A black background with grey leaves

AI-generated content may be incorrect.

**Retail Sales Data Lakehouse – Bootcamp Project 2**

[Document subtitle]

A grey logo on a black background

AI-generated content may be incorrect.

Student Name – Sai Sravan Kumar Repaka

Trainer – Dr. Praveen Sharma

**1. Objective**

The goal of this project is to build a complete end-to-end **data Lakehouse pipeline** using Azure services. The pipeline ingests retail sales data from local i.e. on-prem files, processes it through Bronze → Silver → Gold layers in **Azure Data Lake Storage (Gen2)**, and finally upserts it into a **Dedicated SQL Pool** using Azure Data Factory. The architecture ensures scalable, secure, and efficient data transformation and reporting.

**2. Architecture Overview**

**Components Used:**

* **Self-hosted Integration Runtime (SHIR)** – for secure on-prem file ingestion
* **Azure Data Factory (ADF)** – for orchestration
* **Azure Data Lake Storage (ADLS Gen2)** – for multi-layered data storage (Medallion architecture)
* **ADF Data Flows & Notebooks** – for transformations and upserts
* **Azure Synapse SQL Pool** – for storing gold-level analytics-ready data

**Architecture**

**A diagram of a company

AI-generated content may be incorrect.**

**Complete End to End Pipeline in ADF.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**3. Technologies Used**

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **Component** | | |  | | --- | | **Technology** | |
| |  | | --- | | Data Orchestration | | |  |  |  | | --- | --- | --- | | |  | | --- | | Azure Data Factory |  |  | | --- | |  | | |
| Storage | Azure Data Lake Gen2 |
| Transformation | ADF Data Flow + Notebooks |
| Ingestion Runtime | Self-hosted Integration Runtime |
| Target | Azure Synapse Dedicated SQL Pool |
| File Format | Parquet (Silver and Gold layers) |

**4. Step-by-Step Pipeline Execution**

✅ **Step 1: Bronze Ingestion**

* Used Self-hosted IR to copy files from local laptop to bronze container in ADLS Gen2.
* Files ingested: customers.csv, products.csv, sales.csv.

✅ **Step 2: Bronze to Silver (Transformation Layer)**

* Mounted ADLS container in a notebook.
* Read Bronze files and applied transformations:
  + Schema standardization
  + Null handling
  + Type casting
* Stored the transformed data as Parquet files in Silver layer.A screenshot of a computer

  AI-generated content may be incorrect.

✅ **Step 3: Silver to Gold (Aggregation Layer)**

* Read data from Silver files using a second notebook.
* Performed:
  + Joins between customers, products, and sales
  + Aggregations like total sales per product or customer
* Output written to Gold layer in Parquet format.

A screenshot of a computer

AI-generated content may be incorrect.

✅ **Step 4: Upsert to Dedicated SQL Pool**

* Created a target table in Synapse Dedicated SQL Pool with schema matching the Gold dataset.
* Configured ADF Data Flow:
  + Used Alter Row transformation for upsert logic
  + Added two records (1 new, 1 modified) to test upsert

A screenshot of a computer

AI-generated content may be incorrect.

* Validated that:
  + New record inserted
  + Existing record updated successfully

A screenshot of a computer

AI-generated content may be incorrect.

**5. Key Features**

* Seamless integration from **on-premises** to **cloud**
* Layered Data Lake architecture: **Bronze → Silver → Gold**
* **Parquet** storage for optimized processing
* Upsert functionality using **Data Flow + Alter Row**
* Data validation by manually inserting test rows
* Secure file access using Self-hosted IR

**6. Data Validation**

* Manually added rows to Gold files to test upsert logic
* Verified record count before and after load
* Used preview and SQL queries to confirm updates

**Challenges faced:**

In the data flow, when creating the sink to the table in Synapse dedicated SQL pool, at the time of creating the SQL server by default Azure will give the authentication methods as Use both local and Entra ID.

A screenshot of a computer

AI-generated content may be incorrect.

But during the linked service creation ADF expects only one authentication method and throws an error. In order to resolve this, we need to use the legacy connection version. By default, Azure will provide the Version as 2.0. We need to change it to 1.0

Source: <https://learn.microsoft.com/en-us/answers/questions/1861971/how-to-fix-only-one-valid-authentication-should-be>

A screenshot of a computer

AI-generated content may be incorrect.