AWS Assignment - 1 (Implementation)

- **Objective:** Implement a batch data processing pipeline in AWS to process daily bank transactions stored in JSON files.

- **Problem Statement:**
  Your bank receives daily transaction data from its branches. The data arrives once daily in a JSON file. Your goal is to set up an AWS data processing pipeline that automatically processes this data as soon as it lands on S3, transforming and storing it for querying.

Requirements:

- **Data Source:**
  - A daily JSON file containing bank transactions will be dropped into an S3 bucket.
  - You are required to write a mock script to generate and simulate this data drop.

- **Processing Trigger:**
  - As soon as a file arrives in the source S3 bucket, your data processing should be triggered automatically.

- **Data Transformation:**
  - Use AWS Glue to read the JSON file from the S3 bucket.
  - Implement transformations such as filtering out any transactions with null values, and deduplicating any repeated transactions based on ***transaction_ID***.

    - Convert the JSON format into a columnar format (e.g., Parquet) which is optimized for querying.
    - Store the transformed data back into a separate S3 bucket or prefix.

- **Incremental Processing:**
    - Your solution should ensure that only new data is processed each day, avoiding re-processing of older data.

- **Data Quality Checks:**
    - Before transformation, ensure that all transaction records have a valid transaction ID, date, and amount.
    - Post transformation, verify that the number of records processed matches the number in the source file.
    - Alert if any file is missing on a given day.

- **Querying:**
    - Set up AWS Athena to query the processed data stored in S3.

- **Monitoring and Notification:**
    - Use AWS CloudWatch to monitor the data processing tasks.
    - Set up SNS notifications for any failures in the pipeline or if any data quality checks fail.

- **Sample JSON record**

{
"transaction_id":"T123456",
"account_id":"A12345",
"transaction_date":"2023-09-10",

```
    "amount":500.00,
    "transaction_type":"debit",
    "branch_id":"B1",
    "description":"ATM Withdrawal"
}
```

- **Steps:**
    - Set up an S3 bucket to act as your data source.
    - Write a mock script to generate the above sample JSON data and store it in the S3 bucket daily.
    - Set up AWS Lambda to be triggered as soon as a new file arrives in the S3 bucket.
    - Within Lambda, invoke the Glue job for data transformation.
    - Design the Glue job to transform and cleanse the data. This will involve using Glue Crawlers to infer the schema, Glue ETL jobs to transform, and writing the results back to S3 in Parquet format.
    - Use AWS Athena to set up a table on top of the transformed data in S3.
    - Set up CloudWatch monitors and alarms, linked to SNS, to notify of any issues in the pipeline.