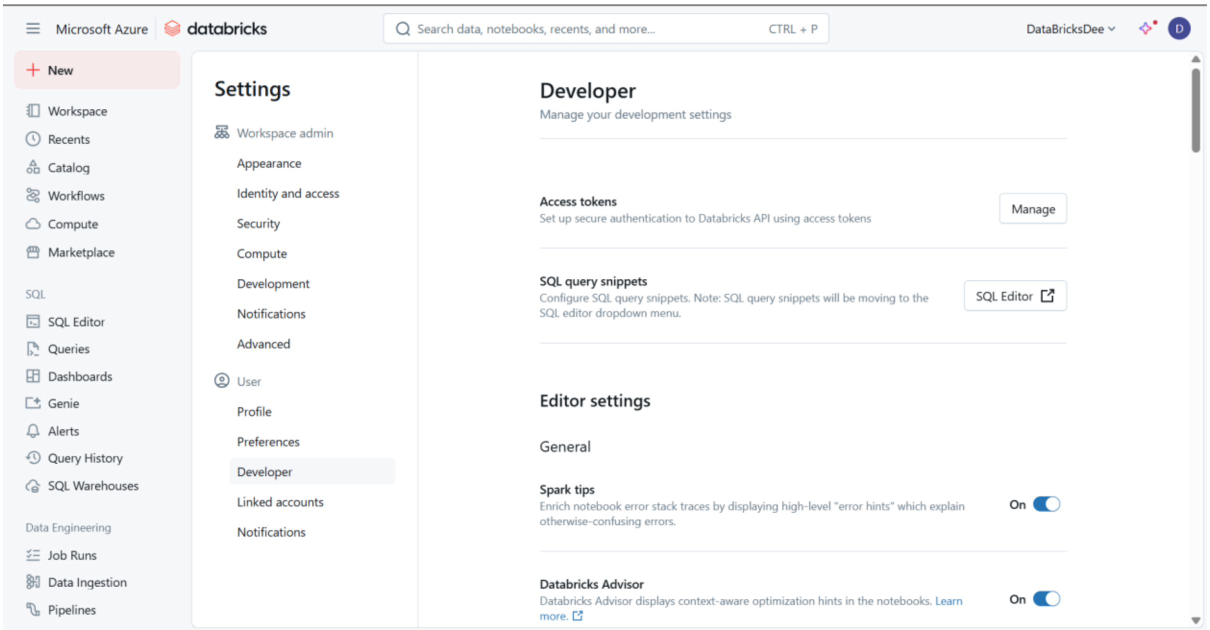
Connect notebook using synapse workspace

First, we have to generate Access token in Data bricks

For that we have to go to profile -> settings -> user -> Developer -> Access token -> Manage



Developer

Manage your development settings

Access tokens

Set up secure authentication to Databricks API using access tokens

Manage

SQL query snippets

Configure SQL query snippets. Note: SQL query snippets will be moving to the SQL editor dropdown menu.

SQL Editor

Editor settings

General

Spark tips

Enrich notebook error stack traces by displaying high-level "error hints" which explain otherwise-confusing errors.

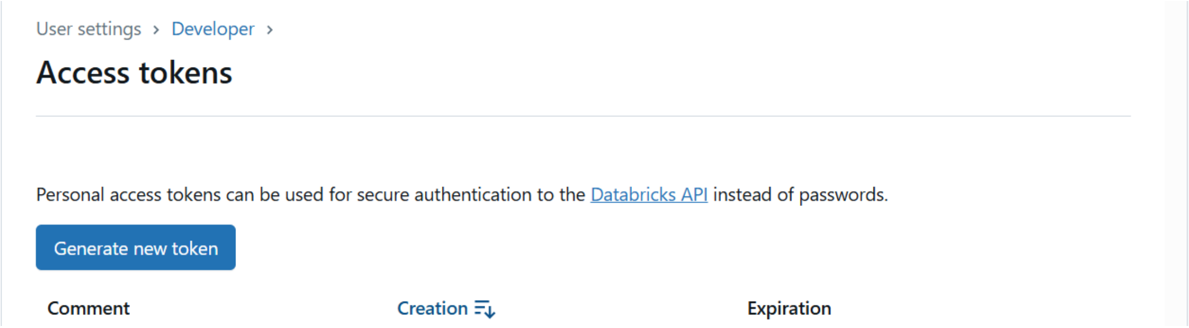
On

Databricks Advisor

Databricks Advisor displays context-aware optimization hints in the notebooks. Learn more.

On

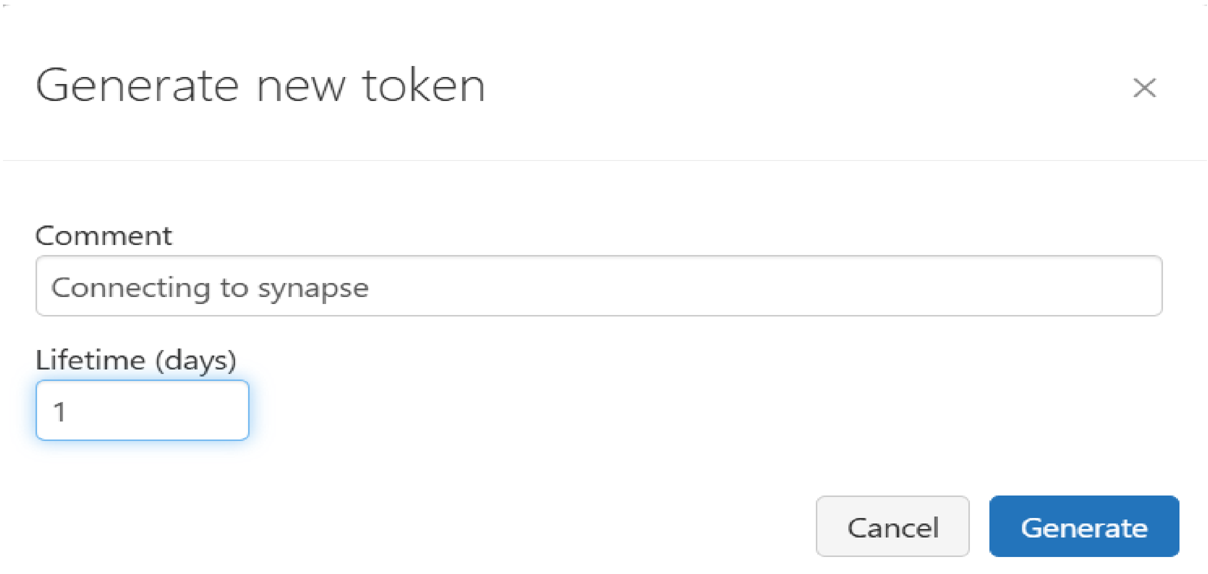
Manage - > Generate new token.



Access tokens

Personal access tokens can be used for secure authentication to the Databricks API instead of passwords.

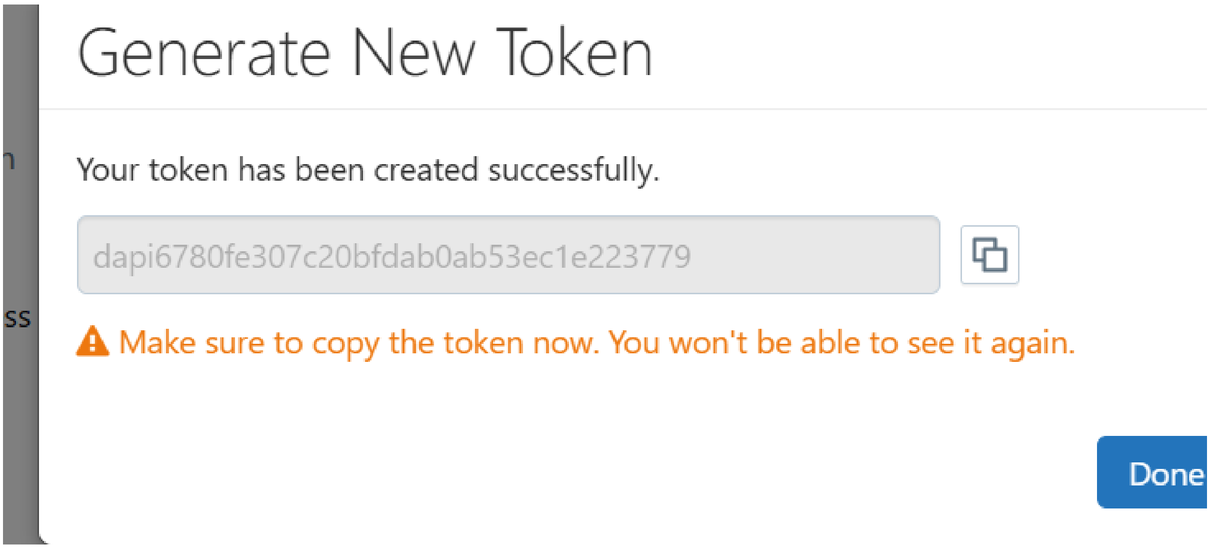
Comment	Creation	Expiration


Generate new token

Comment: Connecting to synapse

Lifetime (days): 1

Copy token: dapi6780fe307c20bfdab0ab53ec1e223779

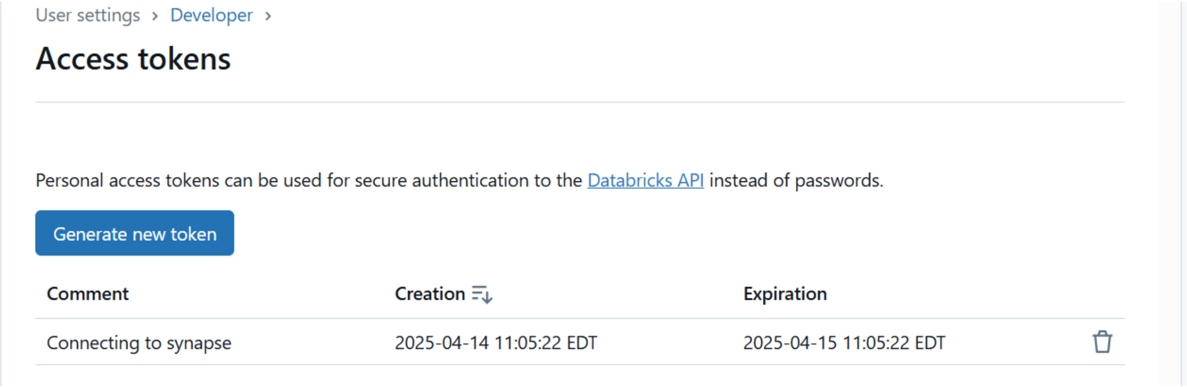


Generate New Token

Your token has been created successfully.

dapi6780fe307c20bfdab0ab53ec1e223779

⚠ Make sure to copy the token now. You won't be able to see it again.



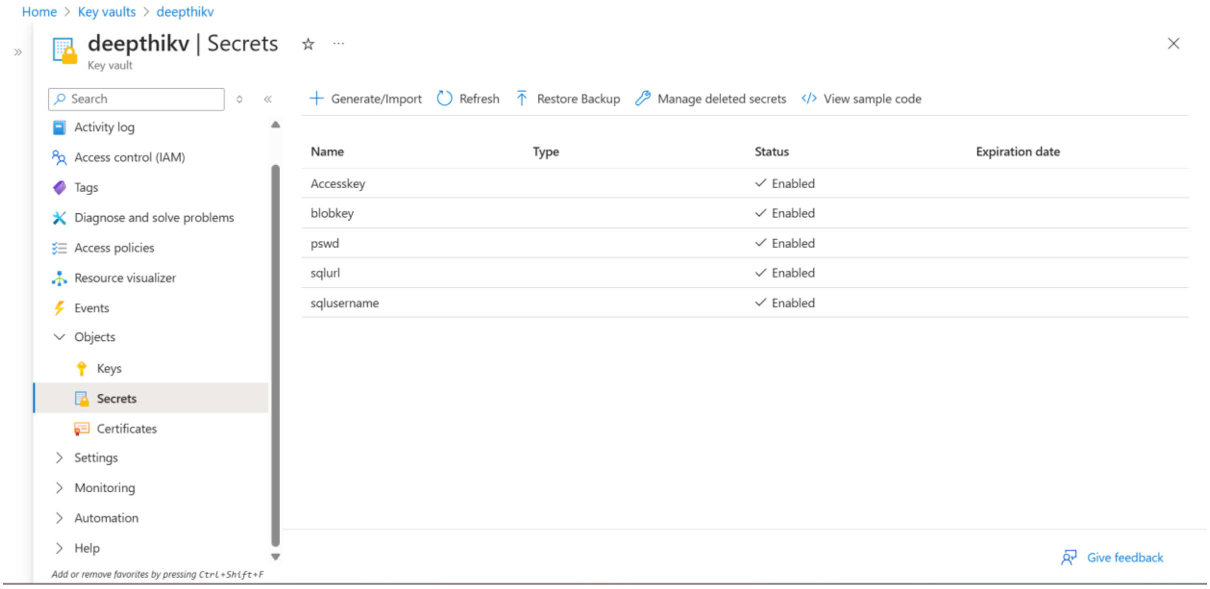
Access tokens

Personal access tokens can be used for secure authentication to the Databricks API instead of passwords.

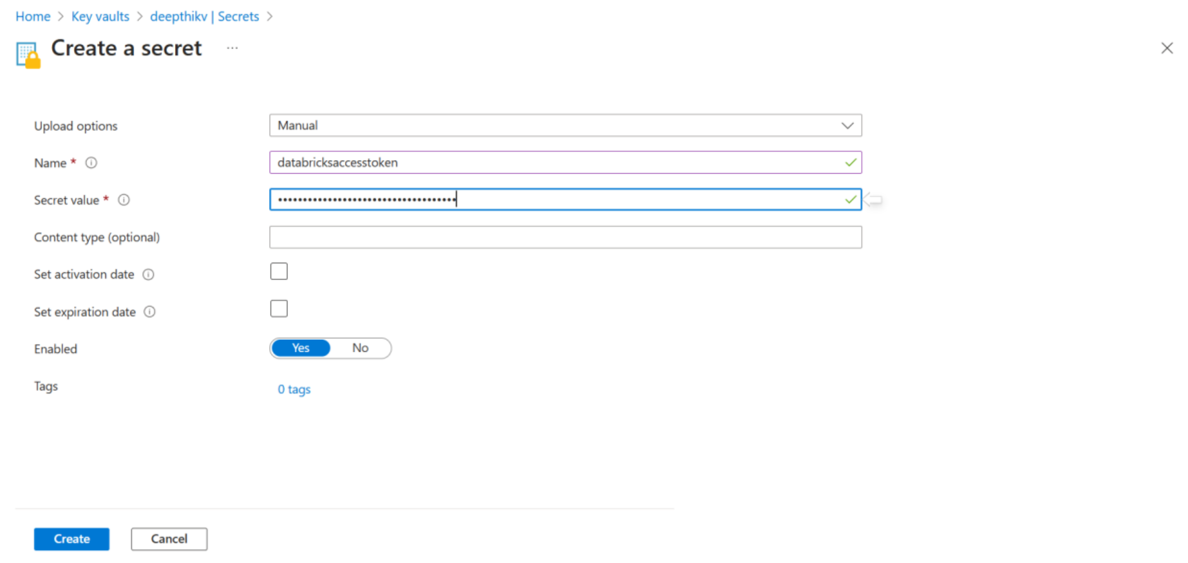
Comment	Creation	Expiration
Connecting to synapse	2025-04-14 11:05:22 EDT	2025-04-15 11:05:22 EDT

Now we have to create a secret key for access token in key vault

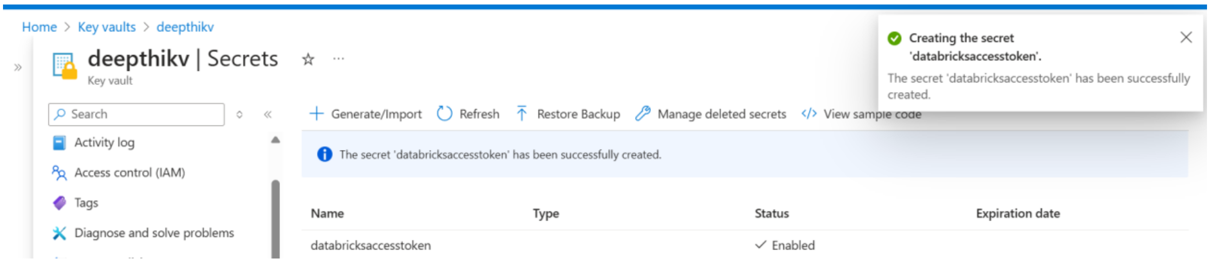
Go to key vault -> Objects -> Secrets -> Generate/ Import



Name	Type	Status	Expiration date
Accesskey		✓ Enabled	
blobkey		✓ Enabled	
pswd		✓ Enabled	
sqlurl		✓ Enabled	
sqlusername		✓ Enabled	



Upload options	Manual
Name *	databricksaccesstoken
Secret value *	••••••••••••••••••••••••••••••••••••
Content type (optional)	
Set activation date	□
Set expiration date	□
Enabled	● Yes ○ No
Tags	0 tags



Creating the secret 'databricksaccesstoken'.

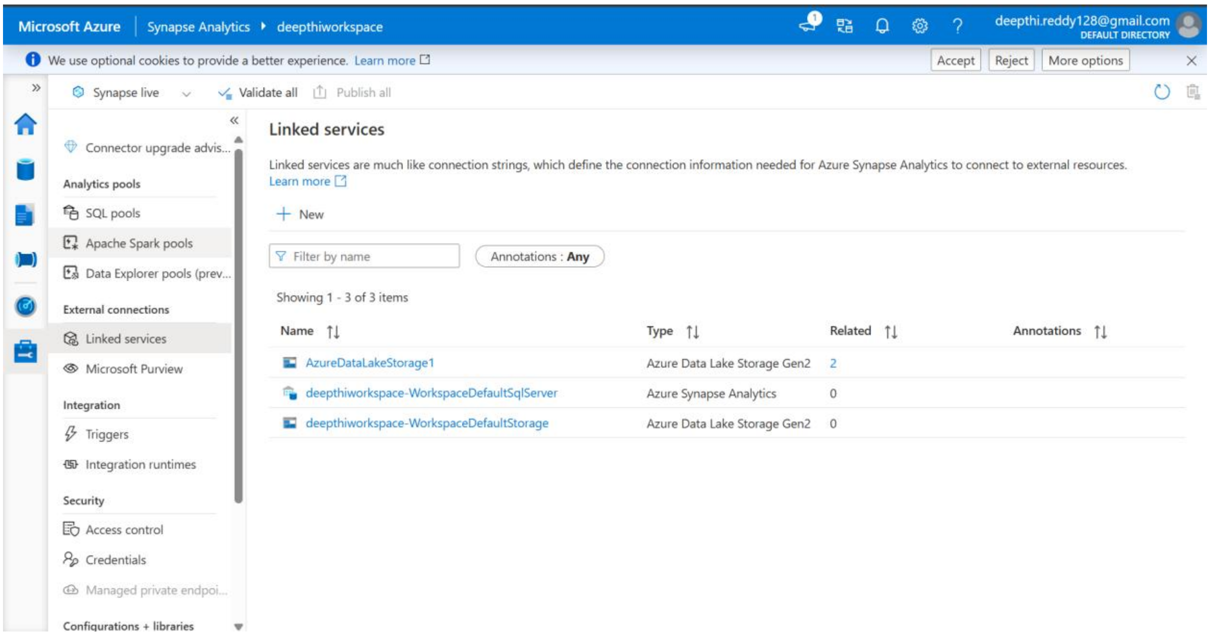
The secret 'databricksaccesstoken' has been successfully created.

The secret 'databricksaccesstoken' has been successfully created.

Name	Type	Status	Expiration date
databricksaccesstoken		✓ Enabled	

Now we have to create linked services for both Key vault and data bricks.

Go to synapse -> Manage -> Linked services -> New



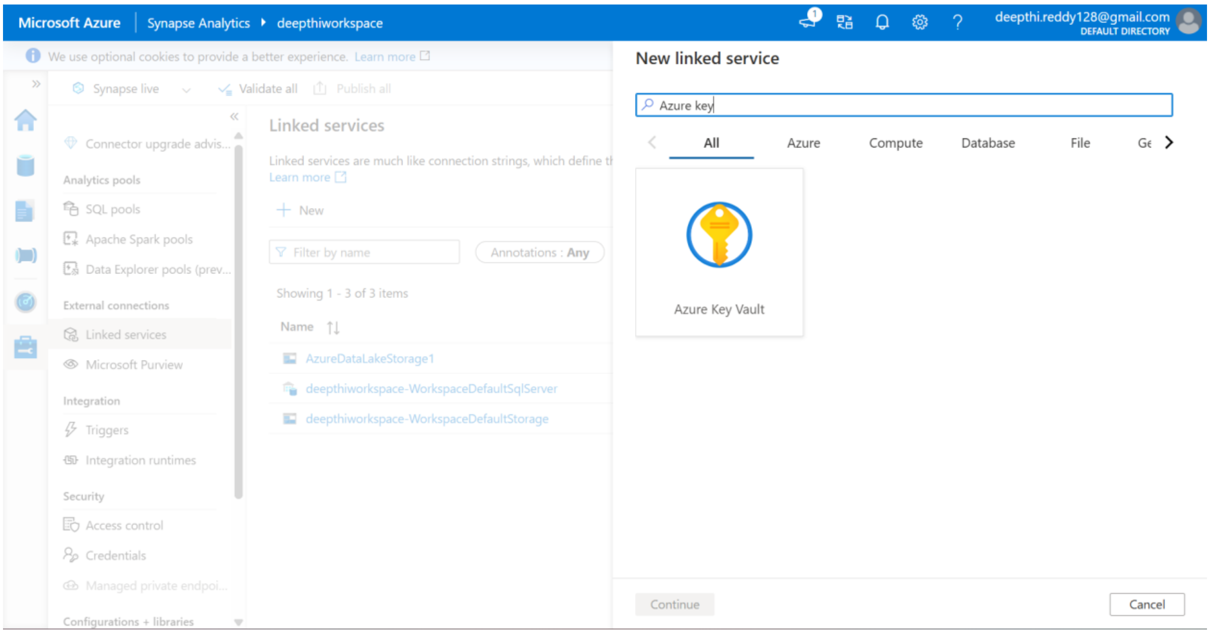
Linked services

Linked services are much like connection strings, which define the connection information needed for Azure Synapse Analytics to connect to external resources. Learn more

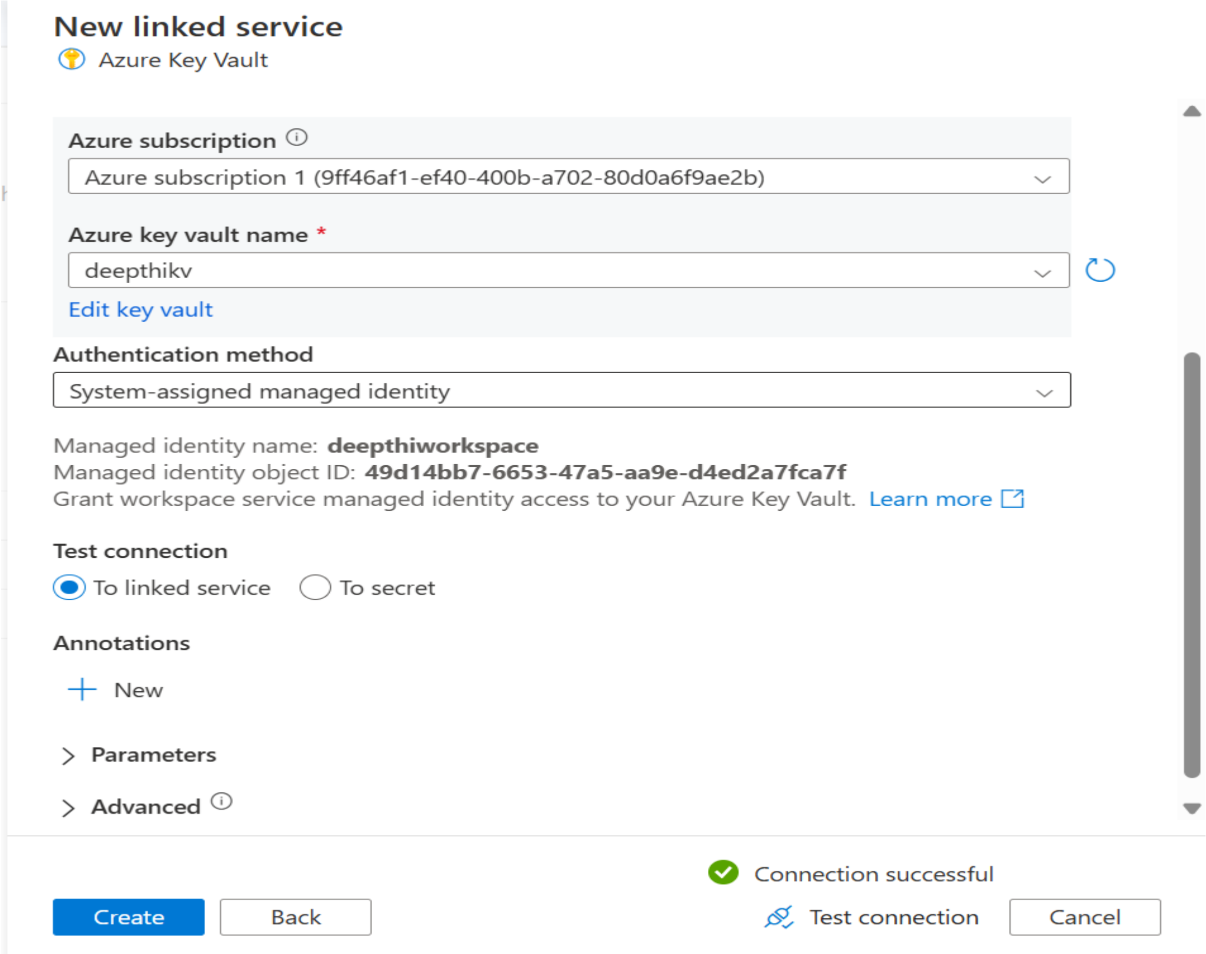
Showing 1 - 3 of 3 items

Name	Type	Related	Annotations
AzureDataLakeStorage1	Azure Data Lake Storage Gen2	2	
deepthiworkspace-WorkspaceDefaultSqlServer	Azure Synapse Analytics	0	
deepthiworkspace-WorkspaceDefaultStorage	Azure Data Lake Storage Gen2	0	

Once we click on new, search for Azure key vault as we have to first create linked services for Azure key vault.



Once you select this option, we have provided all the details, about subscription, what key vault we have to use and then test connection.



New linked service

Azure Key Vault

Azure subscription: Azure subscription 1 (9ff46af1-ef40-400b-a702-80d0a6f9ae2b)

Azure key vault name *: deepthikv

Edit key vault

Authentication method: System-assigned managed identity

Managed identity name: **deepthiworkspace**
Managed identity object ID: **49d14bb7-6653-47a5-aa9e-d4ed2a7fca7f**
Grant workspace service managed identity access to your Azure Key Vault. Learn more

Test connection: ● To linked service ○ To secret

Annotations

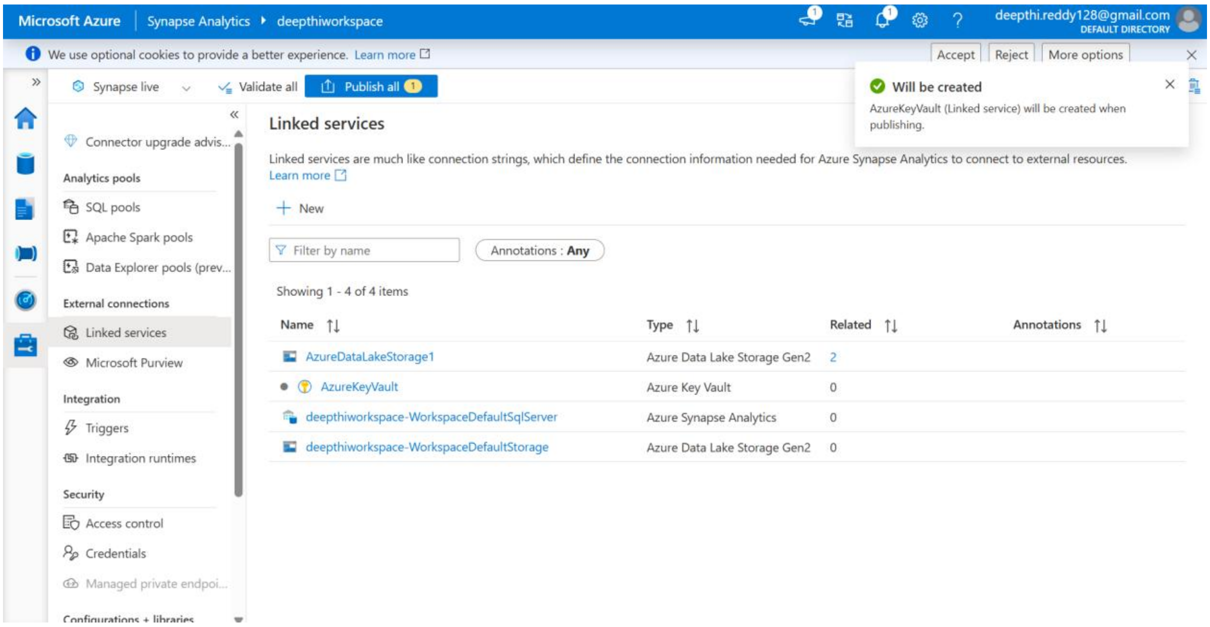
+ New

> Parameters

> Advanced

✓ Connection successful

Once connection is successful, create the linked service.



We use optional cookies to provide a better experience. Learn more

Will be created

AzureKeyVault (Linked service) will be created when publishing.

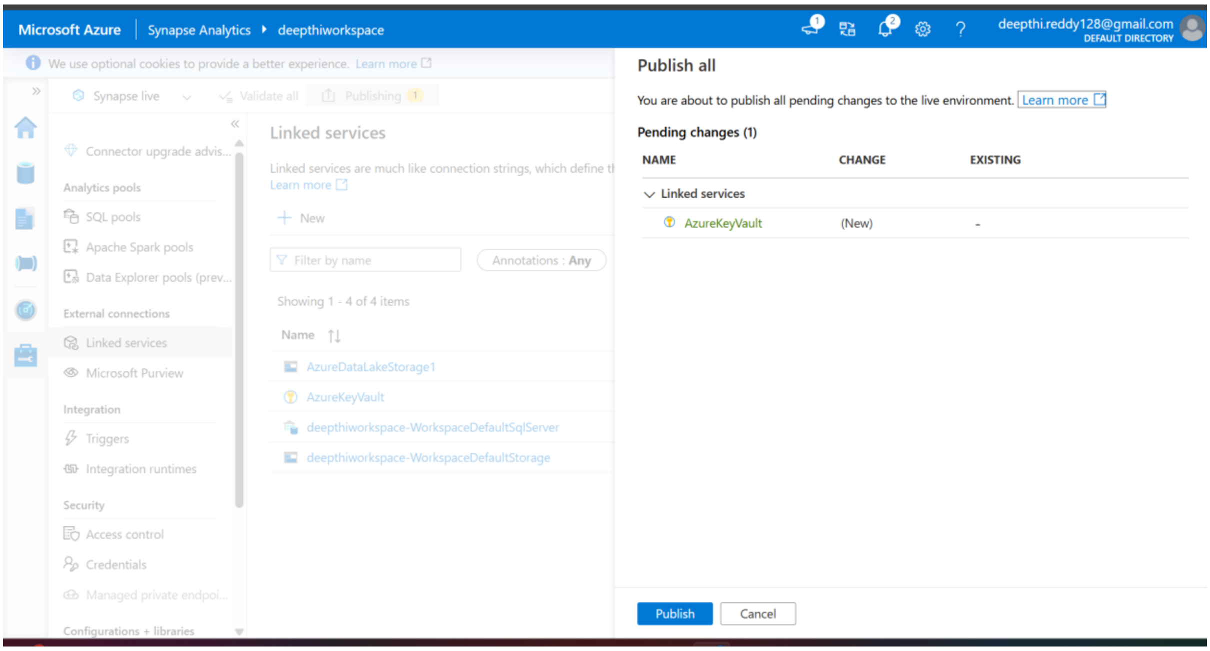
Linked services

Linked services are much like connection strings, which define the connection information needed for Azure Synapse Analytics to connect to external resources. Learn more

Showing 1 - 4 of 4 items

Name	Type	Related	Annotations
AzureDataLakeStorage1	Azure Data Lake Storage Gen2	2	
● AzureKeyVault	Azure Key Vault	0	
deepthiworkspace-WorkspaceDefaultSqlServer	Azure Synapse Analytics	0	
deepthiworkspace-WorkspaceDefaultStorage	Azure Data Lake Storage Gen2	0	

Once it is created publish the changes for it to be saved

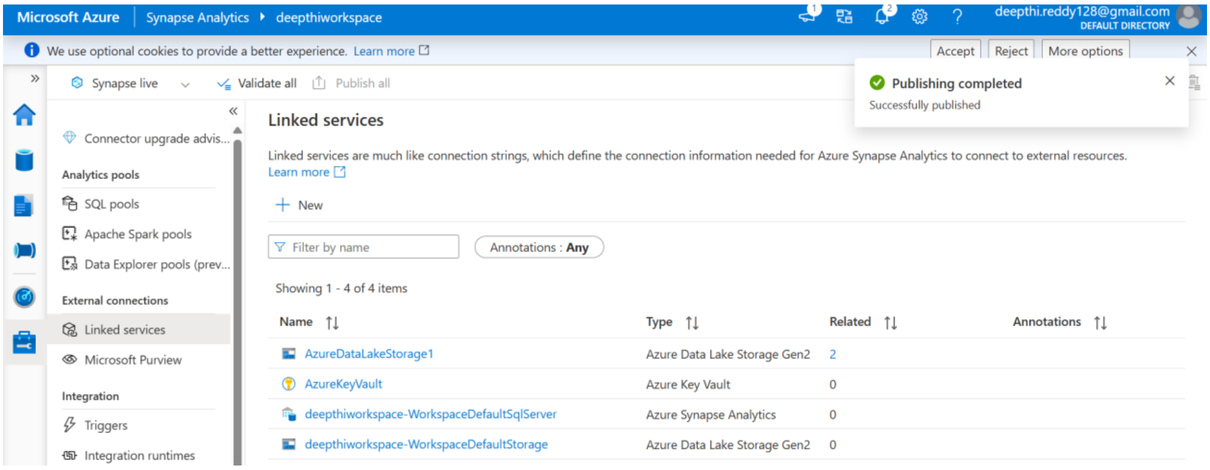


Publish all

You are about to publish all pending changes to the live environment. Learn more

Pending changes (1)

NAME	CHANGE	EXISTING
Linked services		
AzureKeyVault	(New)	-



We use optional cookies to provide a better experience. Learn more

Publishing completed

Successfully published

Linked services

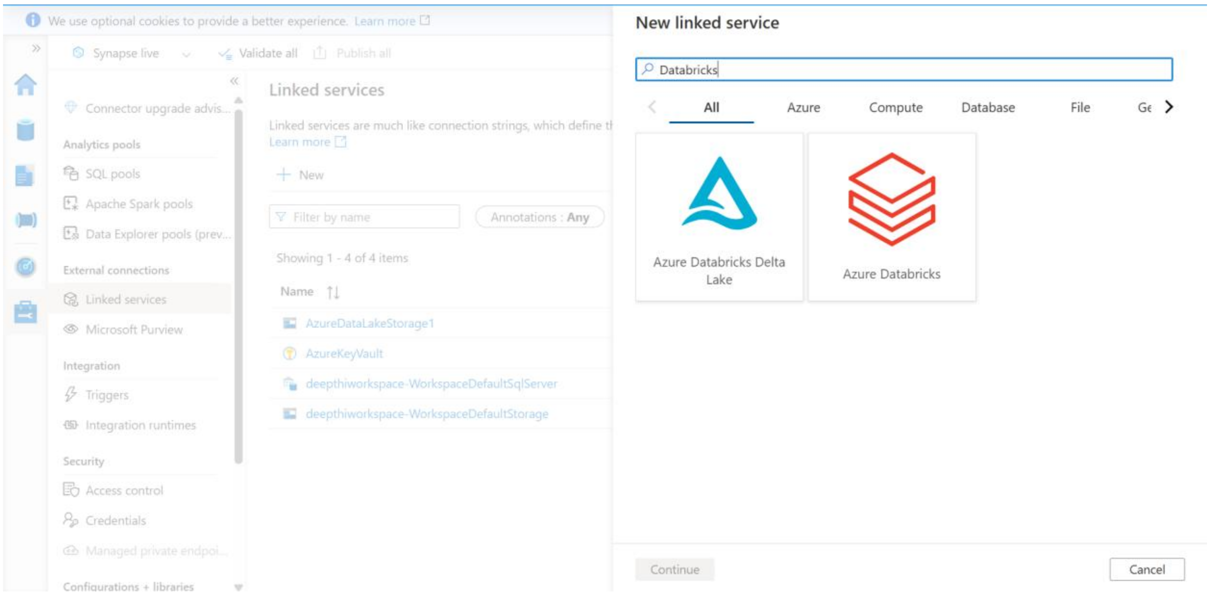
Linked services are much like connection strings, which define the connection information needed for Azure Synapse Analytics to connect to external resources. Learn more

Showing 1 - 4 of 4 items

Name	Type	Related	Annotations
AzureDataLakeStorage1	Azure Data Lake Storage Gen2	2	
AzureKeyVault	Azure Key Vault	0	
deepthiworkspace-WorkspaceDefaultSqlServer	Azure Synapse Analytics	0	
deepthiworkspace-WorkspaceDefaultStorage	Azure Data Lake Storage Gen2	0	

Now create Data bricks linked service

Click on new and search for Databricks.



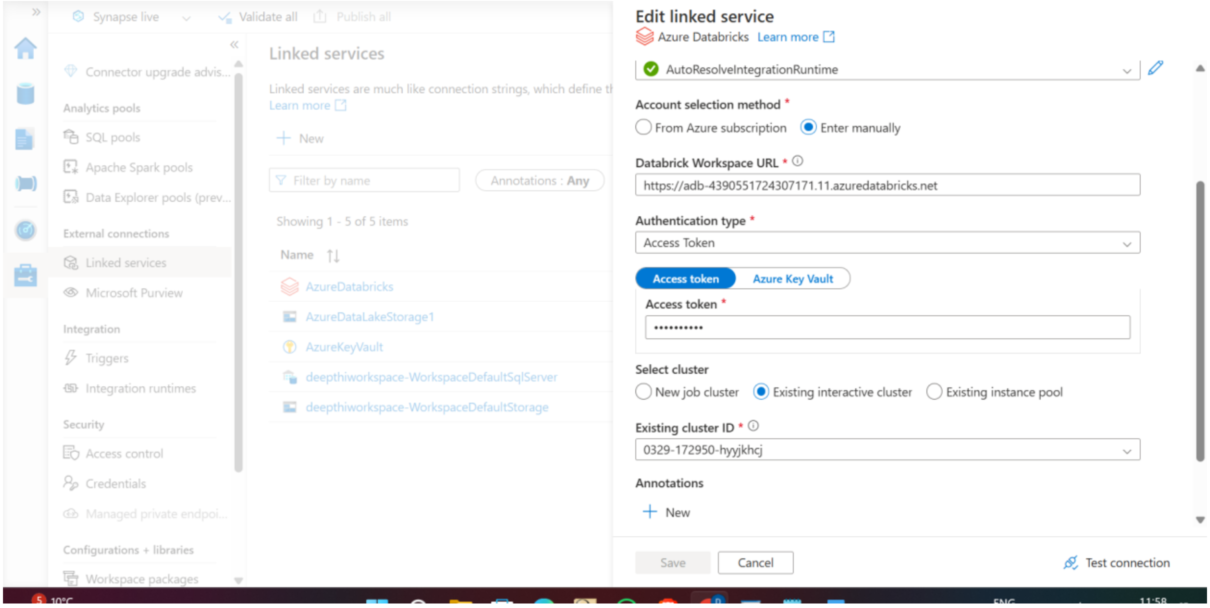
New linked service

Databricks

Azure Databricks Delta Lake

Azure Databricks

Provide all required details and test connection



Edit linked service

Azure Databricks Learn more

AutoResolveIntegrationRuntime

Account selection method *
○ From Azure subscription ● Enter manually

Databrick Workspace URL *
https://adb-4390551724307171.11.azuredatabricks.net

Authentication type *
Access Token

Access token | Azure Key Vault

Access token *
••••••••••

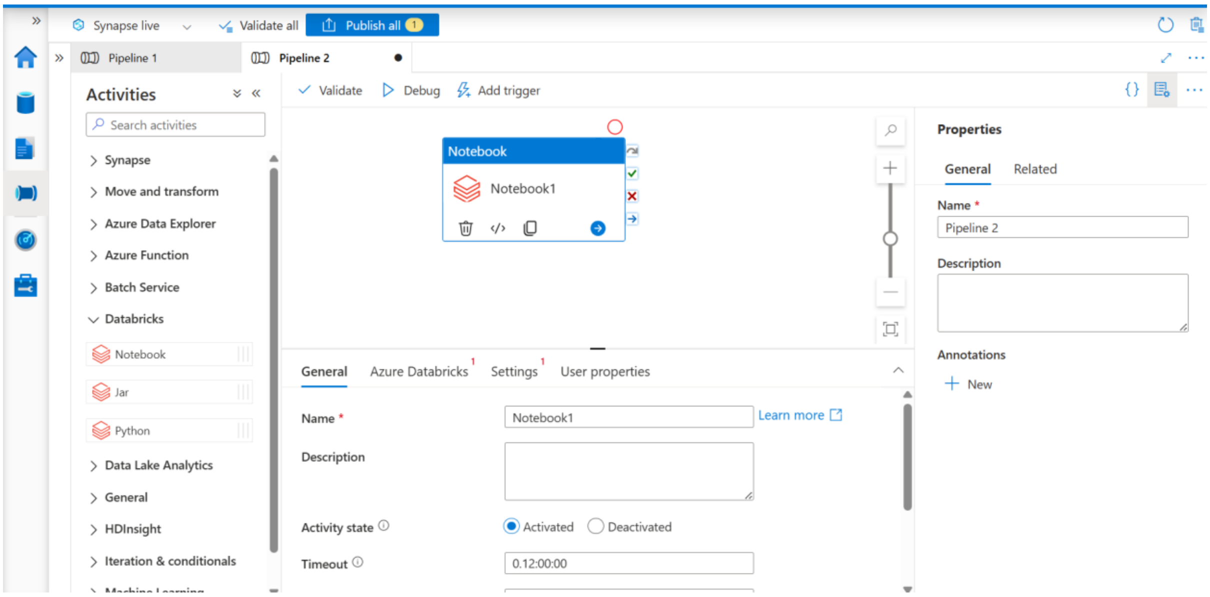
Select cluster
○ New job cluster ● Existing interactive cluster ○ Existing instance pool

Existing cluster ID *
0329-172950-hyyjkhcj

Annotations
New

After linked services are created, create a pipeline, to run the notebook.

Go to Integrate -> new Pipeline -> drag and drop notebook from data bricks



Name * Notebook1

Description

Activity state ● Activated ○ Deactivated

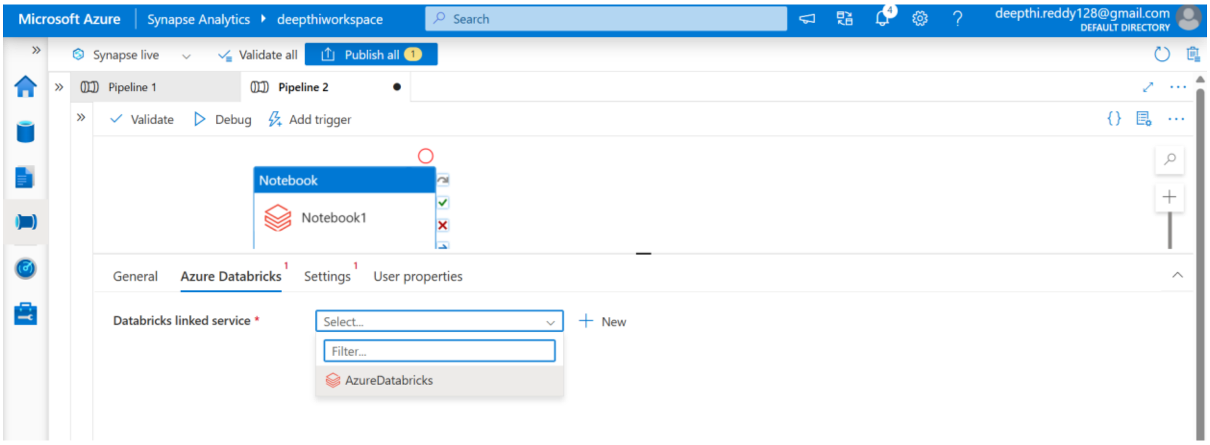
Timeout 0.12:00:00

Name * Pipeline 2

Description

Now go to Azure data bricks section and choose linked service.

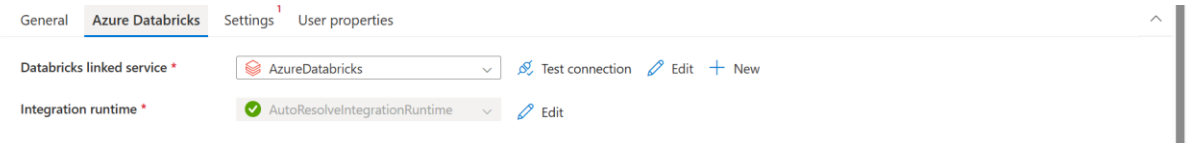
As we already created a linked service, we can directly use it.



Databricks linked service * Select...

Filter...

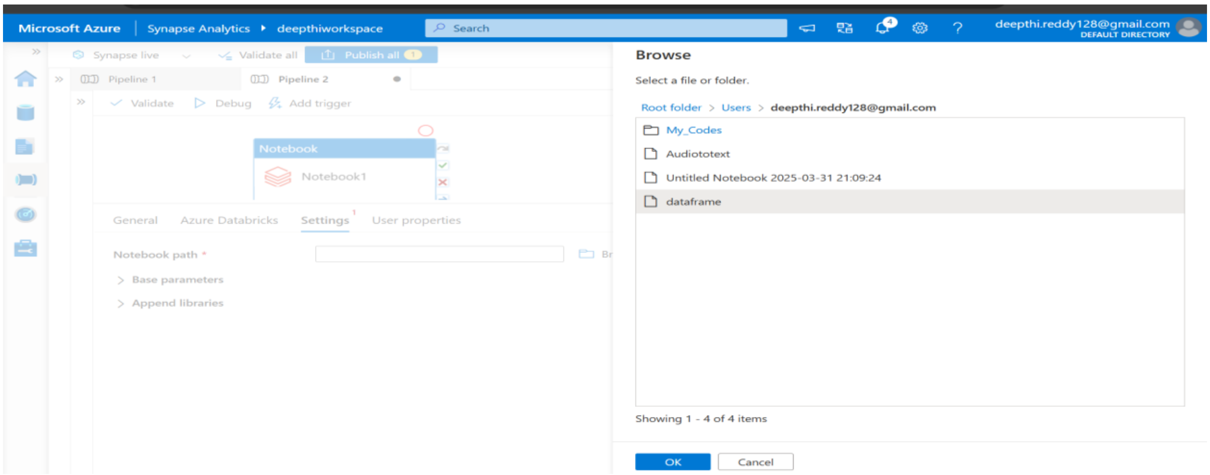
AzureDatabricks



Databricks linked service * AzureDatabricks

Integration runtime * AutoResolveIntegrationRuntime

Now go to settings and choose which note book we want to run.



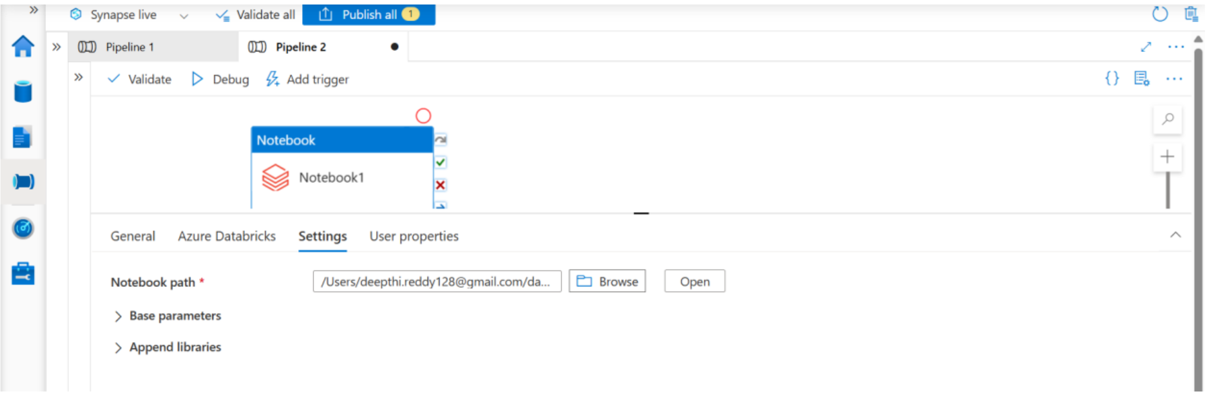
Browse

Select a file or folder.

Root folder > Users > deepthi.reddy128@gmail.com

- My_Codes
- Audiototext
- Untitled Notebook 2025-03-31 21:09:24
- dataframe

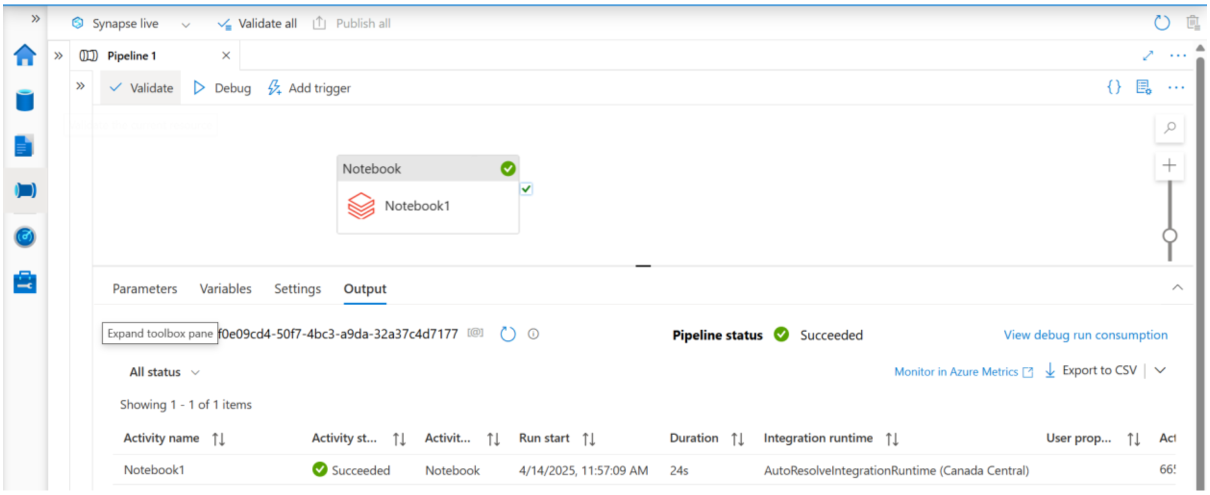
Showing 1 - 4 of 4 items



Notebook path * /Users/deepthi.reddy128@gmail.com/da...

Validate and run pipeline.

If cluster is on it will take less time, but if cluster is off first, cluster will start and once it is up and running then it will run the notebook.



f0e09cd4-50f7-4bc3-a9da-32a37c4d7177

Pipeline status ✓ Succeeded

Showing 1 - 1 of 1 items

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime	User prop...	Act
Notebook1	✓ Succeeded	Notebook	4/14/2025, 11:57:09 AM	24s	AutoResolveIntegrationRuntime (Canada Central)		66!

Pipeline ran successfully.

Explore medallion architecture and write the best way of organizing data

What is Medallion Architecture?

The Medallion Architecture is a **modular approach** to building data lakes using **Delta Lake** (often used with Azure Synapse or Databricks), promoting **data quality, governance, and reuse**.

It splits the data lifecycle into 3 layers:

1. Bronze Layer (Raw Data)

Purpose: Store raw, unfiltered data from source systems.

- Format: CSV, JSON, Avro, or raw Delta
- Minimal transformation (e.g., basic parsing)
- Used for auditing, reprocessing, or backup

2. Silver Layer (Cleansed & Transformed)

Purpose: Cleaned and structured data for analytical use.

- Joins, filters, type conversions
- Deduplication, null handling
- Typically stored in Delta Lake format for reliability

3. Gold Layer (Business/Curated Data)

Purpose: Final layer for analytics, ML, BI, and reporting.

- Aggregated KPIs
- Business-ready metrics
- Denormalized, fast access structure

The **Medallion Architecture** provides a **layered approach** to structuring data in a data lake or Delta Lake environment. It promotes **data quality, scalability, and clarity** by segmenting data into **bronze, silver, and gold layers**. Each layer serves a distinct purpose and follows best practices in terms of storage organization, access control, and data format.

Explain RDD vs Data frames

RDD (Resilient Distributed Dataset):

It is low-level distributed data abstraction, which supports Java, Scala, Python languages.

Characteristics:

- Immutable, distributed collection of objects
- Can perform transformations (e.g., map, filter) and actions (count, collect)
- Offers **fine-grained control** over data

- Best for **unstructured data**, complex logic, or **low-level transformations**

Downsides:

- **No optimization**: No automatic query optimization (no Catalyst or Tungsten)
- **More code** to perform operations
- Less efficient than Data Frames
- Performance is low

Data Frame:

It is high-level abstraction over structured data, and this supports Java, Scala, Python and R languages.

Characteristics:

- Distributed **table-like** structure with rows and named columns (like a SQL table)
- Built on top of RDDs
- Supports **SQL-like operations** using DSL or SQL queries
- Backed by **Catalyst optimizer** (logical + physical optimization)
- Best for **structured/semi-structured data**, analytics, and aggregations

Benefits:

- **Optimized performance**
- **Less code** needed
- Supports various data sources: Parquet, JSON, JDBC, Hive, etc.
- Can interoperate with SQL and BI tools

Explain narrow and wide transformations

Narrow Transformations

Narrow transformations are those where each partition of the parent RDD is used by at most one partition of the child RDD. In simple terms, data from a single input partition only maps to one output partition. These transformations do not require data to be shuffled across the network, making them faster and more efficient.

Examples: map(), filter(), flatMap(), union(), sample()

Since there's no data movement between nodes, narrow transformations allow Spark to pipeline operations and execute them more efficiently.

Wide Transformations

Wide transformations are those where multiple partitions of the parent RDD may contribute to a single partition of the child RDD. This requires data shuffling, where Spark redistributes data across the network. Shuffles are expensive operations as they involve disk I/O, data serialization, and network latency.

Examples: groupByKey(), reduceByKey(), join(), distinct()

These transformations are necessary for tasks that require data aggregation or combining datasets but can significantly impact performance if not handled properly.