Cloud-Based Loan Default Prediction Using Azure Data Engineering

1. Project Overview

This project focuses on building a fully cloud-based data engineering pipeline on Microsoft Azure to predict loan defaults using both historical batch data and real-time streaming data. The end-to-end solution ingests, stores, processes, analyzes, and visualizes loan and customer data, enabling banks to identify high-risk applicants early and reduce potential financial loss.

The implementation uses Azure Data Factory for ingestion, Azure Data Lake Gen2 for storage, Azure Databricks for data processing, Azure Synapse for analytics, Delta Lake for ACID storage, Azure Event Hubs for real-time ingestion, Power BI for dashboards, and Logic Apps/Monitor for alerts.

---

2. Problem Statement

Banks and financial institutions often struggle to evaluate the risk level of loan applicants due to incomplete data analysis and lack of real-time decision systems. The existing manual or legacy systems cannot detect patterns, fraud indicators, or default risk in time.

The objective is to automate loan-risk prediction using cloud-based data engineering, enabling:

- Reliable batch ingestion of historical loan data
- Real-time detection of potentially risky loan applications
- Centralized Lakehouse storage with versioning & auditing
- Analytics dashboard for business decision-makers

3. Azure Services Used

Layer	Azure Service	Purpose
Batch Ingestion	**Azure Data Factory (ADF)**	Copy data from Azure MySQL → ADLS
Real-Time Ingestion	**Azure Event Hubs**	Stream live loan requests
Storage	**Azure Data Lake Storage Gen2 (ADLS)**	Raw, Cleansed, Curated zones
Processing	**Azure Databricks (Spark)**	Batch ETL + ML + streaming
Storage Format	**Delta Lake**	ACID, time-travel, schema enforcement
Analytics	**Azure Synapse SQL / Serverless**	Queries, BI connectivity
Monitoring	**Azure Monitor + Log Analytics**	Metrics, performance, alerts
Alert Automation	**Azure Logic Apps**	Notify on high-risk detection
Visualization	**Power BI**	Dashboard for default prediction insights

4. Project Plan (3-Day Execution)

Day	Focus Area	Output
Day 1	Batch ingestion, Raw zone setup, Schema validation	ADF pipeline + ADLS Bronze data
Day 2	Databricks ETL, Delta optimization, analytics, risk scoring	Silver + Gold zone data + Synapse SQL views
Day 3	Real-time streaming pipeline, alerts, Power BI dashboard	Event Hub streaming + alerts + PBIX report

---

5. Roles and Responsibilities of Data Engineers

Team	Responsibility	Azure Components
Team 1	Data Ingestion & Storage	ADF, ADLS Gen2
Team 2	Data Cleaning & Transformation	Databricks Batch
Team 3	Risk Analytics & Delta Optimization	Databricks + Synapse
Team 4	Real-Time Processing	Event Hubs + Databricks Streaming
Team 5	Dashboard & Alerts	Power BI + Logic Apps

6. Detailed Day-Wise Plan

✅ Day 1 – Data Ingestion, Storage, and Preparation

Objective: Move source data to Azure, validate schema, and store in Bronze zone.

Tasks:

1. Create Azure Storage Account + ADLS Gen2 container
   - Folders: `/bronze/loans`, `/silver/loans`, `/gold/loans`
2. Set up Azure Data Factory pipeline
   - Source: Azure MySQL Flexible Server
   - Sink: ADLS Gen2 (Parquet format)
3. Run validation notebook in Databricks
   - Check nulls, schema drift, duplicate rows
4. Register raw Delta table in Databricks metastore

Deliverables:

- ADF pipeline JSON export
- Raw dataset in `bronze` zone
- Schema validation notebook

✅ Day 2 – Data Processing, Analytics, and Optimization

Objective: Clean data, remove duplicates, enforce schema, build risk rules.

Tasks:

1. Load Bronze → Clean → Write to Silver (Delta format)
2. Feature engineering: EMI/Income, loan-to-income ratio, credit score band
3. Detect outliers using Z-score/IQR in Spark
4. Load Silver → Create curated Gold table (for analytics/models)
5. Register external tables in Synapse SQL
6. Build risk rules (ex: Income < 25K & EMI > 40% → High Risk)

Deliverables:

- Notebooks for Bronze→Silver→Gold
- Delta tables in all 3 zones
- Synapse SQL views for Power BI

✅ Day 3 – Real-Time Streaming, Alerts, and Dashboard

Objective: Detect and flag risky loan applicants in real-time.

Tasks:

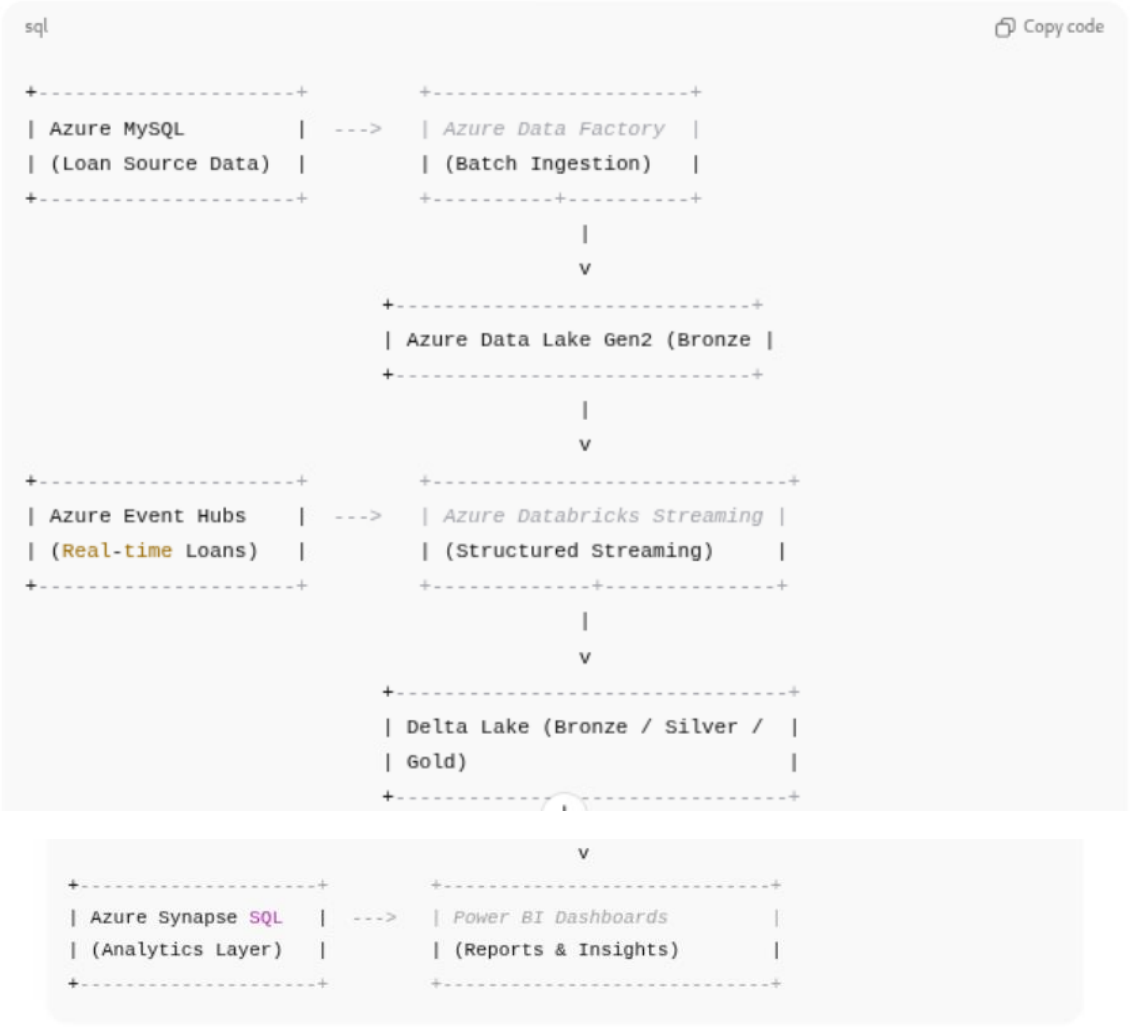
1. Create Azure Event Hub topic: `loan_stream`
2. Create Databricks streaming job to read Event Hub → Delta
3. Apply ML/rule-based classification in streaming mode
4. Insert flagged records into Gold `high_risk_loans` table
5. Build Power BI report using Synapse connector
6. Configure alert with Logic App:
   - If `risk_score >= 0.7` → Email/SMS alert

Deliverables:

- Event Hub → Delta Streaming Notebook
- Power BI dashboard (`.pbix`)
- Automated email alert workflow

7. Azure Architecture Diagram (Text Format)

sql Copy code



8. Sample Code Snippets

✅ Azure Data Factory Pipeline JSON

json Copy code

```json
{
  "name": "Ingest_Loan_Data",
  "properties": {
    "activities": [
      {
        "name": "CopyFromMySQL",
        "type": "Copy",
        "inputs": [{ "referenceName": "MySQLSource" }],
        "outputs": [{ "referenceName": "ADLSDataset" }]
      }
    ]
  }
}
```

✅ Databricks Streaming Code (Event Hub → Delta)

python Copy code

```python
stream_df = (spark.readStream.format("eventhubs")
  .option("eventhubs.connectionString", eh_connection)
  .load())

parsed = stream_df.selectExpr("cast(body as string) as json") \
  .select(from_json("json", loan_schema).alias("data")) \
  .select("data.*")

parsed.writeStream.format("delta") \
  .outputMode("append") \
  .option("checkpointLocation", "/chk/loans") \
  .start("/mnt/gold/streamed_loans")
```

✅ Synapse External Table (Query Delta Lake)

sql Copy code

```sql
CREATE EXTERNAL TABLE loan_gold
USING DELTA
LOCATION 'abfss://gold@storageaccount.dfs.core.windows.net/loans/';
```

9. Deliverables Summary

✅ ADF Pipeline JSON
✅ Databricks Notebooks (.ipynb or .dbc)
✅ Delta Lake Tables (Bronze, Silver, Gold)
✅ Synapse SQL Script
✅ Power BI Dashboard (.pbix)
✅ Real-time alert workflow (Logic Apps)

---

10. Tools & Technologies

Category	Azure Service
Ingestion	Data Factory, Event Hubs
Storage	ADLS Gen2, Delta Lake
Processing	Databricks (Spark)
Analytics	Synapse SQL, Serverless
Dashboard	Power BI
Alerts	Logic Apps, Monitor

11. Dataset Schema (Example)

Column Name	Type	Description
Loan_ID	String	Unique loan identifier
Customer_ID	String	Customer reference
Loan_Amount	Float	Approved loan amount
Annual_Income	Float	Declared annual income
CIBIL_Score	Integer	Credit score (300-900)
Loan_Status	String	Default / Paid
Employment_Status	String	Employed / Self-Employed
EMI	Float	Monthly installment
Term	Integer	Loan duration in months

I will now begin rewriting the full document with the following changes:

🔁 What will be transformed:

Old (Local Hadoop)	New (Azure Cloud)
Sqoop	Azure Data Factory
MySQL (local)	Azure Database for MySQL
HDFS	Azure Data Lake Storage Gen2
Hive External Tables	Synapse Serverless / Dedicated SQL + Delta Lake
Spark (local)	Azure Databricks Spark
Kafka (local)	Azure Event Hubs
Delta Tables (local)	Delta Lake on ADLS
Tableau/Grafana	Power BI
Spark JAR on cluster	Databricks notebooks / jobs
Local alerts	Azure Logic Apps / Monitor alerts