NYC Taxi Data Engineering Pipeline – Project Report (2025)

1. Project Overview

This project delivers a full-scale, industry-ready **end-to-end data engineering pipeline** using the NYC Taxi Trip dataset. Built on **Azure Cloud** and **Databricks**, the pipeline simulates a real-time enterprise workflow covering **data ingestion, transformation, storage, and analytics serving**. The solution emphasizes **dynamic API ingestion**, **medallion architecture**, **Delta Lake features**, and **Power BI connectivity**, equipping learners with hands-on exposure to modern data engineering tools and cloud platforms.

---

2. Objective

To build a cloud-native data pipeline that:

- Ingests monthly NYC Taxi trip data dynamically via API (without manual upload).
- Stores raw data in a structured **medallion architecture** (Bronze → Silver → Gold).
- Applies schema enforcement, transformations, and optimizations using **PySpark on Databricks**.
- Leverages **Delta Lake** for ACID compliance, time travel, and data versioning.
- Exposes curated data for reporting tools like **Power BI**.

---

3. Tools and Technologies

Layer	Technology / Service
Cloud	Microsoft Azure
Orchestration	Azure Data Factory (ADF)

Processing	Azure Databricks (PySpark)
Storage	Azure Data Lake Gen2 + Delta Lake
Ingestion	Public API via HTTP linked service
Format	Parquet (base), Delta (transactional)
Security	Service Principal & Managed Identity
BI Layer	Power BI Desktop

---

4. Architecture

The architecture follows the **Medallion Layer Pattern**:

- **Bronze Layer**: Raw monthly data fetched directly from NYC Taxi API and stored in Parquet format.
- **Silver Layer**: Cleaned and transformed data with enriched schema, split columns, and date parsing.
- **Gold Layer**: Curated Delta Tables with optimized schemas, updates, and deletes enabled for reporting.

Security is ensured using **Azure Active Directory**, **Service Principals**, and **role-based access control**.

---

5. Implementation Steps

Step 1: Azure Setup

- Created Azure Resource Group, Storage Account with Hierarchical Namespace.
- Set up containers: `bronze`, `silver`, and `gold`.

Step 2: Data Ingestion via ADF

- Configured HTTP linked service to connect to NYC Taxi API.
- Parameterized pipeline with:
    - Dataset for year and month input.
    - Dynamic URL expression with conditionals to handle single-digit month formatting.
    - `ForEach` loop to automate data ingestion from Jan to Dec.
- Output written in `.parquet` format to the Bronze zone.

Step 3: Databricks and Silver Layer

- Created a Databricks workspace and cluster.
- Configured Service Principal to authenticate and access ADLS Gen2.
- Ingested `.parquet` files from Bronze using recursive directory reading.
- Applied PySpark transformations:
    - Schema definition using `StructType`.
    - Column renaming and standardization.
    - Splitting composite columns using `split()`.
    - Date extraction (day, month, year) from timestamps.
- Stored cleaned data into the Silver zone using Parquet format in **append** mode.

Step 4: Delta Lake and Gold Layer

- Created **external Delta tables** from Silver layer data:
    - Stored in `gold` container.
    - Metadata managed using Databricks SQL.

- Enabled ACID compliance with:
    - Insert, update, and delete support.
    - Schema evolution.
    - Time travel (query by version or timestamp).
- Demonstrated rollback and audit capabilities via Delta Log inspection.

Step 5: Power BI Integration

- Used **Partner Connect** from Databricks to generate connection to Power BI Desktop.
- Access token used for authentication.
- Imported Delta Tables for report building.

---

6. Key Features Demonstrated

- ✅ Dynamic pipeline generation with parameterized datasets and loops.
- ✅ Automated API ingestion (avoids CSV upload bottleneck).
- ✅ Recursive folder reading for nested Parquet structures.
- ✅ Schema enforcement and transformation using PySpark.
- ✅ Use of Delta Lake for versioning and rollback.
- ✅ Service Principal and RBAC for secure access.
- ✅ Power BI integration for downstream analytics consumption.

---

7. Real-World Concepts Covered

Concept	Application in Project
Medallion Architecture	Bronze, Silver, Gold data zones
Parquet over CSV	Efficient columnar format for large-scale reads
Dynamic ADF Pipelines	Scalable API ingestion across months
Delta Lake	Transactional operations and version control
PySpark Transformations	ETL using DataFrame API
Recursive Ingestion	Reads nested monthly directories dynamically
Role-Based Access Control	Via Managed Identity and Service Principal
Power BI Analytics	Consuming Delta Tables without manual exports

---

8. Output and Results

- Successfully ingested and transformed 12 months of green taxi trip data.
- Created efficient Delta Tables supporting query, update, and rollback.
- Connected data to Power BI for visualization and business insights.