Distributed Music Streams Processing Using Airflow, Spark & DynamoDB

# Executive Summary

This report details an ETL workflow using AWS services: Apache Airflow, AWS Glue (PySpark & Python Shell), Amazon S3, and DynamoDB. The goal of this project is to process and transform music streaming data and store it efficiently in DynamoDB for low-latency retrieval by microservices.

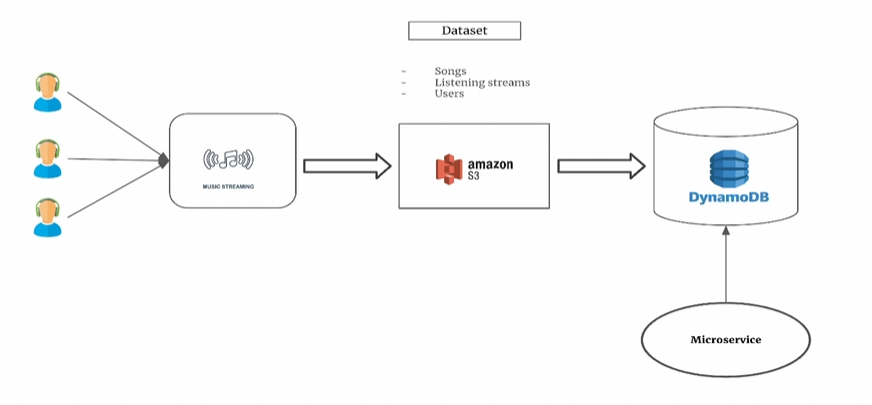
# Introduction

Unlike traditional batch processing, this ETL workflow is designed to handle irregular incoming data intervals and provide rapid data availability. The architecture separates orchestration (Airflow) from heavy data transformation (AWS Glue), leveraging Spark for large-scale processing and DynamoDB as a NoSQL database optimized for high-speed queries.

# Solution Architecture

The architecture consists of three main layers:  
1. \*\*Storage Layer\*\*: Data is ingested from Amazon S3.  
2. \*\*Orchestration Layer\*\*: Airflow manages job execution and data movement.  
3. \*\*Processing Layer\*\*: AWS Glue performs Spark transformations and Python-based ingestion.  
4. \*\*Database Layer\*\*: DynamoDB stores processed metrics for real-time access.

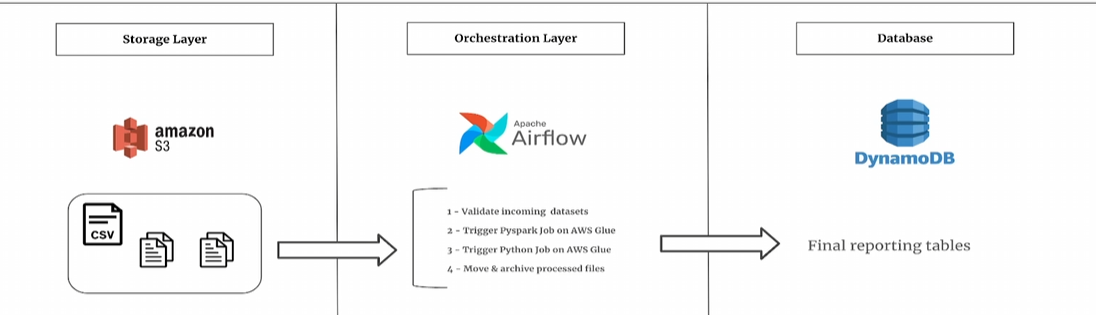
The high-level architecture is depicted below:



# Data Pipeline Workflow

The ETL pipeline consists of the following key steps:  
1. \*\*Data Validation\*\*: Airflow verifies the necessary input files exist.  
2. \*\*Trigger PySpark Job (AWS Glue)\*\*: Reads input from S3, performs transformations, and writes back to S3.  
3. \*\*Trigger Python Job (AWS Glue)\*\*: Reads transformed data from S3 and ingests into DynamoDB.  
4. \*\*Archiving\*\*: Moves processed files to an archive folder to prevent redundant processing.

The workflow pipeline is visually represented below:



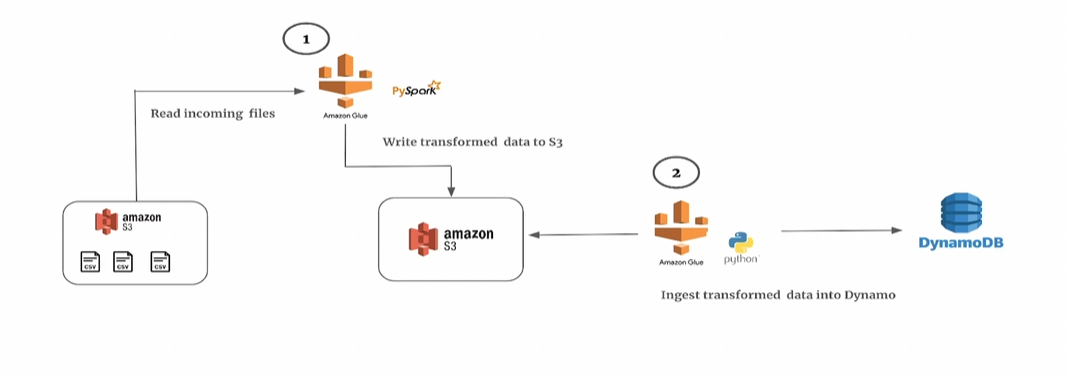
# Technologies Used

- \*\*Amazon S3\*\*: Stores incoming raw data and transformed outputs.  
- \*\*Apache Airflow\*\*: Handles orchestration and task scheduling.  
- \*\*AWS Glue (PySpark & Python Shell)\*\*: Spark handles large-scale transformations; Python job ingests data to DynamoDB.  
- \*\*Amazon DynamoDB\*\*: Serves as a NoSQL database optimized for fast reads/writes.

# Implementation Details

The ETL process is defined in an Airflow DAG that sequentially triggers the necessary transformations. PySpark transformations include aggregating music stream data, computing ranking-based KPIs, and filtering top genre-based records. The Python job then updates DynamoDB with an UPSERT strategy, ensuring data freshness.

A detailed view of the AWS Glue workflow:



# Conclusion & Recommendations

This project demonstrates the efficiency of using AWS services for distributed ETL processing. Future improvements could include:  
- Implementing event-driven triggers instead of scheduled Airflow jobs.  
- Introducing AWS Lambda for lightweight processing where applicable.  
- Enhancing real-time analytics using DynamoDB Streams.