Azure Data Engineering Project Report

Technologies Used

- **Azure Data Factory**: For data orchestration and pipeline management
- **Azure Databricks**: For advanced data processing and analytics
- **Azure Synapse Analytics**: For data warehousing and serverless SQL
- **Azure Storage Account**: For data lake storage
- **Power BI**: For data visualization and reporting

Step 01

Create resource group to keep all the necessary resources

Step 02

Create storage account. Choose Data Lake Gen2 to create the file system in hierarchical structure.

Blob Storage for:

storing files, images, logs, backups, etc.

Don’t need analytics or a hierarchical file structure.

Data Lake Gen2 for:

building a data lake for analytics or ML.

need Hadoop/Spark/Hive integration.

have a file system-like structure with directories.

Step 3

Create Azure Data Factory. Create 3 containers for bronze, silver, gold in Storage Account

Implementing Data Pipeline in Azure Data Factory

1.Static Connection Implementation

The project begins with establishing static connections to data sources. Static connections represent the traditional approach where data files are loaded individually using manual copy operations.

Key Characteristics of Static Approach:

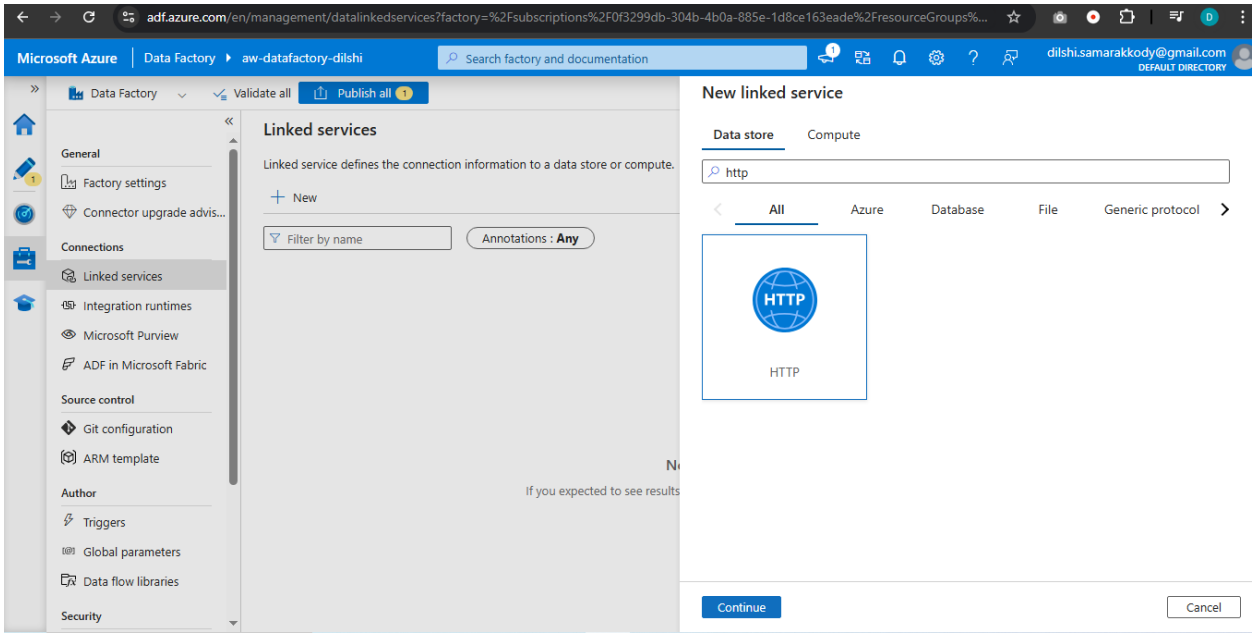
- Manual intervention required for each data file
- Individual file processing
- Limited scalability
- Suitable for small, infrequent data loads

Copy Data Activity Configuration

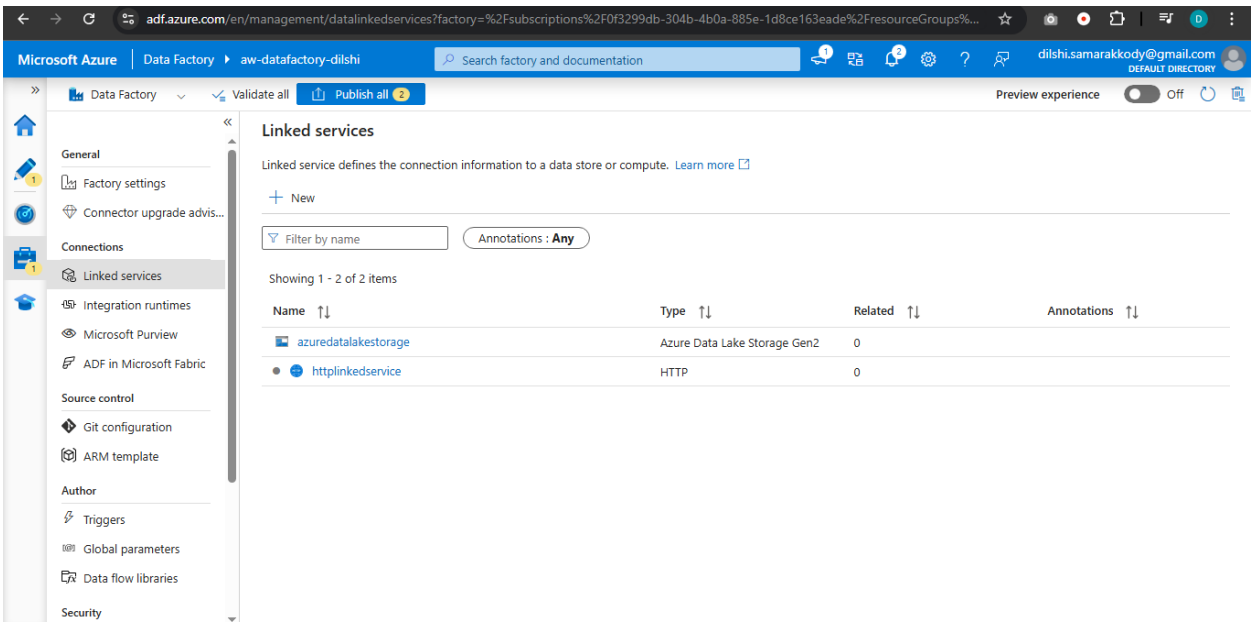
The static implementation uses Azure Data Factory's Copy Data activity with the following configurations:

- Source dataset definition
- Destination dataset configuration

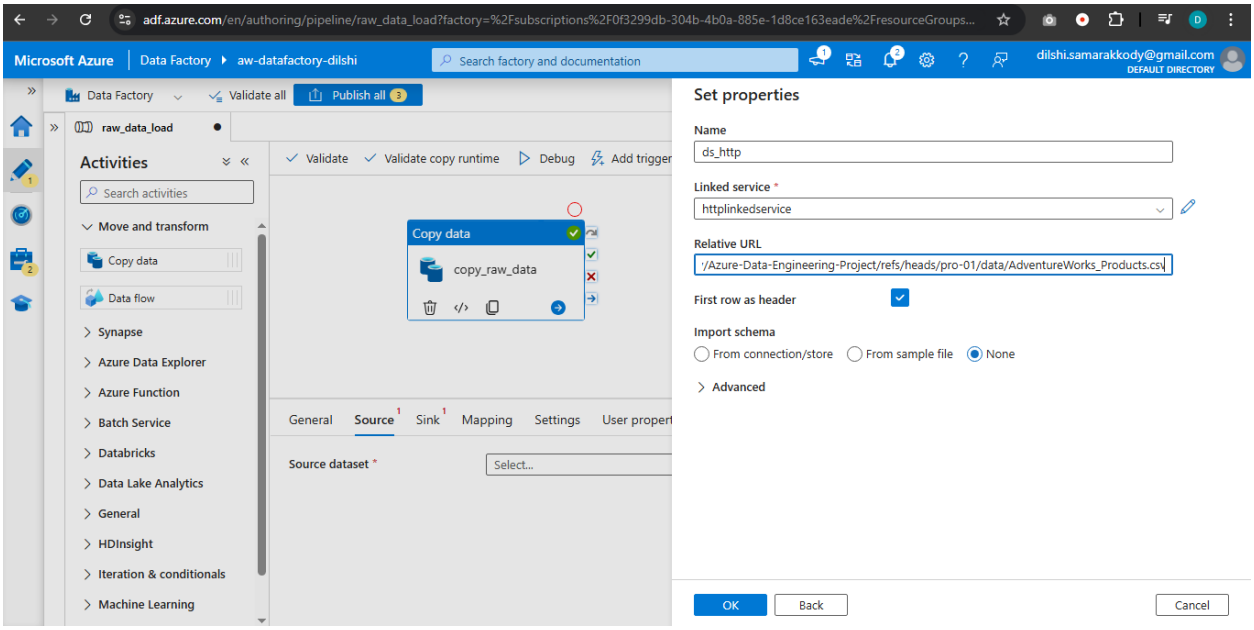
To create a connection with source (Github) create http linked service

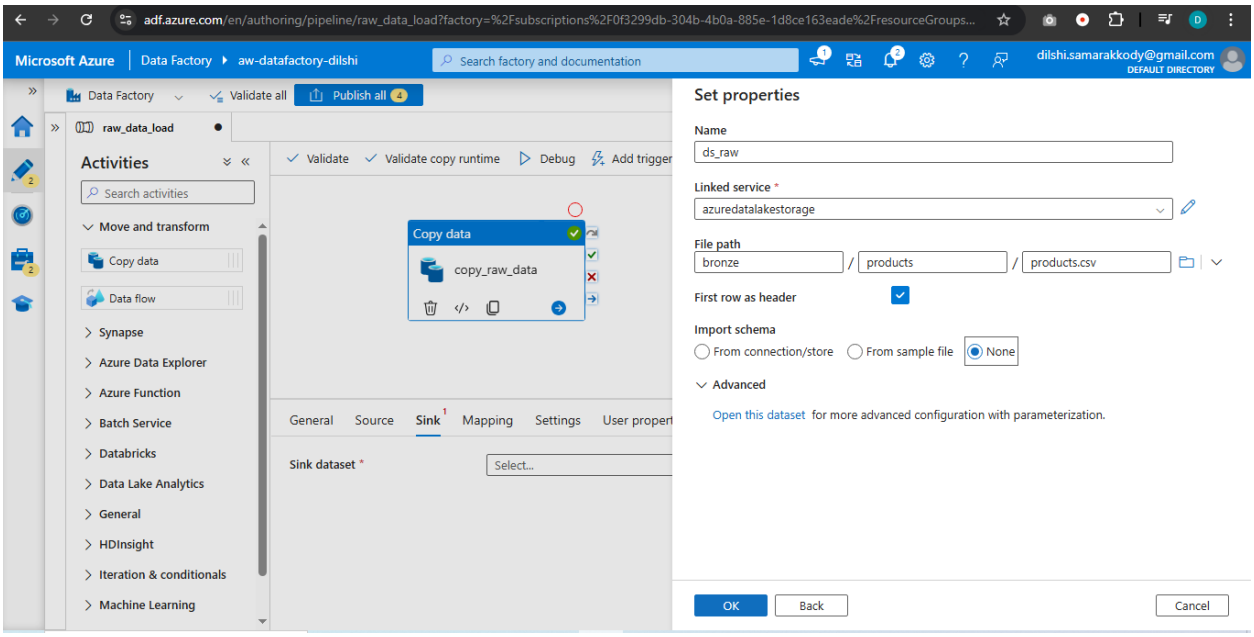


Create Destination linked service for Azure Data Lake Storage Gen2

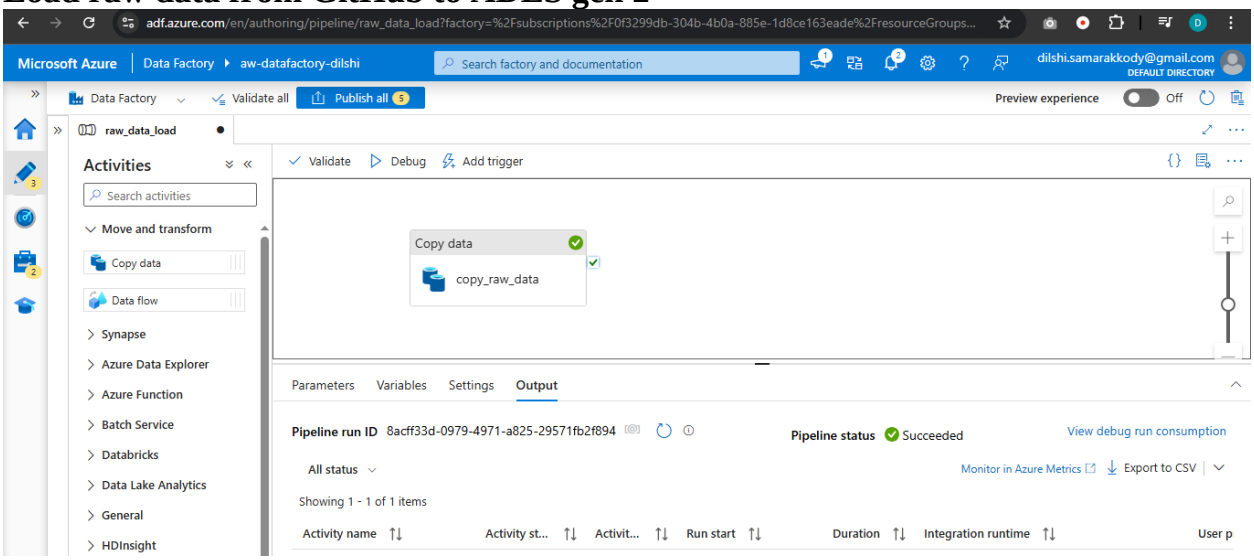


Set properties for source and sink





Load raw data from GitHub to ADLS gen 2



2. Dynamic Pipeline Architecture

The dynamic pipelines represent a significant improvement in automation and scalability. Dynamic pipelines can automatically process multiple files and adapt to changing data requirements.

Key Components:

Lookup Activity: Retrieves metadata and configuration information

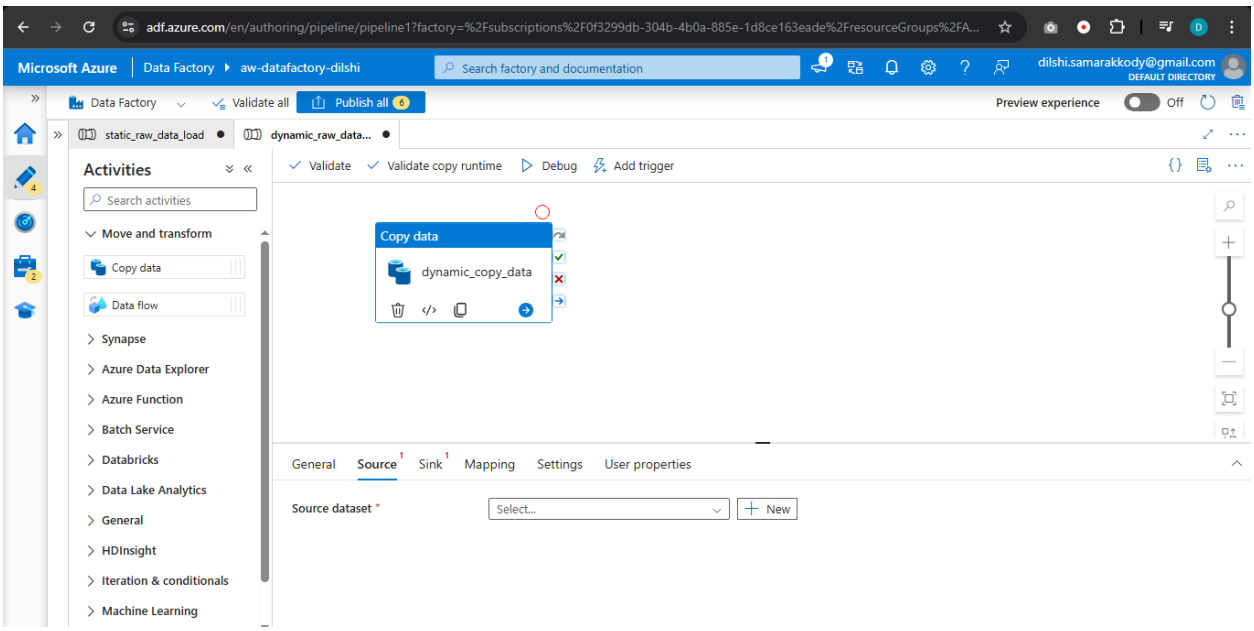
ForEach Activity: Iterates through data sources dynamically

Copy Data Activity: Performs data transfer operations

Data Flow: Handles complex transformations

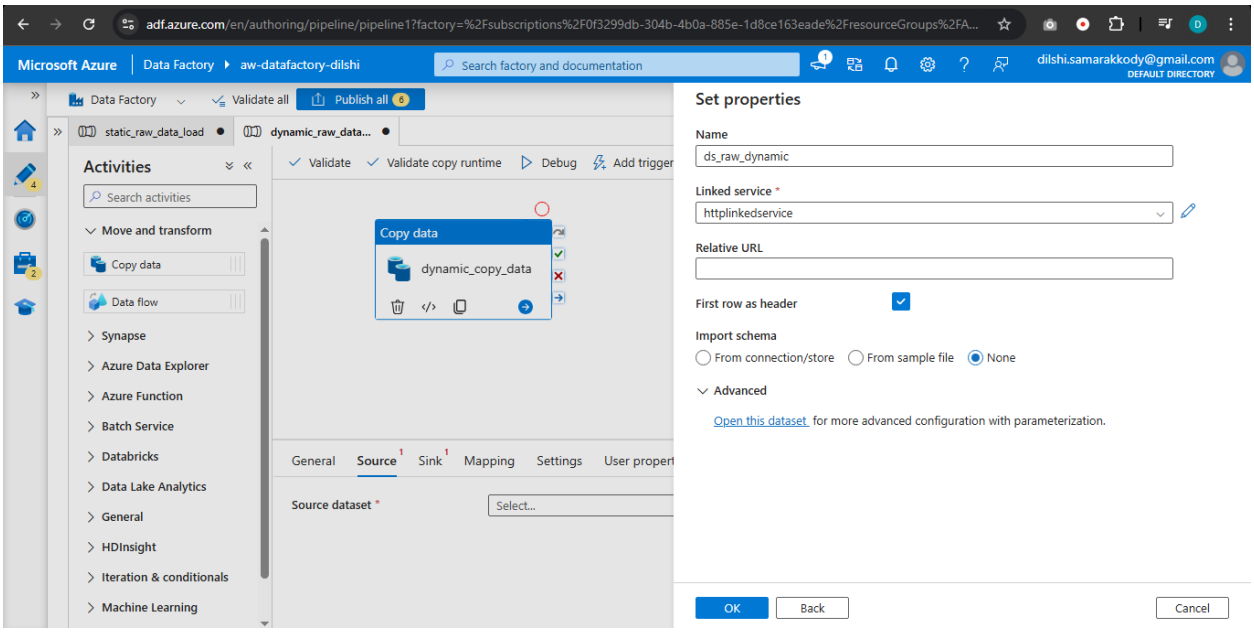
Step 5

Create parameterized copy data tool



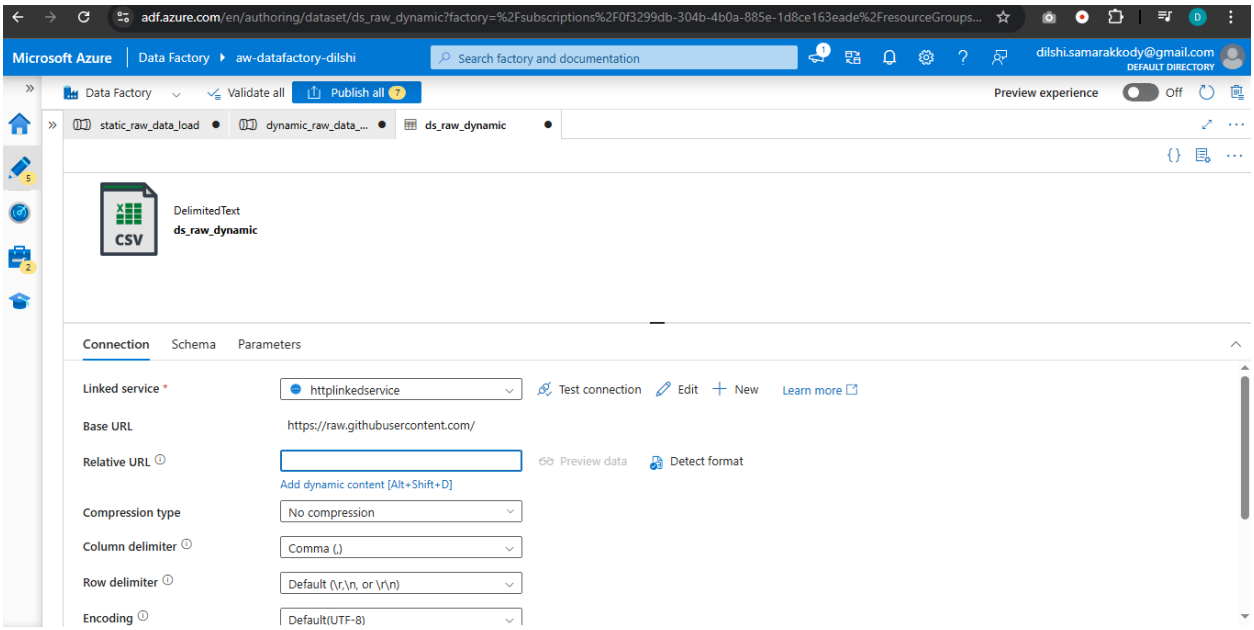
Step 06

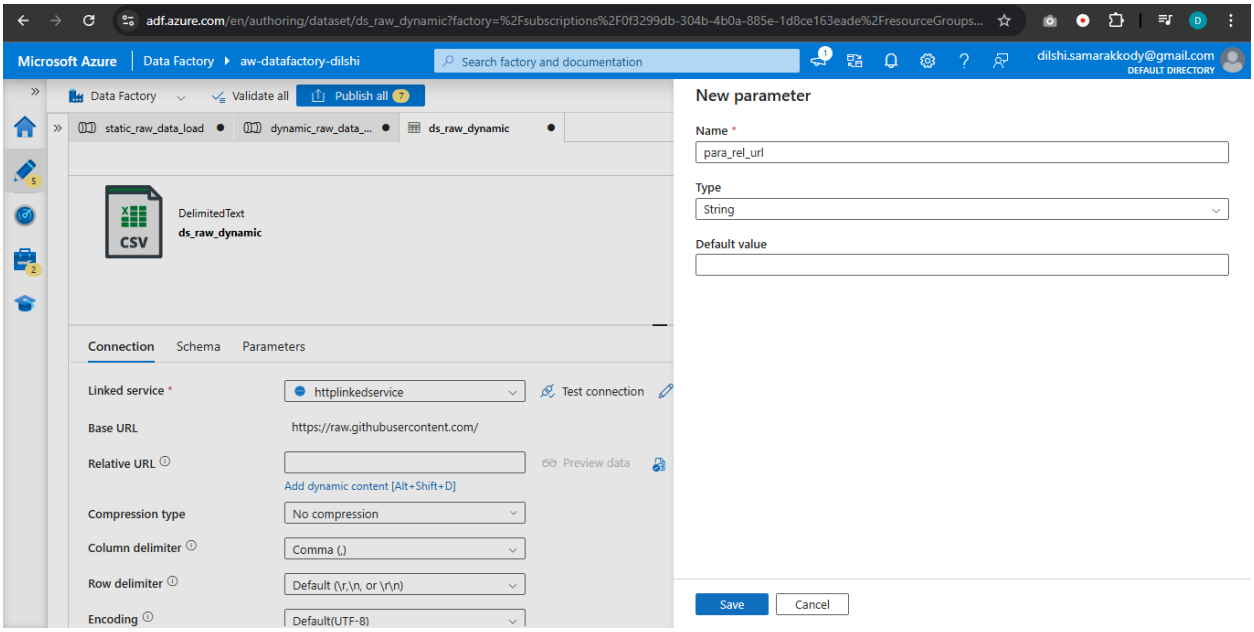
And set properties

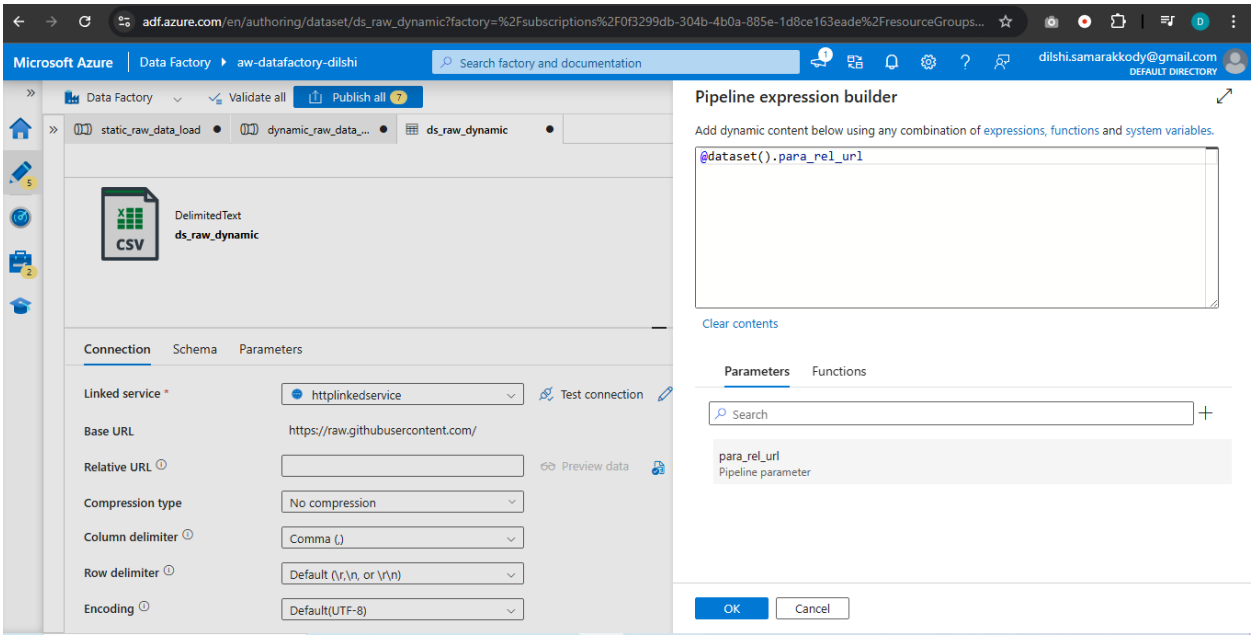


Step 7

Create dynamic parameter for relative URL

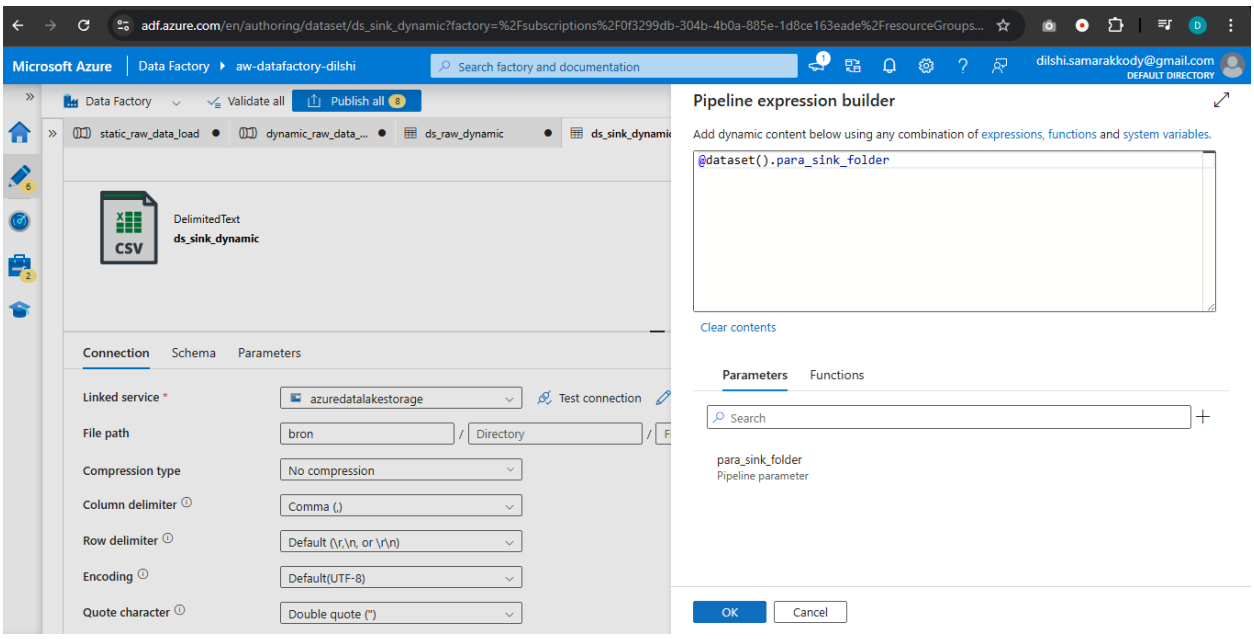






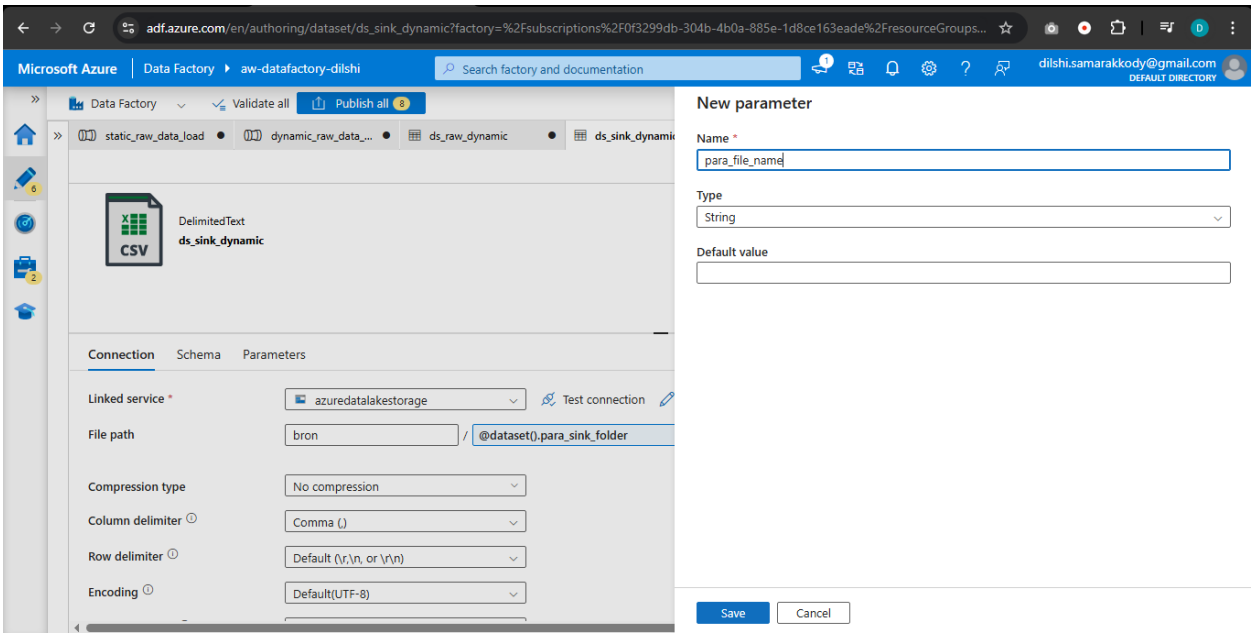
Step 8

Create dynamic parameter for sink folder



Step 9

Create dynamic parameter for sink filename



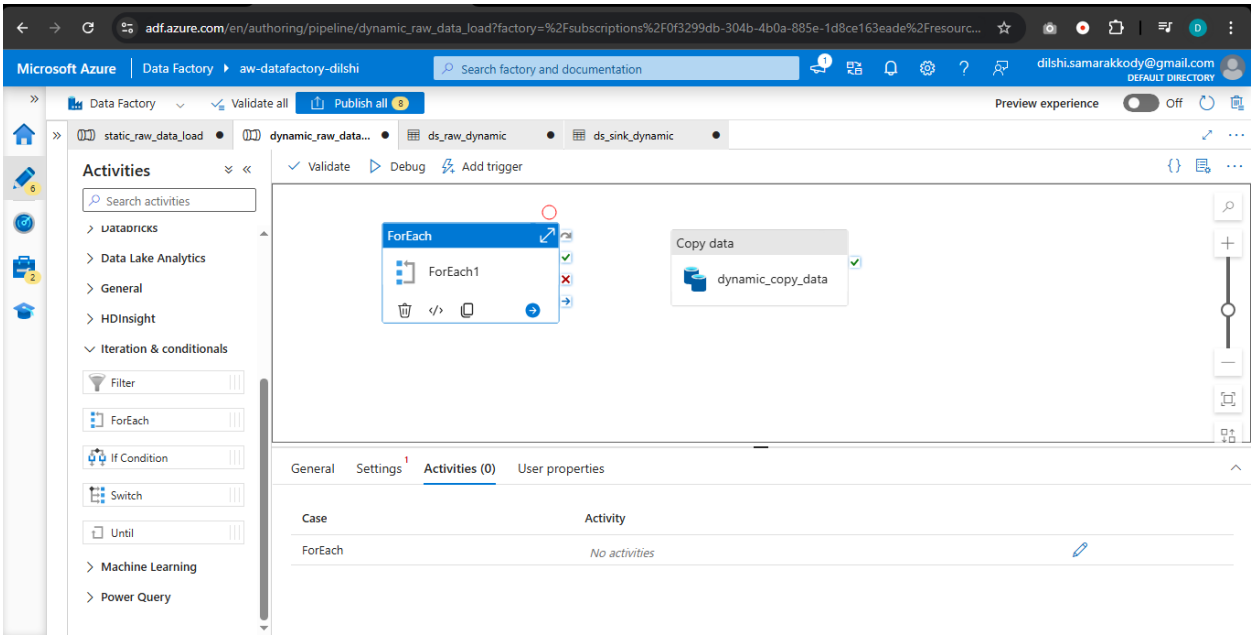
ForEach Loop Implementation and provide values for the dynamic parameters

The ForEach activity enables parallel processing of multiple data sources:

- Iterates through lookup results
- Executes child activities for each item
- Supports sequential and parallel execution modes
- Provides item-level error handling

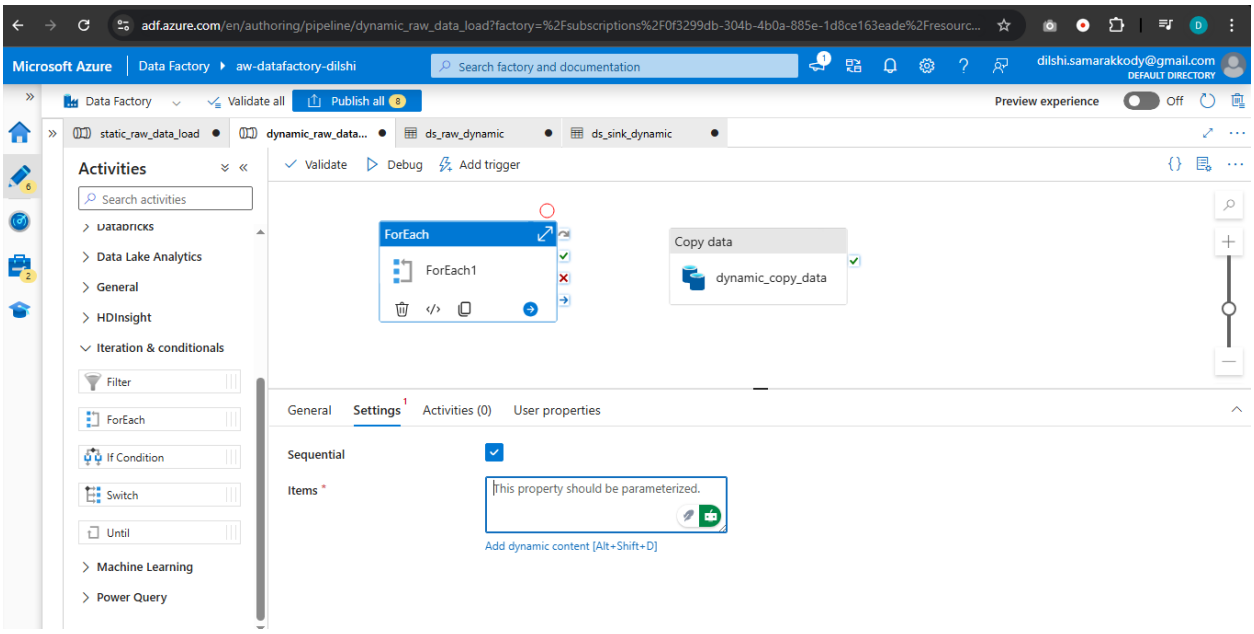
Step 10

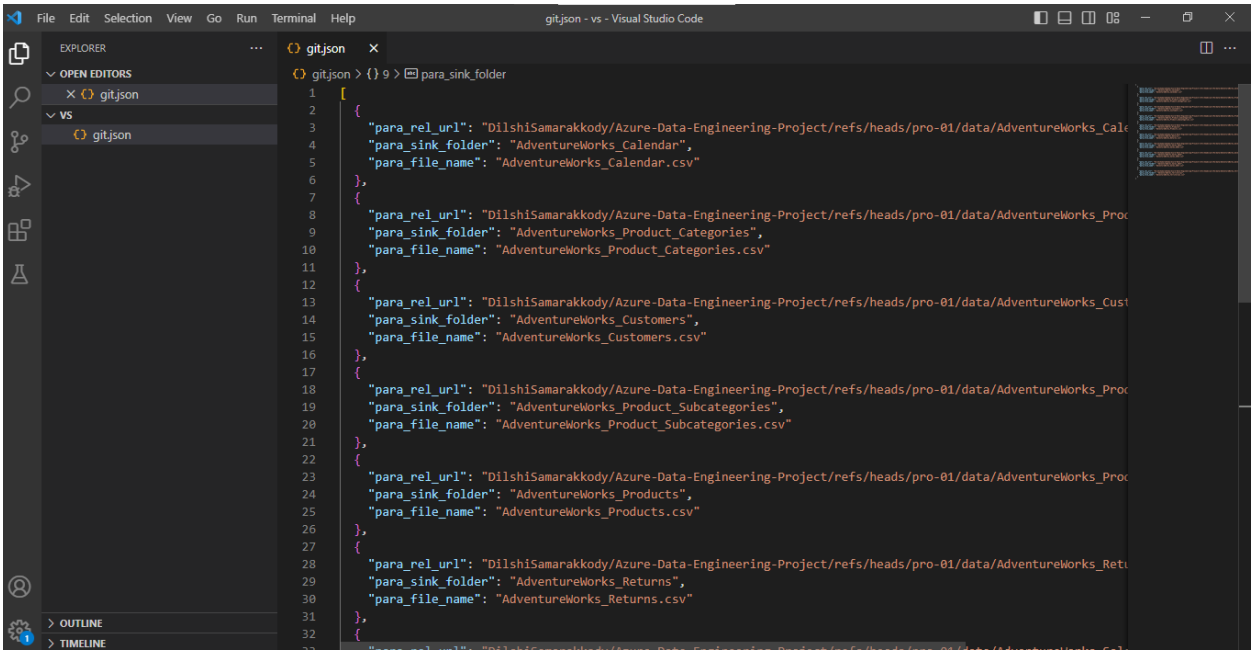
Create ForEach activity



Step 11

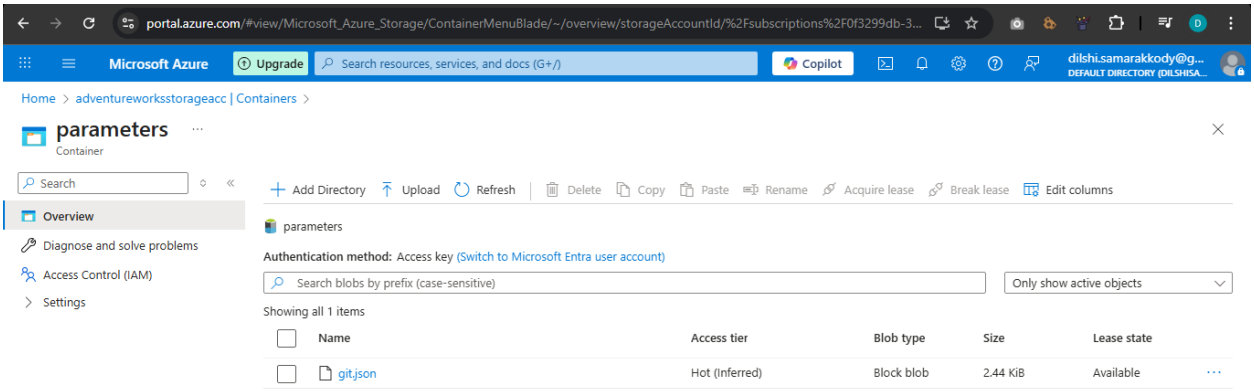
Pass a sequential list of values as an array in json format





Step 12

Create a folder and upload the json file



Lookup Activity Configuration

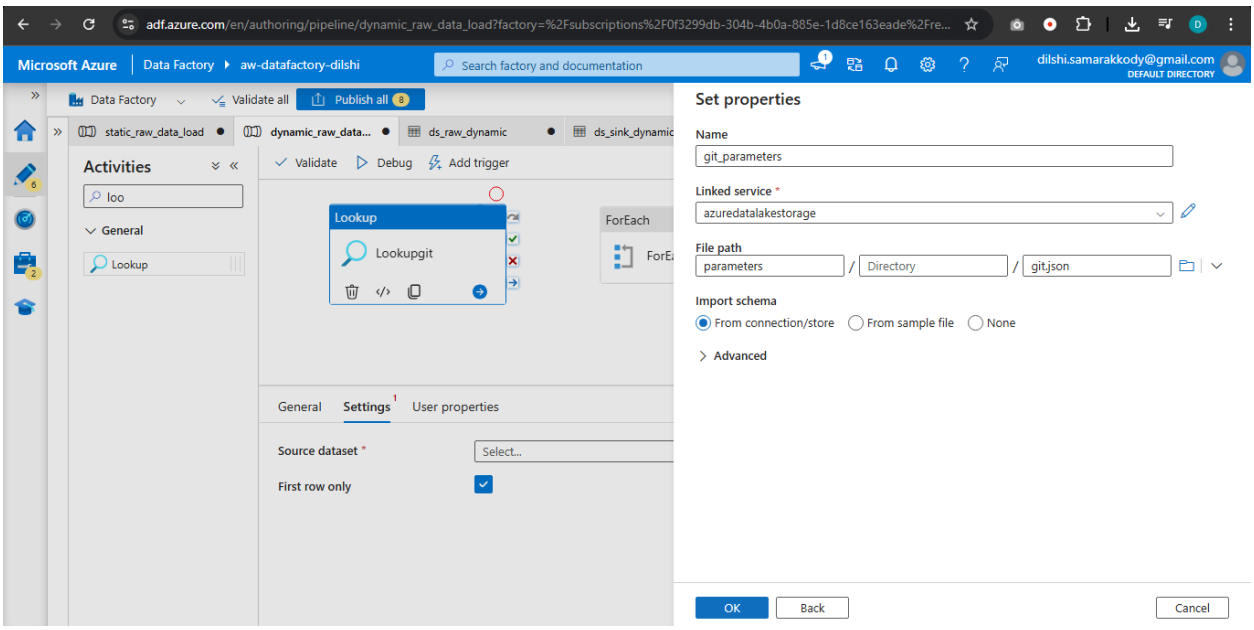
The Lookup activity serves as the foundation for dynamic processing:

- Queries configuration tables or files
- Retrieves source and destination information
- Provides parameters for subsequent activities
- Enables conditional processing logic

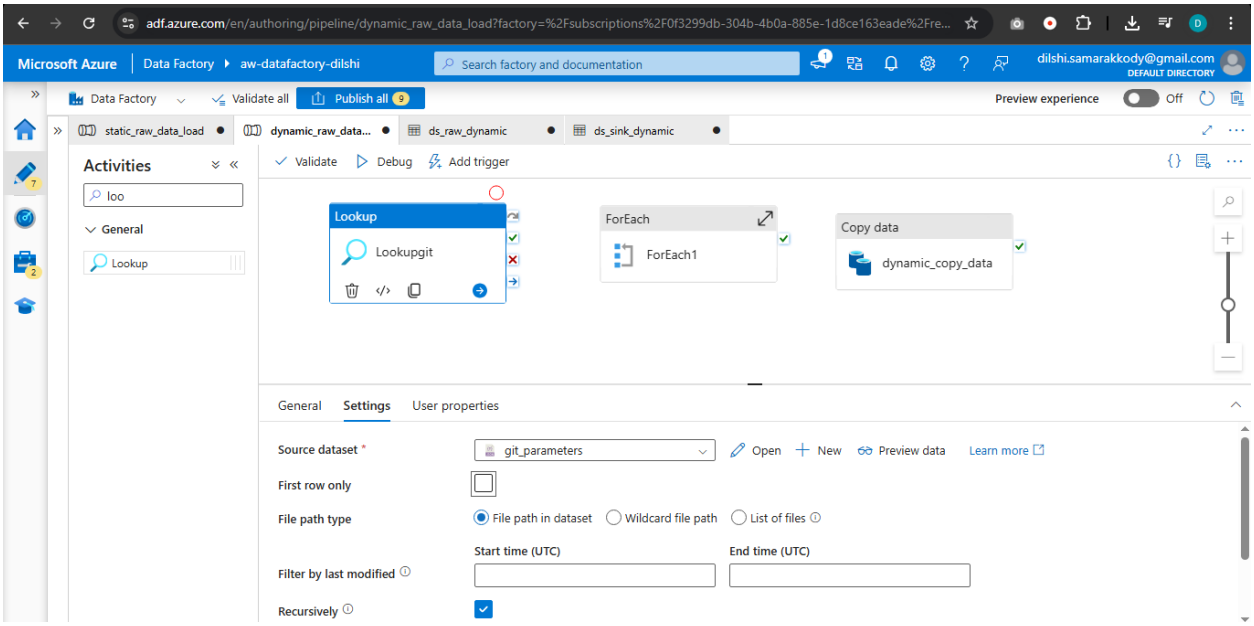
Step 13

Create a Lookup activity to check the output

Select ADLS and Json format for data source

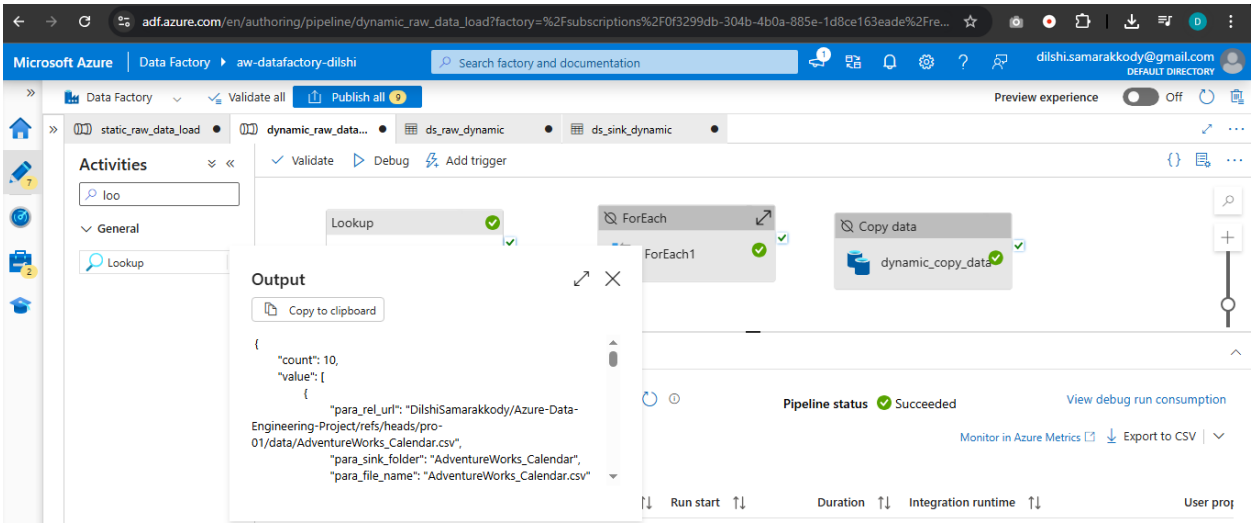


Uncheck first row only



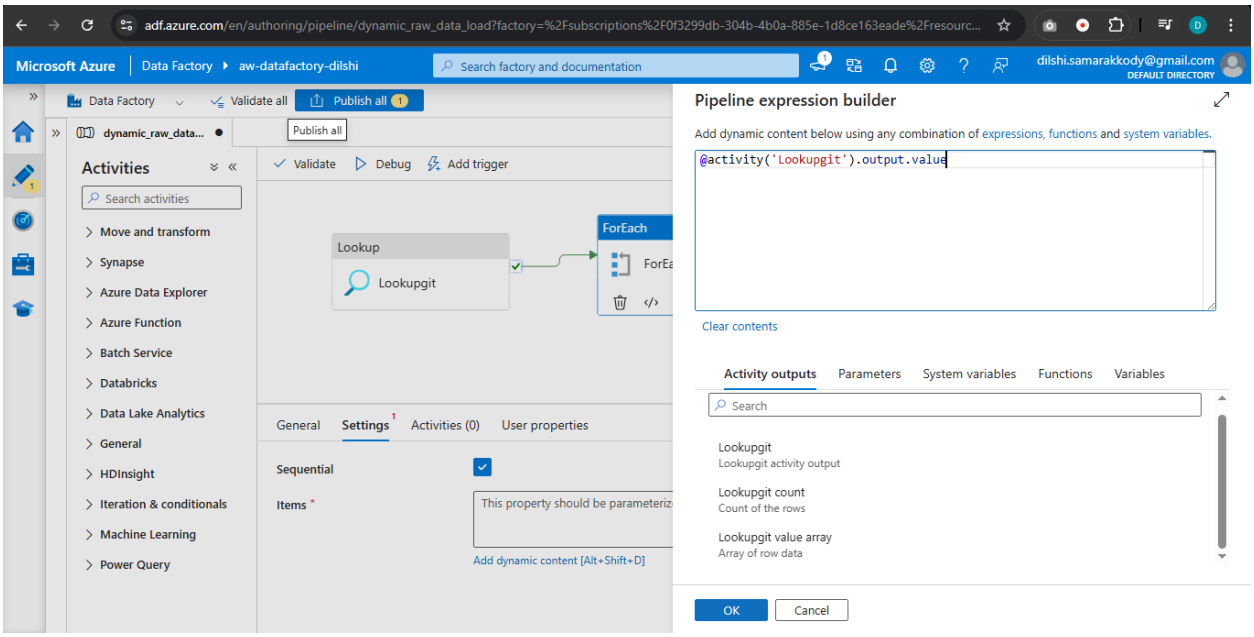
Step 14

Deactivate other activities and Debug only Lookup activity. Check the output.



Step 15

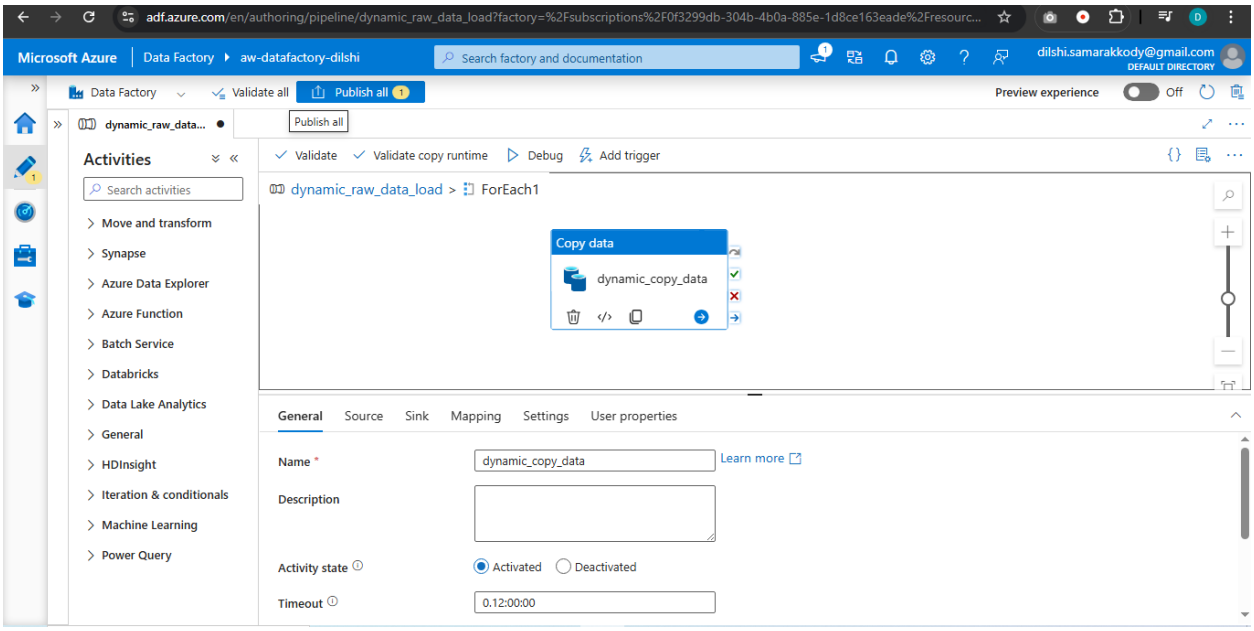
There are 3 outputs choose only value array.



Step 16

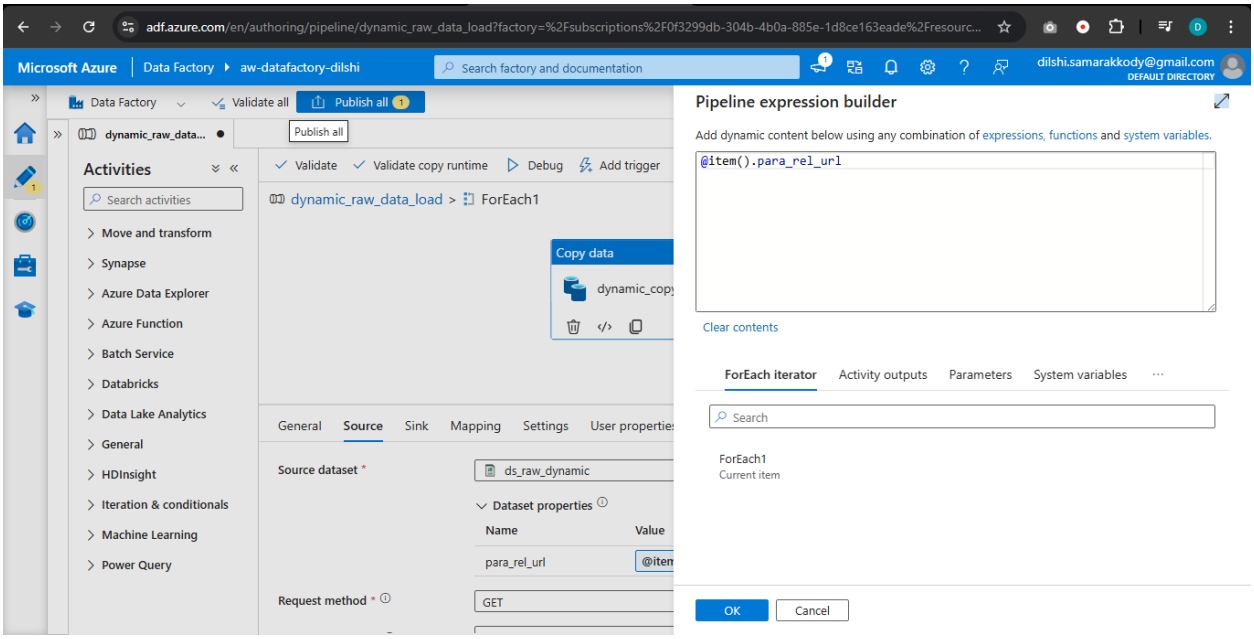
Activate other activities. Connect the Lookup activity to ForEach.

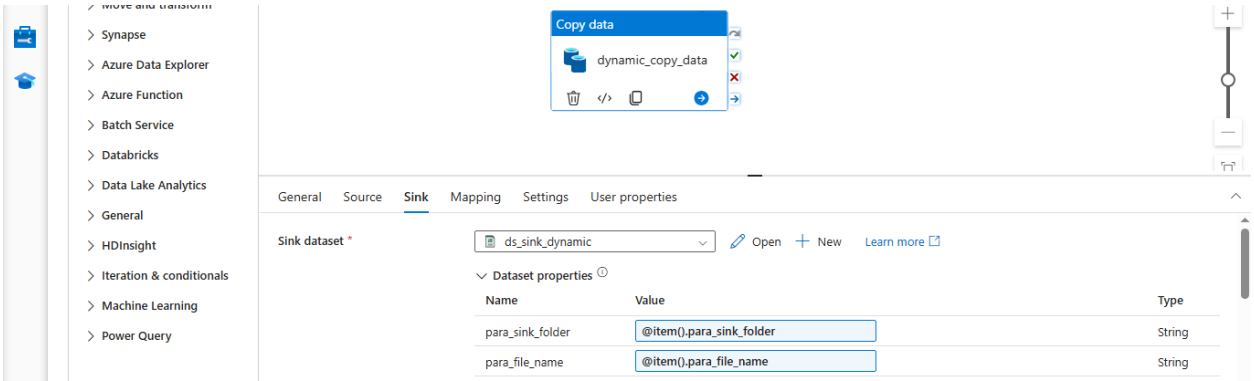
Select ForEach activity and go to activities and copy and paste Copy data activity.



Step 17

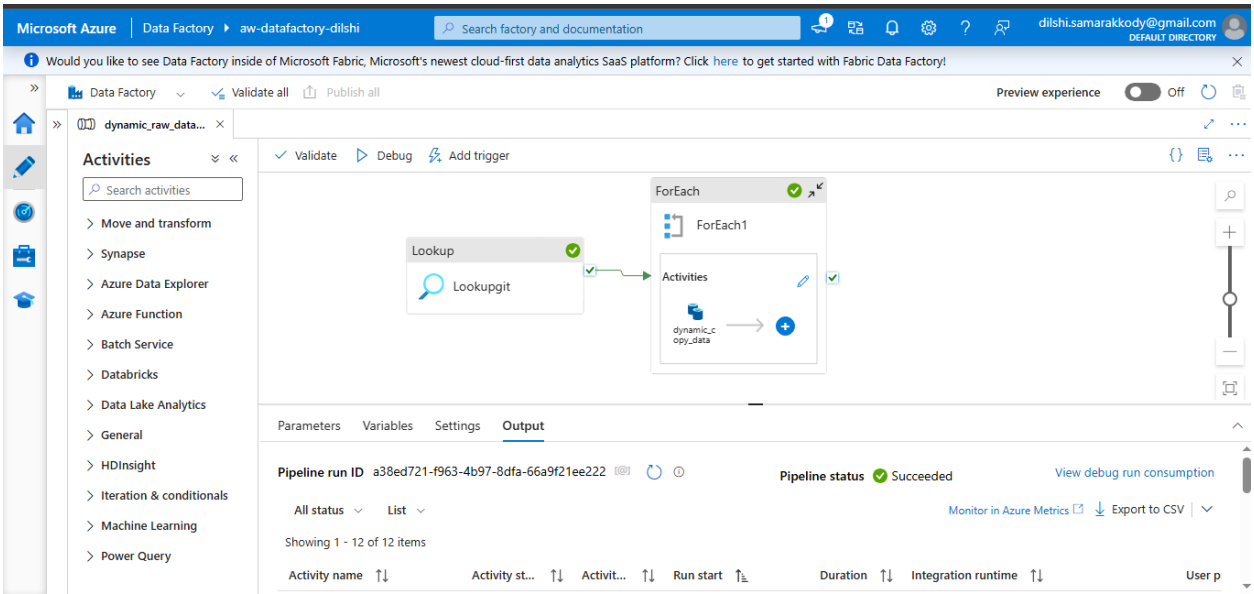
Pass the values for the source and sink dynamic parameters

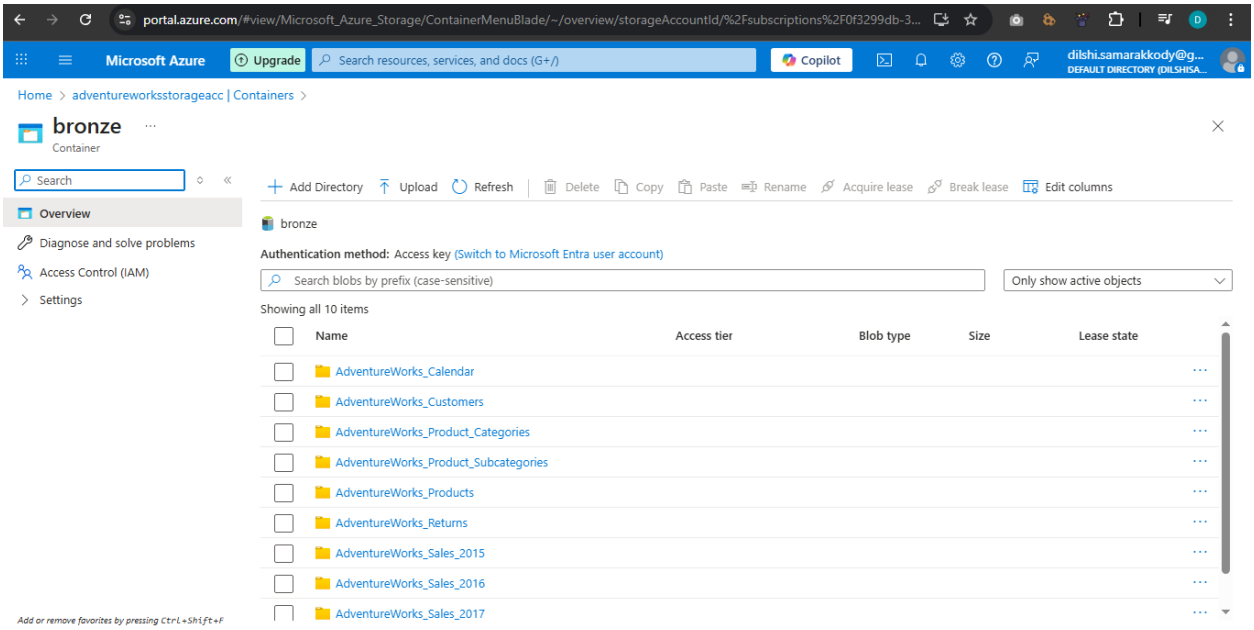




Steps 18

Debug the dynamic pipeline



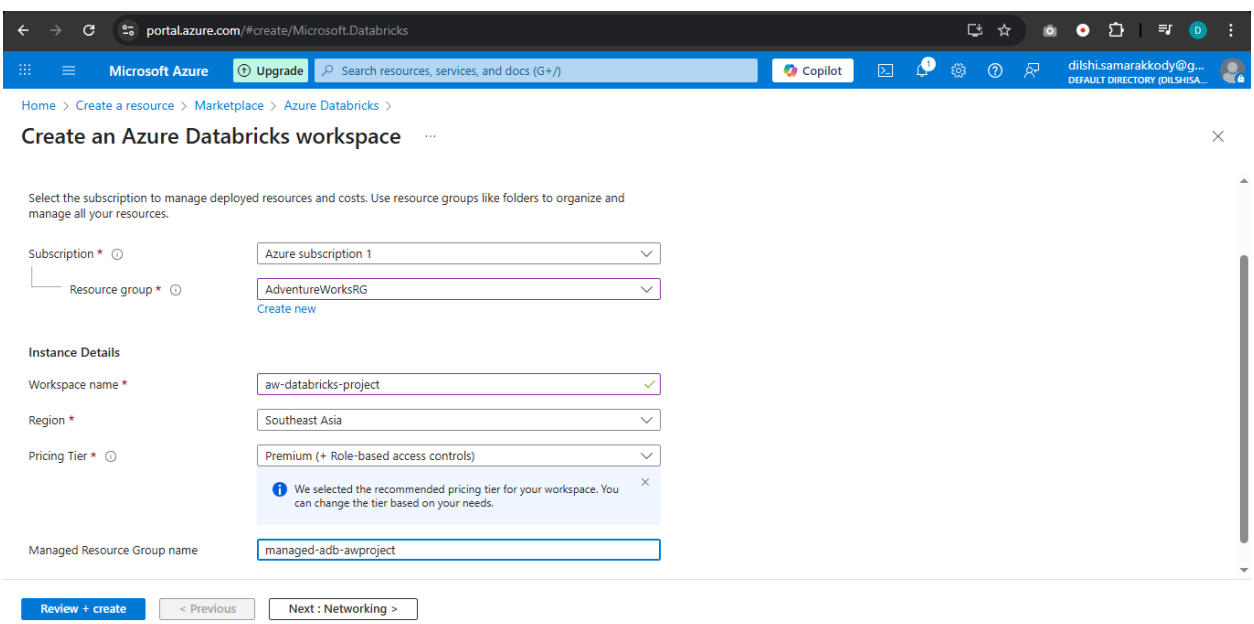


3. Azure Databricks Integration

Azure Databricks provides advanced analytics and machine learning capabilities.

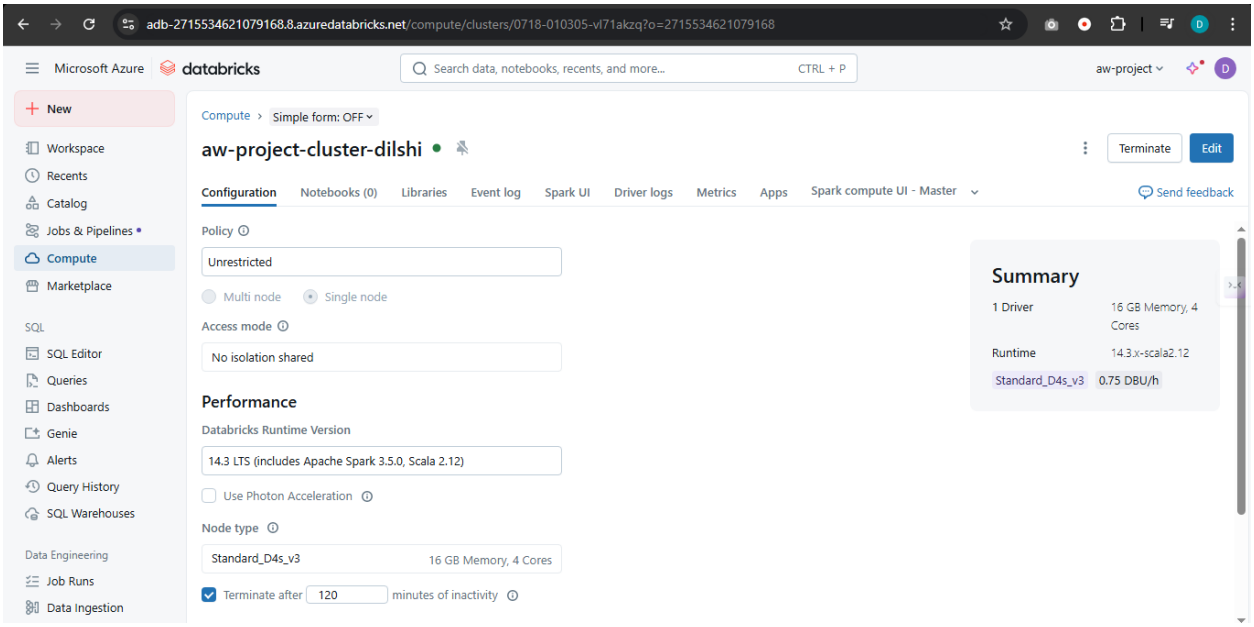
Step 19

Create Databricks workspace.



Step 20

Create a new cluster for workspace

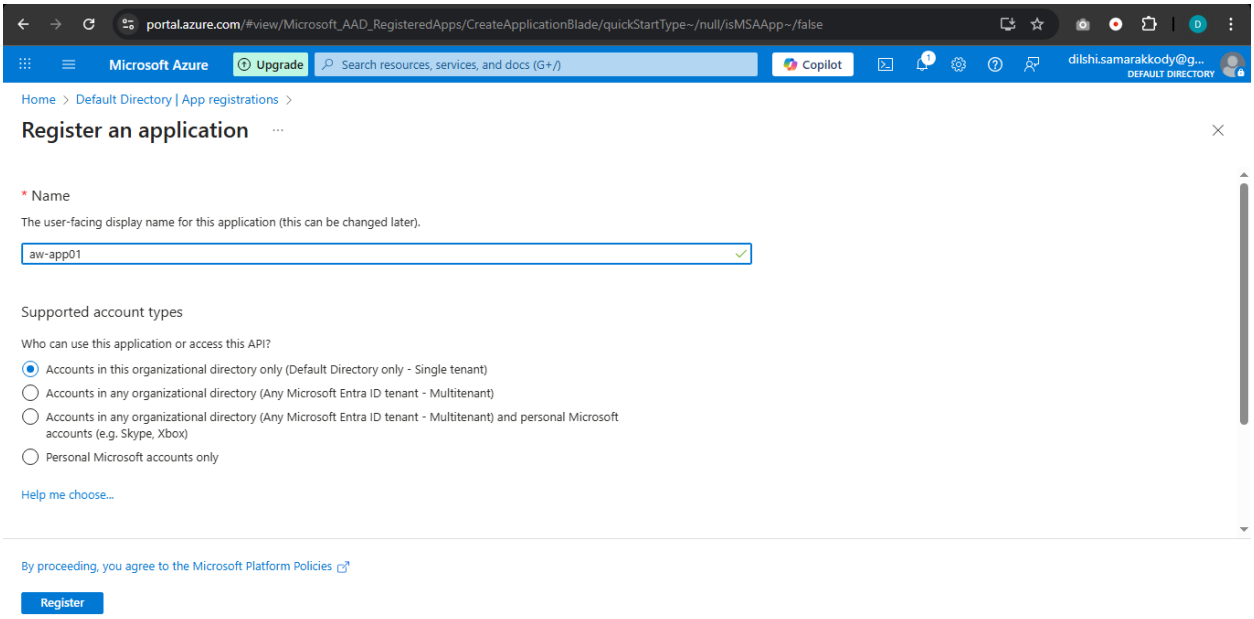


4. Authentication and Authorization

Create an application that accesses data in ADLS and provide credential to Azure Databricks.

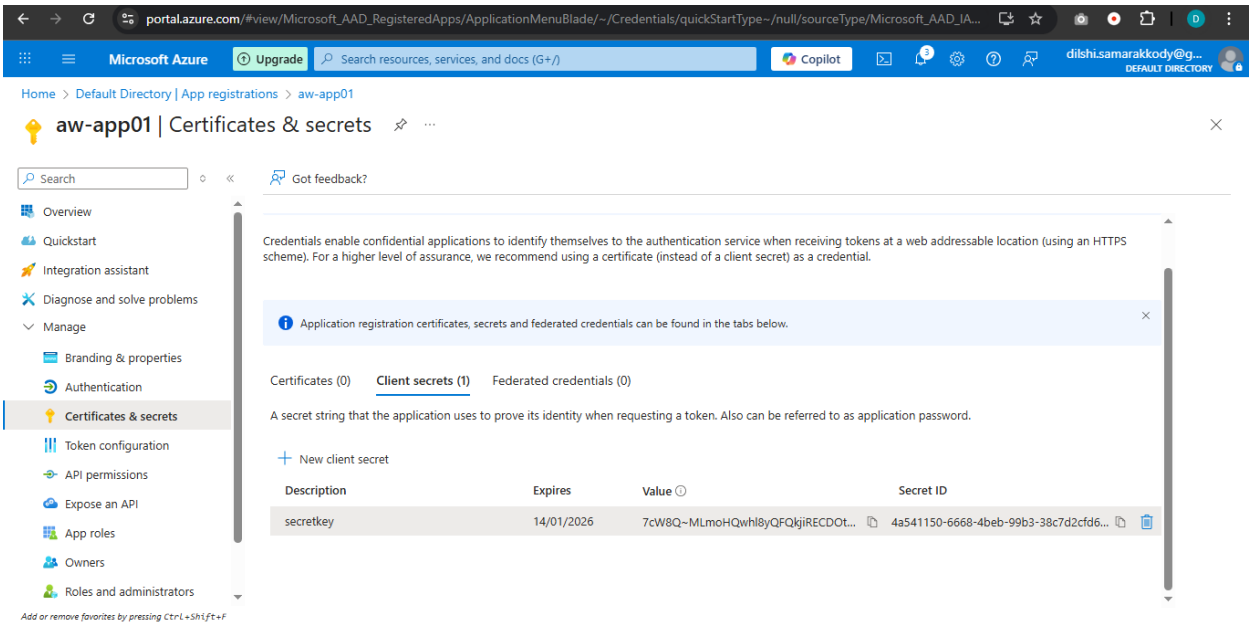
Step 21

Go to App Registration and create a new app



Step 22

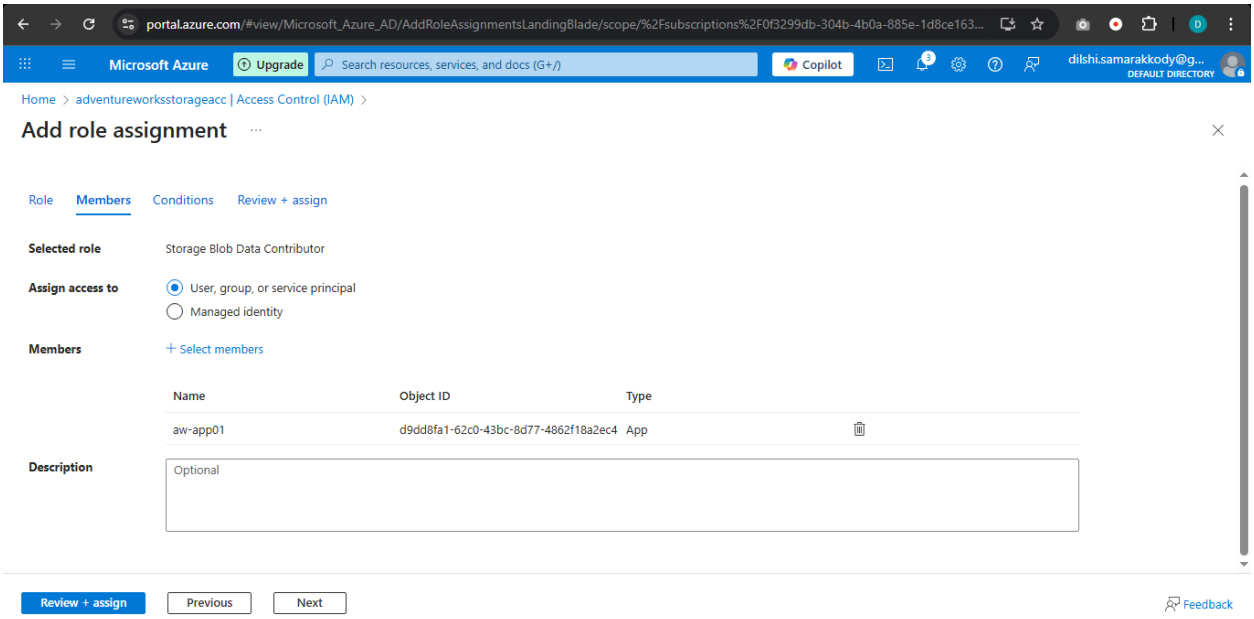
Create a Secret key



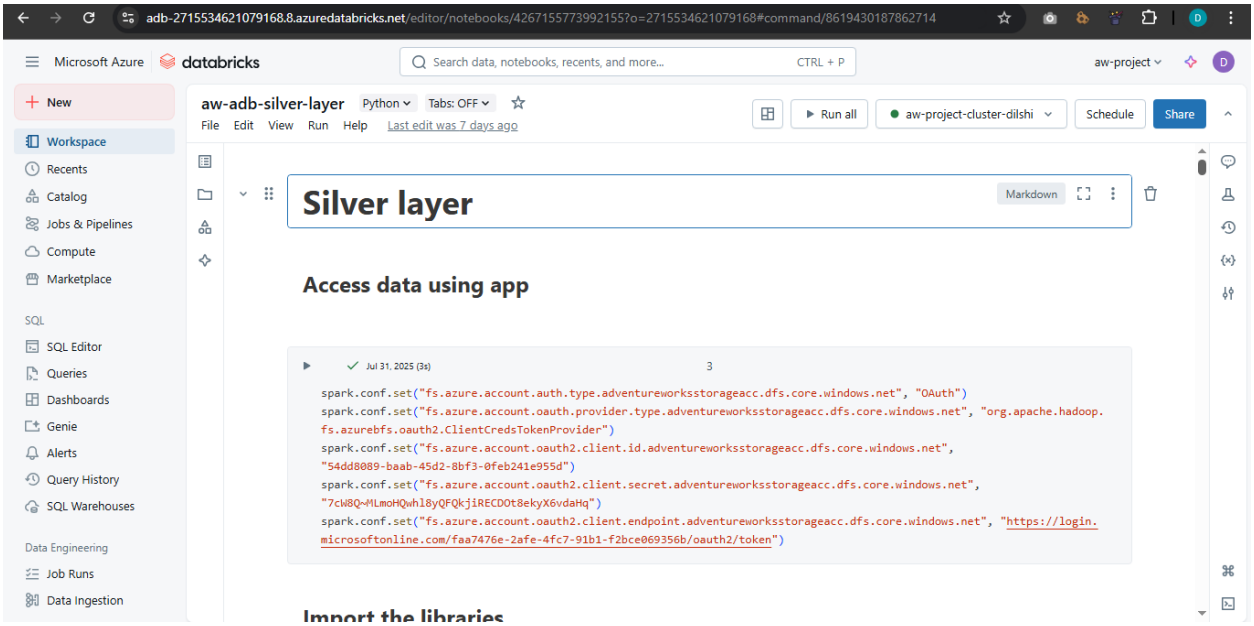
Assign a role to the application that can access the ADLS.

Step 23

Go to storage account and select Access control (IAM). Add a role assignment and select Storage Blob Contributor which gives both read and write permission to ADLS

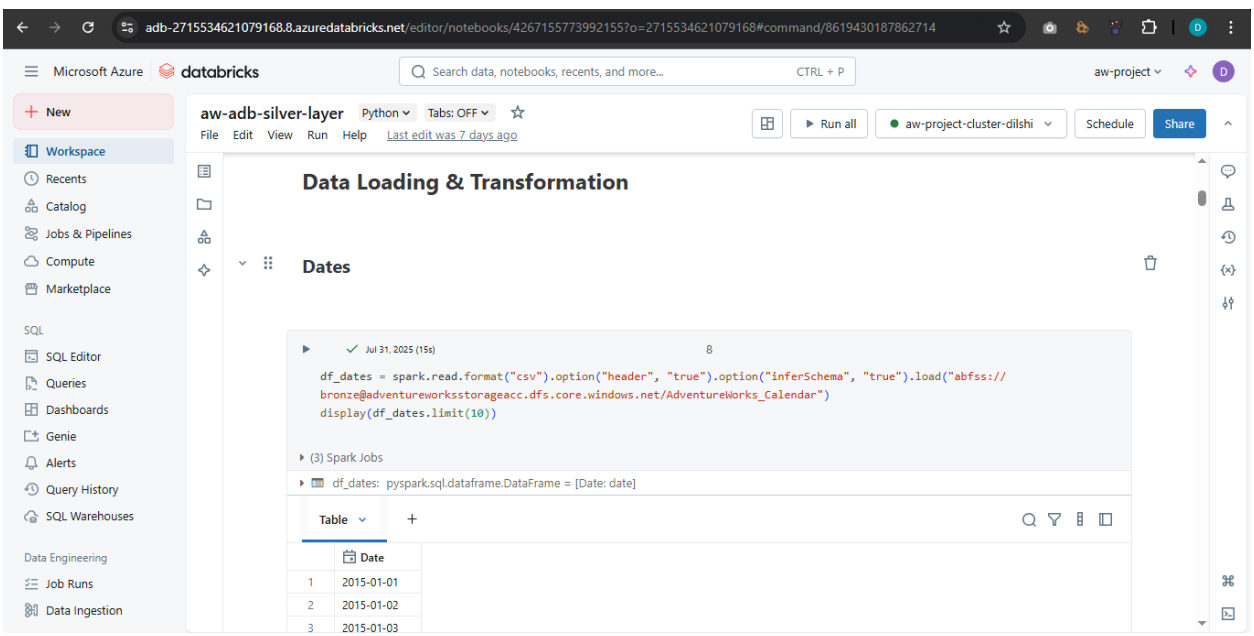


5. Data access using app



Step 24

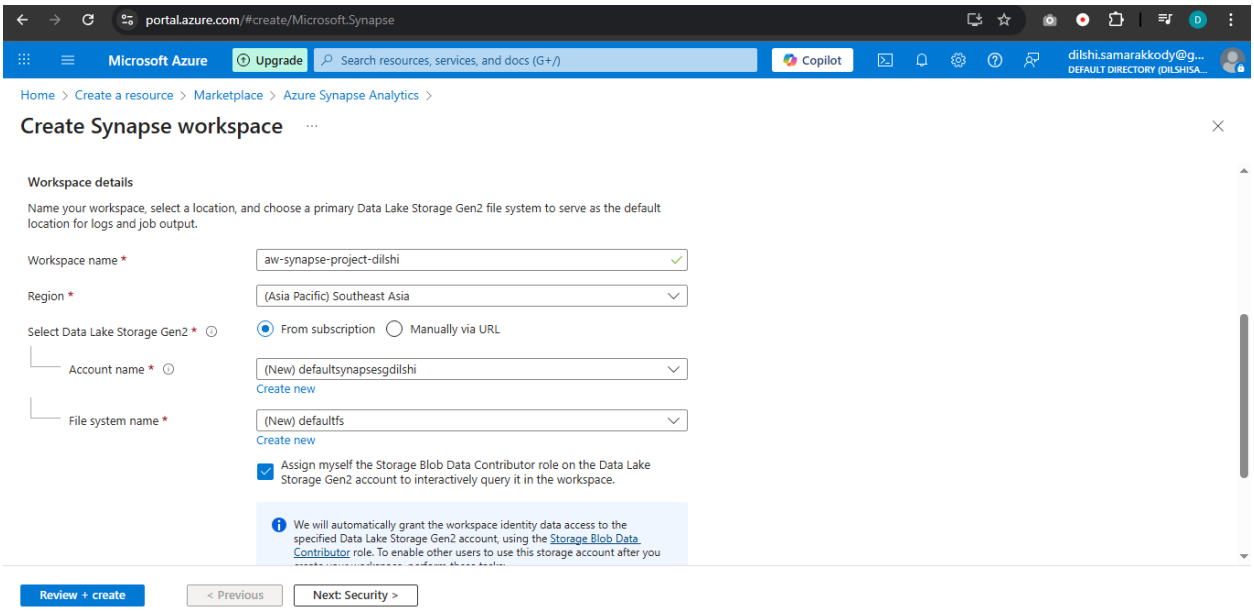
Implement the Silver Layer

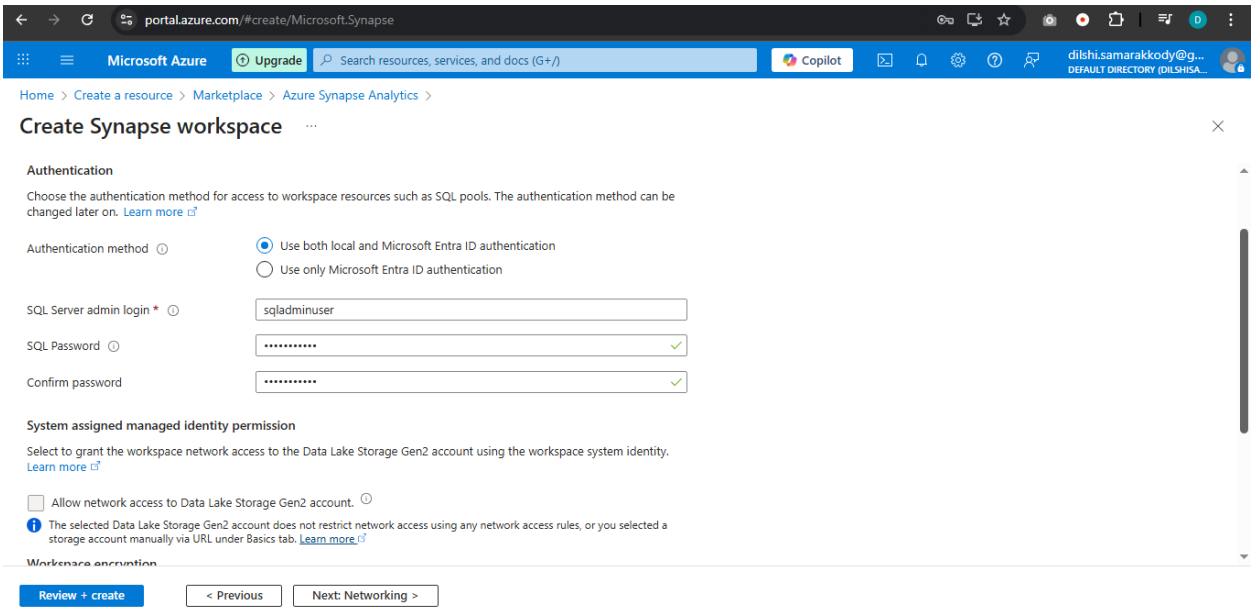


6. Azure Synapse Analytics Implementation

Step 25

Create synapse workspace

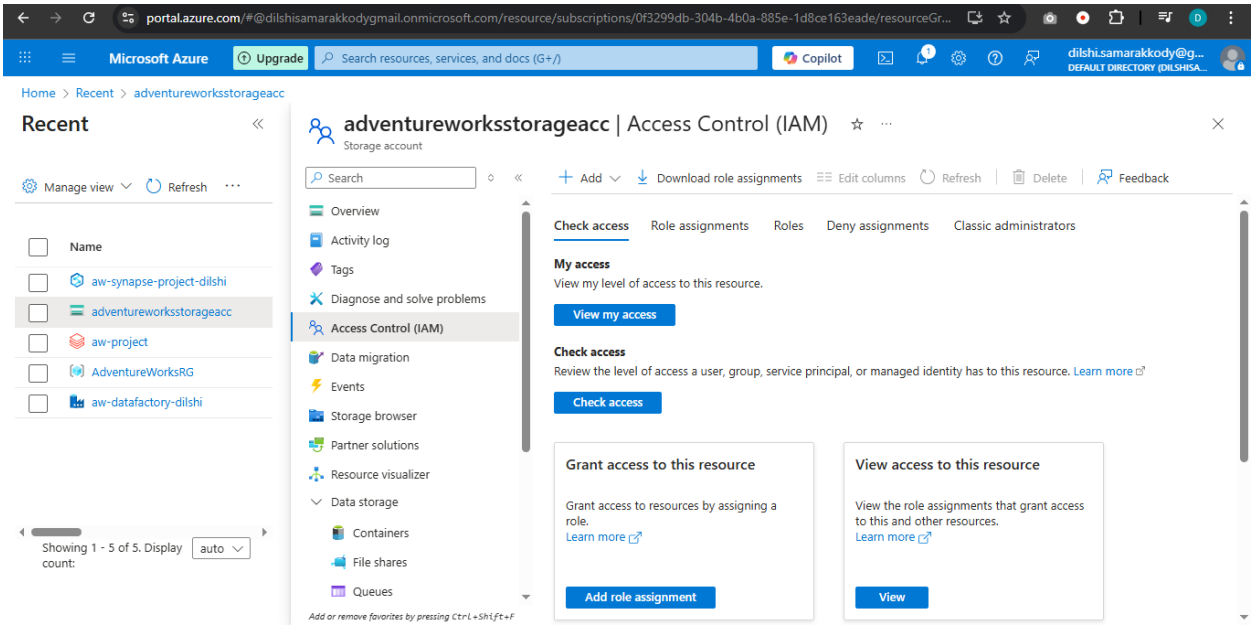




Allow Azure Synapse Analytics to access data stored in ADLS using managed identity

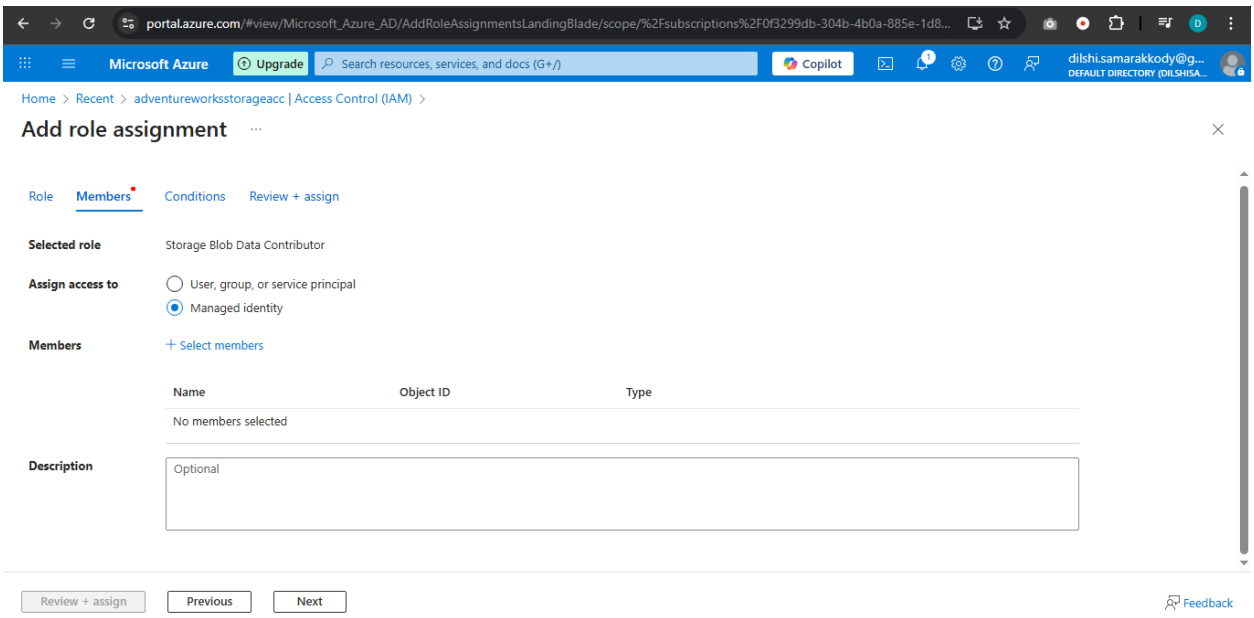
Step 26

Create Managed identity



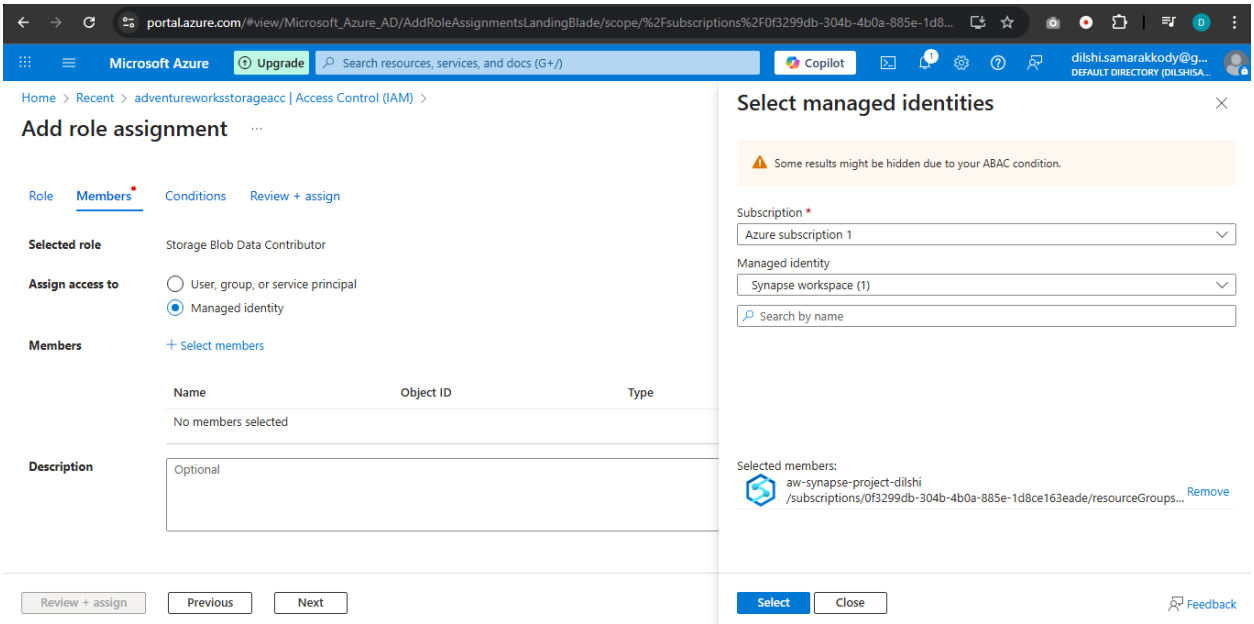
Step 27

Select managed identity



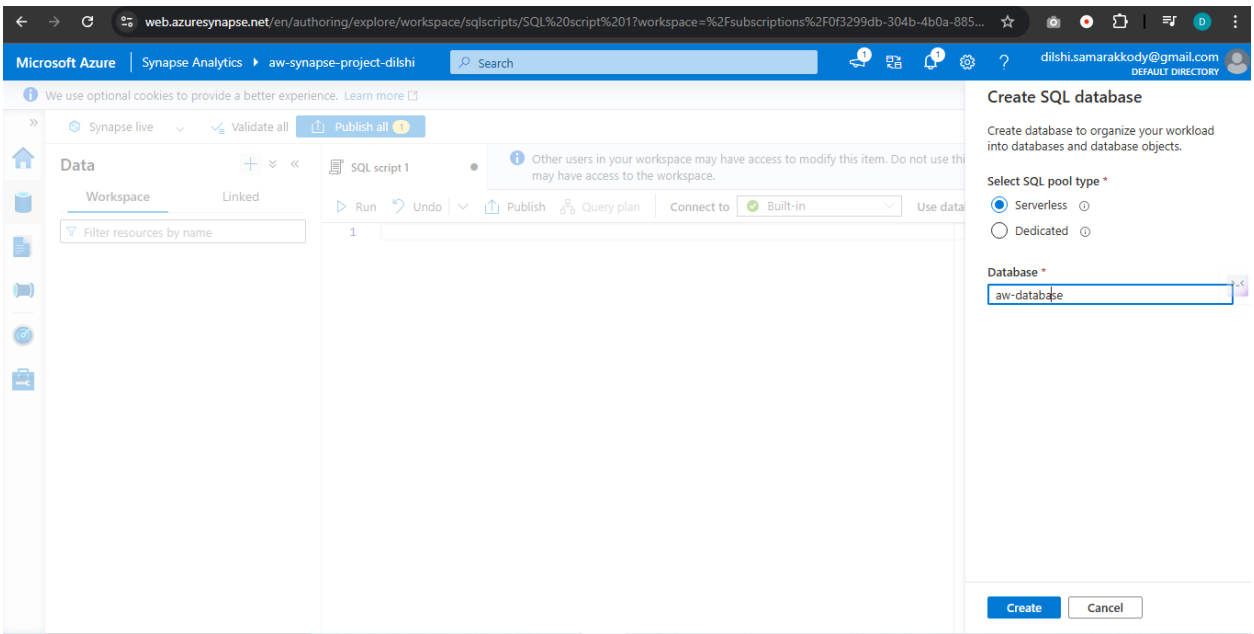
Step 28

Assign a role on Managed Identity



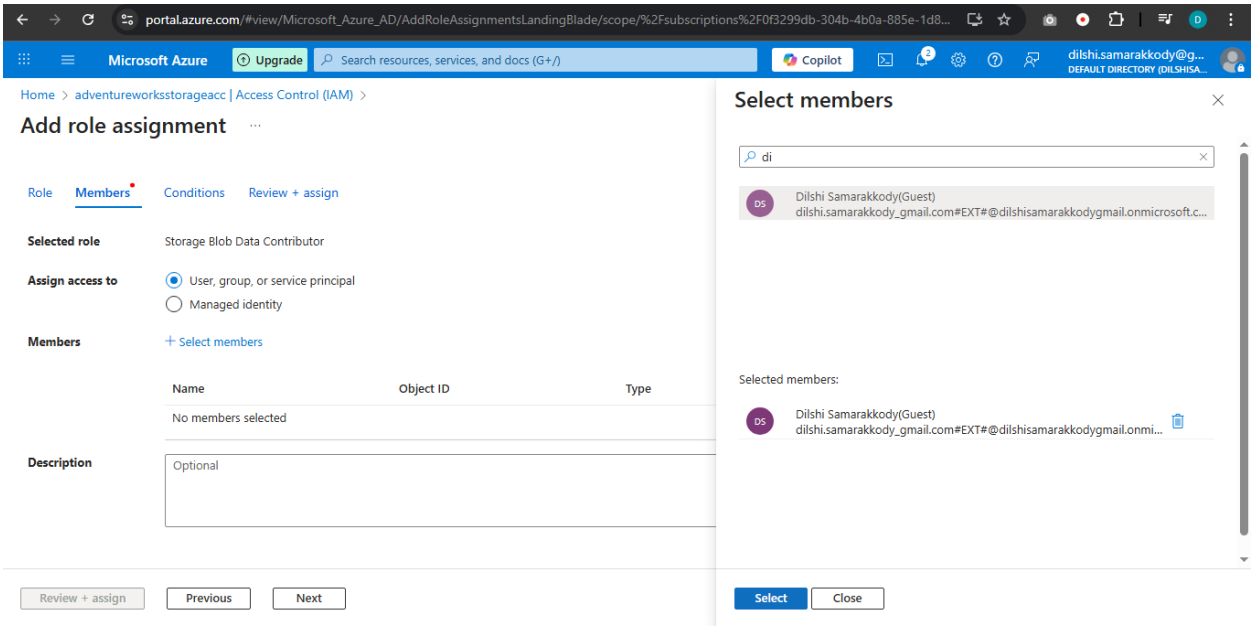
Step 29

Create Serverless database

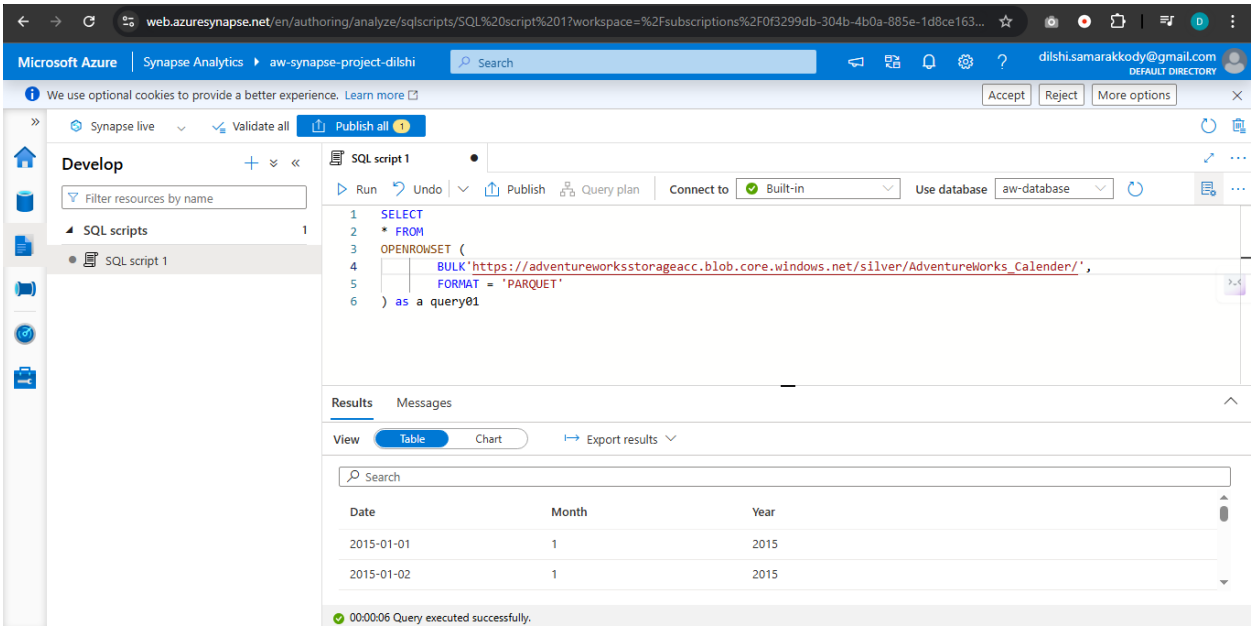


Step 30

Go to storage account and create role assign for me to access data in ADLS

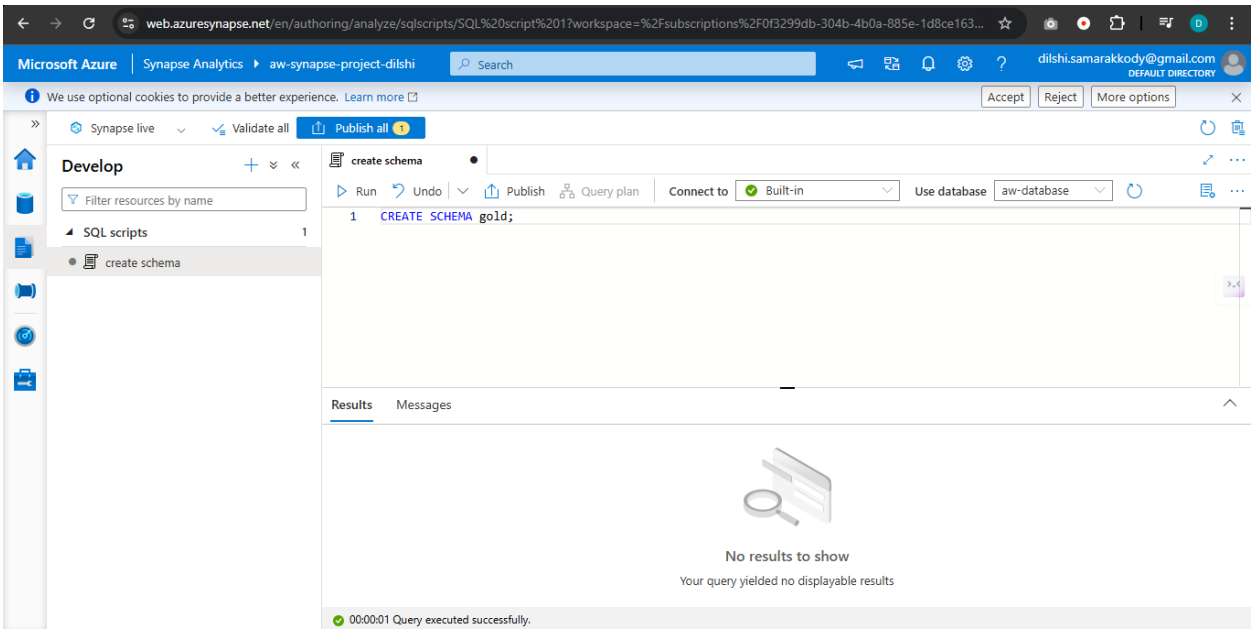


Check data preview



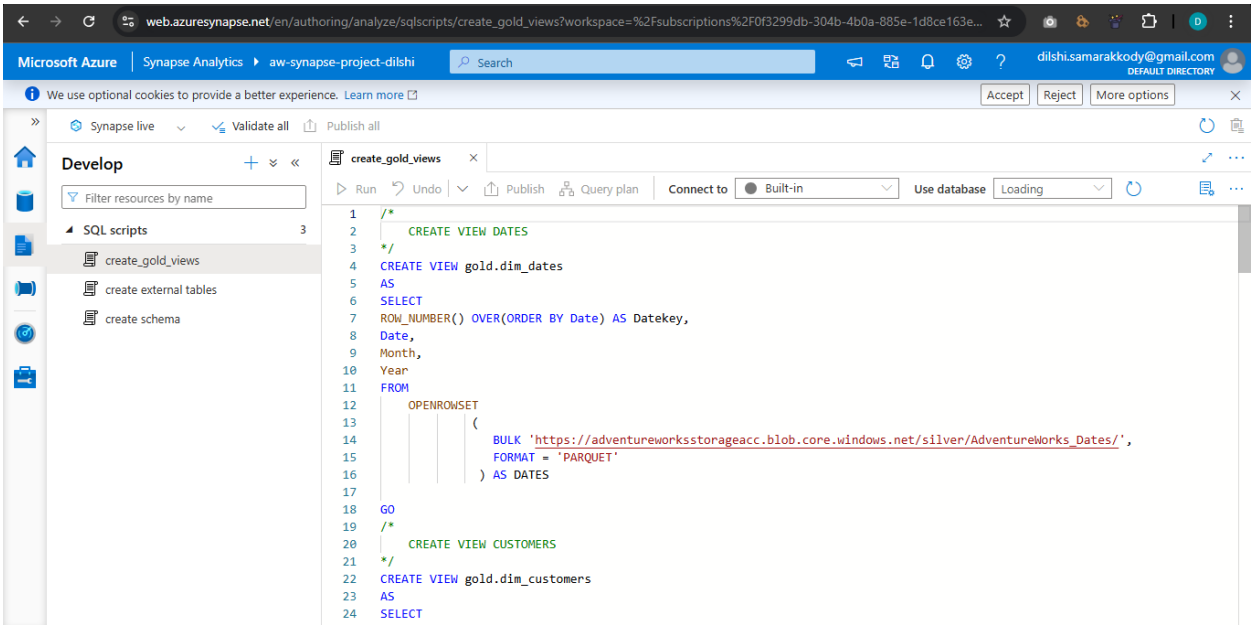
Step 31

Create a schema



Step 32

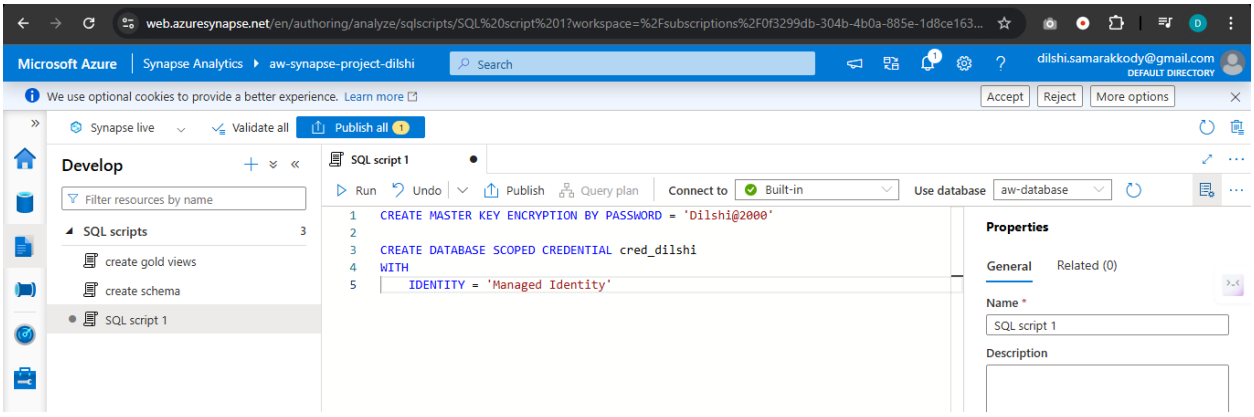
Create views for dimension and fact tables



Create master key if not available

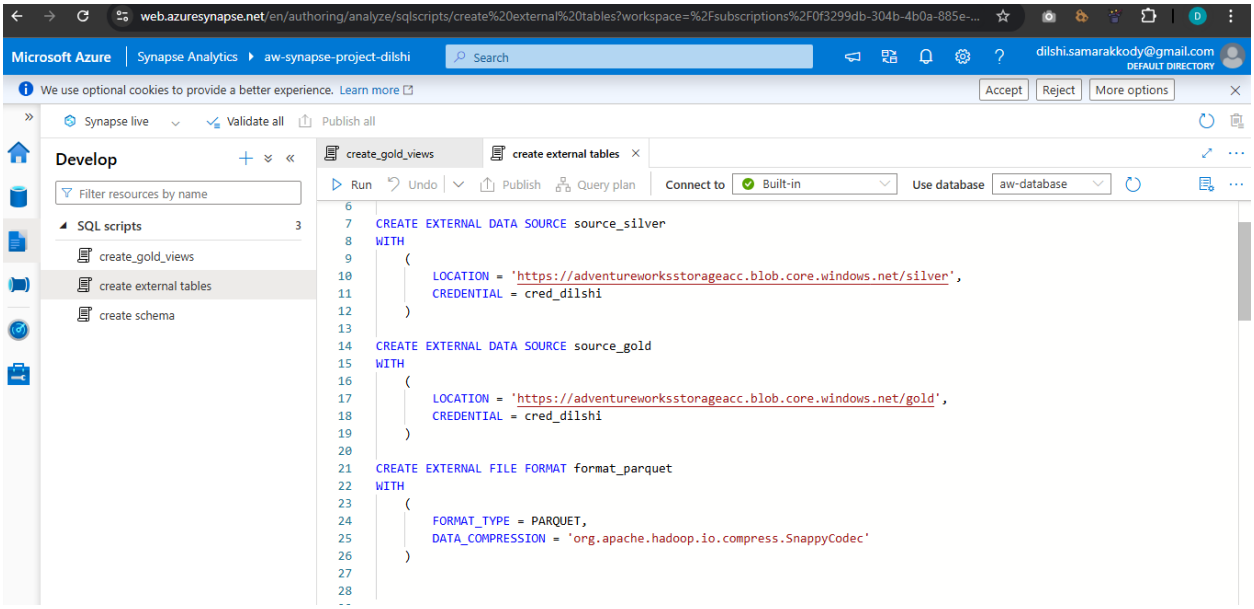
Step 33

Create database scoped credential

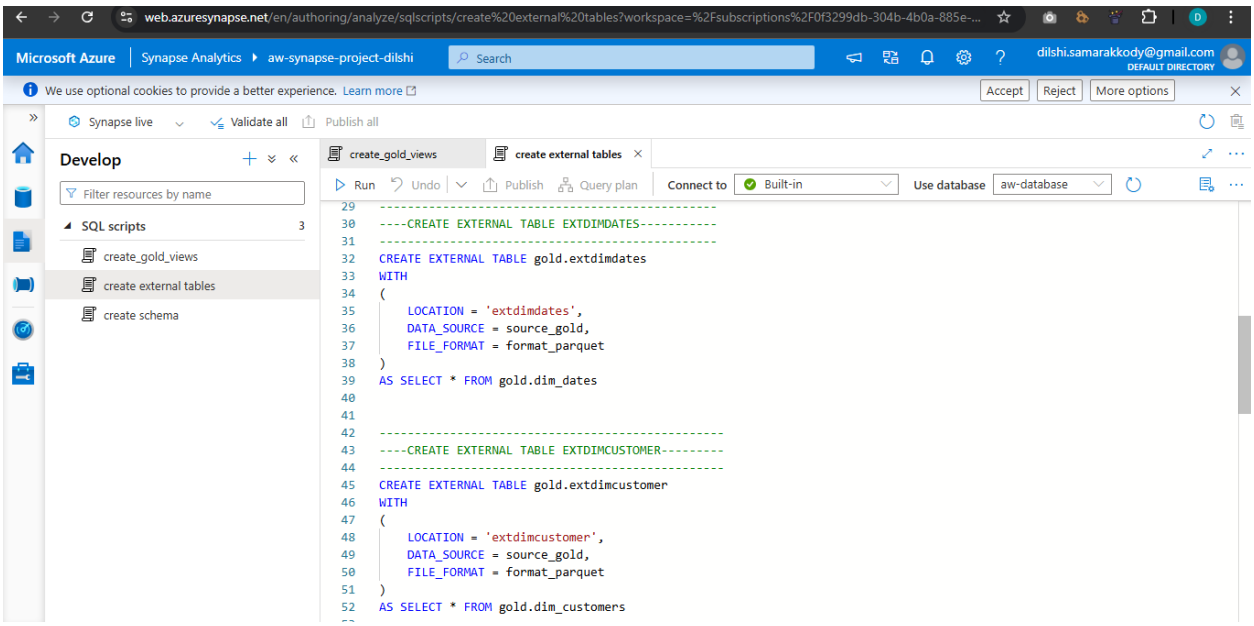


Step 34

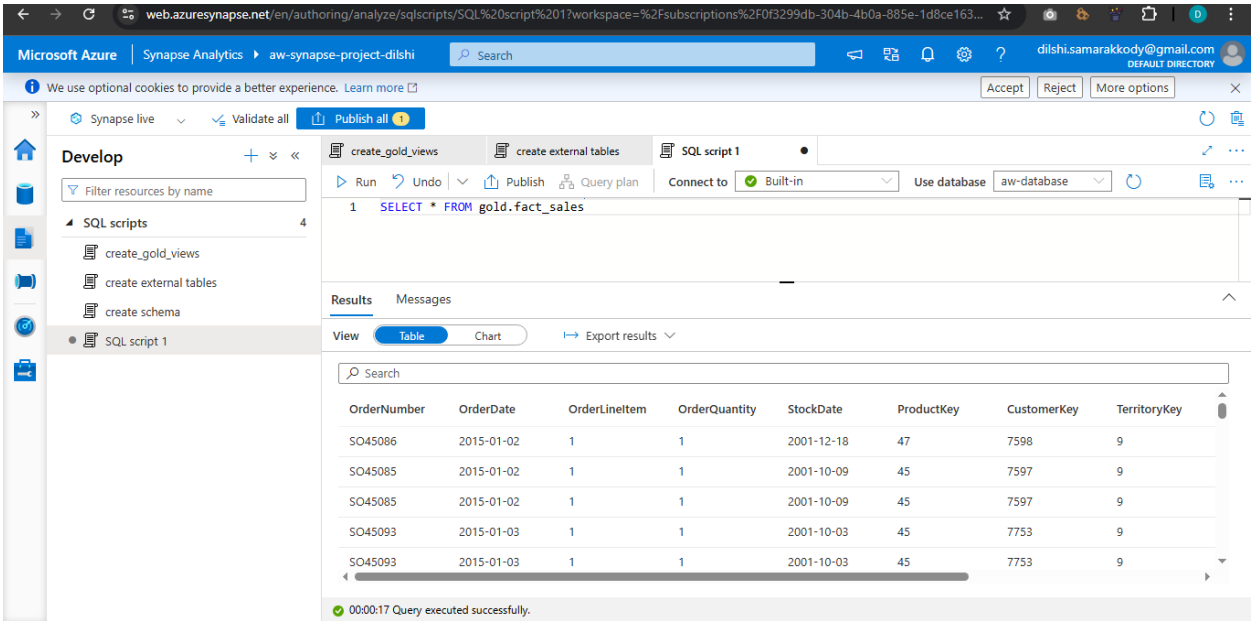
Creating external data source and create external file format



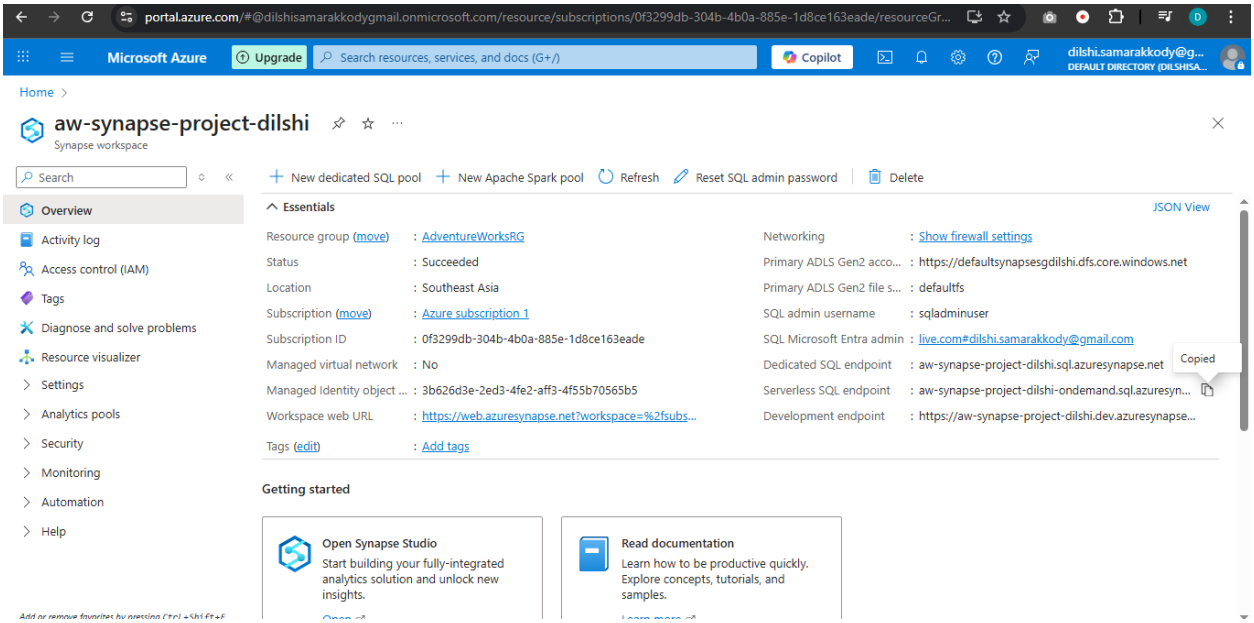
Create external table



Check data in gold layer

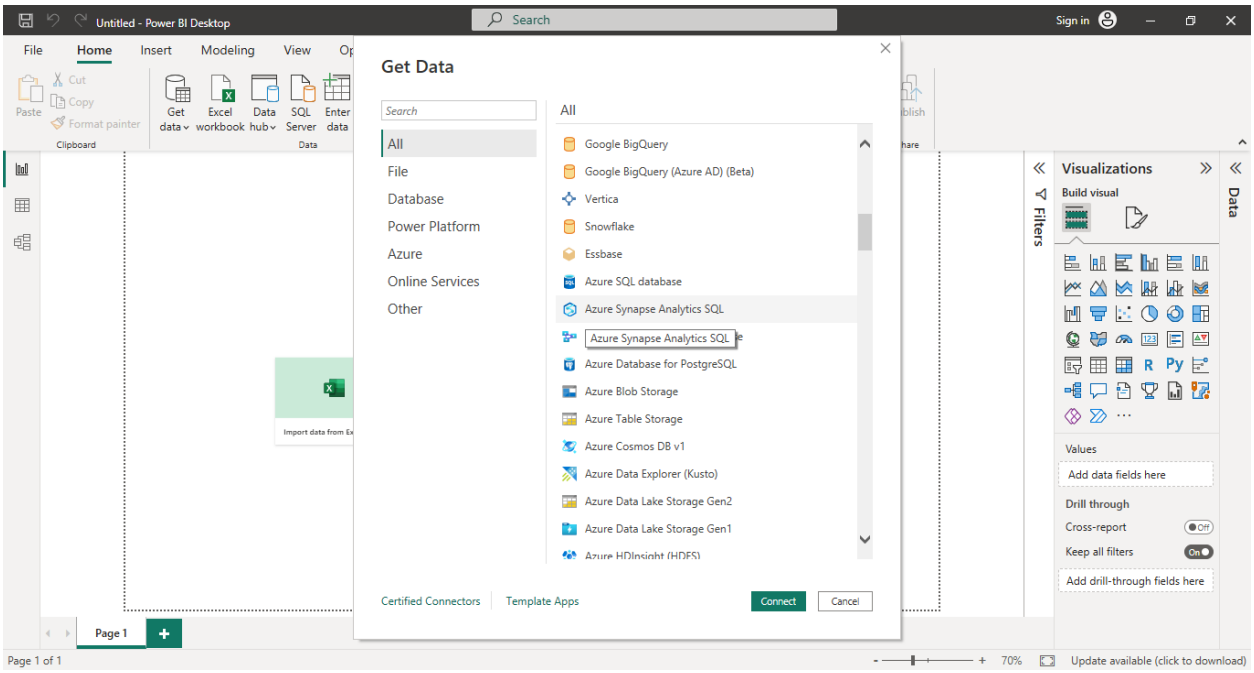


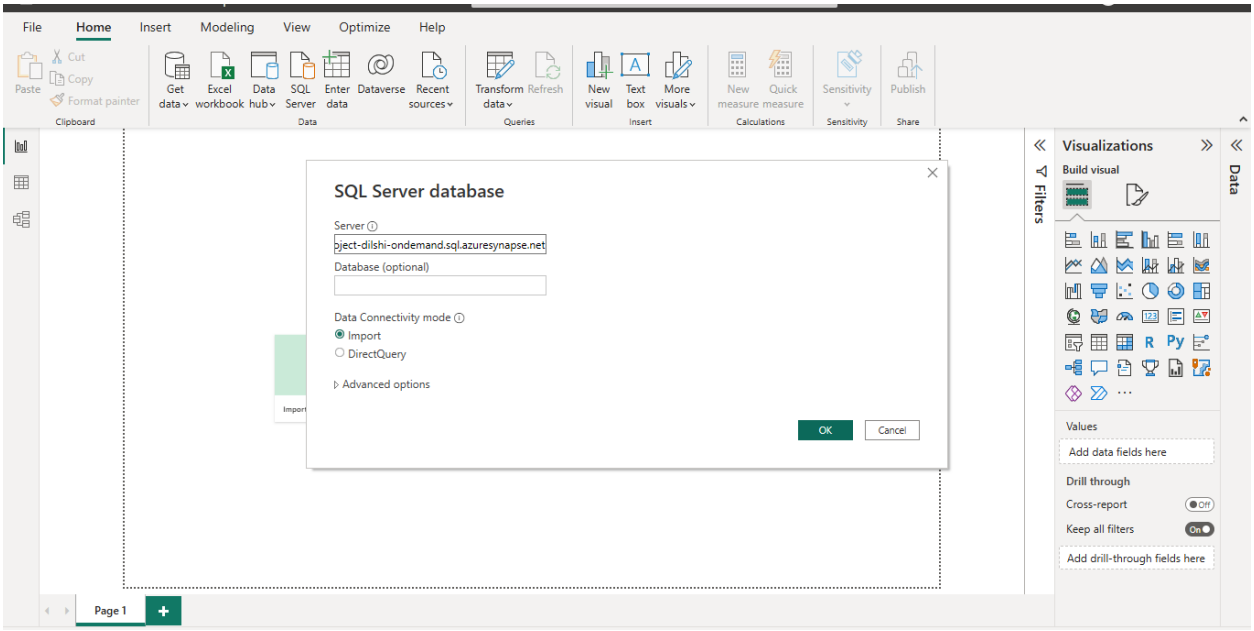
Copy the serverless sql end point and create connection with power bi

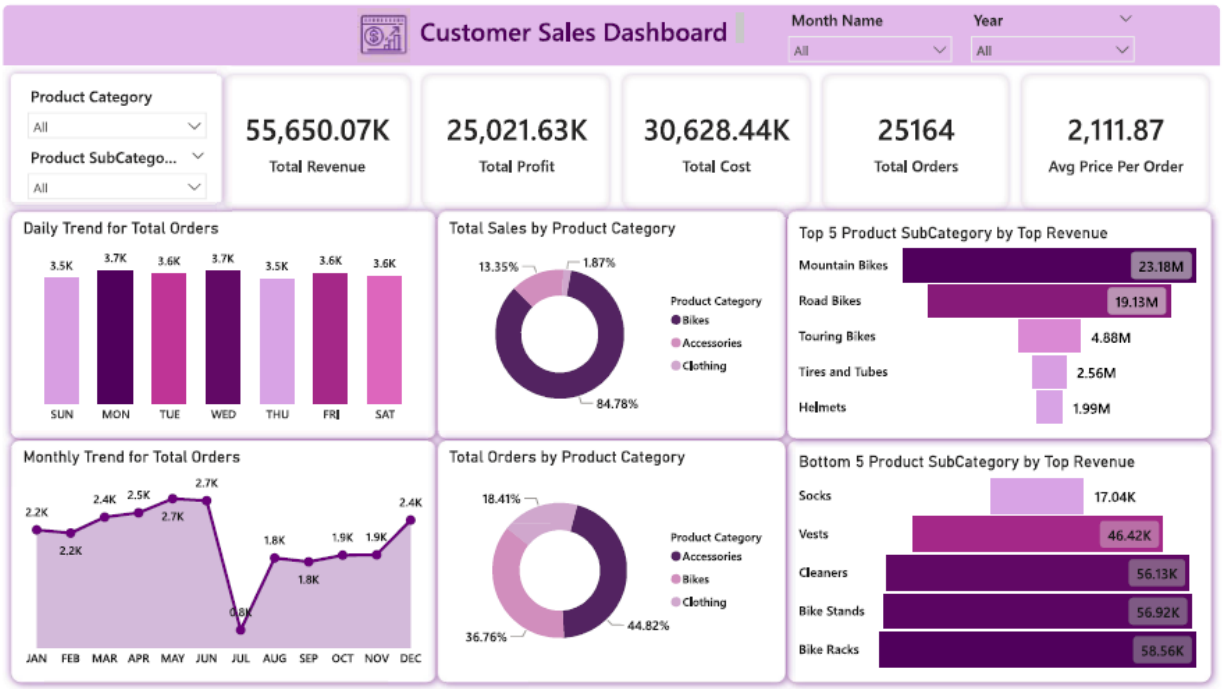


7. Power BI dashboard implementation

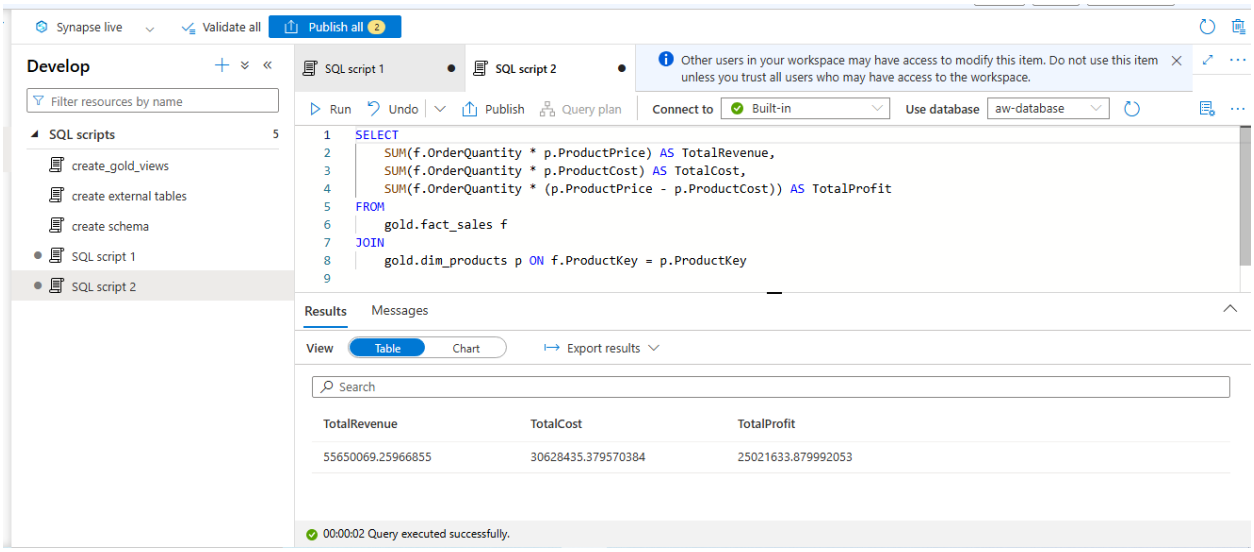
Load data from Azure Synapse into Power BI

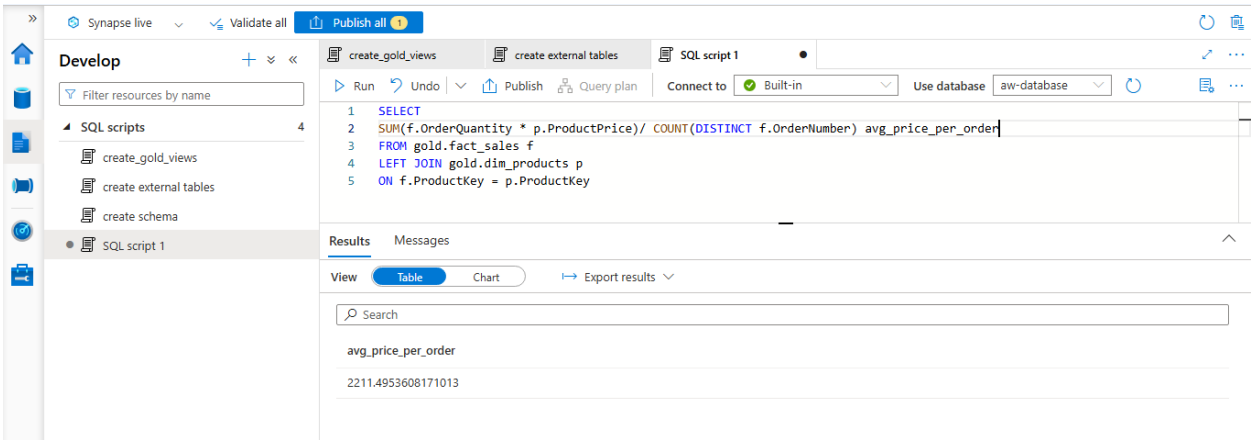






KPI validation





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

Accept | Reject | More options

Synapse live | Validate all | Publish all 2

Develop

Filter resources by name

- SQL scripts 5
  - create_gold_views
  - create external tables
  - create schema
  - SQL script 1
  - SQL script 2

SQL script 1 | SQL script 2

Other users in your workspace may have access to modify this item. Do not use this item unless you trust all users who may have access to the workspace.

Run | Undo | Publish | Query plan | Connect to: Built-in | Use database: aw-database

```
select
p.ProductCategory,
SUM(f.OrderQuantity * p.ProductPrice) as tot_price
from gold.fact_sales f
LEFT JOIN gold.dim_products p
ON f.ProductKey = p.ProductKey
group by p.ProductCategory
```

Results | Messages

View: Table | Chart | Export results

Search

ProductCategory	tot_price
Accessories	7427842.503841094
Bikes	47180481.20360897
Clothing	1041745.5521997635

00:00:02 Query executed successfully.

web.azuresynapse.net/en/authoring/analyze/sqlscripts/SQL%20script%201?workspace=%2Fsubscriptions%2F0f3299db-304b-4b0a-885e-1d8ce163...

Microsoft Azure | Synapse Analytics ▸ aw-synapse-project-dilshi | Search | dilshi.samarakkody@gmail.com DEFAULT DIRECTORY

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

Accept | Reject | More options

Synapse live | Validate all | Publish all 2

Develop

Filter resources by name

- SQL scripts 5
  - create_gold_views
  - create external tables
  - create schema
  - SQL script 1
  - SQL script 2

SQL script 1 | SQL script 2

Other users in your workspace may have access to modify this item. Do not use this item unless you trust all users who may have access to the workspace.

Run | Undo | Publish | Query plan | Connect to: Built-in | Use database: aw-database

```
select
p.ProductSubCategory,
SUM(f.OrderQuantity * p.ProductPrice) as tot_price
from gold.fact_sales f
LEFT JOIN gold.dim_products p
ON f.ProductKey = p.ProductKey
group by p.ProductSubCategory
```

Results | Messages

View: Table | Chart | Export results

Search

ProductSubCategory	tot_price
Bike Racks	58560
Bike Stands	56922
Bottles and Cages	1701606.5899979584

00:00:02 Query executed successfully.