

Optimized Batch & Stream Data Processing for Enhanced Inventory Management

Submitted in partial fulfilment of the requirements for the degree of

Post Graduate Diploma in Data Engineering

by

Sudarshan P (G23AI1046)

Utkarsh Gupta (G23AI1048)

Under the guidance of

Dr. Pradip Samal

IIT Jodhpur

**Indian Institute of Technology Jodhpur
Advance Data Engineering in Cloud
Trimester-3 (July 2024)**



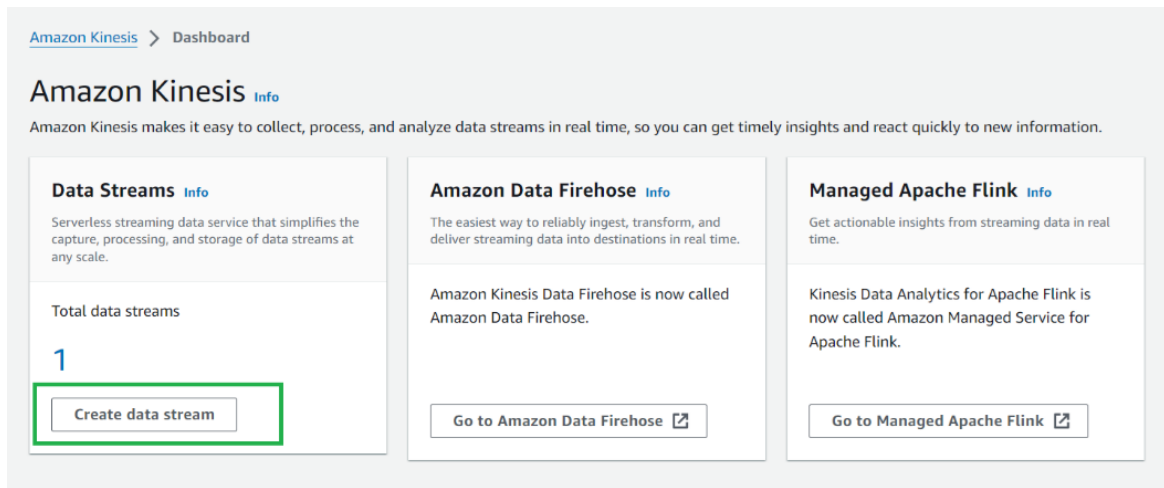
॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Assignment – 2

Part 1: Kinesis Setup

1. Create a Kinesis Data Stream:

- Open the AWS Management Console.
- Navigate to Kinesis > Data Streams.
- Click Create data stream.
- Enter the stream name (e.g., OnlineRetailDataStream).
- Specify the number of shards based on your expected data volume.
- Click Create stream.



2. Enter Details

- a. Give stream name
- b. Choose On-demand or provisioned as per requirement (I have chosen on-demand here)
- c. Click on Create data stream

Create data stream [Info](#)

Data stream configuration

Data stream name

online-retail-stream

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens and periods.

Data stream capacity [Info](#)

Capacity mode

☒ On-demand

Use this mode when your data stream's throughput requirements are unpredictable and variable. With on-demand mode, your data stream's capacity scales automatically.

☐ Provisioned

Use provisioned mode when you can reliably estimate throughput requirements of your data stream. With provisioned mode, your data stream's capacity is fixed.

3. Check the status if it is Active

[Amazon Kinesis](#) > [Data streams](#) > OnlineRetailDataStream

OnlineRetailDataStream [Info](#)

Delete

Data stream summary

Status	Capacity mode	ARN	Creation time
Active	On-demand	arn:aws:kinesis:us-east-1:992382534203:stream/OnlineRetailDataStream	July 08, 2024 at 12:27 GMT+5:30
	Data retention period		
	1 day		

Part 2: Glue Setup to send data to Kinesis

1. Create Python Shell Glue job to create glue catalog for Retail data that we will stream

create_glue_catalog_retail

Last modified on

Script

Job details

Runs

Data quality

Schedules

Version Control

Script

Info

```
1 import boto3
2
3 # Initialize Glue client
4 glue_client = boto3.client('glue')
5
6 # Define parameters
7 database_name = 'online_retail_db'
8 table_name = 'online_retail_test'
9 s3_path = 's3://ade-project/dataset/Online_Retail_test.csv'
10
11 # Define table schema
12 table_input = {
13     'Name': table_name,
14     'StorageDescriptor': {
15         'Columns': [
16             {'Name': 'InvoiceNo', 'Type': 'string'},
17             {'Name': 'StockCode', 'Type': 'string'},
18             {'Name': 'Description', 'Type': 'string'},
19             {'Name': 'Quantity', 'Type': 'int'},
20             {'Name': 'InvoiceDate', 'Type': 'string'},
21             {'Name': 'UnitPrice', 'Type': 'float'},
22             {'Name': 'CustomerID', 'Type': 'string'},
23             {'Name': 'Country', 'Type': 'string'}
```

2. Also, have the dataset in an s3 location where you want to stream the data from
3. Once the job succeeds, check the glue catalog

create_glue_catalog_retail

Last modified on 7/8/2024, 1:08:42 PM

Actions

Save

Run

Script

Job details

Runs

Data quality

Schedules

Version Control

Job runs (1/3)

Info

Last updated (UTC)
July 13, 2024 at 16:24:25

View details

Stop job run

Table View

Card View

Filter job runs by property

< 1 >

⚙

	Run status	Retries	Start time (Local)	End time (Local)	Duration	Capacity...	Worker type
●	✓ Succeeded	0	07/08/2024 13:08:44	07/08/2024 13:09:04	12 s	0.0625 DPU	-

AWS Glue

Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)

Data Catalog
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings

online_retail_db

Last updated (UTC)
July 13, 2024 at 16:25:06

Database properties

Name	Description	Location	Created on (UTC)
online_retail_db	-	-	July 8, 2024 at 07:11:08

Tables (3)

Last updated (UTC)
July 13, 2024 at 16:25:07

View and manage all available tables.

Filter tables

Name	Database	Location	Classification	Deprecated	View data	Data quality
online_retail	online_retail_db	s3://ade-project/tra	-	-	Table data	View data quality
online_retail_test	online_retail_db	s3://ade-project/da	-	-	Table data	View data quality
transformed_data	online_retail_db	s3://ade-project/tra	CSV	-	Table data	View data quality

4. Next Create a Python Shell Glue to send the data to kinesis stream
 - a. Use boto3 Client to connect to Kinesis Stream
 - b. Give s3 location to send the data from

OnlineRetailToKinesisPython

Last modified on 7/8/2024, 6:28:04 PM

Actions Save Run

Script Job details Runs Data quality Schedules Version Control

Script Info

```

1 import boto3
2 import json
3 import csv
4
5 # Initialize boto3 clients
6 s3_client = boto3.client('s3')
7 kinesis_client = boto3.client('kinesis', region_name='us-east-1') # Update region as needed
8
9 # Parameters
10 s3_bucket = "ade-project"
11 s3_key = "dataset/Online_Retail.csv"
12 kinesis_stream_name = "OnlineRetailDataStream"
13
14 # Function to process CSV file from S3 and send records to Kinesis
15 def process_csv_and_send_to_kinesis(s3_bucket, s3_key, kinesis_stream_name):
16     # Download CSV file from S3
17     obj = s3_client.get_object(Bucket=s3_bucket, Key=s3_key)
18     csv_data = obj['Body'].read().decode('utf-8').splitlines()

```

5. Once the job succeeds, proceed to create Data Firehose to get that data

OnlineRetailToKinesisPython

Last modified on 7/8/2024, 6:28:04 PM

Actions Save Run

Script Job details **Runs** Data quality Schedules Version Control

Job runs (1/10) Info

Last updated (UTC)
July 13, 2024 at 16:28:17

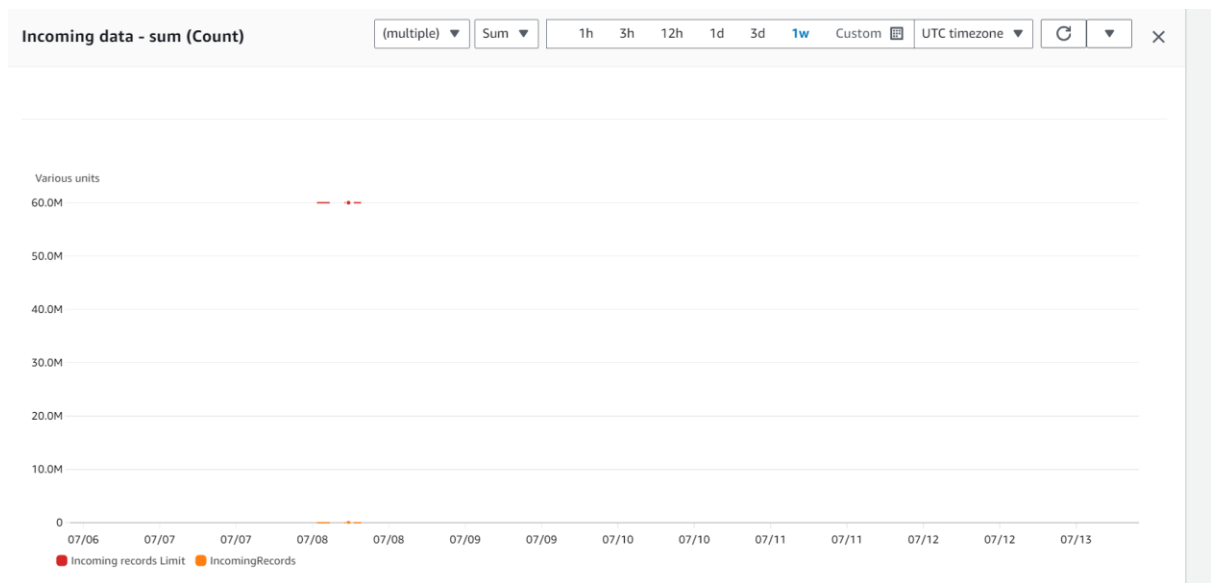
View details Stop job run

Table View Card View

Filter job runs by property

Run status	Retries	Start time (Local)	End time (Local)	Duration	Capacity...	Worker type
Succeeded	0	07/08/2024 18:28:07	07/08/2024 19:39:47	1 h 11 m 31 s	0.0625 DPUs	-

- Also, you can monitor the data has arrived in monitoring tab of kinesis



Part 3: Setup Data Firehose

- Click on Create Firehose stream
- Choose Source and Destination

[Amazon Data Firehose](#) > [Firehose streams](#) > Create Firehose stream

Create Firehose stream [Info](#)

► Amazon Data Firehose: How it works

Choose source and destination
Specify the source and the destination for your Firehose stream. You cannot change the source and destination of your Firehose stream once it has been created.

Source [Info](#)
Amazon Kinesis Data Streams ▼

Destination [Info](#)
Amazon S3 ▼

3. Give source stream

Source settings

Kinesis data stream

[Browse](#)

[Create](#)

Format: arn:aws:kinesis:[Region]:[AccountId]:stream/[StreamName]

4. Give the destination settings

Destination settings [Info](#)

Specify the destination settings for your Firehose stream.

S3 bucket

[Browse](#)

[Create](#)

Format: s3://bucket

New line delimiter

You can configure your Firehose stream to add a new line delimiter between records in objects that are delivered to Amazon S3.

☐ Not enabled

☒ Enabled

Dynamic partitioning [Info](#)

Dynamic partitioning enables you to create targeted data sets by partitioning streaming S3 data based on partitioning keys. You can partition your source data with inline parsing and/or the specified AWS Lambda function. You can enable dynamic partitioning only when you create a new Firehose stream. You cannot enable dynamic partitioning for an existing Firehose stream. Enabling dynamic partitioning incurs additional costs per GiB of partitioned data. For more information, see [Amazon Data Firehose pricing](#).

☒ Not enabled

☐ Enabled

i You can enable dynamic partitioning only when you create a new Firehose stream. You cannot enable dynamic partitioning for an existing Firehose stream.

S3 bucket prefix - *optional*

By default, Amazon Data Firehose appends the prefix "YYYY/MM/dd/HH" (in UTC) to the data it delivers to Amazon S3. You can override this default by specifying a custom prefix that includes expressions that are evaluated at runtime.

stream_data/

You can repeat the same keys in your S3 bucket prefix. Maximum S3 bucket prefix characters: 1024.

S3 bucket error output prefix - *optional*

You can specify an S3 bucket error output prefix to be used in error conditions. This prefix can include expressions for Amazon Data Firehose to evaluate at runtime.

stream_data/error/

S3 bucket and S3 error output prefix time zone [Info](#)

Choose a time zone that you want to use for date and time in S3 prefixes

UTC

5. Provide proper IAM role to access S3 and Kinesis Data Stream

Service access [Info](#)

Edit

Amazon Data Firehose uses this IAM role for all the permissions that the Firehose stream needs. To specify different roles for the different permissions, use the API or the CLI.

IAM role

[KinesisFirehoseServiceRole-KDS-S3-DfiuG-us-east-1-1720425224099](#)

6. Once the setup is done and Firehose is active, whenever data comes in Kinesis Data Stream Firehose will put the data to s3 destination location in part files

[Amazon Data Firehose](#) > [Firehose streams](#) > KDS-S3-DfiuG

KDS-S3-DfiuG [Info](#)


Delete Firehose stream

Firehose stream details

Status
✔ Active

Source
Amazon Kinesis Data Streams

Destination
Amazon S3

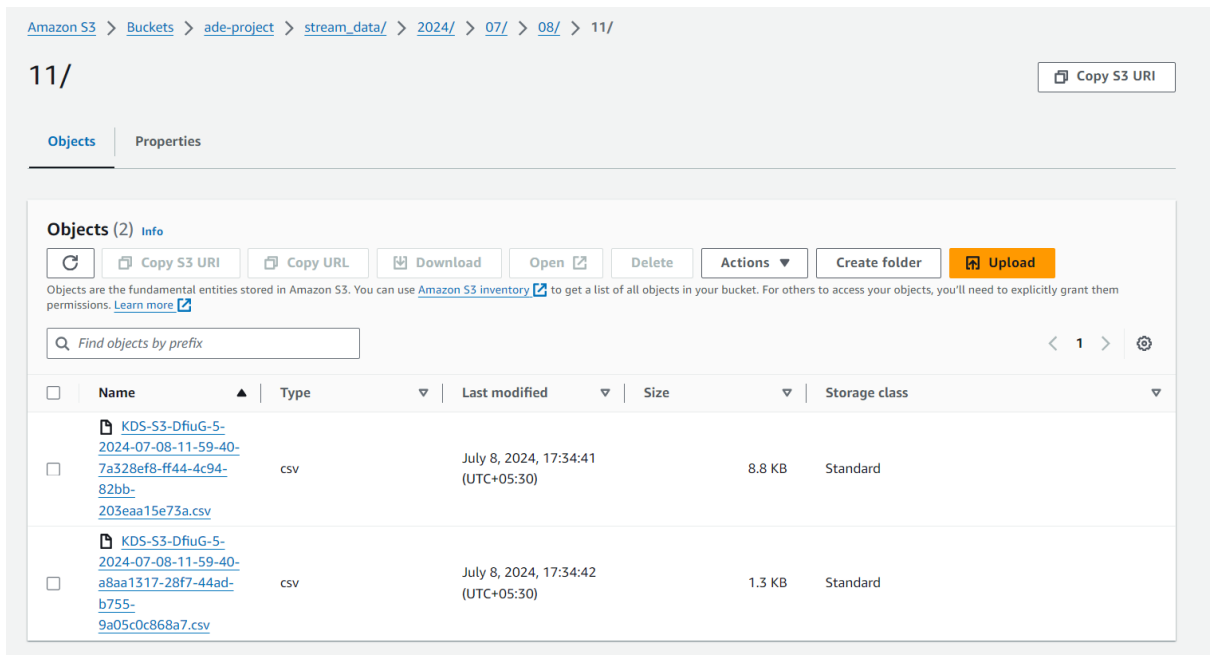
ARN
 `arn:aws:firehose:us-east-1:992382534203:deliverystream/KDS-S3-DfiuG`

Data transformation
Not enabled

Dynamic partitioning
Not enabled

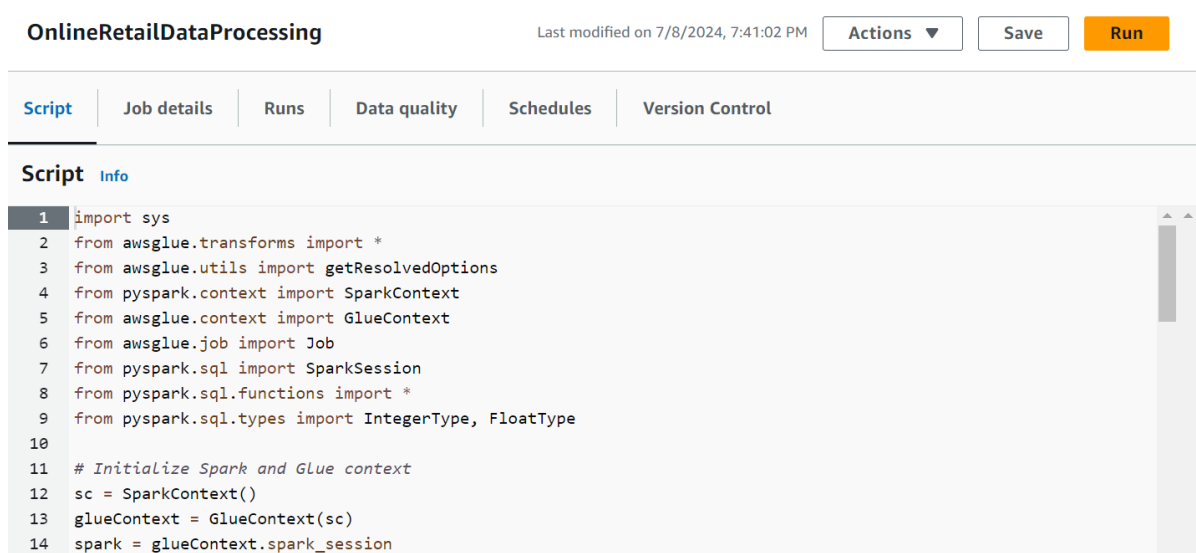
Creation time
July 08, 2024 at 13:26 GMT+5:30

Error logs status
✔ 0 Destination error logs



Part 4: Glue Job to for transform Stream data into Cleansed data

1. Create Spark Glue Job



2. Store the data in an S3 location

3. Check if Job succeeded

OnlineRetailDataProcessing

Last modified on 7/8/2024, 7:41:02 PM

Actions

Save

Run

Script

Job details

Runs

Data quality

Schedules

Version Control

Job runs (1/47) Info

Last updated (UTC)
July 13, 2024 at 16:54:47

View details

Stop job run

Table View

Card View

Filter job runs by property

< 1 2 3 >

⚙

	Run status	Retries	Start time (Local)	End time (Local)	Duration	Capacit...
<input checked="" type="radio"/>	✓ Succeeded	0	07/08/2024 19:43:21	07/08/2024 19:45:12	1 m 43 s	10 DPUs

4. Check s3 for transformed data

Amazon S3

>

Buckets

>

ade-project

>

transformed_data/

transformed_data/

Copy S3 URI

Objects

Properties

Objects (36) 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

< 1 >

⚙

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	 part-00000-6c58300f-b011-471a-ba4d-eb6252e679d2-c000.csv	csv	July 8, 2024, 19:44:59 (UTC+05:30)	1.5 MB	Standard