



Welcome to
What is new in Azure Synapse Analytics



Agenda

Morning

8:00-8:30	Welcome and what's new overview
8:30-9:00	Github integration and Azure DevOps for CI/CD
9:00-9:15	Azure Machine Learning integration
9:15-9:30	Lab environment overview
9:30-9:45	Short Break
9:45-10:45	Lab 01: Machine Learning integration
10:45-11:00	Azure Data Explorer integration
11:00-11:15	Spark Updates
11:15-12:15	Lab 02: Spark
12:15-12:45	Long Break

Afternoon

12:45-1:00	CDM
1:00-1:15	Purview
1:15-2:15	Lab 03: CDM
2:15-2:30	Short Break
2:30-2:45	Monitoring enhancements
2:45-3:00	Migrating existing data warehouses
3:00-3:15	Migrating ADF assets
3:15-3:45	Lab 04: Migrating data factory assets
3:45-4:00	Conclusion

■ Presentation

■ Lab

■ Break



Mentimeter Poll

Scan QR Code

or

Go to www.menti.com and use code

9044 6679

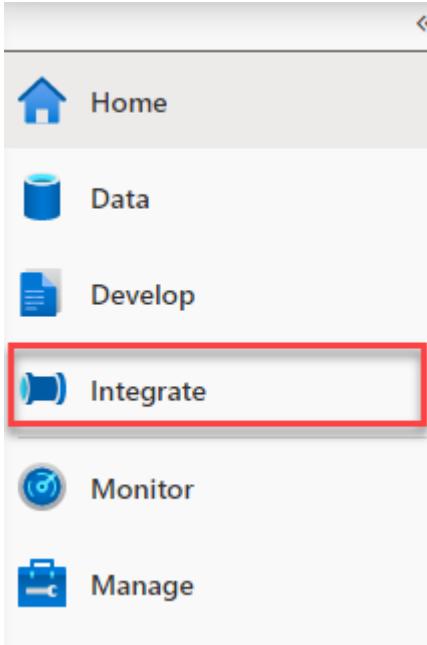




What's new in Azure Synapse Analytics overview

Synapse Studio Features - Naming

- Orchestrate is now Integrate
- Serverless pool renamed to “Built-in”
- SQL pool rebranding: serverless and dedicated
- Apache Spark in Azure Synapse Analytics



The screenshot shows the Azure Synapse Studio interface. On the left, there is a navigation sidebar with the following items: Home, Data, Develop, **Integrate** (which is highlighted with a red box), Monitor, and Manage. The main area is titled "SQL pools". It displays a message: "The serverless SQL pool, Built-in, is immediately available for your workspace. Dedicated SQL pools can be configured". Below this are buttons for "+ New", "Refresh", and "System-assigned managed identity". A "Filter by name" input field is present. The text "Showing 1-5 of 5 items (1 Serverless, 4 Dedicated)" is displayed. A table lists the SQL pools:

Name	Type	Status
Built-in	Serverless	Online
SQLPool01	Dedicated	Paused

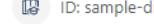
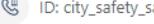
Synapse Studio Features – Knowledge Center

Gallery

Datasets Notebooks SQL scripts Pipelines

Filter by keyword

Tags : All

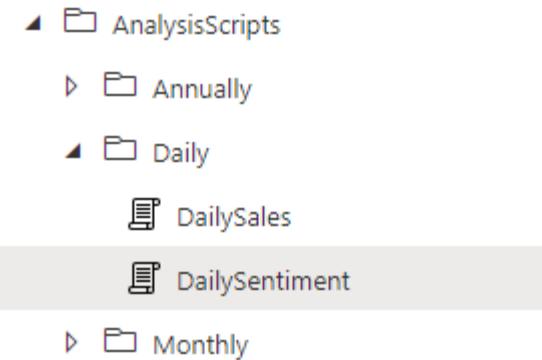
 Bing COVID-19 Data Bing COVID-19 data includes confirmed, fatal, and recovered cases from all regions, updated da... ID: bing-covid-19-data	 Boston Safety Data Read data about 311 calls reported to the city of Boston. This dataset is stored in Parquet format and is up...  ID: city_safety_boston	 COVID Tracking Project The COVID Tracking Project dataset provides the latest numbers on tests, confirmed cases, hospitalizat... ID: covid-tracking	 Chicago Safety Data Read data about 311 calls reported to the city of Chicago. This dataset is stored in Parquet format and is ...  ID: city_safety_chicago	 European Centre for Disease Prevention and Control (ECDC) Covid-19 Cases The latest available public data on... ID: ecdc-covid-19-cases	 NOAA Integrated Surface Data (ISD) NOAA Integrated Surface Data (ISD) provides Worldwide hourly weath...  ID: isd	 NYC Taxi & Limousine Commission - For-Hire Vehicle (FHV) trip records The For-Hire Vehicle trip records i...  ID: nyc_tlc_fhv
 NYC Taxi & Limousine Commission - green taxi trip records The green taxi trip records include...  ID: nyc_tlc_green	 NYC Taxi & Limousine Commission - yellow taxi trip records The yellow taxi trip records includ...  ID: nyc_tlc_yellow	 New York City Safety Data This dataset contains all New York City 311 service requests from 2010 to the present. It's stored in Pa...  ID: city_safety_newyork	 Oxford COVID-19 Government Response Tracker The Oxford Covid-19 Government Response Tracker (OxCGR) database... ID: oxford-covid-19-govern...	 Public Holidays Worldwide public holiday data sourced from PyPI holidays package and Wikipedia, covering 38 countr...  ID: public_holiday	 Sample: Diabetes The Diabetes dataset has 442 samples with 10 features, making it ideal for getting started with mac...  ID: sample-diabetes	 San Francisco Safety Data Fire department calls for service and 311 cases in San Francisco. This dataset contains historical records...  ID: city_safety_sanfrancisco
 Seattle Safety Data Seattle Fire Department 911 dispatches. This dataset is updated daily, and contains historical recor...  ID: city_safety_seattle	 US Consumer Price Index The Consumer Price Index (CPI) is a measure of the average change over time in the prices paid by urb...  ID: us-consumer-price-index	 US Labor Force Statistics US Labor Force Statistics provides Labor Force Statistics, labor force participation rates, and the civilian...  ID: us-labor-force-statistics	 US Local Area Unemployment Statistics The US Local Area Unemployment Statistics datasets provides month...  ID: us-local-area-unemploy...	 US National Employment Hours and Earnings The Current Employment Statistics (CES) program produces detailed i...  ID: us-employment-hours-e...	 US Population by County US population by gender and race for each US county sourced from 2000 and 2010 Decennial Census. ...  ID: us-decenrial-census-co...	 US Population by ZIP Code US population by gender and race for each US ZIP code sourced from 2010 Decennial Census. This datas...  ID: us-decenrial-census-zip

Continue

Close

Synapse Studio Features – File organization

Organize and group files into folders



Rename Synapse artifacts

The image shows two overlapping dialogs. The top dialog is the 'Properties' dialog for a notebook named 'Renamed Notebook'. The 'Name' field is highlighted with a red box. The bottom dialog is the 'Publish all' dialog, which displays pending changes. It shows a table with one item: a 'Notebook' named 'Renamed Notebook' with a status of '(Edited, Renamed)' and the original name 'Notebook 01'.

1

Properties

General Related (0)

Name

Description

Type .ipynb notebook

Size 683 bytes

Notebook settings Include cell output when saving

Session Configure session

2

Publish all

You are about to publish all pending changes to the live environment. [Learn more](#)

Pending changes (1)

NAME	CHANGE	EXISTING
Renamed Notebook	(Edited, Renamed)	Notebook 01

Synapse Studio Features - Security

Expanded roles (preview)

Add role assignment

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

Scope * ⓘ

Workspace Workspace item

Role * ⓘ

Select a role

Filter...

- Synapse Administrator ⓘ
- Synapse SQL Administrator ⓘ
- Synapse Apache Spark Administrator ⓘ
- Synapse Contributor (preview) ⓘ
- Synapse Artifact Publisher (preview) ⓘ
- Synapse Artifact User (preview) ⓘ
- Synapse Compute Operator (preview) ⓘ
- Synapse Credential User (preview) ⓘ

Scoped role assignment (preview)

Add role assignment

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

Scope * ⓘ

Workspace Workspace item

Item type *

Linked service

Item *

asadatalake01

Role * ⓘ

Synapse Credential User (preview)

Select user * ⓘ

Search by name or email address

Selected user(s), group(s), or service principal(s)

No users, groups, or apps selected.

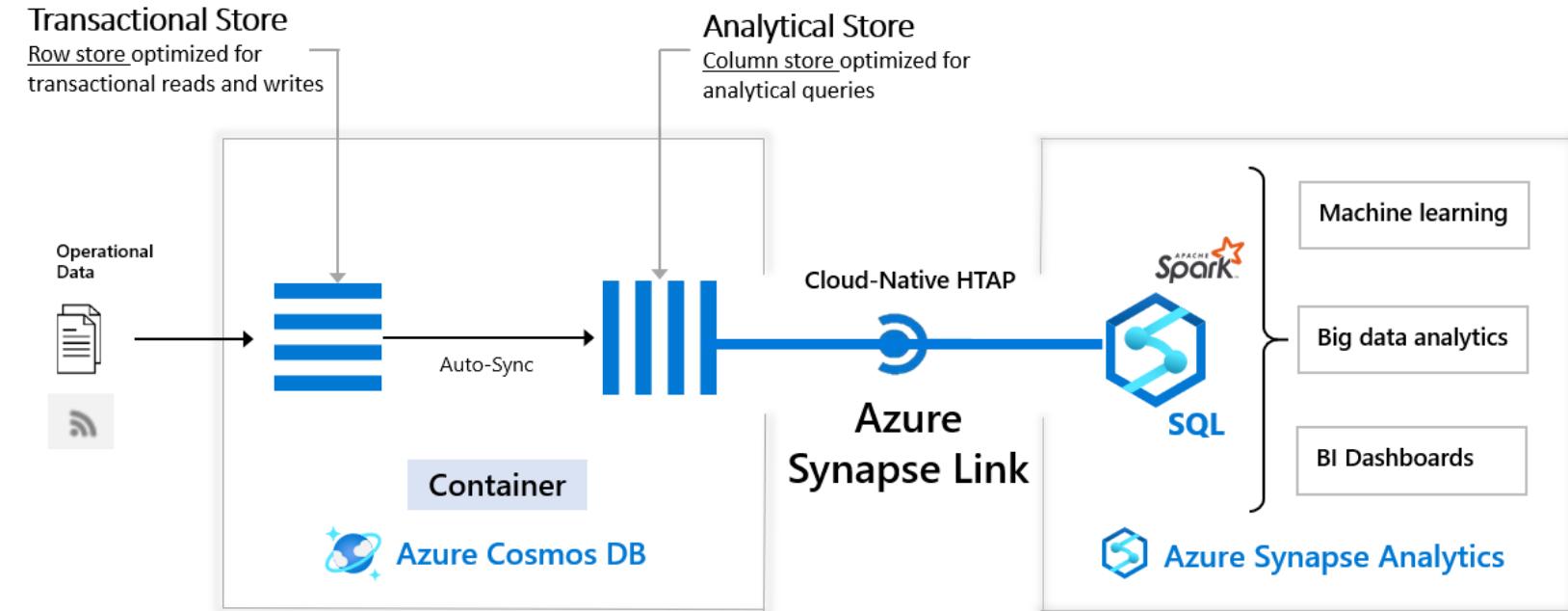


Synapse Studio Features – Security (cont.)

Action	Synapse Administrator	Synapse Contributor	Synapse Artifact Author	Synapse Artifact Reader	Synapse Compute Manager	Synapse Credential User	Synapse Managed Private Endpoint Administrator	Synapse Reader
workspaces/read	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
workspaces/roleAssignments/write, delete	Yes							
workspaces/managedPrivateEndpoint/write, delete	Yes							Yes
workspaces/bigDataPools/useCompute/action	Yes	Yes			Yes			
workspaces/bigDataPools/viewLogs/action	Yes	Yes			Yes			
workspaces/integrationRuntimes/useCompute/action	Yes	Yes			Yes			
workspaces/integrationRuntimes/viewLogs/action	Yes	Yes			Yes			
workspaces/artifacts/read	Yes	Yes	Yes	Yes				
workspaces/notebooks/write, delete	Yes	Yes	Yes					
workspaces/sparkJobDefinitions/write, delete	Yes	Yes	Yes					
workspaces/sqlScripts/write, delete	Yes	Yes	Yes					
workspaces/dataFlows/write, delete	Yes	Yes	Yes					
workspaces/pipelines/write, delete	Yes	Yes	Yes					
workspaces/triggers/write, delete	Yes	Yes	Yes					
workspaces/datasets/write, delete	Yes	Yes	Yes					
workspaces/libraries/write, delete	Yes	Yes	Yes					
workspaces/linkedServices/write, delete	Yes	Yes	Yes					
workspaces/credentials/write, delete	Yes	Yes	Yes					
workspaces/notebooks/viewOutputs/action	Yes	Yes						
workspaces/pipelines/viewOutputs/action	Yes	Yes						
workspaces/linkedServices/useSecret/action	Yes					Yes		
workspaces/credentials/useSecret/action	Yes					Yes		

Azure Synapse Link for Cosmos DB

- Low latency, non-impactful queries
- Support for the SQL API and the API for Mongo DB
- Multi-region support
- Integrates with serverless SQL Pool and Apache Spark



Enable Azure Synapse Link for Cosmos DB

Microsoft Azure

Search resources, services and docs

Home > Azure Cosmos DB > CosmosDB-account - Features

CosmosDB-account - Features

Microsoft

Search (Ctrl+I)

Refresh

Feature	Status
Synapse link	Off

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Quick Start

Notifications

Data Explorer

Settings

Replicate data globally

Default consistency

Backup & Restore

Keys

Features

Add Azure Search

Add Azure Function

Locks

Automation script

Synapse link

Feature

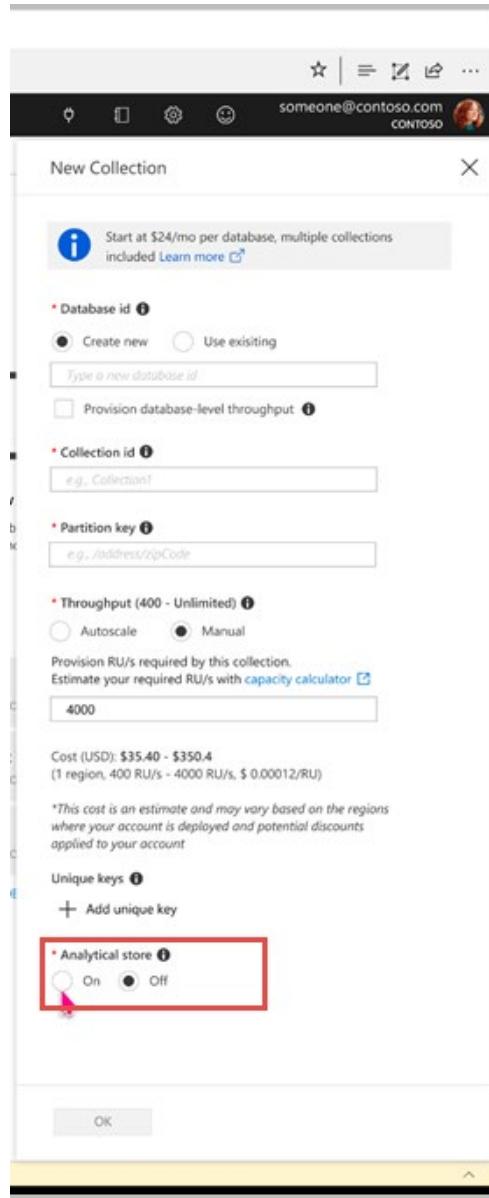
Synapse Link for Cosmos DB creates a tight integration between Azure Cosmos DB and Azure Synapse Analytics enabling customers to run near real-time analytics over their operational data with no-ETL and full performance isolation from their transactional workloads.

By combining the distributed scale of Cosmos DB's transactional processing with built-in analytical store and the computing power of Azure Synapse Analytics, Synapse Link enables Hybrid Transactional/Analytical Processing (HTAP) architectures for optimizing your business processes. This integration eliminates ETL processes, enabling business analysts, data engineers & data scientists to self-serve and run near real-time BI, analytics and ML pipelines over operational data.

Learn more

Enable Cancel

Create a container with analytical store



C#

```
// Create a container with a partition key, and analytical TTL configured to -1 (infinite retention)
ContainerProperties properties = new ContainerProperties()
{
    Id = "myContainerId",
    PartitionKeyPath = "/id",
    AnalyticalStoreTimeToLiveInSeconds = -1,
};
CosmosClient cosmosClient = new CosmosClient("myConnectionString");
await cosmosClient.GetDatabase("myDatabase").CreateContainerAsync(properties);
```

Java

```
// Create a container with a partition key and analytical TTL configured to -1 (infinite retention)
CosmosContainerProperties containerProperties = new CosmosContainerProperties("myContainer", "/myPartitionKey");

containerProperties.setAnalyticalStoreTimeToLiveInSeconds(-1);

container = database.createContainerIfNotExists(containerProperties, 400).block().getContainer();
```

Also with Python, Azure CLI, Powershell...



Analytical Store
Container



Transactional Store
Container

Create a Cosmos DB Linked Service

Microsoft Azure | Synapse Analytics > asaworkspace01

Synapse live Validate all Publish all

Data

Workspace   

SQL database

Linked

Databases

Connect to external data 

Integration dataset

Browse gallery

Connect to external data

Once a connection is created, the underlying data of that connection will be available for analysis in the Data hub or for pipeline activities in the Integrate hub.

Azure Blob Storage	Azure Cosmos DB (MongoDB API) 	Azure Cosmos DB (SQL API) 
--------------------	---	---

New linked service (Azure Cosmos DB (SQL API))

Choose a name for your linked service. This name cannot be updated later.

Name *

CosmosDb1

Description

Connect via integration runtime *

AutoResolveIntegrationRuntime

Connection string 

Account selection method 

From Azure subscription Enter manually

Azure subscription 

Synapse Analytics

Azure Cosmos DB account name * 

asacosmosdb01

Database name * 

[REDACTED]

Additional connection properties

+ New

Annotations

Query analytical Cosmos DB data from a serverless SQL pool

SQL

```
OPENROWSET(
    'CosmosDB',
    '<Azure Cosmos DB connection string>',
    <Container name>
) [ < with clause > ] AS alias
```

SQL

```
CREATE CREDENTIAL MyCosmosDbAccountCredential
WITH IDENTITY = 'SHARED ACCESS SIGNATURE', SECRET = 's5zarR2pT0JWH9k8roipnWxUYBegO
GO
CREATE OR ALTER VIEW Ecdc
AS SELECT *
FROM OPENROWSET(
    PROVIDER = 'CosmosDB',
    CONNECTION = 'Account=synapseslink-cosmosdb-sqlsample;Database=covid',
    OBJECT = 'Ecdc',
    SERVER_CREDENTIAL = 'MyCosmosDbAccountCredential'
) with ( date_rep varchar(20), cases bigint, geo_id varchar(6) ) as rows
```

SQL

```
SELECT
*
FROM
OPENROWSET(
    'CosmosDB',
    'Account=synapseslink-cosmosdb-sqlsample;Database=covid;Key=s5zarR2pT0JW
    Cord19
) WITH ( title varchar(1000) '$.metadata.title',
          authors varchar(max) '$.metadata.authors' ) AS docs
CROSS APPLY OPENJSON ( authors )
           WITH (
               first varchar(50),
               last varchar(50),
               affiliation nvarchar(max) as json
           ) AS a
```

alter database current collate Latin1_General_100_CI_AI_SC_UTF8



Query analytical Cosmos DB data from Spark

Python

```
# To select a preferred list of regions in a multi-region Azure Cosmos DB account

df = spark.read.format("cosmos.olap")\
    .option("spark.synapse.linkedService", "<enter linked service name>")\
    .option("spark.cosmos.container", "<enter container name>")\
    .load()
```

SQL

```
%%sql
-- To select a preferred list of regions in a multi-region Azure Cosmos DB account,
create table call_center using cosmos.olap options (
    spark.synapse.linkedService '<enter linked service name>',
    spark.cosmos.container '<enter container name>'
)
```

Serverless SQL pool enhancements

- Cost control
- Fully supported in SSMS 18.7
- Fully supported in ADS 1.24.0
- **STRING_AGG, OFFSET/FETCH, Pivot/Unpivot, SESSION_CONTEXT and CONTEXT_INFO** are now supported.
- Automatic schema discovery on delimited text files with parser_version 2.0



Mentimeter Poll

Scan QR Code

or

Go to www.menti.com and use code

2104 4036



Deep dives for today

GitHub and DevOps

Azure Machine Learning

Azure Data Explorer integration

Spark Updates

Monitoring enhancements

Migration from existing data warehouses

Migration from Azure Data Factory





GitHub integration and Azure DevOps for CI/CD

Source control in Azure Synapse Studio

Synapse live ▾ Validate all Publish all

Analytics pools SQL pools Apache Spark pools External connections Linked services Integration Triggers Integration runtimes Security Access control Credentials Managed private endpoints Source control **Git configuration**

Configure a repository

Connect your workspace with your Git repository just within few clicks. To learn more about best practices about CI/CD please

Setting Disconnect

No Git repository configured

Connect to a repository for source control and collaboration for work on your workspace pipelines.

Configure

Configure a repository

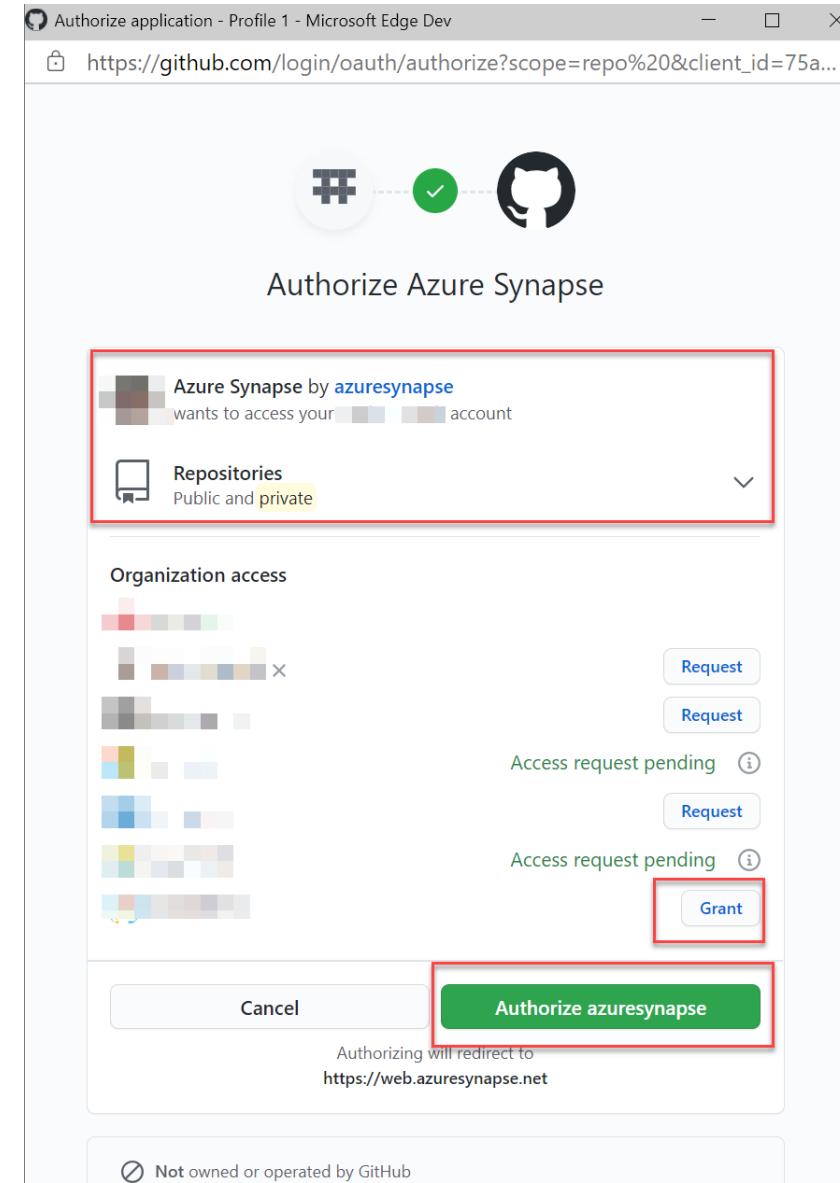
Specify the settings that you want to use when connecting to your repository.

Repository type *

Azure Active Directory

Continue Cancel

GitHub Authorization



Git Branches

Configure a repository

 synapseconnect

Specify the settings that you want to use when connecting to your repository.

Select repository Use repository link

Repository name * 

synapsesdev

Collaboration branch * 

main

Publish branch * 

workspace_publish

Root folder * 

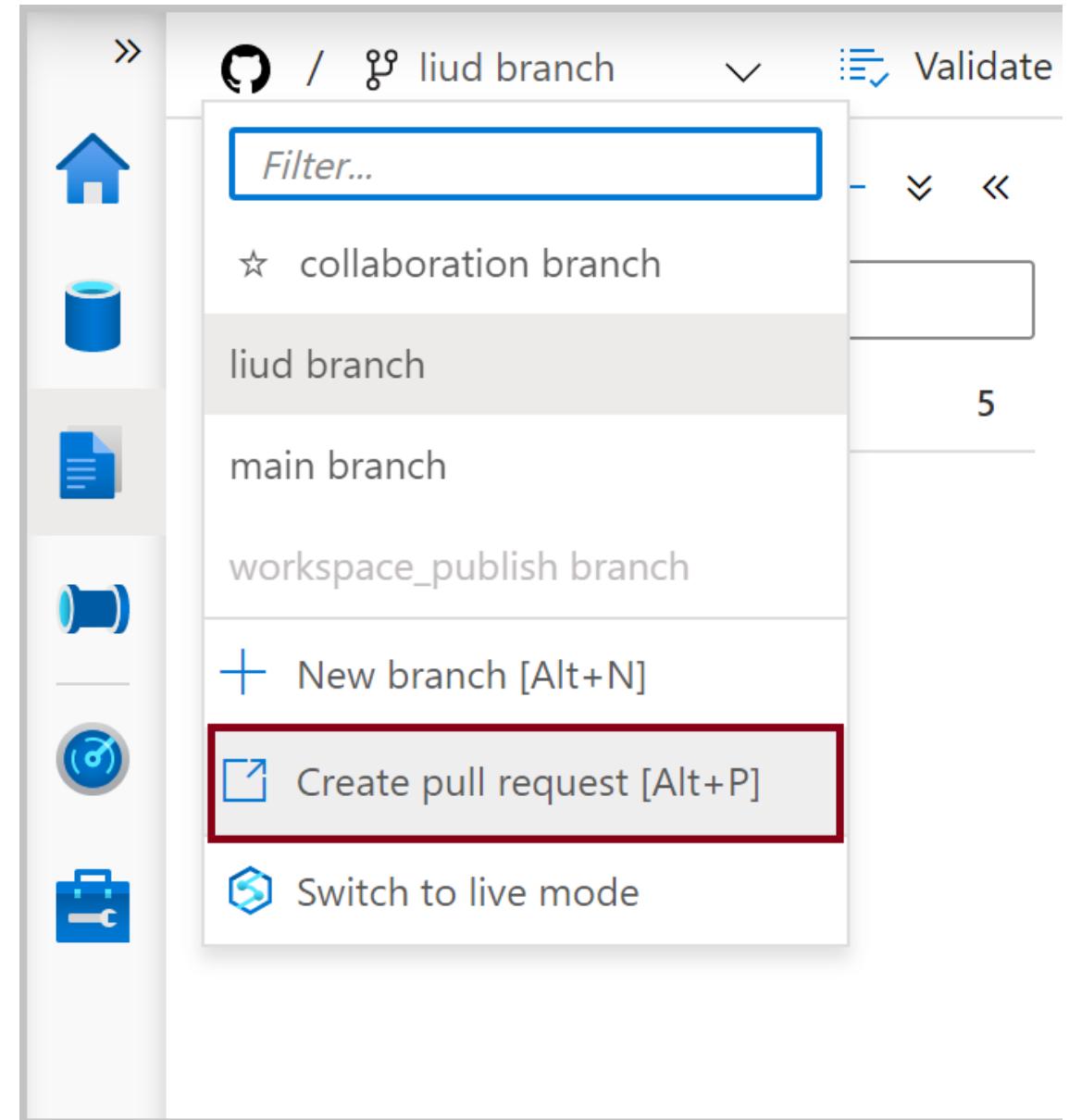
/

Import existing resource

Import existing resources to repository

Import resource into this branch 

main



CI/CD with Azure DevOps

- Establish service connections required (GitHub, ARM, Azure)
- Marketplace extension called 'Synapse workspace deployment' to assist in deploying to a production workspace (and overriding template variable values)

The screenshot shows the 'New release pipeline' configuration screen in Azure DevOps. The pipeline has one task: 'Synapse deployment task for workspace: whats-new-in-synapse'. The task is set to run on an agent. The task configuration includes:

- Synapse workspace deployment**: Task version 1.*
- Display name**: Synapse deployment task for workspace: whats-new-in-synapse-prod
- Template**: \$(System.DefaultWorkingDirectory)/_synapseconnect_synapseworkspace01/whats-new-synapse-dev/TemplateForWorkspace.json
- Template parameters**: \$(System.DefaultWorkingDirectory)/_synapseconnect_synapseworkspace01/whats-new-synapse-dev/TemplateParametersForWorkspace.json
- Synapse workspace connection type**: Microsoft Azure Sponsorship (Scoped to subscription 'Microsoft Azure Sponsorship')
- Synapse workspace resource group**: whats-new-in-synapse
- Synapse workspace name**: whats-new-in-synapse-prod
- OverrideParameters**: -workspaceName whats-new-in-synapse-prod -whats-new-synapse-dev-WorkspaceDefaultStorage_properties_typeProperties_url https://asadatalakecepprod.dfs.core.windows.net



Demo

Synapse source control and CI/CD



Mentimeter Poll

Scan QR Code

or

Go to www.menti.com and use code

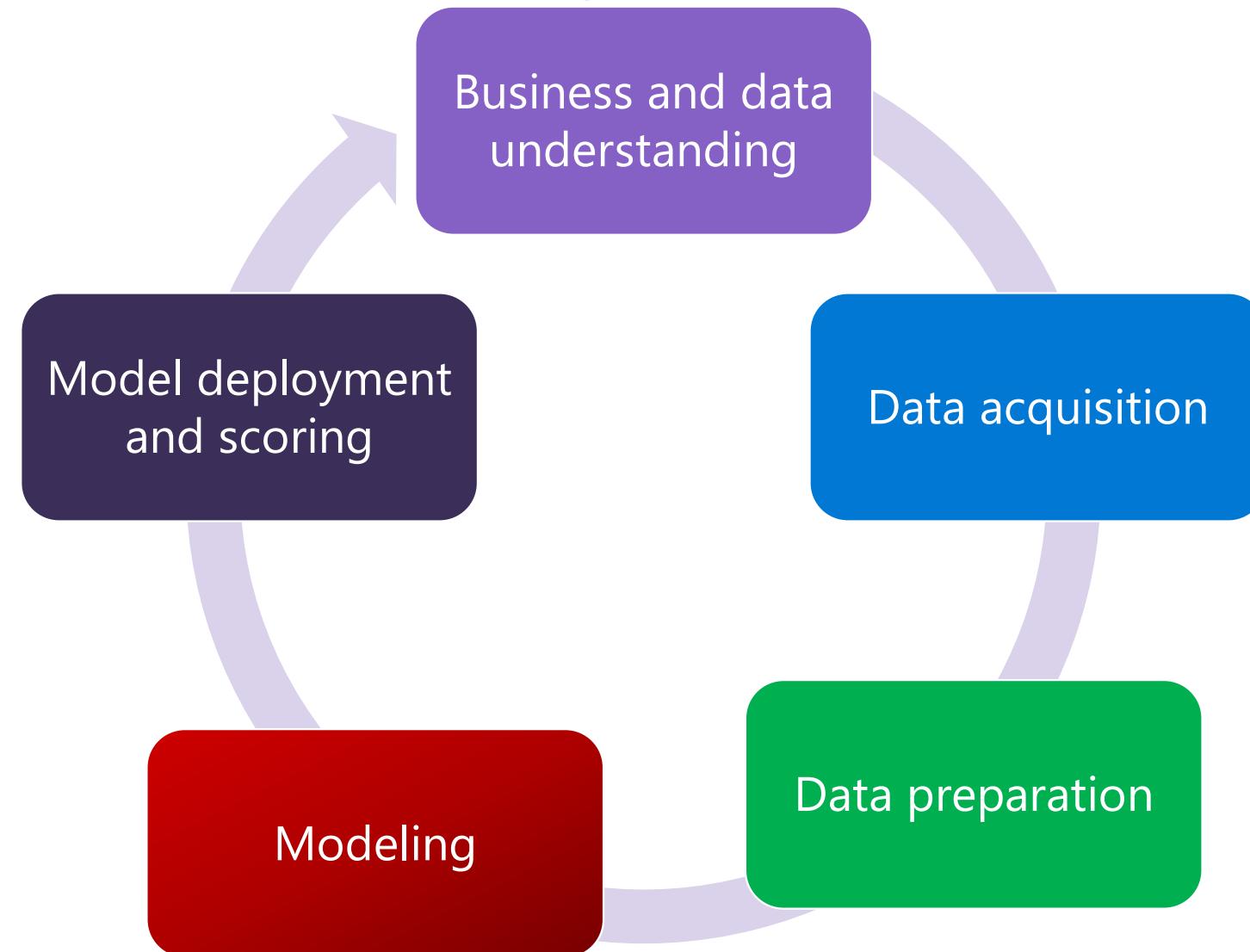
2492 0311





Azure Machine Learning integration

High level machine learning process





Modeling with 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

Microsoft Machine Learning for Apache Spark

v1.0-rc

Microsoft's Open Source
Contributions to Apache Spark



Distributed
Machine Learning



Fast Model
Deployment



Microservice
Orchestration



Multilingual Binding
Generation

www.aka.ms/spark

 Azure/mmlspark

Modeling with Azure Machine Learning – Code Free

nyc_taxi

New SQL script >

New notebook >

Machine Learning > Enrich with new model

Refresh

1

Enrich with new model

nyc_taxi

Configure experiment

This wizard will help you to train a machine learning model using [automated ML in Azure Machine Learning](#). You first need to configure the experiment that will be created and select a Spark pool to be used for training the model. [Learn more](#)

Source data

nyc_taxi

Azure Machine Learning workspace * ①

AML_ws

Experiment name * ①

nyc_taxi-20201116055046

Best model name * ①

nyc_taxi-20201116055046-Best

Target column * ①

fareAmount (double)

Spark pool * ①

Spark1110

2

Enrich with new model

nyc_taxi

Choose a model type

Select the machine learning model type for the experiment based on the question you are trying to answer. Once you have selected the model type, you will be prompted with a few settings before the experiment run is created. [Learn more](#)



Classification

Determine the likelihood of a specific outcome being achieved (binary classification) or identify the category an attribute belongs to (multiclass classification).

Example: Predict if a customer will renew or cancel their subscription.



Regression

Estimate a numeric value based on input variables.

Example: Predict housing prices based on house size.



Time series forecasting

Estimate values and trends based on historical data.

Example: Predict stock market trends over the next year.

3

Enrich with new model

nyc_taxi

Configure regression model

This model will estimate a numeric value based on input variables. [Learn more](#)

Primary metric ①

Spearman correlation

Training job time (hours) ①

3

Max concurrent iterations ①

2

ONNX model compatibility ①

Enable Disable

Create run

Open in notebook

Back

4

Modeling with Azure Machine Learning – With Code

Notebook 3

+ Cell ▾ **Run all** Undo | Publish | Attach to Spark1110 | Language PySpark (Python)

```
1 df = spark.sql("SELECT * FROM default.nyc_taxi")
2
3 datastore = Datastore.get_default(ws)
4 dataset = register_spark_dataframe(df, datastore, name = experiment_name + "-dataset")
5 dataset_train, dataset_test = dataset.random_split(percentage = 0.8)
```

Cell 3

```
[ ] 1 automl_config = AutoMLConfig(spark_context = sc,
2                                task = "regression",
3                                training_data = dataset_train,
4                                label_column_name = "fareAmount",
5                                primary_metric = "spearman_correlation",
6                                experiment_timeout_hours = 3,
7                                max_concurrent_iterations = 2,
8                                enable_onnx_compatible_models = False)
```

Cell 4

```
[ ] 1 run = experiment.submit(automl_config)
```

d all

Notebook 3 ▾ Notebook 4 ▾

+ Cell ▾ Cancel all Undo | Publish | Attach to sparkPoolMedium | Language PySpark (Python) | Azure Notebooks (Preview)

```
1 experiment_timeout_hours = 3,
2 max_concurrent_iterations = 2,
3 enable_onnx_compatible_models = True
```

Command executed in 3mins 11s 578ms by chaxu on 10-27-2020 15:33:48.557 +08:00

Cell 6

```
[6] 1 run = experiment.submit(automl_config)
```

Command executed in 3mins 48s 748ms by chaxu on 10-27-2020 15:34:25.758 +08:00

Running on remote or ADB.

Cell 7

```
[7] 1 displayHTML("<a href={} target='_blank'>Your experiment in Azure Machine Learning portal: {}</a>".format(run.get_portal_url(), run.id))
```

Command executed in 3mins 50s 564ms by chaxu on 10-27-2020 15:34:27.599 +08:00

Your experiment in Azure Machine Learning portal: AutoML_70940235-79fd-4d5d-a3ce-b05cf4e88d80

Cell 8

```
[8] 1 run.wait_for_completion()
```

Running

Cell 9

```
1 # # If you want to register the best model, please uncomment this code snippet and change model_name
2 # import onnxruntime
3 # import mlflow
4 # import mlflow.onnx
5
6 # from mlflow.models.signature import ModelSignature
7 # from mlflow.types import DataType
8 # from mlflow.types.schema import ColSpec, Schema
9
10 # # Get best model from automl run
```

Model deployment and scoring – Code Free

Microsoft Azure | Synapse Analytics > wsazuresynapseanalytics

Data

Workspace Linked

Databases 8

NYCTaxi.Pool (SQL)

- Tables
 - dbo.aml_models
 - dbo.modeldeploy
 - dbo.Models
 - dbo.nyc_taxi
 - Columns
 - dbo.nyctaxi
 - dbo.testload1029
 - dbo.Trips
 - External tables
 - External resources
 - Views

New SQL script >

New notebook >

New data flow

New integration dataset

Machine Learning > Enrich with existing model

Refresh

1

Enrich with existing model

dbo.nyc_taxi

Select the model you want to use to enrich the selected dataset. [Learn more](#)

Azure Machine Learning

Machine Learning workspace * yifso-aml-eus

Name	Version	Created	Created By	Framework
nyc_taxi_tip_predict	1	11:44:52 07/18/2020	[REDACTED]	Custom

2

Enrich with existing model

dbo.nyc_taxi

Map the source table columns to the expected model inputs. [Learn more](#)

Input mapping *

+ New | Delete

Source Column	Model Input	Input Type
fareAmount	fareAmount	real
paymentType	paymentType	bigint
passengerCount	passengerCount	bigint
tripDistance	tripDistance	real
tripTimeSecs	tripTimeSecs	bigint
pickupTimeBin	pickupTimeBin	varchar

Output mapping *

+ New | Delete

Model Output	Output Type
output_label	bigint

Continue Back Cancel

3

Model deployment and scoring – Code Free (cont...)

Enrich with existing model

dbo.nyc_taxi

Stored procedure

A stored procedure will be created once you run the generated script. Specify a name for this stored procedure. [Learn more](#)

Stored procedure name *

nyc_taxi_procedure

Target table

Create a new table or use an existing table to store the model. [Learn more](#)

Select target table *

Existing table Create new

New table *

aml_models

Deploy model + open script Back Cancel

synapseanalytics Search resources

SQL script 3

Run Undo Publish Query plan Connect to NYCTaxi_Pool Use database NYCTaxi_Pool

```
1 CREATE PROCEDURE dbo.test_nyctaxi_scoring112
2 AS
3 BEGIN
4
5 SELECT
6     CAST([fareAmount] AS [real]) AS [fareAmount],
7     CAST([paymentType] AS [bigint]) AS [paymentType],
8     CAST([passengerCount] AS [bigint]) AS [passengerCount],
9     CAST([tripDistance] AS [real]) AS [tripDistance],
10    [tripTimeSecs],
11    CAST([pickupTimeBin] AS [varchar]) AS [pickupTimeBin]
12 INTO [dbo].[#nyc_taxi]
13 FROM [dbo].[nyc_taxi];
14
15 SELECT *
16 FROM PREDICT (MODEL = (SELECT [model] FROM dbo.aml_models WHERE [ID] = 'nyc_taxi_tip_predict:1'),
17 | | | DATA = [dbo].[#nyc_taxi]) WITH ([output_label] [bigint])
18
19 END
20 GO
21
22 EXEC dbo.test_nyctaxi_scoring112
```

Results Messages

View Table Chart Export results

Search

Output_label	FareAmount	PaymentType	PassengerCount	TripDistance	TripTimeSecs	PickupTimeBin
1	5	1	1	0.7	235	PMRush
1	6	1	1	1.06	357	Afternoon
1	9	1	1	1.7	619	Night
0	5.5	2	1	0.52	337	AMRush
1	16.5	1	1	4.17	1186	Night
0	10.5	2	1	3.1	547	Night
0	5	2	1	0.8	265	Afternoon

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.

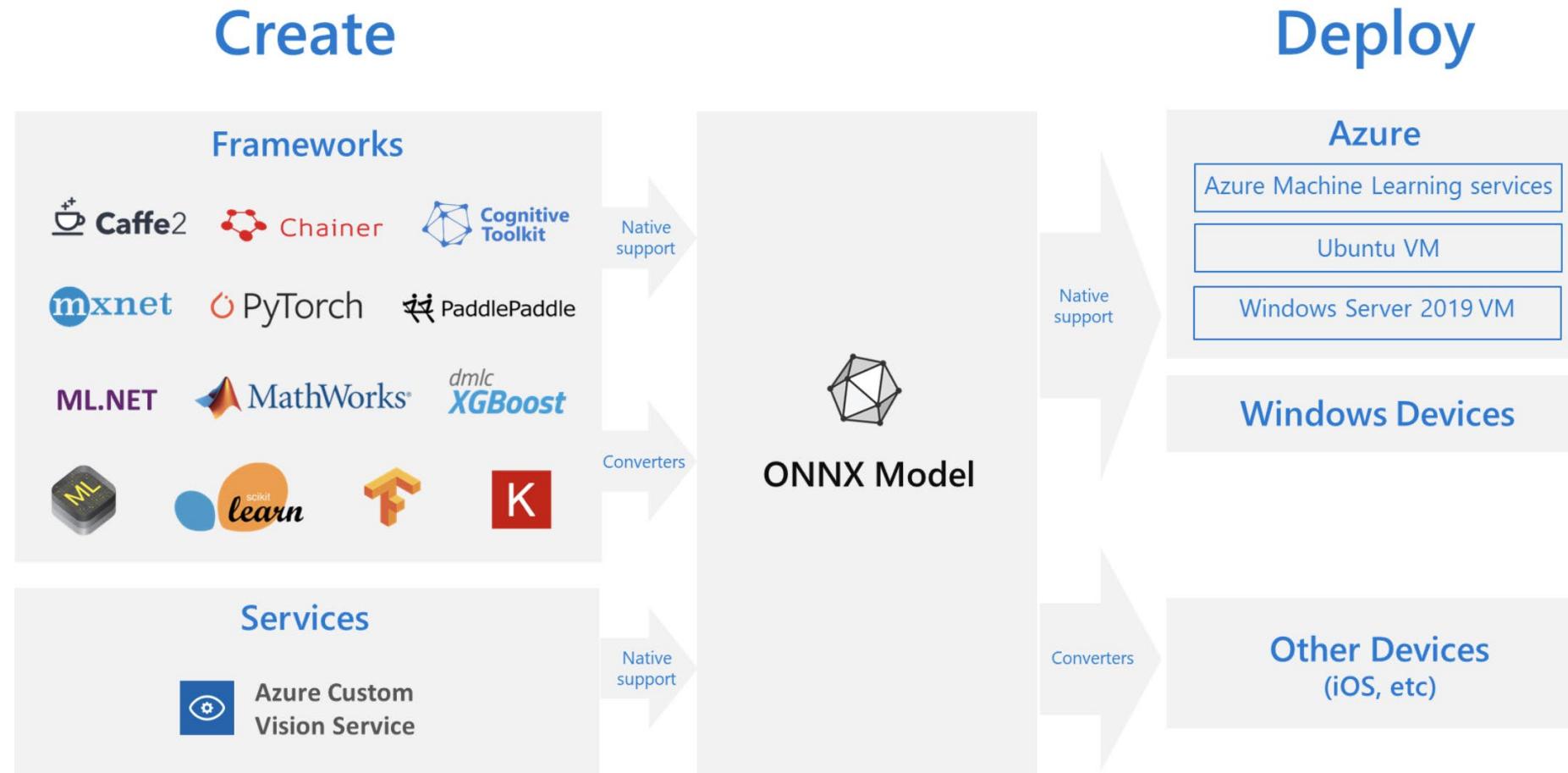
Syntax:

```
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:

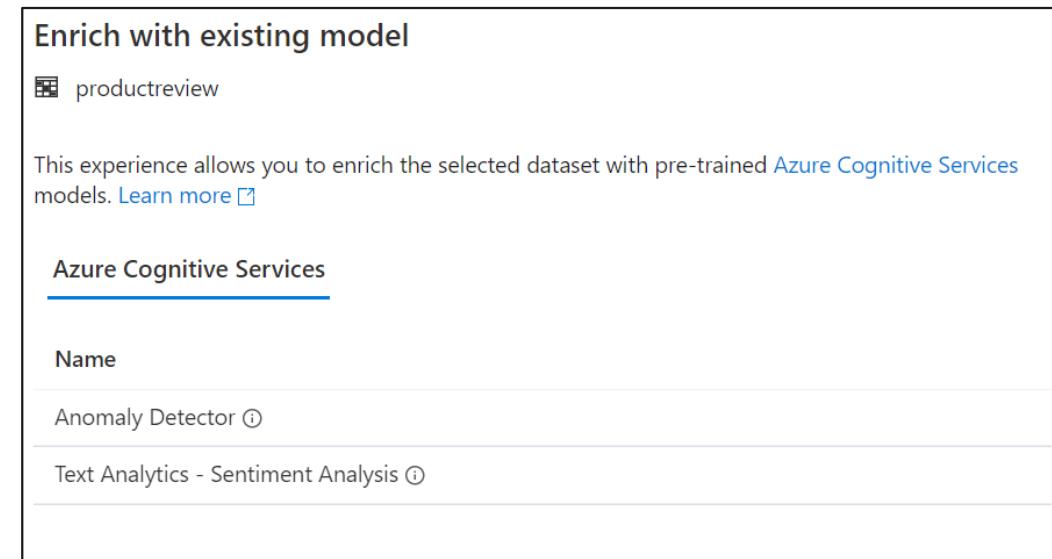
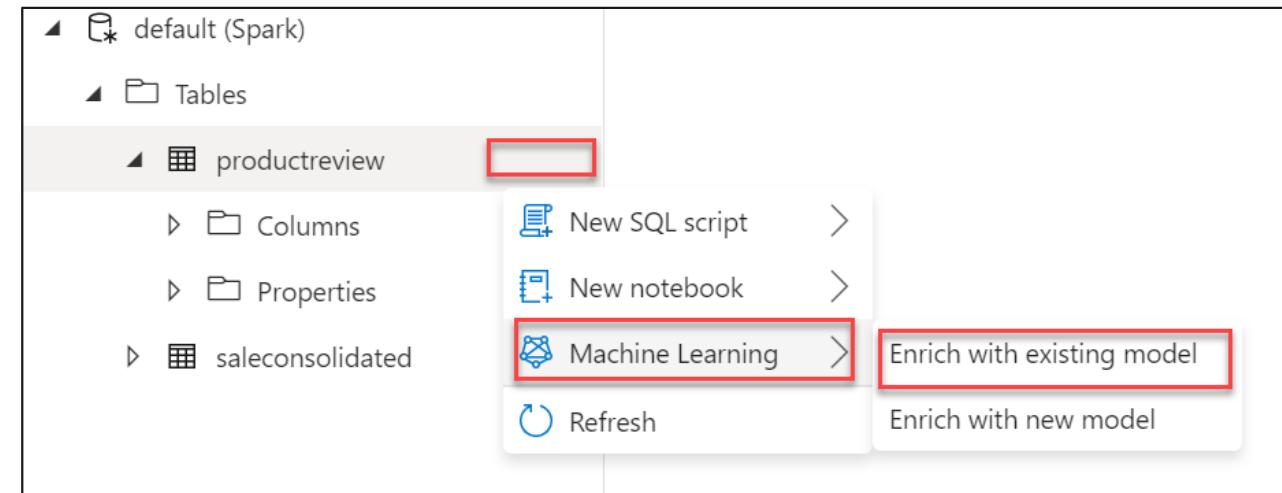
```
DECLARE @model varbinary(max) = (SELECT Model FROM Models WHERE Id = <>);
SELECT d.*, p.Score
FROM PREDICT(MODEL = @model,
    DATA = dbo.mytable AS d) WITH (Score float) AS p;
```

What is ONNX?



Leveraging Cognitive services

- Enrich with existing model
- Select from the Cognitive Services available
- Choose the appropriate Cognitive Service resource
- Cog. Svc. API Key must be stored in a Key Vault
- Set input column
- Generates a PySpark Notebook that applies the cognitive service model to the column chosen





Mentimeter Poll

Scan QR Code

or

Go to www.menti.com and use code

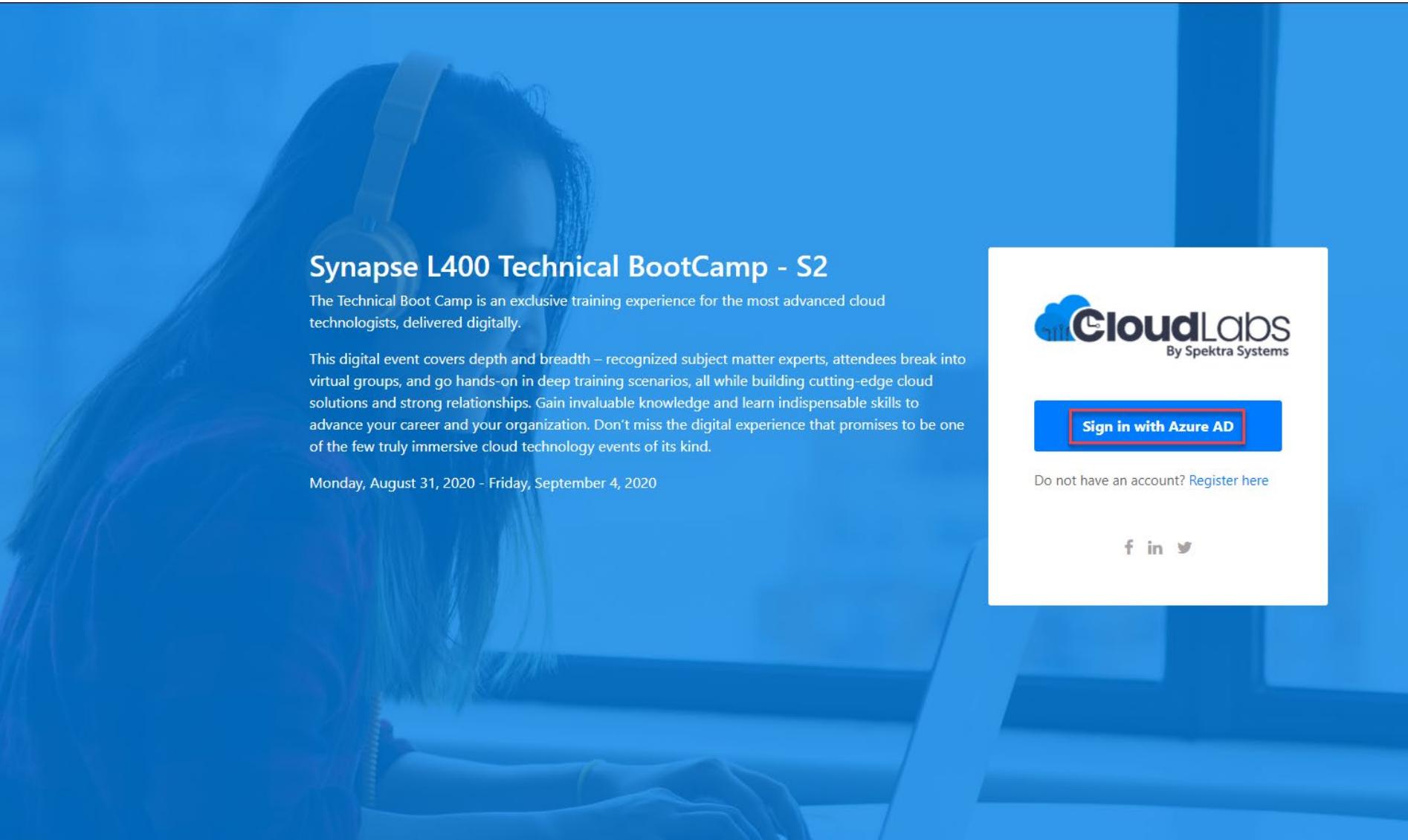
9728 4952





Lab environment overview

Login to the portal

A large, semi-transparent blue rectangular overlay covers the left side of the page, featuring a blurred background image of a person's head and shoulders wearing over-ear headphones. On the right side, there is a white rectangular login form.

Synapse L400 Technical BootCamp - S2

The Technical Boot Camp is an exclusive training experience for the most advanced cloud technologists, delivered digitally.

This digital event covers depth and breadth – recognized subject matter experts, attendees break into virtual groups, and go hands-on in deep training scenarios, all while building cutting-edge cloud solutions and strong relationships. Gain invaluable knowledge and learn indispensable skills to advance your career and your organization. Don't miss the digital experience that promises to be one of the few truly immersive cloud technology events of its kind.

Monday, August 31, 2020 - Friday, September 4, 2020

CloudLabs
By Spektra Systems

[Sign in with Azure AD](#)

Do not have an account? [Register here](#)

f in tw

Sessions for Bootcamp

The screenshot shows the CloudLabs platform interface. At the top, there is a navigation bar with links: My Sessions (highlighted with a red dashed border), My Calendar, My Team, Users, and Teams. On the far right, it says "Hi" followed by a user icon. Below the navigation bar is a banner featuring a calendar on a tablet and a keyboard, with the text "My Sessions" and "Synapse L400 Technical BootCamp - S2".

Below the banner, there are several filter buttons: "All Days" (highlighted in blue), "Mon 31", "Tue 1", "Wed 2", "Thu 3", and a "Add to Calendar" button. There is also a search bar with the placeholder "Search" and a magnifying glass icon.

The main content area displays a single session card:

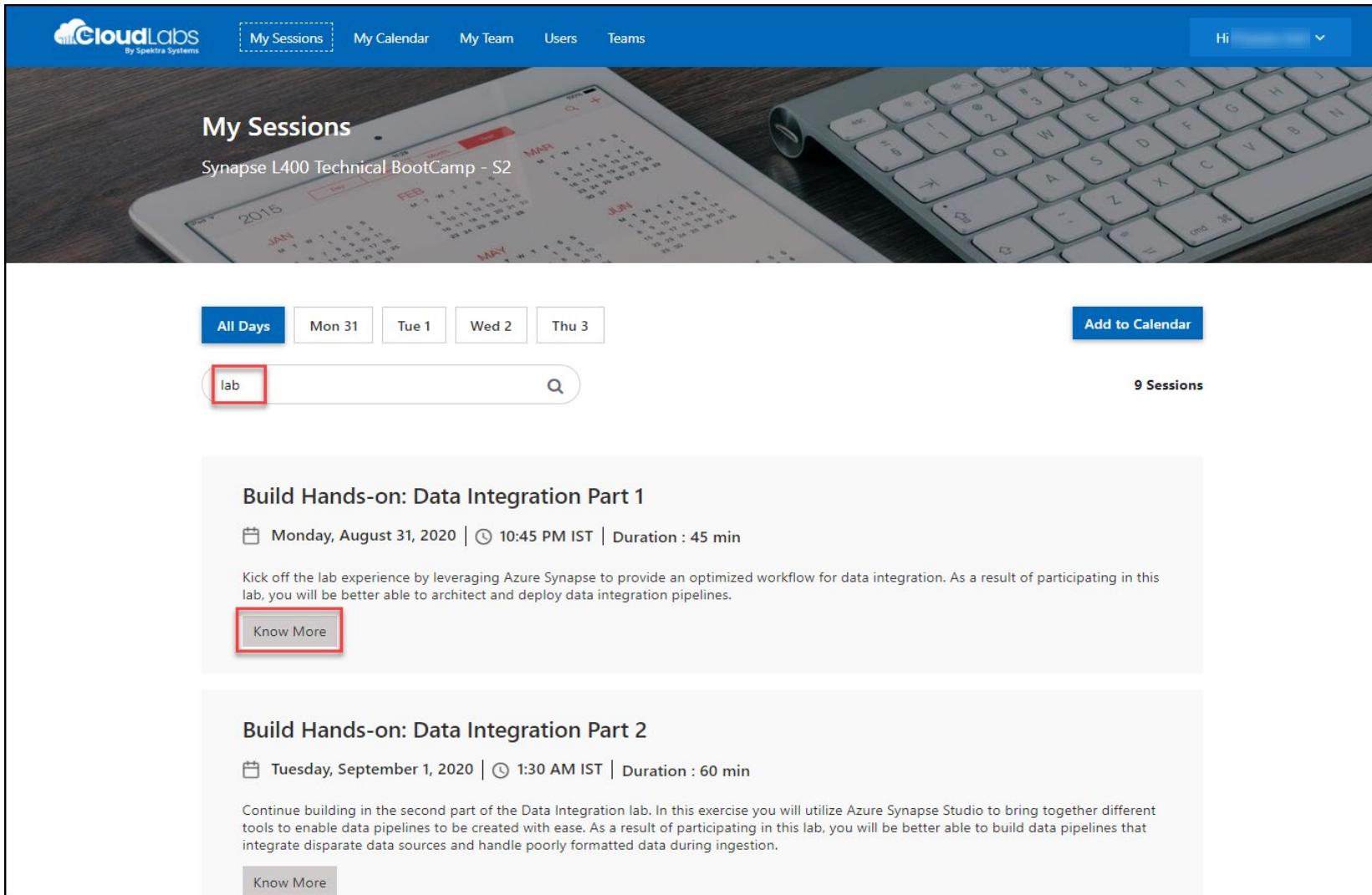
Welcome
Speaker: Leanne Gallagher
Monday, August 31, 2020 | 7:30 PM IST | Duration : 5 min

Welcome to the Azure Synapse Technical Boot Camp! Discover what we will be learning this week, learn about the resources available to you and where to find them, and connect with your peers.

[Know More](#)

At the bottom right of the main content area, it says "37 Sessions".

Search for lab



CloudLabs
By Spektra Systems

My Sessions My Calendar My Team Users Teams Hi

My Sessions

Synapse L400 Technical BootCamp - S2

All Days Mon 31 Tue 1 Wed 2 Thu 3 Add to Calendar

lab

9 Sessions

Build Hands-on: Data Integration Part 1
Monday, August 31, 2020 | 10:45 PM IST | Duration : 45 min
Kick off the lab experience by leveraging Azure Synapse to provide an optimized workflow for data integration. As a result of participating in this lab, you will be better able to architect and deploy data integration pipelines.
[Know More](#)

Build Hands-on: Data Integration Part 2
Tuesday, September 1, 2020 | 1:30 AM IST | Duration : 60 min
Continue building in the second part of the Data Integration lab. In this exercise you will utilize Azure Synapse Studio to bring together different tools to enable data pipelines to be created with ease. As a result of participating in this lab, you will be better able to build data pipelines that integrate disparate data sources and handle poorly formatted data during ingestion.
[Know More](#)

- Search for lab
- Click on Know More



Break Time

Lab 01: Azure Machine Learning integration

Objective: As a result of participating in this lab, you will be better able to use the Azure Machine Learning integration in Synapse Studio

What you will be doing: You will **work individually in your lab environment to complete a set of tasks** to create a machine learning model from within Synapse Studio

Total Activity Time: 60 minutes



Azure Data Explorer integration

What is Azure Data Explorer (ADX)?

- Fully managed cluster-based analytics service
- Optimized for telemetry and streaming data
- KQL query language
- Data service powering Azure Monitor, Time Series Insights, and Windows Defender Advanced Threat Protection



```
1 // Cluster Diagnostics
2 □ .show diagnostics
3 | extend Passed= (IsHealthy) and not(IsScaleOutRequired)
4 | extend Summary = strcat('Cluster is ', iif(Passed, '', 'NOT'), 'healthy.')
5 | Details=pack('MachinesTotal', MachinesTotal, 'DiskCacheCapacity', round(ClusterDataCapacityFactor,1))
6 | project Action = 'Cluster Diagnostics', Category='Info', Summary, Details;
7
8 □ // Permission check
9 .show principal roles
10
```

Action	Category	Summary	Details
Cluster Diagnostics	Info	Cluster is healthy.	{"MachinesTotal":2,"DiskCacheCapacity":58.4}
Permission check	Authorization	Diagnose will examine usage of 1 database(s).	["mainbearingdb"]
CPU utilization by workload type	CPU	CPU consumed past 24h: 4.4 hours. Top 3 consumers: Query (70): 1.5h ExtentsDrop (24): 0.0h	{"Operations":[{"CommandType":"ExtentsDrop","CPU":4.4,"WorkloadType":"Query"}]}
CPU utilization (commands)	CPU	CPU consumed by commands past 24h: 0.0 hours. Top 3 operations: ExtentsDrop (24): 0.0h Top 3 principals: AAD app id=d38bccce-105f-45ea-b8e8-bf799aac8165 (24): 0.0h	{"Operations":[{"CommandType":"ExtentsDrop","CPU":0.0,"WorkloadType":"Commands"}]}
		CPU consumed by queries past 24h: 3.0 hours. Top 3 principals:	



How did ADX come to be?



- 2014 – R&D
- 2016 – Added as the data service for Application Insights
- 2018 – Public preview
- 2019 - GA

Azure Data Explorer and Azure Synapse Analytics

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The top navigation bar includes 'Microsoft Azure' and 'Synapse Analytics > contosoworkspace01'. Below the navigation bar are buttons for 'Publish all', 'Validate all', 'Refresh', and 'Discard all'. The left sidebar has a 'Data' section with 'Workspace' and 'Linked' tabs. Under 'Linked', there is a 'Filter resources by name' search bar and a list of 'Azure Data Explorer' resources. One resource, 'contosoadx_eastus2 (ignite...)', is expanded to show 'TruckTelemetry', 'BuildingTelmetry', 'Occupancy' (with 'StageloTRawData' and 'Thermostats' listed), and 'ML_Models'. A red box highlights the 'New notebook' button, which is also circled with a large number '1'. Other options in the dropdown menu include 'Read DataFrame from table', 'Write DataFrame to table', and 'Write streaming DataFrame to table'.

The screenshot shows a Jupyter Notebook cell titled 'Cell 1'. The cell contains the following PySpark code:

```
1 %%pyspark
2
3
4 # Read data from Azure Data Explorer table(s)
5 # Full Sample Code available at: https://github.com/Azure/azure-kusto-spark/blob/master/samples/pyspark/read\_kusto\_spark.py
6
7 kustoDf = spark.read \
8     .format("com.microsoft.kusto.spark.synapse.datasource") \
9     .option("spark.synapse.linkedService", "contosoadx_eastus2") \
10    .option("kustoDatabase", "Occupancy") \
11    .option("kustoQuery", "Thermostats | take 10") \
12    .load()
13
14 display(kustoDf)
15
```

The first line of code, '%%pyspark', is highlighted with a red box and circled with a large number '2'.



Mentimeter Poll

Scan QR Code

or

Go to www.menti.com and use code

7594 9409





Spark Updates

UI Changes

- Session configuration has been re-located to the top of the notebook.
- Package management moved to the “Packages” section of the action menu on a Spark pool in the manage hub.
- The Notebook experience now defaults to the preview version.

The screenshot illustrates two UI changes. At the top, the notebook header includes session configuration options like 'Attach to' and a gear icon for settings. Below, the 'Manage' view for a 'SparkPool01' pool shows a context menu with an option 'Packages' highlighted by a red box.

Name	Size
SparkPool01	Small (4 vCores / 32 GB) - 3 to 4 nodes

- ... (More options)
- Scale settings
- Auto-pause settings
- Packages** (highlighted with a red box)
- Apache Spark configuration
- Assign tags
- View role assignments
- Delete

Configuration updates

- New node sizes introduced XLarge and XXLarge
- YAML (*.yml) format supported for libraries in addition to requirements.txt
- Workspace packages (custom/private wheel (whl)files or jar files) can be uploaded from the Manage Hub
- Specific workspace packages can then be selected for use on a Spark Pool via the Package management view
- Enabling Session level packages on a Spark pool allows for testing new versions of packages without impacting the underlying pool configuration
- Spark Configuration file upload is now available

Size	vCore	Memory
Small	4	32 GB
Medium	8	64 GB
Large	16	128 GB
XLarge	32	256 GB
XXLarge	64	432 GB

The screenshot shows the Azure portal interface for managing packages in a Spark pool. On the left, a sidebar lists various management options: Integration, Triggers, Integration runtimes, Security, Access control, Credentials, Managed private endpoints, Code libraries, Workspace packages (which is highlighted with a red box), Source control, and Git configuration. On the right, a preview pane titled 'Manage packages' for 'sp01' displays the following information:

- Allow session level packages:** A radio button is set to 'Enable'.
- Requirements files:** Includes 'Upload' and 'Refresh' buttons.
- NAME SIZE DATE**

environment.yml	141B	2/23/2021, 10:06:10 PM
-----------------	------	------------------------
- Workspace packages:** Includes 'Select from workspace packages' and 'Refresh' buttons.
- NAME DATE**

stanford-corenlp-3.3.0.jar	2/23/2021, 9:58:35 PM
----------------------------	-----------------------

At the bottom right are 'Apply' and 'Cancel' buttons.

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.

The screenshot shows a Jupyter Notebook interface with three code cells. Cell 1 shows the import of mssparkutils and its help function. Cell 2 shows the import of credentials and its help function. Cell 3 shows the import of env and its help function. Each cell includes the command, its execution time, and the resulting help documentation for the respective module.

Cell 1:

```
[2] 1 from notebookutils import mssparkutils  
2 mssparkutils.fs.help()
```

Command executed in 437ms by prlangad on 11-24-2020 18:32:02.018 -08:00

mssparkutils.fs provides utilities for working with various FileSystems.

Below is overview about the available methods:

cp(from: String, to: String, recurse: Boolean = false): Boolean -> Copies a file or directory, possibly across FileSystems
mv(from: String, to: String, recurse: Boolean = false): Boolean -> Moves a file or directory, possibly across FileSystems
ls(dir: String): Array -> Lists the contents of a directory
mkdirs(dir: String): Boolean -> Creates the given directory if it does not exist, also creating any necessary parent directories
put(file: String, contents: String, overwrite: Boolean = false): Boolean -> Writes the given String out to a file, encoded in UTF-8
head(file: String, maxBytes: int = 1024 * 100): String -> Returns up to the first 'maxBytes' bytes of the given file as a String encoded in UTF-8
append(file: String, content: String, createFileIfNotExists: Boolean): Boolean -> Append the content to a file
rm(dir: String, recurse: Boolean = false): Boolean -> Removes a file or directory

Use mssparkutils.fs.help("methodName") for more info about a method.

Cell 2:

```
[3] 1 mssparkutils.credentials.help()
```

Command executed in 355ms by prlangad on 11-24-2020 18:32:47.633 -08:00

getToken(audience, name): returns AAD token for a given audience, name (optional)
isValidToken(token): returns true if token hasn't expired
getConnectionStringOrCreds(linkedService): returns connection string or credentials for linked service
getSecret(akvName, secret, linkedService): returns AKV secret for a given AKV linked service, akvName, secret key
getSecret(akvName, secret): returns AKV secret for a given akvName, secret key
putSecret(akvName, secretName, secretValue, linkedService): puts AKV secret for a given akvName, secretName
putSecret(akvName, secretName, secretValue): puts AKV secret for a given akvName, secretName

Cell 3:

```
1 mssparkutils.env.help()
```

Command executed in 472ms by prlangad on 11-24-2020 18:33:14.526 -08:00

getUser(): returns user name
getUser(): returns unique user id
getJobId(): returns job id
getWorkspaceName(): returns workspace name
getPoolName(): returns Spark pool name
getClusterId(): returns cluster id

Spark 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

The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** Cell ▾, Run all, Undo, Publish, Attach to analytics1, Language PySpark (Python).
- Status Bar:** Not started.
- Create indexes:** Cell 4 contains code to create indexes from configurations:

```
[ ] 1 # Create indexes from configurations
2
3 hyperspace.createIndex(emp_DF, emp_IndexConfig)
4 hyperspace.createIndex(dept_DF, dept_IndexConfig1)
5 hyperspace.createIndex(dept DF, dept IndexConfig2)
```
- List indexes:** Cell 6 contains code to list indexes:

```
[ ] 1 hyperspace.indexes().show()
```
- Index usage:** Cell 8 contains code to enable Hypespace and demonstrate its usage with filters:

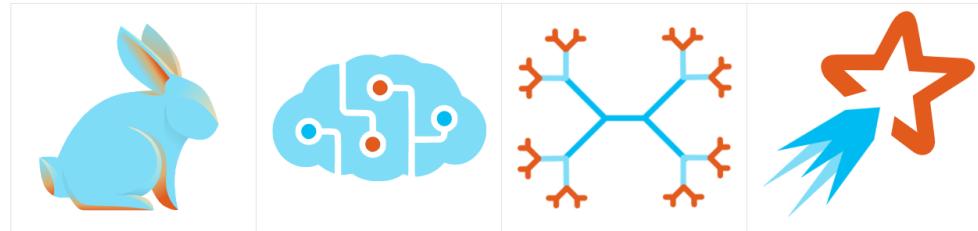
```
[ ] 1 # Enable Hypespace
2 Hyperspace.enable(spark)
3
4 emp_DF = spark.read.parquet(emp_Location)
5 dept_DF = spark.read.parquet(dept_Location)
6
7 emp_DF.show(5)
8 dept_DF.show(5)
9
10 # Filter with equality predicate
11
12 eqFilter = dept_DF.filter("deptId = 20").select("deptName")
13 eqFilter.show()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



MMLSpark



Projects



Vowpal Wabbit on Spark	The Cognitive Services on Spark	LightGBM on Spark	Spark Serving
------------------------	---------------------------------	-------------------	---------------

Fast, Sparse, and Effective Text Analytics	Leverage the Microsoft Cognitive Services at Unprecedented Scales in your existing SparkML pipelines	Train Gradient Boosted Machines with LightGBM	Serve any Spark Computation as a Web Service with Sub-Millisecond Latency
--	--	---	---



HTTP on Spark	CNTK on Spark	Lime on Spark	Spark Binding Autogeneration
---------------	---------------	---------------	------------------------------

An Integration Between Spark and the HTTP Protocol, enabling Distributed Microservice Orchestration	Distributed Deep Learning with the Microsoft Cognitive Toolkit	Distributed, Model Agnostic, Interpretations for Classifiers	Automatically Generate Spark bindings for PySpark and SparklyR
---	--	--	--



Isolation Forest on Spark	CyberML	Conditional KNN
---------------------------	---------	-----------------

Distributed Nonlinear Outlier Detection	Machine Learning Tools for Cyber Security	Scalable KNN Models with Conditional Queries
---	---	--

Lab 02: Spark

Objective: As a result of participating in this lab, you will be better able to understand Spark Hyperspace and mssparkutil

What you will be doing: You will **work individually in your lab environment to complete a set of tasks** to index files in storage for faster read performance.

Total Activity Time: 60 minutes





Break Time

CDM – Common Data Model



What is the Common Data Model

- Standard schemas defining the most used concepts and activities
- Structural Consistency
- Semantic Consistency
- Unified shape
- Extensible standard entities



Common data model documents

- JSON based with the extension “*.cdm.json”
- Model both physical and logical entities
- Create a folder logically named to hold schema files
- Import other documents using _allImports.cdm.json or the “imports”: [{ “corpusPath”: “/foundations.cdm.json”}] property
- Manifest documents *.manifest.cdm.json acts like an entry point to the entities and should be at the root of the folder. Sub-manifests are also supported

```
{  
  "$schema": "../schema.cdm.json",  
  "jsonSchemaSemanticVersion": "1.0.0",  
  "imports": [  
    {  
      "corpusPath": "/foundations.cdm.json"  
    }  
  ],  
  "definitions": [  
    {  
      "entityName": "UserAgent",  
      "extendsEntity": "CdmEntity",  
      "description": "The user agent.",  
      "hasAttributes": [  
        {  
          "name": "browserName",  
          "dataType": "string",  
          "description": "Browser name."  
        },  
        {  
          "name": "browserVersion",  
          "dataType": "string",  
          "appliedTraits": [  
            "means.measurement.version"  
          ],  
          "description": "Browser version."  
        },  
        {  
          "name": "osName",  
          "dataType": "string",  
          "description": "Operating system name."  
        },  
        {  
          "name": "osVersion"  
        }  
      ]  
    }  
  ]  
}
```



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()
```

CDM in Data Flows

- Available in Data flows as an Inline source type
- Source/Sink options allows for the selection of the manifest and root location and Entity

Source settings Source options Projection Optimize Inspect Data preview

Output stream name * Learn more

Source type * Dataset Inline

Inline dataset type * Common Data Model

Linked service * whats-new-synapse-dev-Workspace... Edit New

Integration runtime * AutoResolveIntegrationRuntime

Sampling * Enable Disable

Source settings **Source options** Projection Optimize Inspect Data preview

▲ File settings

Metadata format Manifest Model.json

Root location * /

Manifest file Root location / /

Start time (UTC) End time (UTC)

Filter by last modified

Allow no files found

▲ Entity reference

Entity reference type Local Custom Standard

Entity * Example: Account

Azure Purview integration

What is Azure Purview

Overview

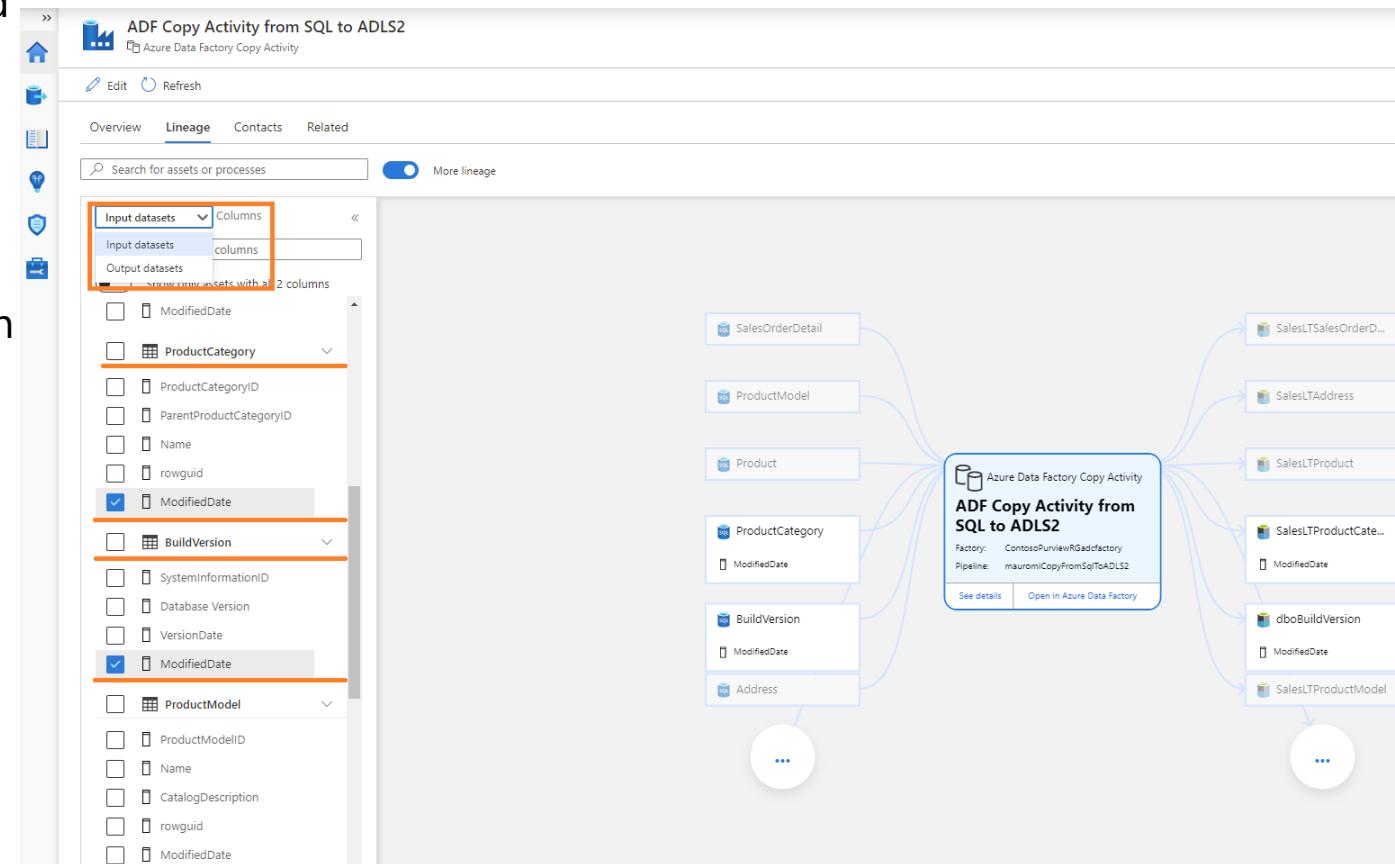
SaaS unified data governance tool for on-premises and cloud data. Provides a holistic and up to date view of your data landscape.

Benefits

Purview Data Map is automatically kept up to date with built-in automated scanning and classification system

Rich User Interface – Azure Purview Data Map. Users can quickly and easily find relevant data based on filters.

Centrally located data documentation. Users can contribute to the catalog to enrich automatically obtained metadata.



Connecting Purview with Azure Synapse

- Manage hub -> External connections -> Azure Purview (Preview)
- Provides a Purview search bar at the top of Synapse Studio in the Data, Develop, and Integrate Hubs.
- For the search to work, data sources must be registered in Azure Purview (ex. ADLS Gen 2, Azure Data Explorer, Dedicated SQL Pool)

Connect to a Purview account

Registering a Purview account to a Synapse workspace will help you discover, understand and explore your organization's data

Account selection method *

From Azure subscription Enter manually

Purview account name *

whats-new-purview

Purview   Search your organization data

Search history

- (L) person's name canada
- (L) customers markets
- (L) report europe
- (L) finance

Recently accessed [View all](#)

 Einat
<https://app.powerbi.com/groups/2b7eb3e5-7468-4600-af03-fdc17641022b>

 measure_data
mssql://jcomssql/MSSQLSERVER/testdb dbo/measure_data



Demo

Azure Purview integration with Synapse

Lab 03: CDM

Objective: As a result of participating in this lab, you will be better able to understand how to use and leverage CDM in a solution.

What you will be doing: You will **work individually in your lab environment to complete a set of tasks** to leverage CDM in a solution.

Total Activity Time: 60 minutes





Break Time

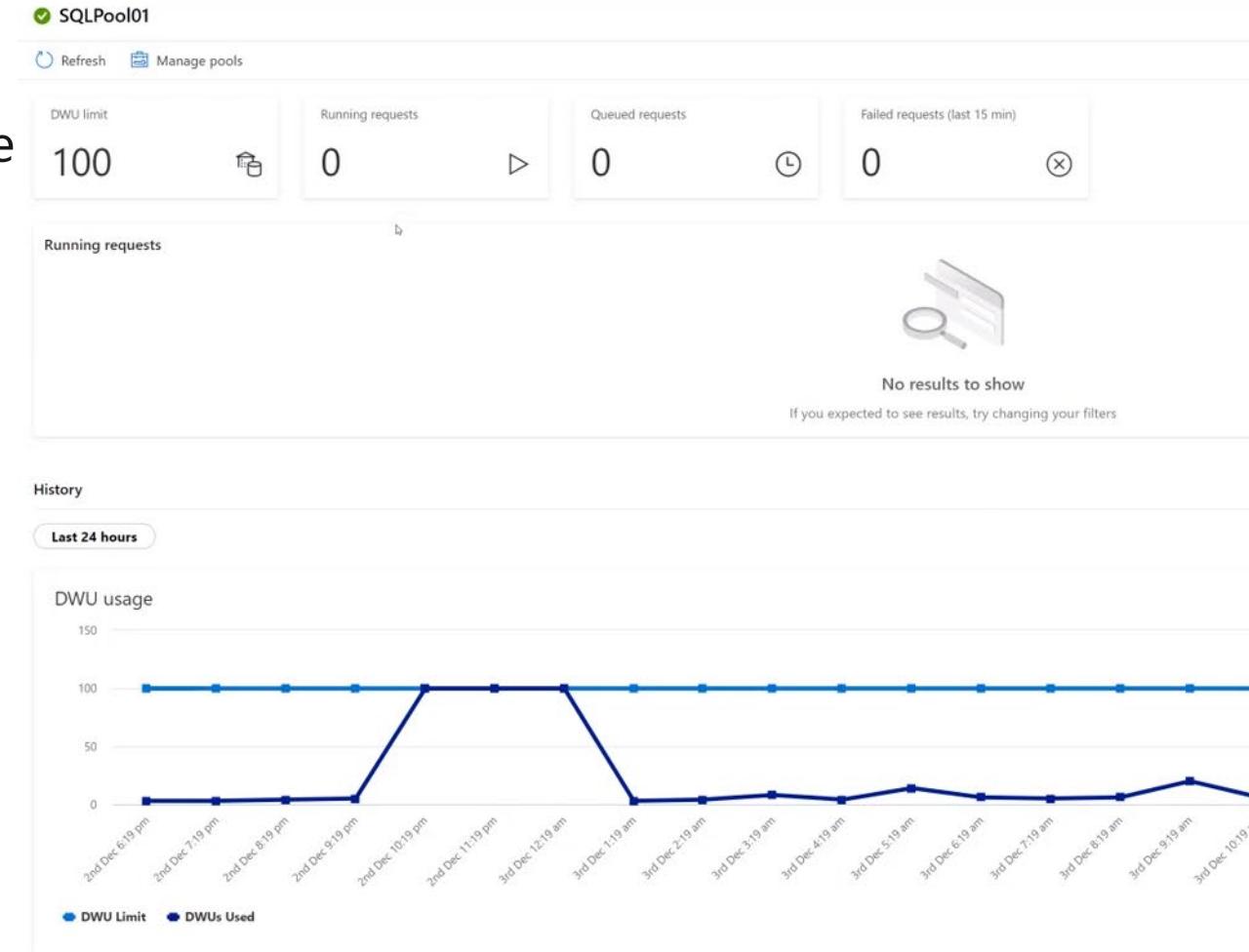


Monitoring enhancements



Monitoring enhancements

- Synapse Studio Monitor Hub now displays resource consumption-based data both real-time and historical
- Spark application playback
- Azure portal Azure monitor improvements
 - More alert conditions
- Azure portal Data classification and discovery UI
- Azure portal Dynamic data masking UI



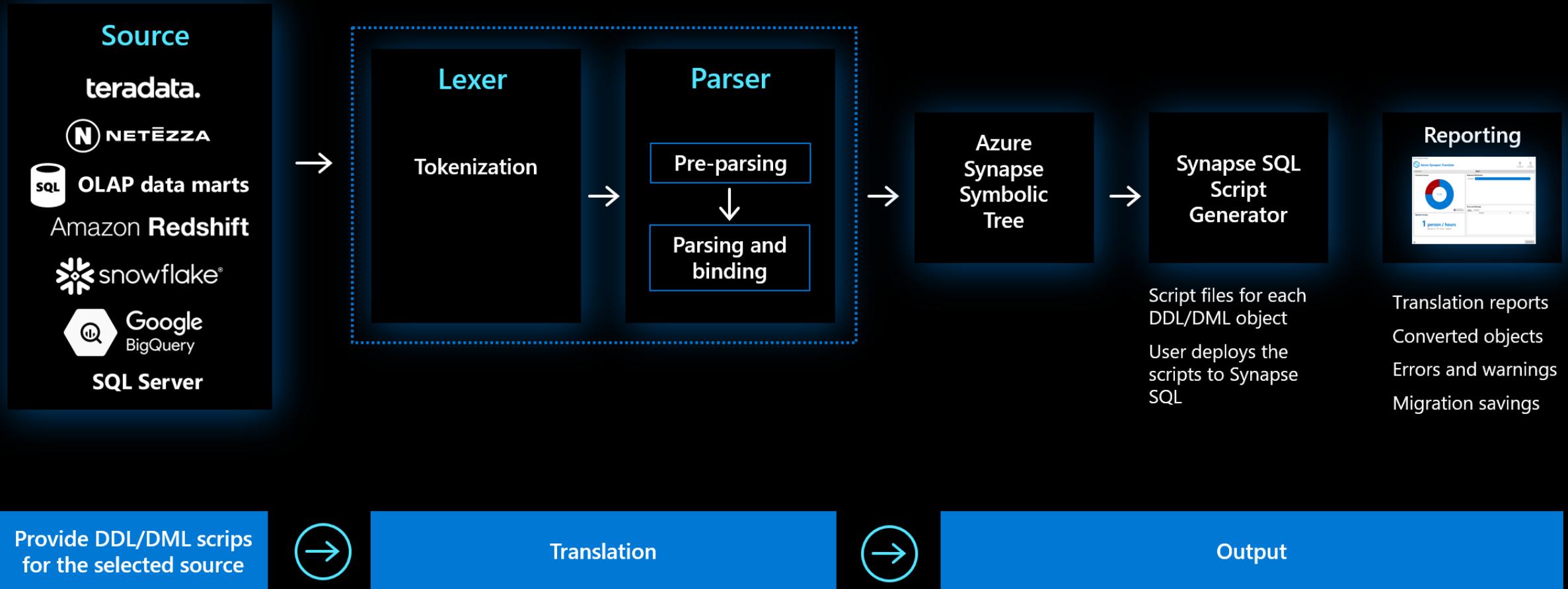


Demo

Monitoring enhancements

Migrating existing data
warehouses

Azure Synapse Pathway code translation



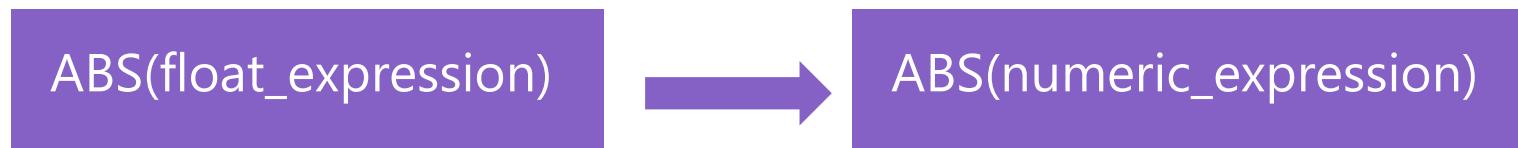


Stage 1 – Lexing and parsing

- Processes SQL input into an augmented Abstract Syntax Tree used in further processing

Stage 2 – Augmented abstract syntax tree (AST)

- Using the augmented Abstract Syntax Tree, code is translated between systems



Stage 3 – Syntax generation

- DDL and load scripts are created for the target system

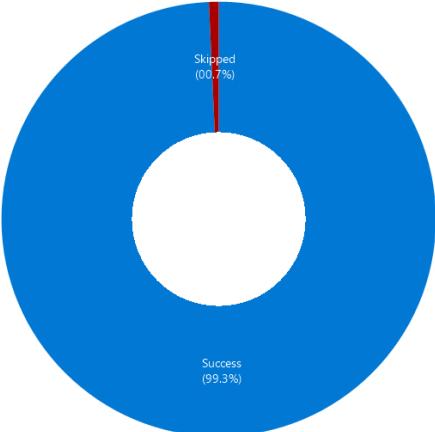
```
DROP TABLE staging.table1;
CREATE TABLE staging.table1
WITH
(
    -- Derived from the original table definition
    DISTRIBUTION = HASH(column1),
    -- Derived from the original table definition
    CLUSTERED COLUMNSTORE INDEX
)
AS SELECT * FROM staging.table2;
```

Translation results

Azure Synapse Pathway

Configuration Report

Conversion Success



Success (99.3%)
Skipped (0.7%)

Migration Savings

69 person hours
Based on 30 mins / object

Statement Distribution

Statement Type	Count
All Statements	138 / 139
AlterDatabase	0 / 1
CreateDatabase	1 / 1
DropDatabase	1 / 1
CreateSchema	8 / 8
CreateTable	128 / 128

Issues

Errors **Warnings**

File	# Warnings
C:\Users\anrampala\OneDrive - Microsoft\Azure Synapse\Azur	1

Issue Details

All Statements

C:\Users\OneDrive - Microsoft\Azure Synapse\Azure Synapse Translate\Input\NZ\nz.sql

Message Line

ERROR_TRANSLATION('ADD COLUMN col1 INT 0')

View Results



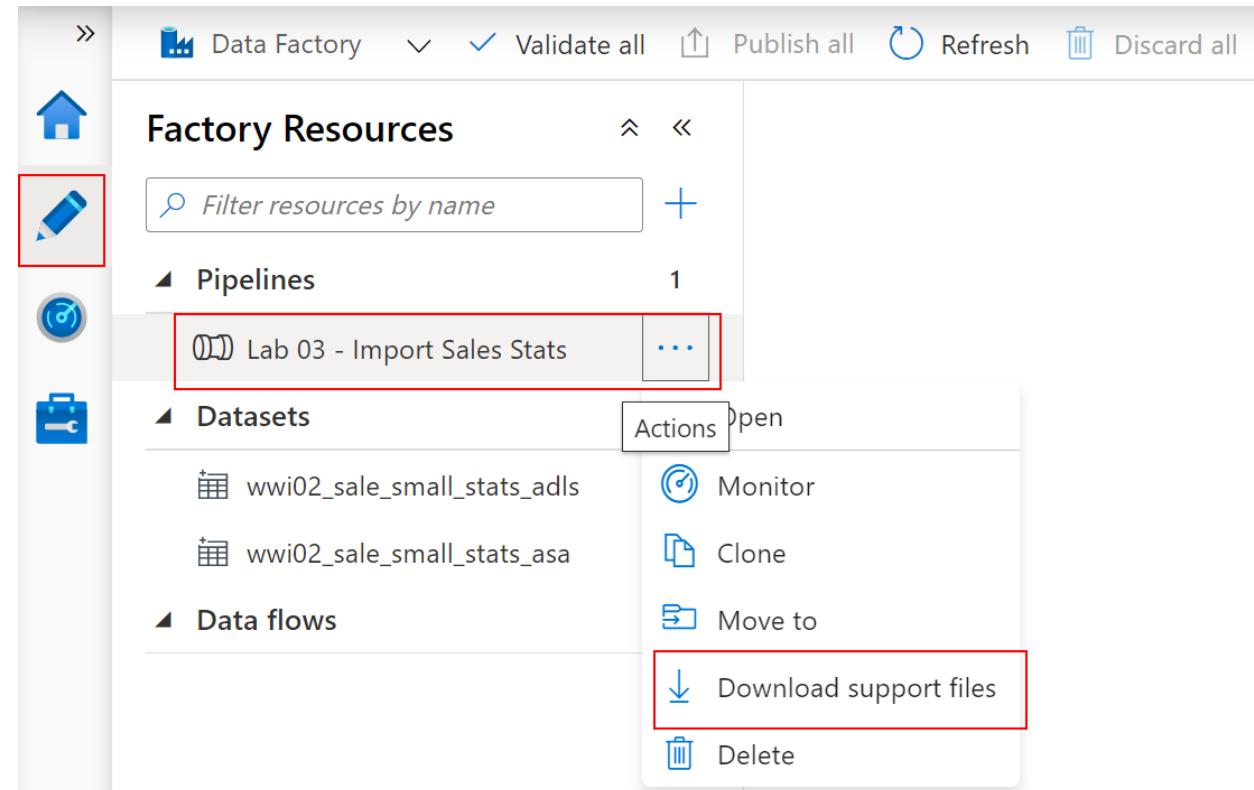
Demo

Migrating existing data warehouses

Migrating from Azure Data Factory

Migration from Azure Data Factory

- Synapse Pipelines has the same code-base
- Behind the scenes all assets in ADF and Azure Synapse are stored as JSON files
- Export ADF artifacts as a Zip file
“Download support files”
- Import ADF artifacts into Azure Synapse
- Automate with PowerShell (Az.Synapse module), or import manually through the UI





Mentimeter Poll

Scan QR Code

or

Go to www.menti.com and use code

5333 4597



Lab 04: Migrating ADF assets

Objective: As a result of participating in this lab, you will be better able to understand how to migrate Azure Data Factory Assets into Azure Synapse Analytics.

What you will be doing: You will **work individually in your lab environment to complete a set of tasks** to import Azure Data Factory assets into Azure Synapse Analytics.

Total Activity Time: 60 minutes





Thank you

