Project-2 : Building a Data Catalog and ETL Pipeline with AWS Glue

Project Description:

In this project, you will create a data catalog and ETL (Extract, Transform, Load) pipeline using AWS Glue. The goal is to automate the process of discovering, cataloging, and transforming data from various sources into a usable format for analysis.

Project Steps:

Data Source Selection:

Choose a few sample datasets from different sources like CSV files, JSON files, and relational databases. These datasets should contain diverse types of data.

Data Catalog Setup:

Set up an AWS Glue Data Catalog. Create databases and tables to organize the datasets. This catalog will store metadata about the data sources.

Crawler Configuration:

Configure AWS Glue Crawlers to automatically discover and catalog the datasets in the Data Catalog. Crawlers analyze the data structure and schema.

ETL Script Development:

Develop ETL scripts using PySpark or AWS Glue ETL jobs to transform the raw data into a format suitable for analysis. Perform data cleaning, filtering, aggregation, and other transformations.

Job Scheduling:

Schedule AWS Glue ETL jobs to run at specific intervals or in response to data updates. This ensures that the data remains up to date in the catalog.

Data Validation:

Implement data validation checks within your ETL scripts. Identify and handle any data quality issues or inconsistencies.

Data Visualization:

Load the transformed data into an Amazon S3 bucket or another data store. You can use visualization tools like Amazon QuickSight or other BI tools to create insightful reports and dashboards.

Monitoring and Error Handling:

Implement monitoring for your ETL jobs. Set up alerts for failures and errors. Implement retry mechanisms to handle transient issues.

**Below are the series of Steps in Detail with screenshots:**

**Step 1: S3 Bucket**

Created two S3 Buckets, one is for Source and other is for destination. And choose one dataset which is in the raw form (i.e., like csv or Json) and upload that file in the Source bucket.

A screenshot of a computer

Description automatically generated

Used a raw file containing Stock details and uploaded in the source bucket as shown in the below picture.

A screenshot of a computer

Description automatically generated

**Step 2: IAM Role**

Created an IAM with appropriate permissions for AWS Glue and S3 Bucket as shown in the below picture.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Step 3: Glue Database**

Created a Glue Database

A screenshot of a computer

Description automatically generated

**Step 4: Glue Crawler**

Created a Glue Crawler using Add tables using Crawler option in Glue Tables.

A screenshot of a computer

Description automatically generated

Once the crawler is created, **run the crawler** and then the csv file that is uploaded in the s3 bucket is shown inside the Glue database as shown in the below figure.  
  
A screenshot of a computer

Description automatically generated

**Step 5: Glue Jobs**

Created an ETL Job to transform the csv file to Parquet format.

Created a job with Visual with a blank Canvas option as shown in the below picture.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Once the job is created you can schedule and run it at required intervals.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Step 6:** We can monitor the job execution using CloudWatch logs if required.

Once the job is successful, please find the transformed parquet file in the destination bucket as shown in the below figure.  
A screenshot of a computer

Description automatically generated

To Visualize the data in Quick Sight, convert the data to csv format and rerun the job again.  
Once the job is successful pull the data to quick sight.

**Step 7:** Visualize the data in Quick Sight.

Create a manifest JSON file with S3 bucket details(parquet/csv transformed file).

Json Code:

{

"entries": [

{

"url": "s3://mounikadestinationbucket/run-1696385401417-part-r-00000",

"meta": {

"description": "Parquet File 1",

"author": "Your Name",

"timestamp": "2023-10-03"

}

}

]

}

A screenshot of a computer

Description automatically generated

Below is the picture of transformed csv file.  
A screenshot of a computer

Description automatically generated

Load the manifest file in QuickSight using S3 Bucket as shown in the below figure.  
A screenshot of a computer

Description automatically generated

Select the columns as required and do Visualisation/Analysys.  
  
A screenshot of a computer

Description automatically generated