Unit Testing

AWS Automated Data Pipeline

Preparation Date: 05-03-24
Prepared By: Ashutosh Kumar Maurya, Brhat B G

Introduction

This document outlines a structured approach to conducting unit tests on the AWS Automated Data Pipeline project, specifically focusing on AWS EMR and AWS Glue implementations. The document is structured to evaluate the project's iterations, examining their capability to handle multibyte delimited files through various methods, ranging from hardcoded schemas in AWS EMR to dynamic schema discovery using AWS Glue. The aim is to ensure accurate and efficient data processing, adherence to predefined constraints, and optimization of features offered by AWS services.

Test Environment Setup

- AWS EMR cluster configured with necessary roles for S3 access and execution.
- AWS Glue job setup with appropriate IAM roles.
- S3 bucket prepared with test data files corresponding to different testing scenarios.

Test Strategy

- Validating the correct execution of scripts within AWS EMR and AWS Glue.
- Ensuring data integrity through the data processing pipeline.
- Testing for adaptability in schema handling.
- Evaluating the efficiency and scalability of solutions.

AWS EMR Solution

Iteration 1: Hardcoded Schema

TC1: Verify Script Execution

Objective: Ensure the PySpark script executes successfully and processes the CSV files as intended.

Expected Outcome: The script completes execution without errors, and the processed data is correctly saved to the designated S3 bucket.

TC2: Validate Data Integrity and Schema Adherence

Objective: Confirm that the data processed adheres to the hardcoded schema and maintains integrity.

Expected Outcome: Data in the S3 bucket matches the predefined schema with no integrity issues.

Iteration 2: Dynamic Bucket Handling

TC1: Test Dynamic File Retrieval

Objective: Validate that the PySpark script dynamically retrieves and processes files from specified S3 buckets.

Expected Outcome: Files are correctly identified and processed, with data integrity maintained post-merging and deduplication.

Iteration 3: Dynamic Schema Inference

TC1: Evaluate Schema Inference Accuracy

Objective: Verify the script’s capability to correctly infer the schema from input files.

Expected Outcome: The schema is accurately inferred, and data processing reflects this dynamic schema understanding.

AWS Glue Solution

Iteration 1: Hardcoded Schema in AWS Glue Jobs

TC1: AWS Glue Job Execution

Objective: Ensure the AWS Glue job executes as planned, with hardcoded schema applied to data transformation.

Expected Outcome: Job completes successfully, and data transformation aligns with the hardcoded schema.

Iteration 2: Dynamic Schema Discovery

TC1: Verify Schema Discovery and Application

Objective: Confirm AWS Glue Crawlers correctly discover and apply schema to data processing.

Expected Outcome: Schema is dynamically discovered and accurately applied, with data processed accordingly.

Advantages and Drawbacks Analysis

For each iteration, document the observed advantages and potential drawbacks based on the outcomes of the unit tests. This analysis will highlight areas of strength and identify opportunities for improvement.

Conclusion

The unit tests conducted across different iterations of the AWS Automated Data Pipeline project demonstrate the varying degrees of adaptability, efficiency, and reliability in handling data processing tasks. Through rigorous testing, areas for improvement such as enhanced error handling, better schema management, and performance optimization have been identified, offering a clear direction for future enhancements to ensure the robustness and scalability of data pipelines in AWS environments.