

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 ॥

End-to-End Project

https://github.com/utkarshgupta98/advance_data_engineering

Problem Statement

An e-commerce company aims to enhance inventory management and supply chain efficiency through analysis of batched sales data from multiple online platforms. The objective is to optimize inventory levels, minimize stockouts, and streamline fulfillment processes.

Real-Life Use Case: Batch Data Processing for Inventory Management

Scenario:

- **Online Sales Platforms:** Transactional data including orders, product details, and customer information.
- **Inventory Systems:** Stock levels, warehouse locations, and logistics data.
- **Supplier Information:** Lead times, pricing, and availability.

Challenges:

1. Data Integration:

- Integrating batch data from diverse online platforms and internal systems.
- Ensuring data consistency and accuracy across different sources.

2. Inventory Management:

- Providing insights into inventory levels and sales trends through batch processing.
- Predicting demand fluctuations and adjusting inventory accordingly.

3. Scalability:

- Handling large volumes of batched transactional data from global online platforms.
- Scaling to accommodate peak shopping periods and seasonal demand variations.

4. **Operational Efficiency:**

- Optimizing procurement and fulfillment processes based on batched data insights.
- Minimizing storage costs and reducing excess inventory.

Solution:

Using AWS Services to Build a Batch Data Inventory Management System:

1. **Data Ingestion:**

- **Amazon S3:** Upload batch data files (CSV, JSON) containing sales, inventory, and supplier information.
- **AWS Kinesis Data Streams or AWS Direct Connect:** Implement the data ingestion mechanism to stream data from a source to Amazon S3.

2. **Data Processing:**

- **AWS Glue:** Develop and test the data processing pipeline using Apache Spark.
 - Apply data transformation and cleansing techniques to prepare the data for aggregation and analysis.
 - Implement data partitioning and indexing strategies to optimize query performance.
 - Update the GitHub repository with the code and configuration files for data ingestion and processing.

3. **Data Storage:**

- **Amazon RDS (Relational Database Service):**
 - Set up an RDS instance (e.g., MySQL, PostgreSQL) to store structured transactional data:

- Sales transactions.
- Product details.
- Inventory levels.
- Supplier information.

4. **Analytics and Reporting:**

- **Amazon Athena:**
 - Query data directly in Amazon S3 for ad-hoc analysis and reporting.
- **Quicksight**
 - Visualize data using Amazon QuickSight for interactive dashboards.

5. **Security and Access Control:**

- **AWS IAM:** Manage access to RDS and other AWS services:
 - Ensure data privacy and compliance with regulatory requirements.

End-to-End Data Engineering Platform:

1. **Integration:**

- Integrate the components developed to create a complete end-to-end data engineering platform.

2. **Data Management:**

- Implement data retention policies and configure data lifecycle management in Amazon S3.

3. **Security Best Practices:**

- Enable encryption for data at rest and in transit.
- Set up proper access controls using AWS IAM for authentication and authorization.

4. **Cost Optimization:**

- Utilize cost-effective storage options like Amazon S3 Glacier for long-term data retention.
- Monitor and optimize resource utilization to reduce costs.

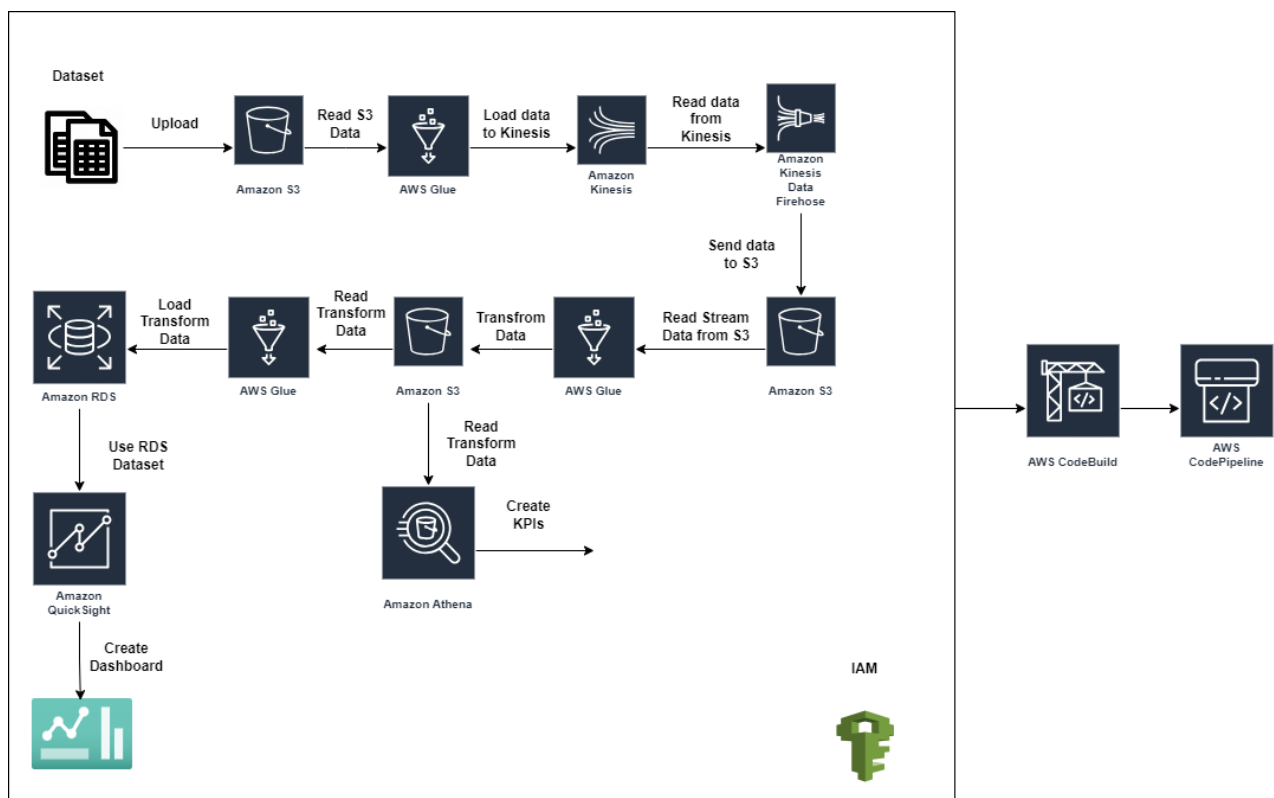
5. Testing and Validation:

- Conduct thorough testing of the platform, including data ingestion, processing, aggregation, and visualization, to ensure data integrity and performance.

6. CI/CD Automation:

- Implement continuous integration and continuous deployment (CI/CD) processes using AWS CodePipeline and AWS CodeBuild to automate deployment and updates.

Architecture Diagram:



Key Performance Indicators (KPIs):

- **Inventory Turnover Ratio:** measures how efficiently inventory is managed by indicating how many times inventory is sold and replaced over a period.

- **Stockout Rate:** measures the percentage of time products are out of stock.
- **COGS to Revenue Ratio:** measures the efficiency of managing inventory costs relative to revenue.
- **Customer Acquisition Cost (CAC):** measures the average cost of acquiring a new customer.
- **Returning Customers:** The total number of unique customers who made purchases on July 11, 2011.
- **Average Order Value (AOV):** Measures the average amount spent per order over the course of the year 2011.
- **Gross Margin:** Measures the profitability of products by comparing the revenue (InvoiceTotal) to the cost of goods sold (Quantity * UnitPrice).
- **COGS to Revenue Ratio:** Efficiency in managing inventory costs relative to revenue.

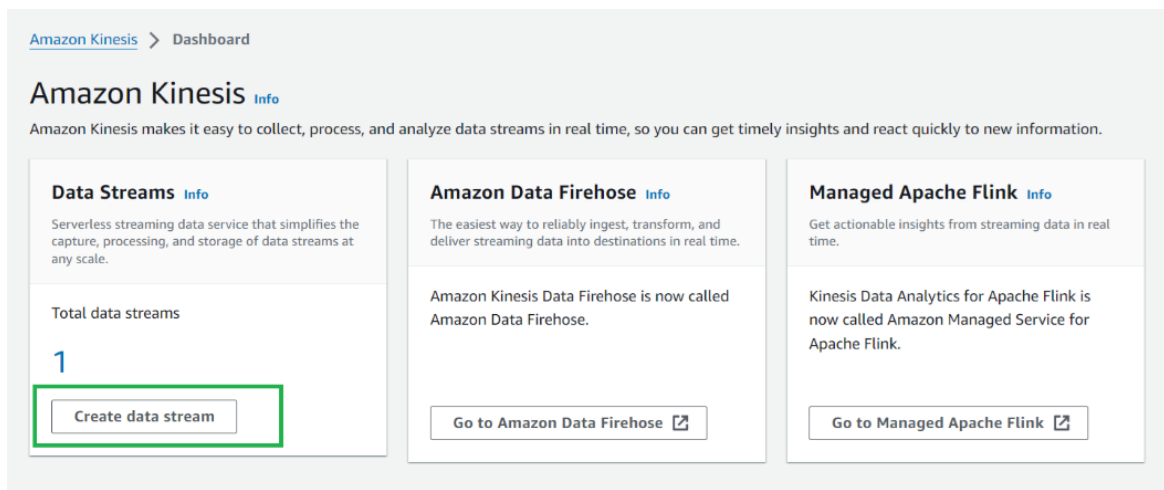
Benefits:

- **Improved Inventory Management:** Enhanced control over inventory levels and reduced stockouts.
- **Efficient Supply Chain Operations:** Optimized procurement and fulfillment processes.
- **Scalability:** AWS services scale with business growth.
- **Cost Efficiency:** Managed services minimize infrastructure costs.
- **Security and Compliance:** Secure handling of data ensures compliance with regulations.

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

	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

<input type="checkbox"/>	Name	Database	Location	Classification	Deprecated	View data	Data quality
<input type="checkbox"/>	online_retail	online_retail_db	s3://ade-project/tra	-	-	Table data	View data quality
<input type="checkbox"/>	online_retail_test	online_retail_db	s3://ade-project/da	-	-	Table data	View data quality
<input type="checkbox"/>	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

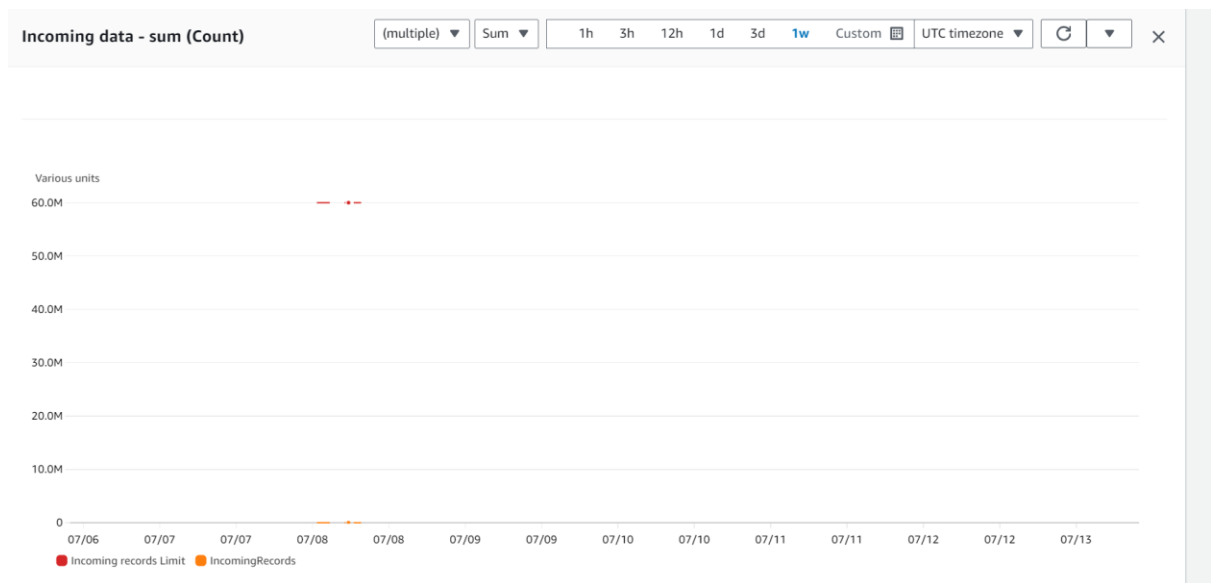
View details Stop job run

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

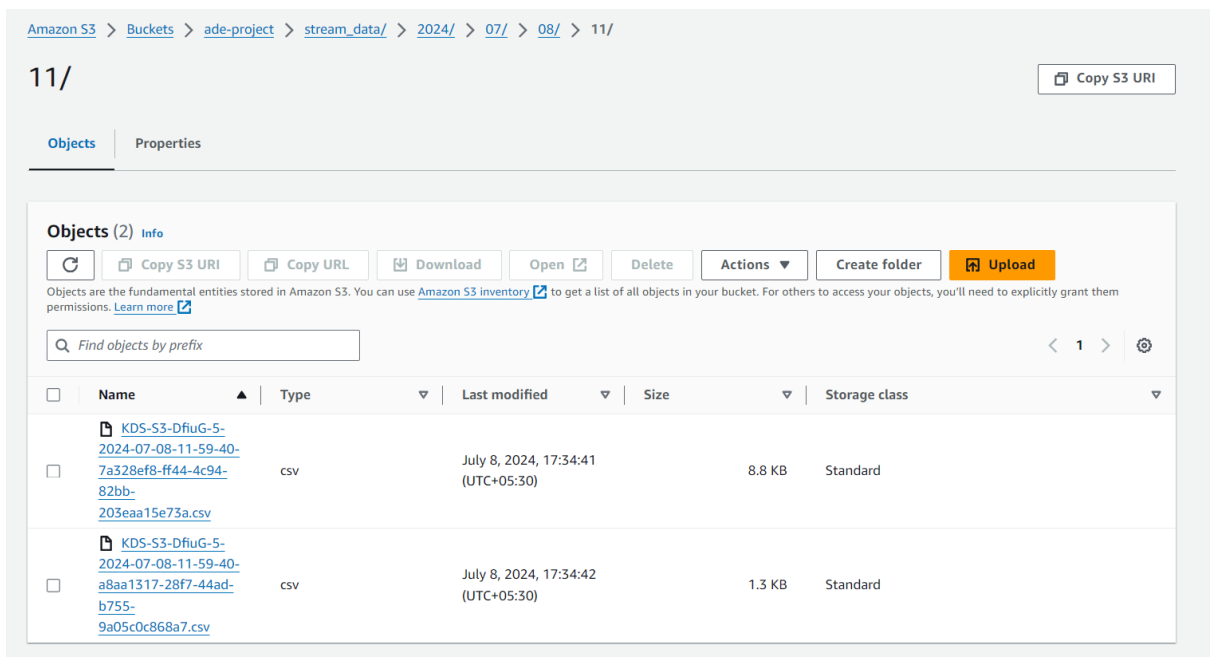
ARN
 `arn:aws:firehose:us-east-1:992382534203:deliverystream/KDS-S3-DfiuG`

Data transformation
Not enabled

Dynamic partitioning
Not enabled

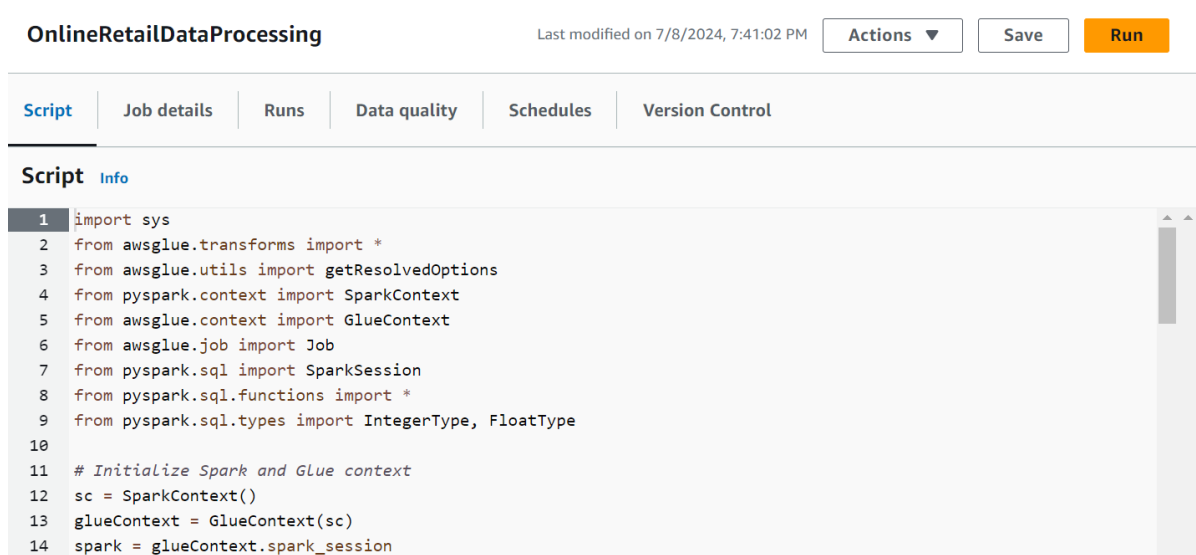
Creation time
July 08, 2024 at 13:26 GMT+5:30

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

Part 5: Create RDS instance

1. Go to RDS console and Create DB instance

[RDS](#) > Create database

Create database

Choose a database creation method [Info](#)


☒ **Standard create**
You set all of the configuration options, including ones for availability, security, backups, and maintenance.


☐ **Easy create**
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.


2. Choose MySQL


Engine options

Engine type [Info](#)

☐ Aurora (MySQL Compatible)


☐ Aurora (PostgreSQL Compatible)


☒ MySQL


☐ MariaDB


3. Choose Free Tier

Templates

Choose a sample template to meet your use case.

☐ **Production**
Use defaults for high availability and fast, consistent performance.

☐ **Dev/Test**
This instance is intended for development use outside of a production environment.

☒ **Free tier**
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
[Info](#)

4. Give DB instance and DB username and password

Settings

DB instance identifier

Info

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username

Info

Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management

You can use AWS Secrets Manager or manage your master user credentials.

☐ Managed in AWS Secrets Manager - *most secure*

RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

☒ Self managed

Create your own password or have RDS create a password that you manage.

Master password

Info

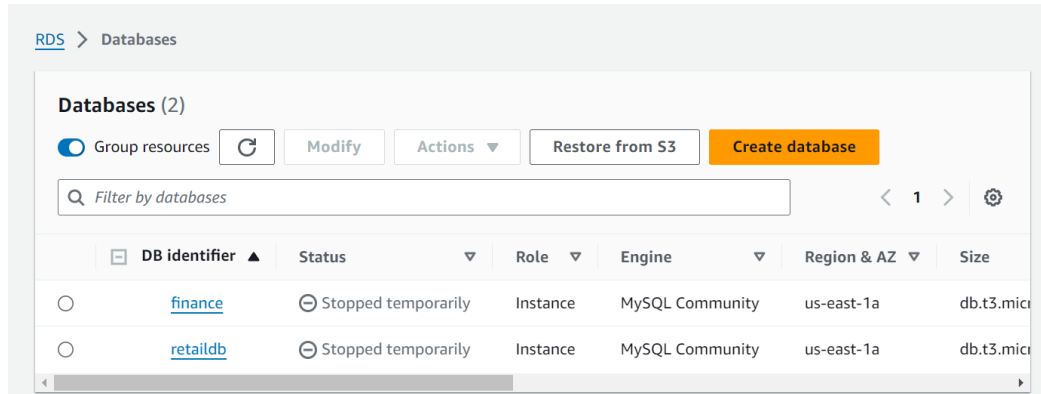
Password strength

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

Confirm master password

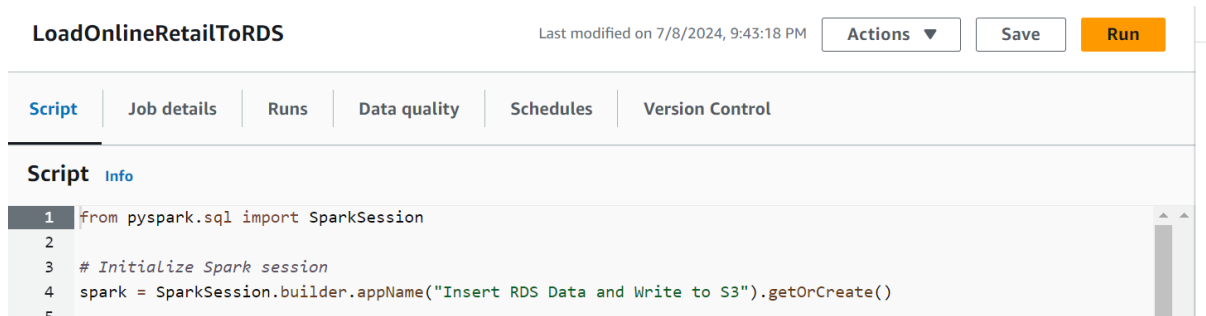
Info

5. Keep everything else as it is.
6. Once the you create it should have status as available (Here I have stopped it for cost purpose)



Part 6: Load Transformed Data To RDS

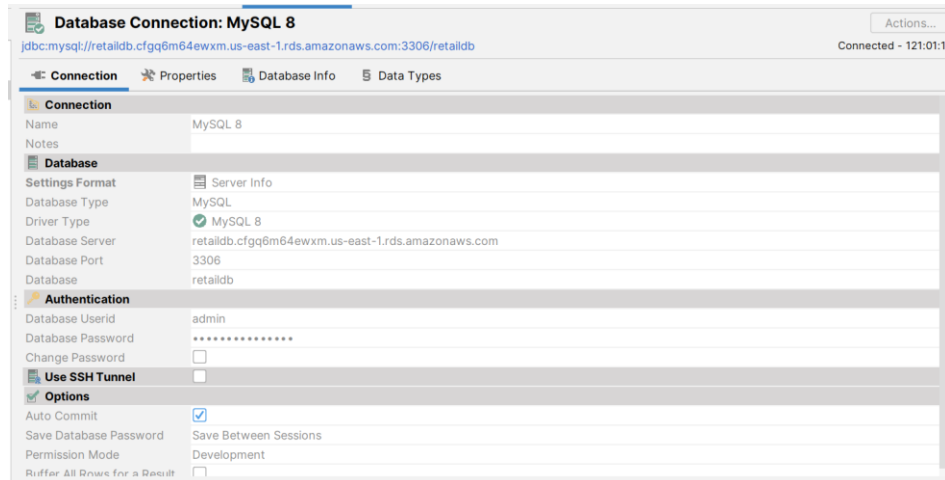
1. Create Spark Glue Job to load s3 data to RDS



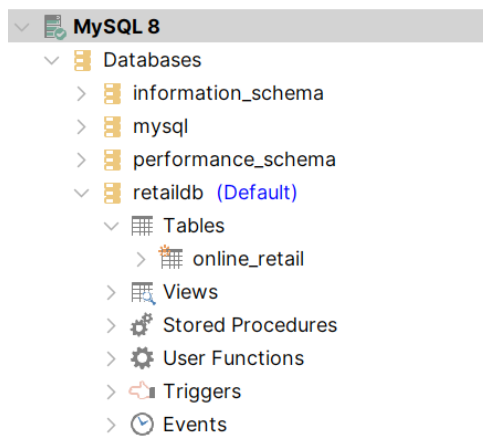
2. Provide connection details

```
rds_options = {
    "url": "jdbc:mysql://retaildb.cf9q6m64ewxm.us-east-1.rds.amazonaws.com:3306/retaildb",
    "user": "admin",
    "password": " ",
    "driver": "com.mysql.cj.jdbc.Driver"
}
```

3. Once the data is loaded check the data from local as well
 - a. Create connection using DBVisualizer



- b. Check if table is created



Part 7: Setup Athena and KPIs

1. Go to Athena Console
2. Use the correct database and Table

Data

↻ <

Data source

AwsDataCatalog ▼

Database

online_retail_db ▼

Tables and views

Create ▼ ⚙

🔍 Filter tables and views

▼ Tables (3) < 1 >

+ online_retail Partitioned ⋮

3. Run aggregation KPI queries:

- Inventory Turnover Ratio** measures how efficiently inventory is managed by indicating how many times inventory is sold and replaced over a period.

1 SELECT

2 SUM(InvoiceTotal) / SUM(Quantity) AS InventoryTurnoverRatio

3 FROM

4 transformed_data

5 WHERE

6 Year = 2011;

7

SQL Ln 1, Col 1

Run again

Explain

Cancel

Clear

Create ▼

☐ Reuse query results
up to 60 minutes ago

Query results

Query stats

✓ Completed

Time in queue: 103 ms

Run time: 942 ms

Data scanned: 55.39 MB

Results (1)

Copy

Download results

🔍 Search rows

< 1 > ⚙

#	InventoryTurnoverRatio
1	1.8605315614265485

- b. **Stockout Rate** measures the percentage of time products are out of stock.

```
1 SELECT
2   COUNT(DISTINCT InvoiceNo) AS StockoutCount
3 FROM
4   transformed_data
5 WHERE
6   Quantity = 0
7   AND Year = 2011;
```

SQL Ln 8, Col 1

[Run again](#) [Explain](#) [Cancel](#) [Clear](#) [Create](#)

☒ Reuse query results up to 60 minutes ago

[Query results](#) | [Query stats](#)

Completed Time in queue: 97 ms Run time: 550 ms Data scanned: 55.39 MB

Results (1) [Copy](#) [Download results](#)

#	StockoutCount
1	0

- c. **COGS to Revenue Ratio** measures the efficiency of managing inventory costs relative to revenue.

```
1 SELECT
2   SUM(Quantity * UnitPrice) / SUM(InvoiceTotal) AS COGSToRevenueRatio
3 FROM
4   transformed_data
5 WHERE
6   Year = 2011;
```

SQL Ln 6, Col 16

[Run again](#) [Explain](#) [Cancel](#) [Clear](#) [Create](#)

☒ Reuse query results up to 60 minutes ago

[Query results](#) | [Query stats](#)

Completed Time in queue: 67 ms Run time: 1.004 sec Data scanned: 55.39 MB

Results (1) [Copy](#) [Download results](#)

#	COGSToRevenueRatio
1	1.00000000674781

- d. **Customer Acquisition Cost (CAC)** measures the average cost of acquiring a new customer.

```
1 SELECT
2   SUM(InvoiceTotal) / COUNT(DISTINCT CustomerID) AS CAC
3 FROM
4   transformed_data
5 WHERE
6   Year = 2011;
```

SQL Ln 7, Col 1

[Run again](#) [Explain](#) [Cancel](#) [Clear](#) [Create](#)

☐ Reuse query results up to 60 minutes ago

[Query results](#) [Query stats](#)

Completed Time in queue: 101 ms Run time: 827 ms Data scanned: 55.39 MB

Results (1) [Copy](#) [Download results](#)

#	CAC
1	2106.045868010359

- e. **Returning Customers:** The total number of unique customers who made purchases on July 11, 2011.

```
1 SELECT
2   COUNT(DISTINCT CustomerID) AS ReturningCustomers,
3   COUNT(DISTINCT CASE WHEN Quantity > 0 THEN CustomerID ELSE NULL END) AS TotalCustomers
4 FROM
5   transformed_data
6 WHERE
7   Year = 2011
8   AND Month = 7
9   AND DayOfMonth = 11;
```

SQL Ln 10, Col 1

[Run again](#) [Explain](#) [Cancel](#) [Clear](#) [Create](#)

☐ Reuse query results up to 60 minutes ago

[Query results](#) [Query stats](#)

Completed Time in queue: 116 ms Run time: 787 ms Data scanned: 55.39 MB

Results (1) [Copy](#) [Download results](#)

#	ReturningCustomers	TotalCustomers
1	49	43

- f. **Average Order Value (AOV):** Measures the average amount spent per order over the course of the year 2011.

```
1 SELECT
2   AVG(InvoiceTotal) AS AOV
3 FROM
4   transformed_data
5 WHERE
6   Year = 2011;
```

SQL Ln 7, Col 1

[Run again](#) [Explain](#) [Cancel](#) [Clear](#) [Create](#)

☐ Reuse query results
up to 60 minutes ago

[Query results](#) | [Query stats](#)

[Completed](#) Time in queue: 69 ms Run time: 679 ms Data scanned: 55.39 MB

Results (1) [Copy](#) [Download results](#)

< 1 > [Settings](#)

#	AOV
1	18.224661789538825

- g. **Gross Margin:** Measures the profitability of products by comparing the revenue (InvoiceTotal) to the cost of goods sold (Quantity * UnitPrice).

```
1 SELECT
2   SUM(InvoiceTotal - (Quantity * UnitPrice)) / SUM(InvoiceTotal) AS GrossMargin
3 FROM
4   transformed_data
5 WHERE
6   Year = 2011;
```

SQL Ln 7, Col 1

[Run again](#) [Explain](#) [Cancel](#) [Clear](#) [Create](#)

☐ Reuse query results
up to 60 minutes ago

[Query results](#) | [Query stats](#)

[Completed](#) Time in queue: 98 ms Run time: 692 ms Data scanned: 55.39 MB

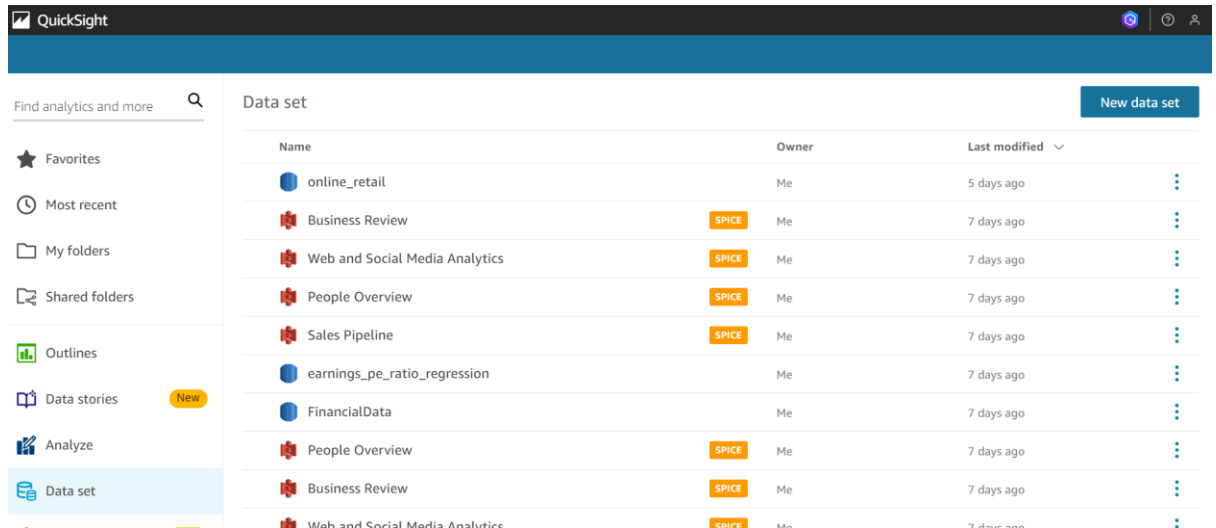
Results (1) [Copy](#) [Download results](#)

< 1 > [Settings](#)

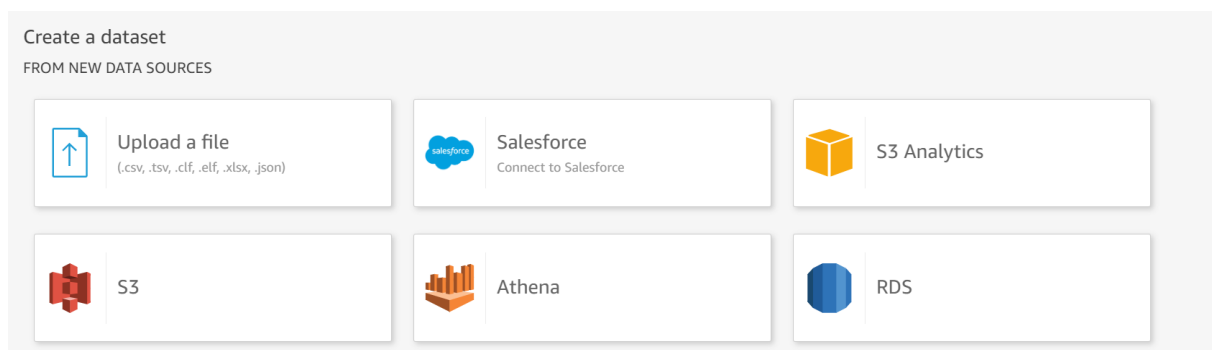
#	GrossMargin
1	-6.747730544346138E-10

Part 8: Quicksight and Visualizations

1. Go to Quicksight and login
2. Choose the dataset to have visualization on



3. Click on new dataset and choose RDS



4. Enter RDS details and test the connection

New RDS data source ×

Data source name

Instance ID

Select an instance ID ▼

Connection type

Public network ▼

Database name

User name

Password

Validate connection

SSL is enabled


Create data source


5. Once created, it will be listed in the datasets


New data set

SPICE capacity for this region: A

Your datasets

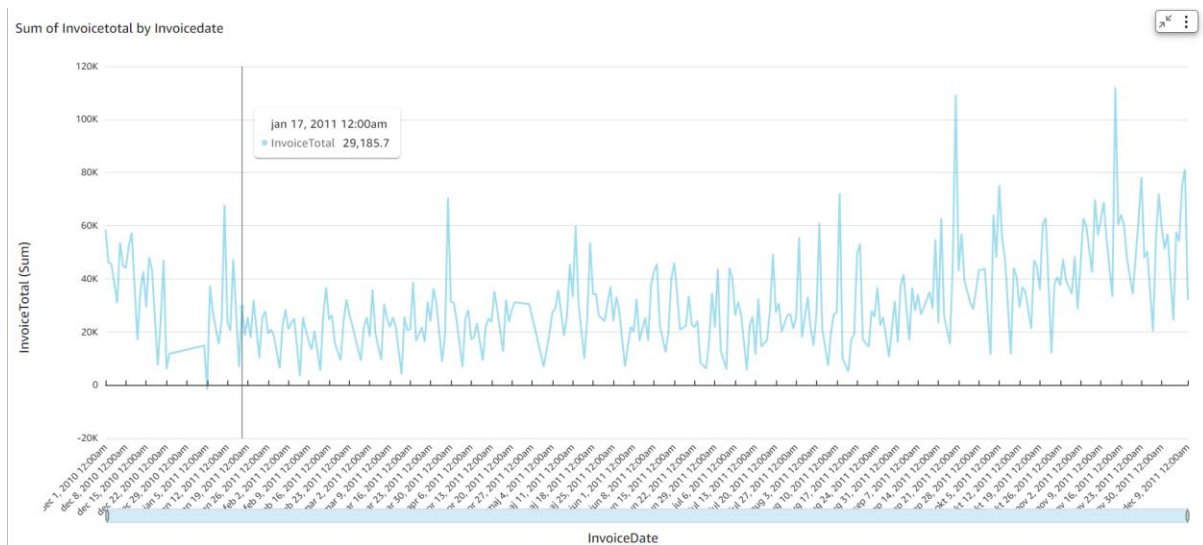
 online_retail

 Business Review
SPICE

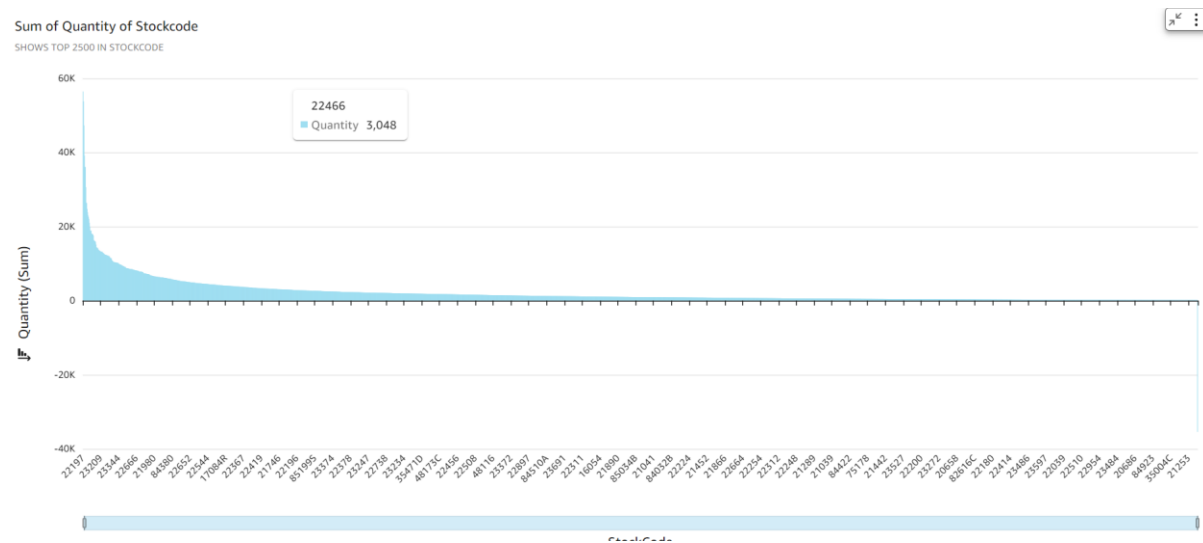
 Web and Social Media An...
SPICE

6. Create Dashboard

a. Sum of Invoice total by invoice date

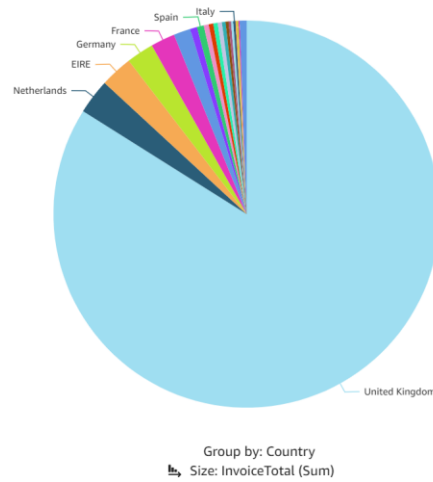


b. Sum of Quantity of Stockcode



c. Sum of invoice total by country

Sum of Invoice total by Country
SHOWS THE TOP 20 IN COUNTRY

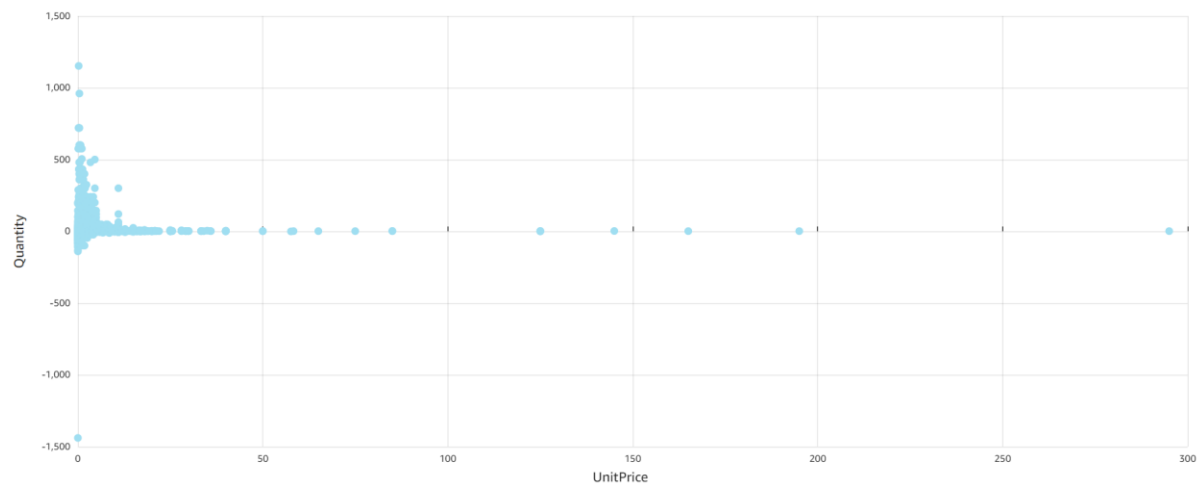


Country

United Kin...
Netherlands
OWNER
Germany
France
Australia
Switzerland
Spain
Belgium
Sweden
Japan
Norway
Portugal
Finland
Channel Isl...
Denmark
Italy
Cyprus
Austria

d. Number of items of Unitprice and Quantity

Number of items of Unitprice and Quantity
DISPLAYS UP TO 2500 DATA POINTS

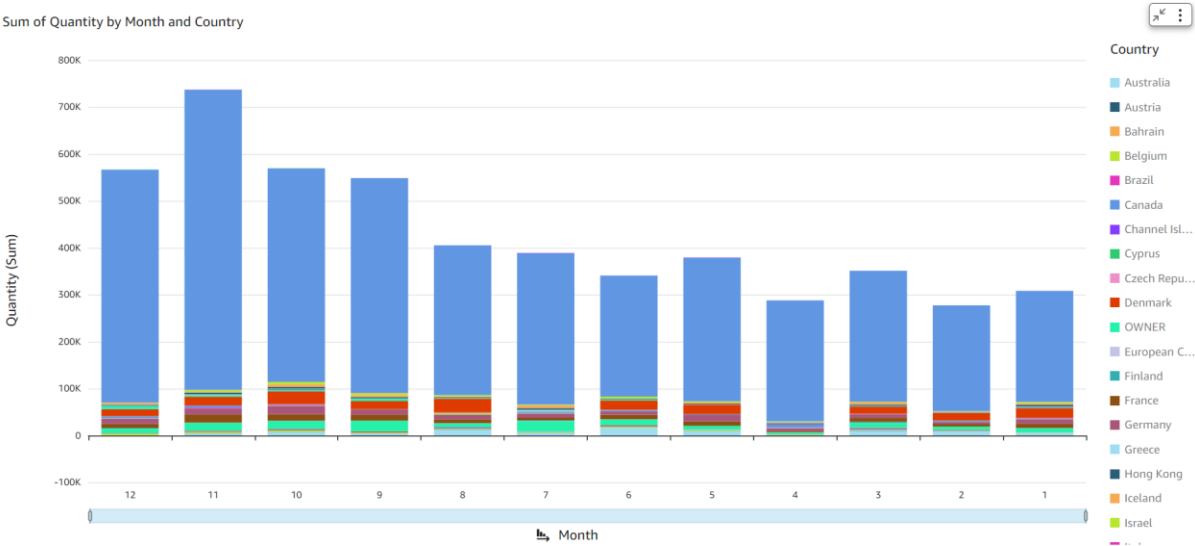


e. Sum of Invoicetotal, Sum of Quantity, and Sum of Unitprice by Month

Sum of Invoicetotal, Sum of Quantity, and Sum of Unitprice by Month

Rows	InvoiceTotal	Quantity	UnitPrice
1	558,448.56	308,281	172,003.69
2	497,026.41	277,374	126,841.95
3	682,013.98	351,165	170,778.3
4	492,367.84	288,237	128,689.46
5	722,094.1	379,652	190,058.09
6	689,977.23	340,945	200,032.62
7	680,156.99	389,051	171,424.58
8	681,386.46	405,450	149,831.85
9	1,017,596.68	548,669	198,308.68
10	1,069,368.23	569,749	261,626.83
11	1,456,145.8	737,182	323,943.25
12	1,179,424.67	566,747	392,533.67

f. Sum of Quantity by Month and Country



Part 9: S3 lifecycle

1. Go to Lifecycle Rule in Management of S3 bucket

[Amazon S3](#) > [Buckets](#) > [ade-project](#) > [Lifecycle configuration](#) > Create lifecycle rule

Create lifecycle rule [Info](#)

Lifecycle rule configuration

Lifecycle rule name

Up to 255 characters

Choose a rule scope

☒ Limit the scope of this rule using one or more filters

☐ Apply to all objects in the bucket

Filter type

You can filter objects by prefix, object tags, object size, or whatever combination suits your usecase.

Prefix

Add filter to limit the scope of this rule to a single prefix.

Don't include the bucket name in the prefix. Using certain characters in key names can cause problems with some applications and protocols. [Learn more](#)

2. Add Rules and Transitions

☒ Move current versions of objects between storage classes

☐ Move noncurrent versions of objects between storage classes

☐ Expire current versions of objects

☒ Permanently delete noncurrent versions of objects

☐ Delete expired object delete markers or incomplete multipart uploads

These actions are not supported when filtering by object tags or object size.

Transition current versions of objects between storage classes

Choose transitions to move current versions of objects between storage classes based on your use case scenario and performance access requirements. These transitions start from when the objects are created and are consecutively applied. [Learn more](#)

Choose storage class transitions	Days after object creation	
<input type="text" value="Standard-IA"/>	<input type="text" value="60"/>	<input type="button" value="Remove"/>
<input type="text" value="Intelligent-Tiering"/>	<input type="text" value="120"/>	<input type="button" value="Remove"/>
<input type="button" value="Add transition"/>		

3. Add Glacier also

Transition current versions of objects between storage classes

Choose transitions to move current versions of objects between storage classes based on your use case scenario and performance access requirements. These transitions start from when the objects are created and are consecutively applied. [Learn more](#)

Choose storage class transitions	Days after object creation	
Standard-IA ▼	60	Remove
Intelligent-Tiering ▼	120	Remove
Glacier Instant Retrieval ▼	180	Remove
<button>Add transition</button>		

4. Add permanent Delete Also

Permanently delete noncurrent versions of objects

Choose when Amazon S3 permanently deletes specified noncurrent versions of objects. [Learn more](#)

Days after objects become noncurrent	Number of newer versions to retain - <i>Optional</i>
365	<i>Number of versions</i>
	Can be up to 100 versions. All other noncurrent versions will be moved.

5. Review transition and expiration actions

Review transition and expiration actions	
<div>Current version actions</div> <div>Day 0</div> <ul style="list-style-type: none">• Objects uploaded <div>↓</div> <div>Day 60</div> <ul style="list-style-type: none">• Objects move to Standard-IA <div>↓</div> <div>Day 120</div> <ul style="list-style-type: none">• Objects move to Intelligent-Tiering <div>↓</div> <div>Day 180</div> <ul style="list-style-type: none">• Objects move to Glacier Instant Retrieval	<div>Noncurrent versions actions</div> <div>Day 0</div> <ul style="list-style-type: none">• Objects become noncurrent <div>↓</div> <div>Day 365</div> <ul style="list-style-type: none">• 0 newest noncurrent versions are retained• All other noncurrent versions are permanently deleted

6. Once create it will be added as below

Amazon S3

>

Buckets

>

ade-project

>

Lifecycle configuration

Lifecycle configuration

Info

To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their lifecycle. A lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. Lifecycle rules run once per day.

Lifecycle rules (1)

Refresh

View details

Edit

Delete

Actions

Create lifecycle rule

Use lifecycle rules to define actions you want Amazon S3 to take during an object's lifetime such as transitioning objects to another storage class, archiving them, or deleting them after a specified period of time.

[Learn more](#)

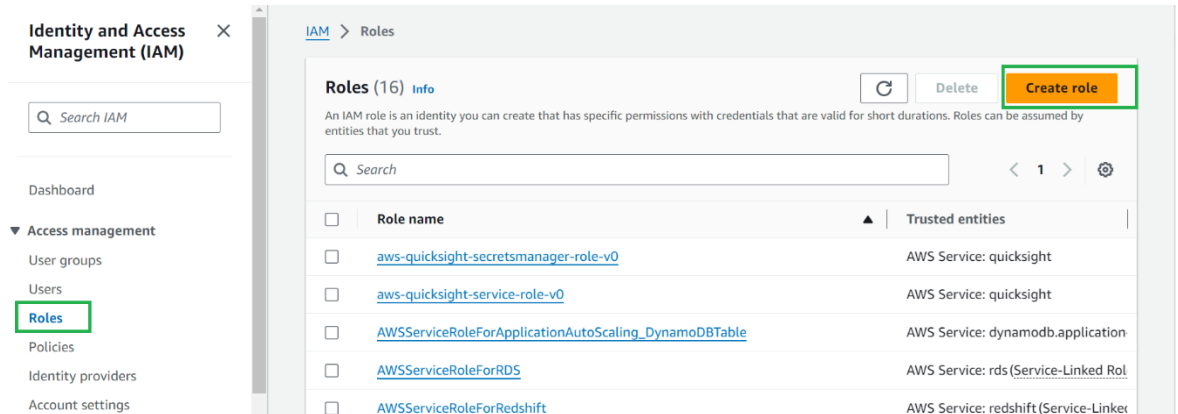
Find lifecycle rules by name

< 1 > ⚙

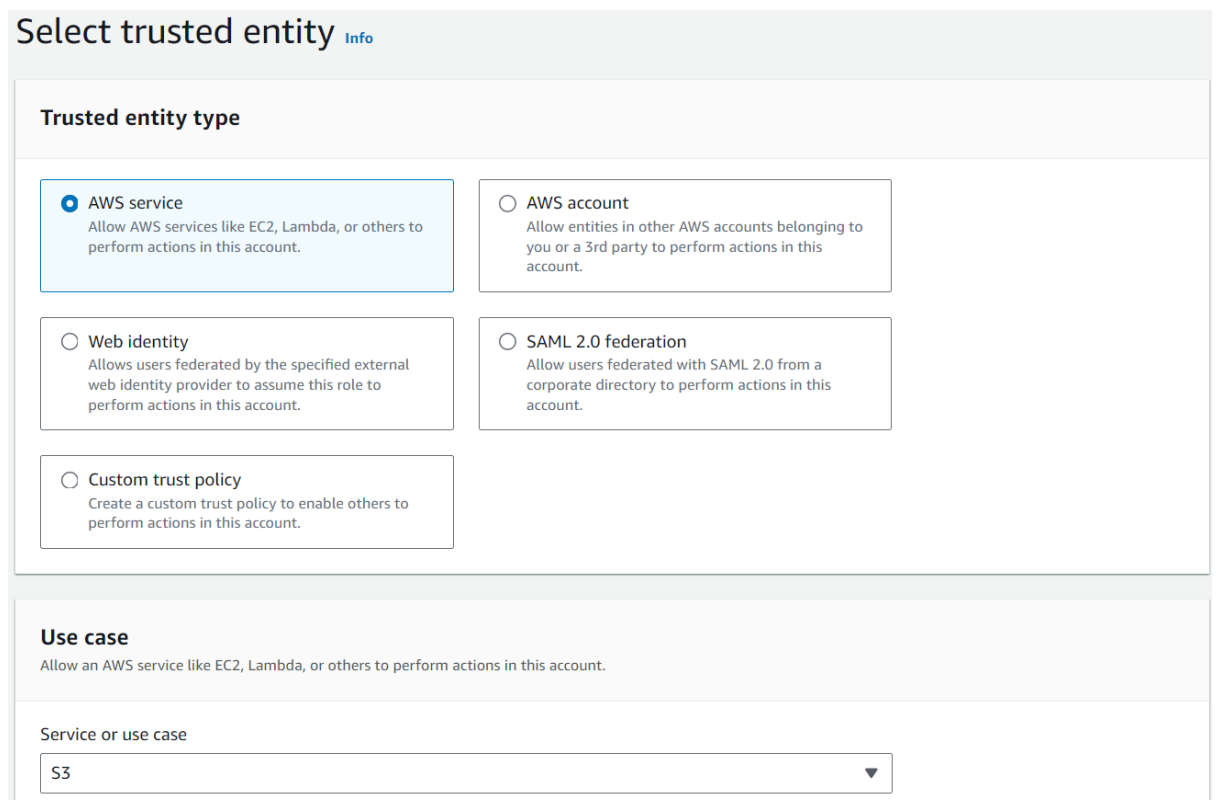
	Lifecycle rul...	Status	Scope	Current ver...	Noncurrent ...	Expired obj...	Incomplete m...
<input type="radio"/>	Policy1	Enabled	Entire bucket	Transition to Standa	Permanently delete	-	-

Part 10: IAM Role

1. Go to IAM Role console
2. Click on Roles and Create Role



3. Choose AWS service and service to add permissions



4. Add necessary permissions

Add permissions [Info](#)

Permissions policies (952) [Info](#)

Choose one or more policies to attach to your new role.

Filter by Type

All types

10 matches

<input type="checkbox"/>	Policy name ↗	Type	Description
<input type="checkbox"/>	AmazonDMSRedshiftS3Role	AWS managed	Provides access to manage S3 settings ...
<input type="checkbox"/>	AmazonS3FullAccess	AWS managed	Provides full access to all buckets via t...
<input type="checkbox"/>	AmazonS3ObjectLambdaExec...	AWS managed	Provides AWS Lambda functions permi...

5. Finally add all the permissions required to the Role as below

Permissions policies (11) [Info](#)

[↻](#) [Simulate](#) [Remove](#) [Add permissions](#)

Filter by Type

All types

<input type="checkbox"/>	Policy name ↗	Type	Attached entities
<input type="checkbox"/>	AmazonKinesisAnalyticsFullAccess	AWS managed	1
<input type="checkbox"/>	AmazonKinesisFirehoseFullAccess	AWS managed	2
<input type="checkbox"/>	AmazonKinesisFullAccess	AWS managed	2
<input type="checkbox"/>	AmazonRDSDataFullAccess	AWS managed	3
<input type="checkbox"/>	AmazonS3FullAccess	AWS managed	4
<input type="checkbox"/>	AWSGlueConsoleFullAccess	AWS managed	4
<input type="checkbox"/>	AWSLambdaKinesisExecutionRole	AWS managed	2
<input type="checkbox"/>	AWSQuicksightAthenaAccess	AWS managed	4
<input type="checkbox"/>	cloudwatch	Customer inline	0
<input type="checkbox"/>	CloudWatchEventsFullAccess	AWS managed	3
<input type="checkbox"/>	CloudWatchLogsFullAccess	AWS managed	3

Part 11: AWS CodeBuild

1. Go to AWS CodeBuild console
2. Click on Create Project
3. Provide the details

Project configuration

Project name

A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.

► **Additional configuration**
Description, Build badge, Concurrent build limit, tags

Source

Add source

Source 1 - Primary

Source provider

GitHub ▼

Repository

☒ Repository in my GitHub account

☐ Public repository

☐ GitHub scoped webhook

4. Create a buildspec.yaml file and push to your github

Buildspec

Build specifications

☐ Insert build commands
Store build commands as build project configuration


☒ Use a buildspec file
Store build commands in a YAML-formatted buildspec file


Buildspec name - optional
By default, CodeBuild looks for a file named buildspec.yaml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).



5. Keep everything else as it is

6. Once created it will come up as below

Developer Tools > CodeBuild > Build projects

Build projects Info  Actions ▾ Create trigger View details Start build ▾ **Create project**

Your projects ▾ < 1 > 

	Name ▾	Source provider	Repository	Latest build status	Description	Last Modified
<input type="radio"/>	OnlineRetail	GitHub	utkarshgupta98/advance_data_engineering 	 Succeeded	-	2 days ago


7. Click on Start build and remember to provide proper permission to IAM

Developer Tools > CodeBuild > Build projects > OnlineRetail

OnlineRetail


Actions ▾ Create trigger Edit Clone Debug build Start build with overrides **Start build**


Configuration


Source provider GitHub	Primary repository utkarshgupta98/advance_data_engineering 	Artifacts upload location -	Service role arn:aws:iam::992382534203:role/service-role/CodeBuildRole
Public builds Disabled			

8. Check if Build is succeeded

Build history Batch history Project details Build triggers Metrics

Build history  Stop build View artifacts View logs Delete builds Retry build

< 1 > 

<input type="checkbox"/>	Build run	Status	Build number	Source version	Submitter	Duration	Completed
<input type="checkbox"/>	OnlineRetail:9ecb4500-983b-47a4-b30e-4a70298a6ef5	 Succeeded	13	-	root	2 minutes 22 seconds	2 days ago

Part 12: AWS CodePipeline

1. Go to AWS CodePipeline console
2. Click on Create Pipeline
3. Give details

Choose pipeline settings [info](#)

Step 1 of 5

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.

No more than 100 characters

Pipeline type

You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.

Execution mode
Choose the execution mode for your pipeline. This determines how the pipeline is run.

☐ Superseded
A more recent execution can overtake an older one. This is the default.

☒ Queued (Pipeline type V2 required)
Executions are processed one by one in the order that they are queued.

☐ Parallel (Pipeline type V2 required)
Executions don't wait for other runs to complete before starting or finishing.

4. Provide Source Details

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

You have successfully configured the action with the provider.

The GitHub (Version 1) action is not recommended
The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. [Learn more](#)

Repository

Branch

5. Add CodeBuild

Build - optional

Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

AWS CodeBuild

Region

US East (N. Virginia)

Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

×

 or

Create project [↗](#)

Environment variables - optional
Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. [Learn more](#) [↗](#)

Add environment variable

Build type

☒ **Single build**
Triggers a single build.

☐ **Batch build**
Triggers multiple builds as a single execution.

6. Keep everything else as it is

7. Once you create a build will be triggered on CodeBuild

Build history							
Build history							
<div><div>⌂</div><div>Stop build</div><div>View artifacts</div><div>View logs</div><div>Delete builds</div><div>Retry build</div></div>							
<div>< 1 > ⚙</div>							
<input type="checkbox"/>	Build run	Status	Build number	Source version	Submitter	Duration	Completed
<input type="checkbox"/>	OnlineRetail:d8299f5d-11c6-4cfc-89ca-3afeecc57106	<div>⌂ In progress</div>	14	arn:aws:s3:::codepipeline-us-east-1-720500140235/OnlineRetailPipeline/SourceArtifacts/x7mlNcK.zip	codepipeline/OnlineRetailPipeline	18 seconds	-

8. Check if all stages are green

OnlineRetailPipeline

Notify

Edit

Stop execution

Clone pipeline

Release change

Pipeline type: V2

Execution mode: QUEUED

Source

Succeeded

Pipeline execution ID: 4dbfea2e-c79b-49ea-bef6-d4ce8e6e9c2a

Source

[GitHub \(Version 1\)](#)

Succeeded - 3 minutes ago

[3d33d7f2](#)

View details

[3d33d7f2](#)

Source: Update buildspec.yml

Disable transition

Build

Succeeded

Start rollback

Pipeline execution ID: 4dbfea2e-c79b-49ea-bef6-d4ce8e6e9c2a

Build

[AWS CodeBuild](#)

Succeeded - Just now