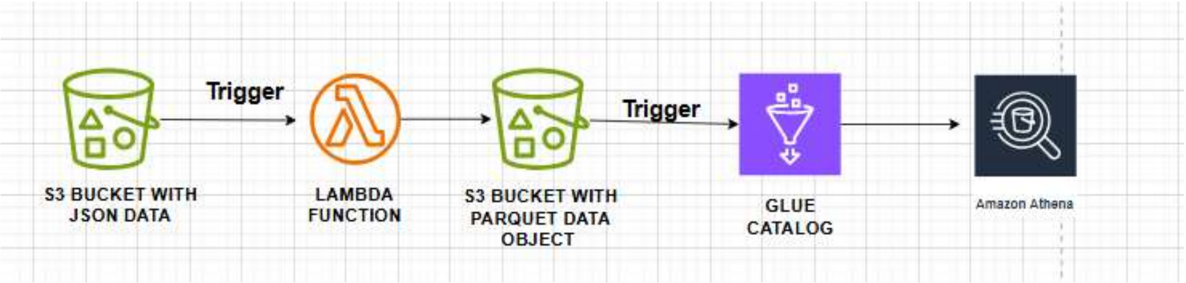
AWS SERVERLESS ETL PIPLELINE

ARCHITECTURE :



This project implements an event-driven serverless data pipeline on AWS to automatically process raw JSON data into a structured format and enable analytical querying without manual intervention.

1. Upload raw JSON data into Amazon S3 bucket.
2. S3 bucket generates an event notification when a new JSON file is uploaded.
3. This event automatically triggers the AWS Lambda function.
4. Lambda reads the uploaded JSON file from the S3 source folder.
5. Lambda flattens the nested JSON data.
6. Flattened JSON is converted into a Pandas DataFrame.
7. DataFrame is transformed into Parquet format.
8. Generated Parquet file is stored in a destination S3 folder.
9. Storage of Parquet file triggers AWS Glue Crawler.Glue Crawler scans the Parquet data.
10. Crawler extracts metadata and infers schema.
11. Metadata is stored in the Glue Data Catalog database.
12. Amazon Athena uses this metadata for querying.
13. Users can run SQL queries directly on processed Parquet data stored in S3.

DETAILED EXPLANATION:

The following AWS services were used to build the serverless data processing pipeline:

1. **Amazon-S3(Simple-Storage-Service)**
   Used to store:
   - Incoming raw JSON data
   - Processed Parquet files
2. **AWS-Lambda**
   Used to:
   - Automatically process uploaded JSON files
   - Flatten nested data
   - Convert JSON data into Parquet format
3. **AWS-Glue-Crawler**
   Used to:
   - Scan processed Parquet files
   - Extract metadata
   - Automatically infer schema
4. **AWS-Glue-Data-Catalog**
   Used to:
   - Store metadata extracted by Glue Crawler
   - Maintain table structure for querying
5. **Amazon-Athena**
   Used to:
   - Perform SQL queries directly on data stored in S3
6. **AWS-IAM-(Identity-and-Access-Management)**
   Used to:
   - Provide necessary permissions to Lambda and Glue services

Step 1: Upload JSON Data into Amazon S3

Raw data in JSON format is uploaded into an Amazon S3 bucket. This bucket acts as the data ingestion layer for storing unstructured input data.

Step 2: S3 Event Trigger Activation

When a new JSON file is uploaded into the S3 bucket (via PUT or POST operation), an event notification is automatically generated.

This event triggers the AWS Lambda function to start the data processing workflow.

Step 3: Data Processing using AWS Lambda

The triggered Lambda function performs the following tasks:

- Reads the uploaded JSON file from S3
- Flattens nested JSON data
- Converts the flattened data into a Pandas DataFrame
- Transforms the DataFrame into Parquet format

Step 4: Store Processed Parquet Data in Destination S3

After transformation:

- The generated Parquet file is renamed dynamically using timestamp
- The file is stored in a separate destination folder inside Amazon S3

This folder contains processed data ready for analysis.

Step 5: Triggering AWS Glue Crawler

Once the Parquet file is stored in the destination S3 bucket, another trigger activates AWS Glue Crawler.

The Glue Crawler:

- Scans the processed Parquet files
- Extracts metadata
- Automatically infers schema

Step 6: Metadata Storage in Glue Data Catalog

The extracted metadata is stored in the Glue Data Catalog in the form of database tables.

This eliminates the need for manually defining schema for querying.

Step 7: Querying using Amazon Athena

Amazon Athena uses the metadata stored in Glue Data Catalog to perform SQL queries directly on the processed Parquet data stored in S3.

Users can now analyze the data without creating any database infrastructure.