# Azure End-To-End Data Engineering Project

## Project Overview

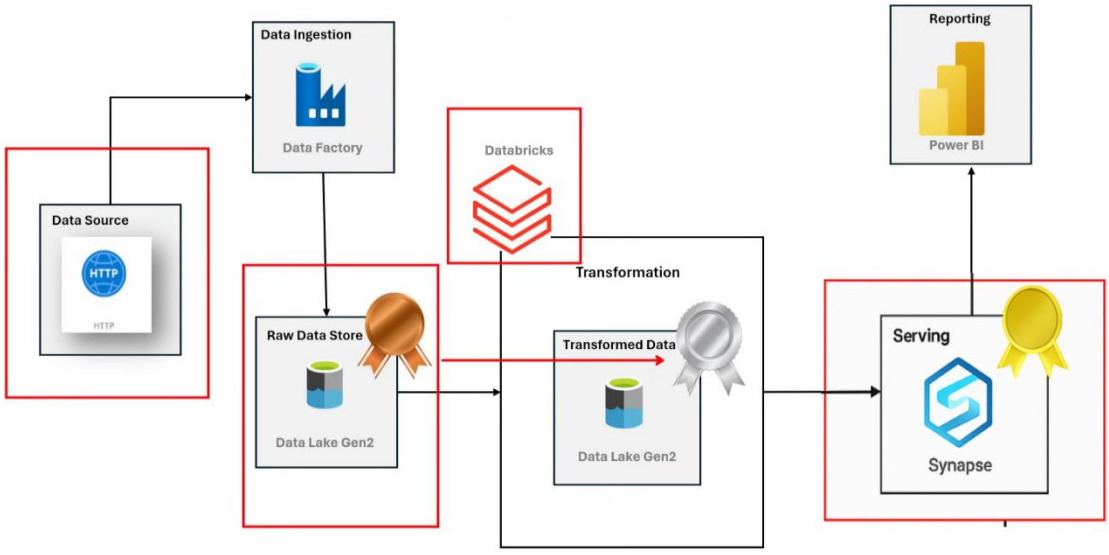
This project demonstrates a comprehensive end-to-end data engineering solution using Microsoft Azure. The goal was to build a scalable data pipeline that ingests, transforms, and serves data for analytics. The project leverages multiple Azure services, including:

- Azure Data Lake Storage (ADLS Gen2) – For storing raw, processed, and serving data.
- Azure Data Factory (ADF) – For orchestrating data movement and transformations.
- Azure Databricks – For performing advanced data transformations using Spark.
- Azure Synapse Analytics – For data warehousing and serving structured data.
- Power BI – For visualization and reporting.

The project follows the Medallion Architecture (Bronze → Silver → Gold) to ensure data quality and reliability.

## Project Architecture

The architecture consists of the following stages:




1. **Data Ingestion (Bronze Layer)**
    - Source: GitHub API (AdventureWorks dataset).

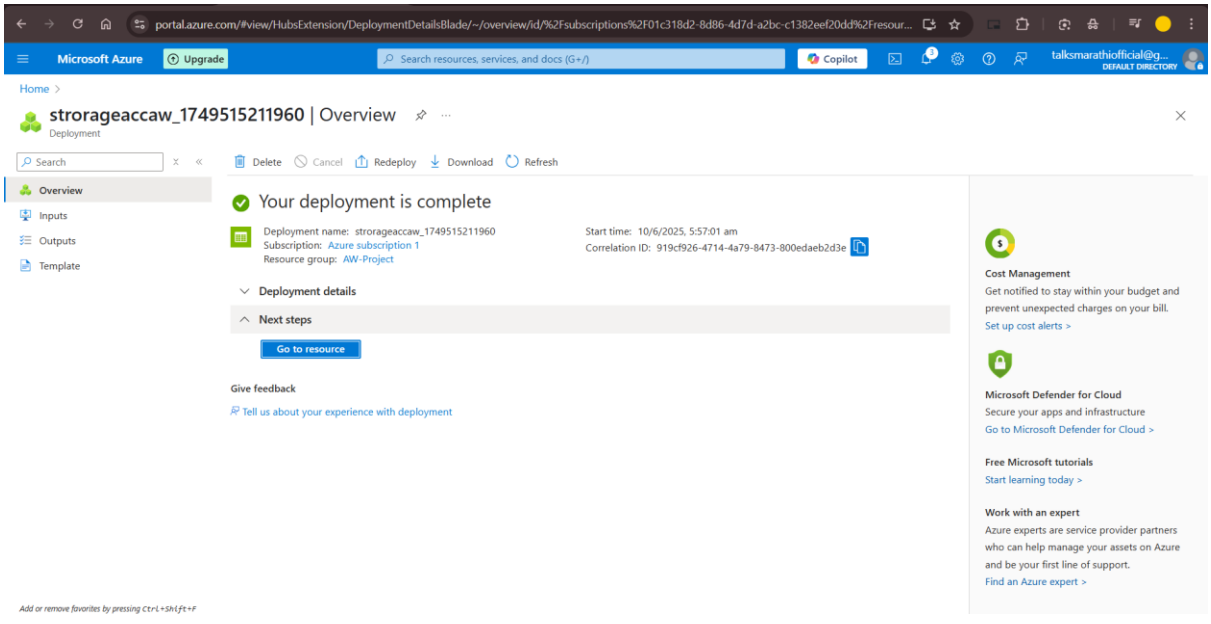
- Tool: Azure Data Factory (ADF) for dynamic data ingestion.
- Storage: Azure Data Lake (Bronze container).

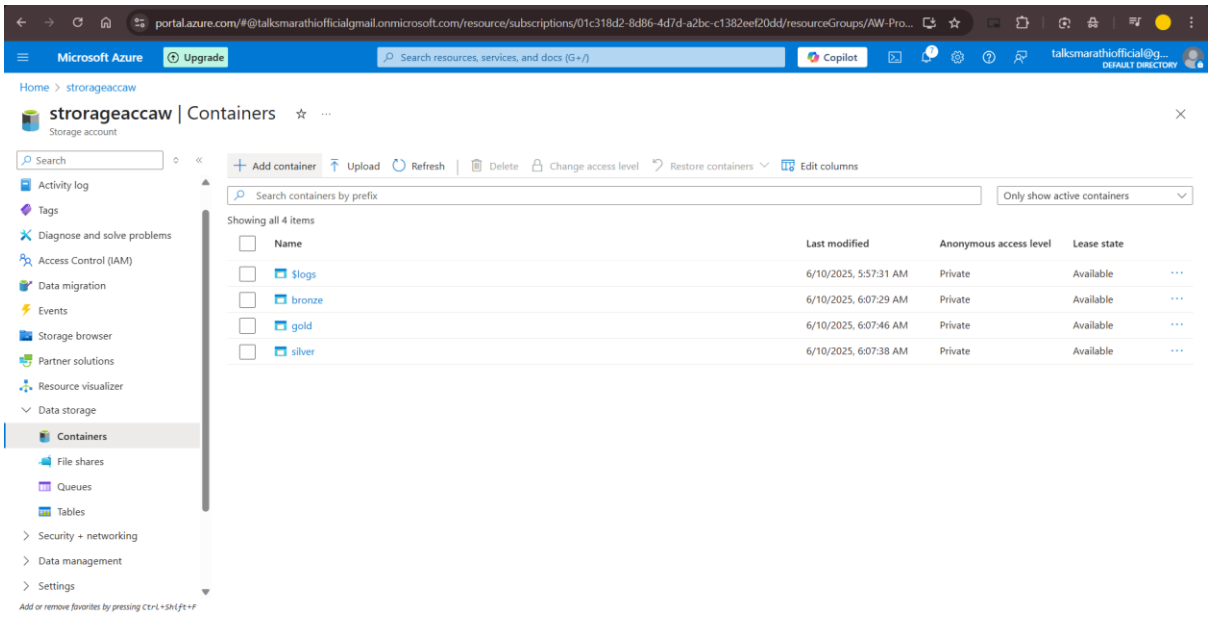
2. **Data Transformation (Silver Layer)**
    - Tool: Azure Databricks for Spark-based transformations.
    - Storage: Azure Data Lake (Silver container).
3. **Data Serving (Gold Layer)**
    - Tool: Azure Synapse Analytics for SQL-based data modeling.
    - Storage: Azure Data Lake (Gold container).
    - Visualization: Power BI for reporting.

## Detailed Implementation

### 1. Setting Up Azure Resources

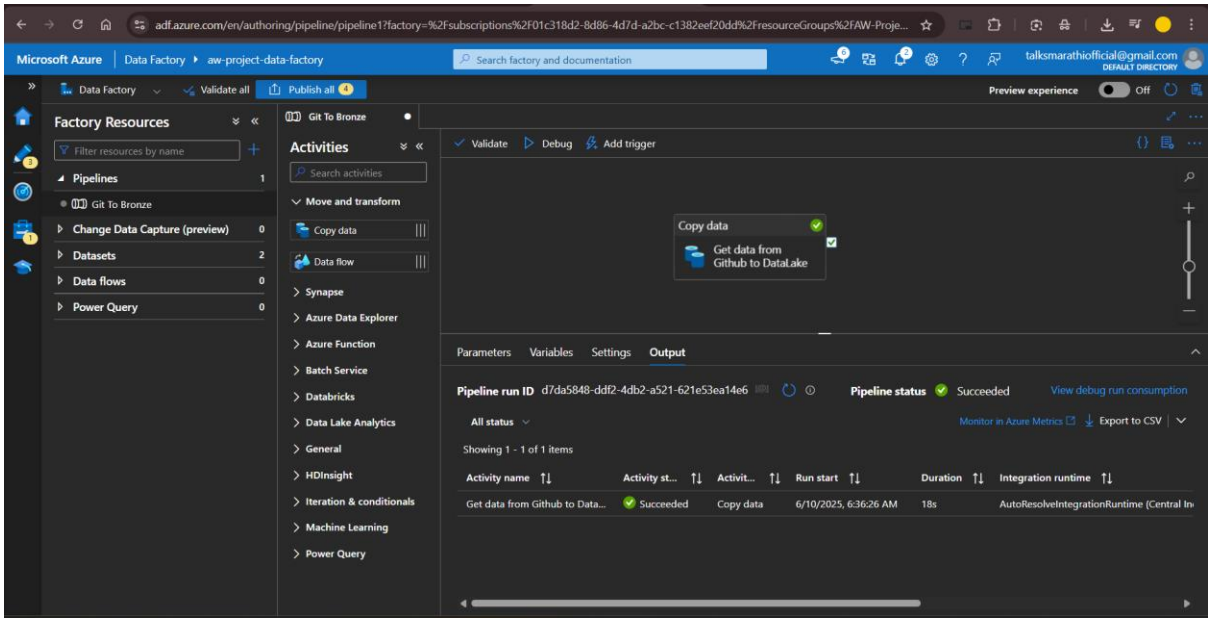
- **Resource Group:** Created to organize all Azure services.
- **Azure Data Lake Storage (ADLS Gen2)**
    - Enabled hierarchical namespace for folder structure.
    - **Created three containers:**
        - Bronze (raw data)
        - Silver (cleaned & transformed data)
        - Gold (serving layer for analytics)



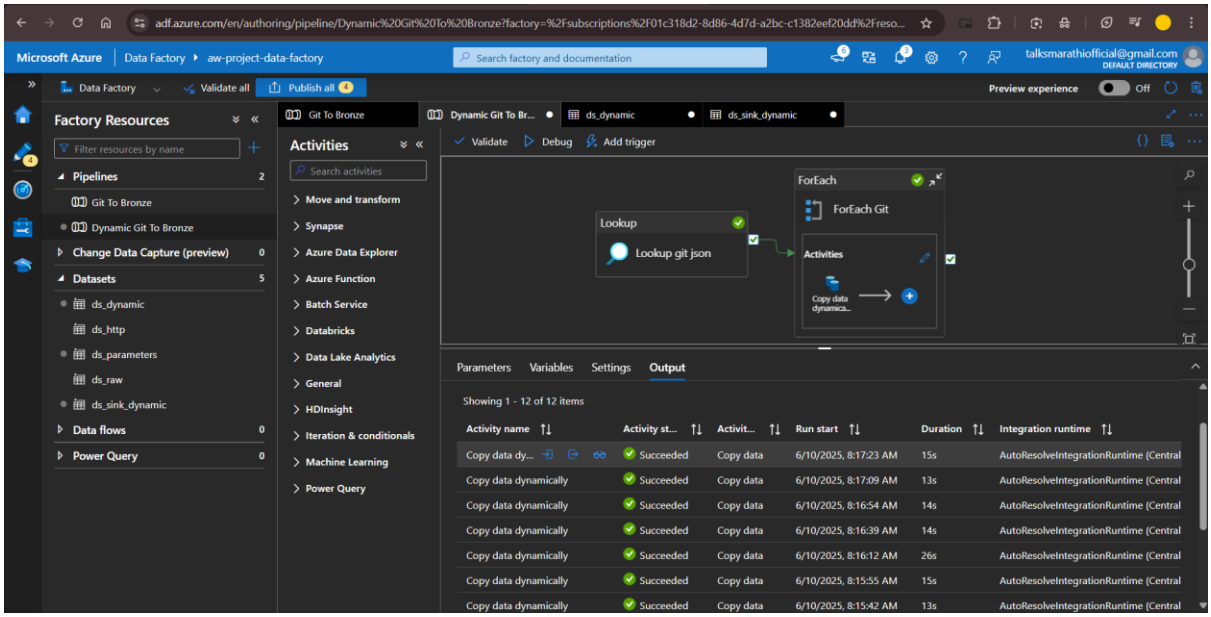


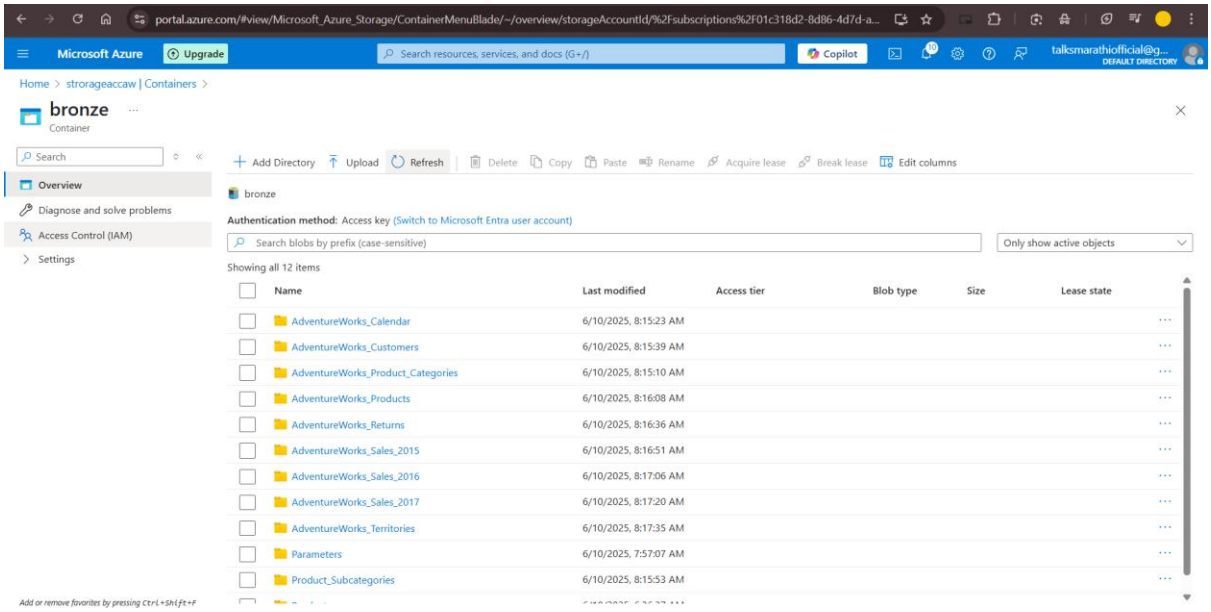

## 2. Data Ingestion with Azure Data Factory (ADF)

- **Static Pipeline:**
    - Pulled data from GitHub (CSV files) into the Bronze layer.



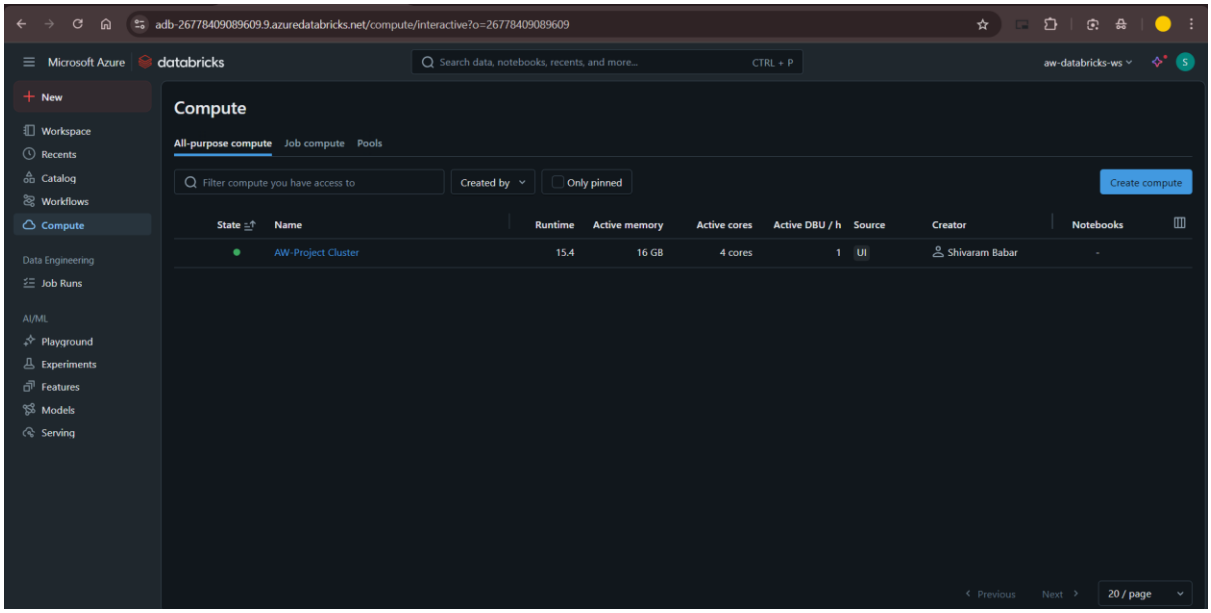

- **Dynamic Pipeline:**
    - Used parameters and loops to automate ingestion.
    - Implemented JSON configuration for flexible file handling.




## 3. Data Transformation with Azure Databricks

- **Cluster Setup**: Configured a Databricks cluster for Spark processing.

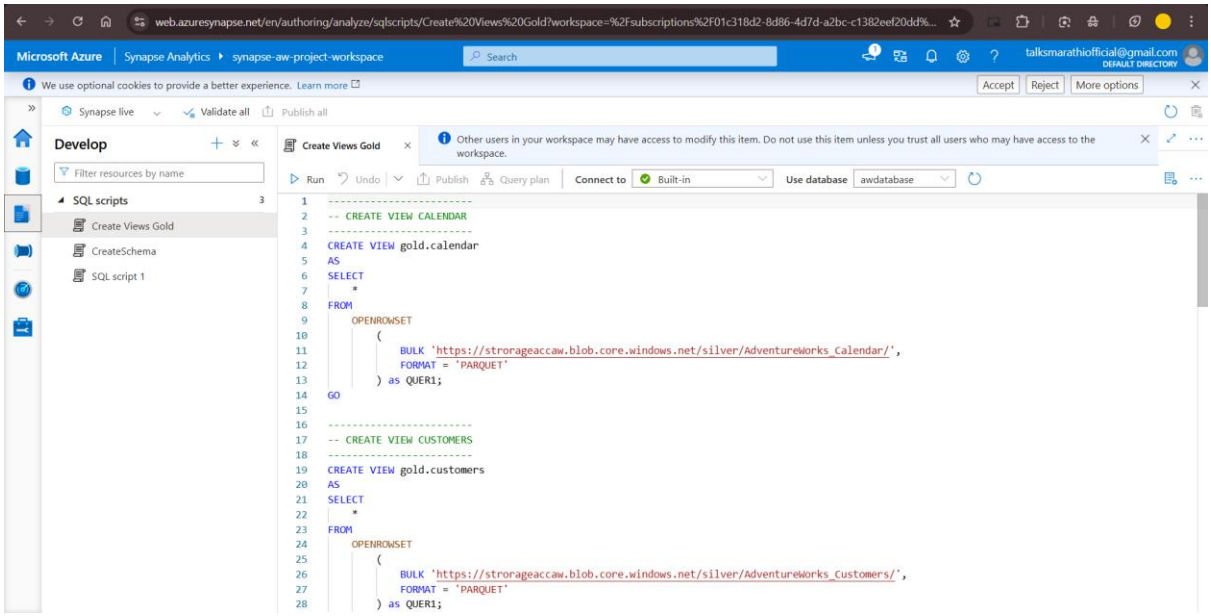



- **Data Loading**: Read data from the Bronze layer into Spark DataFrames.
- **Transformations Applied**:
    - **Date Functions**: Extracted month and year from dates.
    - **String Manipulation**: Concatenated columns (e.g., full customer name).
    - **Split Operations**: Separated product SKU into categories.
    - **Aggregations**: Grouped sales data by date for trend analysis.
- **Data Storage**: Saved transformed data in Parquet format to the Silver layer.

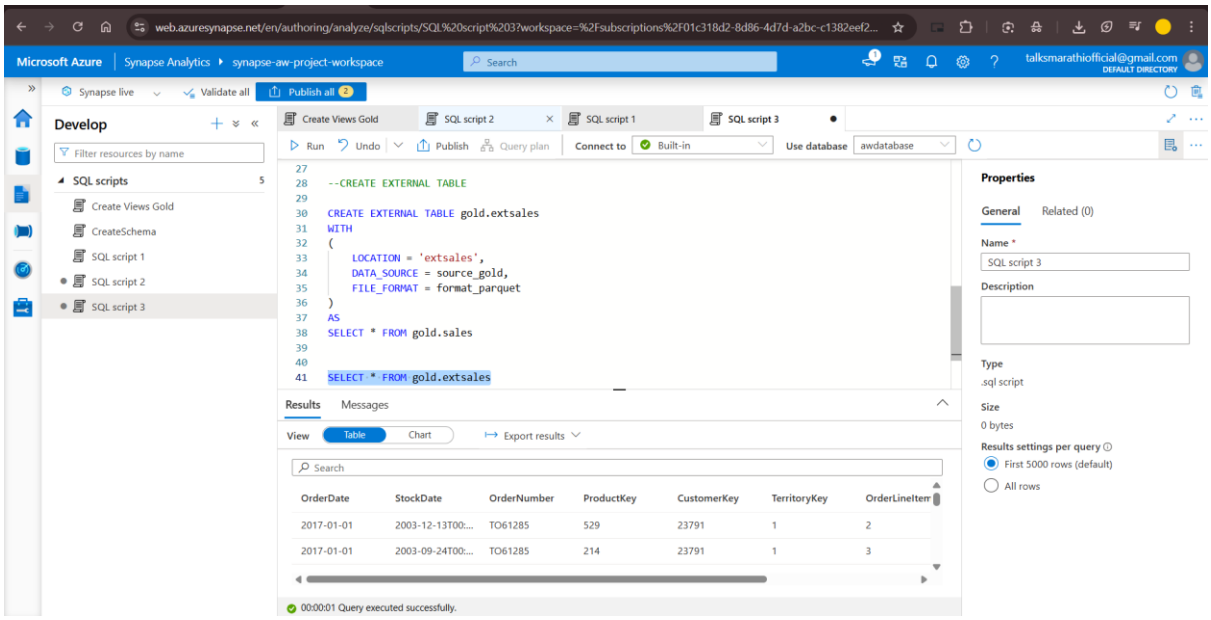
**Notebook Link :** https://github.com/shivaramsb/Adventure-Works-Data-Engineering-Project/blob/main/Silver-layer.ipynb

**4. Data Serving with Azure Synapse Analytics**

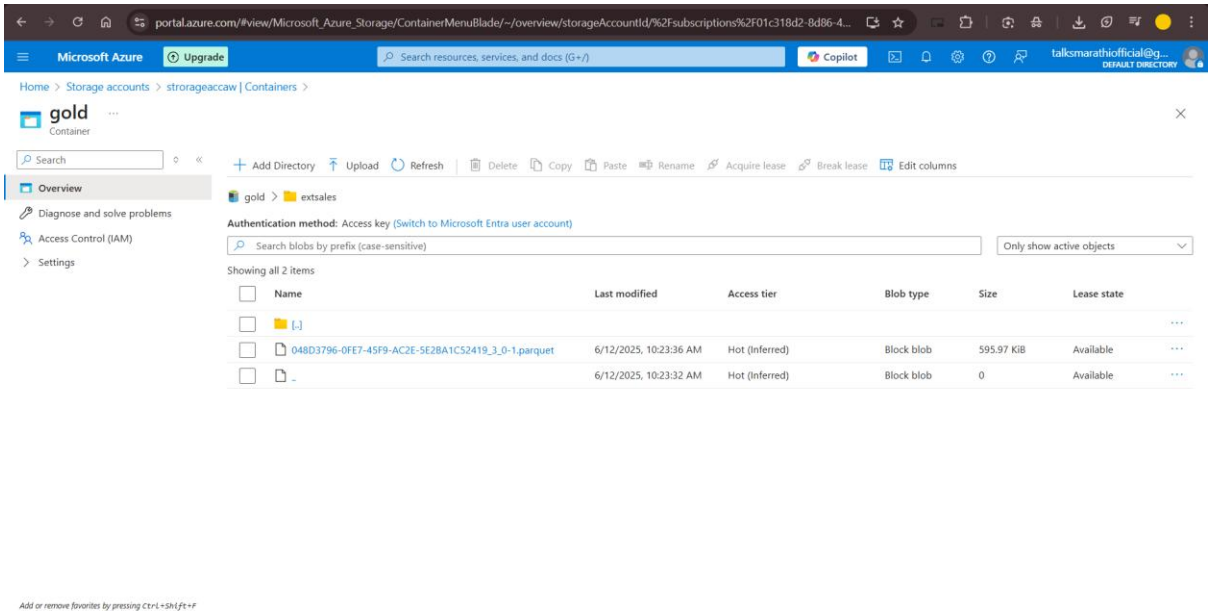
- **Serverless SQL Pool: Used for querying data directly from ADLS.**
- **Views Creation: Defined SQL views on Silver layer data.**




- **External Tables:**
  - **Created using CETAS (CREATE EXTERNAL TABLE AS SELECT).**

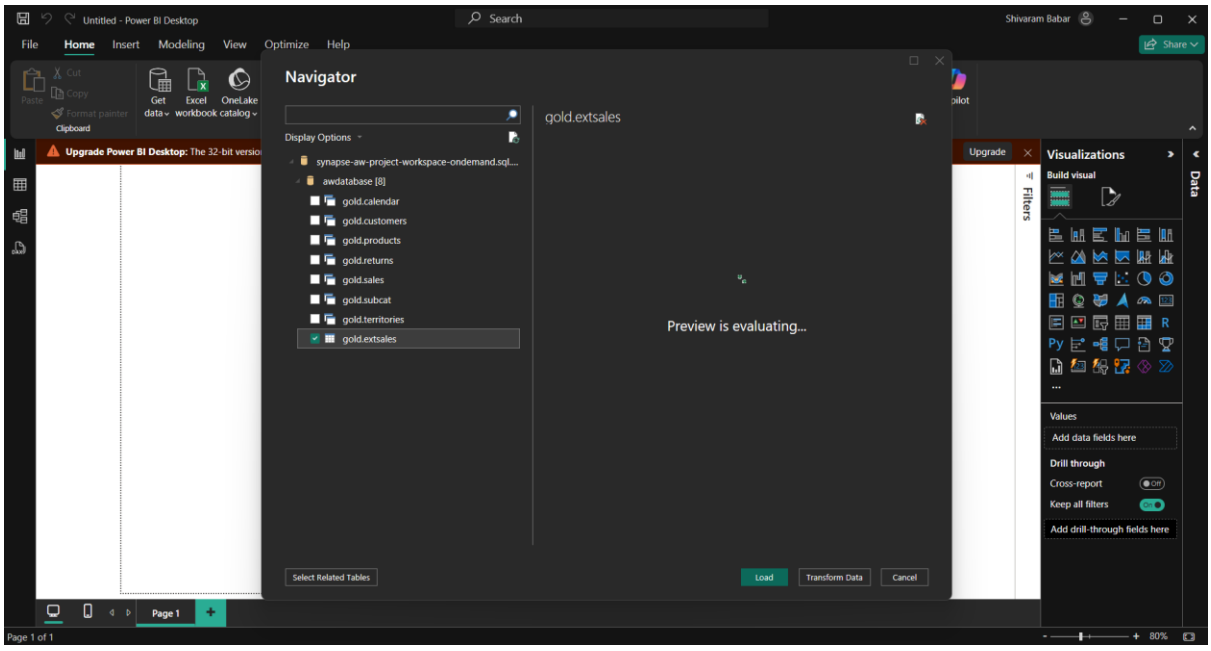



- **Stored final data in the Gold layer for Power BI consumption.**

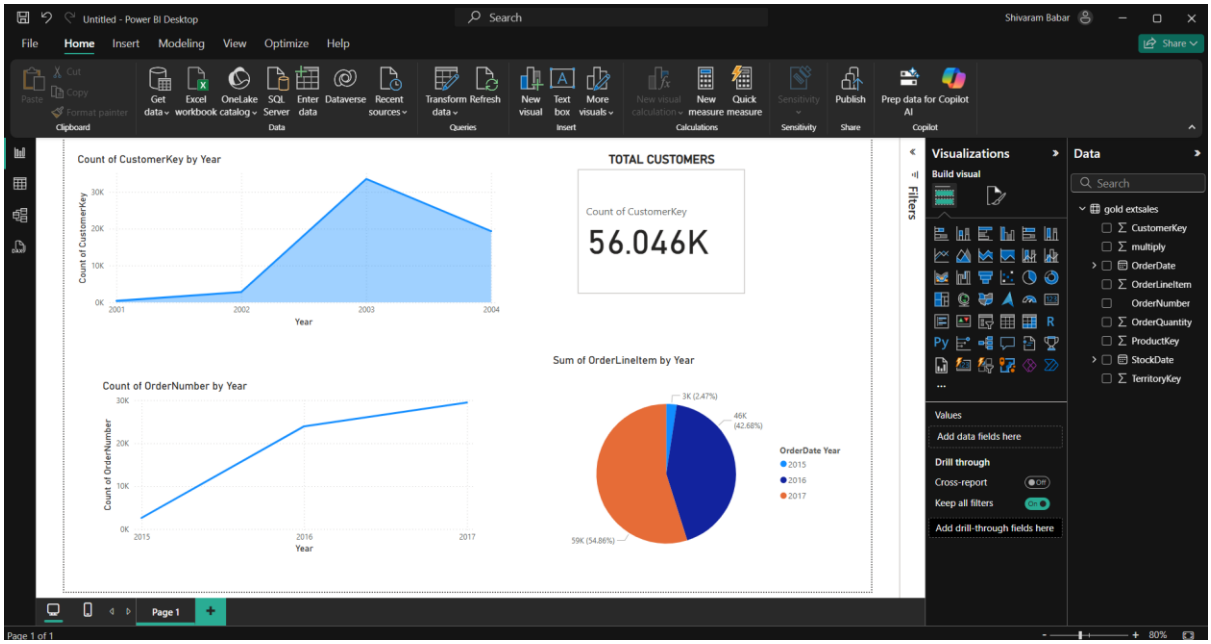



## 5. Visualization with Power BI

- **Connection Setup**: Linked Power BI to Synapse using SQL endpoint.




- **Dashboard Creation**:
    - Built interactive reports (line charts, KPIs).
    - Analyzed sales trends and customer metrics.




## Key Learnings

1. **Dynamic Pipelines in ADF**:
    - Learned to use parameters, loops, and JSON configurations for flexible data ingestion.
    - Challenges: Debugging pipeline failures due to incorrect parameter mappings.
2. **Databricks Transformations**:
    - Mastered Spark functions (withColumn, groupBy, split).
    - Challenges: Schema inference issues with CSV files.
3. **Synapse & Lakehouse Concept**:
    - Understood the difference between **dedicated SQL pool** (traditional DW) and **serverless SQL pool** (Lakehouse).
    - Challenges: Managing external tables and credentials.
4. **Power BI Integration**:
    - Successfully connected Synapse to Power BI using SQL endpoints.

## Conclusion

This project successfully implemented a **scalable, automated, and end-to-end data pipeline** on Azure. Key achievements:

- **Efficient Data Flow**: From raw ingestion to analytics-ready datasets.
- **Cost Optimization**: Leveraged serverless components (Synapse, Databricks).
- **Real-World Applicability**: Used dynamic pipelines and Lakehouse architecture.

## GitHub & Code References

- Github
- Linkdin