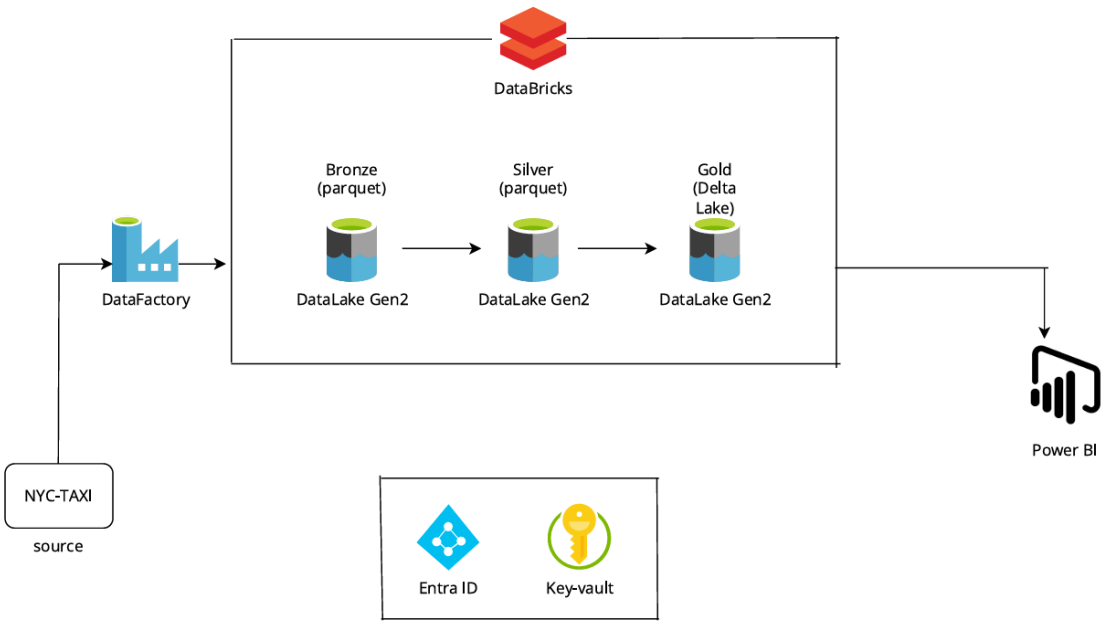
Architecture:



Environment Setup:

1. Create a Resource Group for the Project.

2. Create a Azure Data Factory.

3. Create a Azure Data Lake gen 2 storage account.

    Under storage account created bronze, silver and gold container.
    Bronze → to store the raw data
    Silver → to store the processed data
    Gold → to store the cleaned data that connect to power bi for end users

4. Create a Azure Databricks.

5. Create a Azure Key Vault.

6. Create a Service Principal to connect the ADLS gen2 storage account with Databricks.

7. Power BI

Dataset required:

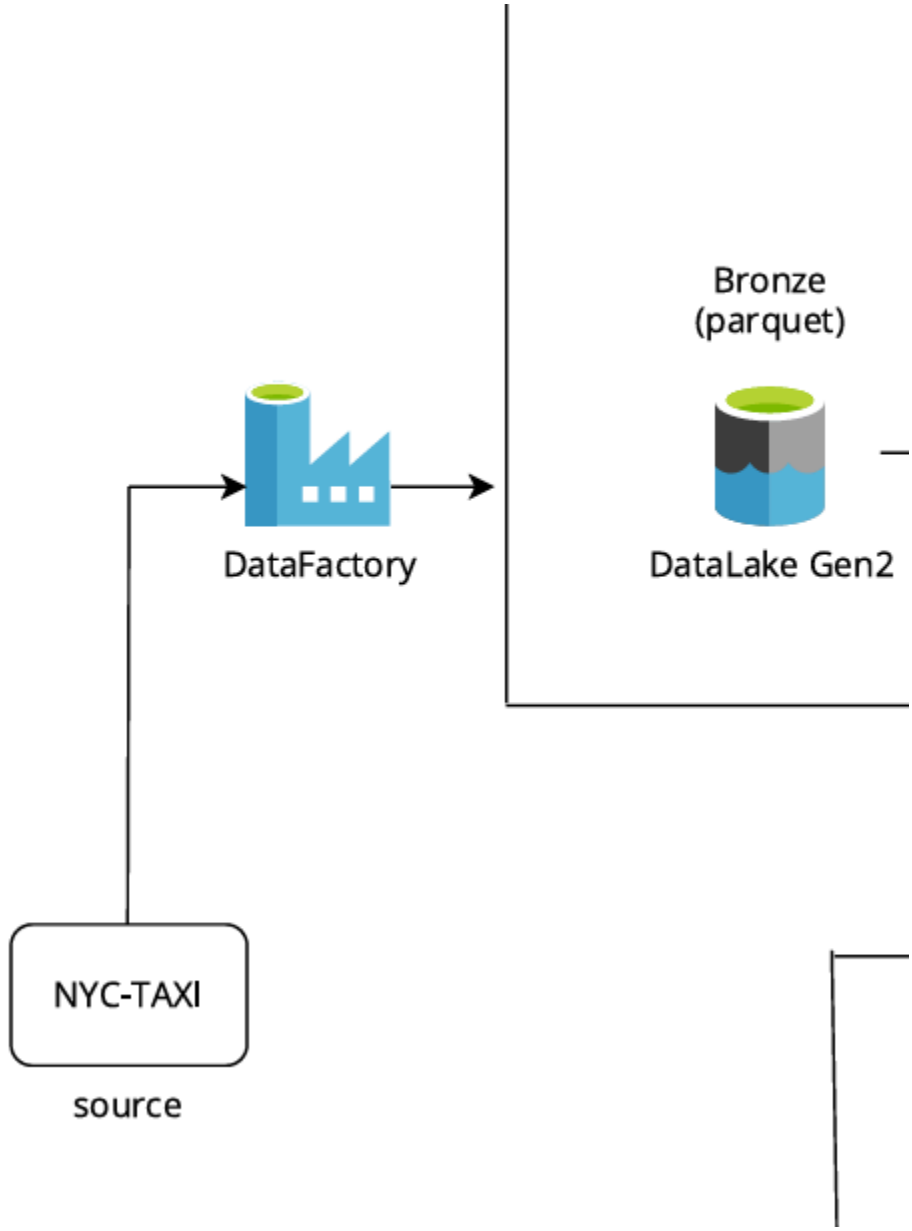
Manually copying the below raw csv files to ADLS gen2 storage account in bronze folder:

1. Trip_type.csv
2. Taxi_zone_lookup.csv

Copying the below trip_data raw parquet files from http to ADLS gen2 storage account in bronze folder using Azure data Factory:

3. https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page → copying 2023 green taxi trip records.

PHASE 1:



In phase 1, we have created a pipeline to copy the data from https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page to ADLS gen2 storage account using Azure Data Factory.

Base URL: https://d37ci6vzurychx.cloudfront.net

Relative URL:/trip-data/green_tripdata_2023-01.parquet

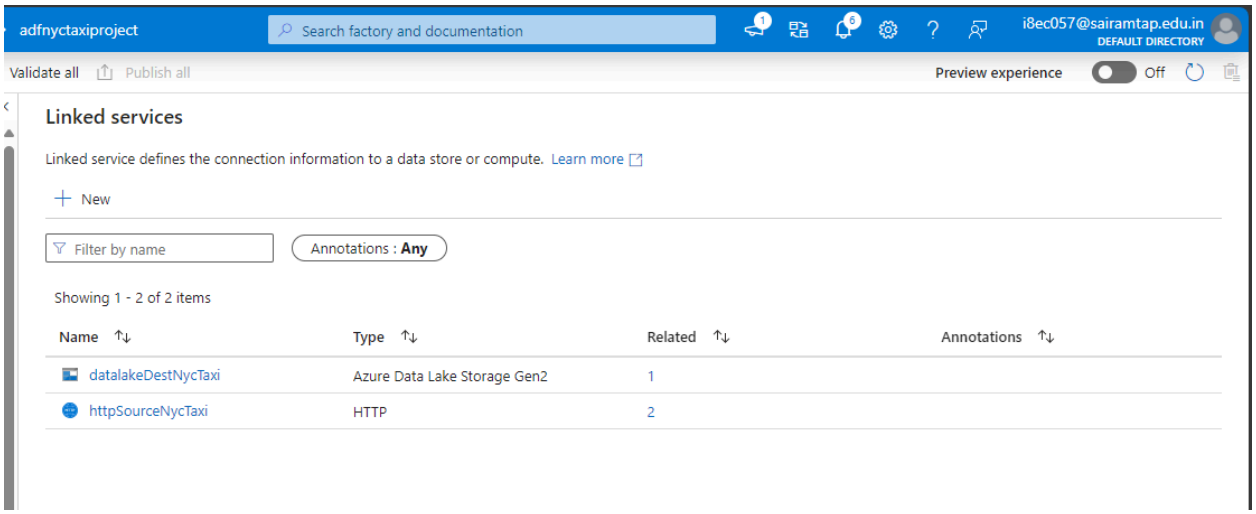
Here, we need to copy the 2023 tripdata from January to December. So, the Base URL will be

the same and the relative url will change.

Relative URL for january:/trip-data/green_tripdata_2023-01.parquet
Relative URL for february:/trip-data/green_tripdata_2023-02.parquet
Relative URL for march:/trip-data/green_tripdata_2023-03.parquet
Relative URL for april:/trip-data/green_tripdata_2023-04.parquet
Relative URL for may:/trip-data/green_tripdata_2023-05.parquet
Relative URL for june:/trip-data/green_tripdata_2023-06.parquet
Relative URL for july:/trip-data/green_tripdata_2023-07.parquet
Relative URL for august:/trip-data/green_tripdata_2023-08.parquet
Relative URL for september:/trip-data/green_tripdata_2023-09.parquet
Relative URL for october:/trip-data/green_tripdata_2023-10.parquet
Relative URL for november:/trip-data/green_tripdata_2023-11.parquet
Relative URL for december:/trip-data/green_tripdata_2023-12.parquet

I have created a single dynamic end to end pipeline to copy all these parquet files from the API to ADLS gen2 storage account.

1. Created a Linked service for http(source) and ADLS gen2(destination).

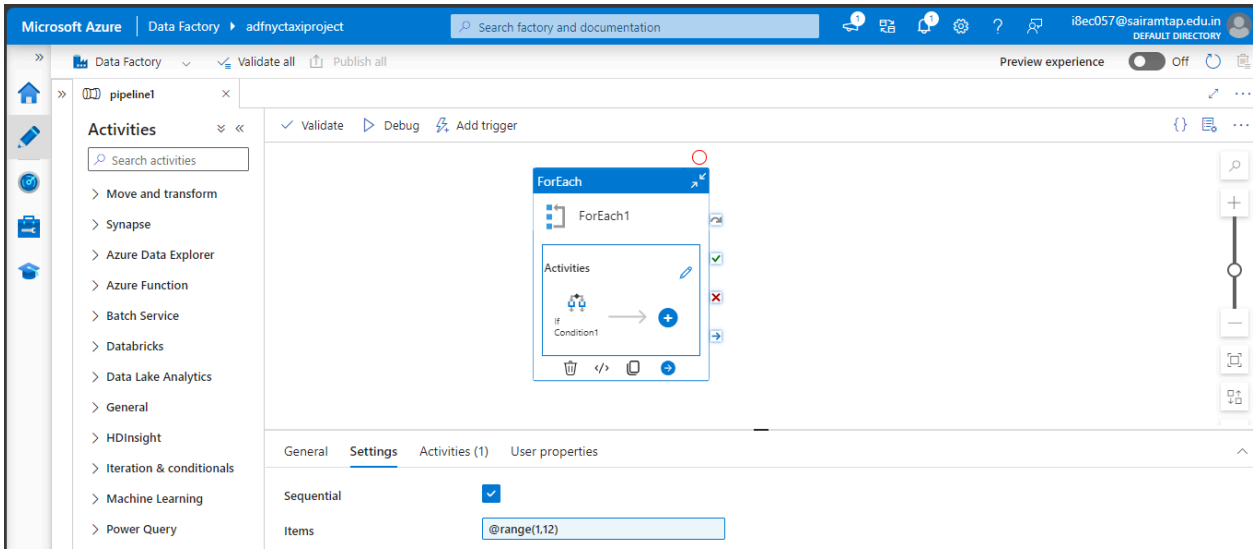


Configured the base url in http(source) linked service.

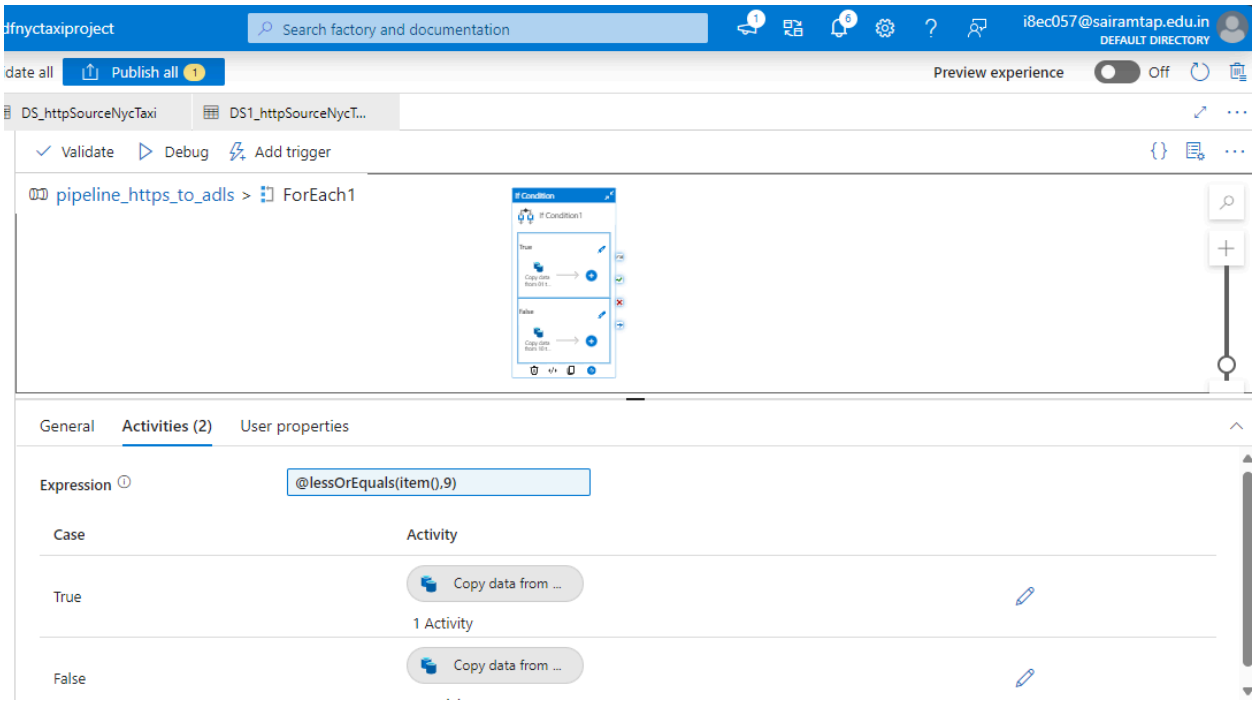
2. Created Dynamic pipeline.

Here Relative URL: /trip-data/green_tripdata_2023-**01**.parquet is changing from 01 to 12.

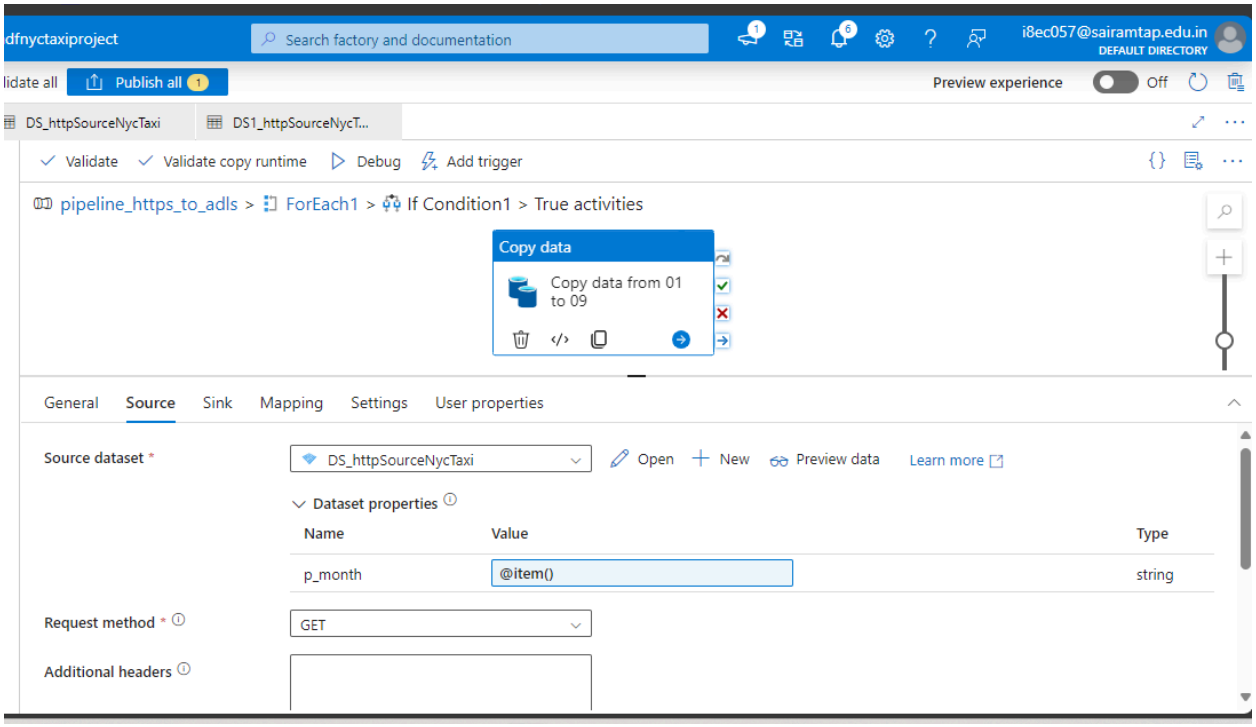
For that, I have created on ForEach activity with the range from 1 to 12.



Inside ForEach activity, I have created a If condition activity to check if the output from the ForEach activity is less than or equal to 9.



If it is true, then Copy activity will be executed from 1 to 9.



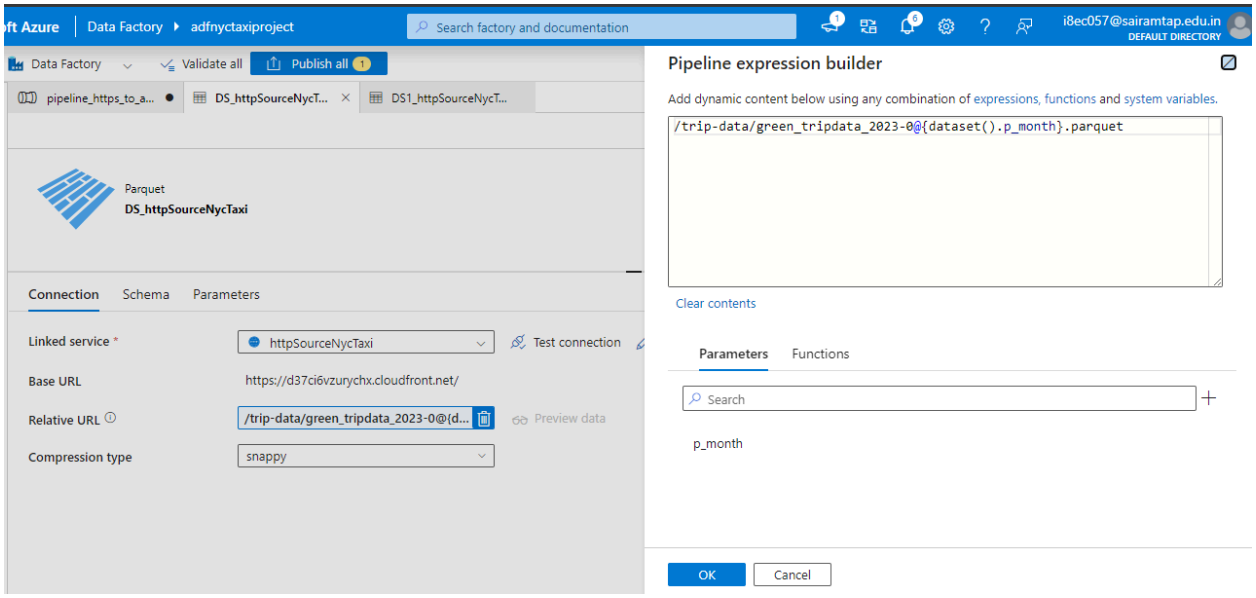
Source Dataset 1:

In source dataset 1, I have created a parameter called **p_month**, the output of the ForEach activity will be assigned to p_month.

http://trip-data/green_tripdata_2023-0**@{dataset().p_month}**.parquet

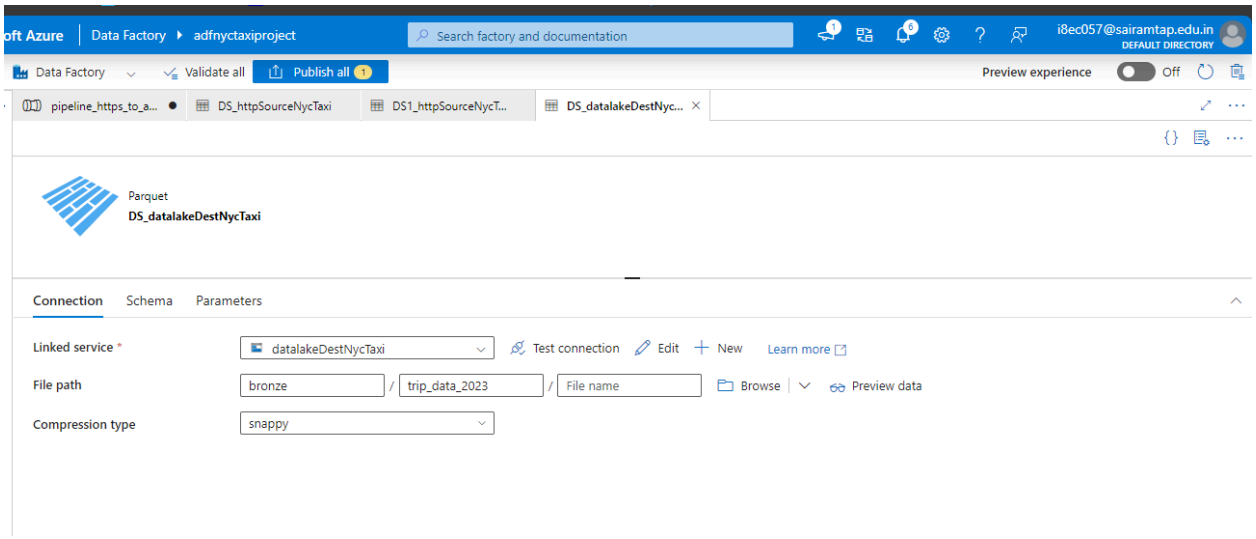
Relative URL: /trip-data/green_tripdata_2023-0**1**.parquet

Here the 1 will be replace by the output of the ForEach activity(p_month).

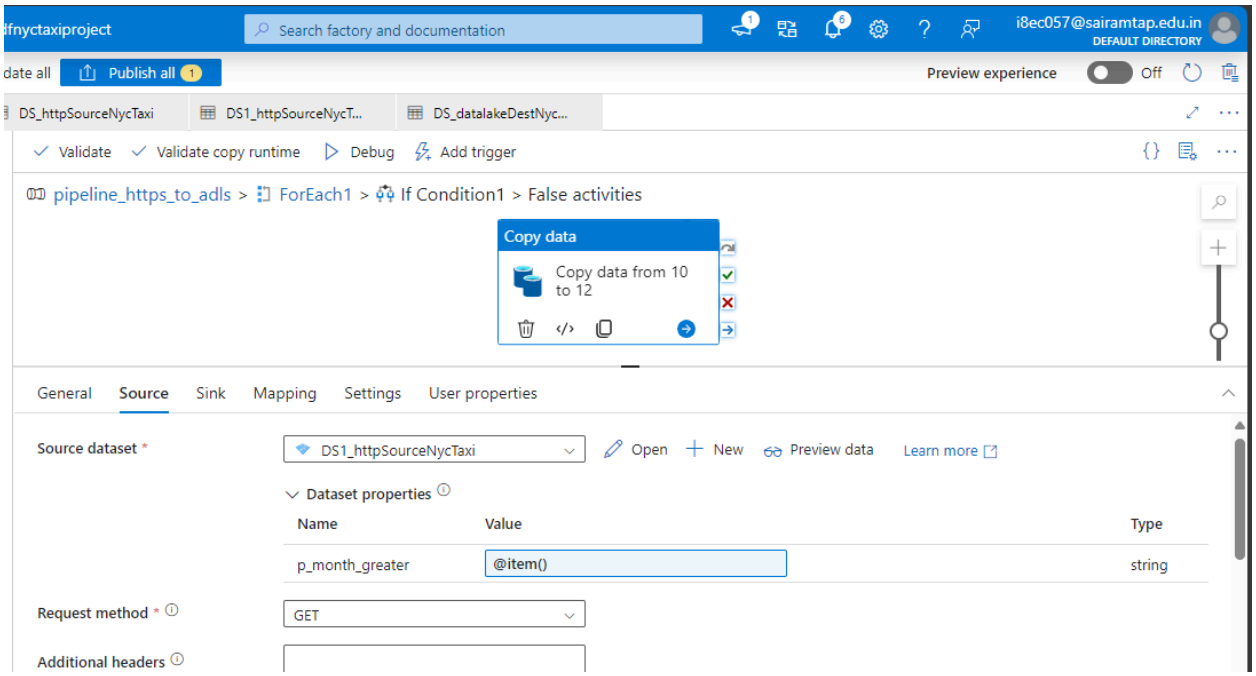


Sink Dataset:

The parquet files from API will then stored in the bronze container.



If it is false, i.e. If the output from the ForEach activity greater than 9 then Copy activity will be executed from 10 to 12.



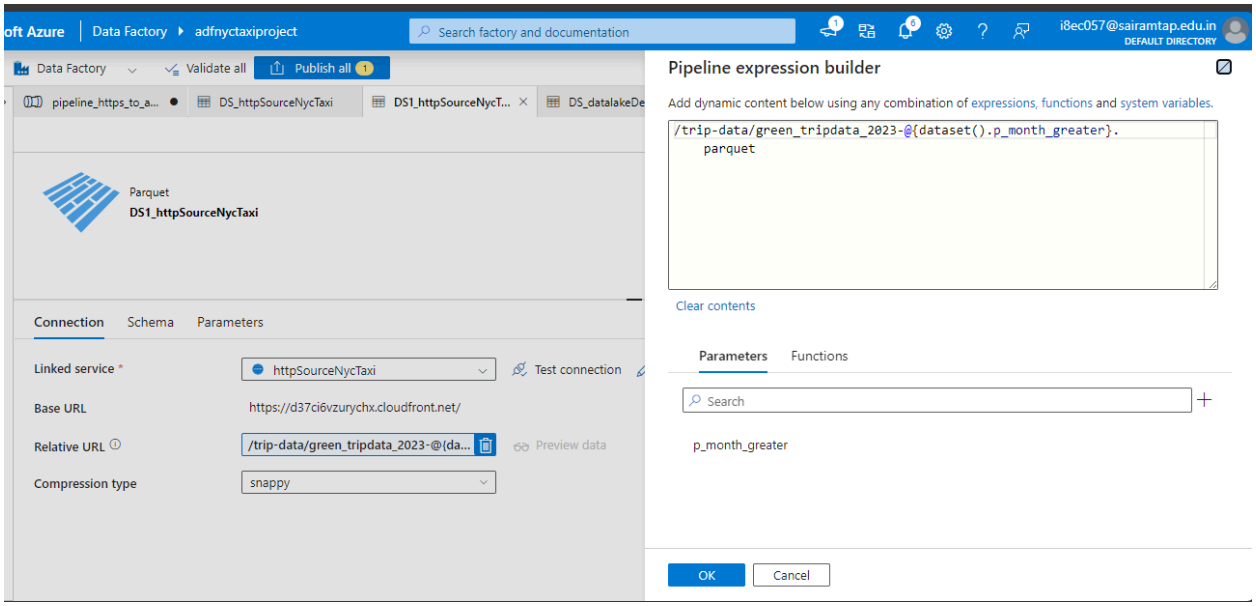
Source Dataset 2:

In source dataset 2, I have created a parameter called **p_month**, the output of the ForEach activity will be assigned to p_month.

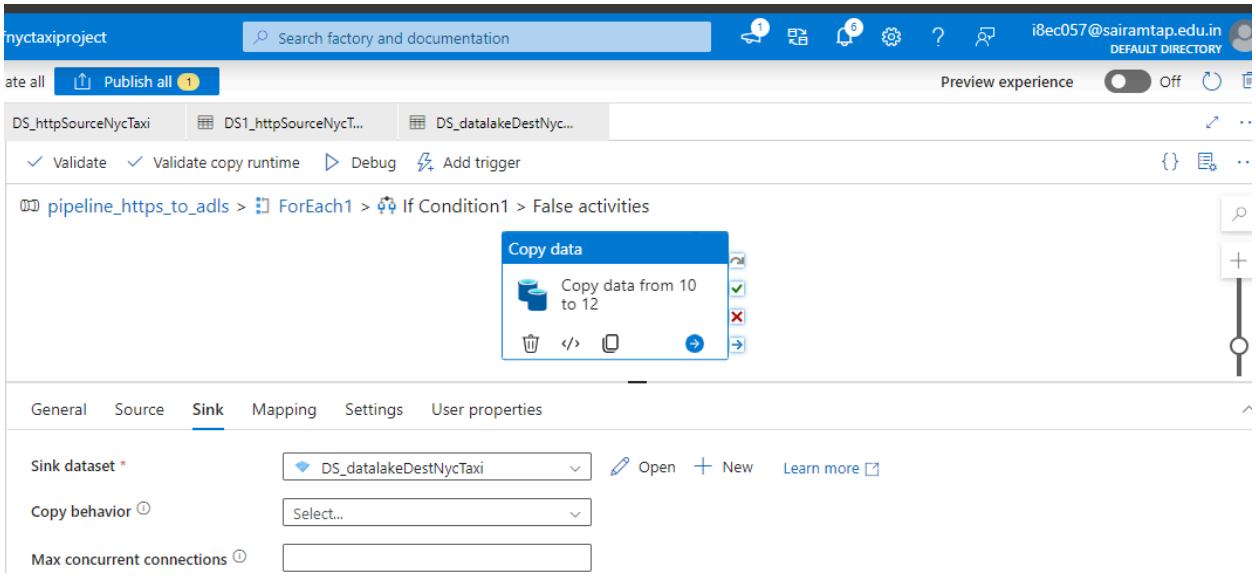
http://trip-data/green_tripdata_2023-**@{dataset().p_month}**.parquet

Relative URL: /trip-data/green_tripdata_2023-**10**.parquet

Here the 10 will be replace by the output of the ForEach activity(p_month).

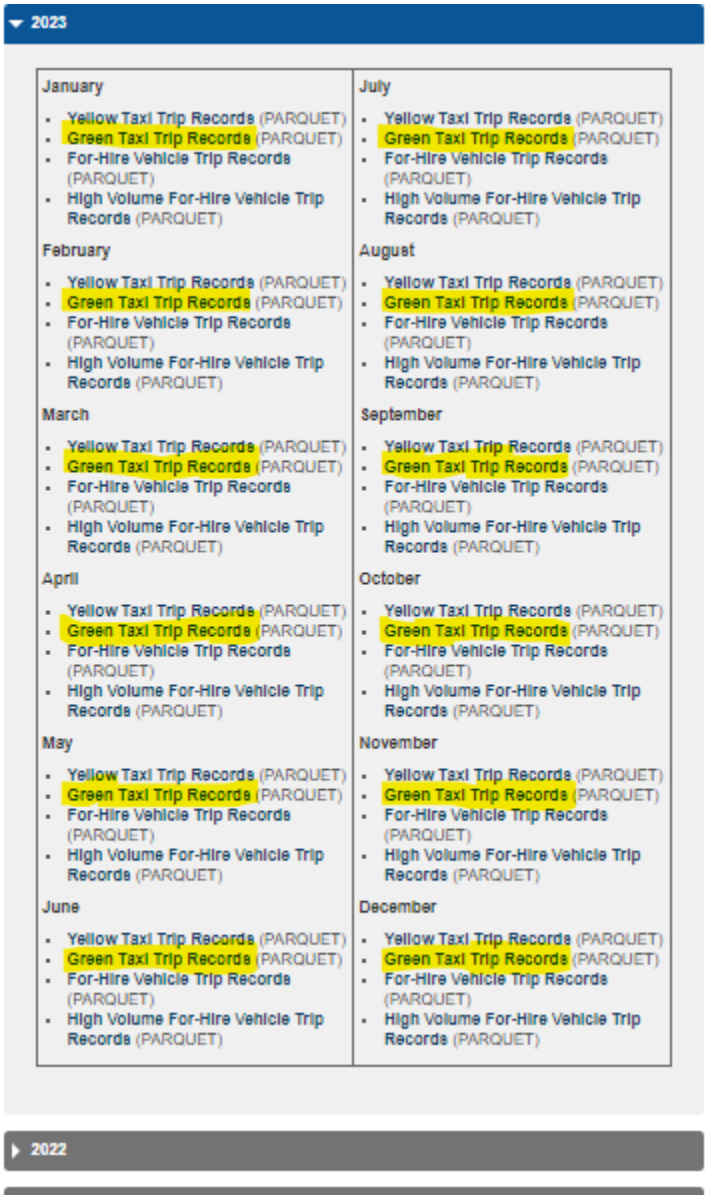


Using the same sink dataset to store the parquet files from API to bronze container.

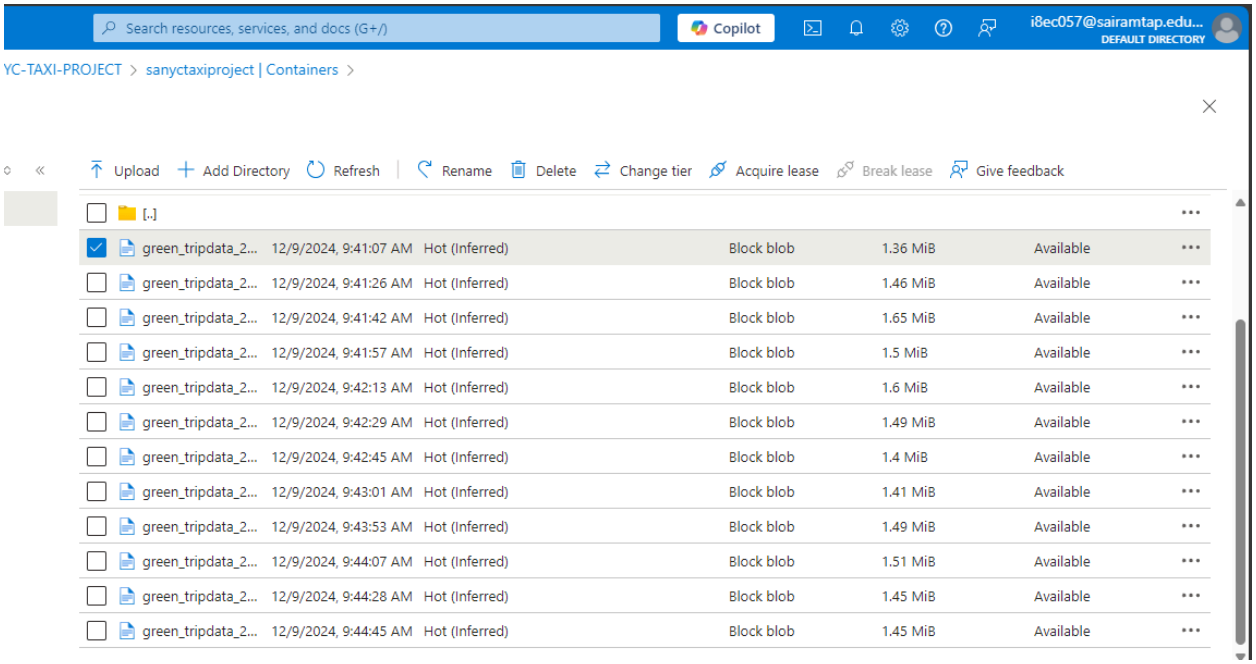


The above pipeline will copy the below parquet files from API to bronze container in ADLS gen2.

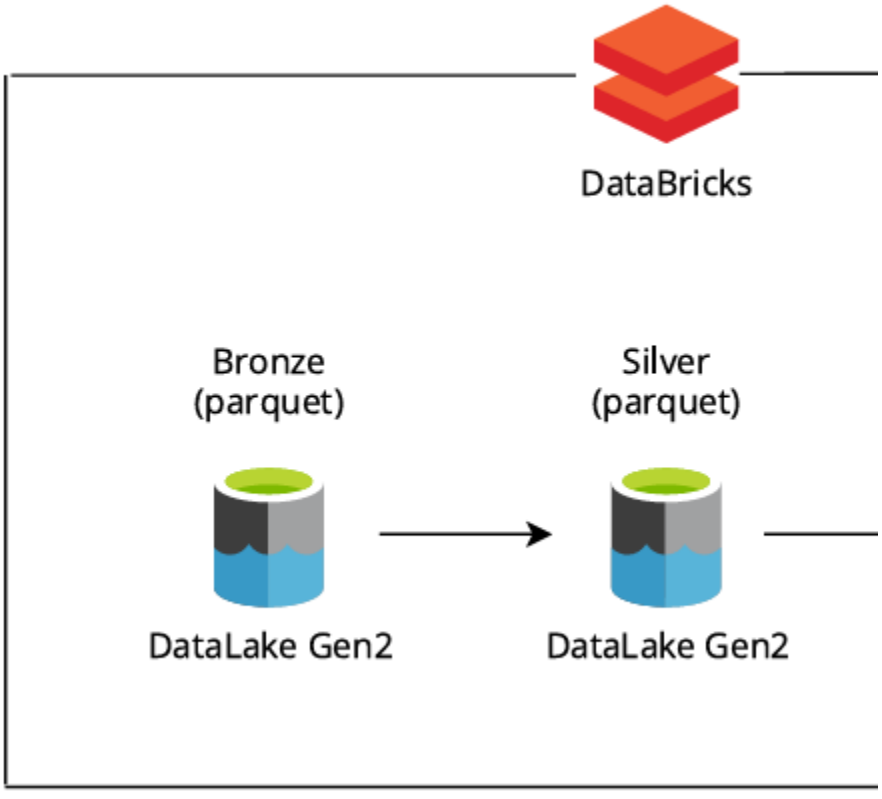
From API:



To bronze container:


Search resources, services, and docs (G+/)
Copilot
i8ec057@sairamtap.edu...
DEFAULT DIRECTORY
YC-TAXI-PROJECT > sanyctaxiproject | Containers >
Upload
Add Directory
Refresh
Rename
Delete
Change tier
Acquire lease
Break lease
Give feedback
[..]
green_tripdata_2... 12/9/2024, 9:41:07 AM Hot (Inferred) Block blob 1.36 MiB Available
green_tripdata_2... 12/9/2024, 9:41:26 AM Hot (Inferred) Block blob 1.46 MiB Available
green_tripdata_2... 12/9/2024, 9:41:42 AM Hot (Inferred) Block blob 1.65 MiB Available
green_tripdata_2... 12/9/2024, 9:41:57 AM Hot (Inferred) Block blob 1.5 MiB Available
green_tripdata_2... 12/9/2024, 9:42:13 AM Hot (Inferred) Block blob 1.6 MiB Available
green_tripdata_2... 12/9/2024, 9:42:29 AM Hot (Inferred) Block blob 1.49 MiB Available
green_tripdata_2... 12/9/2024, 9:42:45 AM Hot (Inferred) Block blob 1.4 MiB Available
green_tripdata_2... 12/9/2024, 9:43:01 AM Hot (Inferred) Block blob 1.41 MiB Available
green_tripdata_2... 12/9/2024, 9:43:53 AM Hot (Inferred) Block blob 1.49 MiB Available
green_tripdata_2... 12/9/2024, 9:44:07 AM Hot (Inferred) Block blob 1.51 MiB Available
green_tripdata_2... 12/9/2024, 9:44:28 AM Hot (Inferred) Block blob 1.45 MiB Available
green_tripdata_2... 12/9/2024, 9:44:45 AM Hot (Inferred) Block blob 1.45 MiB Available

PHASE 2:


DataBricks
Bronze
(parquet)
Silver
(parquet)
DataLake Gen2
DataLake Gen2

Here in phase 2, I have transformed the data in the bronze container using Databricks and moved the transformed data to silver container.

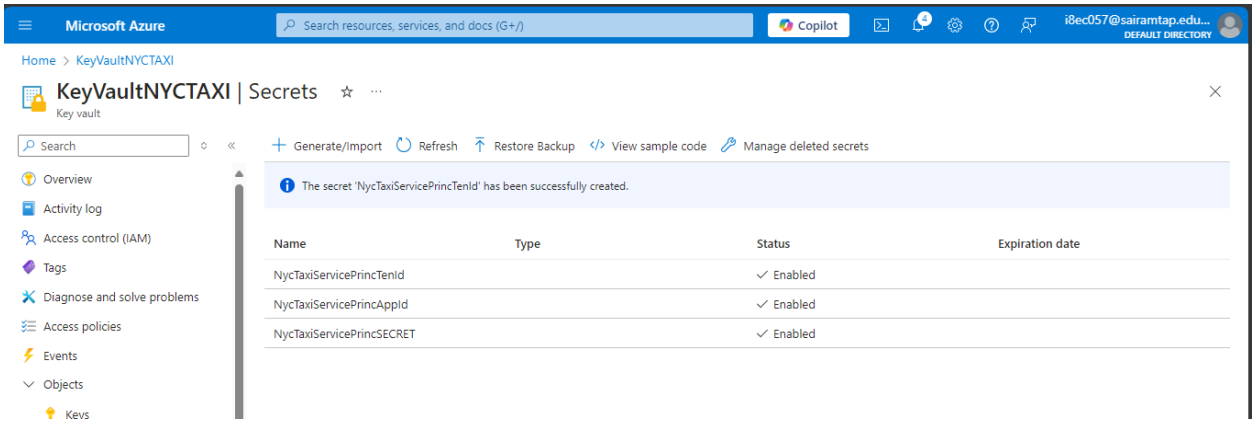
Databricks should have the access to read the files from bronze container (adls gen2 storage account), so I have created a Service principal in Microsoft entra ID to provide the access for the ADLS gen2 storage account to Databricks.

I have assigned the blob contributor role to the Service principal in the ADLS gen2 storage account.

To connect the ADLS gen2 to Databricks, we need the below details:

Application_id of ServicePrincipal
Directory_id of ServicePrincipal
Secret of ServicePrincipal

In order to prevent directly using the above details in databricks notebook, I have created a Azure Key Vault to store the above details securely.



In order to use these secrets in the databricks notebook. First we need to create a azure key vault backed secret scope in databricks.

To create a azure key vault backed secret scope in databricks:

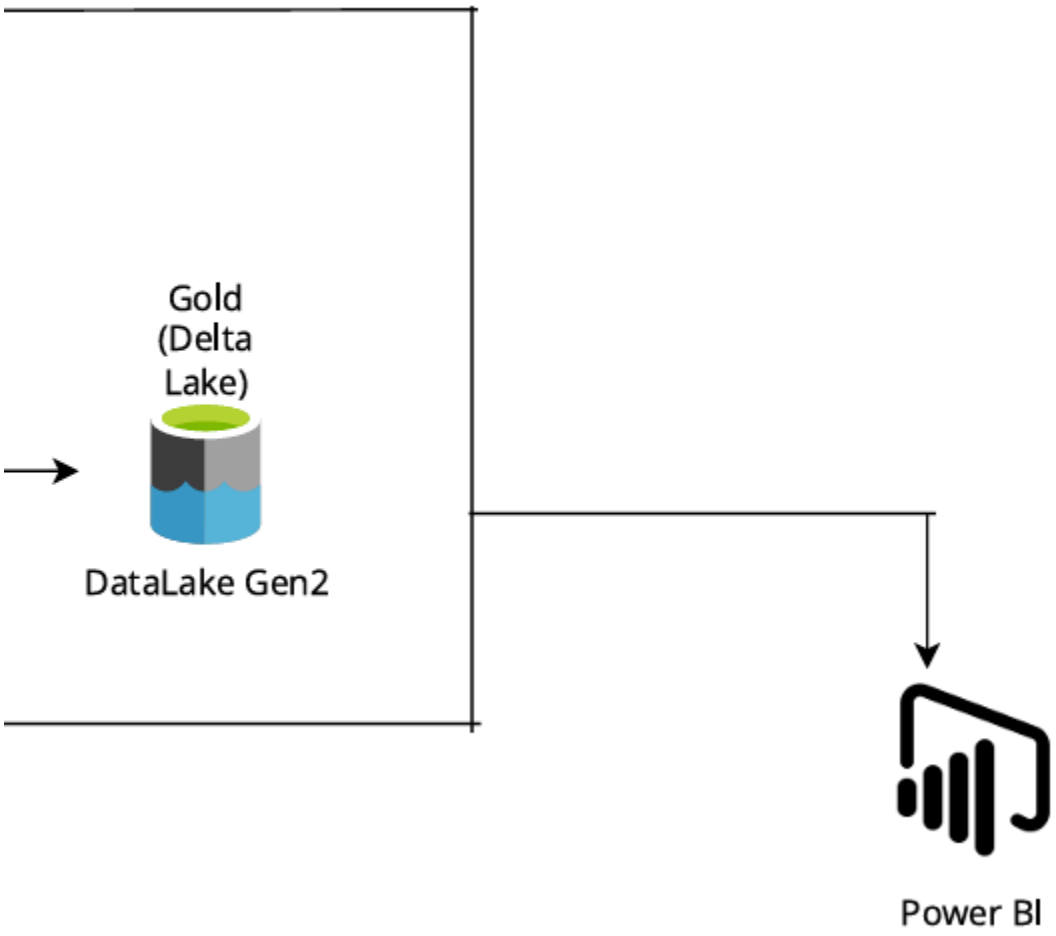
Add the **secrets/createScope** at the end of the databricks instance url.

Go to https://<databricks-instance>**#secrets/createScope**

REFER THE **NYC_TAXI_SILVER** notebook:

Now the data are transformed and stored in the Silver container:

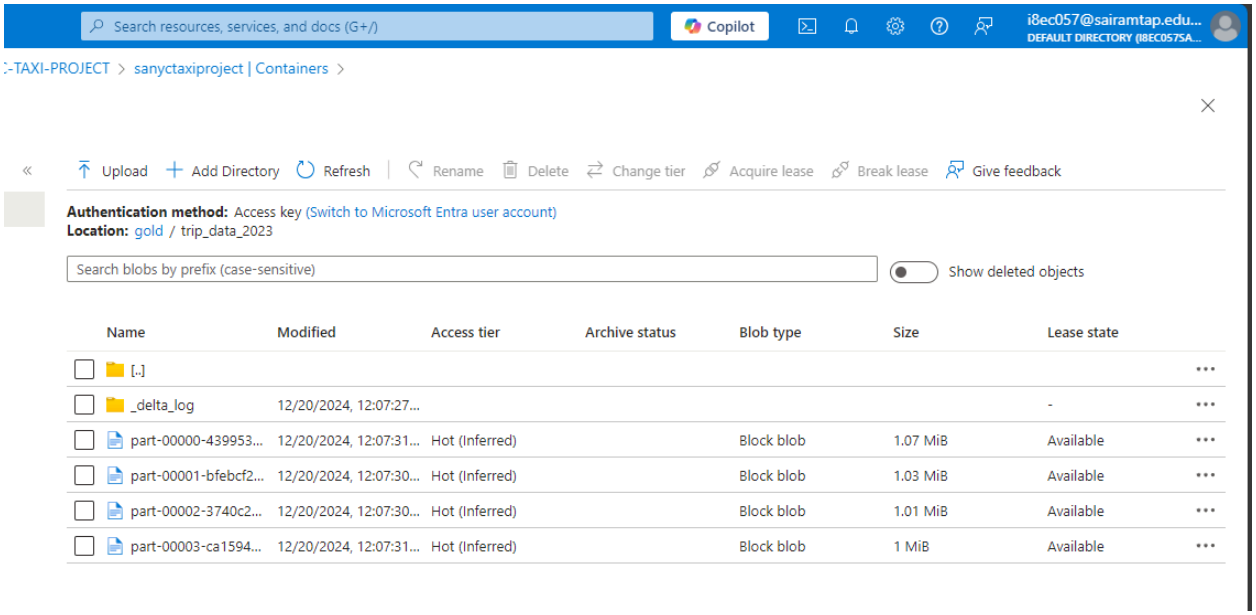
PHASE 3:



Copying the data to the Delta Lake and connect the final data with the Power Bi.

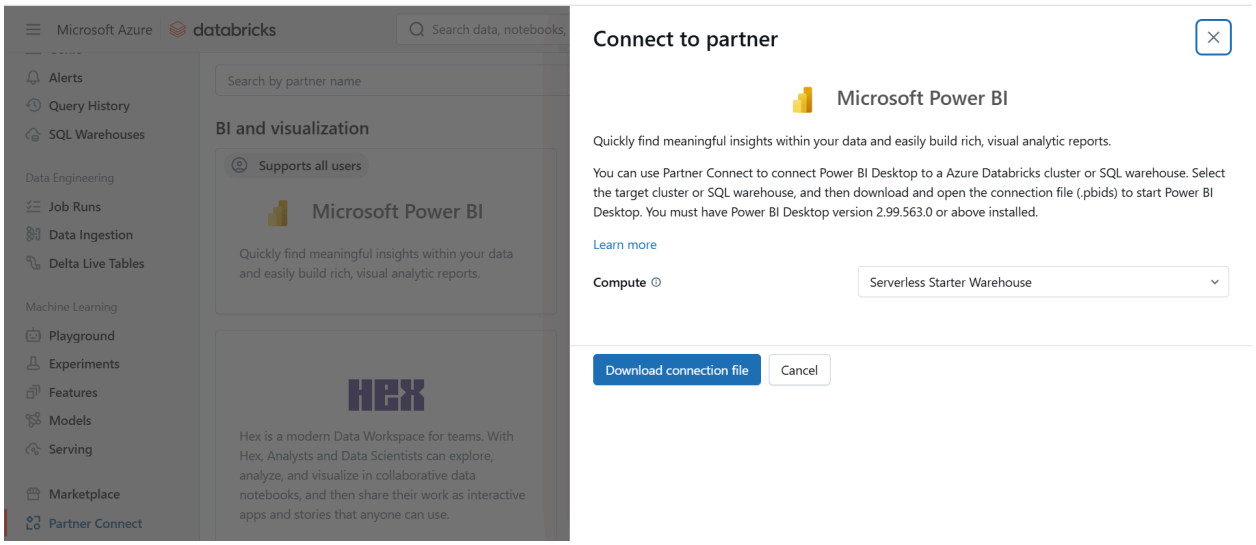
REFER THE **NYC_TAXI_GOLD** notebook:

Now the data are moved to the Gold container in Delta Format.

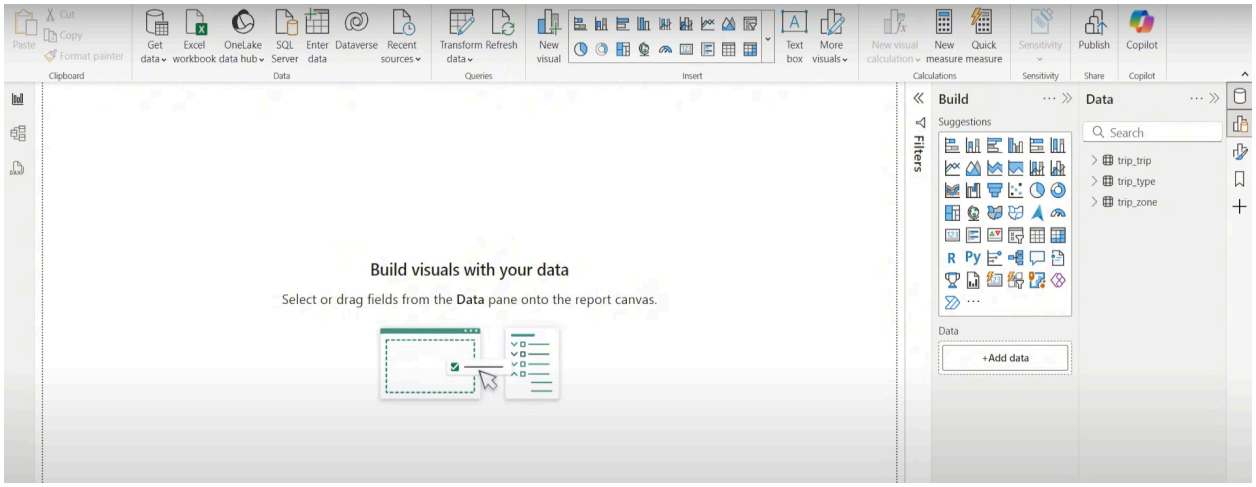


To connect with Power BI, I have installed power BI desktop on my machine.

In Databricks, go to Partner connect and select Power BI desktop, then download the connection file. It will create a empty power BI file which has all the connection connection strings.



We need to create a databricks token and provide the token in the Power BI, after that we can see the gold container data's in power BI.


Paste
Cut
Copy
Format painter
Clipboard
Get data
Excel workbook
OneLake data hub
SQL Server
Enter data
Dataverse
Recent sources
Data
Transform data
Refresh
Queries
New visual
Text box
More visuals
Insert
New visual calculation
New measure
Quick measure
Calculations
Sensitivity
Publish
Share
Copilot
Build visuals with your data
Select or drag fields from the Data pane onto the report canvas.
Filters
Build
Suggestions
Data
+Add data
Data
Search
trip_trip
trip_type
trip_zone