# Netflix ELT Pipeline | Azure Data Engineering

A production-style data engineering pipeline that ingests, transforms, and streams Netflix data using a fully automated and modular design.

## Tech Stack

• Azure Data Factory (ADF) – Orchestration & dynamic pipeline control  
• Azure Databricks – Transformation notebooks with parameterization  
**• Unity Catalog** – Access control, metastore, external locations  
• Delta Lake + Delta Live Tables (**DLT**) – Bronze/Silver/Gold architecture  
• Azure Data Lake Storage (ADLS Gen2) – Layered storage with ABFSS  
• GitHub – Version control and metadata integration

## Key Features

* ✅ Dynamic & Parameterized Pipelines

Ingests files from GitHub dynamically using ADF’s Web Activity. Uses ForEach + Set Variable activities for looped file processing. Databricks notebooks are fully parameterized to support scalable reuse.

* ✅ Conditional Execution Based on Weekday

Uses a `weekday\_lookup` logic to determine the day of execution. Runs heavy transformations (`silver\_Master\_Data`) only on Sundays. On other days, it conditionally exits via a `FalseNotebook` — boosting efficiency.

* ✅ Layered ELT with Delta Architecture

Bronze: Raw ingestion via Auto Loader  
Silver: Cleansing, flag creation, type casting, enrichment  
Gold: Aggregated tables with KPIs using DLT including rank(), groupBy(), and filtering.

* ✅ Delta Live Tables & Unity Catalog

DLT pipelines visualize end-to-end table flow using DLT Graph. Unity Catalog is used to secure tables and manage access centrally.

* ✅ Fully Integrated & Git-Backed

GitHub metadata is validated and looped dynamically for ingestion. All notebooks are versioned and public at: https://github.com/AlishaRuqshan/azure-databricks-netflix-pipeline-DLT-databricks

## Visual Snapshots

• DLT table lineage and streaming record flow  
• Conditional logic via Databricks workflows  
• Parameter-driven ADF and notebook chains  
\*Images provided in the repository for full walkthrough\*