



GPS Data/AI Strategy FY23

Delivered by CSA Team



Franck Gaillard
Cloud Solution Architect
Data AI
frgail@microsoft.com



Narjes Majdoub
Cloud Solution Architect
Data AI
nmajdoub@microsoft.com



Ali Bouhaddou
Cloud Solution Architect
Data Analytics
albouhad@microsoft.com



Frederic Gisbert
Cloud Solution Architect
Data Analytics
frgisber@microsoft.com

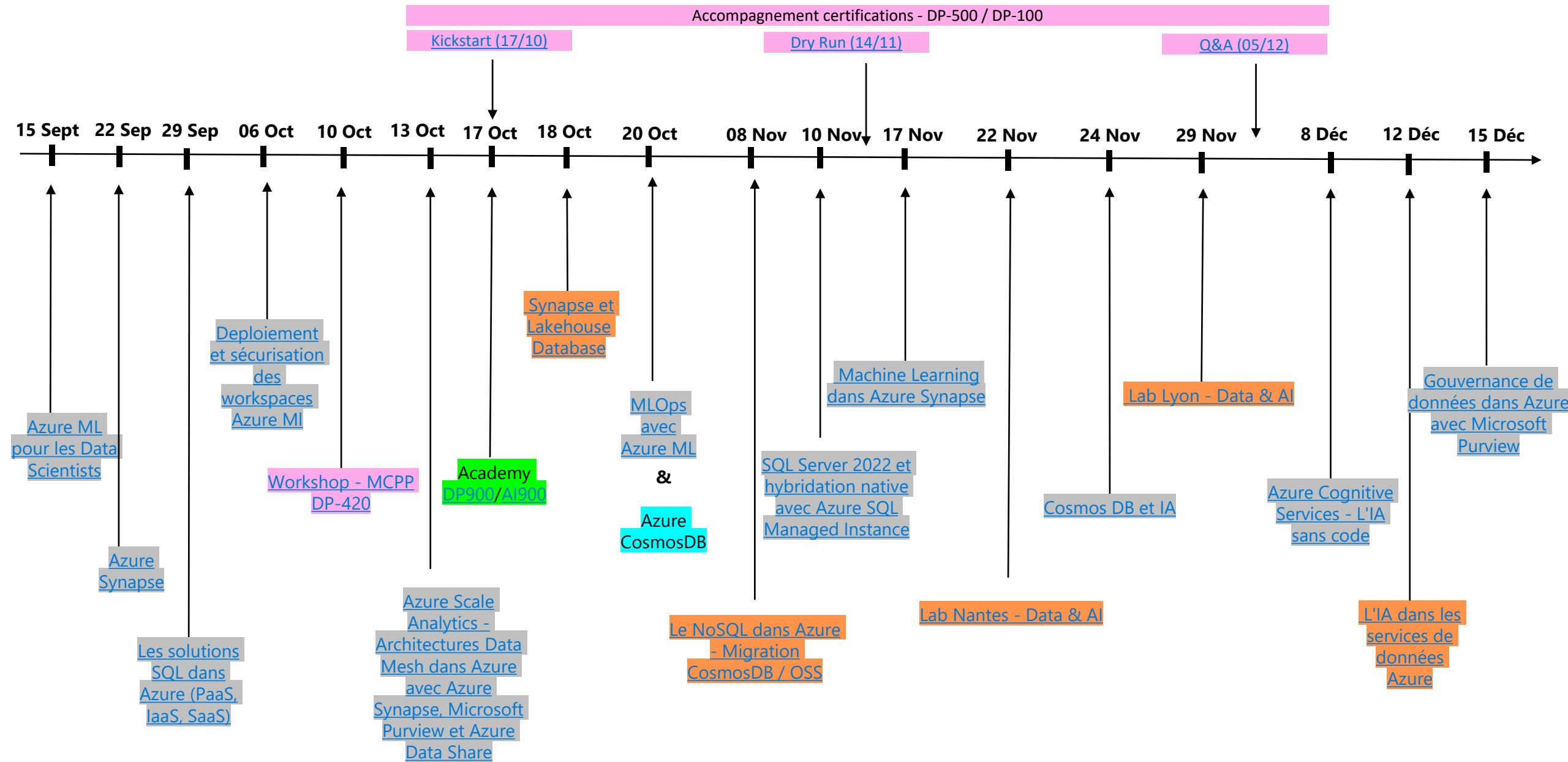


Azure Data & AI technical intensity plan

- From June 2022 to June 2023
- Focus on "Azure Data & AI" tech intensity
- Many content, from L100 Beginner to L400 Expert level:
 - Academy L100
 - Webinar L200/L300
 - Workshop L300/L400
 - Certification kickstart L300/L400
 - Openhack / Microhack L400

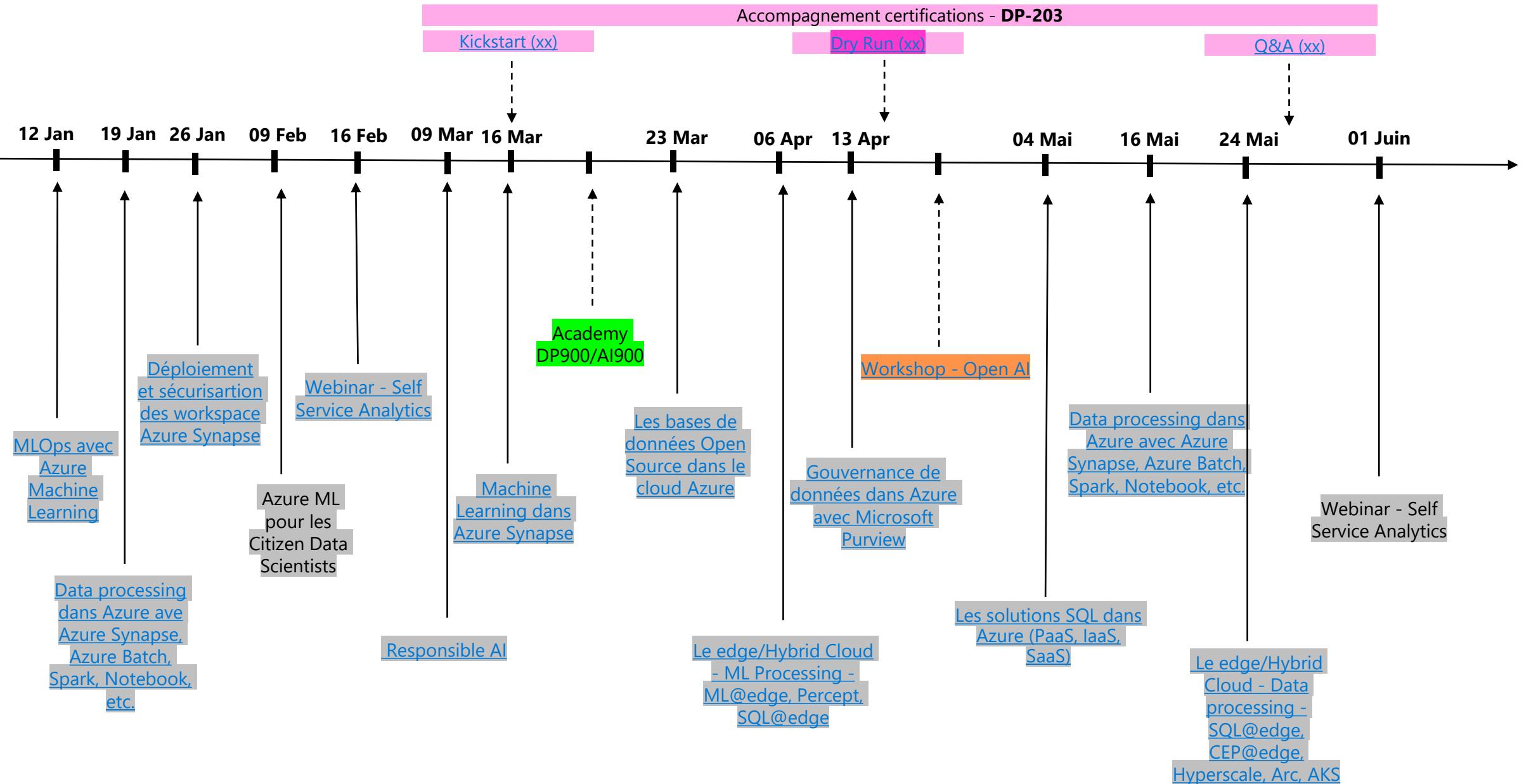
Data & AI events timeline – H1

Webinar/Academy - L 200/300
Workshop/ Openhack/ Certifications - L 300/400



Data & AI events timeline – H2

Webinar/Academy - L 200/300
Workshop/ Openhack/ Certifications - L 300/400



Liste des évènements de type Webinar 2H

Event Webinar (Les jeudis de la Data & AI) - L200/300	Date	Duration (min)	Link
Azure Machine Learning pour les Data Scientists	15/09/2022	120	https://msevents.microsoft.com/event?id=2454281594
Azure Synapse	22/09/2022	120	https://msevents.microsoft.com/event?id=857781749
Les solutions SQL dans Azure (PaaS, IaaS, SaaS)	29/09/2022	120	https://msevents.microsoft.com/event?id=502366997
Déploiement et sécurisation des workspaces Azure Machine learning	06/10/2022	120	https://msevents.microsoft.com/event?id=1505714138
Azure Scale Analytics - Architectures Data Mesh dans Azure avec Azure Synapse, Microsoft Purview et Azure Data Share	13/10/2022	120	https://msevents.microsoft.com/event?id=139685175
MLOps avec Azure Machine Learning	20/10/2022	120	https://msevents.microsoft.com/event?id=1245885767
SQL Server 2022 et hybridation native avec Azure SQL Managed Instance	10/11/2022	120	https://msevents.microsoft.com/event?id=145826476
Machine Learning dans Azure Synapse Analytics	17/11/2022	120	https://msevents.microsoft.com/event?id=3637723312
Azure Cosmos DB et IA	24/11/2022	120	https://msevents.microsoft.com/event?id=2646013445
Azure et les Services Cognitifs	08/12/2022	120	https://msevents.microsoft.com/event?id=3772037220
La gouvernance de données dans Azure avec Microsoft Purview	15/12/2022	120	https://msevents.microsoft.com/event?id=1499560981
MLOps avec Azure Machine Learning	12/01/2023	120	https://msevents.microsoft.com/event?id=4115194515
Data processing dans Azure ave Azure Synapse, Azure Batch, Spark, Notebook, etc.	19/01/2023	120	https://msevents.microsoft.com/event?id=1537241181
Déploiement et sécurisation des workspace Azure Synapse	26/01/2023	120	https://msevents.microsoft.com/event?id=1806467748
Azure Machine Learning pour les Citizen Data Scientists	09/02/2023	120	En cours
PowerBI - Self Service Analytics	16/02/2023	120	https://msevents.microsoft.com/event?id=1401519679
L'IA responsable avec Azure machine learning	09/03/2023	120	https://msevents.microsoft.com/event?id=2072953112
Machine Learning dans Azure Synapse Analytics	16/03/2023	120	https://msevents.microsoft.com/event?id=3413014857
Les bases de données Open Source dans le cloud Azure	23/03/2023	120	https://msevents.microsoft.com/event?id=2727487131
Hybridation des services de Machine Learning Azure	06/04/2023	120	https://msevents.microsoft.com/event?id=1624914222
La gouvernance de données dans Azure avec Microsoft Purview	13/04/2023	120	https://msevents.microsoft.com/event?id=3909342839
Les solutions SQL dans Azure (PaaS, IaaS, SaaS)	04/05/2023	120	https://msevents.microsoft.com/event?id=1162207895
Data processing dans Azure ave Azure Synapse, Azure Batch, Spark, Notebook, etc.	16/05/2023	120	https://msevents.microsoft.com/event?id=3517068442
Hybridation des services de données Azure	24/05/2023	120	https://msevents.microsoft.com/event?id=2996507398
Self Service Analytics	01/06/2023	120	En cours

Liste des évènements de type Workshop/Prepa Cert/Academy

Event Workshop L300/400	Date	Duration (min)	Link
Synapse et Lakehouse Database	18/10/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdURE1RMVgwTDNISTE1TDFYSDVLR0cyS1kwWS4u
Le NoSQL dans Azure - Migration CosmosDB / OSS	08/11/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdURE1RMVgwTDNISTE1TDFYSDVLR0cyS1kwWS4u
Lab Lyon - Data & AI	22/11/2022	240	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdUMIZZOURET0RSWjcyTERYRkJGTIFFUJaUi4u
Lab Nantes - Data & AI	29/11/2022	240	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdUMIZZOURET0RSWjcyTERYRkJGTIFFUJaUi4u
L'IA dans les services de données Azure	12/12/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdURE1RMVgwTDNISTE1TDFYSDVLR0cyS1kwWS4u
Open AI	H2	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdURE1RMVgwTDNISTE1TDFYSDVLR0cyS1kwWS4u

Event Academy, kickstart certifications, workshop certifications	Date	Duration (min)	Link
MCPP - DP-420	10/10/2022	420	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdUMkJSIRKSU1RRFA0OVgzSFdTSTY0RE9WQy4u
Micro Hack CosmosDB	20/10/2022	420	H1 - Inscriptions PTA
Academy DP900	17-21/10/2022	300	https://msevents.microsoft.com/event?id=3250818161
Academy AI900	17-21/10/2022	300	https://msevents.microsoft.com/event?id=2717528090
Kickstart DP-500	17/10/2022	60	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdUNEk3WFQ1TEdNNTQ2Uk85V0cxQzM3TE9ZRS4u
Dry Run DP-500	14/11/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdUNEk3WFQ1TEdNNTQ2Uk85V0cxQzM3TE9ZRS4u
Q&A DP-500	05/12/2022	90	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdUNEk3WFQ1TEdNNTQ2Uk85V0cxQzM3TE9ZRS4u
Kickstart DP-100	17/10/2022	60	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdUNDAxV0hSN0FHM1YzUzI3OUNMFYxSkRIMi4u
Dry Run DP-100	14/11/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdUNDAxV0hSN0FHM1YzUzI3OUNMFYxSkRIMi4u
Q&A DP-100	05/12/2022	90	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdUNDAxV0hSN0FHM1YzUzI3OUNMFYxSkRIMi4u
Kickstart DP-203	17/10/2022	60	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdUOVFWOUVCNFcyQk5SVjFBUFczNktCUFpLMi4u
Dry Run DP-203	14/11/2022	120	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdUOVFWOUVCNFcyQk5SVjFBUFczNktCUFpLMi4u
Q&A DP-203	05/12/2022	90	https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHB3zwJTO3s11AuaqpNnBbrwdUOVFWOUVCNFcyQk5SVjFBUFczNktCUFpLMi4u



Azure Synapse Analytics & Lake Database

18/10/2022

Speaker info



Frederic Gisbert

Cloud Solutions Architect
Frederic.gisbert@microsoft.com



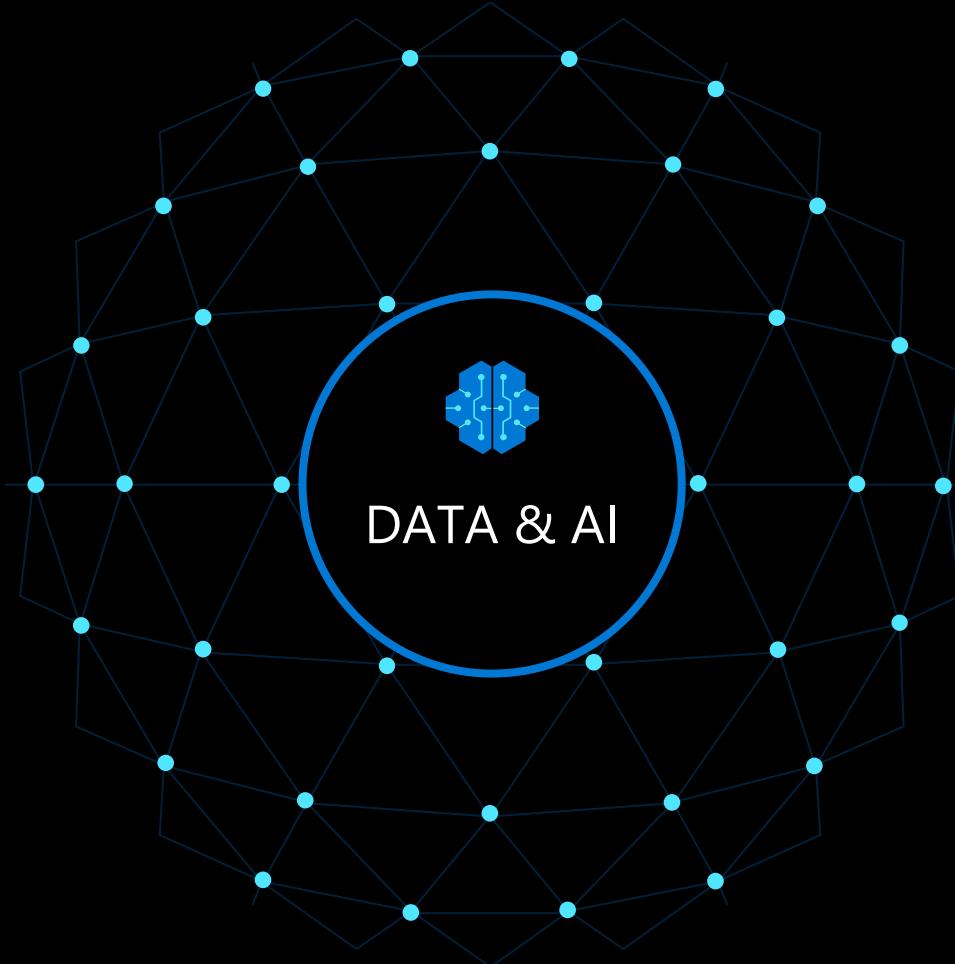
Ali Bouhaddou

Cloud Solutions Architect
albouhad@microsoft.com

The Data Driven Enterprise

Engage customers 

Transform products 



 Optimize operations

 Empower people

Approche serverless de données

Besoin d'expliquer finement l'approche analytique moderne

Besoin d'expliquer les nouvelles avancées des services de données dans le cloud

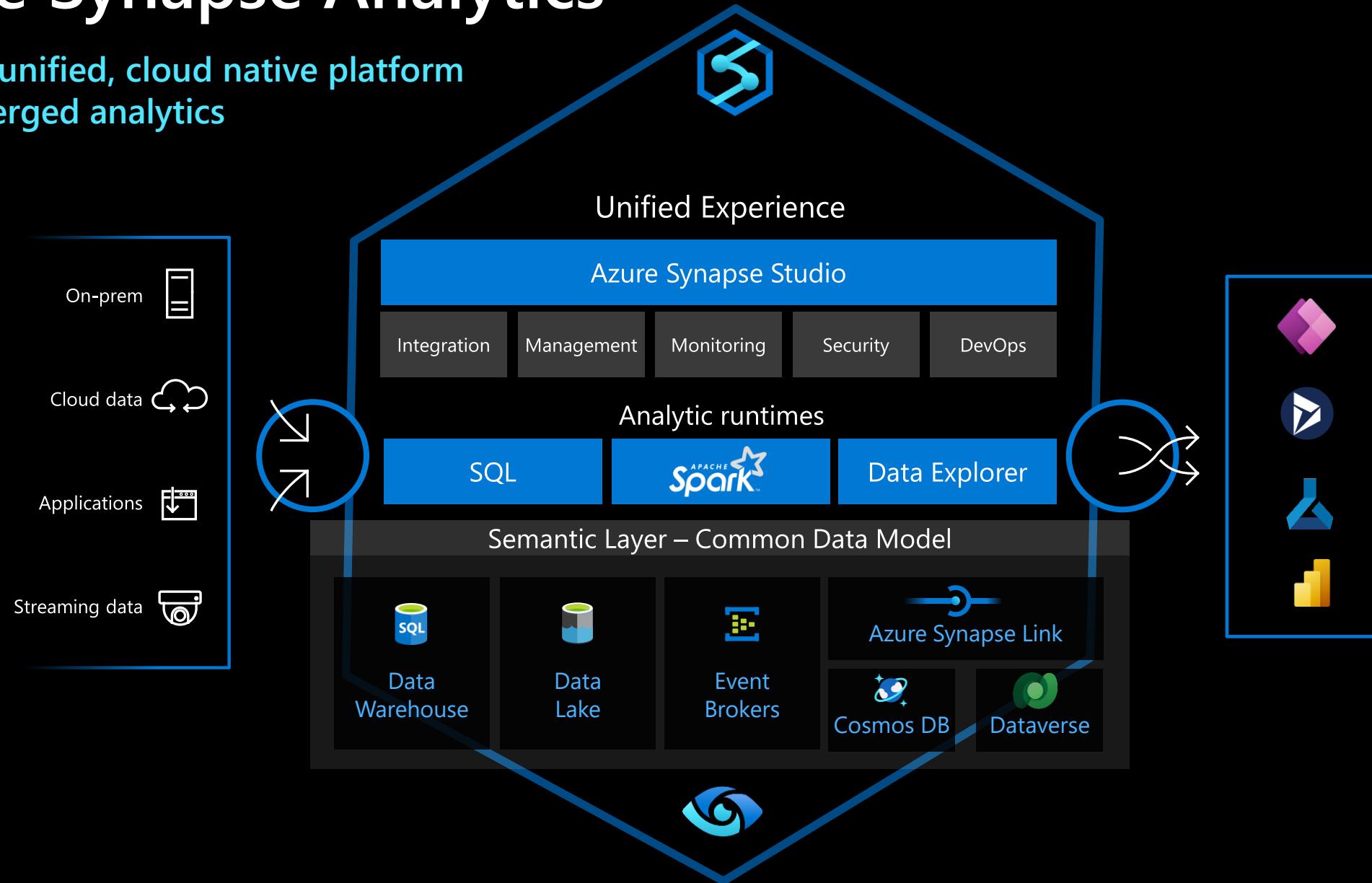
- Détachement de la scalabilité des services, plus de "capacity planning"
- Détachement de la notion de performance
- Consommation "à la demande" de "l'analytique"
- Les limites d'une telle approche
- **Réflexion sur "où" est calculé l'indicateur, modèles hybrides/composites**
- Couts variables pour les métiers (dépendant de l'utilisation), besoin de changer l'approche budgétaire.

L'approche Lakehouse, multi moteur analytique



Azure Synapse Analytics

The first **unified, cloud native platform** for converged analytics





Azure Synapse Analytics

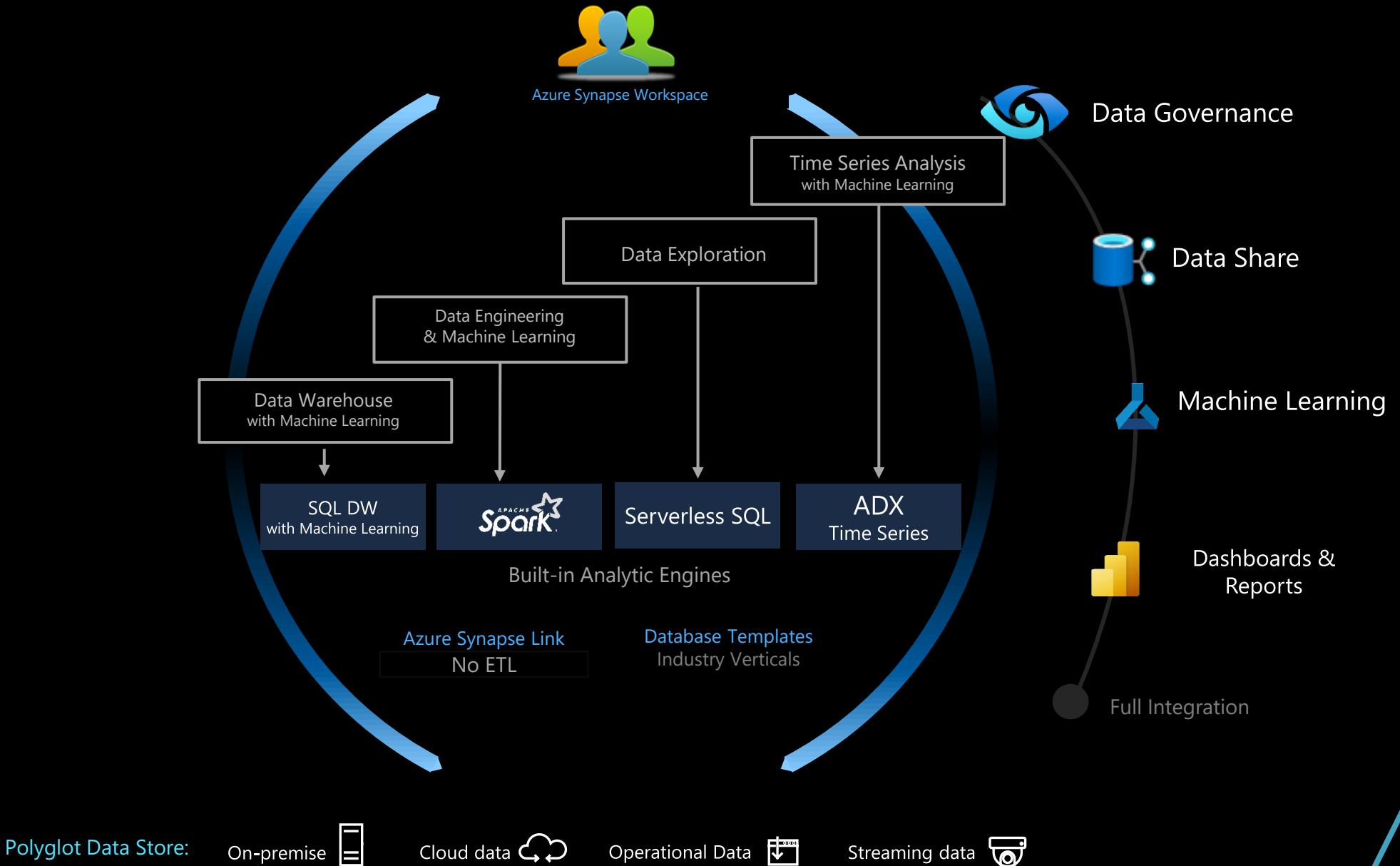
Unified, Secured, and Integrated Ecosystem





Azure Synapse Analytics

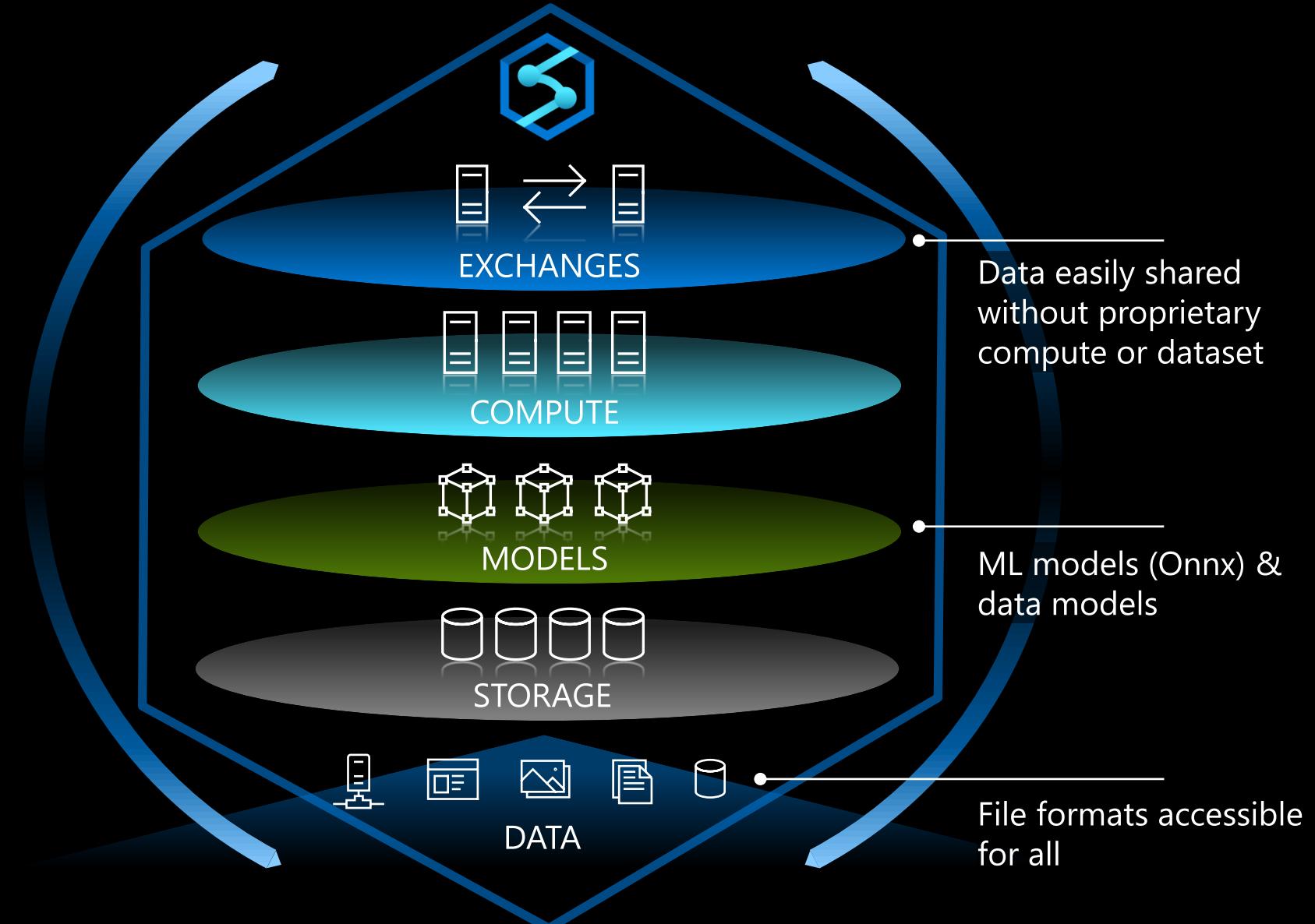
Unified, Secured, and Integrated Ecosystem



Why Synapse Open Hub

Hub & Spoke Model

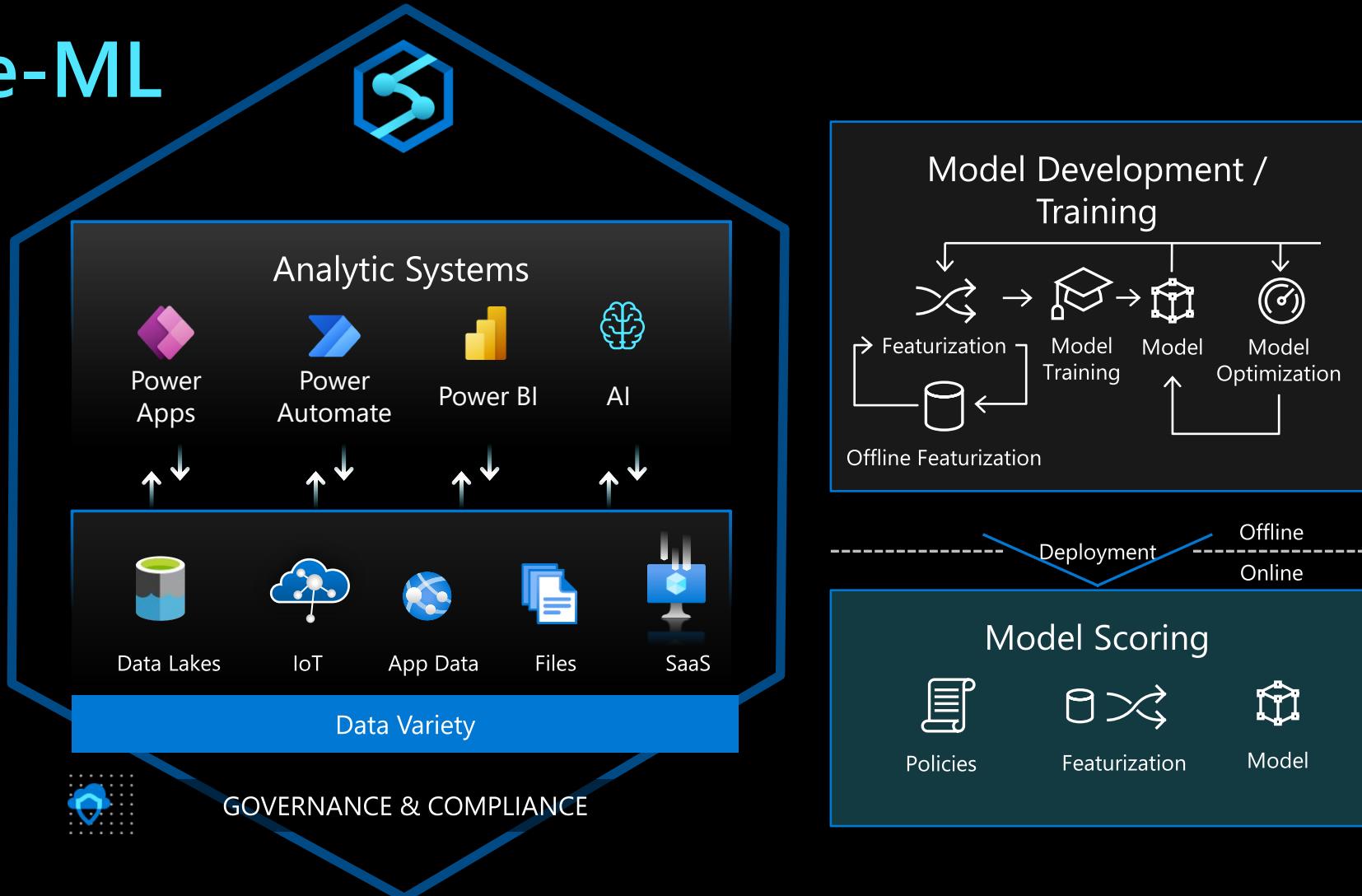
- Integrated with hundreds of ISVs
- Compute choices using both proprietary and open-source tech
- Open lake format
- Open file formats



Why Synapse Enterprise Grade-ML

Productizing AI

- Train in the cloud within the Hub
- Scoring with operational systems
- Governance everywhere (models, lineage)
- Ethical AI
- Control over deployment
- Deployment across Apps, BI, Processes
- Exchange of models (ONNX)
- Enabling Reinforcement learning





Azure Synapse Analytics

Apache Spark



Azure Synapse Apache Spark - Summary

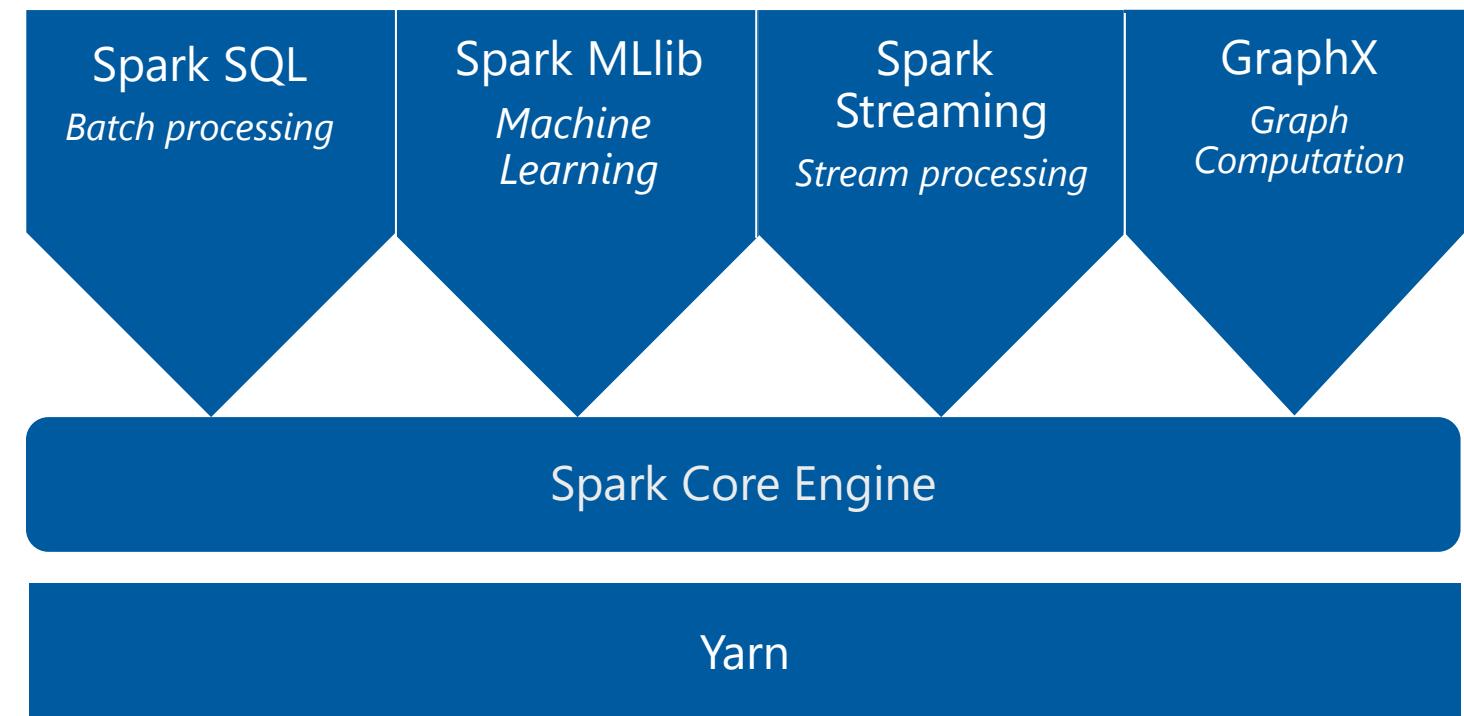
- **Apache Spark 2.4/3.1 derivation**
 - Linux Foundation Delta Lake 0.6/1.2 support
 - .Net Core 3.0/3.1 support
 - Python 3.6/3.8 + Anacondas support
- **Tightly coupled to other Azure Synapse services**
 - Integrated security and sign on
 - Integrated Metadata
 - Integrated and simplified provisioning
 - Integrated UX including Jupyter based notebooks
 - Fast load of Synapse SQL (provisioned) 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

Apache Spark

An unified, open source, parallel, data processing framework for Big Data Analytics

Spark Unifies:

- Batch Processing
- Interactive SQL
- Real-time processing
- Machine Learning
- Deep Learning
- Graph Processing



<http://spark.apache.org>

Motivation for Apache Spark

Traditional Approach: MapReduce jobs for complex jobs, interactive query, and online event-hub processing involves lots of (slow) disk I/O

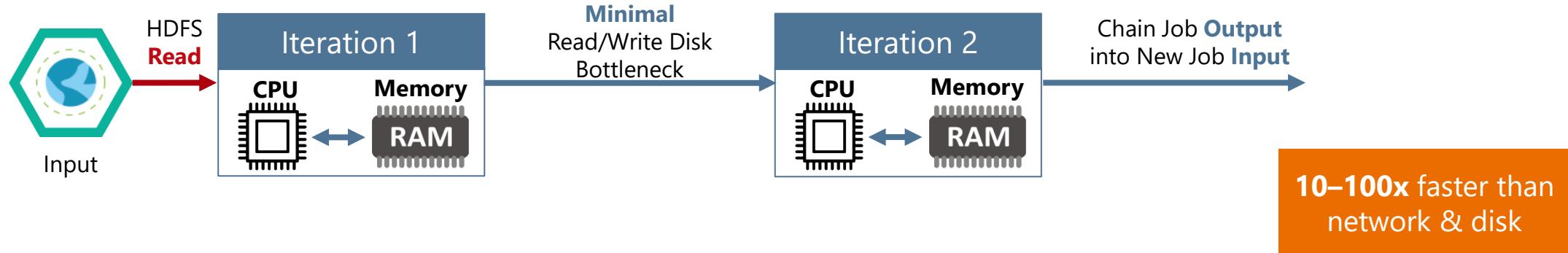


Motivation for Apache Spark

Traditional Approach: MapReduce jobs for complex jobs, interactive query, and online event-hub processing involves lots of **(slow) disk I/O**

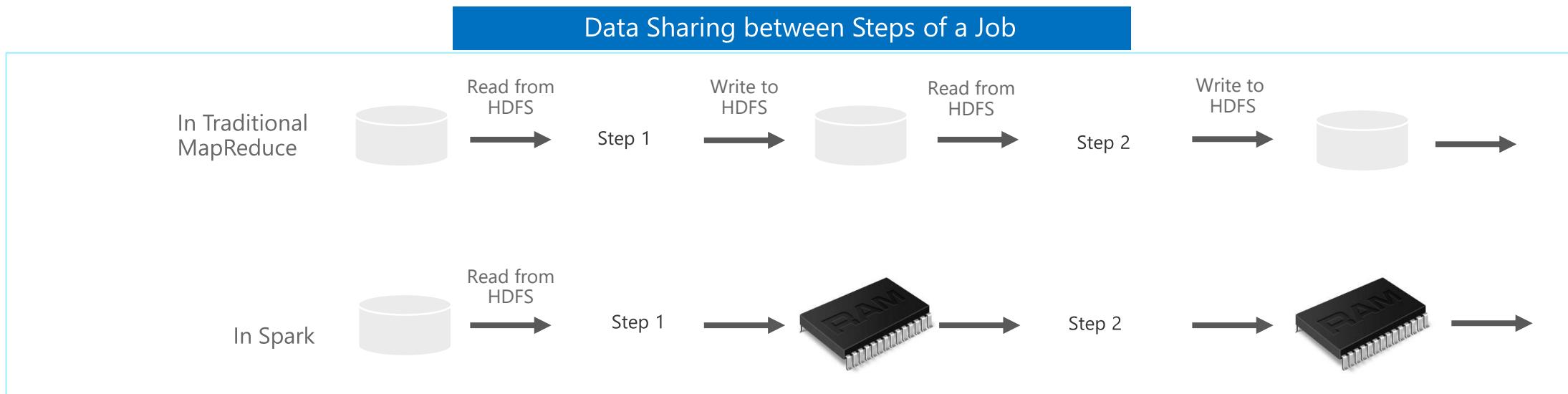


Solution: Keep data **in-memory** with a new distributed execution engine



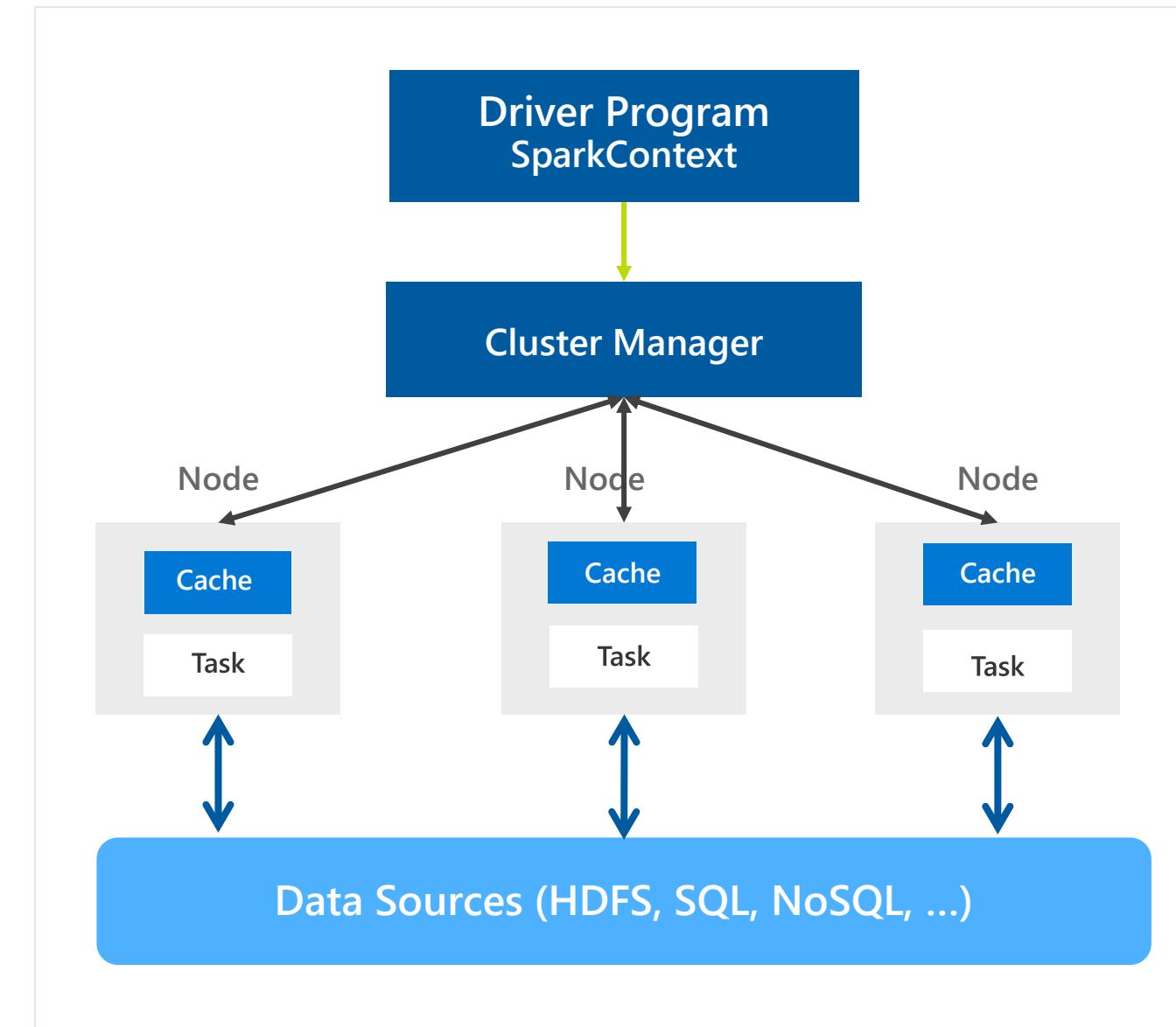
What makes Spark fast

- **In-memory cluster computing:** Spark provides primitives for *in-memory* cluster computing. A Spark job can *load and cache* data into memory and query it repeatedly (iteratively) much quicker than disk-based systems.
- **Scala Integration:** Spark integrates into the Scala programming language, letting you manipulate distributed datasets like local collections. No need to structure everything as map and reduce operations
- **Faster Data-sharing:** Data-sharing between operations is faster as data is in-memory:
 - In (traditional) Hadoop data is shared through HDFS which is expensive. HDFS maintains three replicas.
 - Spark stores data in-memory *without any replication*.



General Spark Cluster Architecture

- 'Driver' runs the user's 'main' function and executes the various parallel operations on the worker nodes.
- The results of the operations are collected by the driver
- The worker nodes read and write data from/to Data Sources including HDFS.
- Worker node also cache transformed data in memory as RDDs (Resilient Data Sets).
- Worker nodes and the Driver Node execute as VMs in public clouds (AWS, Google and Azure).



Spark Component Features

Spark SQL

- Unified data access: Query structured data sets with SQL or DataFrame APIs
- Fast, familiar query language across all your enterprise data
- Use BI tools to connect and query via JDBC or ODBC drivers

Mlib/SparkML

- Predictive and prescriptive analytics
- Machine learning algorithms for:
 - Clustering
 - Classification
 - Regression
 - etc.
- Smart application design from pre-built, out-of-the-box statistical and algorithmic models

Spark Streaming

- Micro-batch event processing for near-real time analytics
- e.g. Internet of Things (IoT) devices, Twitter feeds, Kafka (event hub), etc.
- Spark's engine drives some action or outputs data in batches to various data stores

GraphX

- Represent and analyze systems represented by graph nodes
- Trace interconnections between graph nodes
- Applicable to use cases in transportation, telecommunications, road networks, modeling personal relationships, social media, etc.



Azure Synapse Apache Spark

Architecture Overview

Creating a Spark pool (1 of 2)

Provision Spark Pool through Azure Portal with default settings or per requirements

Basic Settings – Minimum details required from user

Home > Synapse workspaces > euang-synapse-nov-ws - Apache Spark pools > Create Apache Spark pool

Create Apache Spark pool

Basics * Additional settings * Tags Summary

Create a Synapse Analytics Apache Spark pool with your preferred configurations. Complete the Basics tab then go to Review + create to provision with smart defaults, or visit each tab to customize.

Apache Spark pool details

Name your Apache Spark pool and choose its initial settings.

Apache Spark pool name *

Enter Apache Spark pool name

Node size family

MemoryOptimized

Node size *

Medium (8 vCPU / 64 GB)

Autoscale * ⓘ

Enabled Disabled

Number of nodes *

3 40

Only required field from user

Default Settings

Creating a Spark pool (2 of 2) - optional

Additional Settings offer optional settings to customize Spark pool

Customize component versions, auto-pause

Import libraries by providing text file containing library name and version

Home > prlangadws2 > Create Apache Spark pool

Create Apache Spark pool

Basics * Additional settings * Tags Summary

Customize additional configuration parameters including autoscale and component versions.

Auto-pause

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

Auto-pause * ⓘ Enabled Disabled

Number of minutes idle * 15

Component versions

Select the Apache Spark version for your Apache Spark pool.

Component	Version
Apache Spark *	2.4
Python	3.6.1
Scala	2.11.12
Java	1.8.0_222
.NET Core	3.1
.NET for Apache Spark	0.10.0
Delta Lake	0.5.0

Packages

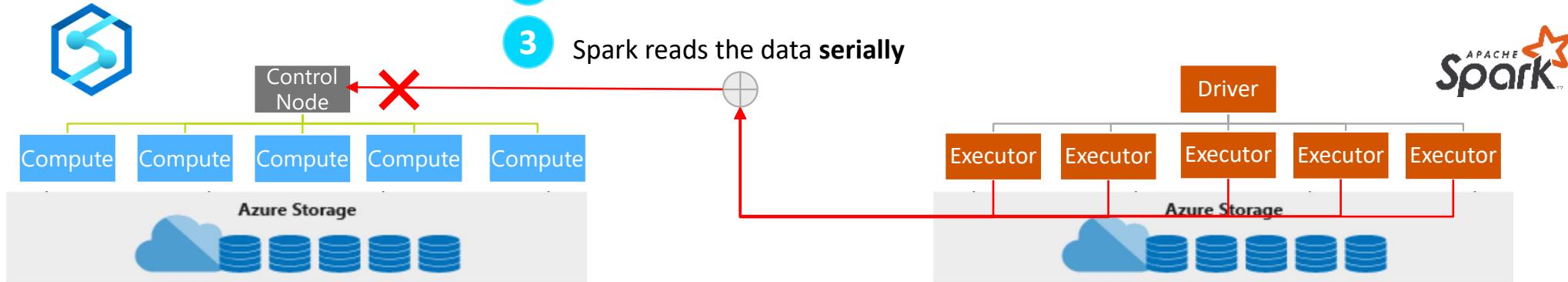
Upload environment configuration file ("PIP freeze" output).

File upload Select a file

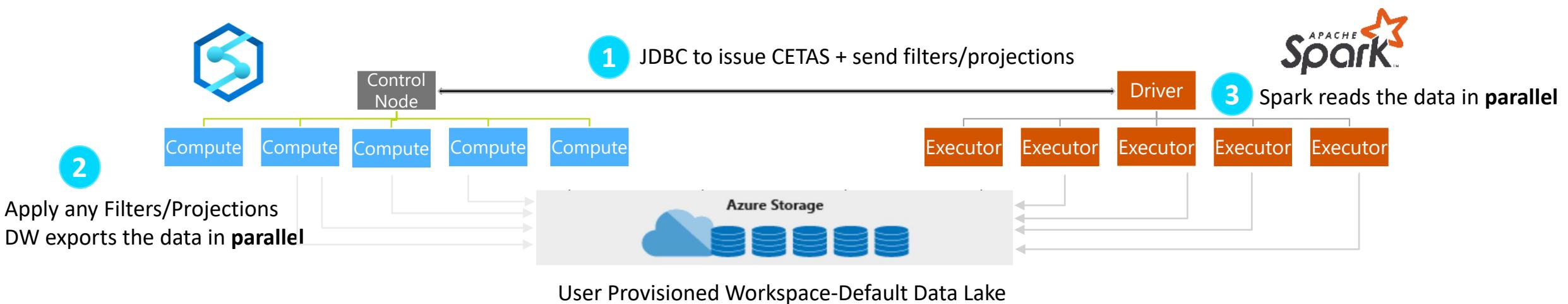
Upload

Review + create < Previous Next: Tags >

Existing Approach: JDBC



New Approach: JDBC and Polybase



Code-Behind Experience

Existing Approach

```
val jdbcUsername = "<SQL DB ADMIN USER>"  
val jdbcPwd = "<SQL DB ADMIN PWD>"  
val jdbcHostname = "servername.database.windows.net"  
val jdbcPort = 1433  
val jdbcDatabase = "<AZURE SQL DB NAME>"  
  
val jdbc_url =  
  s"jdbc:sqlserver://${jdbcHostname}:${jdbcPort};database=${jdbcDatabase};"  
  encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.databas  
e.windows.net;loginTimeout=60;"  
  
val connectionProperties = new Properties()  
  
connectionProperties.put("user", s"${jdbcUsername}")  
connectionProperties.put("password", s"${jdbcPwd}")  
  
val sqlTableDf = spark.read.jdbc(jdbc_url, "dbo.Tbl1", connectionProperties)
```

New Approach

```
// Construct a Spark DataFrame from dedicated SQL pool  
var df = spark.read.sqlAnalytics("sql1.dbo.Tbl1")  
  
// Write the Spark DataFrame into dedicated SQL pool  
df.write.sqlAnalytics("sql1.dbo.Tbl2")
```

Create Notebook on files in storage

The screenshot illustrates the process of creating a Notebook on files stored in Azure Storage.

Left Panel (Storage Explorer):

- Shows the Azure Data blade.
- Selected Storage account: **prlangaddemo (Primary)**.
- Container: **nyctic**.
- File: **part-00055** (highlighted with a red box).
- Context menu options for the file include: **New SQL script**, **New notebook** (highlighted with a red box), **Copy ABFSS path**, and **Manage Access...**.

Right Panel (Job History):

- Shows the **Data** blade.
- Attached to **priangadSpark2**.
- Language: **PySpark (Python)**.
- Cell 1 code:

```
[3] 1 %%pyspark
2 data_path = spark.read.load('abfss://nyctic@prlangaddemo.dfs.core.windows.net/yellow/puYear=2015/puMonth=3/part-00133-tid-210938564719836543-aea5b543-5e83-')
3 data_path.show(10)
```
- Job execution status:
 - Job 0**: load at NativeMethodAccessorImpl.java:0 - Succeeded (Duration: 7s)
 - Job 1**: showString at NativeMethodAccessorImpl.java:0 - Succeeded (Duration: 1s)
 - Job 2**: showString at NativeMethodAccessorImpl.java:0 - Succeeded (Duration: 11s)
- Job output preview (partial):

Vendor ID	Pickup Date Time	Pickup Off Date Time	Passenger Count	Trip Distance	Pu Location ID	Do Location ID	Start Lon	Start Lat	End Lon	End Lat
2	2015-02-28 23:53:18	2015-03-01 00:00:29	6	1.63	null	null	-74.00084686279297	40.73069381713867	-73.9841537475586	40.74470520019531
1	N	1	7.5	0.5	0.5	1	0.3	1.76	10.56	null
1	2015-03-28 19:21:05	2015-03-28 19:28:31	1	2.2	null	null	-73.97765350341797	40.763160705566406	-73.95502471923828	40.78600311279297
1	N	1	8.5	0.8	0.5	1	0.3	2.3	0.0	11.6
2	2015-02-28 23:53:19	2015-03-01 00:12:08	5	3.23	null	null	-73.96012878417969	40.76215744018555	-73.9881591796875	40.72818896484375
1	N	1	14.5	0.5	0.5	1	0.3	4.74	0.0	28.54
1	2015-03-28 19:21:05	2015-03-28 19:37:02	1	2.1	null	null	-73.98143005371094	40.7815055847168	-74.000891552734375	40.76177215576172

Develop Hub - Notebooks

Notebooks

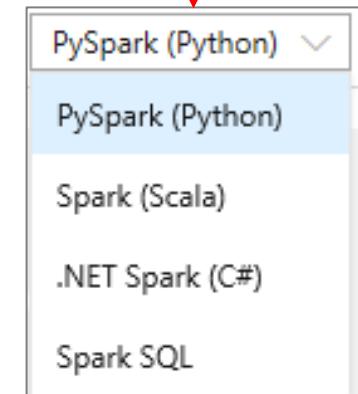
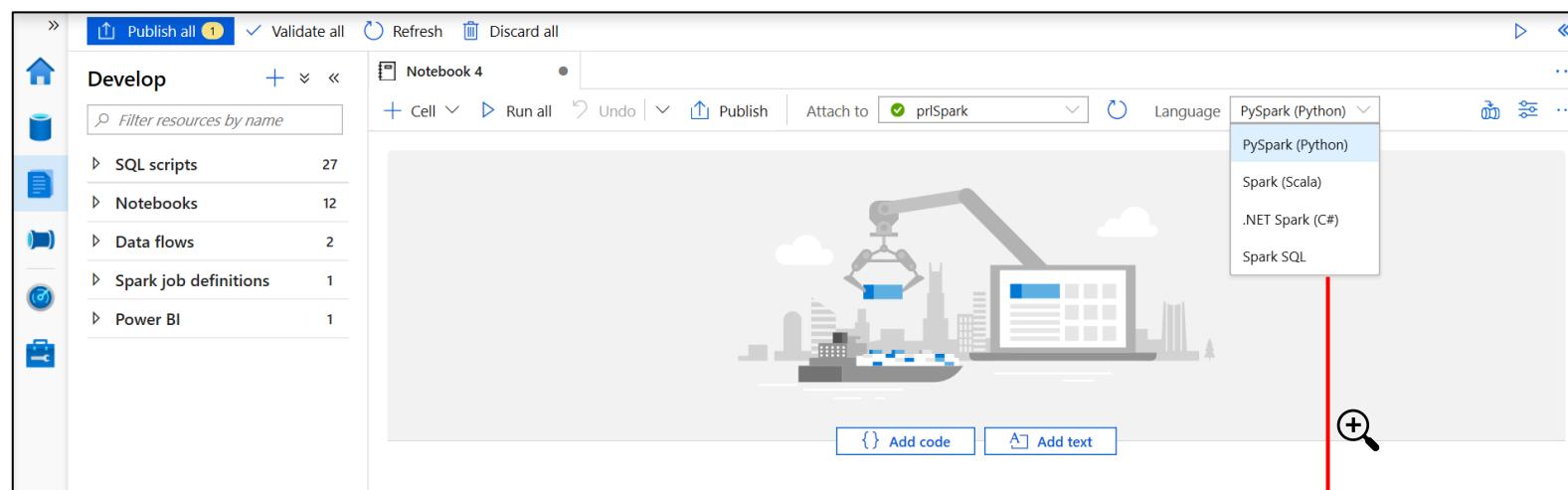
Allows to write multiple languages in one notebook

`%%<Name of language>`

Offers use of temporary tables across languages

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

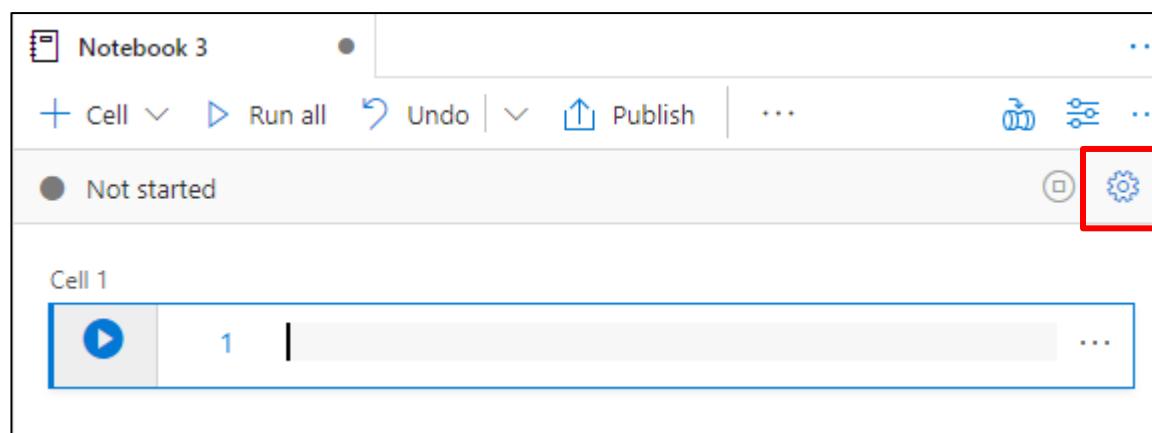
Export results



Develop Hub - Notebooks

Configure session allows developers to control how many resources are devoted to running their notebook.

Provides quick links to monitor session and Spark history server



Configure session

Livy session ID

Status
Not started

Attach to * ⓘ
analytics1

analytics1
Refresh at 12:04:28 AM

Medium (8 vCores / 56 GB) 3 - 10 nodes
0.00% utilized

Available session sizes ⓘ

Small	19 executors	Use
Medium	9 executors	Use

Executor size * ⓘ
Small (4 vCores, 28GB memory)

Executors * ⓘ
2

Driver size * ⓘ
Small (4 vCores, 28GB memory)

Session timeout (minutes) * ⓘ
30

[Apply](#) [Cancel](#)

Develop Hub - Notebooks

As notebook cells run, the underlying Spark application status is shown. Providing immediate feedback and progress tracking.

The screenshot shows the Microsoft Azure Synapse Analytics Develop Hub - Notebooks interface. At the top, there's a navigation bar with 'Microsoft Azure' and 'Synapse Analytics'. A search bar says 'Search resources' and a user profile 'prlangad@microsoft.com MICROSOFT' is on the right. Below the navigation is a toolbar with 'Publish all', 'Validate all', 'Refresh', and 'Discard all' buttons. The main area shows a notebook titled 'opendataset' with a cell named 'Notebook 1'. The cell contains the following PySpark code:

```

1 %pyspark
2 data_path = spark.read.load('abfss://opendataset@internalsandboxwe.dfs.core.windows.net/holidays/part-00000-bd1ab'
3 data_path.show(100)

```

Below the code, it says 'Command executed in 2mins 44s 998ms by prlangad on 03-19-2020 11:31:56.458 -07:00'. Underneath, there's a section titled 'Job execution Succeeded Spark 2 executors 8 cores' with three entries:

ID	DESCRIPTION	STATUS	STAGES	TASKS	SUBMISSION TIME	DURATION
▶ Job 0	load at NativeMethodAccessorImpl.java:0	✓ Succeeded	1/1	<div style="width: 100%; background-color: #2e7131;"></div>	3/19/2020, 11:31:35 AM	6s
▶ Job 1	showString at NativeMethodAccessorImpl.java:0	✓ Succeeded	1/1	<div style="width: 100%; background-color: #2e7131;"></div>	3/19/2020, 11:31:43 AM	1s
▶ Job 2	showString at NativeMethodAccessorImpl.java:0	✓ Succeeded	1/1	<div style="width: 100%; background-color: #2e7131;"></div>	3/19/2020, 11:31:45 AM	9s

At the bottom, there's a preview of the data being processed:

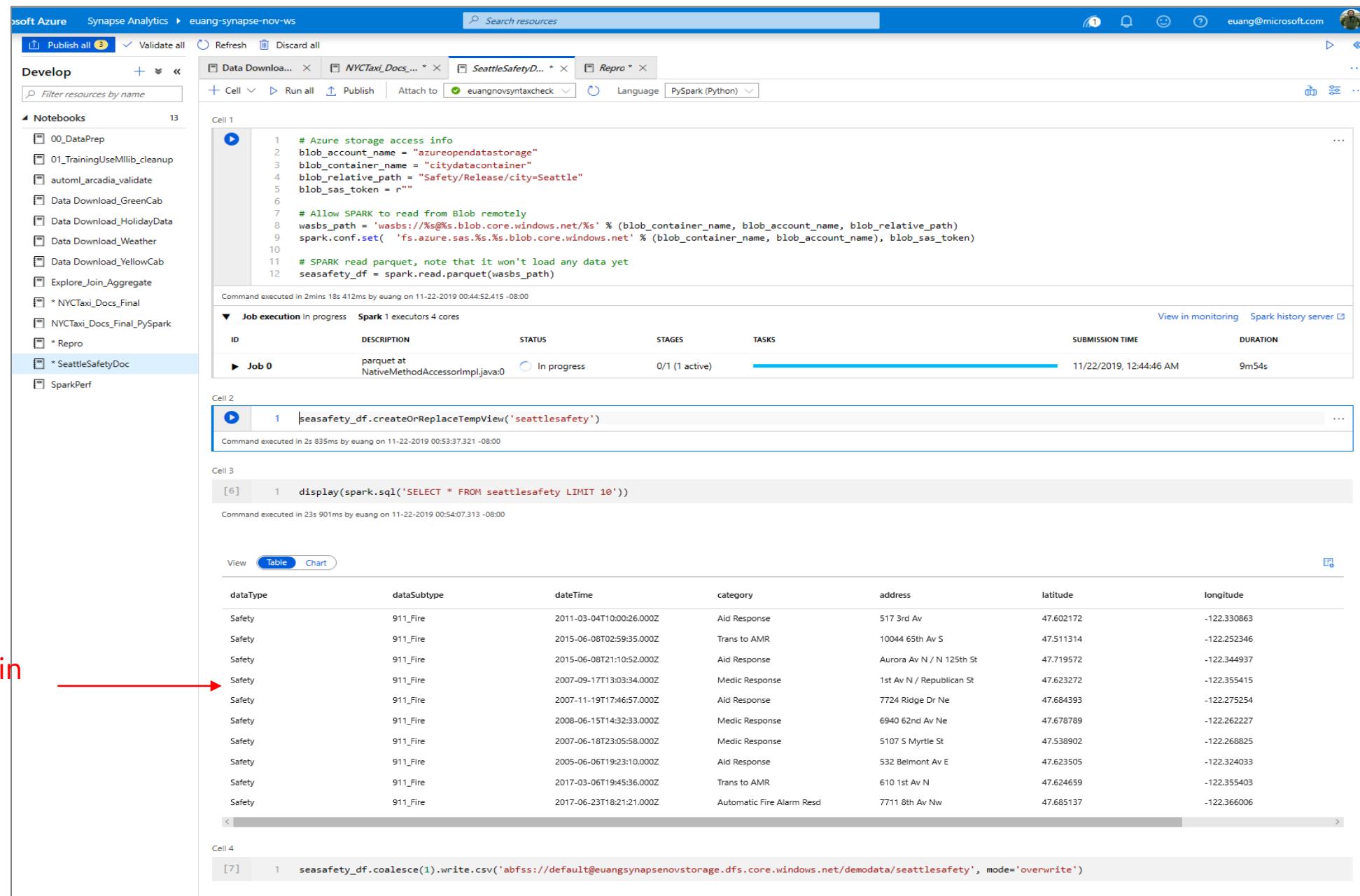
```

+---+-----+-----+-----+-----+-----+-----+-----+
| vendorID | tpepPickupDateTime | tpepDropoffDateTime | passengerCount | tripDistance | puLocationId | doLocationId | startLon | startLat |
| endLon | endLat | rateCodeId | storeAndFwdFlag | paymentType | fareAmount | extra | mtaTax | improvementSurcharge | tipAmount | tollsAmount |
| totalAmount |
+---+-----+-----+-----+-----+-----+-----+-----+
| CMT | 2009-04-30 23:59:52 | 2009-05-01 00:11:14 | 0 | 1 | Credit | 1.9 | null | null | -73.984708 | null | 1.8 |
| 40.760237 | -73.960426 | 40.761527 | null | 0 | 1 | Credit | 8.5 | 0.0 | null | null | null | 0.0 |
| 10.3 |
| CMT | 2009-05-07 01:03:26 | 2009-05-07 01:14:11 | 0 | 1 | Credit | 3.4 | null | null | null | -73.956527 | null |
| 40.771307 | -73.941002 | 40.80763 | null | 0 | 1 | Credit | 9.7 | 0.0 | null | null | null | 0.0 |
| 12.25 |
| CMT | 2009-04-30 23:50:42 | 2009-05-01 00:06:43 | 1 | 2.2 | null | null | null | null | -74.009102 |
| 0.0 |

```

At the very bottom, there are buttons for 'Ready' (with a checkmark), 'Stop session', and 'Configure session'.

View results in table format 



```

# Azure storage access info
blob_account_name = "azureopendatastorage"
blob_container_name = "citydatacontainer"
blob_relative_path = "Safety/Release/city=Seattle"
blob_sas_token = r""

# Allow SPARK to read from Blob remotely
wasbs_path = 'wasbs://{}@{}.blob.core.windows.net/{}'.format(blob_container_name, blob_account_name, blob_relative_path)
spark.conf.set('fs.azure.sas.{}.blob.core.windows.net'.format(blob_container_name), blob_sas_token)

# SPARK read parquet, note that it won't load any data yet
seasafety_df = spark.read.parquet(wasbs_path)

```

Command executed in 2mins 18s 412ms by euang on 11-22-2019 00:44:52.415 -08:00

Job execution In progress Spark 1 executors 4 cores

ID	DESCRIPTION	STATUS	STAGES	TASKS	SUBMISSION TIME	DURATION
Job 0	parquet at NativeMethodAccessImpl.java:0	In progress	0/1 (1 active)		11/22/2019, 12:44:46 AM	9m54s

Cell 2

```
1 seasafety_df.createOrReplaceTempView('seattlesafety')
```

Command executed in 2s 835ms by euang on 11-22-2019 00:53:37.321 -08:00

Cell 3

```
[6] 1 display(spark.sql('SELECT * FROM seattlesafety LIMIT 10'))
```

Command executed in 23s 901ms by euang on 11-22-2019 00:54:07.313 -08:00

View **Table** **Chart**

dataType	dataSubtype	dateTime	category	address	latitude	longitude
Safety	911_Fire	2011-03-04T10:00:26.000Z	Aid Response	517 3rd Av	47.602172	-122.330863
Safety	911_Fire	2015-06-08T02:59:35.000Z	Trans to AMR	10044 65th Av S	47.511314	-122.252346
Safety	911_Fire	2015-06-08T21:10:52.000Z	Aid Response	Aurora Av N / N 125th St	47.719572	-122.344937
Safety	911_Fire	2007-09-17T13:03:34.000Z	Medic Response	1st Av N / Republican St	47.623272	-122.355415
Safety	911_Fire	2007-11-19T17:46:57.000Z	Aid Response	7724 Ridge Dr Ne	47.684393	-122.275254
Safety	911_Fire	2008-06-15T14:32:33.000Z	Medic Response	6940 62nd Av Ne	47.678789	-122.262227
Safety	911_Fire	2007-06-18T23:05:58.000Z	Medic Response	5107 S Myrtle St	47.538902	-122.268825
Safety	911_Fire	2005-06-06T19:23:10.000Z	Aid Response	532 Belmont Av E	47.623505	-122.324033
Safety	911_Fire	2017-03-06T19:45:36.000Z	Trans to AMR	610 1st Av N	47.624659	-122.355403
Safety	911_Fire	2017-06-23T18:21:21.000Z	Automatic Fire Alarm Resd	7711 8th Av Nw	47.685137	-122.366006

Cell 4

```
[7] 1 seasafety_df.coalesce(1).write.csv('abfss://default@euangsynapsenovstorage.dfs.core.windows.net/demodata/seattlesafety', mode='overwrite')
```

View results in chart format

SQL support

The screenshot shows the Azure Synapse Analytics workspace interface. At the top, there's a navigation bar with 'Synapse Analytics' and a user profile. Below it is a toolbar with various icons like 'Publish all', 'Validate all', 'Refresh', and 'Discard all'. A search bar labeled 'Search resources' is also present.

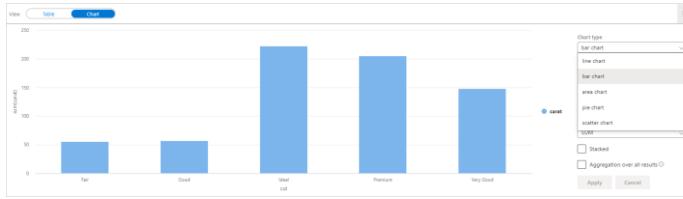
The main area is a notebook editor with four cells:

- Cell 1:** Shows PySpark code for reading data from Azure Blob Storage. The language dropdown is highlighted with a red box. The code includes setting up storage access info, defining a blob path, and reading a parquet file. It also includes a command to view monitoring and Spark history server.
- Cell 2:** Shows a single line of code: `seasafety_df.createOrReplaceTempView('seattlesafety')`.
- Cell 3:** Shows a single line of code: `display(spark.sql('SELECT * FROM seattlesafety'))`. This line is highlighted with a red box and has a red arrow pointing to the text "SQL support". The result is a pie chart visualizing data from the temp view. The chart is titled "longitude" and has several segments labeled with incident types: "Aid Response" (blue), "Medic Response" (green), "Trans to AMR" (black), and others like "Automatic Fire Alarm False", "Medic Response, 7 per Rule", etc. A configuration panel on the right allows setting chart type (pie chart), X axis column (category), Y axis columns (longitude), Aggregation (COUNT), Y axis label (Total), X axis label (category), and a button to "Apply".
- Cell 4:** Shows the final step of writing the data back to Azure Blob Storage: `seasafety_df.coalesce(1).write.csv('abfss://default@euangsynapsenovstorage.dfs.core.windows.net/demodata/seattlesafety', mode='overwrite')`.

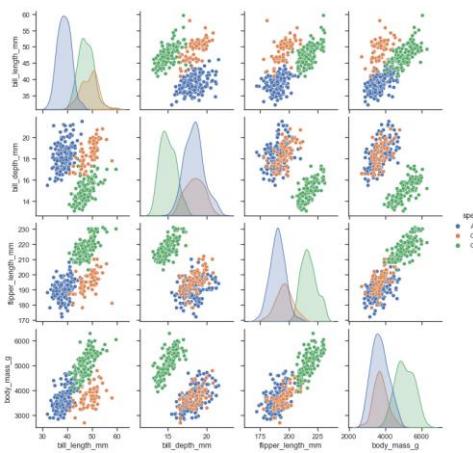
Develop Hub

Explore data by using native visuals in Spark notebooks

Fonction display(df)



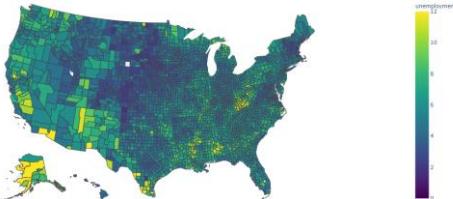
Seaborn



```
import seaborn as sns
sns.set_theme(style="ticks")
```

```
df = sns.load_dataset("penguins")
sns.pairplot(df, hue="species")
```

Plotly



```
import json
with
urlopen('https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-fips.json') as response:
    counties = json.load(response)
```

```
import pandas as pd
df =
pd.read_csv("https://raw.githubusercontent.com/plotly/datasets/master/fips-unemp-16.csv",
            dtype={"fips": str})
```

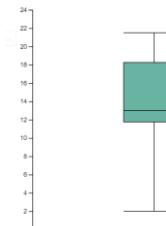
```
import plotly
import plotly.express as px
```

```
fig = px.choropleth(df, geojson=counties, locations='fips',
color='unemp',
        color_continuous_scale="Viridis",
        range_color=(0, 12),
        scope="usa",
        labels={'unemp':'unemployment rate'}
    )
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
```

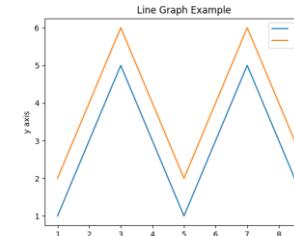
```
# create an html document that embeds the Plotly plot
h = plotly.offline.plot(fig, output_type='div')
```

```
# display this html
displayHTML(h)
```

displayHTML() option



Matplotlib



```
# Bar chart
```

```
import matplotlib.pyplot as plt
```

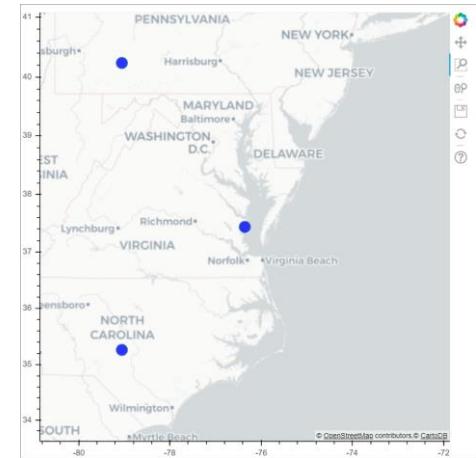
```
x1 = [1, 3, 4, 5, 6, 7, 9]
y1 = [4, 7, 2, 4, 7, 8, 3]
```

```
x2 = [2, 4, 6, 8, 10]
y2 = [5, 6, 2, 6, 2]
```

```
plt.bar(x1, y1, label="Blue Bar", color='b')
plt.bar(x2, y2, label="Green Bar", color='g')
plt.plot()
```

```
plt.xlabel("bar number")
plt.ylabel("bar height")
plt.title("Bar Chart Example")
plt.legend()
plt.show()
```

Bokeh



```
from bokeh.plotting import figure, output_file
from bokeh.tile_providers import get_provider, Vendors
from bokeh.embed import file_html
from bokeh.resources import CDN
from bokeh.models import ColumnDataSource
```

```
tile_provider = get_provider(Vendors.CARTODBPOSITRON)

# range bounds supplied in web mercator coordinates
p = figure(x_range=(-9000000,-8000000),
y_range=(4000000,5000000),
x_axis_type="mercator", y_axis_type="mercator")
p.add_tile(tile_provider)
```

```
# plot datapoints on the map
source = ColumnDataSource(
    data=dict(x=[ -8800000, -8500000 , -8800000],
              y=[4200000, 4500000, 4900000])
)
```

```
p.circle(x="x", y="y", size=15, fill_color="blue", fill_alpha=0.8,
source=source)
```

```
# create an html document that embeds the Bokeh plot
html = file_html(p, CDN, "my plot1")
```

```
# display this html
displayHTML(html)
```

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

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

Constraints

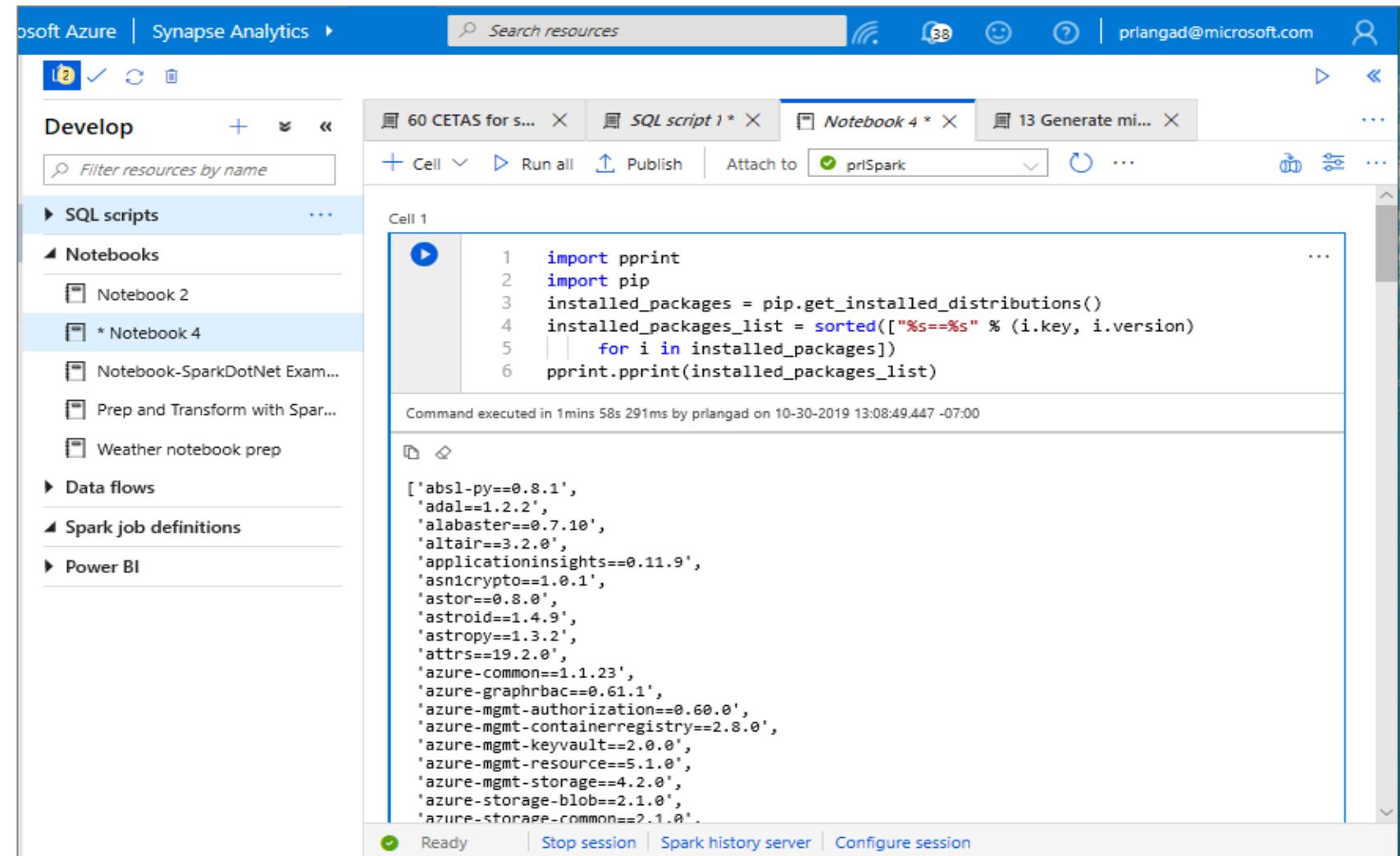
The library version must exist on PyPI repository

Version downgrade of an existing library not allowed

The screenshot shows the Microsoft Azure Synapse Analytics Library Management interface. The left sidebar includes options like Analytics pools, SQL pools, Apache Spark pools (which is selected), External connections, Linked services, Orchestration, Triggers, Integration runtimes, Security, Access control, and Managed private endpoints. The main area displays 'Apache Spark pools' with three items listed: 'priLangadSpark2', 'priLang-syntaxcheck', and 'priSpark'. A 'Properties' panel on the right shows details for 'priSpark': Name (priSpark), URL (/subscriptions/56f8824d-32b0-4825-9825-02fa6a801546/resourceGroups/priLangadrg/pro...), Creation date (10/30/2019, 12:50:37 PM), Configuration, and Workspace. Below these is a 'Packages' section with a red box around it, containing a 'Upload environment config file' button and a 'Refresh' button. A message states 'No user-provided packages currently uploaded. You can upload "environment config file".' At the bottom right is a 'Close' button.

Library Management - Python

Get list of installed libraries with version information



The screenshot shows the Azure Synapse Analytics interface with the 'Develop' workspace selected. In the center, a notebook titled 'Notebook 4' is open. A code cell in this notebook contains the following Python script:

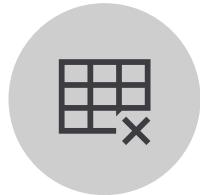
```
1 import pprint
2 import pip
3 installed_packages = pip.get_installed_distributions()
4 installed_packages_list = sorted(['%s==%s' % (i.key, i.version)
5         for i in installed_packages])
6 pprint.pprint(installed_packages_list)
```

The output of this command is displayed below the code cell, listing numerous Python packages and their versions. The output starts with:

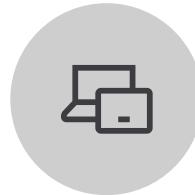
```
['absl-py==0.8.1',
 'adal==1.2.2',
 'alabaster==0.7.10',
 'altair==3.2.0',
 'applicationinsights==0.11.9',
 'asn1crypto==1.0.1',
 'astor==0.8.0',
 'astroid==1.4.9',
 'astropy==1.3.2',
 'attrs==19.2.0',
 'azure-common==1.1.23',
 'azure-graphrbac==0.61.1',
 'azure-mgmt-authorization==0.60.0',
 'azure-mgmt-containerregistry==2.8.0',
 '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 of the interface, there are buttons for 'Ready', 'Stop session', 'Spark history server', and 'Configure session'.

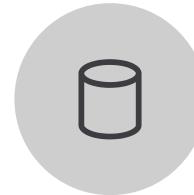
Dataflow Capabilities



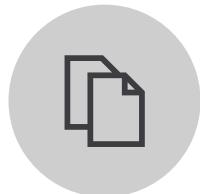
Handle upserts, updates, deletes on sql sinks



Add new partition methods



Add schema drift support



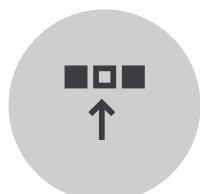
Add file handling (move files after read, write files to file names described in rows etc)



New inventory of functions (for e.g. Hash functions for row comparison)



Commonly used ETL patterns(Sequence generator/Lookup transformation/SCD...)



Data lineage – Capturing sink column lineage & impact analysis(invaluable if this is for enterprise deployment)

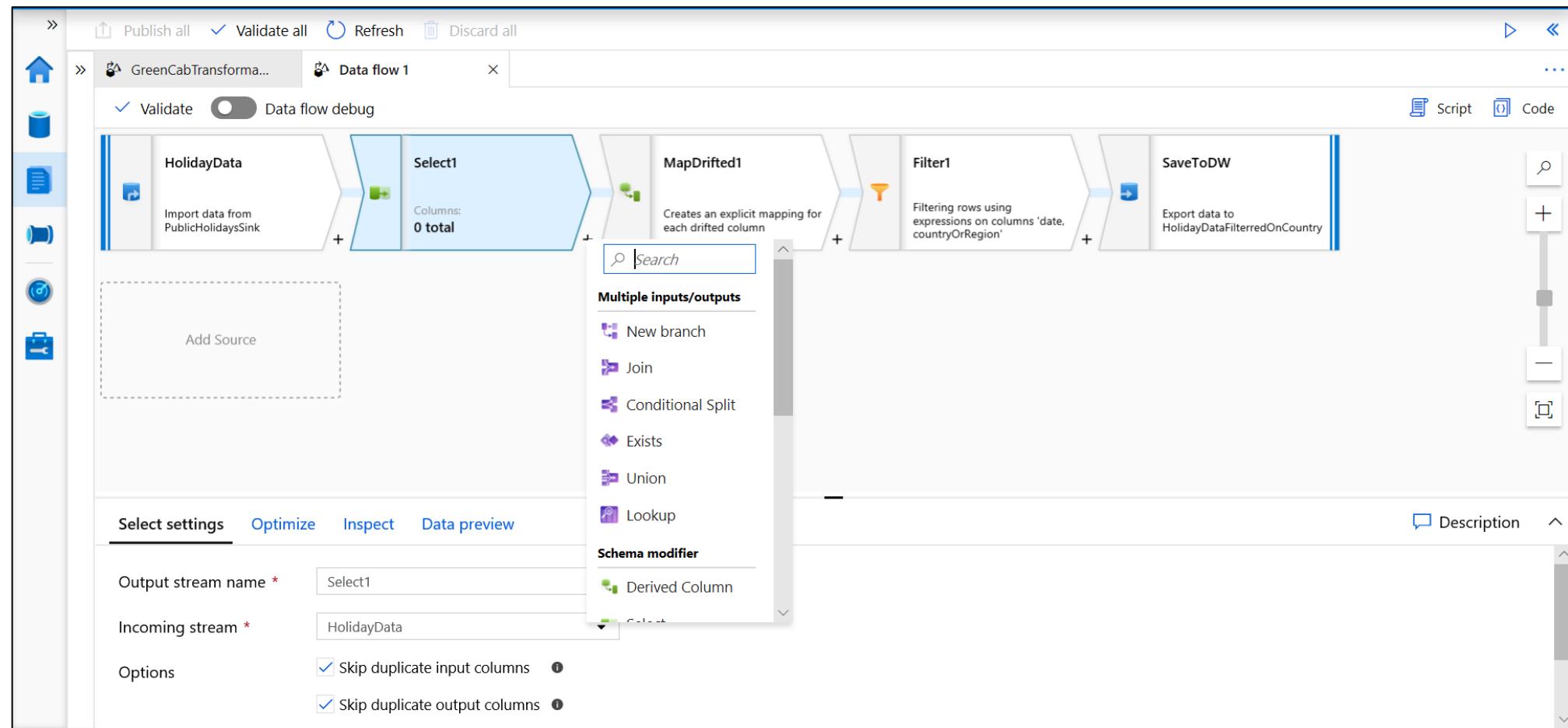


Implement commonly used ETL patterns as templates(SCD Type1, Type2, Data Vault)

Develop Hub - Data Flows

Data flows are a visual way of specifying how to transform data.

Provides a code-free experience.



Spark ML Algorithms

Spark ML Algorithms

Classification and Regression	<ul style="list-style-type: none">• Linear Models (SVMs, logistic regression, linear regression)• Naïve Bayes• Decision Trees• Ensembles of trees (Random Forest, Gradient-Boosted Trees)• Isotonic regression
Clustering	<ul style="list-style-type: none">• k-means and streaming k-means• Gaussian mixture• Power iteration clustering (PIC)• Latent Dirichlet allocation (LDA)
Collaborative Filtering	<ul style="list-style-type: none">• Alternating least squares (ALS)
Dimensionality Reduction	<ul style="list-style-type: none">• SVD• PCA
Frequent Pattern Mining	<ul style="list-style-type: none">• FP-growth• Association rules
Basic Statistics	<ul style="list-style-type: none">• Summary statistics• Correlations• Stratified sampling• Hypothesis testing• Random data generation

SynapseML

Simple and Distributed Machine Learning

Get Started



Coming from [MMLSpark](#)? We have been renamed to **SynapseML**!

[Cognitive Services](#) [Deep Learning](#) [Responsible AI](#) [LightGBM](#) [OpenCV](#)

```
from synapse.ml.cognitive import *

sentiment_df = (TextSentiment()
    .setTextCol("text")
    .setLocation("eastus")
    .setSubscriptionKey(key)
    .setOutputCol("sentiment")
    .setErrorCol("error")
    .setLanguageCol("language")
    .transform(input_df))
```

Read more

Microsoft Spark Utilities

Overview

It provides utilities for working with file systems, including ADLS Gen2 and Azure Blob Storage.

Benefits

It supports multiple methods for file systems such as List, Copy, Move, Write, Append, Delete file or directory, View file properties, Create new directory, Preview file content.

It supports environment utilities to get username, user id, job id, workspace name, pool name, cluster id.

It supports to get the access tokens of linked services and manage secrets in Azure Key Vault.

<https://github.com/solliancenet/azure-synapse-analytics-ga-content-packs/blob/main/hands-on-labs/lab-02/README.md#task-3---explore-the-data-lake-storage-with-the-mssparkutil-library>

The screenshot shows a Jupyter Notebook interface with three code cells:

- Cell 1:** Contains the command `[2] 1 from notebookutils import mssparkutils
2 mssparkutils.fs.help()`. The output shows the command was executed in 437ms by prlangad on 11-24-2020 18:32:02.018 -08:00. Below the command, it says "mssparkutils.fs provides utilities for working with various FileSystems." and lists several methods: cp, mv, ls, mkdirs, putfile, headfile, appendfile, and rm.
- Cell 2:** Contains the command `[3] 1 mssparkutils.credentials.help()`. The output shows the command was executed in 355ms by prlangad on 11-24-2020 18:32:47.633 -08:00. Below the command, it lists methods for getting and putting AKV secrets.
- Cell 3:** Contains the command `[4] 1 mssparkutils.env.help()`. The output shows the command was executed in 472ms by prlangad on 11-24-2020 18:33:14.526 -08:00. Below the command, it lists methods for getting environment variables: getUsername, getUserId, getJobId, getWorkspaceName, getPoolName, and getClusterId.

Hypespace

Overview

Hypespace introduces the ability for Apache Spark users to create indexes on their data

Benefits

It helps accelerate your workloads or queries containing filters on predicates with high selectivity or a join that requires heavy shuffles.

Maintain the indexes through a multi-user concurrency model.

Leverage these indexes automatically, within your Spark workloads, without any changes to your application code for query/workload acceleration.

It supports index operations as create index, list index, restore index, delete index, vacuum index

Languages supported: Scala, Python, .NET

<https://github.com/solliancenet/azure-synapse-analytics-ga-content-packs/blob/main/hands-on-labs/lab-02/README.md#task-2---index-the-data-lake-storage-with-hypespace>

```

Create indexes
Cell 4
[ ] 1 # Create indexes from configurations
2 hyperspace.createIndex(emp_DF, emp_IndexConfig)
3 hyperspace.createIndex(dept_DF, dept_IndexConfig1)
4 hyperspace.createIndex(dept_DF, dept_IndexConfig2)

List indexes
Cell 6
[ ] 1 hyperspace.indexes().show()

Index usage
Cell 8
[ ] 1 # Enable Hypespace
2 Hyperspace.enable(spark)
3
4 emp_DF = spark.read.parquet(emp_Location)
5 dept_DF = spark.read.parquet(dept_Location)
6
7 emp_DF.show(5)
8 dept_DF.show(5)
9
10 # Filter with equality predicate
11 eqFilter = dept_DF.filter("deptId = 20").select("deptName")
12 eqFilter.show()
13
14 hyperspace.explain(eqFilter, True, displayHTML)
15

```

Spark CDM connector

Overview

It offers Spark dataframes to read and write entities in a CDM folder.

Benefits

It supports use of Managed Identities for Azure resources to mediate access to the Azure datalake storage.

Writes from a Spark dataframe to an entity in a CDM folder based on dataframe schema or CDM entity definition.

Supports data in Apache Parquet format and CSV format.

The CDM connector is pre-installed and supports languages: Python, Scala

```

1 # Explicit write, creating an entity in a CDM folder based on a pre-defined model
2
3 # Case 1: Using an entity definition defined in the CDM Github repo
4
5 data = [
6     ["1", "2", "3", 4],
7     ["4", "5", "6", 8],
8     ["7", "8", "9", 4],
9     ["10", "11", "12", 8],
10    ["13", "14", "15", 4]
11 ]
12
13 schema = (StructType()
14     .add(StructField("teamMembershipId", StringType(), True))
15     .add(StructField("systemUserId", StringType(), True))
16     .add(StructField("teamId", StringType(), True))
17     .add(StructField("versionNumber", LongType(), True))
18 )
19
20 df = spark.createDataFrame(spark.sparkContext.parallelize(data,1), schema)
21
22 (df.write.format("com.microsoft.cdm")
23     .option("storage", storageAccountName)
24     .option("manifestPath", container + "/explicitTest/root.manifest.cdm.json")
25     .option("entity", "TeamMembership")
26     .option("entityDefinitionPath", "core/applicationCommon/TeamMembership.cdm.json/TeamMembership")
27     .option("useCdmStandardModelRoot", True) # sets the model root to the CDM CDN schema documents folder
28     .option("useSubManifest", True)
29     .mode("overwrite")
30     .save())
31
32 readDf = (spark.read.format("com.microsoft.cdm")
33     .option("storage", storageAccountName)
34     .option("manifestPath", container + "/explicitTest/root.manifest.cdm.json")
35     .option("entity", "TeamMembership")
36     .load())
37
38 readDf.select("*").show()

```

IntelliJ IDE



Use the Azure Toolkit for IntelliJ plug-in to develop Apache Spark applications and submit applications to Spark pools.

The screenshot shows the IntelliJ IDEA interface with the following details:

- File Bar:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help.
- Project Bar:** DataAnalysis_v3_Maven > src > main > scala > DataAnalysis.scala.
- Editor:** The code editor displays a Scala file named DataAnalysis.scala. The code initializes a SparkSession and performs basic data processing.
- Run/Debug Configurations Dialog:**
 - Name:** DataAnalysisRunConfig
 - Common Run Parameters:** Main class name: DataAnalysis.
 - Locally Run:** Selected tab.
 - VM options:** Empty.
 - Program arguments:** Empty.
 - Working directory:** C:\Users\prlangad\IdeaProjects\DataAnalysis_v3_Maven
 - Environment variables:** HADOOP_HOME=C:\winutils
 - Use classpath of module:** DataAnalysis_v3_Maven
 - Data repositories directory:** C:\Users\prlangad\IdeaProjects\DataAnalysis_v3_Maven\data
 - Data default repo directory:** C:\Users\prlangad\IdeaProjects\DataAnalysis_v3_Maven\data_default
 - Hadoop user default directory:** C:\Users\prlangad\IdeaProjects\DataAnalysis_v3_Maven\data_default_user\current
 - WINUTILS.exe location:** C:\winutils\bin\winutils.exe
 - Enable parallel execution:** Unchecked.
- Sidebar:** Project, Z-Structure, Azure Explorer, External Libraries, Scratches and Consoles.

Synapse Notebook: Connect to AML workspace

The screenshot shows the Azure Synapse Notebook interface. The left sidebar lists resources under 'Develop': SQL scripts, Notebooks (selected), Data flows, Spark job definitions, and Power BI. The main area displays a notebook cell titled 'Check the Azure ML Core SDK Version to Validate Your Installation'. The code in Cell 3 is:

```
[5] 1 import azureml.core  
2 print("SDK Version:", azureml.core.VERSION)
```

The output shows the command was executed in 1s 258ms by balapv on 11-12-2019 14:41:52.805 -08:00, and the output is 'SDK Version: 1.0.69'.

Below this, a section titled 'Connect to Azure Workspace' is shown. Cell 5 contains the following code:

```
[6] 1 ## Import the Workspace class and check the Azure ML SDK version.  
2 from azureml.core import Workspace  
3  
4 ws = Workspace(subscription_id = "6560575d-fa06-4e7d-95fb-f962e74efd7a",  
5 | | | | | resource_group = "balapv-synapse-rg", workspace_name = "AML-WS-synapse")  
6  
7 print(ws.name, ws.location, ws.resource_group, sep='\t')
```

The output shows the command was executed in 3s 909ms by balapv on 11-12-2019 14:41:55.491 -08:00, and the output is 'AML-WS-synapse westus2 balapv-synapse-rg'.

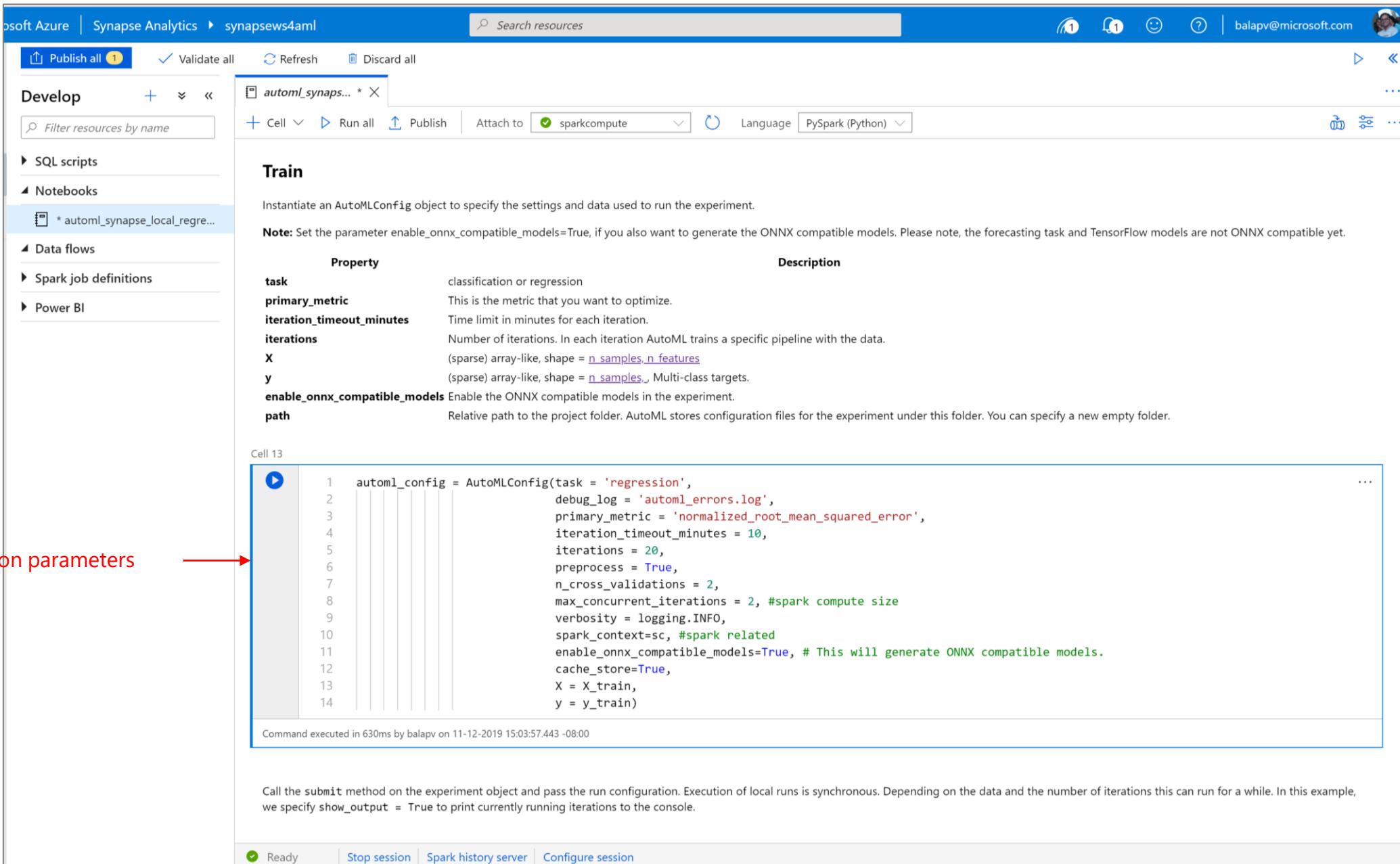
Cell 6 contains the following code:

```
[7] 1 # import modules  
2 import azureml.core  
3 import pandas as pd  
4 from azureml.core.authentication import ServicePrincipalAuthentication  
5 from azureml.core.workspace import Workspace  
6 from azureml.core.experiment import Experiment
```

At the bottom of the notebook, there are buttons for 'Running', 'Stop session', 'Spark history server', and 'Configure session'.

A red arrow points from the text 'Simple code to connect workspace' to the code in Cell 5.

Synapse Notebook: Configure AML job to run on Synapse



The screenshot shows the Azure Synapse Analytics notebook interface. On the left, the sidebar lists resources under 'Develop': SQL scripts, Notebooks (with 'automl_synapse_local_regr...' selected), Data flows, Spark job definitions, and Power BI. The main area displays the 'Train' section of the 'automl_synapse_local_regr...' notebook. It includes a table of configuration parameters with their descriptions and a code cell showing the Python code for creating an AutoMLConfig object.

Configuration parameters

```

1 automl_config = AutoMLConfig(task = 'regression',
2                               debug_log = 'automl_errors.log',
3                               primary_metric = 'normalized_root_mean_squared_error',
4                               iteration_timeout_minutes = 10,
5                               iterations = 20,
6                               preprocess = True,
7                               n_cross_validations = 2,
8                               max_concurrent_iterations = 2, #spark compute size
9                               verbosity = logging.INFO,
10                              spark_context=sc, #spark related
11                              enable_onnx_compatible_models=True, # This will generate ONNX compatible models.
12                              cache_store=True,
13                              X = X_train,
14                              y = y_train)

```

Call the `submit` method on the experiment object and pass the run configuration. Execution of local runs is synchronous. Depending on the data and the number of iterations this can run for a while. In this example, we specify `show_output = True` to print currently running iterations to the console.

Bottom navigation bar: Ready, Stop session, Spark history server, Configure session.

Synapse Notebook: Run AML job

The screenshot shows the Azure Synapse Notebook interface. The top navigation bar includes 'Microsoft Azure | Synapse Analytics > synapsews4aml', a search bar, 'Show notifications' with a bell icon, and a user profile for 'balapv@microsoft.com'. The left sidebar under 'Develop' lists resources: SQL scripts, Notebooks, Data flows, Spark job definitions, and Power BI. The main workspace is titled 'automl_synapse_local_regression' and shows a 'Run AutoML job' section. Cell 15 contains the Python code:

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

. Below the code, it says 'Command executed in 12mins 34s 972ms by balapv on 11-12-2019 15:17:53.089 -08:00'. The output shows the progress of the AutoML experiment, including iterations, pipelines evaluated, duration, metric values, and best scores. A red arrow points from the text 'ML job execution result' to the output table.

ML job execution result

ITERATION	PIPELINE	DURATION	METRIC	BEST
1	StandardScalerWrapper ElasticNet	0:00:38	0.0021	0.0021
2	StandardScalerWrapper ElasticNet	0:00:32	0.0054	0.0021
0	StandardScalerWrapper ElasticNet	0:01:20	0.0004	0.0004
4	StandardScalerWrapper RandomForest	0:00:33	0.0179	0.0004
3	StandardScalerWrapper ElasticNet	0:00:36	0.0036	0.0004
5	StandardScalerWrapper LightGBM	0:00:28	0.0109	0.0004
6	MaxAbsScaler DecisionTree	0:00:34	0.0168	0.0004
7	MaxAbsScaler RandomForest	0:00:41	0.0104	0.0004
8	MaxAbsScaler DecisionTree	0:01:05	0.0077	0.0004
9	MaxAbsScaler DecisionTree	0:00:48	0.0086	0.0004
10	StandardScalerWrapper DecisionTree	0:00:39	0.0058	0.0004
11	MaxAbsScaler DecisionTree	0:00:45	0.0096	0.0004
13	MaxAbsScaler ExtremeRandomTrees	0:00:47	0.0147	0.0004
12	MaxAbsScaler ExtremeRandomTrees	0:01:54	0.0096	0.0004
14	StandardScalerWrapper ElasticNet	0:00:39	0.0027	0.0004
15	StandardScalerWrapper ElasticNet	0:00:54	0.0010	0.0004
16	StandardScalerWrapper ElasticNet	0:00:48	0.0023	0.0004
17	MaxAbsScaler ElasticNet	0:00:31	0.0239	0.0004
18	StandardScalerWrapper ElasticNet	0:00:53	0.0014	0.0004
19	VotingEnsemble	0:01:59	0.0004	0.0004

Get Azure Portal URL for Monitoring Runs

Running | Stop session | Spark history server | Configure session

Predict

Overview

It provides ability to import existing machine learning models and score them within provisioned SQL. It takes ONNX (Open Neural Network Exchange) and data as inputs and generates prediction based on model.

Benefits

1. It empowers data engineers to successfully deploy machine learning models with the familiar T-SQL interface
2. It offers seamless collaboration with data scientists
3. It generates new columns, but the number of columns and their data types depends on the type of model that was used for prediction.

Syntax:

```
PREDICT
(
    MODEL = @model | model_literal,
    DATA = object AS <table_alias>
)
WITH ( <result_set_definition> )
<result_set_definition> ::= 
{
    { column_name
        data_type
    }
    [,...n]
}
MODEL = @model | model_literal
```

Example:

```
DECLARE @model varbinary(max) = (SELECT Model FROM Models WHERE Id = <>);
SELECT d.*, p.Score
FROM PREDICT(MODEL = @model,
    DATA = dbo.mytable AS d) WITH (Score float) AS p;
```

Monitor Hub - Spark applications

Overview

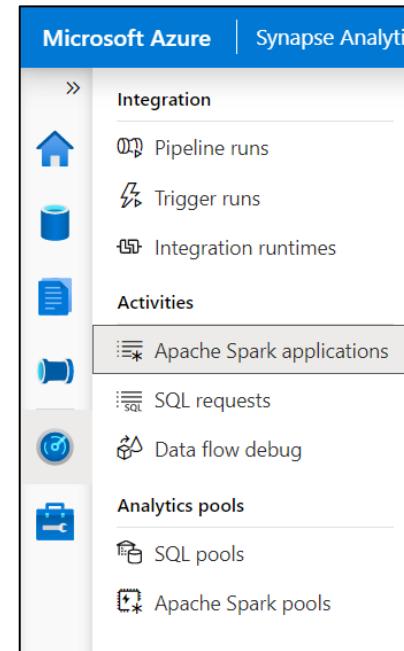
Monitor Spark pools, Spark applications for the status of activities

Benefits

Apply filter for pool to get Apache Spark Applications per pool

Additional available filters include

1. Application Name
2. Livy ID
3. Status
4. End time



Application name	Submitter	Submit time	Status	Pool
Synapse_automlpool_1...	negust@microsoft.com	11/30/20, 7:23:26 PM	Stopped	automlpool
Synapse_automlpool_1...	negust@microsoft.com	11/30/20, 7:14:00 PM	Stopped	automlpool
Synapse_automlpool_1...	negust@microsoft.com	11/30/20, 5:09:00 PM	Stopped (session timed ou...	automlpool
Notebook 3_analyticscp...	negust@microsoft.com	11/30/20, 12:52:51 PM	Stopped	analyticspool
Notebook 4_analytics1...	prlangad@microsoft.com	11/24/20, 6:29:41 PM	Stopped	analytics1
Synapse_hbpool_16059...	charlesf@microsoft.com	11/20/20, 2:30:44 PM	Stopped	hbpool
Synapse_analyticspool_...	negust@microsoft.com	11/20/20, 11:20:30 AM	Stopped	analyticspool



Azure Synapse Analytics

Synapse (SQL & Spark) serverless SQL pool

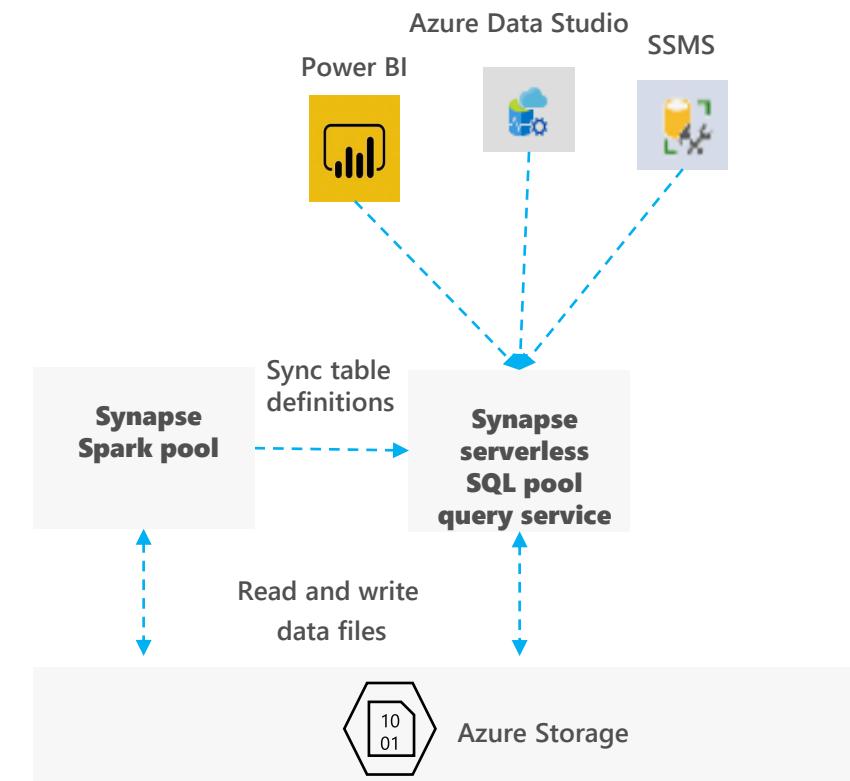
serverless SQL pool

Overview

An interactive query service that enables you to use standard T-SQL queries over files in Azure storage.

Benefits

- Use SQL to work with files on Azure storage
 - Directly query files on Azure storage using T-SQL
 - Logical Data Warehouse on top of Azure storage
 - Easy data transformation of Azure storage files
- Supports any tool or library that uses T-SQL to query data
- Automatically synchronize tables from Spark
- Serverless
 - No infrastructure, no upfront cost, no resource reservation
 - Pay only for query execution (per data processed)



Recommended usage scenarios

Quick data exploration

- Easily explore schema and data in files on Azure storage
- Supports various file formats (Parquet, CSV, JSON)
- Direct connector to Azure storage for large BI ecosystem

Logical Data Warehouse

Model raw files as virtual tables and views

Use any tool that works with SQL to analyze files

Use enterprise-grade security model

Easy data transformation

Transform CSV to parquet format

Move data between containers and accounts

Save the results of queries on external storage

Easily explore files on storage

The screenshot illustrates the integration of Azure Storage and Azure Synapse Analytics. On the left, the Azure portal navigation bar shows 'Microsoft Azure | Synapse Analytics > internalsandboxwe5'. The main area is divided into two panes:

- Left Pane (File Explorer):** Shows the 'Data' section with 'Storage accounts' (internalsandboxwe), 'Databases' (3), and 'Datasets' (5). The 'opendataset' folder under 'Storage accounts' contains several files, including '_SUCCESS', 'part-00000...', 'part-00001...', 'part-00002...', and 'part-00003...'. A context menu is open over the 'New SQL script - Select TOP 100 rows' file.
- Right Pane (Query Editor):** Displays a 'SQL script 1' tab with the following T-SQL code:


```

1 SELECT
2     TOP 100 *
3     FROM
4     OPENROWSET(
5         BULK 'https://internalsandboxwe.dfs.core.windows.net/opendataset/holidays/part-0001-bd1aba93-a85a-4909-8bf4-f79afb6c946f-c000.snappy.parquet'
6         FORMAT='PARQUET'
7     ) AS [r];
      
```

The 'Connect to' dropdown is set to 'SQL on-demand'. Below the code, the 'Results' tab shows a table of data with columns: VENDORID, TPEPICKUPDATETIME, TPEPDROPOFFDATETIME, PASSENGERCOUNT, TRIPDISTANCE, PULOCATIONID, and DOLOCATIONID. The results table displays four rows of taxi trip data from New York City.

Easily query files in various formats

Overview

Use OPENROWSET function to access data stored in various file formats

Benefits

Enables you to read CSV, parquet, Delta and JSON files

Provides unified T-SQL interface for all file types

Use standard SQL language to transform and analyze returned data

- Use JSON functions to get the data from underlying files.
- Use JSON functions to get data from PARQUET nested types

```
SELECT TOP 10 *
FROM OPENROWSET(
    BULK 'https://XYZ.blob.core.windows.net/csv/taxi/*.csv',
    FORMAT = 'CSV')
WITH (
    country_code VARCHAR(4),
    country_name VARCHAR(50),
    year INT,
    population INT
) AS nyc
```

```
SELECT TOP 10 *
FROM OPENROWSET(
    BULK 'https://XYZ.blob.core.windows.net/csv/taxi/*.parquet',
    FORMAT = 'PARQUET') AS nyc
```

```
SELECT TOP 10 *
    JSON_VALUE(jsonContent, '$.countryCode') AS country_code,
    JSON_VALUE(jsonContent, '$.countryName') AS country_name,
    JSON_VALUE(jsonContent, '$.year') AS year
    JSON_VALUE(jsonContent, '$.population') AS population
FROM OPENROWSET(
    BULK 'https://XYZ.blob.core.windows.net/json/taxi/*.json',
    FORMAT='CSV',
    FIELDTERMINATOR = '0x0b',
    FIELDQUOTE = '0x0b',
    ROWTERMINATOR = '0x0b'
)
WITH ( jsonContent varchar(MAX) ) AS json_line
```

	country_code	country_name	year	population
1	LU	Luxembourg	2017	594130

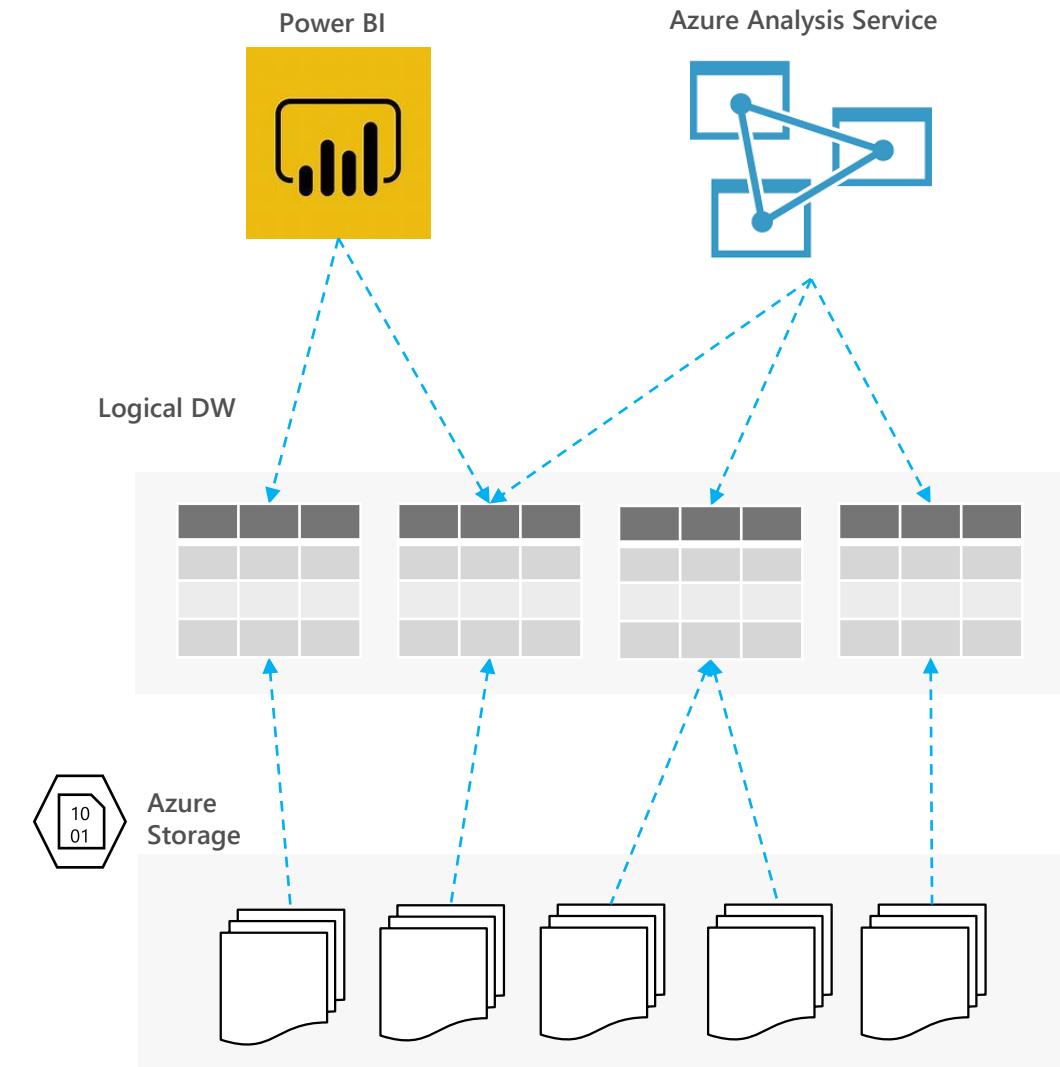
Synapse serverless SQL pool as a logical data warehouse

Overview

Logical relational layer on top of physical files in Azure Storage.

Benefits

- Abstract physical storage and file formats using well understandable relational concepts such as tables and views.
- Direct connector to Azure storage for large ecosystem of BI tools
- BI tools that use SQL can work with files on storage
 - Analytic tools use external tables that represent proxy to actual files.
 - No need for custom connectors in BI tools.
- Provides complex data processing (joining and aggregation) on top of raw files.
- Apply enterprise-ready security model and access control using battle-tested SQL Server permission model on top of Azure storage files



Logical Data Warehouse views

Overview

serverless SQL pool logical data warehouse views are created on external files placed in customer Azure storage

Benefits

Create SQL views on externally stored data

Access files using the view from various tools and language

Leverage rich T-SQL language to process and analyze data in external files exposed via views

Create PowerBI reports on the views created on external data

```
USE [mydbname]
GO

DROP VIEW IF EXISTS populationView
GO

CREATE VIEW populationView AS
SELECT *
FROM OPENROWSET(
    BULK 'https://XYZ.blob.core.windows.net/csv/population/*.csv',
    FORMAT = 'CSV',
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n'
)
WITH (
    [country_code] VARCHAR (5),
    [country_name] VARCHAR (100),
    [year] smallint,
    [population] bigint
) AS [r]
```

```
SELECT
    country_name, population
FROM populationView
WHERE
    [year] = 2019
ORDER BY
    [population] DESC
```

	country_name	population
1	China	1389618778
2	India	1311559204
3	United States	331883986
4	Indonesia	264935824
5	Pakistan	210797836
6	Brazil	210301591
7	Nigeria	208679114
8	Bangladesh	161062905
9	Russia	141944641
10	Mexico	127318112

Creating views

The screenshot illustrates the process of creating a view in Azure Synapse Analytics using the Serverless SQL pool.

Top Left: The main Azure Synapse Analytics interface shows the "internalsandbox..." workspace. The "Data" section is selected, displaying storage accounts, databases, and datasets. A "Develop" tab is open, showing the code for creating a view named "yellow_2017". The code uses OPENROWSET to bulk load data from a Parquet file located at https://internalsandboxwe.dfs.core.windows.net/opendataset/nyctlc/yellow/puYear=2017/*/*.

```

CREATE VIEW yellow_2017 AS
Select *
FROM
OPENROWSET(
    BULK 'https://internalsandboxwe.dfs.core.windows.net/opendataset/nyctlc/yellow/puYear=2017/\*/\*',
    FORMAT='PARQUET'
) AS [];

```

Bottom Left: A separate "Results" tab shows the execution of the same query. The results table displays the count of passengers per year (2017) and the total count (CNT).

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

Right Side: Another "Develop" tab shows the same view creation code. Below it, the "Results" tab displays the chart of passenger counts by year. The chart has "passengerCount" on the x-axis (0 to 10) and "cnt" on the y-axis (0 to 100M). The data shows a sharp peak at 1 passenger in 2017, followed by a decline and then a small rise at higher passenger counts.

Legend:

- PassengerCount (Blue line with dots)
- CNT (Green line with squares)

Chart Configuration:

- Chart type: Line
- Category column: (none)
- Legend (series) columns: Column 0, passengerCount, cnt
- Legend position: center - bottom
- Legend (series) label: (empty)

Logical Data Warehouse - tables

Overview

Create external tables that reference external files in your serverless SQL pool logical data warehouse

Benefits

Create external tables that reference set of files on Azure storage.

Join and transform multiple tables in the same query.

Enables you to analyze external files with the same experience that you have in classic databases.

Manage column statistics in external tables.

Manage access rights per table.

Create PowerBI reports on the views created on external data

```
USE [mydbname]
```

```
GO
```

```
DROP TABLE IF EXISTS dbo.Population
```

```
GO
```

```
CREATE EXTERNAL TABLE dbo.Population (
```

```
country_code VARCHAR (5) COLLATE Latin1_General_BIN2,  
country_name VARCHAR (100) COLLATE Latin1_General_BIN2,  
year smallint,  
population bigint
```

```
)
```

```
WITH(
```

```
LOCATION = '/csv/population/population-* .csv',
```

```
DATA_SOURCE = MyAzureStorage,
```

```
FILE_FORMAT = MyAzureCSVFormat
```

```
)
```

```
CREATE STATISTICS stat_country_name  
ON dbo.Population(country_name);
```

```
SELECT
```

```
country_name, population
```

```
FROM population
```

```
WHERE year = 2019
```

```
ORDER BY population DESC
```

	country_name	population
1	China	1389618778
2	India	1311559204
3	United States	331883986
4	Indonesia	264935824
5	Pakistan	210797836
6	Brazil	210301591
7	Nigeria	208679114
8	Bangladesh	161062905
9	Russia	141944641
10	Mexico	127318112

Easy data transformation

Overview

Easily perform data transformations of Azure Storage files using SQL queries

Optimize data pipeline - achieve more using serverless SQL pool

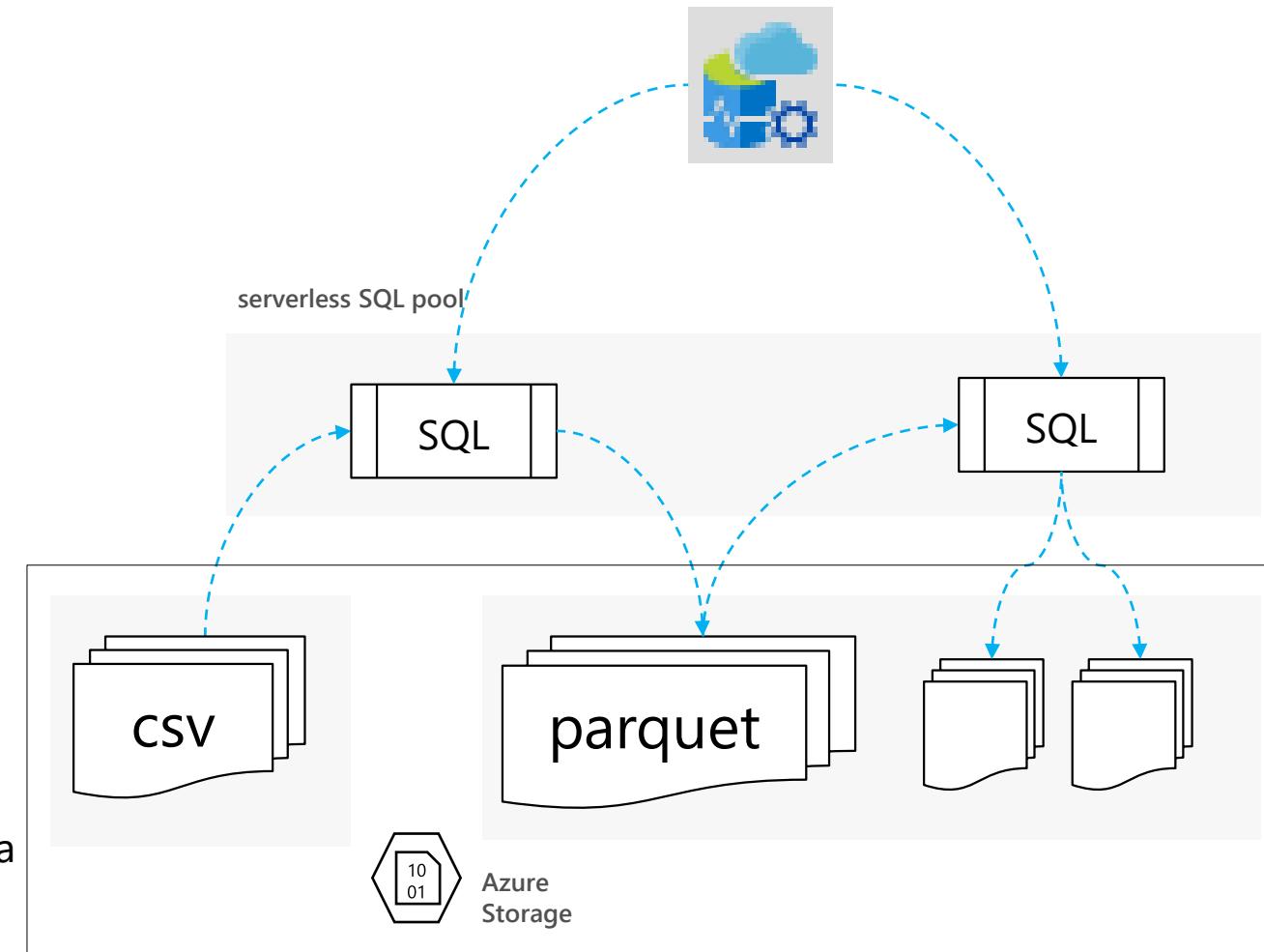
Benefits

Single statement transformations:

- convert CSV or JSON files to Parquet
- copy files from one storage account to another
- re-partition data to new location(s)
- store results of your query on Azure Storage

SQL ETL pipelines

- Use SQL commands to transform data
- Chain SQL statement for build ETL process
- Materialize reports created on the current snapshot of data



Automatic syncing of Spark tables

Overview

Tables created in Spark pool are automatically created as external tables that reference external files in your serverless SQL pool logical data warehouse

Benefits

Tables designed using Spark languages are immediately available in serverless SQL pool.

Schema definition matches original

Spark table updates are applied in serverless SQL pool

No need to manually create SQL tables that match Spark tables

Spark and serverless SQL pool tables references the same external files.

The screenshot shows the Azure Synapse Analytics studio environment. On the left is a dark sidebar with various icons. The main area has two panes: a top pane titled "Create external table" containing a code editor with the following SQL:

```

%%sql
create table data1017 using parquet
location 'abfss://container@demostorage.dfs.core.windows.net/data/'

```

Below this is a bottom pane titled "SQLQuery_1 - sqlikon...oud!SA" showing a running query:

```

SELECT TOP (10) [ExtractId]
,[DayOfWeekID]
,[DayOfWeekDescr]
,[DayOfWeekDescrShort]
,[ExtractDateTime]
,[LoadTS]
,[DeltaActionCode]
FROM [default]..[data1017]

```

The results pane shows the following data:

	ExtractId	DayOfWeekID	DayOfWeekDescr	DayOfWeekDescrShort	ExtractDateT
1	6b86b273ff34fce19d6b804eff5a...	1	Sunday	Sun	2020-01-22 00:00:00
2	d4735e3a265e16eee03f5a718h9b...	2	Monday	Mon	2020-01-22 00:00:00
3	4e07408562bedb8b60ce05c1uect...	3	Tuesday	Tue	2020-01-22 00:00:00
4	4b22777d4dd1fc61c6f884f4864...	4	Wednesday	Wed	2020-01-22 00:00:00
5	ef2d127de37b942baad06145e54b...	5	Thursday	Thu	2020-01-22 00:00:00
6	e7f6c011776e8db7cd330b54174f...	6	Friday	Fri	2020-01-22 00:00:00

Logical Data Warehouse - tables

Overview

Create external tables that reference external files in your serverless SQL pool logical data warehouse

Benefits

Create external tables that reference set of files on Azure storage.

Join and transform multiple tables in the same query.

Enables you to analyze external files with the same experience that you have in classic databases.

Manage column statistics in external tables.

Manage access rights per table.

Create PowerBI reports on the views created on external data

```
USE [mydbname]
GO

DROP TABLE IF EXISTS dbo.Population
GO

CREATE EXTERNAL TABLE dbo.Population (
    country_code VARCHAR (5) COLLATE Latin1_General_BIN2,
    country_name VARCHAR (100) COLLATE Latin1_General_BIN2,
    year smallint,
    population bigint
)
WITH(
    LOCATION = '/csv/population/population-* .csv',
    DATA_SOURCE = MyAzureStorage,
    FILE_FORMAT = MyAzureCSVFormat
)
```

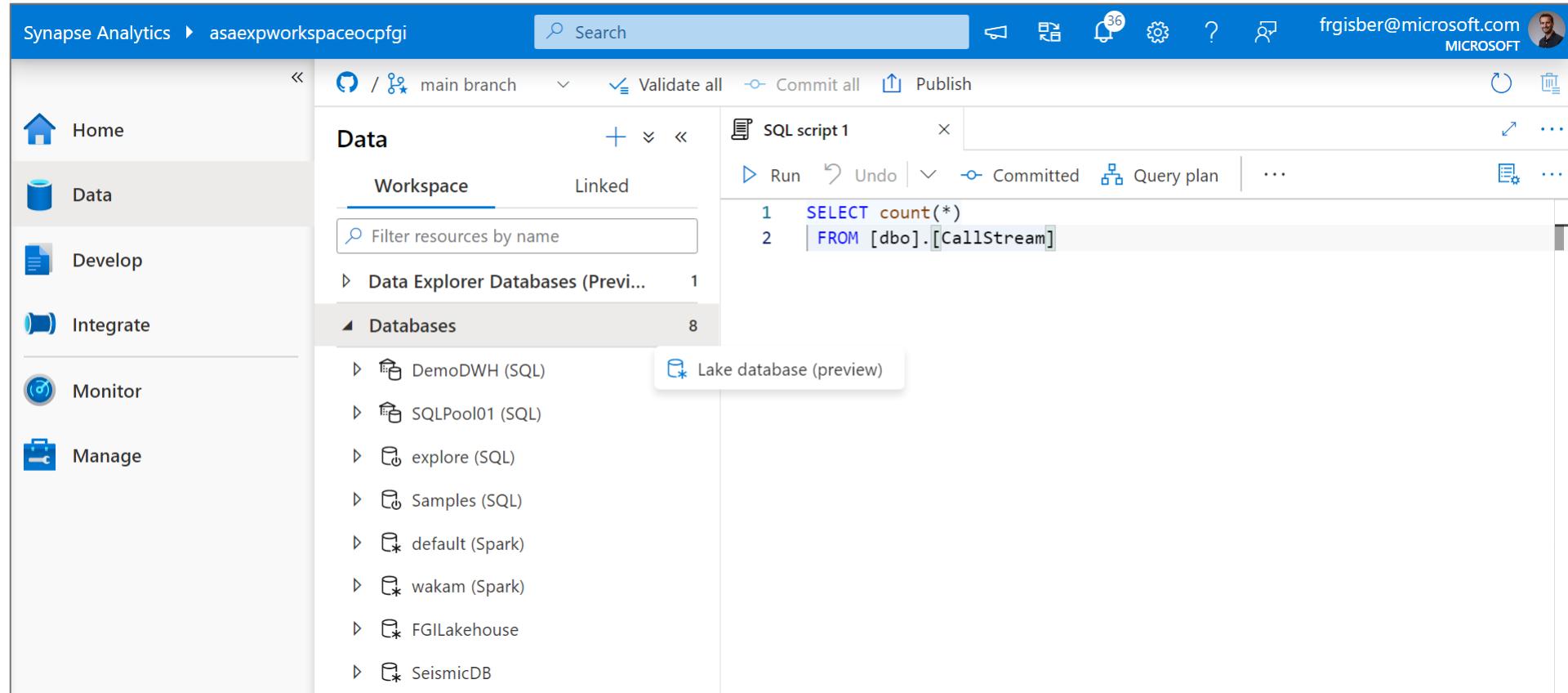
```
CREATE STATISTICS stat_country_name
ON dbo.Population(country_name);
```

```
SELECT
    country_name, population
FROM population
WHERE year = 2019
ORDER BY population DESC
```

	country_name	population
1	China	1389618778
2	India	1311559204
3	United States	331883986
4	Indonesia	264935824
5	Pakistan	210797836
6	Brazil	210301591
7	Nigeria	208679114
8	Bangladesh	161062905
9	Russia	141944641
10	Mexico	127318112

Lakehouse Database Engine

New Lakehouse Database capabilities based on Synapse Spark



Data Lakehouse database

Based on entities templates or custom design

The screenshot shows the 'Add from template' interface in Microsoft Synapse Analytics. The top navigation bar includes 'Synapse Analytics', a search bar, and user information for 'frgisber@microsoft.com MICROSOFT'. On the left, there's a sidebar with icons for Home, Databases, Workspaces, and Jupyter Notebooks. The main area is titled 'Add from template' under 'Database 1'.

The interface displays a grid of 16 entity templates, each with an icon and a brief description:

- Agriculture**: For companies engaged in growing crops, raising livestock and dairy production.
- Automotive**: For companies manufacturing automobiles, heavy vehicles, tires, and other automotive components.
- Banking**: For companies who are analyzing banking data.
- Consumer Goods**: For manufacturers or producers of goods bought and used by consumers.
- Energy & Commodity Trading**: For traders of energy, commodities, or carbon credits.
- Freight & Logistics**: For companies providing freight and logistics services.
- Fund Management**: For companies managing investment funds on behalf of investors.
- Genomics**: For companies acquiring and analyzing genomic data about human beings or other species.
- Life Insurance & Annuities**: For companies who provide life insurance, sell annuities, or both.
- Manufacturing**: For companies engaged in discrete manufacturing of a wide range of products.
- Oil & Gas**: For companies involved in various phases of the Oil & Gas value chain.
- Pharmaceuticals**: For companies engaged in creating, manufacturing, and marketing pharmaceutical and bio-pharmaceutical products and medical devices.
- Property & Casualty Insurance**: For companies who provide insurance against risks to property and various forms of liability coverage.
- R&D and Clinical Trials**: For companies involved in research and development and clinical trials of pharmaceutical products or devices.
- Retail**: For sellers of consumer goods or services to customers through multiple channels.
- Utilities**: For gas, electric and water utilities and power generators and water desalination.

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

Microsoft Azure | Synapse Analytics > asaexpworkspaceocpfgi

main branch | Validate all | Commit all | Publish | Search

Data | SQL script 1 | Database 1 | SeismicDB

Table | Committed

Tables

- Filter resources by name
- Data Explorer Databases (Prev...)** 1
 - kustopool (kustopool)
- Databases** 9
 - Database 1
 - DemoDWH (SQL)
 - SQLPool01 (SQL)
 - explore (SQL)
 - Samples (SQL)
 - default (Spark)
 - wakam (Spark)
 - FGILakehouse
- SeismicDB**
 - Tables**
 - SeismicChannelType
 - SeismicDataAcquisitionEvent
 - SeismicShot
 - ShotChannel
 - ShotFile

SeismicShot

- SeismicShotId PK
- SeismicShotTimestamp
- SeismicShotEnergySourceA...
- SeismicShotLocationId
- GeographicAreaId
- GeographicAreaPolygonVer...
- SeismicShotSourceArrayTy...
- SeismicShotRecordingPolar...
- SeismicShotSeismicEnergyT...
- SeismicShotSeismicSensorT...

SeismicData AcquisitionEvent

- SeismicDataAcquisitionEve... PK
- SeismicDataAcquisitionEve...
- SeismicDataAcquisitionEve...
- SeismicDataAcquisitionEve...
- SeismicDataAcquisitionEve...
- SurveyMethodTypeId
- NumberOfChannels

ShotFile

- SeismicDataFileId PK
- SeismicShotId PK,FK
- ShotFileNote

ShotChannel

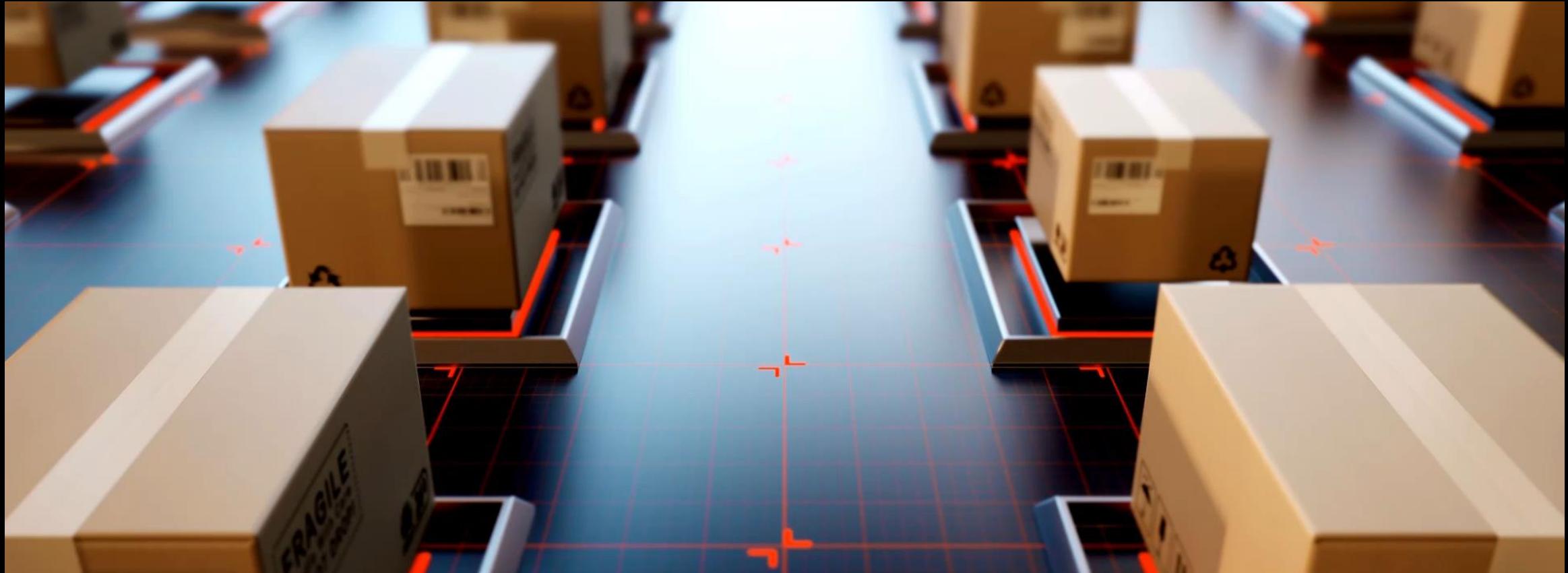
- SeismicShotId PK,FK
- ShotChannelId PK
- SeismicDataFileId PK,FK
- PeriodStartTimeStamp PK
- PeriodEndTimeStamp
- SeismicChannelTypeId FK
- ShotChannelNote

General Columns Relationships

Filter by keyword | + Column | Clone | Delete

Name	Keys	Description	Nullability	Data type	Format / Length
SeismicDataFileId	PK	The unique identifier of a seismic data file.	Null	integer	1024
SeismicShotId	PK, FK	The unique identifier of a seismic shot.	Null	integer	1024
ShotFileNote	PK	A note, comment or additional information regarding the shot file.	Null	string	1024

Demo





Enregistrez vous dès maintenant au prochain Webinars Data AI

Event Webinar (Les jeudis de la Data & AI) - L200/300	Date	Duration (min)	Link
Azure Synapse	22/09/2022	120	https://msevents.microsoft.com/event?id=857781749
Les solutions SQL dans Azure (PaaS, IaaS, SaaS)	29/09/2022	120	https://msevents.microsoft.com/event?id=502366997
Déploiement et sécurisation des workspaces Azure Machine learning	06/10/2022	120	https://msevents.microsoft.com/event?id=1505714138
Azure Scale Analytics - Architectures Data Mesh dans Azure avec Azure Synapse, Microsoft Purview et Azure Data Share	13/10/2022	120	https://msevents.microsoft.com/event?id=139685175
MLOps avec Azure Machine Learning	20/10/2022	120	https://msevents.microsoft.com/event?id=1245885767
SQL Server 2022 et hybridation native avec Azure SQL Managed Instance	10/11/2022	120	https://msevents.microsoft.com/event?id=145826476
Machine Learning dans Azure Synapse Analytics	17/11/2022	120	https://msevents.microsoft.com/event?id=3637723312
Azure Cosmos DB et IA	24/11/2022	120	https://msevents.microsoft.com/event?id=2646013445
Azure et les Services Cognitifs	08/12/2022	120	https://msevents.microsoft.com/event?id=3772037220
La gouvernance de données dans Azure avec Microsoft Purview	15/12/2022	120	https://msevents.microsoft.com/event?id=1499560981
MLOps avec Azure Machine Learning	12/01/2023	120	https://msevents.microsoft.com/event?id=4115194515
	19/01/2023	120	https://msevents.microsoft.com/event?id=1537241181
Data processing dans Azure ave Azure Synapse, Azure Batch, Spark, Notebook, etc.	26/01/2023	120	https://msevents.microsoft.com/event?id=1806467748
Déploiement et sécurisation des workspace Azure Synapse	09/02/2023	120	En cours
Azure Machine Learning pour les Citizen Data Scientists	16/02/2023	120	https://msevents.microsoft.com/event?id=1401519679
L'IA responsable avec Azure machine learning	09/03/2023	120	https://msevents.microsoft.com/event?id=2072953112
Machine Learning dans Azure Synapse Analytics	16/03/2023	120	https://msevents.microsoft.com/event?id=3413014857
Les bases de données Open Source dans le cloud Azure	23/03/2023	120	https://msevents.microsoft.com/event?id=2727487131
Hybridation des services de Machine Learning Azure	06/04/2023	120	https://msevents.microsoft.com/event?id=1624914222
La gouvernance de données dans Azure avec Microsoft Purview	13/04/2023	120	https://msevents.microsoft.com/event?id=3909342839
Les solutions SQL dans Azure (PaaS, IaaS, SaaS)	04/05/2023	120	https://msevents.microsoft.com/event?id=1162207895
	16/05/2023	120	https://msevents.microsoft.com/event?id=3517068442
Data processing dans Azure ave Azure Synapse, Azure Batch, Spark, Notebook, etc.	24/05/2023	120	https://msevents.microsoft.com/event?id=2996507398
Self Service Analytics	01/06/2023	120	En cours

END

