Serverless Batch Big Data Processing Pipeline using AWS

Introduction

This document presents a serverless big data processing pipeline implemented using AWS Free Tier services. The project demonstrates how raw transactional data can be stored, processed, and analyzed efficiently without managing servers.

Problem Statement

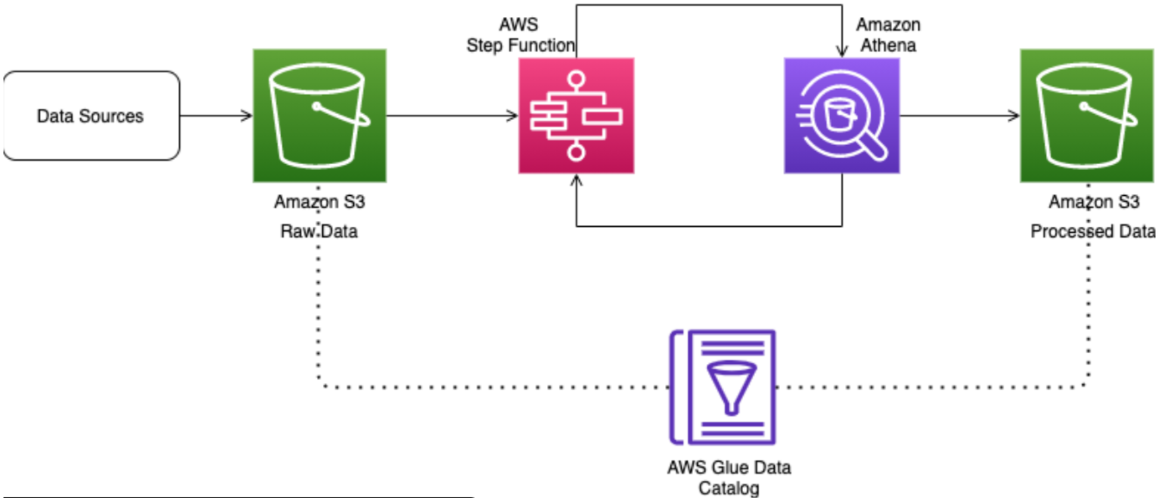
Organizations generate large volumes of raw data that is difficult to analyze directly. Traditional solutions are costly and complex. This project addresses the need for a low-cost, scalable analytics pipeline.

Objectives

- Design a serverless big data pipeline using AWS Free Tier
- Store raw data in Amazon S3
- Process data using AWS Glue
- Convert JSON data to Parquet format
- Analyze data using Amazon Athena

System Architecture

Local Python Script → Amazon S3 (Raw Data) → AWS Glue Crawler → AWS Glue ETL Job → Amazon S3 (Processed Data) → Amazon Athena

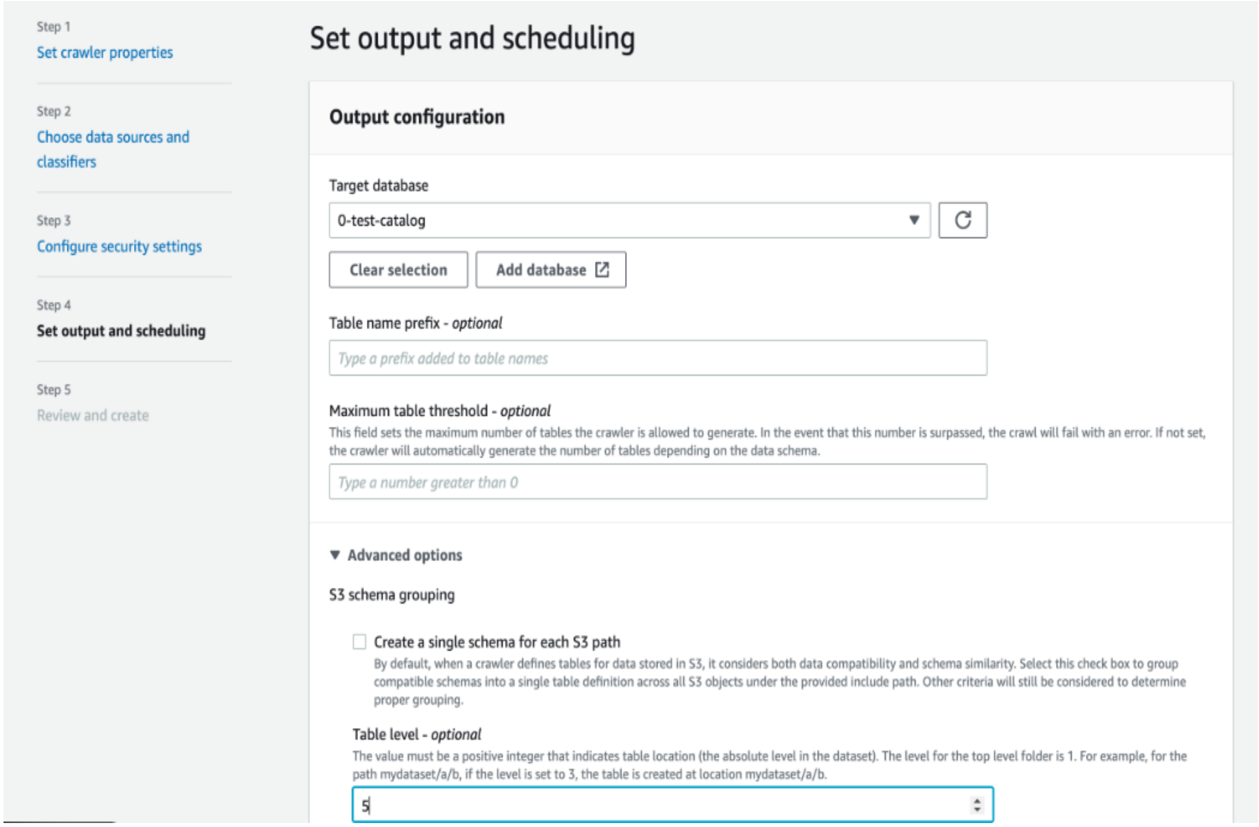


AWS Services Used

Amazon S3 – Data lake storage
AWS Glue – Schema detection and ETL processing
AWS Glue Data Catalog – Metadata management
Amazon Athena – SQL-based analytics
IAM – Security and permissions

Implementation Details

1. Data generated using Python in JSON format
2. Uploaded to S3 raw data bucket
3. Glue Crawler detects schema
4. Glue ETL converts data to Parquet
5. Athena queries processed data



Set output and scheduling

Output configuration

Target database
0-test-catalog

Table name prefix - *optional*
Type a prefix added to table names

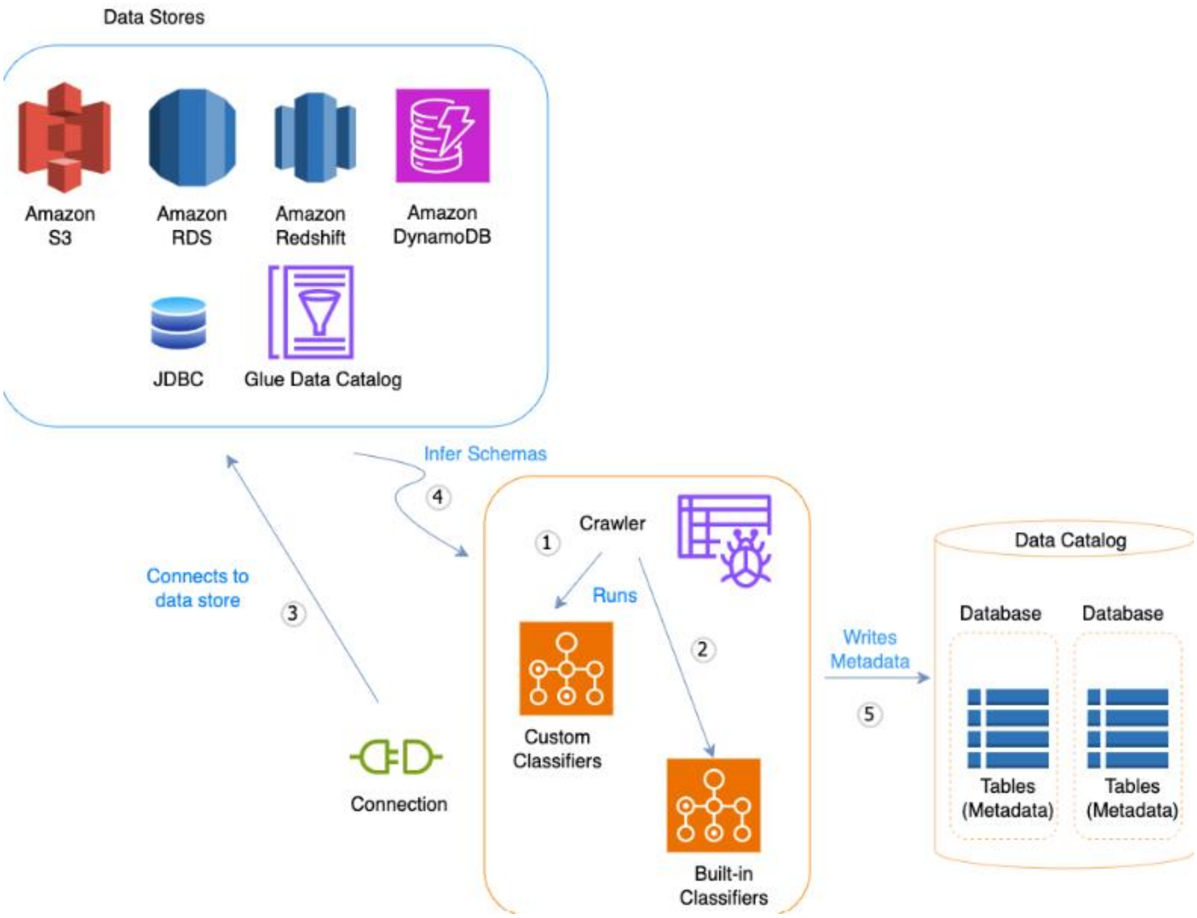
Maximum table threshold - *optional*
This field sets the maximum number of tables the crawler is allowed to generate. In the event that this number is surpassed, the crawl will fail with an error. If not set, the crawler will automatically generate the number of tables depending on the data schema.
Type a number greater than 0

▼ Advanced options

S3 schema grouping

□ Create a single schema for each S3 path
By default, when a crawler defines tables for data stored in S3, it considers both data compatibility and schema similarity. Select this check box to group compatible schemas into a single table definition across all S3 objects under the provided include path. Other criteria will still be considered to determine proper grouping.

Table level - *optional*
The value must be a positive integer that indicates table location (the absolute level in the dataset). The level for the top level folder is 1. For example, for the path mydataset/a/b, if the level is set to 3, the table is created at location mydataset/a/b.
5



Sample Queries

SELECT SUM(price) FROM sales;
SELECT country, SUM(price) FROM sales GROUP BY country;

Conclusion

This project successfully demonstrates a complete serverless big data pipeline using AWS services.