

# Azure End-To-End Data Engineering Project

## Project Overview

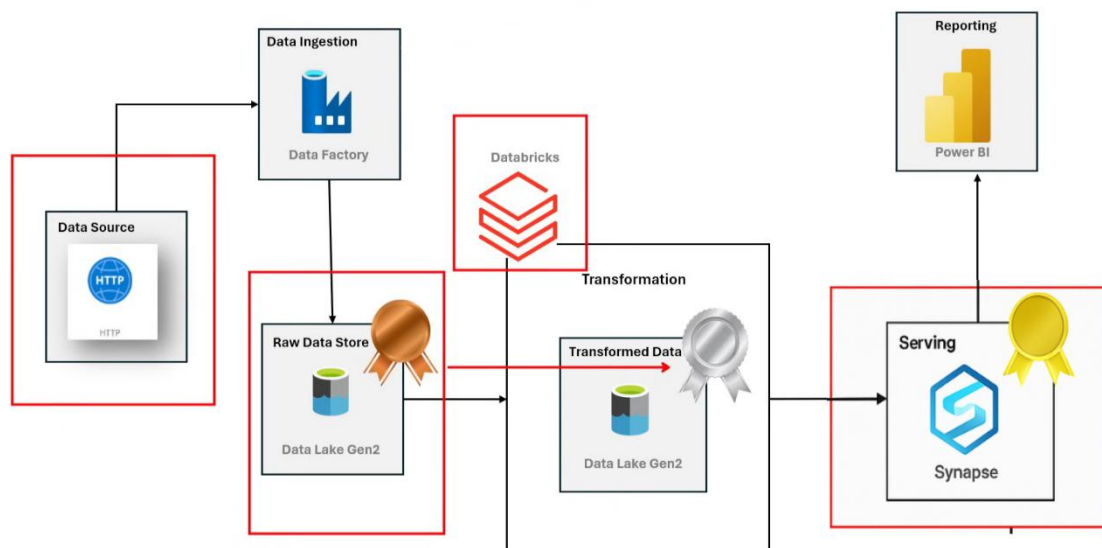
This project demonstrates a comprehensive end-to-end data engineering solution using Microsoft Azure. The goal was to build a scalable data pipeline that ingests, transforms, and serves data for analytics. The project leverages multiple Azure services, including:

- 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).

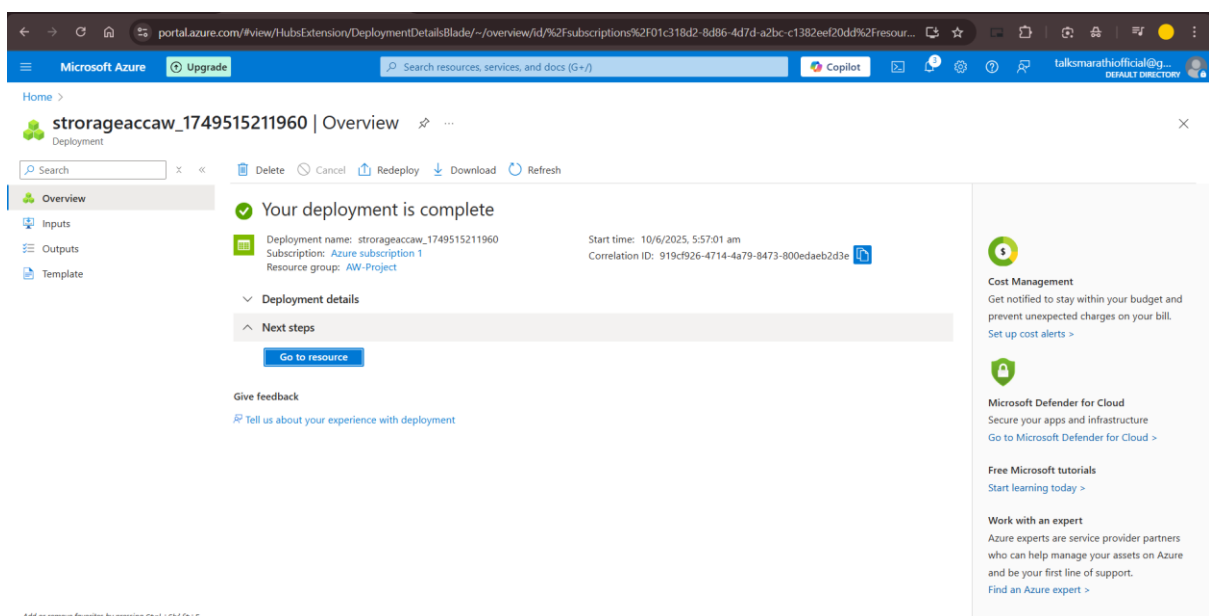
## 3. Data Serving (Gold Layer)

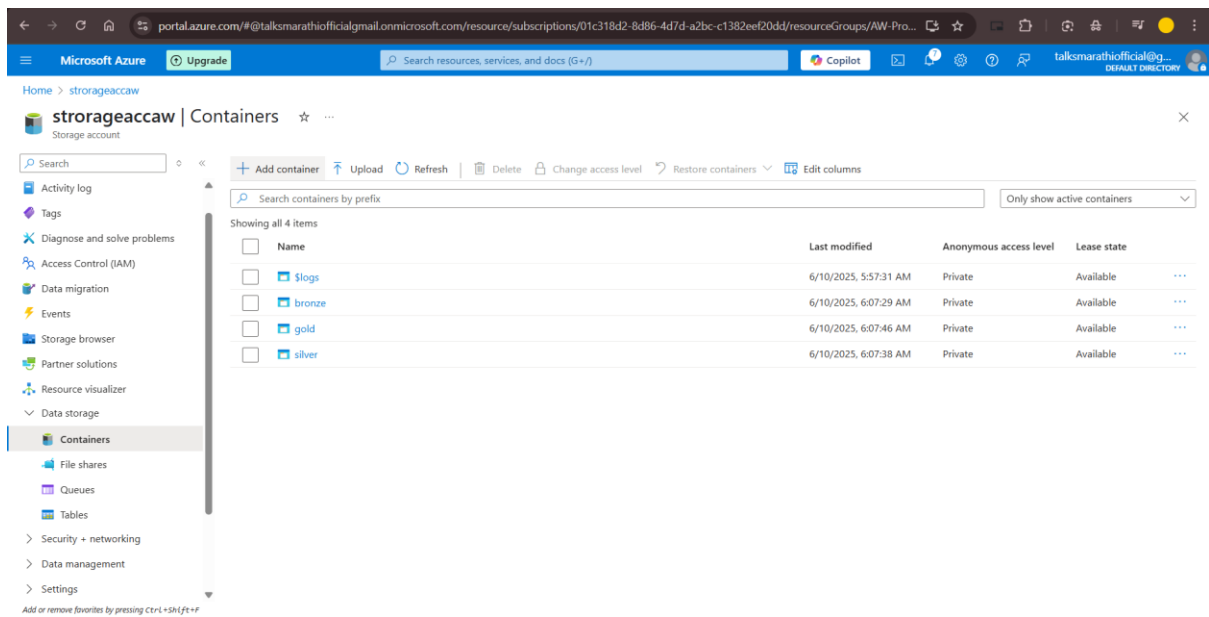
- Tool: Azure Synapse Analytics for SQL-based data modeling.
- Storage: Azure Data Lake (Gold container).
- Visualization: Power BI for reporting.

# Detailed Implementation

## 1. Setting Up Azure Resources

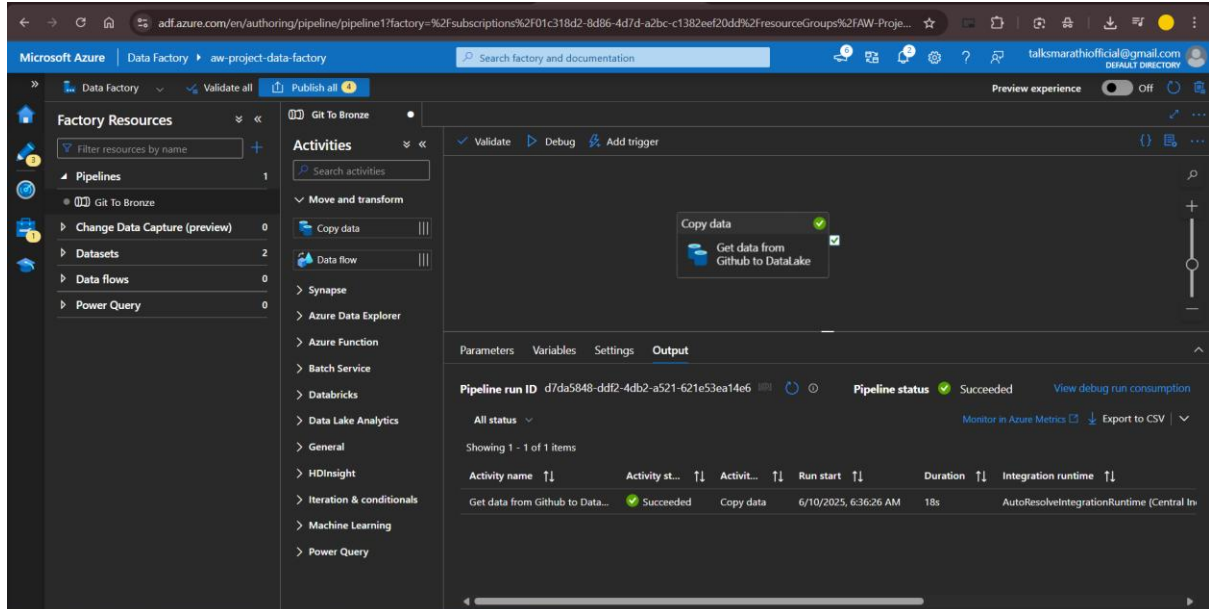
- **Resource Group:** Created to organize all Azure services.
- **Azure Data Lake Storage (ADLS Gen2)**
  - Enabled hierarchical namespace for folder structure.
  - **Created three containers:**
    - Bronze (raw data)
    - Silver (cleaned & transformed data)
    - Gold (serving layer for analytics)





## 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.

The screenshot displays the Microsoft Azure Data Factory interface. On the left, the 'Factory Resources' pane shows a list of pipelines, including 'Git To Bronze' and 'Dynamic Git To Bronze'. The main workspace shows a pipeline diagram with a 'Lookup' activity (labeled 'Lookup git json') and a 'ForEach' loop (labeled 'ForEach Git') containing a 'Copy data' activity. The 'Output' tab at the bottom shows a table of activity results.

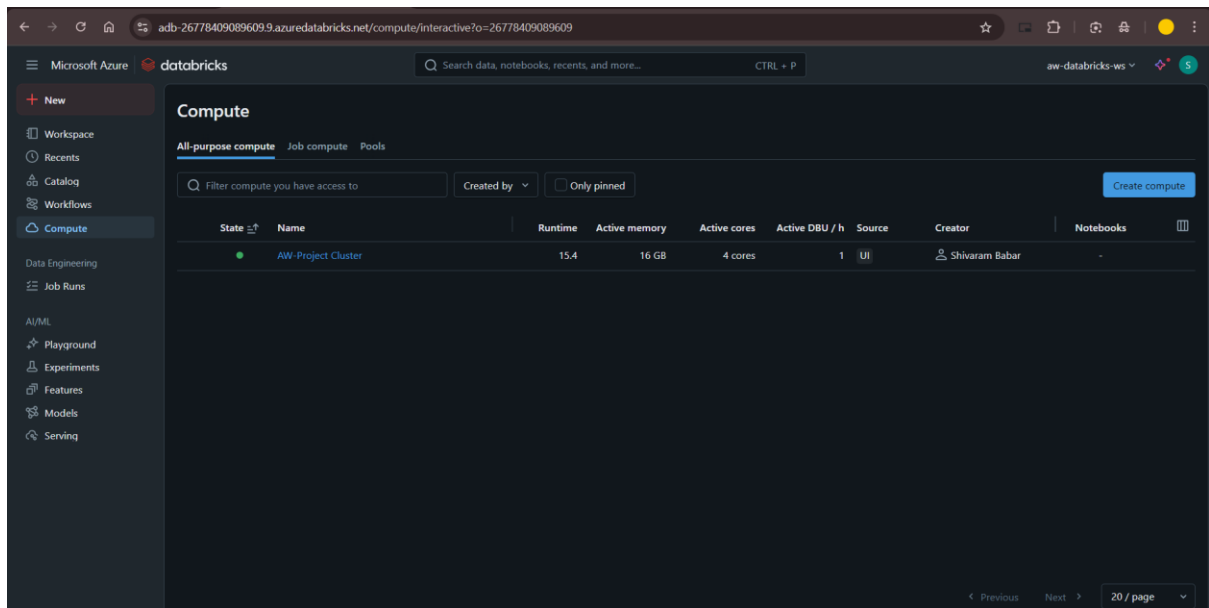
Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime
Copy data dy...	✓ Succeeded	Copy data	6/10/2025, 8:17:23 AM	15s	AutoResolveIntegrationRuntime (Centra
Copy data dynamically	✓ Succeeded	Copy data	6/10/2025, 8:17:09 AM	13s	AutoResolveIntegrationRuntime (Centra
Copy data dynamically	✓ Succeeded	Copy data	6/10/2025, 8:16:54 AM	14s	AutoResolveIntegrationRuntime (Centra
Copy data dynamically	✓ Succeeded	Copy data	6/10/2025, 8:16:39 AM	14s	AutoResolveIntegrationRuntime (Centra
Copy data dynamically	✓ Succeeded	Copy data	6/10/2025, 8:16:12 AM	26s	AutoResolveIntegrationRuntime (Centra
Copy data dynamically	✓ Succeeded	Copy data	6/10/2025, 8:15:55 AM	15s	AutoResolveIntegrationRuntime (Centra
Copy data dynamically	✓ Succeeded	Copy data	6/10/2025, 8:15:42 AM	13s	AutoResolveIntegrationRuntime (Centra

The screenshot shows the Microsoft Azure portal interface for a storage account named 'bronze'. The 'Overview' page is displayed, showing a list of blobs and folders. The 'Authentication method' is set to 'Access key'. The 'Showing all 12 items' section lists the following items:

Name	Last modified	Access tier	Blob type	Size	Lease state
AdventureWorks_Calendar	6/10/2025, 8:15:23 AM				...
AdventureWorks_Customers	6/10/2025, 8:15:39 AM				...
AdventureWorks_Product_Categories	6/10/2025, 8:15:10 AM				...
AdventureWorks_Products	6/10/2025, 8:16:08 AM				...
AdventureWorks>Returns	6/10/2025, 8:16:36 AM				...
AdventureWorks_Sales_2015	6/10/2025, 8:16:51 AM				...
AdventureWorks_Sales_2016	6/10/2025, 8:17:06 AM				...
AdventureWorks_Sales_2017	6/10/2025, 8:17:20 AM				...
AdventureWorks_Territories	6/10/2025, 8:17:35 AM				...
Parameters	6/10/2025, 7:57:07 AM				...
Product_Subcategories	6/10/2025, 8:15:53 AM				...

### 3. Data Transformation with Azure Databricks

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

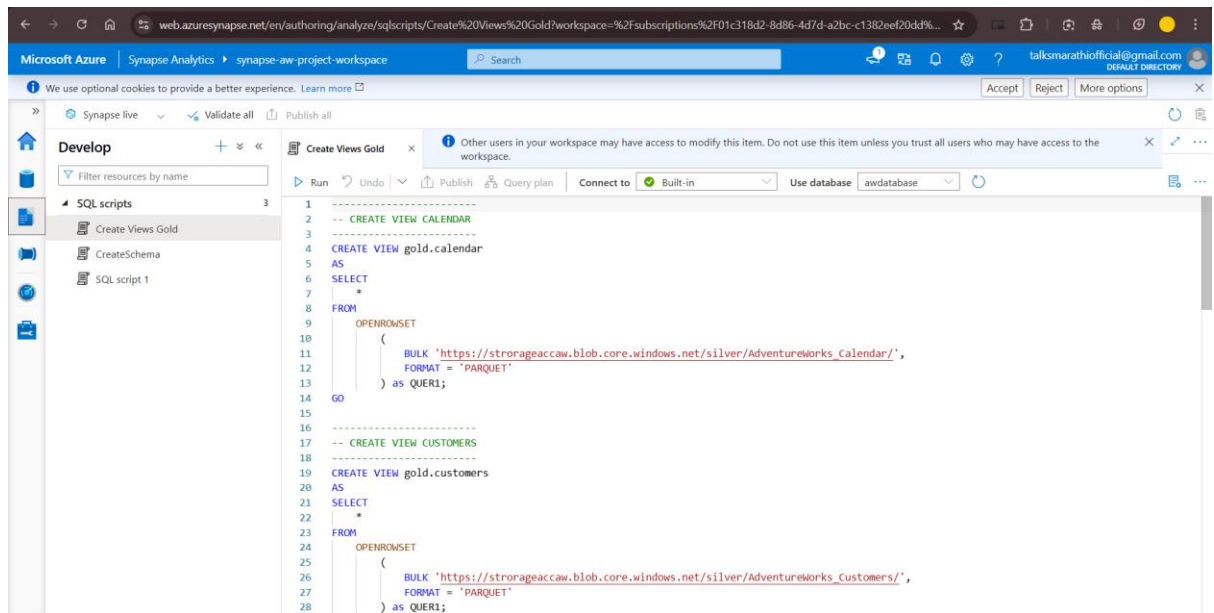


- **Data Loading:** Read data from the Bronze layer into Spark DataFrames.
- **Transformations Applied:**
  - **Date Functions:** Extracted month and year from dates.
  - **String Manipulation:** Concatenated columns (e.g., full customer name).
  - **Split Operations:** Separated product SKU into categories.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

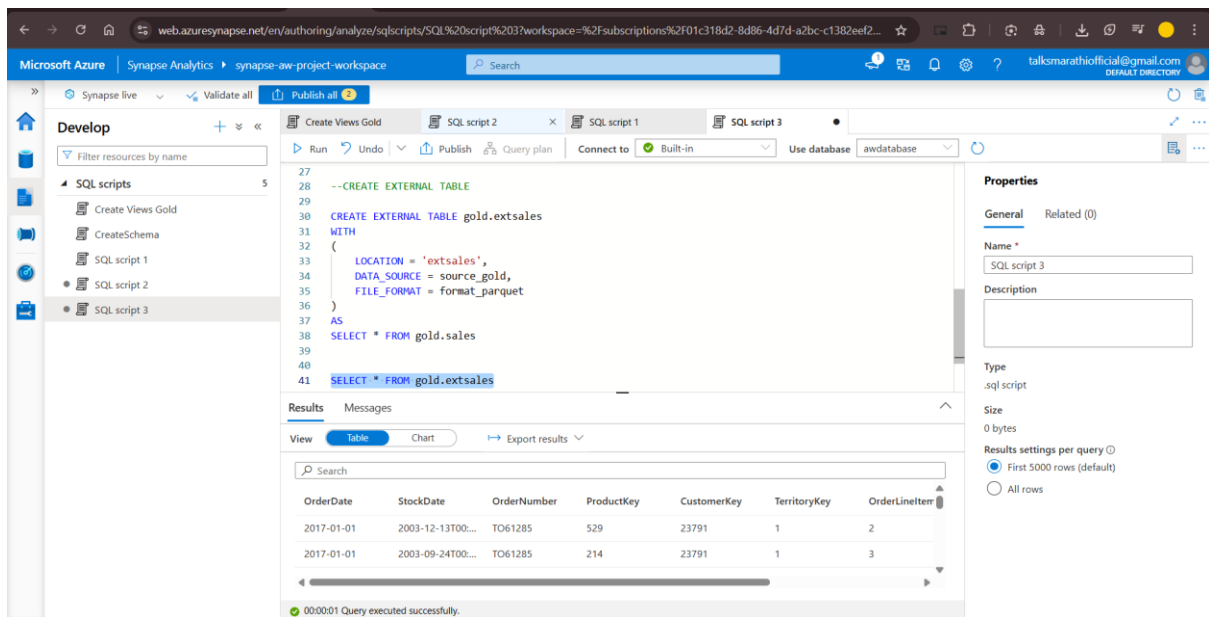
**Notebook Link :** <https://github.com/shivaramsb/Adventure-Works-Data-Engineering-Project/blob/main/Silver-layer.ipynb>

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

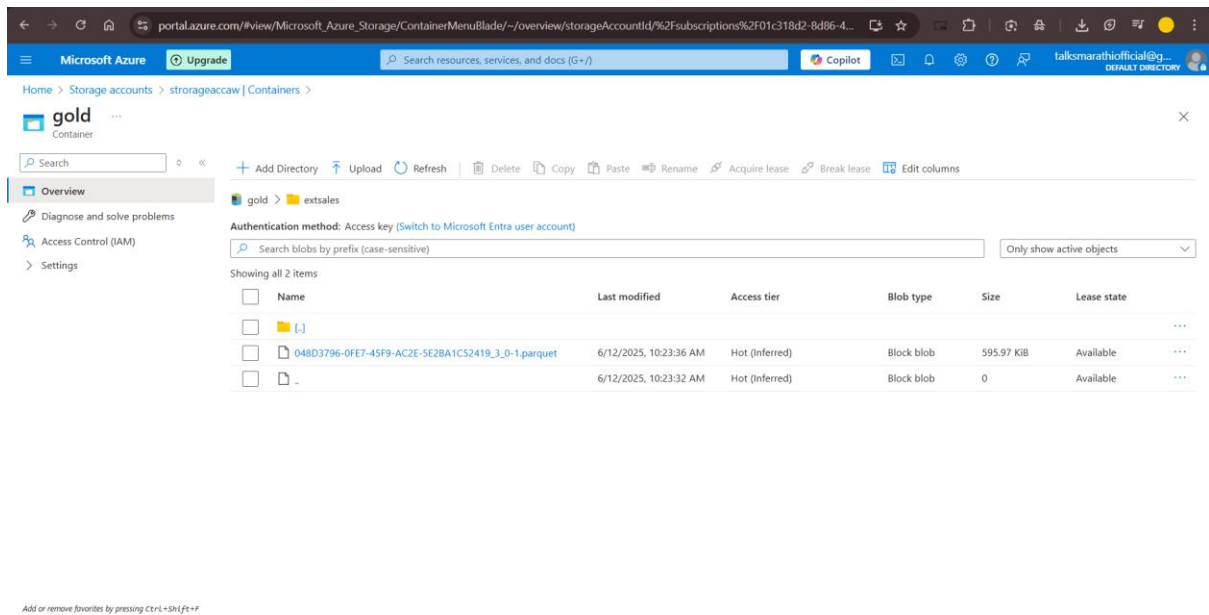
- **Serverless SQL Pool:** Used for querying data directly from ADLS.
- **Views Creation:** Defined SQL views on Silver layer data.



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

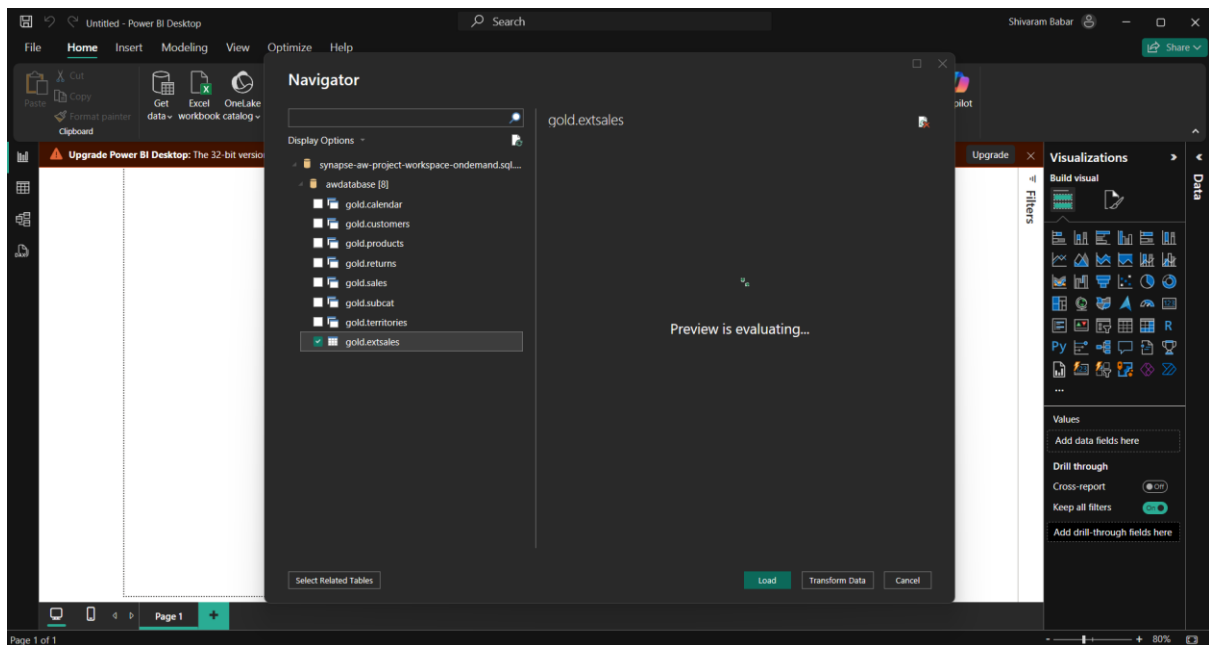


- **Stored final data in the Gold layer for Power BI consumption.**

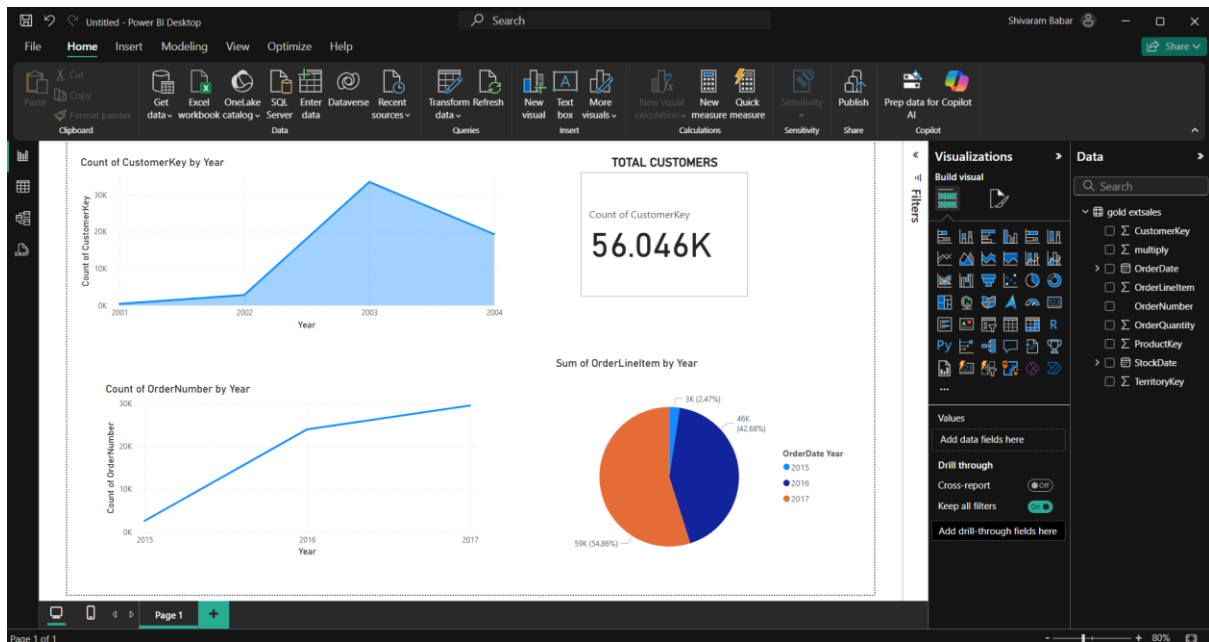


## 5. Visualization with Power BI

- **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](#)