## **Day 13 - Azure Lifecycle & ETL Process**

## **Date -** June 25, 2025

### **1. Azure Data Lifecycle Overview**

You mapped out a clear, structured flow of data processing stages:

* **Source** Raw data origins: CRM, IoT devices, medical scans, APIs, logs.
* **Ingestion** Tool: Azure Data Factory (ADF)  
   Actions: Batch/stream ingestion into Azure Data Lake Storage Gen2.
* **Exploration** Tools: SQL Data Warehouse, PolyBase, Azure Databricks  
   Actions: Data profiling, lightweight transformations, anomaly detection.
* **Prepare & Train** Tools: Azure Databricks (SQL, PySpark/Python)  
   Actions: ETL operations, feature engineering, ML model training.
* **Model & Serve** Tools: Azure SQL, Azure Synapse Analytics  
   Actions: Store and serve refined datasets for consumption.
* **BI & Analytics** Tools: Power BI, Tabular Models, real-time analytics tools  
   Actions: Visualization, periodic or real-time analytics, business-driven decisions.
* **Underlying Storage** Centralized on ADLS Gen2 — supports all stages with scalable, secure storage.

✅ **End-to-end Flow**:  
 Source → Ingest (ADF) → Explore (Databricks/SQL) → Prepare & Train → Model & Serve → BI  
 **Objective**: Scalable and efficient analytics pipelines suited for AI and BI workloads.

### **2. Hands-On Setup: Farm Data ETL Pipeline**

#### **Phase 1: Storage Account & Blob Container Setup**

* Created Azure Storage: specified subscription, resource group, unique name.
* Added Blob container: farmdatasets (private access).
* Uploaded sample CSV files: weather\_data.csv, soil\_data.csv, fertilizer\_usage.csv.

#### **Phase 2: Databricks Workspace & Compute Setup**

* Provisioned an Azure Databricks Workspace named (e.g.) greenfarm-databricks.
* Launched workspace and created a single-node cluster (greenfarm-cluster) with latest runtime.

#### **Phase 3: Storage Mounting in Databricks**

* Retrieved storage account access key.

Mounted blob container using:  
  
 dbutils.fs.mount(

source="wasbs://farmdatasets@<storage>.blob.core.windows.net/",

mount\_point="/mnt/farmdatasets",

extra\_configs={

"fs.azure.account.key.<storage>.blob.core.windows.net": "<account-key>"

}

)

* Verified mount: visible 3 CSV files via display(dbutils.fs.ls("/mnt/farmdatasets"))

#### **Phase 4: PySpark-based ETL in Databricks**

1. **Extract** Loaded CSVs into Spark DataFrames with schema inference.
2. **Transform**
   * Aggregated weather data (e.g., average temperature, total rainfall by farm).
   * Joined datasets (weather, soil, fertilizer) on farm\_id.
   * Handled nulls by replacing with 0.
3. **Load** (Though not fully described, implied steps to write cleaned data back to storage or Delta tables.)

## **Key Achievements**

| **Aspect** | **Completed Actions** |
| --- | --- |
| **Infrastructure Setup** | ADLS Gen2 containers created, Databricks workspace & cluster provisioned |
| **Data Connectivity** | Containers mounted to Databricks |
| **ETL Pipeline** | Extracted, transformed, aggregated, and prepared data via PySpark |
| **Data Engineering Blueprint** | Aligned with Medallion Architecture concept: raw ingestion → processing → readying for analytics/ML |

## **✅ Next Steps & Recommendations**

* **Delta Lake Optimization**: Persist transformed/aggregated tables in Delta format for versioning and performance.
* **Gold Layer Integration**: Clean final datasets into structured tables (e.g., farm\_summary) for BI consumption.
* **BI Integration**: Connect outputs to Power BI dashboards.
* **ML Workflows**: Use prepared data for predictive modeling (crop yield, soil analysis, weather impact).