

Bank Transactions Event-Driven Data Processing with AWS

1. This project was about implementing an event-driven data processing pipeline in AWS to process daily bank transactions stored in JSON files. Goal was 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.
2. A daily JSON file containing bank transactions was being dropped into an S3 bucket. Once new file uploaded, data processing triggered automatically with AWS Lambda function.

```
[
  {
    "transaction_id": "T123456",
    "account_id": "A18796",
    "transaction_date": "2023-09-20",
    "amount": 217.86,
    "transaction_type": "debit",
    "branch_id": "B3",
    "description": "ATM Withdrwal"
  }
]
```

Amazon S3 > Buckets > bank-transactions-gds > input/

input/ Copy S3 URI

Objects Properties

Objects (3) Info Refresh Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	bank_transactions_new.json	json	May 26, 2024, 13:47:33 (UTC+05:30)	9.1 KB	Standard
<input type="checkbox"/>	bank_transactions_neww.json	json	May 26, 2024, 13:57:06 (UTC+05:30)	9.1 KB	Standard
<input type="checkbox"/>	bank_transactions.json	json	May 26, 2024, 13:43:36 (UTC+05:30)	9.1 KB	Standard

3. Within Lambda, I invoked the Glue job. Following step by step starting, used AWS Glue to read the JSON file from the S3 bucket

Successfully updated the function `s3_to_lambda_trigger`.

```
1 import json
2 import boto3
3 import pandas as pd
4
5 s3_client = boto3.client('s3')
6 glueclient = boto3.client('glue')
7
8 sns_client = boto3.client('sns')
9 sns_arn = 'arn:aws:sns:us-east-1:339713857891:dq_checks_sns'
10
11 def lambda_handler(event, context):
12     gluejobname="s3i_to_s3o_etl"
13     # print('Event is {}'.format(event))
14     # TODO implement
15     try:
16         bucket_name = event['Records'][0]['s3']['bucket']['name']
17         s3_file_key = event['Records'][0]['s3']['object']['key']
18         print('Bucket Name is {}'.format(bucket_name))
19         print('Bucket key is {}'.format(s3_file_key))
20         res = s3_client.get_object(Bucket=bucket_name, Key=s3_file_key)
21
22         df_s3_data = pd.read_json(res['Body'])
23         print(df_s3_data)
24
25         # trigger etl job
26         runId = glueclient.start_job_run(JobName=gluejobname)
27         status = glueclient.get_job_run(JobName=gluejobname, RunId=runId['JobRunId'])
28         print("Job Status : ", status['JobRun']['JobRunState'])
29
30         message = 'Input S3 file has been processed successfully !!'
31         print(message)
32         response = sns_client.publish(Subject="SUCCESS - Daily Data Processing", TargetArn=sns_arn, Message=message, MessageStructure='text')
33     except Exception as err:
34         print(err)
35         message = 'Input S3 file processing failed !!'
36         print(message)
37         response = sns_client.publish(Subject="FAILED - Daily Data Processing", TargetArn=sns_arn, Message=message, MessageStructure='text')
```

4. Implemented transformations such as filtering out any transactions with null values, and deduplicating any repeated transactions based on transaction_ID. Then converted the JSON format into a columnar format which is considered more optimized for querying

`s3i_to_s3o_etl` Last modified on 5/25/2024, 7:34:47 PM Actions Save Run

Visual Script Job details Runs Data quality - updated Schedules Version Control

Transform

S3 Source Glue DC X
Catalog - DataSource

Associate an alias with each input source [Info](#)
Edit the aliases used for the inputs to this node.

Input sources SQL aliases

S3 Source Glue DC s3_bt_source

SQL query
Enter a SQL statement to add to your job.

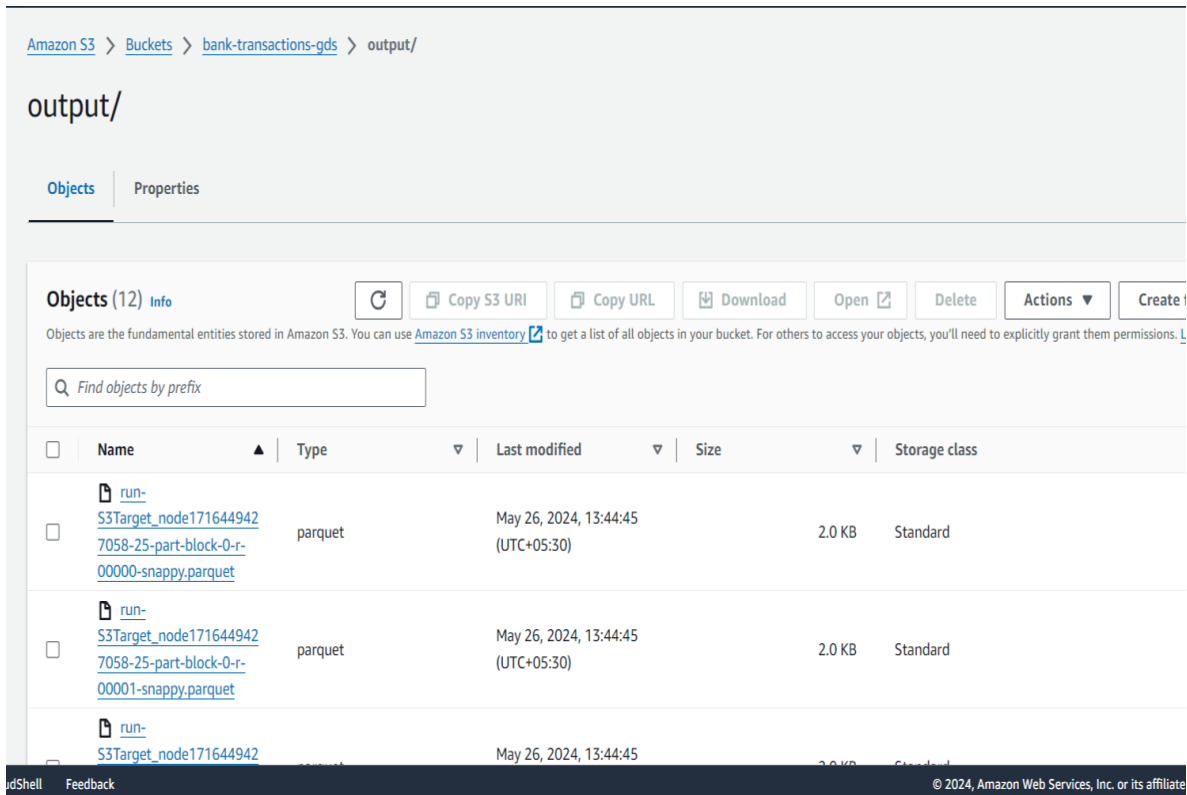
```
1 with cte as (
2 select *,
3 row_number() over(partition by transaction_id order by tra
4 count(transaction_id) over(partition by transaction_id) as
5 from s3_bt_source
6 where transaction_id is not null
7 )
8 select * from cte where t_rank=1
```

Data preview (100) Info READY End session Previewing 9 of 9 fields

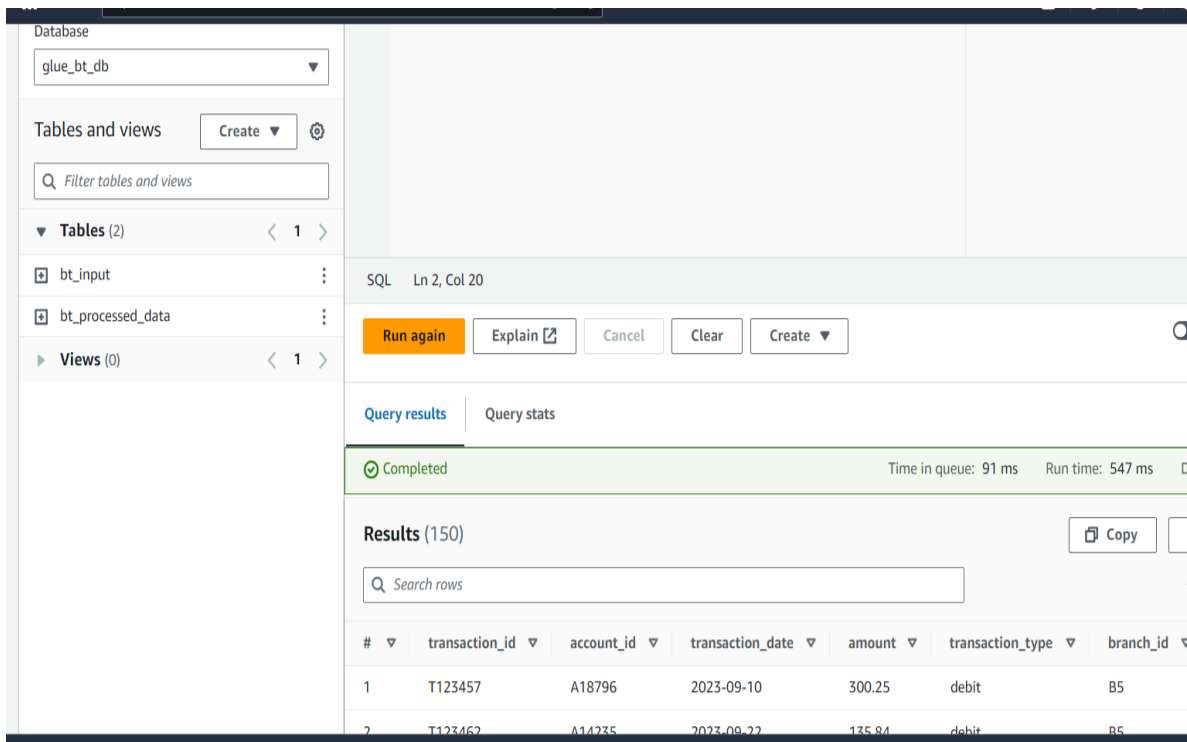
Filter sample dataset

transaction_id	account_id	transaction_date	amount	transaction_type
T123458	A69781	2023-09-22	250.23	debit
T123465	A14235	2023-09-20	439.63	debit

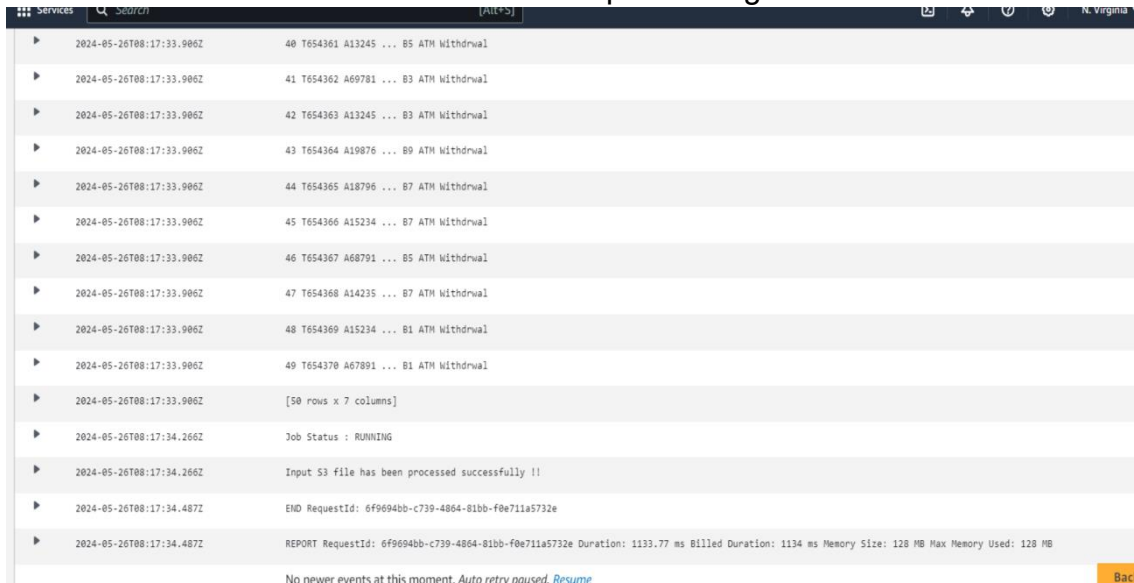
5. Further storing the transformed data back into a separate S3 bucket in parquet form. Also enabled the etl job bookmark to get only new or updated data.



6. At last I set up AWS Athena to set up a table and query on the top of transformed data stored in S3.

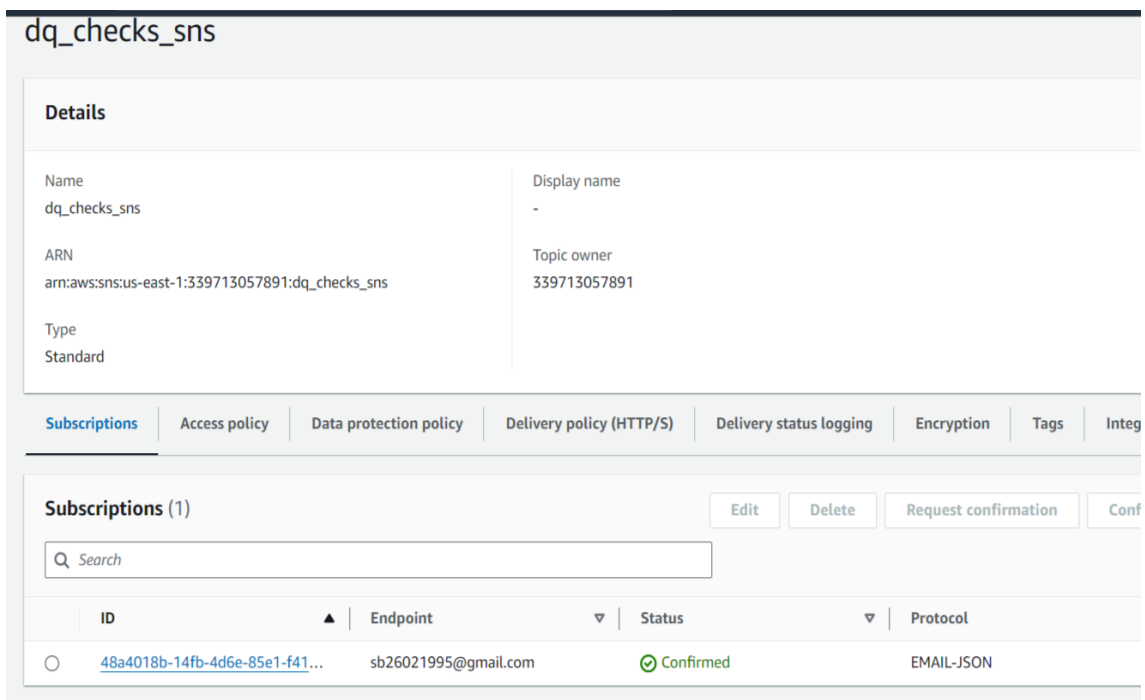


7. Used aws cloudWatch to monitor the data processing tasks



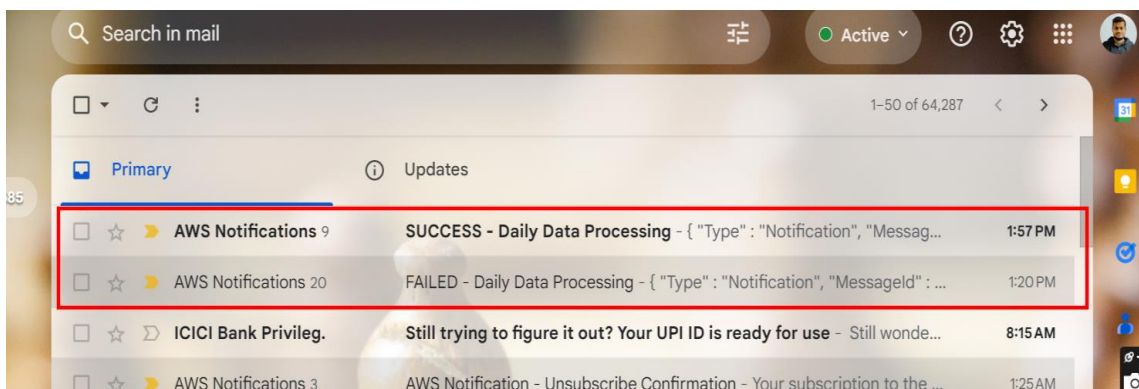
Timestamp	Event ID	Description
2024-05-26T08:17:33.906Z	40 T654361 A13245 ... B5	ATH Withdrwal
2024-05-26T08:17:33.906Z	41 T654362 A69781 ... B3	ATH Withdrwal
2024-05-26T08:17:33.906Z	42 T654363 A13245 ... B3	ATH Withdrwal
2024-05-26T08:17:33.906Z	43 T654364 A19876 ... B9	ATH Withdrwal
2024-05-26T08:17:33.906Z	44 T654365 A18796 ... B7	ATH Withdrwal
2024-05-26T08:17:33.906Z	45 T654366 A15234 ... B7	ATH Withdrwal
2024-05-26T08:17:33.906Z	46 T654367 A68791 ... B5	ATH Withdrwal
2024-05-26T08:17:33.906Z	47 T654368 A14235 ... B7	ATH Withdrwal
2024-05-26T08:17:33.906Z	48 T654369 A15234 ... B1	ATH Withdrwal
2024-05-26T08:17:33.906Z	49 T654370 A67891 ... B1	ATH Withdrwal
2024-05-26T08:17:33.906Z	[50 rows x 7 columns]	
2024-05-26T08:17:34.266Z	Job Status :	RUNNING
2024-05-26T08:17:34.266Z	Input S3 file has been processed successfully !!	
2024-05-26T08:17:34.487Z	END RequestId: 6f9694bb-c739-4864-81bb-f0e711a5732e	
2024-05-26T08:17:34.487Z	REPORT RequestId: 6f9694bb-c739-4864-81bb-f0e711a5732e	Duration: 1133.77 ms Billed Duration: 1134 ms Memory Size: 128 MB Max Memory Used: 128 MB

8. set up SNS notifications for any failures in the pipeline or if any data quality checks fail



dq_checks_sns			
Details			
Name	dq_checks_sns		
ARN	arn:aws:sns:us-east-1:339713057891:dq_checks_sns		
Type	Standard		
Display name	-		
Topic owner	339713057891		

Subscriptions (1)			
ID	Endpoint	Status	Protocol
48a4018b-14fb-4d6e-85e1-f41...	sb26021995@gmail.com	Confirmed	EMAIL-JSON



From	Subject	Time
AWS Notifications 9	SUCCESS - Daily Data Processing - { "Type": "Notification", "Messag...	1:57 PM
AWS Notifications 20	FAILED - Daily Data Processing - { "Type": "Notification", "Messageld": ...	1:20 PM
ICICI Bank Privileg.	Still trying to figure it out? Your UPI ID is ready for use - Still wonde...	8:15 AM
AWS Notifications 3	AWS Notification - Unsubscribe Confirmation - Your subscription to the ...	1:25 AM