Automating a Data Pipeline with Databricks Jobs and Workflows

1. Data pipeline architecture.

The pipeline is built on a multi-layered Delta Lake architecture using Databricks Workflows to orchestrate three main notebooks: ingestion, transformation, and storage. The data flows through the following layers:

Bronze Layer: Raw CSV files from ADLS container are ingested using the ingestion notebook and stored as Delta tables at /mnt/datalake/raw/.

Silver Layer: The transformation notebook cleans and aggregates the bronze data. It produces cleaned datasets saved at `/mnt/datalake/output/`.

Gold Layer: The storage notebook joins customer and transaction data to produce final analytics-ready datasets. Results are registered as a table total_revenue.

2. Workflow Configuration.

A Databricks multi-task workflow orchestrates the three notebooks in a sequential manner:

Task 1: Data Ingestion Notebook
Task 2: Data Transformation Notebook (dependent on task 1)
Task 3: Data Storage Notebook (dependent on task 2)

Workflow is configured within a Databricks Job and runs daily at midnight using the built in scheduler (6:30 pm UTC).

3. Challenge faced and how you resolved them.

- **Invalid Column Names with Spaces:**

  Some CSV files had column names with spaces (e.g., 'Customer ID', 'Store ID'), which led to runtime errors during transformations and joins in Spark.

  Resolution:

  Renamed columns using sanitized names (e.g., 'Customer_ID', 'Store_ID') right after reading the DataFrames, and used aliasing during joins to prevent naming conflicts.