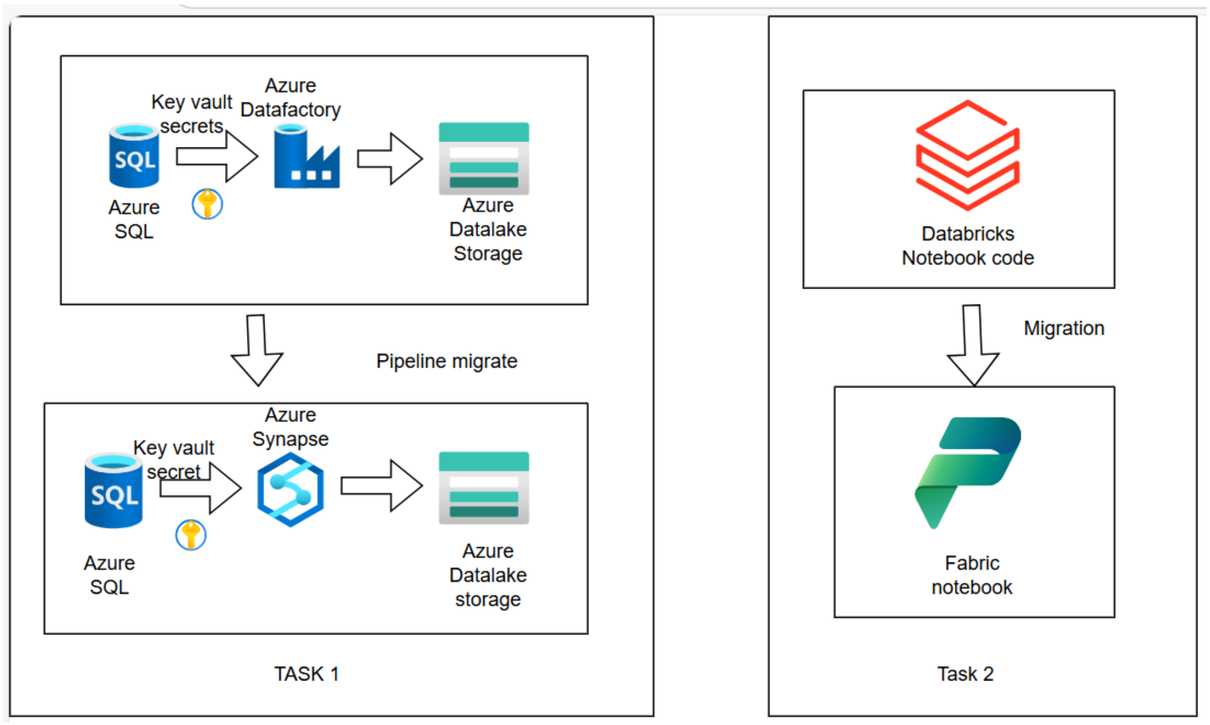
Bootcamp Project 5

Project Title: Migrating pipelines from ADF to Synapse

Problem Statement:

Develop ADF pipelines (copy activity, foreach loop, look up) and migrate the pipelines, datasets, linked services to Azure Synapse.

Architecture Diagram:



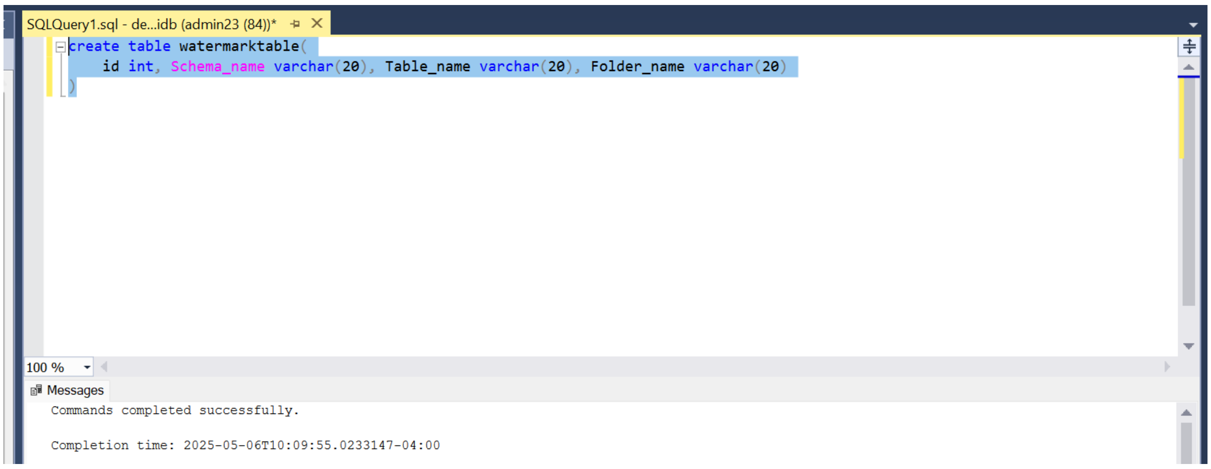
Tools & Technologies:

- Azure Data factory
- Azure Synapse
- Azure SQL
- Azure data lake Gen2
- Azure key vault
- Draw.io
- Databricks
- Fabric

Task 1:

Migrating pipeline from ADF to Azure synapse.

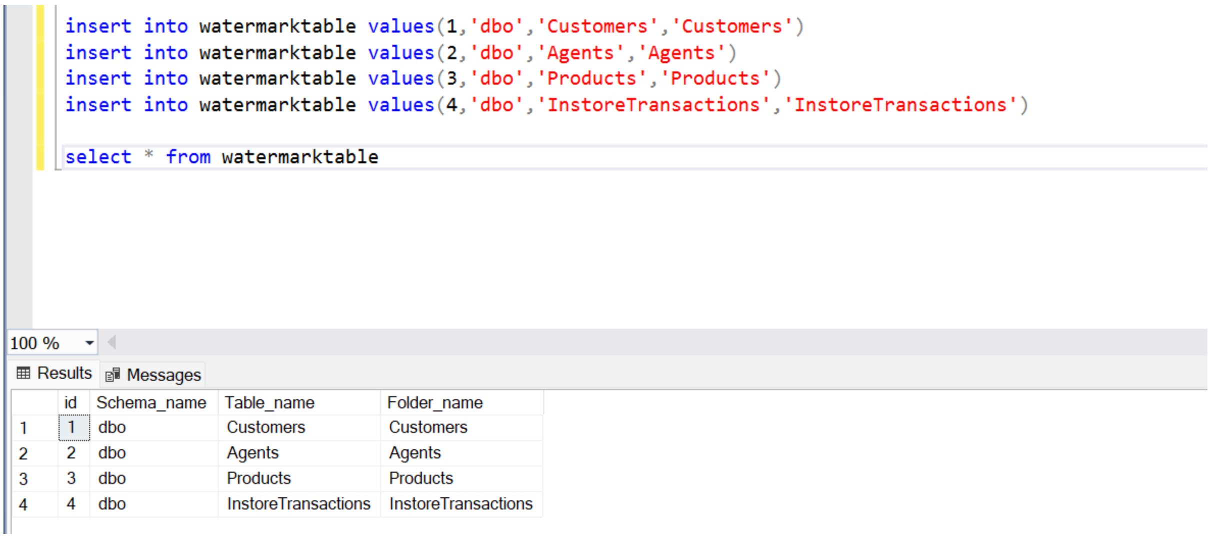
Creating a meta_table



```sql
create table watermarktable(
    id int, Schema_name varchar(20), Table_name varchar(20), Folder_name varchar(20)
)
```

```
Commands completed successfully.

Completion time: 2025-05-06T10:09:55.0233147-04:00
```

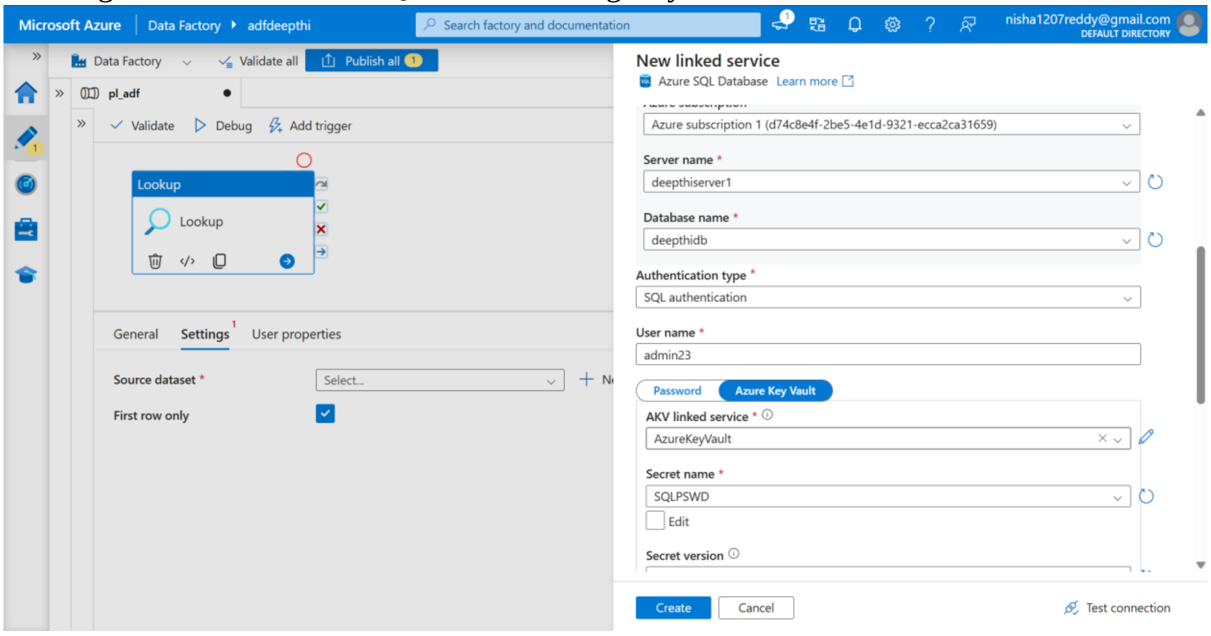
Values inserted into table



```sql
insert into watermarktable values(1,'dbo','Customers','Customers')
insert into watermarktable values(2,'dbo','Agents','Agents')
insert into watermarktable values(3,'dbo','Products','Products')
insert into watermarktable values(4,'dbo','InstoreTransactions','InstoreTransactions')

select * from watermarktable
```

	id	Schema_name	Table_name	Folder_name
1	1	dbo	Customers	Customers
2	2	dbo	Agents	Agents
3	3	dbo	Products	Products
4	4	dbo	InstoreTransactions	InstoreTransactions

Creating linked services for SQL database using Key vault.



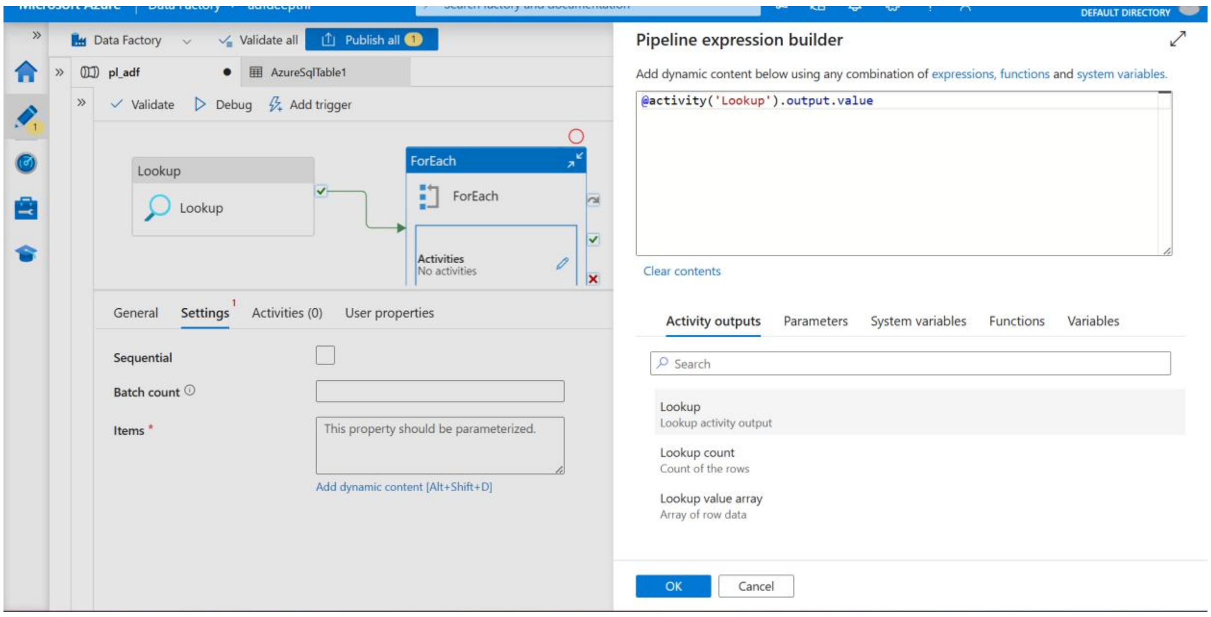
New linked service

Azure SQL Database Learn more

Source dataset *	Select...
First row only	✓

Azure subscription	Azure subscription 1 (d74c8e4f-2be5-4e1d-9321-ecca2ca31659)
Server name *	deepthiserver1
Database name *	deepthidb
Authentication type *	SQL authentication
User name *	admin23
Password / Azure Key Vault	Azure Key Vault
AKV linked service *	AzureKeyVault
Secret name *	SQLPSWD
Edit	☐
Secret version	

Connecting lookup to foreach activity



Sequential

Batch count

Items *: This property should be parameterized.

Add dynamic content [Alt+Shift+D]

Pipeline expression builder

Add dynamic content below using any combination of expressions, functions and system variables.

```
@activity('Lookup').output.value
```

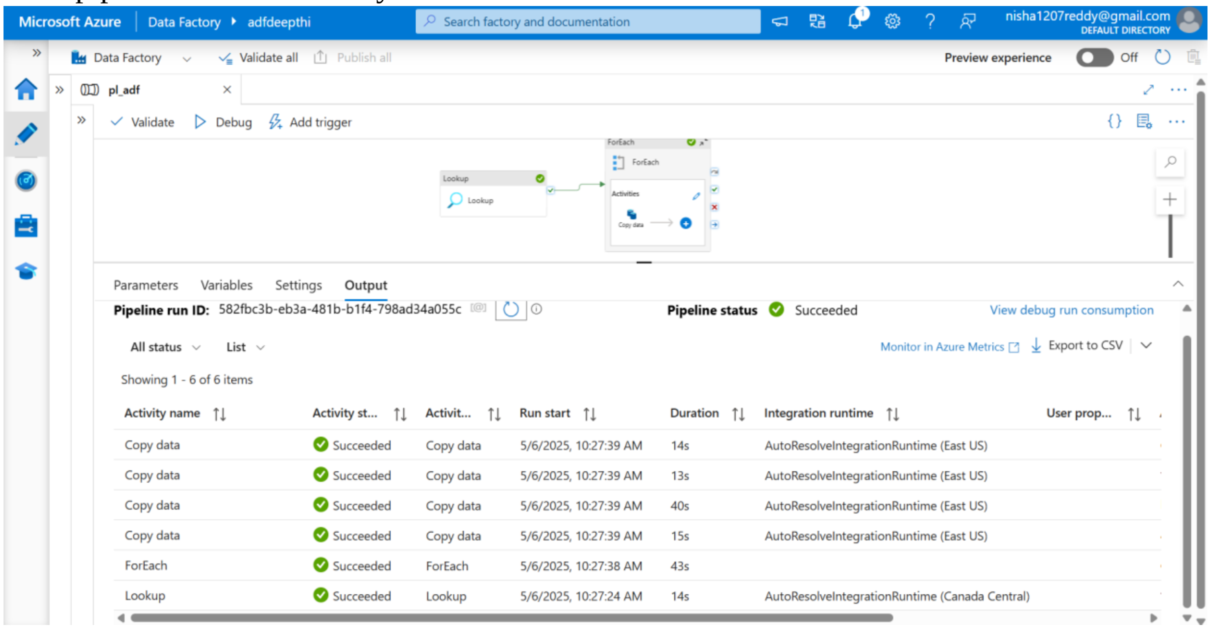
Clear contents

Lookup — Lookup activity output

Lookup count — Count of the rows

Lookup value array — Array of row data

ADF pipeline ran successfully



Pipeline run ID: 582fbc3b-eb3a-481b-b1f4-798ad34a055c

Pipeline status: Succeeded

View debug run consumption

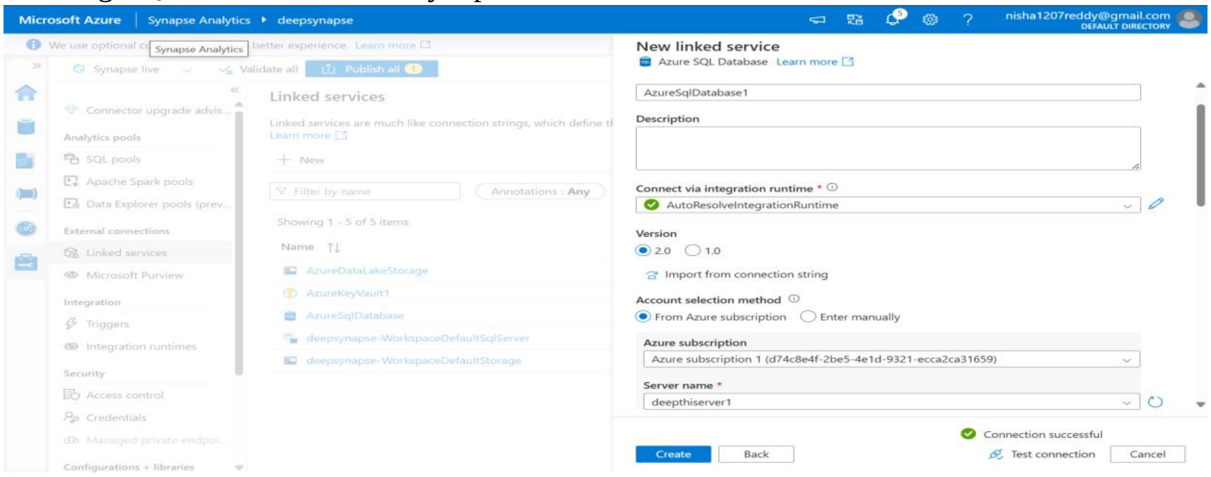
All status · List · Monitor in Azure Metrics · Export to CSV

Showing 1 - 6 of 6 items

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime	User prop...
Copy data	Succeeded	Copy data	5/6/2025, 10:27:39 AM	14s	AutoResolveIntegrationRuntime (East US)	
Copy data	Succeeded	Copy data	5/6/2025, 10:27:39 AM	13s	AutoResolveIntegrationRuntime (East US)	
Copy data	Succeeded	Copy data	5/6/2025, 10:27:39 AM	40s	AutoResolveIntegrationRuntime (East US)	
Copy data	Succeeded	Copy data	5/6/2025, 10:27:39 AM	15s	AutoResolveIntegrationRuntime (East US)	
ForEach	Succeeded	ForEach	5/6/2025, 10:27:38 AM	43s		
Lookup	Succeeded	Lookup	5/6/2025, 10:27:24 AM	14s	AutoResolveIntegrationRuntime (Canada Central)	

Synapse Pipeline:

Creating SQL linked service in Synapse



Linked services are much like connection strings, which define t... Learn more

Showing 1 - 5 of 5 items

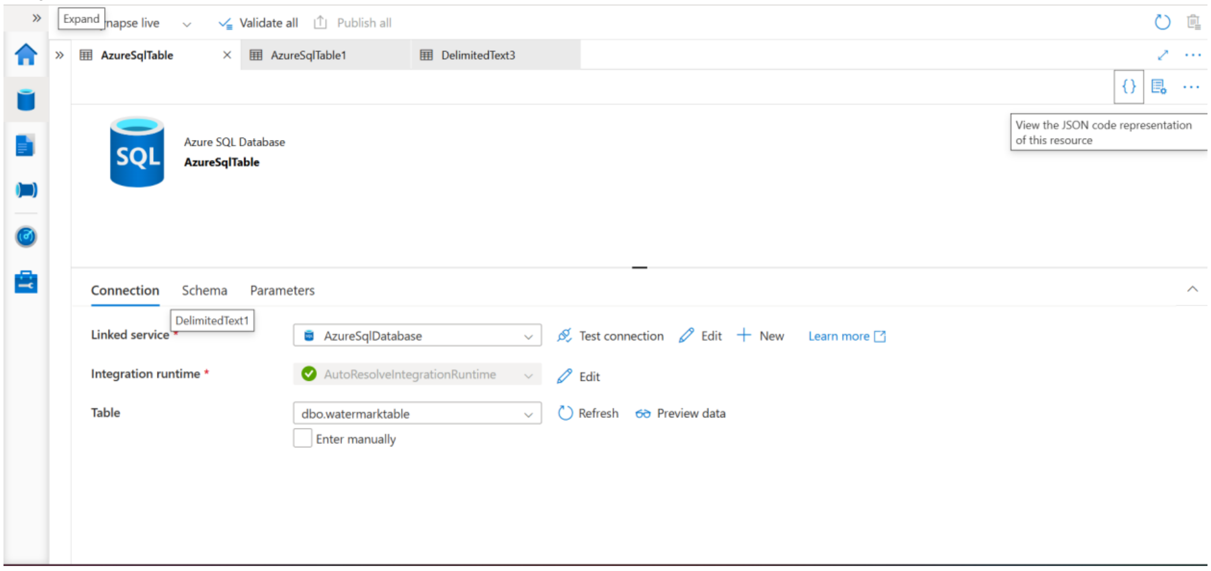
Name
AzureDataLakeStorage
AzureKeyVault1
AzureSqlDatabase
deepsynapse-WorkspaceDefaultSqlServer
deepsynapse-WorkspaceDefaultStorage

New linked service

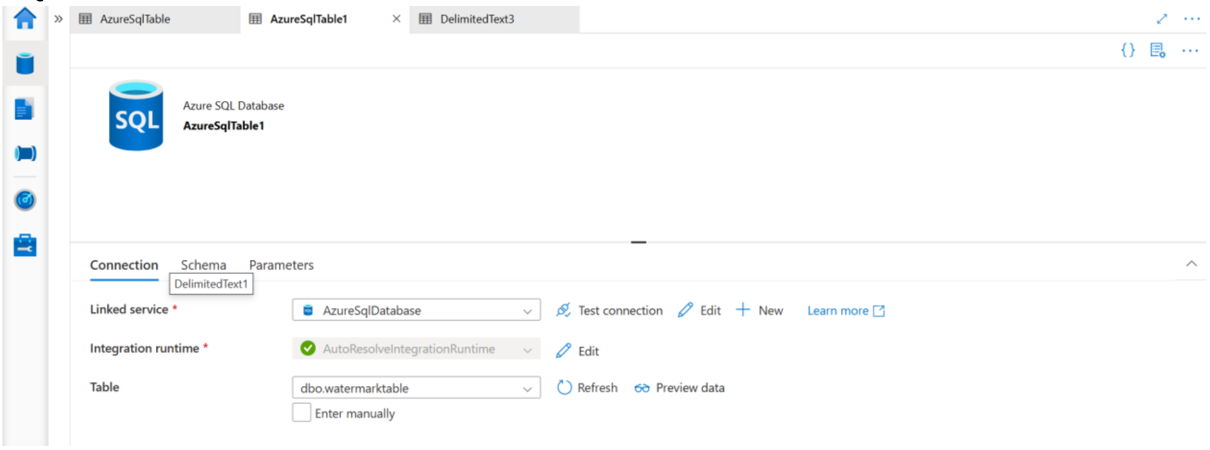
Azure SQL Database Learn more

Name	AzureSqlDatabase1
Description	
Connect via integration runtime *	AutoResolveIntegrationRuntime
Version	[✓ 2.0] 1.0
	Import from connection string
Account selection method	[✓ From Azure subscription] Enter manually
Azure subscription	Azure subscription 1 (d74c8e4f-2be5-4e1d-9321-ecca2ca31659)
Server name *	deepthiserver1

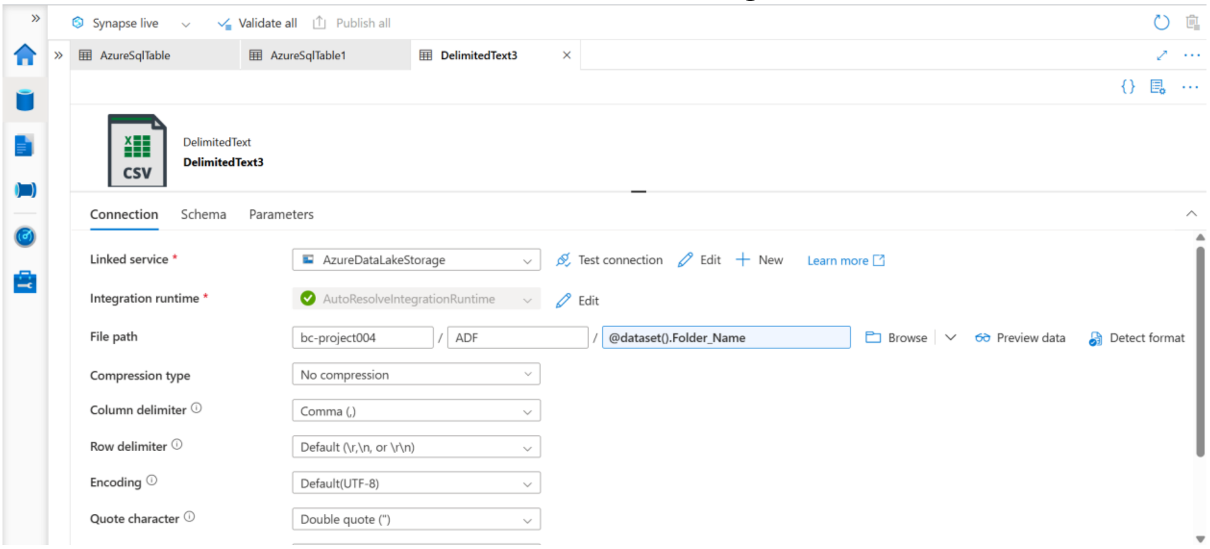
Connection successful

Creating Data sets which are in Data factory in Synapse as well.
SQL Dataset 1


Linked service * AzureSqlDatabase
Integration runtime * AutoResolveIntegrationRuntime
Table dbo.watermarktable
Enter manually

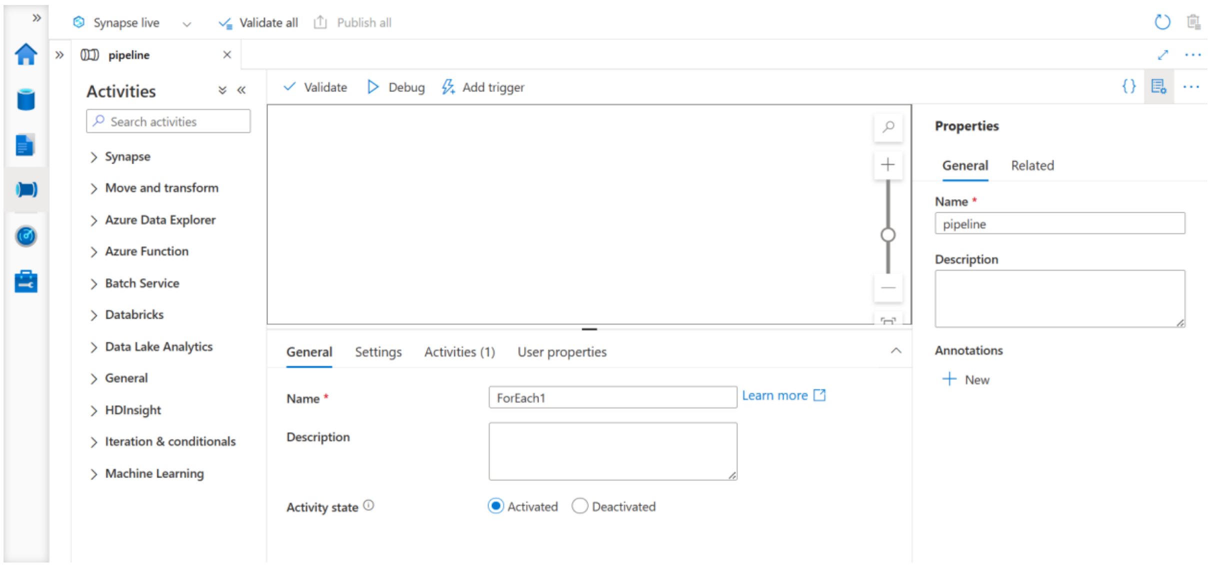
SQL Dataset 2


Linked service * AzureSqlDatabase
Integration runtime * AutoResolveIntegrationRuntime
Table dbo.watermarktable
Enter manually

Delimited Text 3 dataset for Azure Data Lake Gen2 Storage.


Linked service * AzureDataLakeStorage
Integration runtime * AutoResolveIntegrationRuntime
File path bc-project004 / ADF / @dataset().Folder_Name
Compression type No compression
Column delimiter Comma (,)
Row delimiter Default (\r,\n, or \r\n)
Encoding Default(UTF-8)
Quote character Double quote (")

Now created a pipeline in Synapse and named it same as pipeline in ADF.



Name * ForEach1

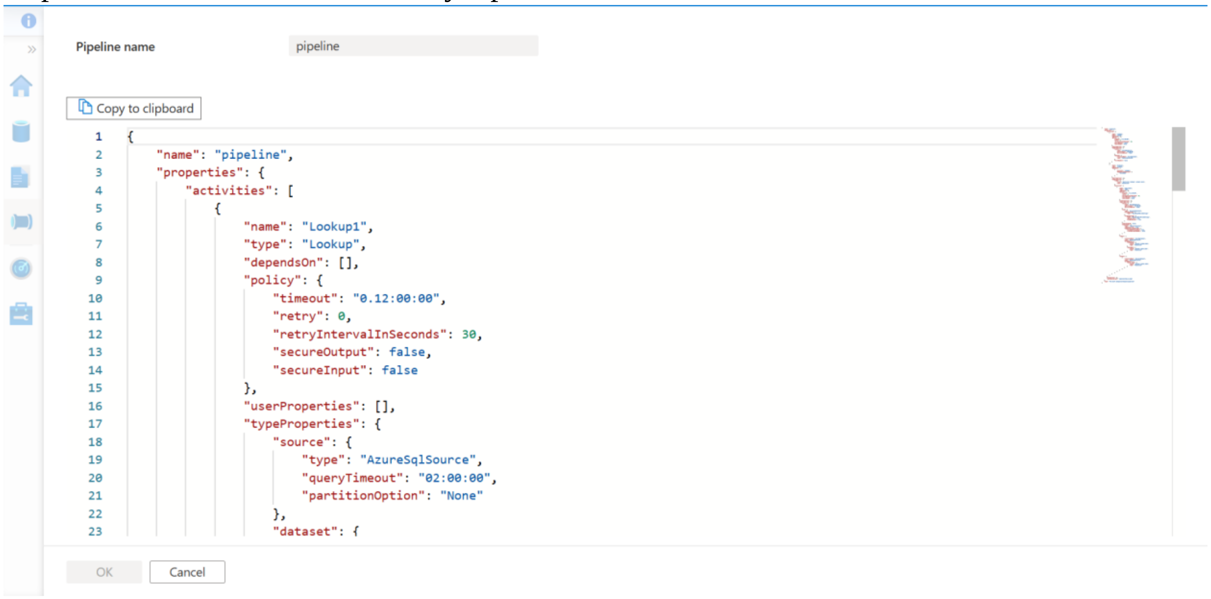
Description

Activity state: [✓ Activated] Deactivated

Properties — Name * pipeline

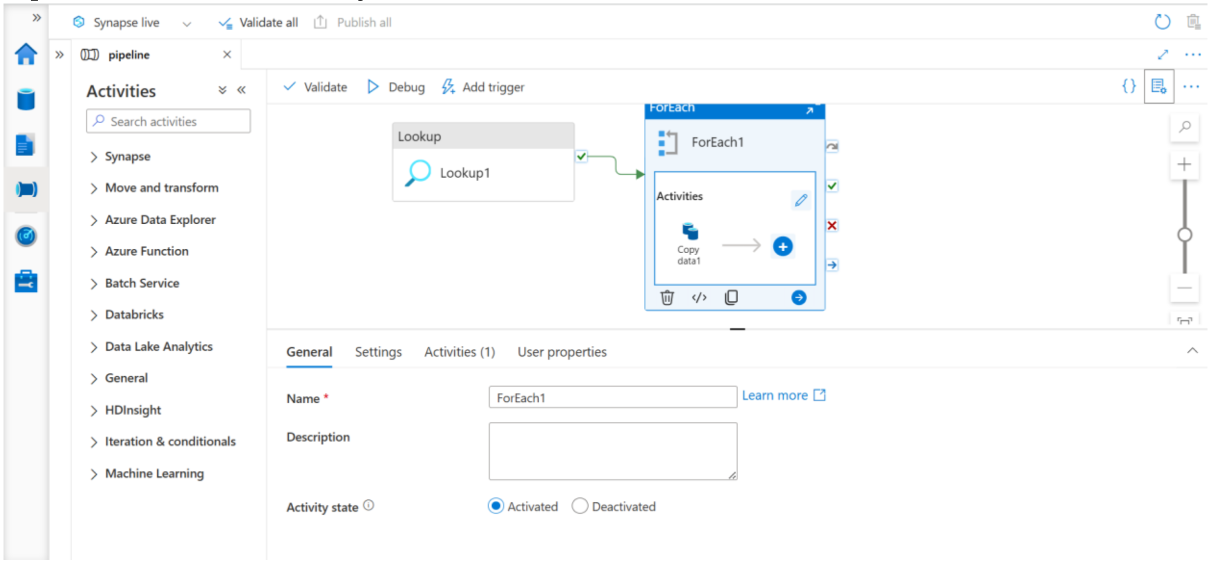
Description

Copied JSON code from ADF to synapse


Pipeline name: pipeline

```
{
    "name": "pipeline",
    "properties": {
        "activities": [
            {
                "name": "Lookup1",
                "type": "Lookup",
                "dependsOn": [],
                "policy": {
                    "timeout": "0.12:00:00",
                    "retry": 0,
                    "retryIntervalInSeconds": 30,
                    "secureOutput": false,
                    "secureInput": false
                },
                "userProperties": [],
                "typeProperties": {
                    "source": {
                        "type": "AzureSqlSource",
                        "queryTimeout": "02:00:00",
                        "partitionOption": "None"
                    },
                    "dataset": {
```

Pipeline created successfully.



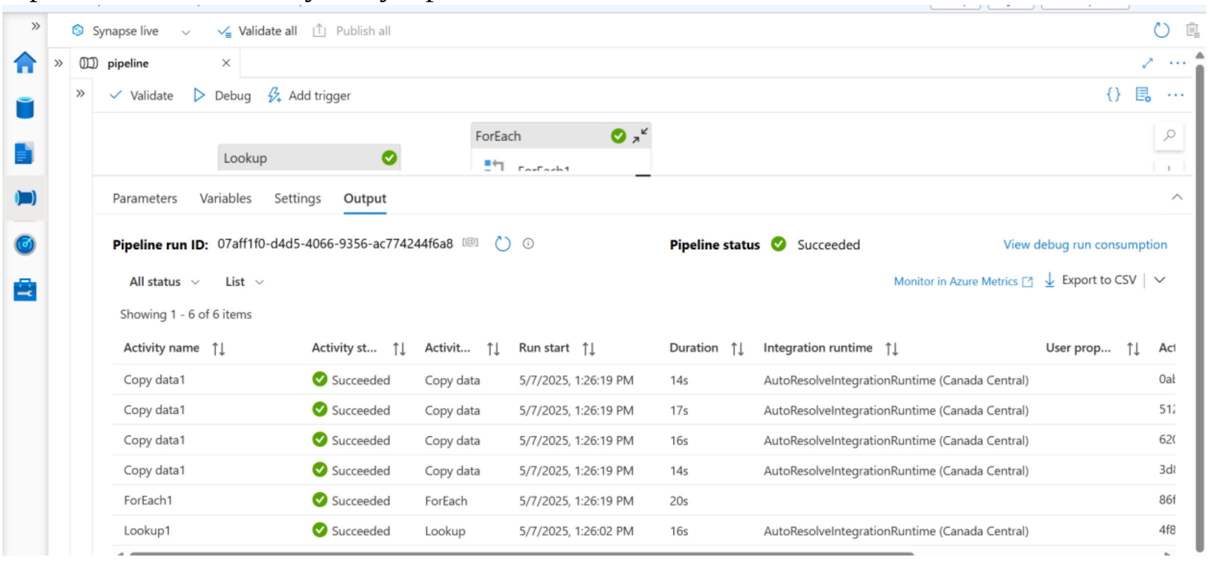
Name * ForEach1

Description

Activity state: [✓ Activated] Deactivated

Published pipeline and ran pipeline.

Pipeline ran successfully in Synapse.



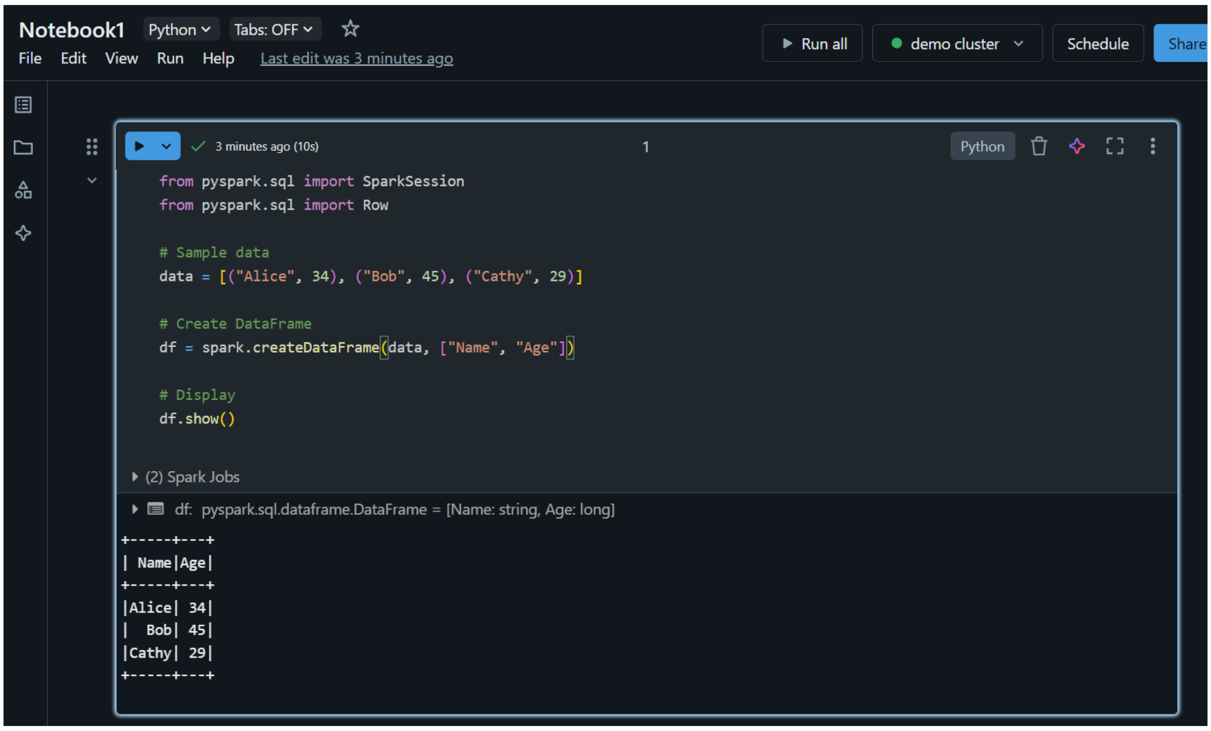
Pipeline run ID: 07aff1f0-d4d5-4066-9356-ac774244f6a8 **Pipeline status** Succeeded View debug run consumption

All status List Monitor in Azure Metrics Export to CSV

Showing 1 - 6 of 6 items

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime	User prop...	Act
Copy data1	Succeeded	Copy data	5/7/2025, 1:26:19 PM	14s	AutoResolveIntegrationRuntime (Canada Central)		0al
Copy data1	Succeeded	Copy data	5/7/2025, 1:26:19 PM	17s	AutoResolveIntegrationRuntime (Canada Central)		51
Copy data1	Succeeded	Copy data	5/7/2025, 1:26:19 PM	16s	AutoResolveIntegrationRuntime (Canada Central)		62
Copy data1	Succeeded	Copy data	5/7/2025, 1:26:19 PM	14s	AutoResolveIntegrationRuntime (Canada Central)		3d
ForEach1	Succeeded	ForEach	5/7/2025, 1:26:19 PM	20s			86
Lookup1	Succeeded	Lookup	5/7/2025, 1:26:02 PM	16s	AutoResolveIntegrationRuntime (Canada Central)		4f8

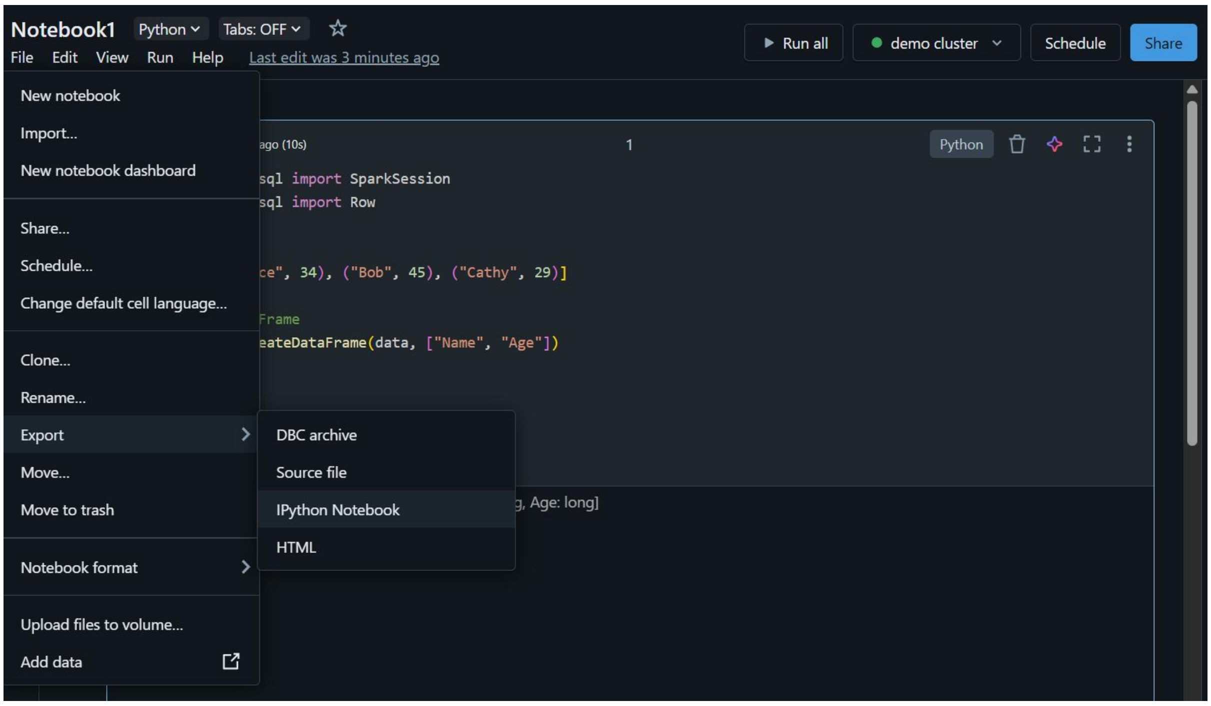
Task 2:
Migrating Databricks notebook to Fabric notebook.
Creating notebooks in Databricks
Notebook 1.



```
from pyspark.sql import SparkSession
from pyspark.sql import Row

# Sample data
data = [("Alice", 34), ("Bob", 45), ("Cathy", 29)]

# Create DataFrame
df = spark.createDataFrame(data, ["Name", "Age"])

# Display
df.show()
```

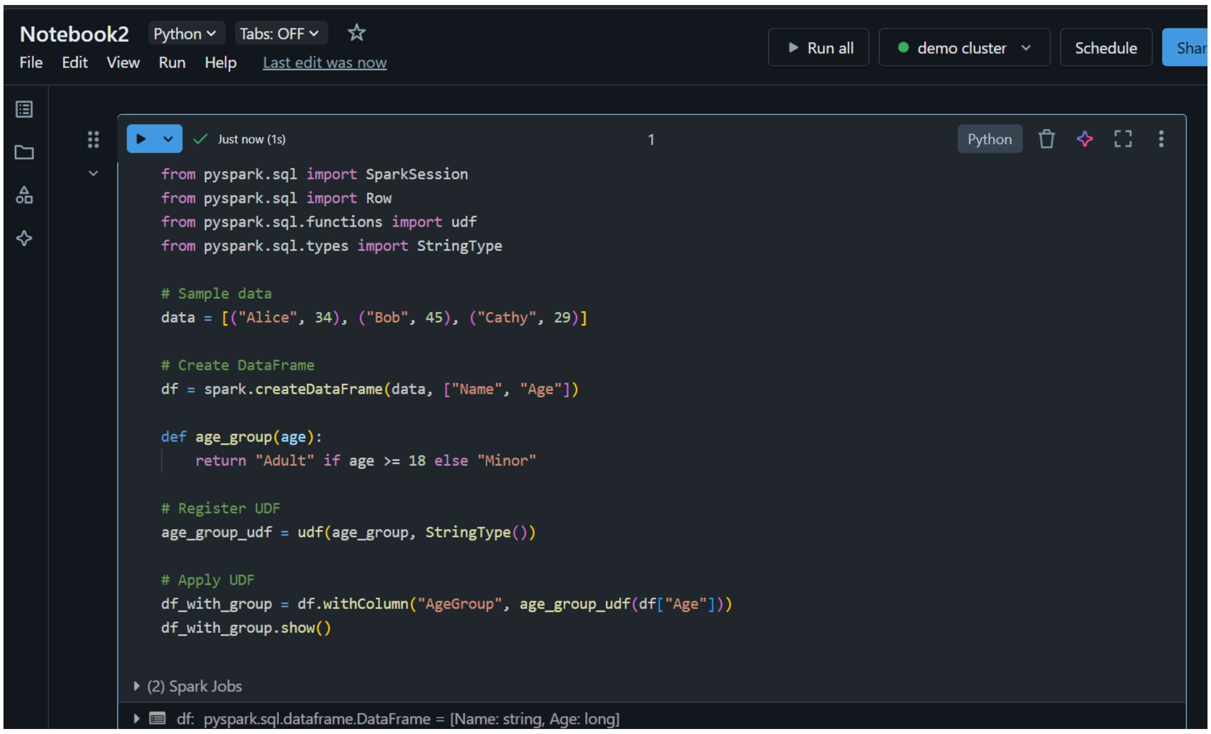
```
(2) Spark Jobs
df: pyspark.sql.dataframe.DataFrame = [Name: string, Age: long]
+-----+---+
| Name|Age|
+-----+---+
|Alice| 34|
|  Bob| 45|
|Cathy| 29|
+-----+---+
```

Importing this Notebook as ipynb file.



```
sql import SparkSession
sql import Row

ce", 34), ("Bob", 45), ("Cathy", 29)]

Frame
eateDataFrame(data, ["Name", "Age"])
```

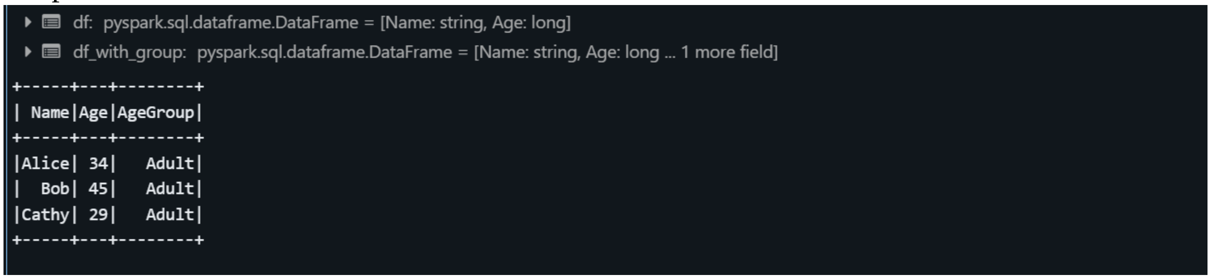
Creating Notebook 2



```
from pyspark.sql import SparkSession
from pyspark.sql import Row
from pyspark.sql.functions import udf
from pyspark.sql.types import StringType

# Sample data
data = [("Alice", 34), ("Bob", 45), ("Cathy", 29)]

# Create DataFrame
df = spark.createDataFrame(data, ["Name", "Age"])

def age_group(age):
    return "Adult" if age >= 18 else "Minor"

# Register UDF
age_group_udf = udf(age_group, StringType())

# Apply UDF
df_with_group = df.withColumn("AgeGroup", age_group_udf(df["Age"]))
df_with_group.show()
```

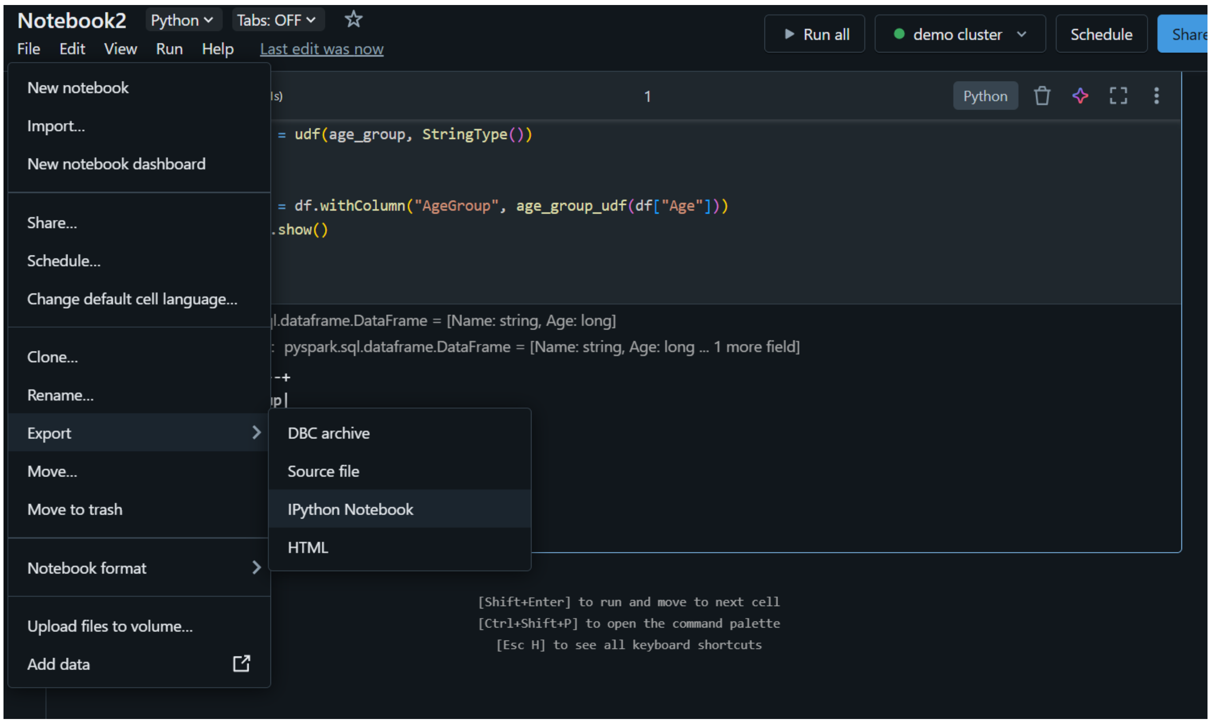
```
(2) Spark Jobs
df: pyspark.sql.dataframe.DataFrame = [Name: string, Age: long]
```

Output:



```
df: pyspark.sql.dataframe.DataFrame = [Name: string, Age: long]
df_with_group: pyspark.sql.dataframe.DataFrame = [Name: string, Age: long ... 1 more field]

+-----+---+--------+
| Name|Age|AgeGroup|
+-----+---+--------+
|Alice| 34|   Adult|
|  Bob| 45|   Adult|
|Cathy| 29|   Adult|
+-----+---+--------+
```

Exporting Notebook2.

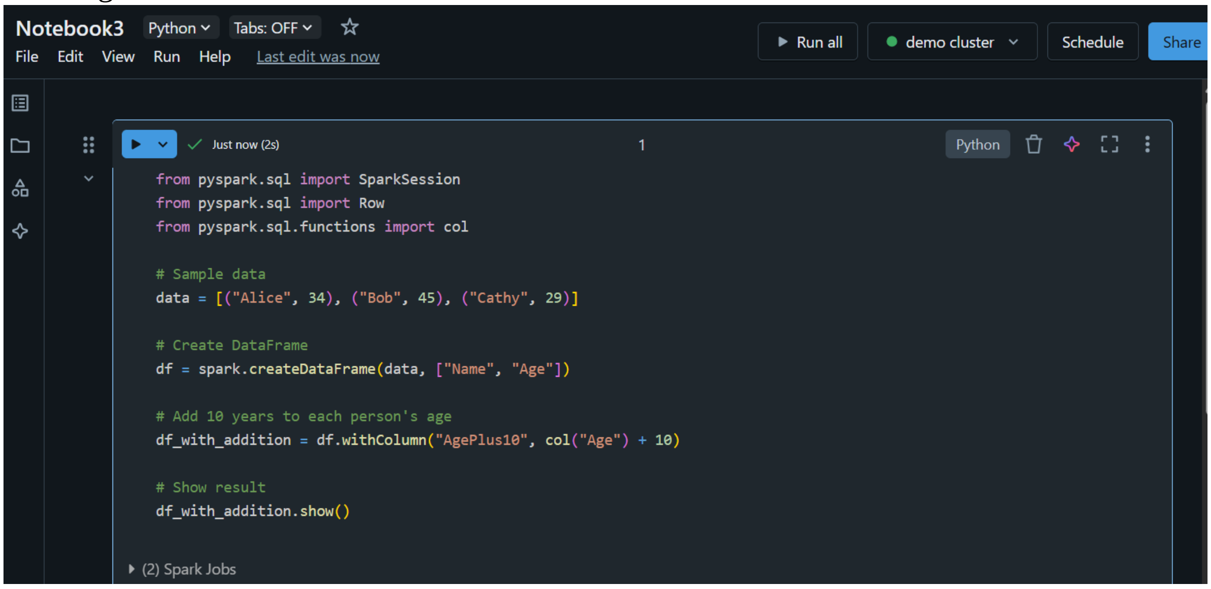


```
= udf(age_group, StringType())

= df.withColumn("AgeGroup", age_group_udf(df["Age"]))
.show()
```

```
l.dataframe.DataFrame = [Name: string, Age: long]
: pyspark.sql.dataframe.DataFrame = [Name: string, Age: long ... 1 more field]
-+
p|
```

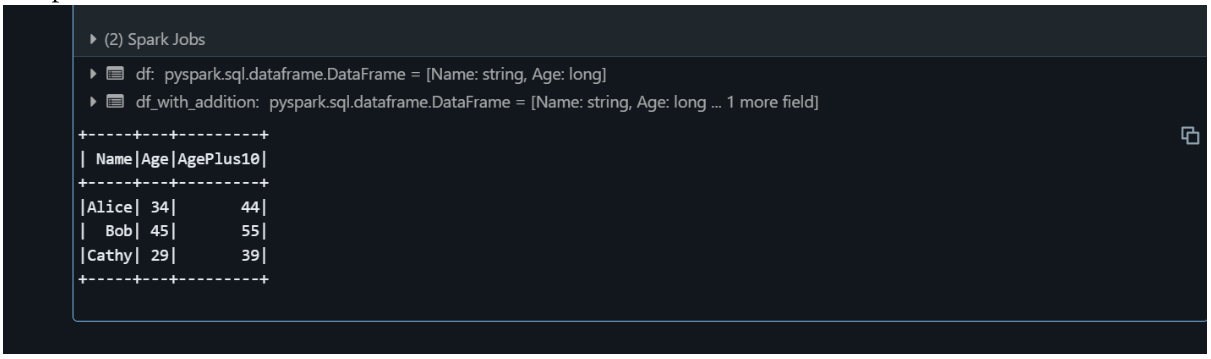
```
[Shift+Enter] to run and move to next cell
[Ctrl+Shift+P] to open the command palette
[Esc H] to see all keyboard shortcuts
```

Creating Notebook3.



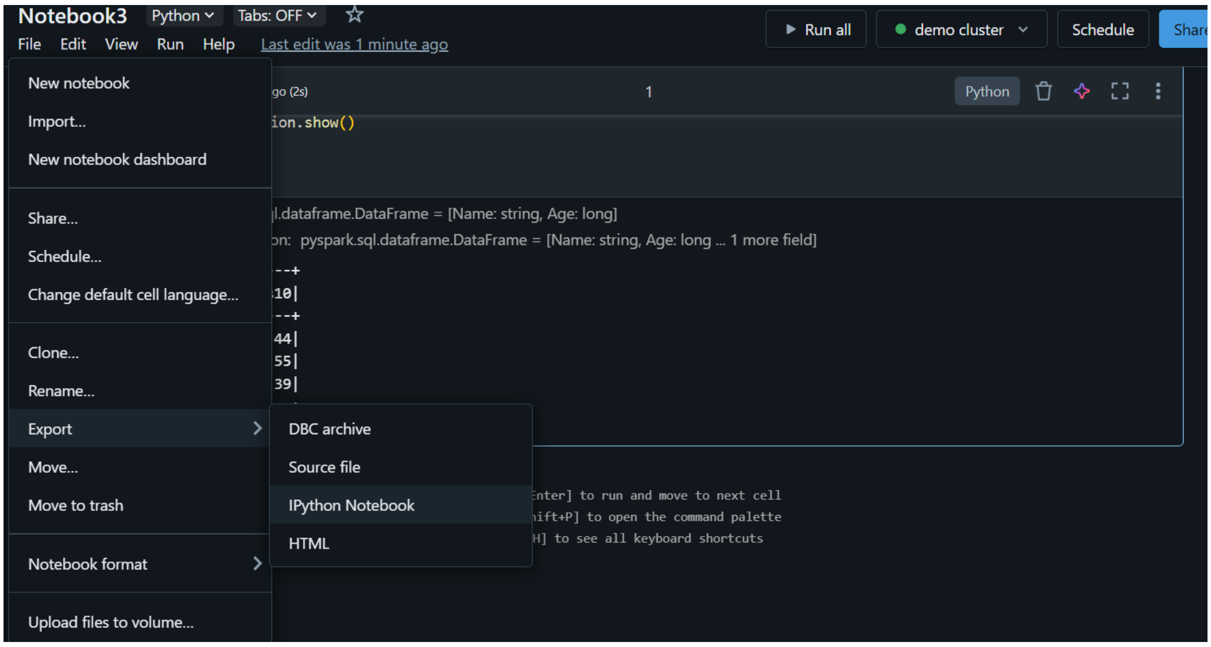
```python
from pyspark.sql import SparkSession
from pyspark.sql import Row
from pyspark.sql.functions import col

# Sample data
data = [("Alice", 34), ("Bob", 45), ("Cathy", 29)]

# Create DataFrame
df = spark.createDataFrame(data, ["Name", "Age"])

# Add 10 years to each person's age
df_with_addition = df.withColumn("AgePlus10", col("Age") + 10)

# Show result
df_with_addition.show()
```

Output:



```
(2) Spark Jobs
df: pyspark.sql.dataframe.DataFrame = [Name: string, Age: long]
df_with_addition: pyspark.sql.dataframe.DataFrame = [Name: string, Age: long ... 1 more field]
+-----+---+---------+
| Name|Age|AgePlus10|
+-----+---+---------+
|Alice| 34|       44|
|  Bob| 45|       55|
|Cathy| 29|       39|
+-----+---+---------+
```

Exporting this Notebook3.

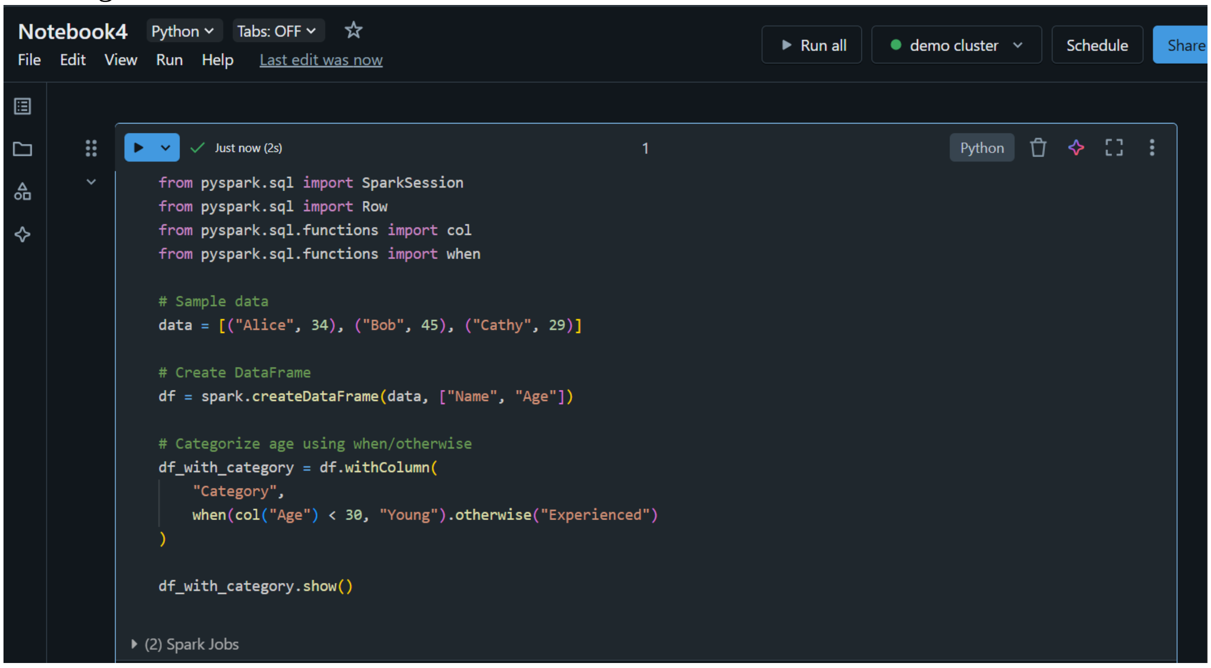


```
ion.show()
```

```
l.dataframe.DataFrame = [Name: string, Age: long]
on:  pyspark.sql.dataframe.DataFrame = [Name: string, Age: long ... 1 more field]
--+
10|
--+
44|
55|
39|
```

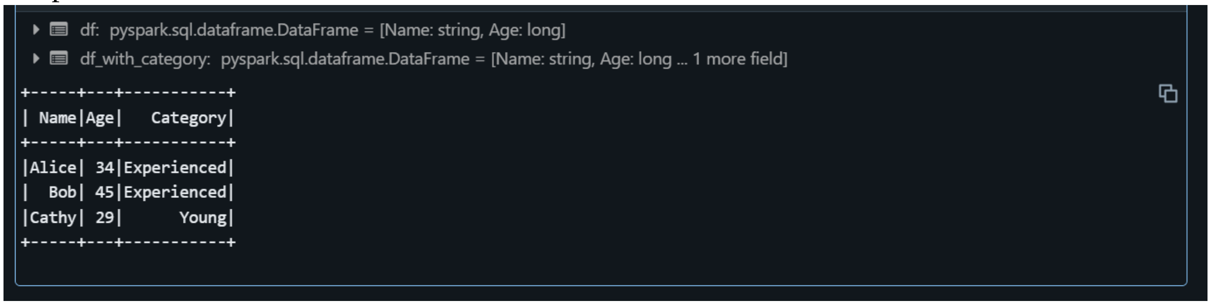
```
Enter] to run and move to next cell
ift+P] to open the command palette
H] to see all keyboard shortcuts
```

Creating notebook 4



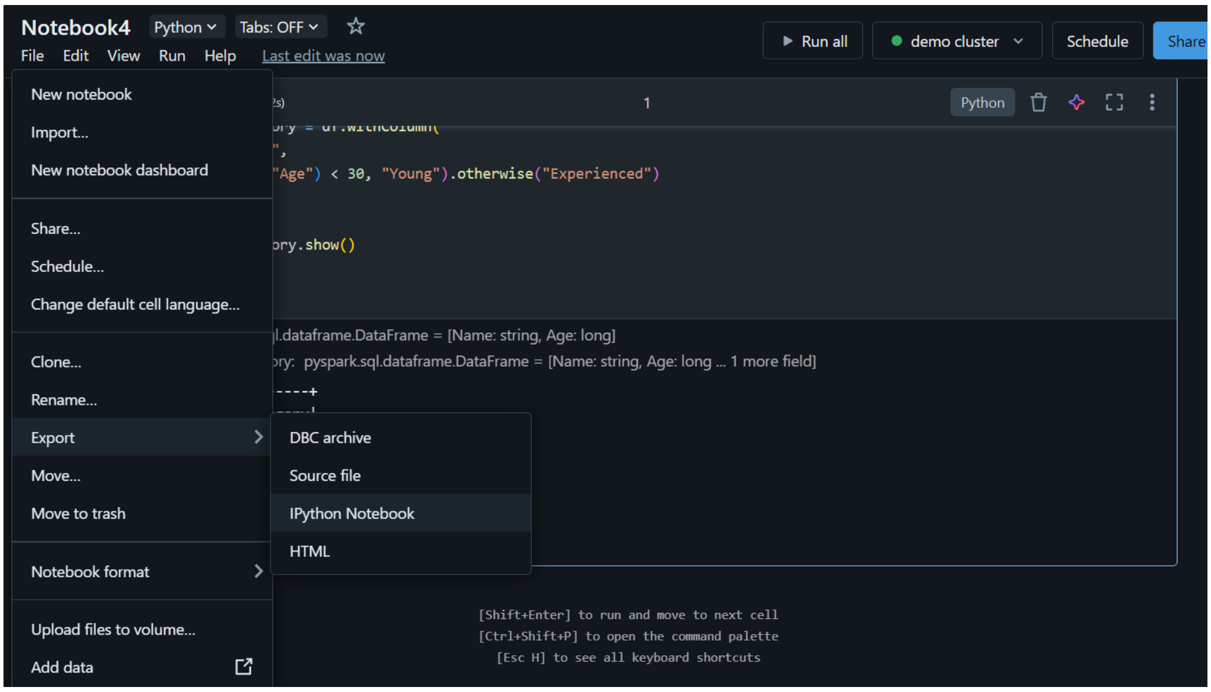
```python
from pyspark.sql import SparkSession
from pyspark.sql import Row
from pyspark.sql.functions import col
from pyspark.sql.functions import when

# Sample data
data = [("Alice", 34), ("Bob", 45), ("Cathy", 29)]

# Create DataFrame
df = spark.createDataFrame(data, ["Name", "Age"])

# Categorize age using when/otherwise
df_with_category = df.withColumn(
    "Category",
    when(col("Age") < 30, "Young").otherwise("Experienced")
)

df_with_category.show()
```

Output:



```
df:  pyspark.sql.dataframe.DataFrame = [Name: string, Age: long]
df_with_category:  pyspark.sql.dataframe.DataFrame = [Name: string, Age: long ... 1 more field]
+-----+---+-----------+
| Name|Age|   Category|
+-----+---+-----------+
|Alice| 34|Experienced|
|  Bob| 45|Experienced|
|Cathy| 29|      Young|
+-----+---+-----------+
```

Exporting Notebook4

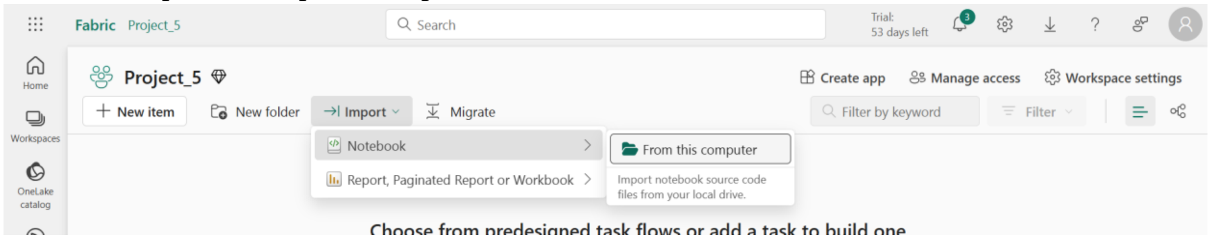


```
ory = df.withColumn(
",
"Age") < 30, "Young").otherwise("Experienced")

ory.show()
```

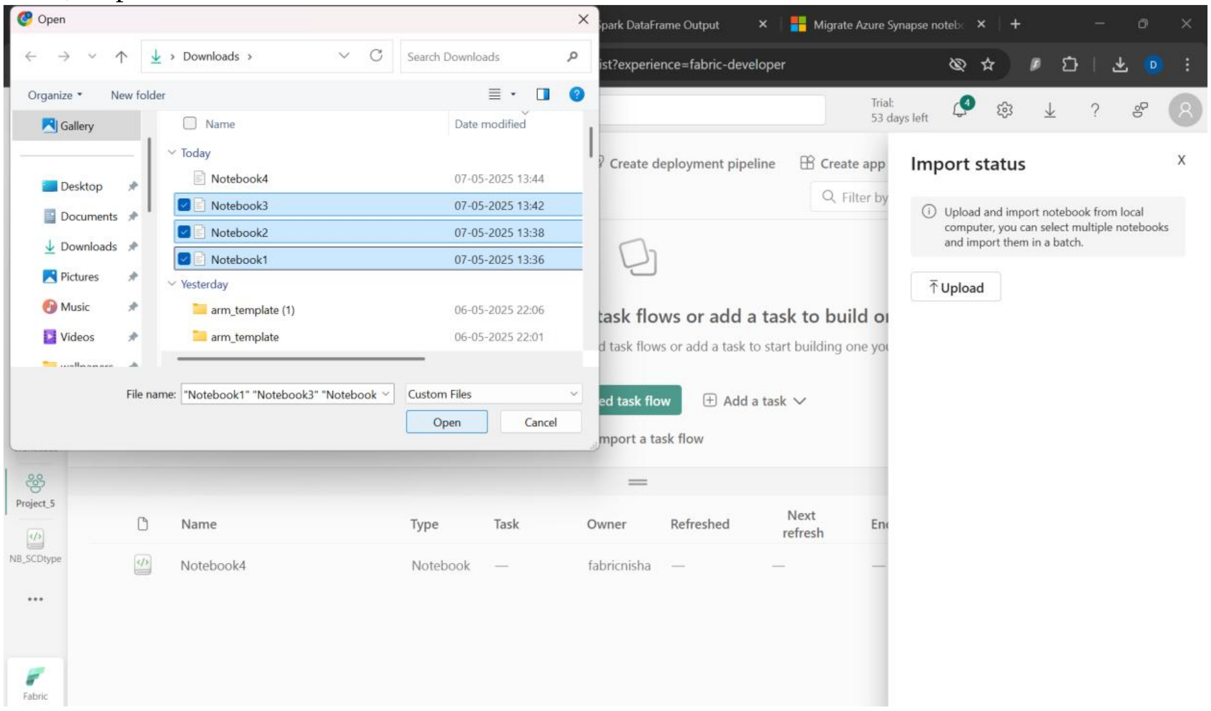
```
l.dataframe.DataFrame = [Name: string, Age: long]
ory:  pyspark.sql.dataframe.DataFrame = [Name: string, Age: long ... 1 more field]
----+
```

```
[Shift+Enter] to run and move to next cell
[Ctrl+Shift+P] to open the command palette
[Esc H] to see all keyboard shortcuts
```

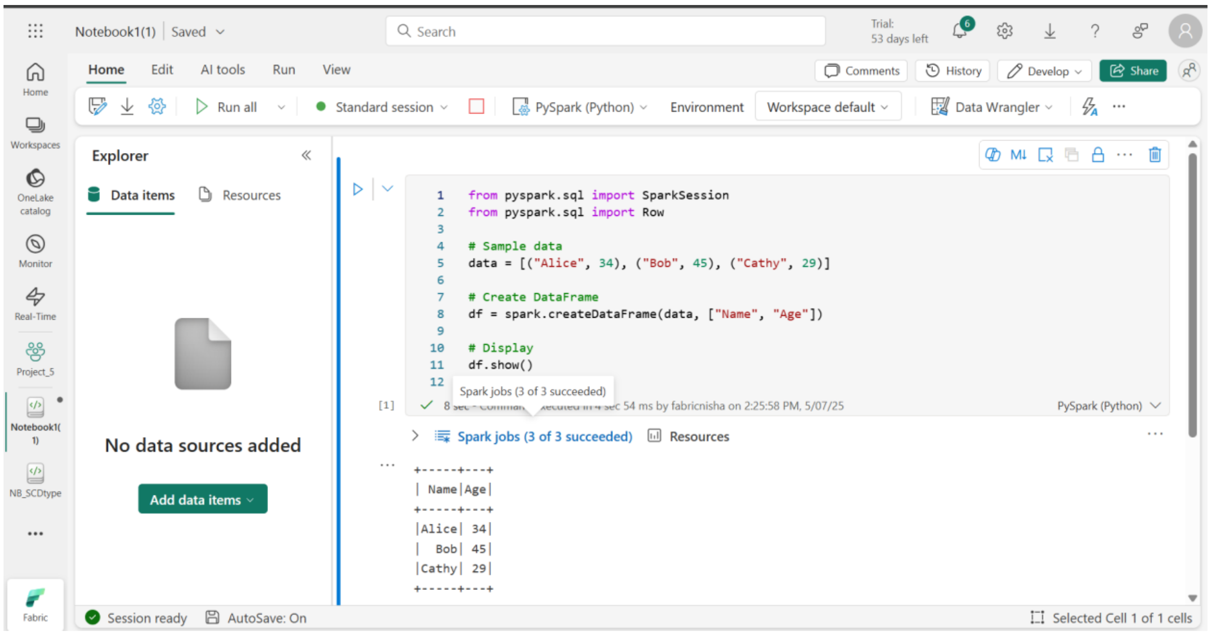
Now will import exported notebooks from Databricks to Fabric.
Go to workspace -> import-> import notebook.



Now, import all Notebooks at once


Open
Downloads
Search Downloads
Organize
New folder
Gallery
Desktop
Documents
Downloads
Pictures
Music
Videos
Name
Date modified
Today
Notebook4 07-05-2025 13:44
Notebook3 07-05-2025 13:42
Notebook2 07-05-2025 13:38
Notebook1 07-05-2025 13:36
Yesterday
arm_template (1) 06-05-2025 22:06
arm_template 06-05-2025 22:01
File name: "Notebook1" "Notebook3" "Notebook
Custom Files
Open
Cancel
park DataFrame Output
Migrate Azure Synapse noteb
ist?experience=fabric-developer
Trial: 53 days left
Create deployment pipeline
Create app
Filter by
Import status
Upload and import notebook from local computer, you can select multiple notebooks and import them in a batch.
Upload
task flows or add a task to build o
d task flows or add a task to start building one yo
ed task flow
Add a task
mport a task flow
Project_5
NB_SCDtype
Fabric
Name Type Task Owner Refreshed Next refresh En
Notebook4 Notebook — fabricnisha — —

Now open Notebooks and run them.
Notebook 1 ran successfully.



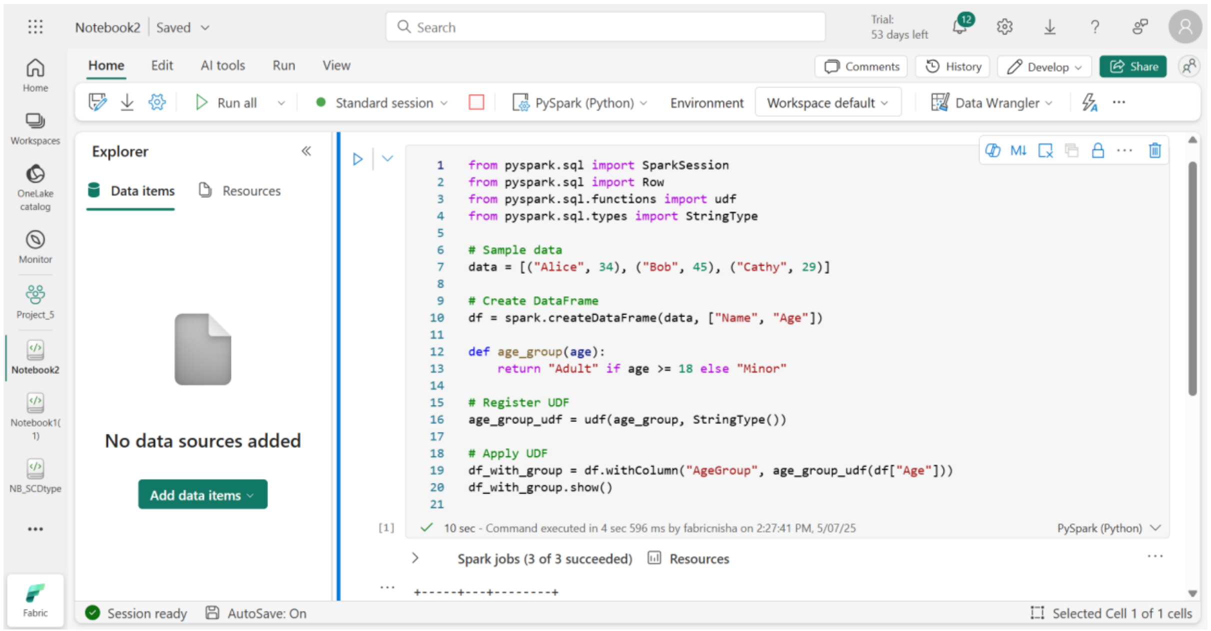
```
from pyspark.sql import SparkSession
from pyspark.sql import Row

# Sample data
data = [("Alice", 34), ("Bob", 45), ("Cathy", 29)]

# Create DataFrame
df = spark.createDataFrame(data, ["Name", "Age"])

# Display
df.show()
```

[1] ✓ 8 sec - Command executed in 4 sec 54 ms by fabricnisha on 2:25:58 PM, 5/07/25 PySpark (Python)

Spark jobs (3 of 3 succeeded) Resources

```
+-----+---+
| Name|Age|
+-----+---+
|Alice| 34|
|  Bob| 45|
|Cathy| 29|
+-----+---+
```

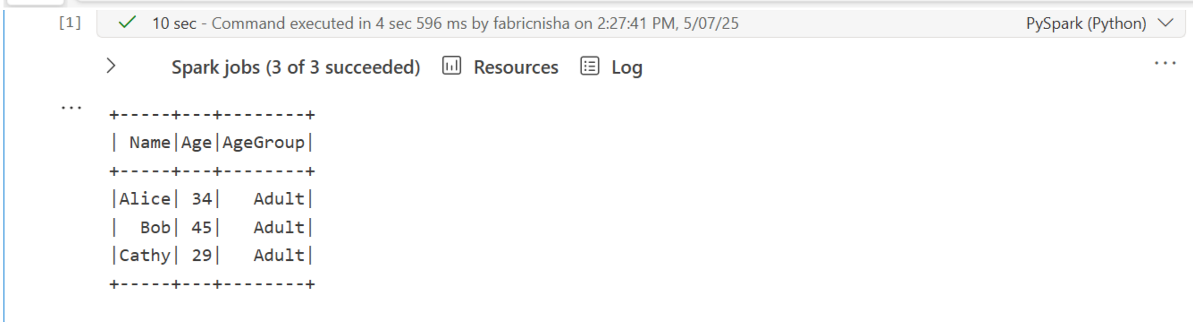
Run Notebook2



```
from pyspark.sql import SparkSession
from pyspark.sql import Row
from pyspark.sql.functions import udf
from pyspark.sql.types import StringType

# Sample data
data = [("Alice", 34), ("Bob", 45), ("Cathy", 29)]

# Create DataFrame
df = spark.createDataFrame(data, ["Name", "Age"])

def age_group(age):
    return "Adult" if age >= 18 else "Minor"

# Register UDF
age_group_udf = udf(age_group, StringType())

# Apply UDF
df_with_group = df.withColumn("AgeGroup", age_group_udf(df["Age"]))
df_with_group.show()
```

[1] ✓ 10 sec - Command executed in 4 sec 596 ms by fabricnisha on 2:27:41 PM, 5/07/25 PySpark (Python)

Spark jobs (3 of 3 succeeded) Resources



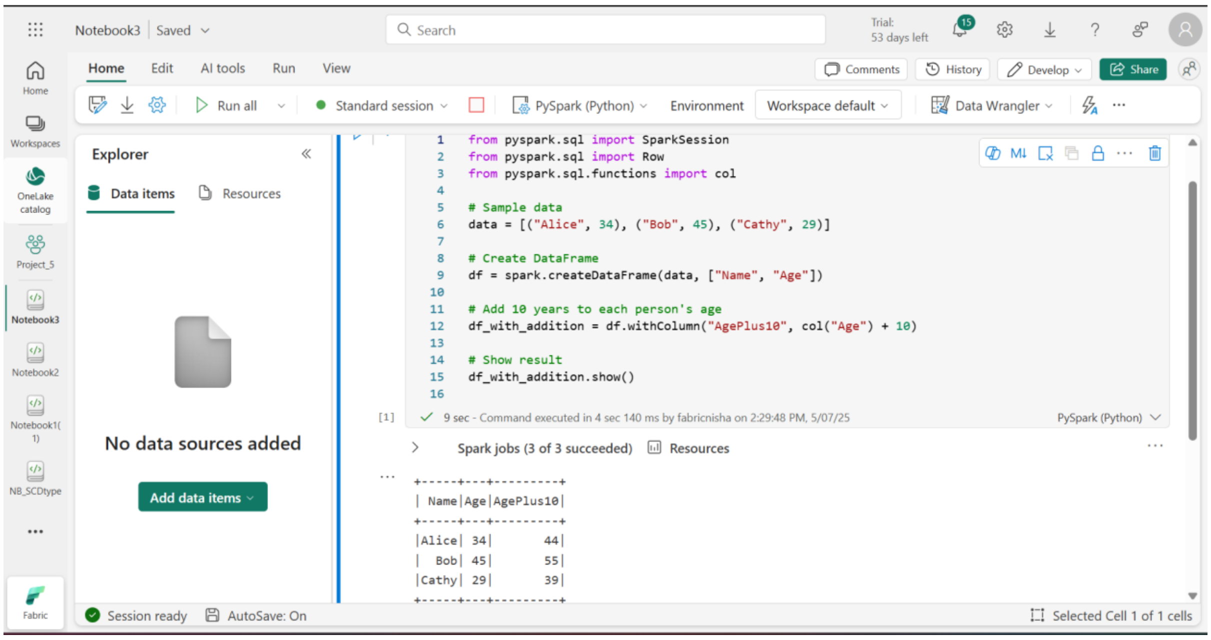
[1] ✓ 10 sec - Command executed in 4 sec 596 ms by fabricnisha on 2:27:41 PM, 5/07/25 PySpark (Python)

Spark jobs (3 of 3 succeeded) Resources Log

```
+-----+---+--------+
| Name|Age|AgeGroup|
+-----+---+--------+
|Alice| 34|   Adult|
|  Bob| 45|   Adult|
|Cathy| 29|   Adult|
+-----+---+--------+
```

Notebook 2 ran successfully.

Run Notebook 3



```python
from pyspark.sql import SparkSession
from pyspark.sql import Row
from pyspark.sql.functions import col

# Sample data
data = [("Alice", 34), ("Bob", 45), ("Cathy", 29)]

# Create DataFrame
df = spark.createDataFrame(data, ["Name", "Age"])

# Add 10 years to each person's age
df_with_addition = df.withColumn("AgePlus10", col("Age") + 10)

# Show result
df_with_addition.show()
```

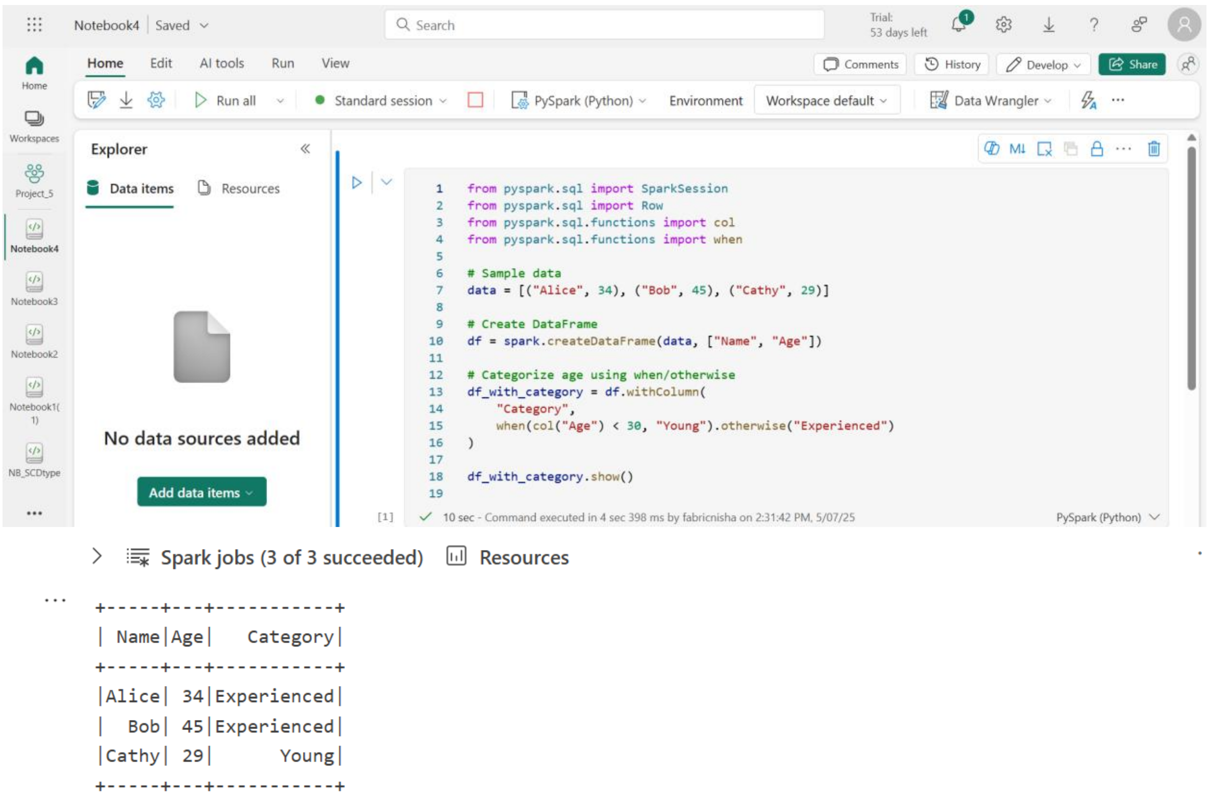
[1] ✓ 9 sec - Command executed in 4 sec 140 ms by fabricnisha on 2:29:48 PM, 5/07/25 PySpark (Python)

Spark jobs (3 of 3 succeeded) Resources

```
+-----+---+---------+
| Name|Age|AgePlus10|
+-----+---+---------+
|Alice| 34|       44|
|  Bob| 45|       55|
|Cathy| 29|       39|
+-----+---+---------+
```

Notebook3 ran successfully.

Run Notebook4



```python
from pyspark.sql import SparkSession
from pyspark.sql import Row
from pyspark.sql.functions import col
from pyspark.sql.functions import when

# Sample data
data = [("Alice", 34), ("Bob", 45), ("Cathy", 29)]

# Create DataFrame
df = spark.createDataFrame(data, ["Name", "Age"])

# Categorize age using when/otherwise
df_with_category = df.withColumn(
    "Category",
    when(col("Age") < 30, "Young").otherwise("Experienced")
)

df_with_category.show()
```

[1] ✓ 10 sec - Command executed in 4 sec 398 ms by fabricnisha on 2:31:42 PM, 5/07/25 PySpark (Python)

Spark jobs (3 of 3 succeeded) Resources

```
+-----+---+-----------+
| Name|Age|   Category|
+-----+---+-----------+
|Alice| 34|Experienced|
|  Bob| 45|Experienced|
|Cathy| 29|      Young|
+-----+---+-----------+
```

Notebook 4 ran successfully.