

Cloud Datapipeline

Yash Bhatnagar

Abstract

This project, titled "Cloud Data Pipeline Mastery: End-to-End Analytics with AWS," focuses on building a versatile and scalable data engineering pipeline using AWS services to analyze and visualize any dataset. For this implementation, we used the Spotify dataset, but the architecture is designed to accommodate any dataset, making it suitable for a wide range of real-world applications in data-driven decision-making. By leveraging AWS Glue, S3, Athena, CloudWatch and QuickSight, this project provides an efficient, scalable, and cost-effective solution for processing and analyzing data, offering valuable insights to stakeholders.

Introduction

In this project, we will build a comprehensive data engineering pipeline using AWS cloud services. While we demonstrate the pipeline with the Spotify dataset, the project architecture is flexible enough to handle any dataset. The focus is on processing and analyzing data using various AWS tools like S3, Glue, Athena, and QuickSight.

Project Architecture Overview

- **Staging Layer:** Raw data is stored in an S3 bucket.
- **ETL Pipeline:** AWS Glue processes and transfers data from the staging layer to the data warehouse.
- **Data Warehouse:** Processed data is stored in another S3 bucket.
- **Data Catalog:** AWS Glue Crawler creates a database and tables for the data warehouse.
- **Data Analysis:** AWS Athena queries the processed data.
- **Data Visualization:** AWS QuickSight visualizes the data.

AWS Services Used

- **Amazon S3:** For storing raw and processed data.
- **AWS Glue:** For building and managing ETL pipelines.
- **AWS Athena:** For querying data using SQL-like syntax.
- **AWS QuickSight:** For visualizing data.
- **AWS CloudWatch** monitoring, logging, and alerting across our data pipeline.

Data Source

The data used in this project is sourced from the [Spotify Dataset 2023](#) available on Kaggle. The dataset, created by Tony Gordon Jr., includes detailed information about Spotify albums, artists, tracks, and various audio features like danceability, energy, loudness, and more. Although this project uses the Spotify dataset, the pipeline is designed to be dataset-agnostic.

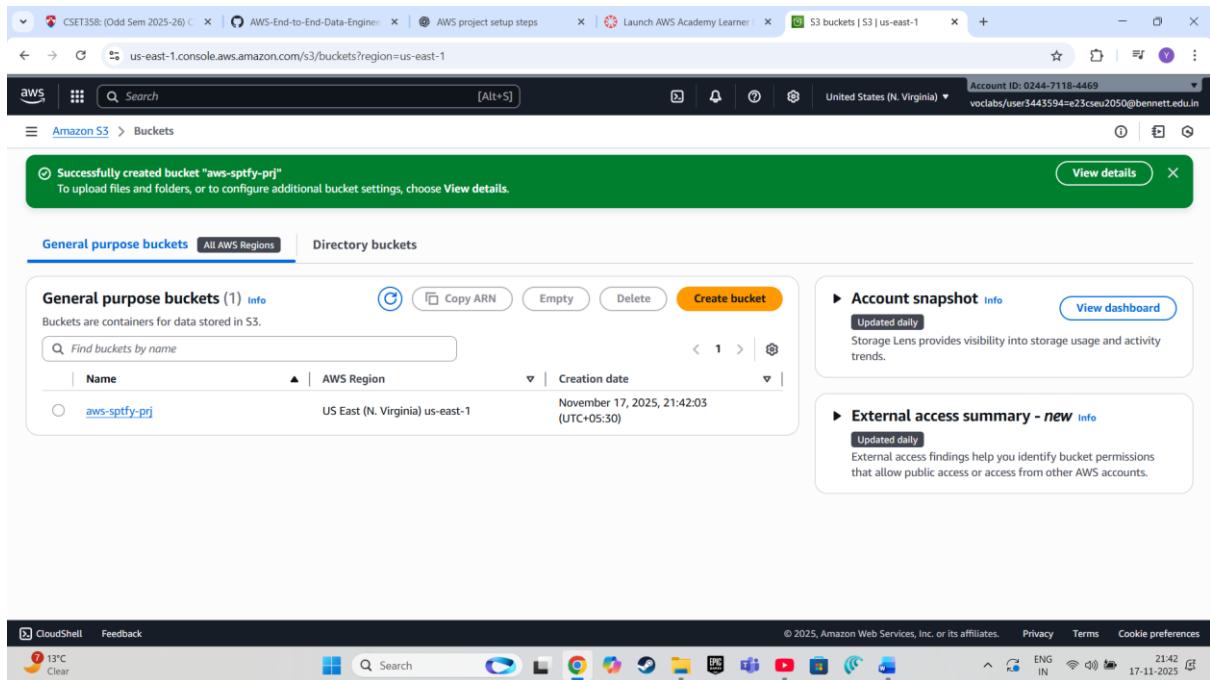
Data Description

- **Albums:** Contains details of all the albums, including album ID, name, popularity, and release date.
- **Artists:** Contains information about the artists, including their names, number of followers, and genres.
- **Tracks:** Contains track-level data, including track ID, popularity, and other features like danceability and energy.
- **Spotify Features:** Contains various audio features like loudness, mode, speechiness, and valence

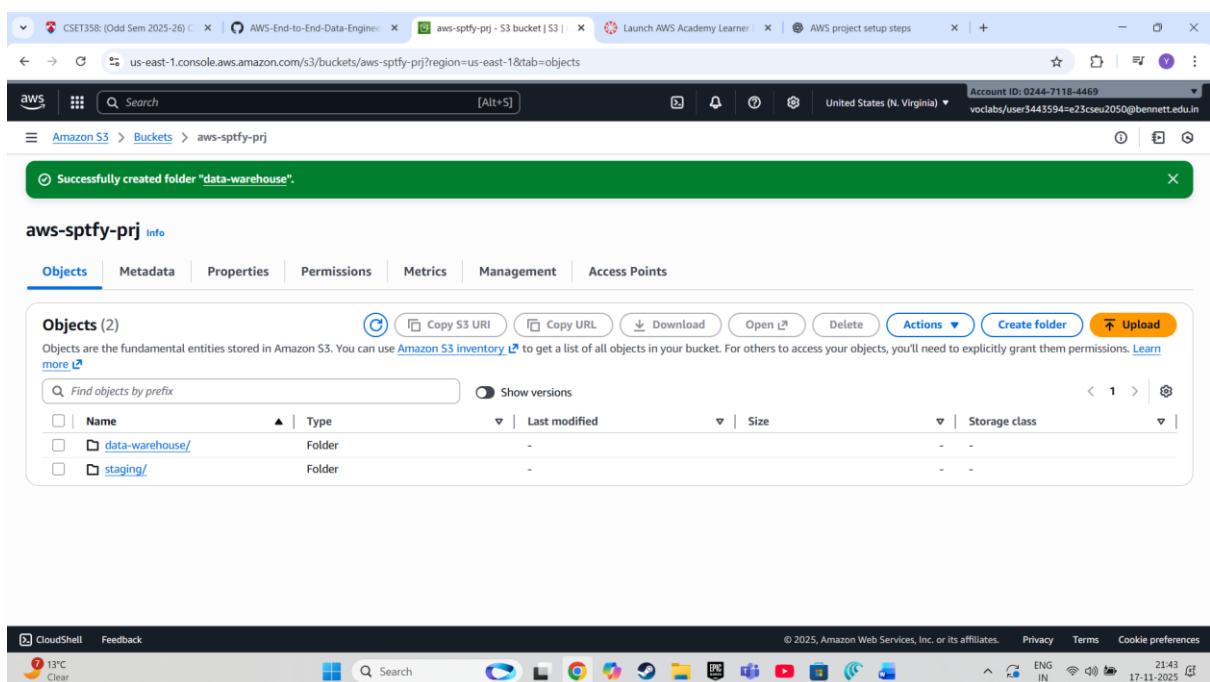
Project Implementation

Creating S3 Buckets

We set up S3 buckets to store raw and processed data. The staging folder in the S3 bucket will hold the raw data files, while the data-warehouse folder will store the processed data. This separation ensures organized storage and easy retrieval of data at different stages of the pipeline.

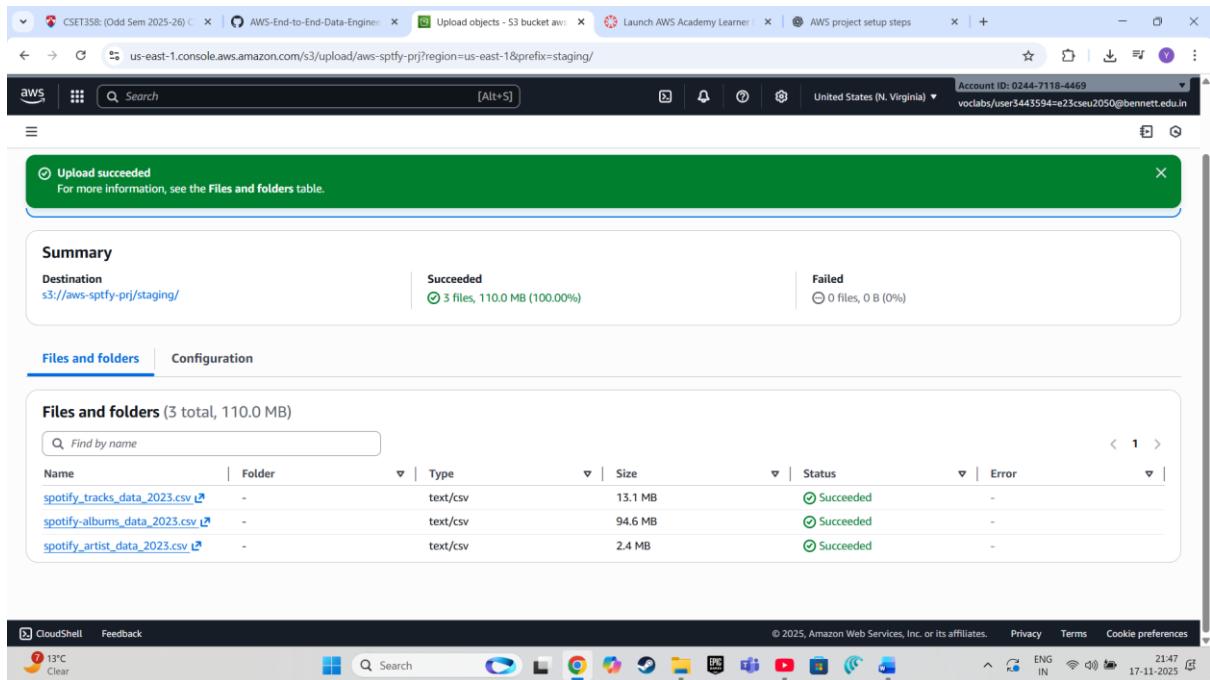


The screenshot shows the AWS S3 console with a green success message: "Successfully created bucket 'aws-sptfy-prj'". The bucket details table shows one item: "aws-sptfy-prj" in the Name column, "US East (N. Virginia) us-east-1" in the AWS Region column, and "November 17, 2025, 21:42:03 (UTC+05:30)" in the Creation date column. To the right, there are "Account snapshot" and "External access summary - new" sections.



The screenshot shows the AWS S3 console with a green success message: "Successfully created folder 'data-warehouse'". The objects list shows two items: "data-warehouse/" and "staging/" both listed as Folders.

We upload the csv files in the staging folder



Upload succeeded
For more information, see the [Files and folders](#) table.

Summary

Destination	Succeeded	Failed
s3://aws-sptfy-prj/staging/	3 files, 110.0 MB (100.00%)	0 files, 0 B (0%)

Files and folders (3 total, 110.0 MB)

Name	Folder	Type	Size	Status	Error
spotify_tracks_data_2023.csv	-	text/csv	13.1 MB	Succeeded	-
spotify_albums_data_2023.csv	-	text/csv	94.6 MB	Succeeded	-
spotify_artist_data_2023.csv	-	text/csv	2.4 MB	Succeeded	-

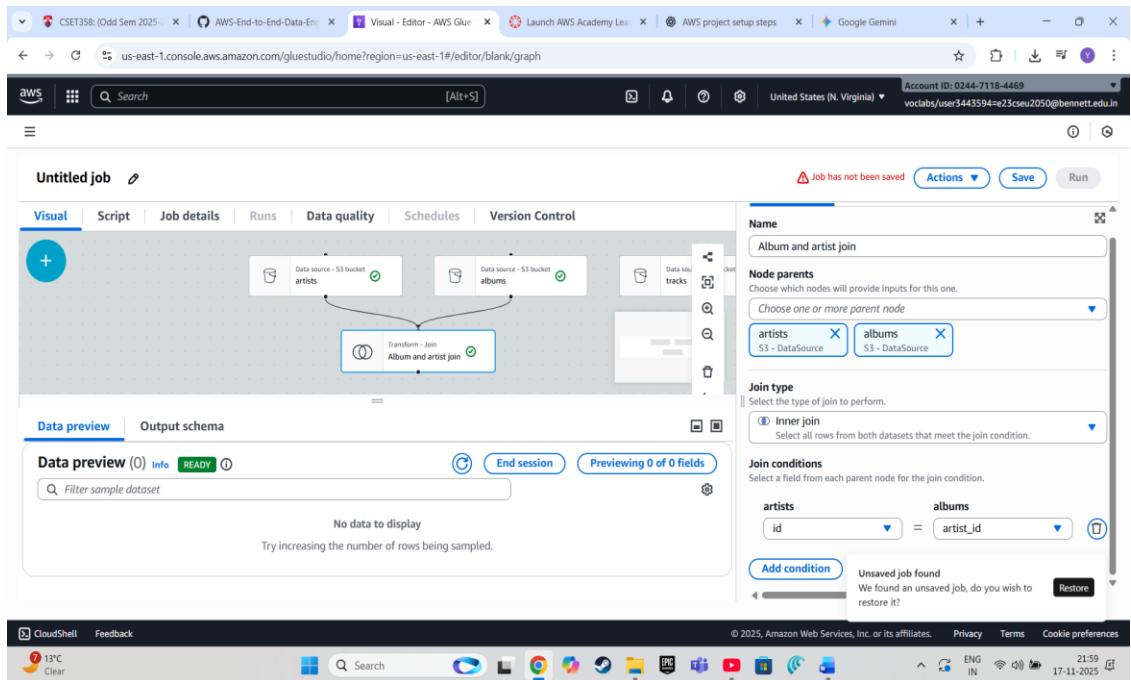
Setting Up AWS Glue for ETL

AWS Glue is used to create a data pipeline that automates the process of extracting data from the staging folder, transforming it by joining and cleaning the data, and then loading it into the data-warehouse folder. This step highlights the power of Glue in managing ETL (Extract, Transform, Load) operations seamlessly.

- Select the **First Amazon S3 bucket** and rename it with “**artist**” and click on **browse** and select the file from the **s3://spotify-aws-prj/staging/spotify_artist_data_2023.csv**. Select the **Data format** as **CSV**.
- Repeat the step for the other two

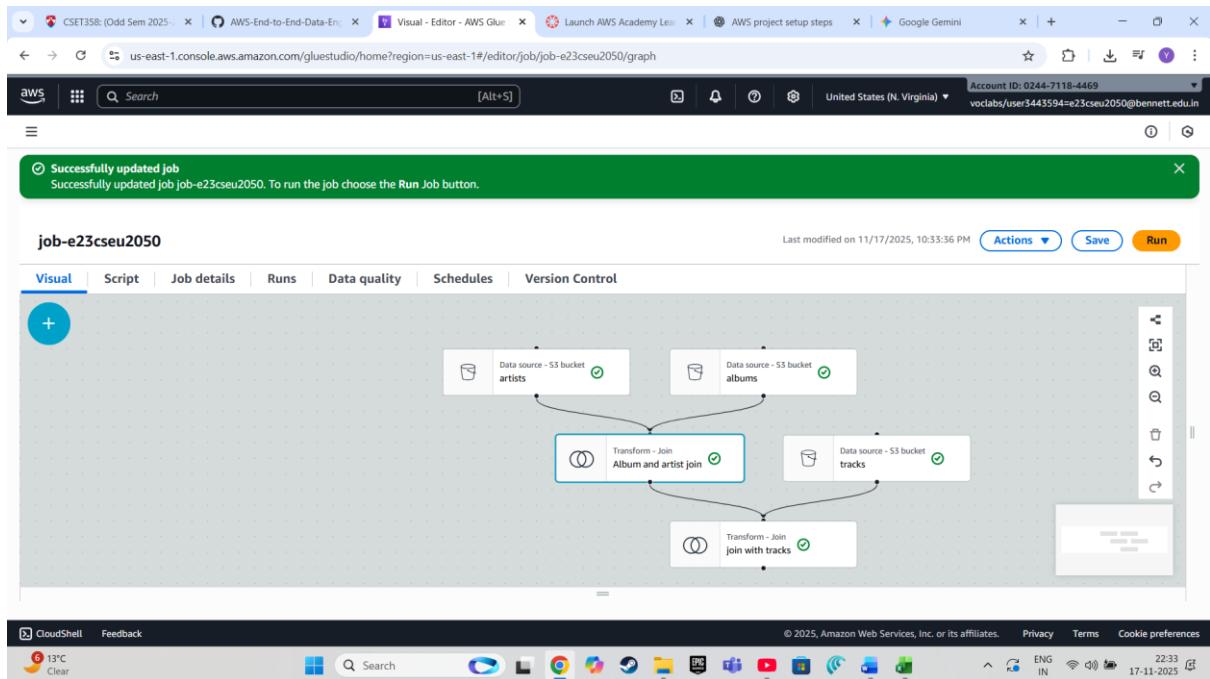
4. Configure Data Transformations:

- Now to join **album** and **artist**, click on the **ADD symbol**, select **join** from **Transforms** and connect nodes as per the image below.

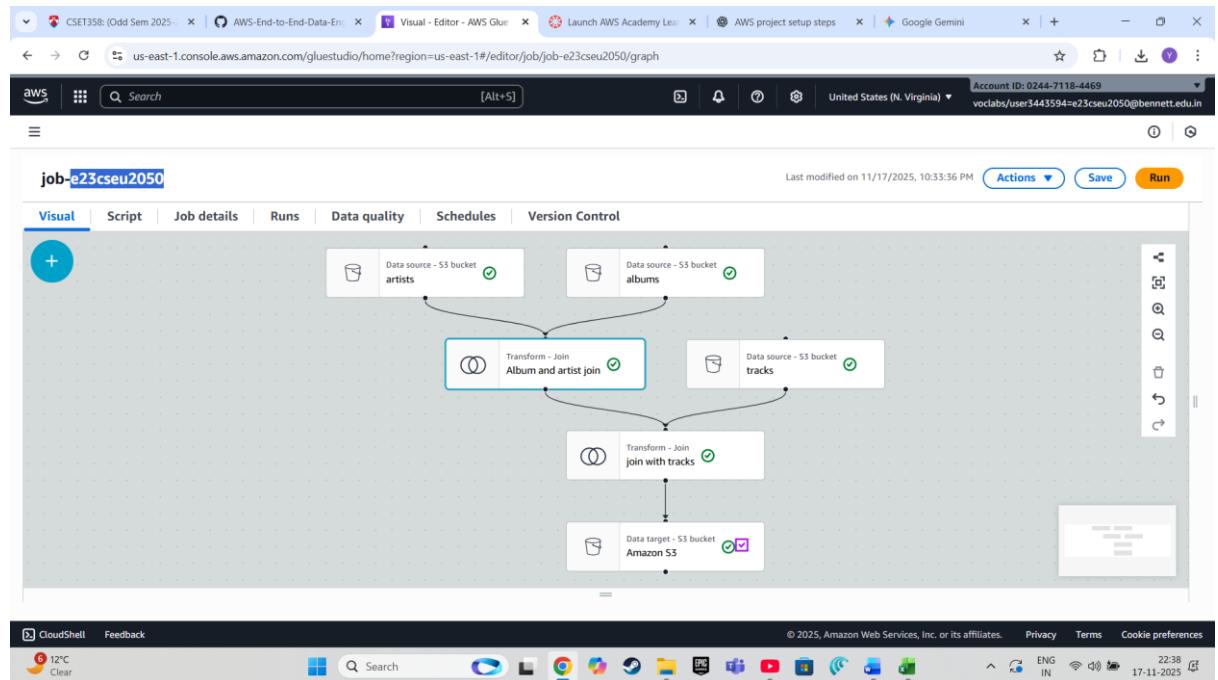


- After Joining the nodes of both the buckets to the **Join Transform**. Add a Condition where **artist 'id' = albums 'artist_id'** and rename the join **“Album and Artist Join”**.

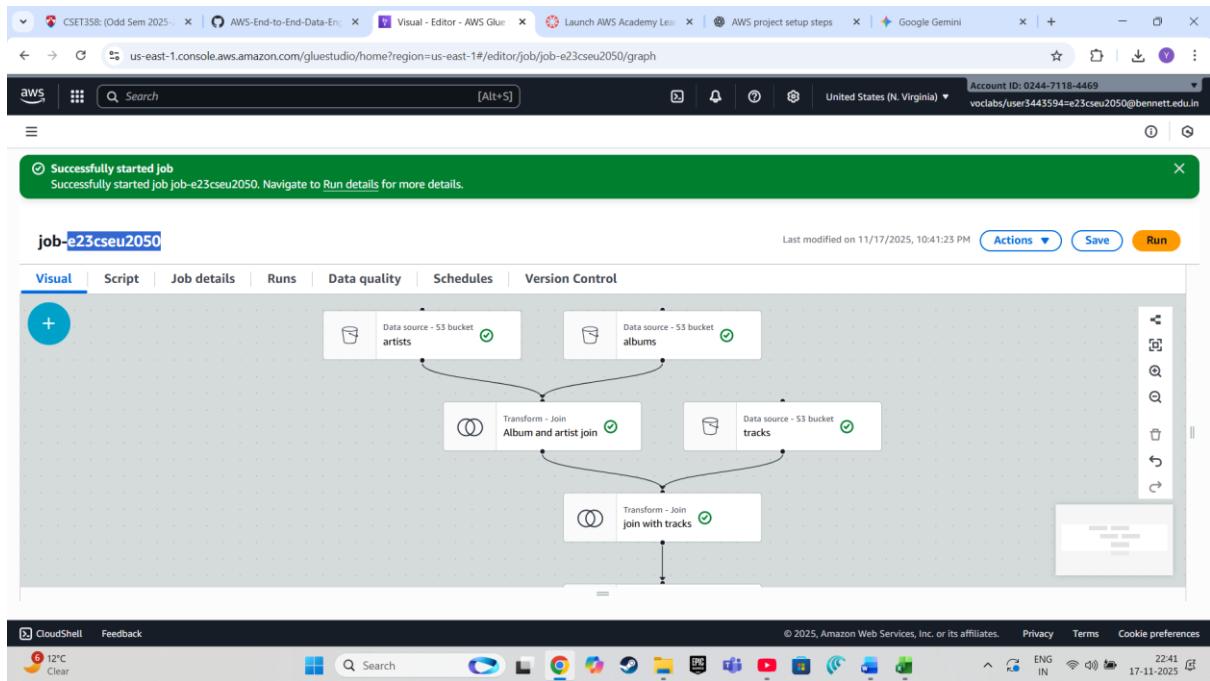
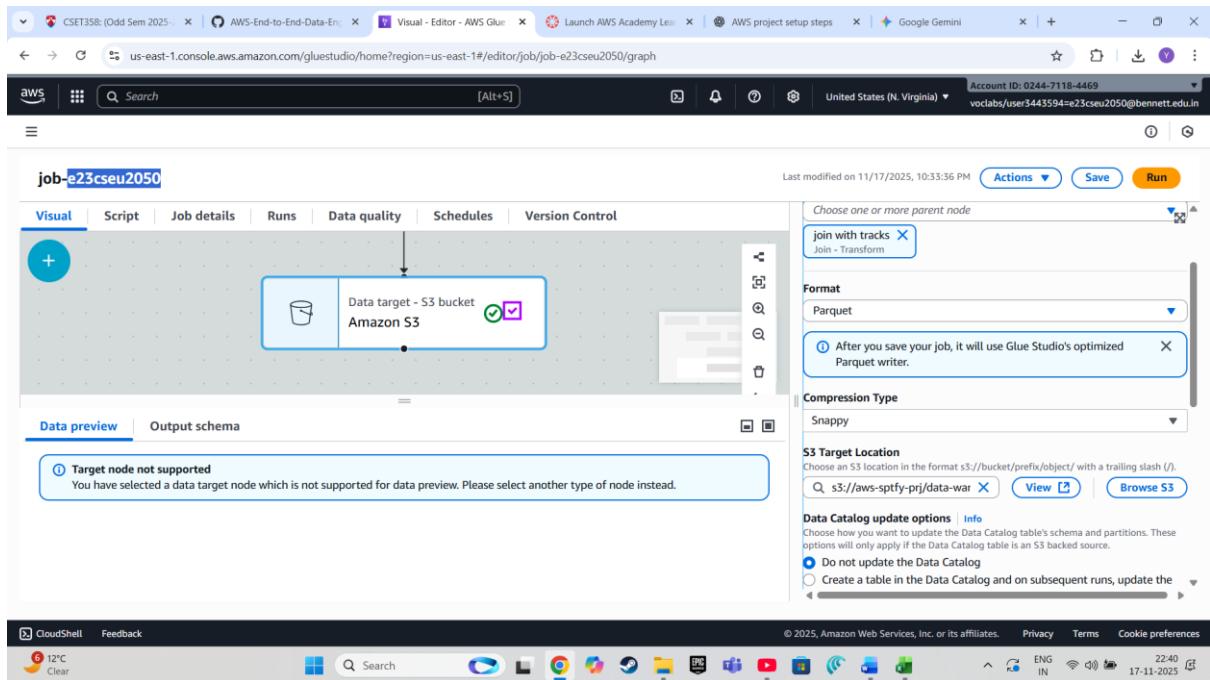
Now add another **Join Transform** to join **‘track’ S3 bucket** and **‘Album and Artist’ Join**.



Now select the **Join** and add the condition **Album and Artist Join ‘track_id’ = tracks ‘id’** and rename the join as **‘Join with Tracks’**.



Rename the Destination node and add the Target location and make sure the compression type is Snappy.



6. Run the Glue Job:

- Review your job settings.
- Click "Run job" to start the ETL process, transforming and moving data from the staging layer to the data warehouse.

The screenshot shows the AWS Glue Studio 'Runs' editor. The top navigation bar includes tabs for 'Visual', 'Script', 'Job details', 'Runs' (which is selected), 'Data quality', 'Schedules', and 'Version Control'. The main content area is titled 'Job runs (1/1) Info' and shows a single run: 'Succeeded' (11/17/2025 22:41:27 - 11/17/2025 22:43:35, 2 m, 10 DPU, G.1X, 5.0). Below this is a detailed table of run parameters. The bottom of the screen shows the AWS navigation bar and a taskbar.

Check S3 bucket if the files are Parsed.

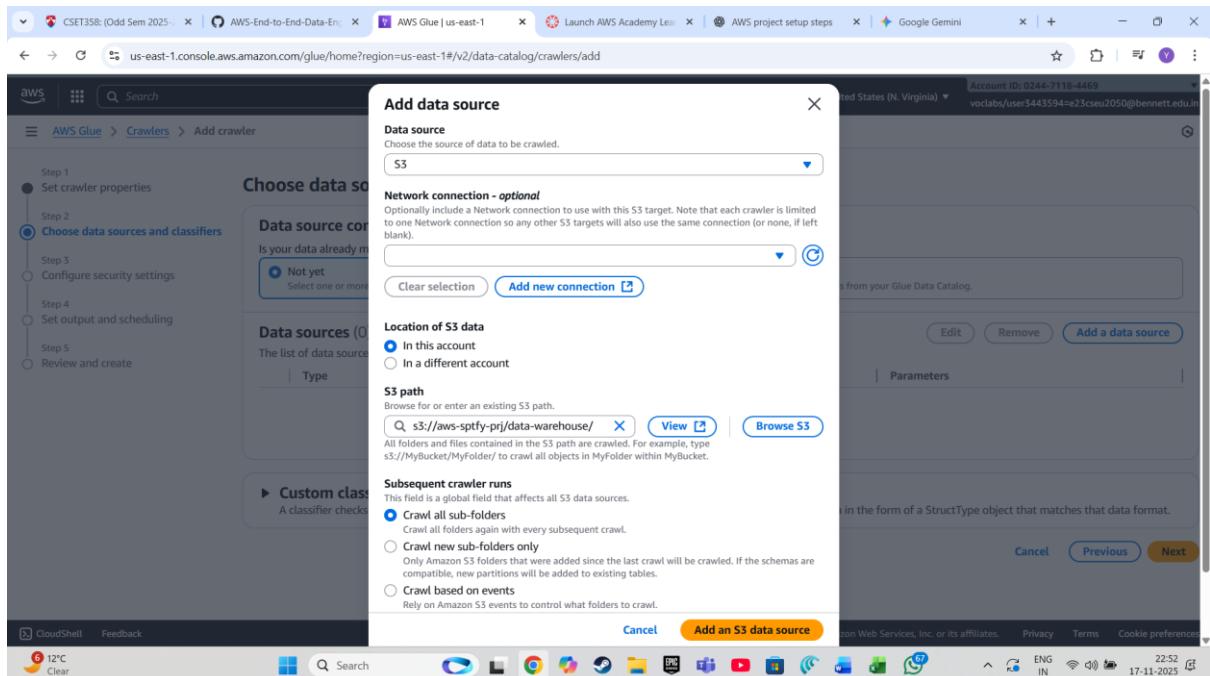
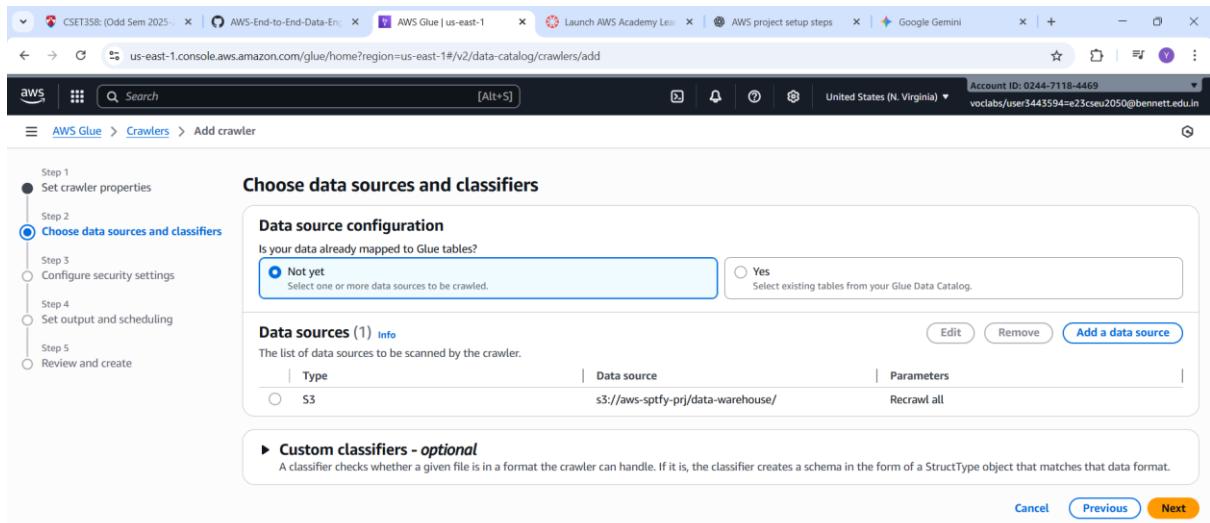
The screenshot shows the AWS S3 console. The left sidebar shows 'Amazon S3' with sections for 'General purpose buckets' (including 'aws-sptfy-prj'), 'Storage Lens', and 'Block Public Access settings'. The main content area shows the 'aws-sptfy-prj' bucket with a 'data-warehouse' folder. The 'Objects' tab is selected, showing 36 parquet files. The table includes columns for Name, Type, Last modified, Size, and Storage class. The bottom of the screen shows the AWS navigation bar and a taskbar.

Step 4: Creating a Data Catalog with AWS Glue Crawler

1. Create a New Crawler:

- In the AWS Glue dashboard, click on "Crawlers" under the "Data catalog" section.

- Click "Create crawler."



2. Create a New Database:

- Open the duplicate tab. Go to the AWS Glue > Data catalog > Databases, and create a new database.

The screenshot shows the AWS Glue Databases console. At the top, there are several tabs: CSET358: (Odd Sem), AWS-End-to-End-Da..., AWS Glue | us-east-1, Databases - AWS Glue, Launch AWS Academy, AWS project setup, Google Gemini, and a blank tab. The main content area is titled "Databases (1)". It displays a table with one row for "sptfy_database2050". The table columns are "Name", "Description", "Location URI", and "Created on (UTC)". The "Name" column shows a dropdown menu with "Filter databases". The "Created on (UTC)" column shows "November 17, 2025 at 17:23:38". At the top right of the table, there are "Edit", "Delete", and "Add database" buttons. The status bar at the bottom shows "CloudShell Feedback", the AWS logo, a search bar, and various system icons.

The screenshot shows the AWS Glue Crawlers console. At the top, there are several tabs: CSET358: (Odd Sem), AWS-End-to-End-Da..., AWS Glue - AWS Glue, Databases - AWS Glue, Launch AWS Academy, AWS project setup, Google Gemini, and a blank tab. The main content area is titled "Crawlers - AWS Glue". It shows a success message: "One crawler successfully created. The following crawler is now created: 'sptfy_crawler_2050'". Below this, the crawler "sptfy_crawler_2050" is listed. The "Crawler properties" section shows details: Name (sptfy_crawler_2050), IAM role (LabRole), Database (sptfy_database2050), State (READY), Description (empty), Security configuration (empty), Lake Formation configuration (empty), and Table prefix (empty). The "Advanced settings" section is collapsed. Below the properties, there are tabs for "Crawler runs", "Schedule", "Data sources", "Classifiers", and "Tags". The "Crawler runs" tab is selected, showing a list with "(0)" entries. At the bottom right, there are buttons for "Run crawler", "Edit", and "Delete". The status bar at the bottom shows "CloudShell Feedback", the AWS logo, a search bar, and various system icons.

The screenshot shows the AWS Glue Crawler configuration page. At the top, a green banner indicates that the crawler 'sptfy_crawler_2050' is 'Crawler successfully starting'. Below this, the crawler's properties are listed, including its name (sptfy_crawler_2050), IAM role (LabRole), database (sptfy_database2050), and state (READY). The crawler is currently in the 'READY' state. The 'Crawler runs' tab is selected, showing one run listed. The run details include a 'Stop run' button, a 'View CloudWatch logs' button, and a 'View run details' button. The browser status bar at the bottom shows the URL as 'us-east-1.console.aws.amazon.com/glue/home?region=us-east-1#v2/data-catalog/crawlers/view/sptfy_crawler_2050' and the account ID as '0244-7118-4469'.

The crawler will scan the data in the **data-warehouse** folder and create corresponding tables in the **spotify_data** database.

Step 5: Querying Data with AWS Athena

1. Set Up Query Result Storage:
2. Navigate to **AWS Athena** from the **AWS Management Console**
3. Before running any queries, you must specify a location for query results. Create a new **S3 bucket** named **sptfy-proj-athena-output**.

Settings successfully updated.

Query result location: <s3://sptfy-proj-athena-output/>

Encrypt query results

Expected bucket owner

Assign bucket owner full control over query results
Turned off

The results will be displayed in the **query editor**, and the output will be saved in the specified **S3 bucket (athena-output)**.

The screenshot shows the AWS Athena Query Editor interface. On the left, the 'Data' sidebar is open, showing the 'Data source' set to 'AwsDataCatalog', 'Catalog' set to 'None', and 'Database' set to 'sptfy_database2050'. Below this, the 'Tables and views' section shows 'Tables (1)' with 'data_warehouse' selected, and 'Views (0)'. The main area is a SQL editor with the query: '1 select*from data_warehouse limit 10;'. Below the editor, the status bar indicates 'SQL Ln 1, Col 1'. Below the editor, there are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. The 'Query results' tab is selected, showing a green status bar with 'Completed', 'Time in queue: 116 ms', 'Run time: 549 ms', and 'Data scanned: 2.17 MB'. Below this, the 'Results (10)' section shows a table with 10 rows. At the bottom of the results table, there are 'Copy' and 'Download results CSV' buttons. The bottom of the screen shows the Windows taskbar with various pinned icons.

The screenshot shows the AWS S3 console. The left sidebar is titled 'Amazon S3' and includes sections for 'General purpose buckets', 'Access Points', 'Access Points (General Purpose Buckets, FSx file systems)', 'Access Points (Directory Buckets)', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'IAM Access Analyzer for S3', and 'Storage Lens'. The main content area shows a list of objects in a bucket named 'sptfy-proj-athena-output'. The list shows two objects: '48521b32-1372-4a3b-9844-cafab093ec1.csv' (Type: csv, Last modified: November 17, 2025, 23:08:55 (UTC+05:30), Size: 4.1 KB, Storage class: Standard) and '48521b32-1372-4a3b-9844-cafab093ec1.csv.metadata' (Type: metadata, Last modified: November 17, 2025, 23:08:55 (UTC+05:30), Size: 2.0 KB, Storage class: Standard). The bottom of the screen shows the Windows taskbar with various pinned icons.

Step 6: Visualizing Data with AWS QuickSight

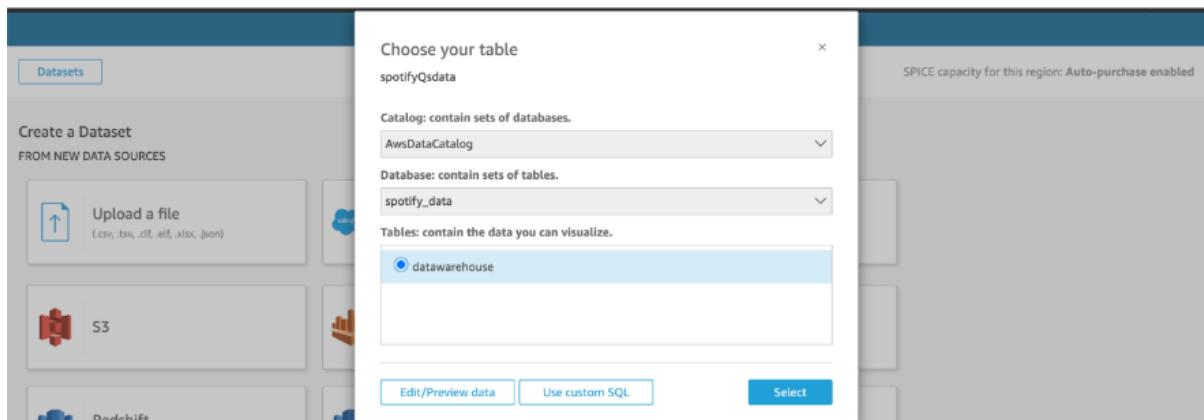
Connect QuickSight to Athena:

Once signed in, go to "**Datasets**" and click "**New dataset**."



② Select "**Athena**" as the data source and name it with **SpotifyQsData**

② Choose the **spotify_data** database and the **data_warehouse** table.



Create Visualizations:

- After importing the data, you can create various types of visualizations (e.g., **bar charts**, **line charts**, **pie charts**) using the fields from the **data_warehouse** table.
- **Example:** Create a bar chart to visualize the popularity of tracks by artist.

4. Publish and Share Dashboards:

- Once your visualizations are ready, you can **publish the dashboard** and share it with stakeholders.
- Click "**Publish dashboard**" and follow the prompts to share it via email or a link.

Step 7: AWS CloudWatch

Screenshot of the AWS CloudWatch Log Events page for the log group /aws-glue/crawlers/sptfy_crawler_2050. The page shows log entries for a crawler named 'sptfy_crawler_2050'.

Log events

Timestamp | Message

- 2025-11-17T22:54:48.059+05:30 [644e33f-a6eb-43fc-a047-940a53e906f9] BENCHMARK : Running Start Crawl for Crawler sptfy_crawler_2050
- 2025-11-17T22:54:54.044+05:30 [644e33f-a6eb-43fc-a047-940a53e906f9] BENCHMARK : Classification complete, writing results to database sptfy_database2050
- 2025-11-17T22:54:54.048+05:30 [644e33f-a6eb-43fc-a047-940a53e906f9] INFO : Crawler configured with Configuration {"Version":1.0,"CreatePartitionIndex":true} and S...
- 2025-11-17T22:55:15.564+05:30 [644e33f-a6eb-43fc-a047-940a53e906f9] INFO : Created table data_warehouse in database sptfy_database2050
- 2025-11-17T22:55:15.841+05:30 [644e33f-a6eb-43fc-a047-940a53e906f9] BENCHMARK : Finished writing to Catalog
- 2025-11-17T22:55:15.870+05:30 [644e33f-a6eb-43fc-a047-940a53e906f9] BENCHMARK : Crawler has finished running and is in state READY
- 2025-11-17T22:55:15.876+05:30 [644e33f-a6eb-43fc-a047-940a53e906f9] INFO : Run Summary For TABLE:
- 2025-11-17T22:55:15.876+05:30 [644e33f-a6eb-43fc-a047-940a53e906f9] INFO : ADD: 1

No newer events at this moment. *Auto retry paused. Resume*

CloudWatch Log groups Log Anomalies Live Tail Logs Insights Contributor Insights

CloudShell Feedback 12°C Clear 12:48 17-11-2025

Screenshot of the AWS CloudWatch Log streams page for the log group /aws-glue/jobs/error. The page shows 10 log streams.

Log streams (10)

By default, we only load the most recent log streams.

Log stream	Last event time
ir_46153e3c95f056b8fc4de6bde714744a0f7f94ea95f4a5840fd30e3e36517bf9	2025-11-17 22:43:24 (UTC+05:30)
ir_46153e3c95f056b8fc4de6bde714744a0f7f94ea95f4a5840fd30e3e36517bf9_g-be	2025-11-17 22:43:23 (UTC+05:30)
ir_46153e3c95f056b8fc4de6bde714744a0f7f94ea95f4a5840fd30e3e36517bf9_g-be	2025-11-17 22:43:23 (UTC+05:30)
ir_46153e3c95f056b8fc4de6bde714744a0f7f94ea95f4a5840fd30e3e36517bf9_g-65	2025-11-17 22:43:23 (UTC+05:30)
ir_46153e3c95f056b8fc4de6bde714744a0f7f94ea95f4a5840fd30e3e36517bf9_g-13	2025-11-17 22:43:23 (UTC+05:30)
ir_46153e3c95f056b8fc4de6bde714744a0f7f94ea95f4a5840fd30e3e36517bf9_g-16	2025-11-17 22:43:23 (UTC+05:30)

CloudWatch Log groups Log Anomalies Live Tail Logs Insights Contributor Insights

CloudShell Feedback 12°C Clear 12:48 17-11-2025

Course: CSET358: Cloud Computing | AWS-End-to-End-Design | Create alarm | Alarms | AWS project setup | Launch AWS Academy | Google Gemini | Top 5 Open Source | +

us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:create?~{Page~MetricSelection~AlarmType~MetricAlarm~AlarmData~Namespace~Glue~MetricName~g... Account ID: 0244-7118-4469

aws Search [Alt+S] United States (N. Virginia) Account ID: 0244-7118-4469

vocabs/user3443594=e23cseu2050@bennett.edu.in

CloudWatch > Alarms > Create alarm

Step 1 Specify metric and conditions

Step 2 Configure actions

Step 3 Add alarm details

Step 4 Preview and create

Specify metric and conditions

Metric

Graph
This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes.

No unit

10

5.16

0.329

21:00 21:30 22:00 22:30 23:00 23:30

glue.1.system.cpuSystemLoad

Namespace
Glue

Metric name
glue.1.system.cpuSystemLoad

Type
gauge

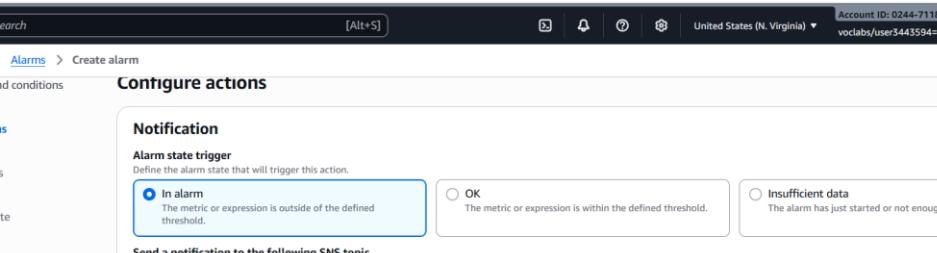
JobRunId
ALL

JobName
job-e23cseu2050

Statistic
Average

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

12°C Clear ENG IN 17-Nov-2025



Specify metric and conditions

Step 2 **Configure actions**

Step 3

Step 4

Preview and create

Configure actions

Notification

Alarm state trigger
Define the alarm state that will trigger this action.

In alarm
The metric or expression is outside of the defined threshold.

OK
The metric or expression is within the defined threshold.

Insufficient data
The alarm has just started or not enough data is available.

Remove

Send a notification to the following SNS topic
Define the SNS (Simple Notification Service) topic that will receive the notification.

Select an existing SNS topic

Create new topic

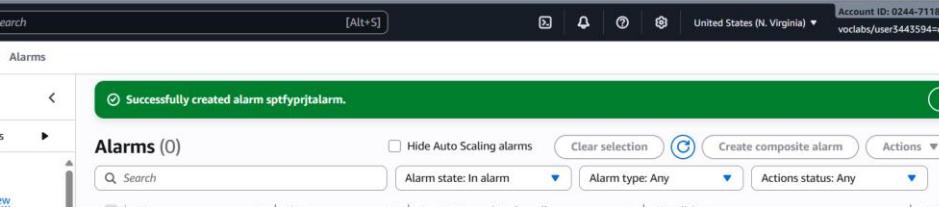
Use topic ARN to notify other accounts

Send a notification to...

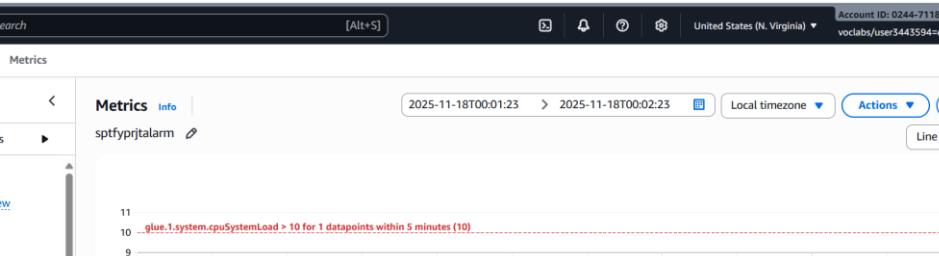
Default_CloudWatch_Alarms_Topic

Only topics belonging to this account are listed here. All persons and applications subscribed to the selected topic will receive notifications.

Email (endpoints)
e23cseu2050@bennett.edu.in - [View in SNS Console](#)



The screenshot shows the AWS CloudWatch Alarms page. The left sidebar is collapsed, and the main content area displays a green success message: "Successfully created alarm sptfypjtlalarm." Below this, the "Alarms (0)" section is shown. The page includes a search bar, filter buttons for "Hide Auto Scaling alarms", "Clear selection", "Create composite alarm", and "Actions", and a prominent "Create alarm" button. The main table area is empty, showing the message "No alarms" and a link to "Read more about Alarms".



CloudWatch Metrics

Metrics Info

2025-11-18T00:01:23 > 2025-11-18T00:02:23 Local timezone Actions Investigate

Account ID: 0244-7118-4469
vocabs/user3443594e23cse2050@bennett.edu.in

Line

11
10 glue.1.system.cpuSystemLoad > 10 for 1 datapoints within 5 minutes (10)
9

00:01:00 00:01:05 00:01:10 00:01:15 00:01:20 00:01:25 00:01:30 00:01:35 00:01:40 00:01:45 00:01:50 00:01:55 00:02:00

Browse Multi source query Graphed metrics (1) Options Source Add math Add query

Add dynamic label Info Statistic: Average Period: 1 minute Clear graph

Label	Details	Statistic	Period	Y axis	Actions
<input checked="" type="checkbox"/> glue.1.system.cpuSystemLoad	Glue • glue.1.system.cpuSystemLoad • Ty	Average	1 minute	< >	Edit Delete Copy Share

Conclusion

This document serves as a runbook for the **Spotify Data Engineering Project**. Follow each step to set up and run your end-to-end data pipeline.

Real-World Applications

The insights gained from this project can be applied in various real-world scenarios:

1. **Music Recommendations:** By analyzing the popularity of tracks and artists, platforms can improve their recommendation engines.
2. **Market Analysis:** Record labels and music producers can use the data to understand trends and make informed decisions on which genres or artists to promote.
3. **User Engagement:** Streaming services can analyze user preferences and behavior to enhance user engagement through personalized playlists and features.
4. **Business Intelligence:** The visualizations and insights derived can help business analysts make data-driven decisions to optimize marketing strategies and improve user retention