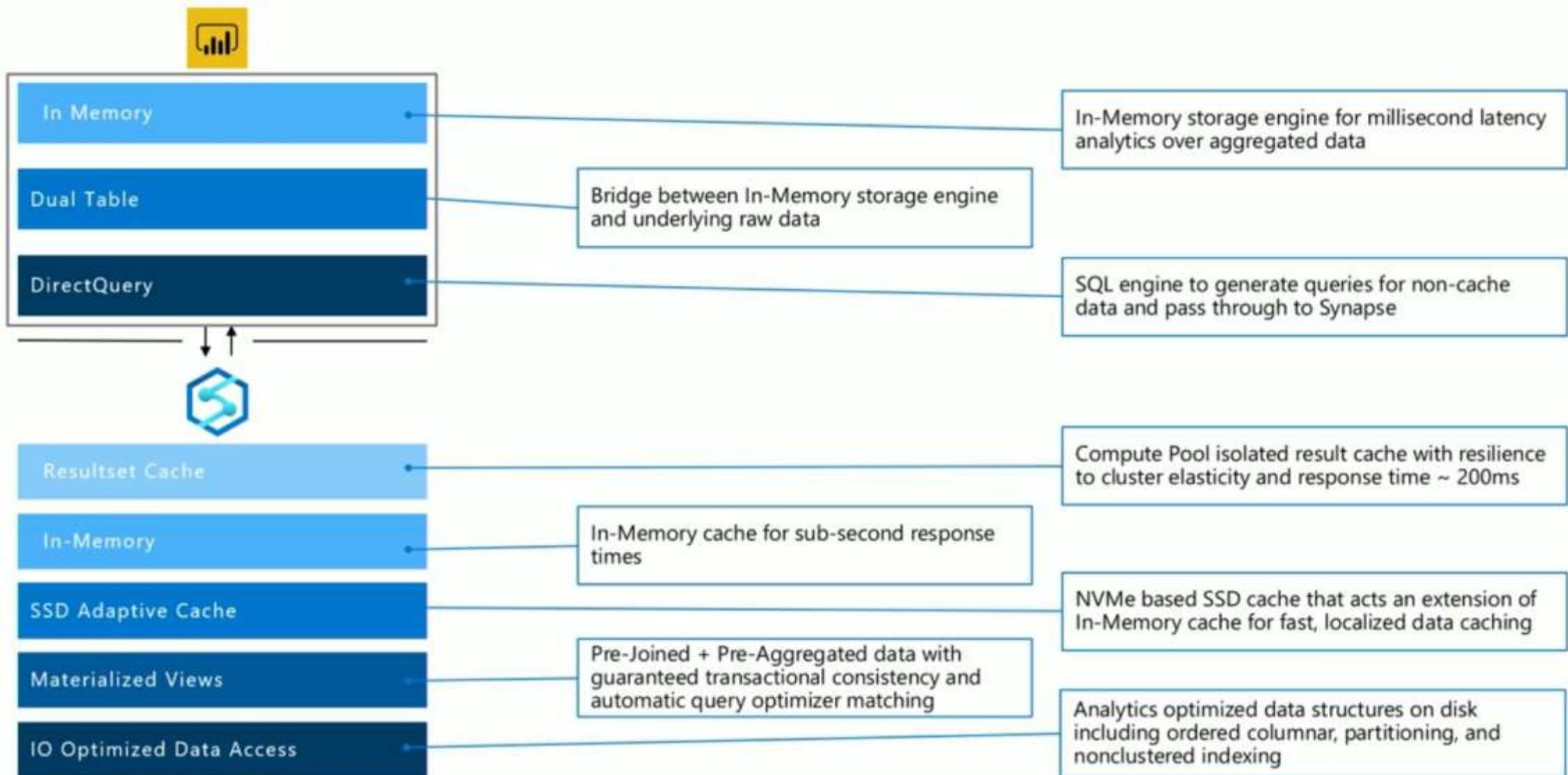
Have access to published reports in workspace

Update reports real time from Synapse workspace to get it reflected on Power BI service

Visually explore and analyze data



Power BI Aggregations and Synapse query performance



Power BI Report Overview: SynapseNYIgnite2019

The report displays a time series chart showing the relationship between GreenCab and YellowCab market share over time. The Y-axis represents Market Share (%) and the X-axis represents Date.

Key Data Points:

- GreenCab Market Share: ~15% (blue line)
- YellowCab Market Share: ~85% (yellow line)

Report Structure:

- Develop**: Current workspace view.
- yellowcabprep**: Data flow.
- PrepareTaxiData**: Data flow.
- 1 Marketshare**: Data flow.
- 2 MostTripsHoli...**: Data flow.
- AutoML**: Machine Learning model.
- SynapseNYIgnite...**: Power BI Report.

Report Details:

- Visualizations:** A collection of various chart and table icons.
- Fields:** A list of data fields:
 - dimHoliday
 - dimNYCLocations
 - Fhv
 - GreenCab
 - PredictedValues
 - vwFhvMarketShare
 - vwGnnCabMarketS...
 - vwMarketShareBy...
 - vwPredictedValues
 - vwYelCabMarketSh...
 - weather
 - YellowCab
 - YellowCabTripsHol...

Report Footer:

Page 1

Power BI Report Overview:

Report Title: yellowcabprep

Visuals:

- Line chart showing "Trips, Dispatches and Returns by Date/Price". The chart displays three data series over time: "Fhv" (blue), "YellowCab" (yellow), and "GreenCab" (green). The Y-axis represents the count of trips, dispatches, or returns.
- Bar chart showing "Trips by Holiday Name". The chart displays the number of trips for various holidays. The top categories include "New Year's Day", "Martin Luther King Jr. Day", "Memorial Day", "Independence Day", and "Labor Day".

Filters:

- Filters on this visual:**
 - holidayName: is (All)
 - numTrips: is (All)
- Filters on this page:** Add data fields here
- Filters on all pages:** Add data fields here

Visualizations:

- Line chart: Fhv, YellowCab, GreenCab
- Bar chart: Trips by Holiday Name

Fields:

- Axis:** holidayName
- Legend:** Add data fields here
- Value:** numTrips
- Tooltips:** Add data fields here
- DRILLTHROUGH:** numTrips
- Cross-report:** Add data fields here

Available Fields:

- dimHoliday
- dimNYCLocations
- Fhv
- GreenCab
- PredictedValues
- vwFhvMarketShare
- vwGrnCabMarketS...
- vwMarketShareBy...
- vwPredictedValues
- vwYelCabMarketSh...
- weather
- YellowCab
- YellowCabTripsHoli...
- date
- holidayName
- numTrips
- year

Report Navigation:

- Page 1
- Page 2

Power BI Report Overview:

Report Title: yellowcabprep

Visuals:

- Line chart showing "Total Daily Trips" over time (Date).
- Bar chart showing "Trips by holidayName" for various holidays.

Filters:

- Filters on this visual:**
 - holidayName: is (All)
 - numTrips: is (All)
- Filters on this page:** Add data fields here
- Filters on all pages:** Add data fields here

Visualizations:

- Line chart: Total Daily Trips
- Bar chart: Trips by holidayName
- Pyramid chart: Py
- Calculated table: Axis
- Calculated table: Legend
- Calculated table: Value
- Calculated table: numTrips
- Calculated table: Tooltips
- Calculated table: DRILLTHROUGH
- Calculated table: Cross-report

Fields:

- dimHoliday
- dimNYCLocations
- Fhv
- GreenCab
- PredictedValues
- vwFhvMarketShare
- vwGnCabMarketS...
- vwMarketShareBy...
- vwPredictedValues
- vwYelCabMarketSh...
- weather
- YellowCab
- YellowCabTripsHoli...
- date
- holidayName
- numTrips
- year

Report Structure:

- Develop
- SQL Scripts: YellowCabExploration_sqld
- Notebooks:
 - AMLAutoMLPredict
 - AutoML
 - Data Download_Weather
 - * PrepareTaxiData
 - yellowcabprep
 - YellowCabPrepare
- Data flows:
 - PrepareCabDataFlow
- Spark job definitions
- Power BI:
 - SynapseNYTaxiInsights
 - Power BI Datasets
 - Power BI Reports:
 - SynapseNYignite2019
 - SynapseNYignite2019 (1)

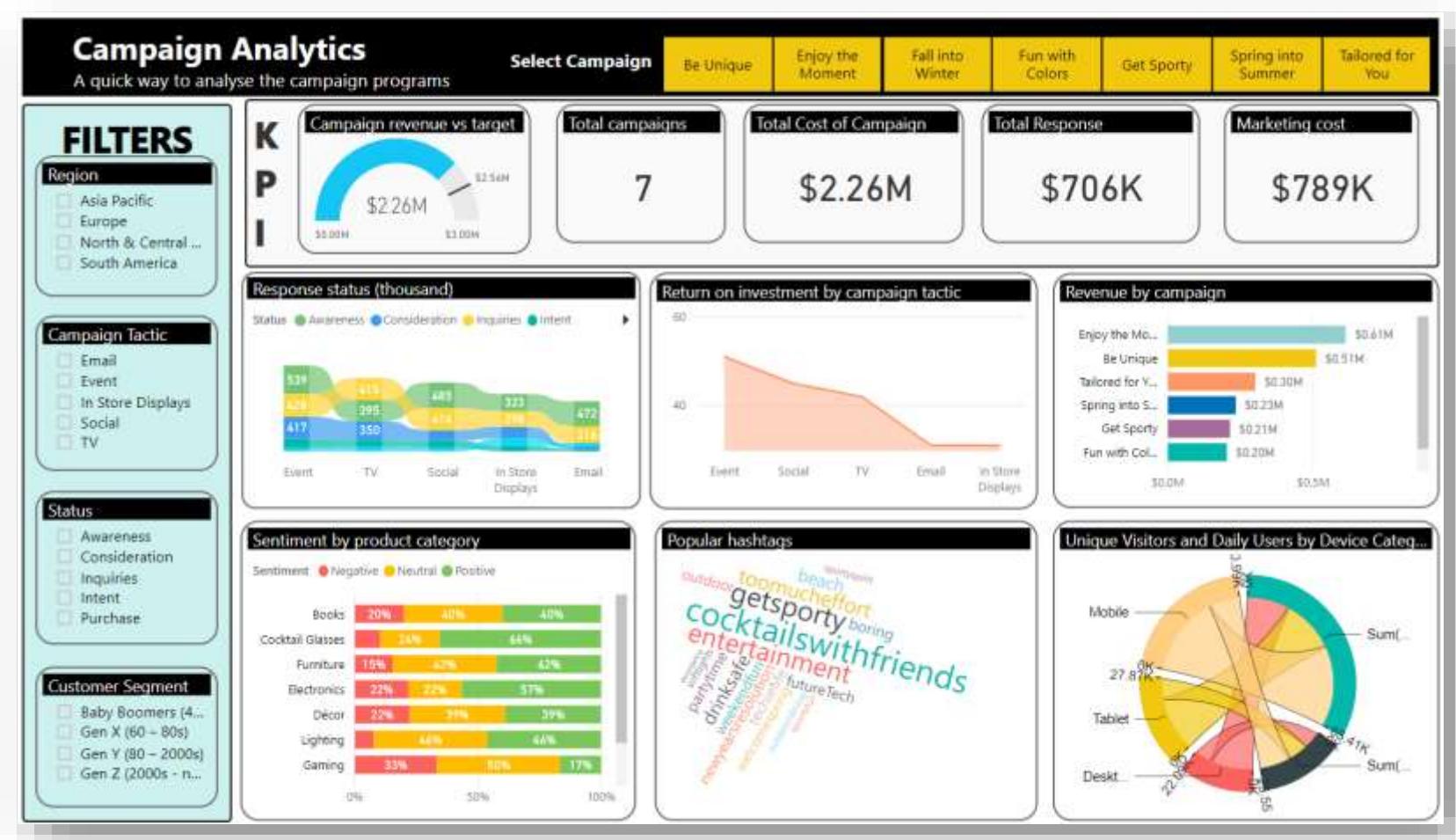
Actions:

- Publish all (2)
- Validate all
- Refresh
- Discard all

Amazing insights made possible by migration to Azure Synapse Analytics



Campaign analytics insights based on historical data



Top documentation links

- What is SQL on-demand?: [link](#)
- What is Apache Spark in Azure Synapse Analytics?: [link](#)
- Best practices for SQL pool in Azure Synapse Analytics: [link](#)
- Best practices for SQL on-demand in Azure Synapse Analytics: [link](#)
- Azure Synapse Analytics shared metadata: [link](#)
- Use maintenance schedules to manage service updates and maintenance: [link](#)
- Cheat sheet for Azure Synapse Analytics (formerly SQL DW): [link](#)
- Best practices for SQL Analytics in Azure Synapse Analytics (formerly SQL DW): [link](#)
- Synapse Analytics documentation is here: aka.ms/SynapseDocs

Q & A

Adrián J. Fernández Zenteno

Sr. Cloud Solution Architect - Azure Advanced Analytics

Email: adrian.fernandez@microsoft.com

Twitter: @AdrianFZ10

