

# **Implementing a Streamlined Real-time Data Streaming and Analytics Pipeline with AWS Kinesis, Firehose, Data Streams, AWS kinesis Analytics Application, Glue Crawler, Glue ETL, and Athena for Querying S3-backed Databases.**

Naveen Raju Sreerama Raju Govinda Raju  
[naveenraju100@gmail.com](mailto:naveenraju100@gmail.com) | +1 224 7067718 | +91 9886157101 |  
<https://www.linkedin.com/in/naveen-raju-s-g-bb1486124>  
<https://naveenrajusg.github.io/Portfolio/>

## **Table of Contents**

- 1) Overview**
- 2) S3 bucket creation**
- 3) Delivery stream creation for Data FireHose**
- 4) Testing delivery stream by sending demo data**
- 5) Create and configure Amazon kinesis Analytics applications**
  - A) Create SQL application**
    - I) Configure Source stream
    - II) Configure Run real time analytics with SQL code
    - III) Configure destination
    - IV) Create delivery stream
    - V) Configure destination
  - B) View results in S3 buckets**
- 6) Configure, create and run AWS Glue crawler**
  - I) Set Crawler properties
  - II) Choose data sources and classifiers
  - III) Configure security settings
  - IV) Set output and Scheduling
  - V) Create and run Crawler
  - VI) View results in AWS Glue tables
- 7) Configure, create and run AWS Glue studio ETL job**
  - I) Configure Data Source-S3 bucket
  - II) Configure Transform-Apply Mapping
  - III) Configure Data target - s3 bucket
  - IV) Configure Job details
  - V) View results in S3 buckets
- 8) Configure, create and run AWS Athena interactive query service**
  - I) Launch query editor and edit settings
  - II) Config editor
  - III) Write queries and view results

## **9) Deleting resources**

- A) Stop the application created**
- B) Delete delivery stream (stream that delivers results to S3 buckets)**
- C) Delete delivery stream (ingest data from producers that are configured to send data to Kinesis Data FireHose)**
- D) Delete S3 buckets**
- E) Delete SQL application (Analytics applications)**
- F) Delete Crawler**
- G) Delete AWS Glue Studio Job**
- H) Delete AWS Glue Databases**
- I) Delete S3 buckets**

Please zoom in to view the screenshots clearly

## **1) Overview**

Amazon Kinesis is a suite of real-time data streaming services provided by Amazon Web Services (AWS). It is designed to make it easy for developers to collect, process, and analyze large amounts of streaming data in real-time. Core components of Amazon Kinesis include Amazon Kinesis Streams, Amazon Kinesis Firehose, Amazon Kinesis Analytics, Amazon Kinesis Video Streams.

AWS Glue is a fully managed Extract, Transform, and Load (ETL) service provided by Amazon Web Services (AWS). It is designed to help users efficiently and securely prepare and load their data for analytics and data-driven tasks. AWS Glue can handle both structured and semi-structured data, making it ideal for processing diverse data sources.

Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 and other federated data sources using standard SQL.

## **2) S3 bucket creation**

Create a s3 bucket named ticker-123.

The screenshot shows the AWS S3 console interface. At the top, there is a green banner message stating "Successfully created bucket 'ticker-123'. To upload files and folders, or to configure additional bucket settings choose View details." Below this, the main S3 dashboard displays an "Account snapshot" section and a table of existing buckets. The table includes columns for Name, AWS Region, Access, and Creation date. Two buckets are listed: "aws-logs-207468113308-us-east-2" (Region: US East (Ohio) us-east-2, Access: Bucket and objects not public, Created: April 27, 2023, 20:29:20 (UTC-05:00)) and "ticker-123" (Region: Asia Pacific (Mumbai) ap-south-1, Access: Bucket and objects not public, Created: May 18, 2023, 03:30:30 (UTC-05:00)). A "Create bucket" button is visible at the top right of the bucket list area.

## **3) Delivery stream creation for Data FireHose**

Delivery stream ingest data from producers that are configured to send data to Kinesis Data FireHose. It can also optionally transform records using AWS lambda functions and finally it will deliver data to configured destination in our case it is S3 bucket.

In the AWS console navigate to Amazon Kinesis --> Data Firehose --> Create delivery stream.

Delivery stream configuration:

Source : Direct PUT (which means we have data source that generates data in real-time)

Destination : Amazon S3

Delivery stream name : ml-raju

Destination settings -> S3 bucket : s3://ticker-123 (path to s3 bucket created)

Dynamic partitioning : Not enabled

(Dynamic partitioning enables to create targeted datasets by partitioning streaming data based on partitioning keys. We can do this with help of AWS Lambda function.)

S3 bucket prefix : ticker/demo

(By default, kinesis Data Firehouse appends prefix “YYYY/MM/dd/HH” (in UTC) to the data it delivers to Amazon S3. We can override this default by specifying a custom prefix.)

S3 bucket error output prefix : ticker/error

Buffer size : 1Mb

(Kinesis Data Firehouse buffers incoming records before delivering to S3 buckets. Record delivery is triggered once the set size limit is reached. Higher buffer size may be lower in cost with high latency. The lower buffer size will be faster in delivery with higher cost and less latency.)

Buffer interval : 60 seconds

(The higher interval allows more time to collect data and size of data may be bigger. The lower interval sends the data more frequently and may be more advantageous when looking at shorter cycles of data activity.)

Compressionsn for Data records : Not enabled

(Kinesis Data Firehouse can compress records before delivering them to S3 buckets.)

Encryptions for data records : Not enabled

(Compress record gets encrypted in the s3 bucket using KMS master key.)

Amazon CloudWatch: Enabled

(Kinesis Data FireHose to log record delivery errors to CloudWatch Logs)

The screenshot shows the AWS CloudShell interface with the following details:

- Header:** AWS Services Search [Alt+S] Mumbai Naveen Raju S.G.
- Breadcrumbs:** Amazon Kinesis > Data Firehose > Create delivery stream
- Title:** Create delivery stream [Info](#)
- Section: Amazon Kinesis Data Firehose: How it works**
  - Input:** Configure your data producers (see below) to send data to Kinesis Data Firehose.
  - Transform (optional):** Transform source records using an AWS Lambda function. You can also convert the record format.
  - Load:** Deliver data to a specified destination, including Amazon S3, Amazon OpenSearch Service, Amazon Redshift, and various HTTP endpoint destinations.
- Data producers:** Amazon Kinesis Data Streams, Amazon Kinesis Agent Tap, Amazon and AWS Services (CloudWatch, IoT Core, DeviceFarm, etc), Direct PUT, and SDK API for custom Apps.
- Choose source and destination:** Specify the source and the destination for your delivery stream. You cannot change the source and destination of your delivery stream once it has been created.
  - Source:** [Info](#) Choose a source
  - Destination:** [Info](#) Choose a destination
- Buttons:** Cancel, **Create delivery stream** (highlighted in orange).

Services Search [Alt+S]

Services Search [Alt+S]

**Source** Info

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

Source: Info Direct PUT Info

Destination: Info Amazon S3 Info

**Delivery stream name**

Delivery stream name: mri-raju Info

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

**Transform and convert records - optional**

Configure Kinesis Data Firehose to transform and convert your record data.

**Transform source records with AWS Lambda** Info

Kinesis Data Firehose can invoke an AWS Lambda function to transform, filter, un-compress, convert and process your source data records. The specified AWS Lambda function can also be used to provide dynamic partitioning keys for the incoming source data before its delivery to the specified destination.

Enable data transformation

**Convert record format** Info

Data in Apache Parquet or Apache ORC format is typically more efficient to query than JSON. Kinesis Data Firehose can convert your JSON-formatted data into Apache Parquet or Apache ORC format. For records that aren't in JSON format, create a Lambda function that converts them to JSON in the transform source records with AWS Lambda section above.

Enable record format conversion

**Destination settings** Info

Specify the destination settings for your delivery stream.

**S3 bucket**

S3 bucket: \$3://ticker-123 Browse Create

Format: \$3//\$bucket

**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 delivery stream. You cannot enable dynamic partitioning for an existing delivery stream. Enabling dynamic partitioning adds costs per GB of partitioned data. For more information, see Kinesis Data Firehose pricing.

Not enabled

Enabled

**S3 bucket prefix - optional**

By default, Kinesis 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.

Enter a prefix

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 a S3 bucket error output prefix to be used in error conditions. This prefix can include expressions for Kinesis Data Firehose to evaluate at runtime.

Enter a prefix

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

**Buffer hints, compression and encryption**

The fields below are pre-populated with the recommended default values for S3. Pricing may vary depending on storage and request costs.

**Advanced settings**

Server-side encryption not enabled; error logging enabled; IAM role KinesisFirehoseServiceRole-raj-raj-ap-south-1-164859859108; no tags.

**Destination settings** Info

Specify the destination settings for your delivery stream.

**S3 bucket**

S3 bucket: \$3://ticker-123 Browse Create

Format: \$3//\$bucket

**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 delivery stream. You cannot enable dynamic partitioning for an existing delivery stream. Enabling dynamic partitioning adds costs per GB of partitioned data. For more information, see Kinesis Data Firehose pricing.

Enable record format conversion

**Destination settings** Info

Specify the destination settings for your delivery stream.

**S3 bucket**

S3 bucket: \$3://ticker-123 Browse Create

Format: \$3//\$bucket

**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 delivery stream. You cannot enable dynamic partitioning for an existing delivery stream. Enabling dynamic partitioning adds costs per GB of partitioned data. For more information, see Kinesis Data Firehose pricing.

Not enabled

Enabled

**S3 bucket prefix - optional**

By default, Kinesis 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.

ticker/demo

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 a S3 bucket error output prefix to be used in error conditions. This prefix can include expressions for Kinesis Data Firehose to evaluate at runtime.

ticker/error

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

**Buffer hints, compression and encryption**

The fields below are pre-populated with the recommended default values for S3. Pricing may vary depending on storage and request costs.

**S3 buffer hints**

Kinesis Data Firehose buffers incoming records before delivering them to your S3 bucket. Record delivery is triggered once the value of either of the specified buffering hints is reached.

**Buffer size**

The higher buffer size may be lower in cost with higher latency. The lower buffer size will be faster in delivery with higher cost and less latency.

1 MB MIB

Minimum: 1 MB; maximum: 128 MB. Recommended: 5 MB.

**Buffer interval**

The higher interval allows more time to collect data and the size of data may be larger. The lower interval sends the data more frequently and may be more advantageous when looking at shorter cycles of data activity.

60 seconds seconds

Minimum: 60 seconds; maximum: 900 seconds. Recommended: 300 seconds.

Kinesis Data Firehose uses AWS Lambda to process your data. You can specify an AWS Lambda function as a target for your delivery stream.

**Delivery stream name**

Delivery stream names must be unique across all Kinesis Data Firehose delivery streams in your account. The name must contain between 3 and 128 characters, and can't contain spaces or punctuation.

**Delivery stream type**

Delivery stream type defines the destination where your data is sent. You can choose to send data to an Amazon S3 bucket or an AWS Lambda function.

**S3 bucket**

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

**Enter a prefix**

**Buffer hints, compression and encryption**

The fields below are pre-populated with the recommended default values for S3. Pricing may vary depending on storage and request costs.

**S3 buffer hints**

Kinesis Data Firehose incoming records before delivering them to your S3 bucket. Record delivery is triggered once the value of either the sum of the specified buffers or hints is reached.

**Buffer size**

The higher buffer size may be lower in cost with higher latency. The lower buffer size will be faster in delivery with higher cost and less latency.

1	MB
---	----

Minimum: 1 MB; maximum: 128 MB. Recommended: 5 MB.

**Buffer interval**

The higher interval allows more time to collect data and the size of data may be bigger. The lower interval sends the data more frequently and may be more advantageous when looking at shorter cycles of data activity.

60	seconds
----	---------

Minimum: 60 seconds; maximum: 900 seconds. Recommended: 300 seconds.

**S3 compression and encryption**

Kinesis Data Firehose can compress records before delivering them to your S3 bucket. Compressed records can also be encrypted in the S3 bucket using an AWS Key Management Service (KMS) master key.

**Compression for data records**

Kinesis Data Firehose can compress records before delivering them to your S3 bucket.

Not enabled

- GZIP
- Snappy
- Zip
- Hadoop-Compatible Snappy

**Encryption for data records**

Compressed record gets encrypted in the S3 bucket using a KMS master key.

Not enabled

Enabled

**Advanced settings**

Server-side encryption not enabled, error logging enabled, IAM role KinesisFirehoseServiceRole-ml-raju-ap-south-1-1684398559108, no tags.

**Server-side encryption**

You can use AWS Key Management Service (KMS) to create and manage Customer Master Keys (CMK) and to control the use of encryption keys in a wide range of AWS services in your applications.

Enable server-side encryption for source records in delivery stream

**Amazon CloudWatch error logging**

CloudWatch error logging is disabled if you want Kinesis Data Firehose to log record delivery errors to CloudWatch Logs.

Not enabled

Enabled

**Service access**

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

Create or update IAM role KinesisFirehoseServiceRole-ml-raju-ap-south-1-1684398559108

Create a new role or updates an existing one and adds the required policy to it, and enables Kinesis Data Firehose to assume it.

Choose existing IAM role

The role that you chose must have policies that include the permissions that Kinesis Data Firehose needs.

**Tags**

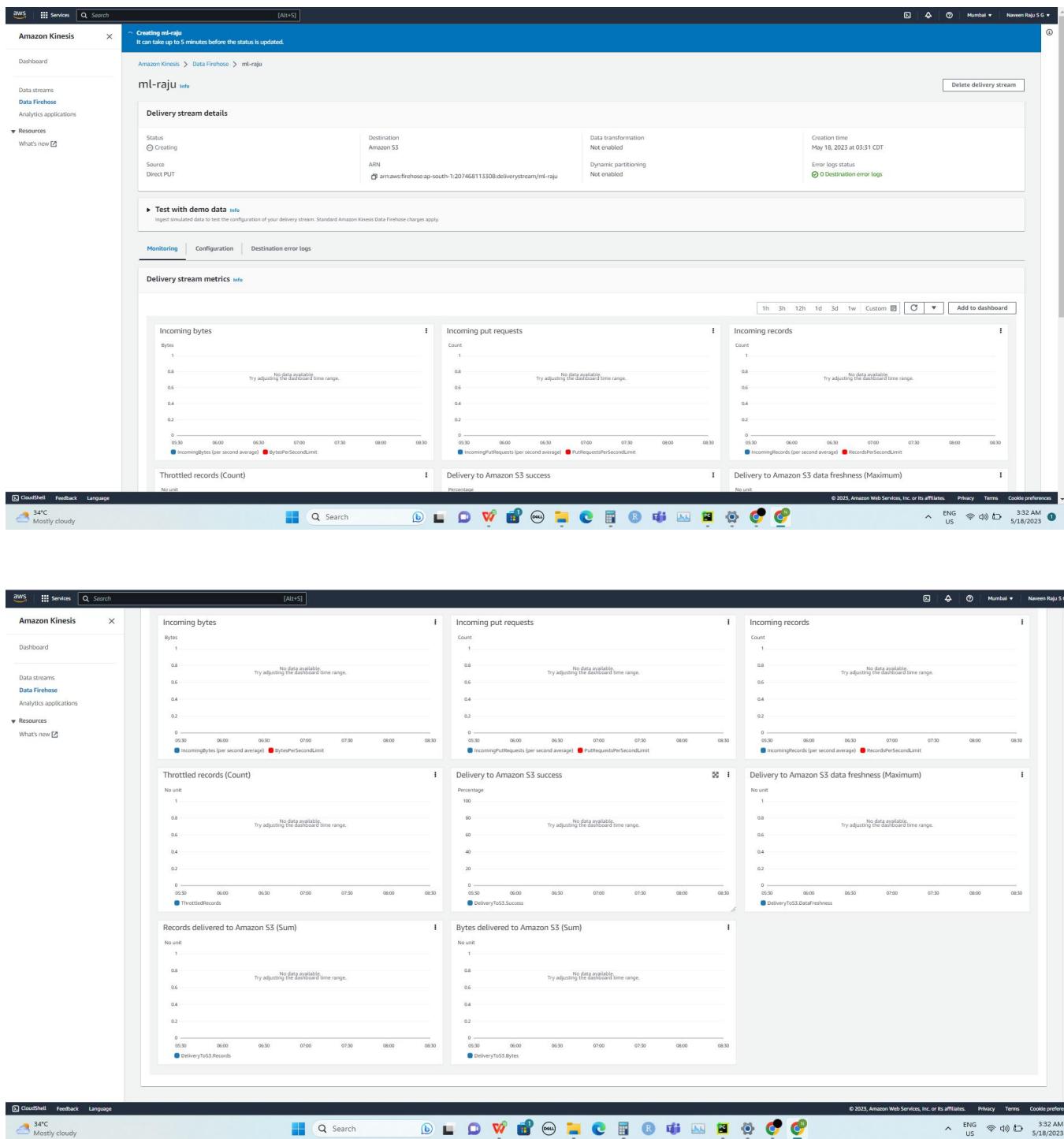
You can add tags to organize your AWS resources, track costs, and control access.

No tags associated with the resource.

**Add new tag**

You can add up to 50 more tags.

## Delivery stream has been created :



The screenshot shows the AWS Kinesis Data Firehose console. A banner at the top introduces the new Kinesis Data Firehose console experience. Below it, a message says 'ml-raju was successfully created.' The main area displays 'Delivery stream details' for 'ml-raju'. It shows the status as 'Active', destination as 'Amazon S3', and ARN as 'arn:aws:firehose:ap-south-1:207468113308:deliverystream/ml-raju'. The 'Data transformation' and 'Dynamic partitioning' are both set to 'Not enabled'. The 'Creation time' is 'May 18, 2023 at 03:37 CDT'. Under the 'Monitoring' tab, there are three line charts: 'Incoming bytes' (Bytes/s), 'Incoming put requests' (Count), and 'Incoming records' (Count). The 'Incoming bytes' chart has a Y-axis from 0 to 1,000 and an X-axis from 06:00 to 08:30. The 'Incoming put requests' chart has a Y-axis from 0 to 1,000 and an X-axis from 06:00 to 08:30. The 'Incoming records' chart has a Y-axis from 0 to 100k and an X-axis from 06:00 to 08:30.

## 4) Testing delivery stream by sending demo data

After delivery stream is created click on drop down “Test with demo data”. This runs a script in the browser to put demo data in Kinesis Data FireHose delivery stream, which then sends to the Amazon S3 destination.

The screenshot shows the 'Test with demo data' section of the AWS Kinesis Data Firehose console. It includes a sample JSON script for generating demo data:

```

1 {
2   "TICKER_SYMBOL": "QXZ",
3   "SECTOR": "HEALTHCARE",
4   "CHANGE": 0.05,
5   "PRICE": 84.51
6 }

```

Below the script are two sections: 'Step 1' and 'Step 2'. 'Step 1' contains a button 'Start sending demo data'. 'Step 2' contains a button 'Stop sending demo data'. At the bottom, there is a 'Destination error logs details' section with a link to 'Log group in Amazon CloudWatch Logs' and a log stream 'LogStreamName'. The status is shown as '0 Destination error logs'.

Click on start sending demo data.

The screenshot shows the AWS Kinesis Data Firehose console interface. The main header bar includes the AWS logo, Services, a search bar, and navigation links for Mumbai and Naveen Raju SG. The left sidebar has a tree view with 'Amazon Kinesis' selected, under which 'Data streams' and 'Data Firehose' are listed. Below that are 'Analytics applications', 'Resources', and 'What's new'. The main content area is titled 'ml-raju' and shows 'Delivery stream details'. It lists the status as 'Active', the destination as 'Amazon S3', and the ARN as 'arn:aws:kinesisfirehose:ap-south-1:207468113308:deliverystream/ml-raju'. It also indicates 'Data transformation Not enabled' and 'Dynamic partitioning Not enabled'. The creation time is 'May 18, 2023 at 03:37 CDT' and there are '0 Destination error logs'. A section titled 'Test with demo data' contains a script to ingest simulated data and a 'Sending demo data' button. Below this is a 'Step 1' section with a 'Start sending demo data' button and a 'Step 2' section with a 'Stop sending demo data' button. The browser address bar shows the URL 'https://ap-south-1.console.aws.amazon.com/kinesisanalytics/home?region=ap-south-1#applications/dashboard'. The system tray at the bottom shows the date and time as '5/8/2023 3:38 AM'.

## 5) Create and configure Amazon kinesis Analytics applications

### A) Create SQL application

Create and configure Amazon kinesis Analytics applications

In AWS console navigate to Amazon kinesis -> Analytics applications -> SQL applications

Click on Create SQL application.

Config : Application name : tiger-analytics.

Click on Create Legacy SQL application.

Kinesis Data Analytics continuously reads and analyzes data from a connected streaming source in real time.

aws Services Search [Alt+S] Mumbai Navneet Raju SG v

**Amazon Kinesis**

Kinesis Data Analytics Blueprints for Apache Flink

Kinesis Data Analytics Blueprints are a curated collection of Apache Flink applications which is extensible and can be leveraged to create more complex applications to solve your business challenges in Apache Flink.

Blueprints

Dashboard

Data streams

Data Firehose

Analytics applications

Streaming applications

Studio notebooks

SQL applications (legacy)

Resources

What's new

AWS Streaming Data Solution for Amazon Kinesis

AWS Glue Schema Registry

Customer survey

Amazon Kinesis > SQL applications (legacy)

SQL applications (legacy) (0) info Run Stop Delete Create SQL application (legacy)

Find SQL applications (legacy) Application name Last updated Status Create SQL application (legacy)

No SQL applications (legacy)

For new applications, we recommend that you use Kinesis Data Analytics Studio instead of Kinesis Data Analytics for legacy SQL applications for running SQL queries. Kinesis Data Analytics Studio provides advanced analytical capabilities, enabling you to build sophisticated stream processing applications in minutes. Learn more

Create legacy SQL application info

Kinesis Data Analytics continuously reads and analyzes data from a connected streaming source in real time. Kinesis Data Analytics resources are not covered under the AWS Free Tier, and usage-based charges apply. For more information, see Kinesis Data Analytics pricing

Application configuration

Application name: ticker-analytics

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

Description - optional

Enter description

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs. Learn more

No tags associated with this application

Add tag You can add up to 50 tags.

Cancel Create legacy SQL application

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG US 3:39 AM 5/18/2023

aws Services Search [Alt+S] Mumbai Navneet Raju SG v

**Amazon Kinesis**

Kinesis Data Analytics Blueprints for Apache Flink

Kinesis Data Analytics Blueprints are a curated collection of Apache Flink applications which is extensible and can be leveraged to create more complex applications to solve your business challenges in Apache Flink.

Blueprints

Dashboard

Data streams

Data Firehose

Analytics applications

Streaming applications

Studio notebooks

SQL applications (legacy)

Resources

What's new

AWS Streaming Data Solution for Amazon Kinesis

AWS Glue Schema Registry

Customer survey

Amazon Kinesis > SQL applications (legacy) > ticker-analytics

ticker-analytics

Steps to configure your application info

Application details

Status: Ready	ARN: arn:aws:kinesisanalytics:south-1:207468113308:application/ticker-analytics	Runtime: SQL
Application version ID: 1	Last updated: May 18, 2023 at 03:39 CDT	Description: -
Created time: May 18, 2023 at 03:39 CDT		

Source | Real-time analytics | Destinations | Tags

Source stream info

Configure

Source: -	Stream: -	In-application stream name: -
ID: -	Record preprocessing: -	IAM role for reading source stream: -

Reference data info

Remove | Configure

CloudShell Feedback Language © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG US 3:40 AM 5/18/2023

After application is created click on “Steps to configure the application” drop down and then select “Configure source stream”

The screenshot shows the AWS Kinesis Data Analytics console with the 'ticker-analytics' application selected. In the main content area, there's a section titled 'Steps to configure your application' with three steps: 'Step 1 Configure source stream', 'Step 2 Run real-time analytics with your SQL code', and 'Step 3 Configure destinations'. The 'Configure source stream' button is highlighted with an orange box. Below this, there's a 'Application details' section with various metadata like ARN, Status, and Last updated. At the bottom of the configuration section, there are tabs for 'Source', 'Real-time analytics', 'Destinations', and 'Tags', with 'Source' being the active tab.

## I) Configure Source stream

Now we should configure data source for the application

Source : Kinesis DataFirehose delivery stream  
Delivery stream : ml-raju

Schema -> click on Discover schema (Schema discovery can generate a schema using records from the source. Schema column names are the same as in source, unless they contain special characters, repeated column names, or reserved keywords.)

The screenshot shows the 'Configure source for ticker-analytics' configuration page. It has sections for 'Source' (with options for Kinesis data stream or Kinesis Data Firehose delivery stream), 'Delivery stream' (set to 'arn:aws:firehose:ap-south-1:207468113308:deliverystream/ml-raju'), 'Record preprocessing with AWS Lambda' (set to 'Off'), and 'IAM role for reading source stream' (with options for creating a new role or choosing an existing one). At the bottom, there's a 'Schema' section with a 'Discover schema' button, which is highlighted with an orange box. A status message below it says 'Discovering schema for stream arn:aws:firehose:ap-south-1:207468113308:deliverystream/ml-raju with starting point { startingPoint }'.

**IAM role for reading source stream**

Create or choose IAM role with the required permissions. Learn more [Create / update IAM role kinesis-analytics-ticker-analytics-ap-south-1 with required policies](#)

Off

On

**Schema info**

Schema discovery can generate a schema from the source. Schema column names are the same as in the source, unless they contain special characters, repeated column names, or reserved keywords.

**Schema was successfully discovered.**

Use discovered schema

Customize schema

**Schema preview**

**Schema (5)**

CHANGE	PRICE	TICKER_SYMBOL	SECTOR
0.99	18.16	AZL	HEALTHCARE
1.62	757.13	AMZN	TECHNOLOGY
-1.03	23.75	ABC	RETAIL
17.59	222.59	HUV	ENERGY
3.34	58.91	SAC	ENERGY

**Actions**

Cancel Save changes

## II) Configure Run real time analytics with SQL code

On the same web page navigate to “Steps to configure your application” -> “Run real time analytics with your SQL code”

Click on Configure SQL

**Successfully created application ticker-analytics.**

**Kinesis Data Analytics Blueprints for Apache Flink**

Kinesis Data Analytics Blueprints are a curated collection of Apache Flink applications which is extensible and can be leveraged to create more complex applications to solve your business challenges in Apache Flink.

**Application ticker-analytics has been successfully updated.**

**Steps to configure your application**

**Step 1** Configure source stream

Choose an existing Kinesis data stream or Kinesis Data Firehose delivery stream as input. Kinesis Data Analytics ingests the data, automatically recognizes standard data formats, and suggests a schema.

**Step 2** Run real-time analytics with your SQL code

Use the Kinesis Data Analytics SQL editor and built-in templates to write queries to process streaming data.

**Step 3** Configure destinations

Point to the destinations where you want the results of your SQL code to be loaded.

**Configure SQL**

**Application details**

Status: Ready

ARN: arn:aws:kinesisanalytics:ap-south-1:207468113308:application/ticker-analytics

Runtime: SQL

Application version ID: 2

Last updated: May 18, 2023 at 03:46 CDT

Created time: May 18, 2023 at 03:39 CDT

Description: -

**Source** **Real-time analytics** **Destinations** **Tags**

**Source stream**

Connect to an existing Kinesis data stream or Firehose delivery stream, or easily create and connect to a new demo Kinesis data stream. Each application can connect to one streaming data source.

**Configure**

Click on the button “Add SQL from templates” - > “Aggregate function in sliding time window”

The screenshot shows the 'Configure SQL code for ticker-analytics' page. At the top, there's a success message: 'Successfully created application ticker-analytics.' Below it, a note about 'Kinesis Data Analytics Blueprints for Apache Flink'. The main area is titled 'Configure SQL code for ticker-analytics' with a sub-section 'SQL code info'. A code editor contains the following SQL template:

```
1 /*  
2 * Welcome to the SQL editor  
3 * -----  
4 *  
5 * The SQL code you write here will continuously transform your streaming data  
6 * when your application is running.  
7 *  
8 * Get started by clicking "Add SQL from templates" or pull up the  
9 * documentation and start writing your own custom queries.  
10 */  
11
```

Below the code editor are tabs for 'View raw SQL', 'Add SQL from templates', 'Save application', and 'Save and run application'. The status bar shows 'SQL' and 'Ln 1, Col 1' with 'Errors: 0' and 'Warnings: 0'. Below this, there are tabs for 'Output' (selected) and 'Input'. Under 'Output', there's a section for 'Output streams (0)' with a search bar and a 'Connect to destination' button. The bottom of the screen shows the AWS navigation bar and system tray.

The screenshot shows the same 'Configure SQL code for ticker-analytics' page, but the 'Add SQL code from templates' section is now active. It lists various template options under 'SQL template':

- Choose template
- Continuous Filter
- Aggregate function in a tumbling time window
- Aggregate function in a sliding time window** (this option is selected)
- Aggregate function in a sliding row window
- Multi-step application
- Anomaly detection
- Approximate top-K items
- Approximate distinct count
- Data enrichment (join)
- Aggregate using two time windows
- Simple alert
- Parse and aggregate Apache logs

The bottom of the screen shows the AWS navigation bar and system tray.

The screenshot shows the AWS CloudShell interface with the Amazon Kinesis service selected. On the left, there's a sidebar with 'Resources' and 'AWS Streaming Data Solution for Amazon Kinesis'. The main content area is titled 'Configure SQL code' and contains a code editor with the following SQL template:

```

1 -- ** Aggregate (COUNT, AVG, etc.) + Sliding time window **
2 -- .. Perform functions on the aggregate rows over a 10 second sliding window for a specified column.
3 --
4 --          | SOURCE   |-->| INSERT  |-->| DESTIN. |
5 -- Source->| STREAM  |-->| PUMP    |-->| STREAM  |-->Destination
6 --
7 --
8 -- STREAM (in-application): a continuously updated entity that you can SELECT from and INSERT into like a TABLE
9 -- PUMP: an entity used to continuously 'SELECT ... FROM' a source STREAM, and INSERT SQL results into an output STREAM
10 -- Create output stream, which can be used to send data to a destination
11 CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM" (ticker_symbol VARCHAR(4), ticker_symbol_count INTEGER);
12 -- Create a pump which continuously selects from a source stream (SOURCE_SQL_STREAM_001)
13 -- performs an aggregate count that is grouped by columns ticker over a 10-second sliding window
14 CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM"
15     SELECT STREAM ticker_symbol, COUNT(*) OVER TEN_SECONDS_SLIDING_WINDOW AS ticker_symbol_count
16     FROM "SOURCE_SQL_STREAM_001"
17     -- Results partitioned by ticker_symbol and a 10-second sliding time window
18     PARTITION BY ticker_symbol
19     RANGE INTERVAL '10' SECOND PRECEDING;
20
21 
```

Below the code editor are buttons for 'Cancel', 'Replace current SQL', and 'Append SQL to the editor'. The bottom of the screen shows the AWS CloudShell toolbar with various icons and status information.

## Description of SQL code:

This template performs aggregate on rows over a 10 seconds sliding window.

Basically the application has 3 parts : Source stream, Select and Insert (Pump), create a output stream which can be sent to a destination.

STREAM(in-application): a continuously updated entity that we select from and insert into Table

PUMP : an entity used to continuously ‘SELECT … FROM’ a source STREAM, and insert SQL results into like a TABLE

Create a output stream, which can be used to send to a destination

[Below line - Create pump which continuously selects from a source stream (SOURCE\_SQL\_STREAM\_001) and performs an aggregate count that is grouped by columns ticker over a 10-second sliding window]

CREATE OR REPLACE STREAM “DESTINATION\_SQL\_STREAM” (ticker\_symbol VARCHAR(4), ticker\_symbol\_count INTEGER);

[Below line does COUNT | AVG | MAX | MIN | SUM | STDDEV\_POP | STDDEV\_SAMP | VAR\_POP | VAR\_SAMP]

CREATE OR REPLACE PUMP “STREAM\_PUMP” AS INSERT INTO “DESTINATION\_SQL\_STREAM”

SELECT STREAM ticker\_symbol, COUNT(\*) OVER TEN\_SECONDS\_SLIDING\_WINDOW AS ticker\_symbol\_count  
FROM “SOURCE\_SQL\_STREAM\_001”

[The below code create results partitioned by ticker\_symbol and a 10-second sliding time window]

WINDOW TEN\_SECOND\_SLIDING\_WINDOW AS (  
 PARTITION BY ticker\_symbol  
 RANGE INTERVAL '10' SECOND PRECEDING);

Now click on “Save and run application”

The screenshot shows the AWS Kinesis Data Analytics console. The application 'ticker-analytics' has been successfully created and updated. The SQL code defines a windowed aggregate query that inserts data into a destination stream. The output stream is set to 'DESTINATION\_SQL\_STREAM'. The AWS navigation bar includes 'CloudShell', 'Feedback', 'Language', and the date '5/18/2023'. The status bar at the bottom right shows '3:49 AM'.

### III) Configure destination

On the same web page navigate to “Steps to configure your application” -> “Configure destination”

The screenshot shows the 'Steps to configure your application' section. Step 1: 'Configure source stream' is marked as 'Configured'. Step 2: 'Run real-time analytics with your SQL code' is marked as 'Configured'. Step 3: 'Configure destinations' is shown with a 'Add destination' button. The 'Application details' section shows the ARN, runtime (SQL), and description. The 'Source stream' section shows the source as 'Kinesis Data Firehose' and the stream name as 'SOURCE\_SQL\_STREAM\_001'. The AWS navigation bar includes 'CloudShell', 'Feedback', 'Language', and the date '5/18/2023'. The status bar at the bottom right shows '3:51 AM'.

## **IV) Create delivery stream:**

Navigate through Amazon Kinesis -> Data Firehose -> Create delivery stream

Source : Direct PUT (which means we have data source that generates data in real-time)

Destination : Amazon S3

Delivery stream name : tiger-analytics

Destination settings -> S3 bucket : s3://ticker-123 (path to s3 bucket created)

S3 bucket prefix : tiger\_analytics/

(By default, kinesis Data Firehose appends prefix “YYYY/MM/dd/HH” (in UTC) to the data it delivers to Amazon S3. We can override this default by specifying a custom prefix.)

S3 bucket error output prefix : tiger\_analytics\_error/

Buffer size : 1Mb

Buffer interval : 60 seconds

Compressionsn for Data records : Not enabled

Encryptions for data records : Not enabled

Amazon CloudWatch: Enabled

(Kinesis Data FireHose to log record delivery errors to CloudWatch Logs)

The screenshot shows the 'Create delivery stream' wizard in the AWS Management Console. The steps completed are 'Choose source and destination' and 'Delivery stream name'. The 'Source' is set to 'Direct PUT' and the 'Destination' is 'Amazon S3'. The 'Delivery stream name' is 'ticker\_analytics'. Under 'Transform and convert records - optional', there is a section for 'Transform source records with AWS Lambda'. The 'Convert record format' section is also visible. At the bottom, the 'Destination settings' section is partially visible.

The screenshot shows the AWS CloudShell interface with multiple tabs open. The active tab is titled "Amazon Kinesis Data Firehose" and displays the configuration for creating a new delivery stream. The configuration includes:

- Delivery stream name:** ticker\_analytics
- Transform and convert records - optional:** This section contains options for AWS Lambda transformation, including "Enable data transformation" and "Enable record format conversion".
- Destination settings:** Set to "Amazon S3".
  - S3 bucket:** Choose a bucket or enter a bucket URI: ticker-123.
  - Format:** s3://bucket

At the bottom right of the CloudShell interface, there is a status bar showing "CloudShell Feedback Language" and system information like "34°C Mostly cloudy" and "CloudShell ENG US 3:52 AM 5/18/2023".

This screenshot shows the continuation of the delivery stream configuration. The "Create delivery stream" button is visible at the bottom of the page.

**S3 buffer hints**

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

**Buffer hints/error**

**Buffer hints, compression and encryption**

The fields below are pre-populated with the recommended default values for S3. Pricing may vary depending on storage and request costs.

**S3 buffer hints**

Kinesis Data Firehose buffers incoming records before delivering them to your S3 bucket. Record delivery is triggered once the value of either of the specified buffering hints is reached.

**Buffer size**

The higher buffer size may be lower in cost with higher latency. The lower buffer size will be faster in delivery with higher cost and less latency.

**Buffer interval**

The higher interval allows more time to collect data and the size of data may be bigger. The lower interval sends the data more frequently and may be more advantageous when looking at shorter cycles of data activity.

**S3 compression and encryption**

Kinesis Data Firehose can compress records before delivering them to your S3 bucket. Compressed records can also be encrypted in the S3 bucket using an AWS Key Management Service (KMS) master key.

**Compression for data records**

Kinesis Data Firehose can compress records before delivering them to your S3 bucket.

Not enabled

GZIP

Snappy

Zip

Hadoop-Compatible Snappy

**Encryption for data records**

Compressed record gets encrypted in the S3 bucket using a KMS master key.

Not enabled

Enabled

**Advanced settings**

Server-side encryption not enabled; error logging enabled; IAM role KinesisFirehoseDeliveryRole-ticker\_analytic-ap-south-1-1048579938414; no tags.

**Create delivery stream**

## V) Configure destination

Continue “Steps to configure your application” -> “Configure destination”

Set delivery stream to tiger\_analytics stream that we created

Under In\_Application stream name section :

Select Choose an existing in-application stream -> In\_application stream name -> DESTINATION\_SQL\_STREAM

Output format : JSON

Now a Analytics application has been created:

**Kinesis Data Analytics Blueprints for Apache Flink**

Kinesis Data Analytics Blueprints are a curated collection of Apache Flink applications which is extensible and can be leveraged to create more complex applications to solve your business challenges in Apache Flink.

**Application ticker-analytics has been successfully updated.**

**ticker-analytics**

**Steps to configure your application**

**Application details**

Status: <b>Running</b>	ARN: arn:aws:kinesisanalytics:ap-south-1:207468113308:application/ticker-analytics	Runtime: SQL
Application version ID: 4	Last updated: May 18, 2023 at 03:56 CDT	Description: -
Created time: May 18, 2023 at 03:59 CDT		

**Destinations (1)**

Destination AWS service	Destination Resource	In-application stream	ID
Kinesis Firehose Streams	ticker_analytics	DESTINATION_SQL_STREAM	4.1

Navigate to application that we created then in Configure SQL code section select Destination SQL stream in the output section, here we can see the results being generated.

aws Services Search [Alt+5] Mumbai Naveen Raj S G 5

Amazon Kinesis X

Dashboard

Data streams

Data Firehose

Analytics applications

Streaming applications

Studio notebooks

SQL applications (legacy)

Resources

What's new AWS Streaming Data Solution for Amazon Kinesis AWS Glue Schema Registry Customer survey

Amazon Kinesis > SQL applications (legacy) > ticker-analytics > Configure SQL code

## Configure SQL code for ticker-analytics

SQL code info

View raw SQL Add SQL from templates Save application Save and run application

```
10 /*
11 .. ** Aggregate (COUNT, AVG, etc.) + Sliding time window */
12 .. Performs function on the aggregated rows over a 10 second sliding window for a specified column.
13 .. .
14 .. .
15 .. .
16 .. .
17 .. .
18 .. .
19 -- STREAM (In-application): a continuously updating entity that you can SELECT from and INSERT into like a TABLE
20 -- PUMP: an entity used to continuously 'SELECT ... FROM' a source STREAM, and INSERT SQL results into an output STREAM
21 -- Create output stream, which can be used to send to a destination
22 -- CREATE OR REPLACE STREAM DESTINATION_SQL_STREAM AS (SELECT * FROM SOURCE_SQL_STREAM WHERE symbol VARCHAR(4), ticker_symbol_count INTEGER);
23 -- Create a pump (continuously) selects From a source stream (SOURCE_SQL_STREAM)
24 -- perform an aggregate count that is grouped by columns ticker over a 10-second sliding window
25 -- CREATE OR REPLACE PUMP STREAM_PUMP AS INSERT INTO DESTINATION_SQL_STREAM
26 -- SELECT COUNT(*) OVER (PARTITION BY symbol, ticker_symbol) AS COUNT_SYMBOL_POPUP [STORED_POPULATE];
27 SELECT STREAM ticker_symbol, COUNT(*) OVER TEH_SECOND_10SECOND_WINDOW AS ticker_symbol_count
28 FROM [SOURCE_SQL_STREAM_001]
29 WHERE results partitioned by ticker_symbol and a 10-second sliding time window
30 ▶ PARTITION_BY ticker_symbol
31 PARTITION_BY ticker_symbol
32 RANGE INTERVAL '10' SECOND PRECEDING;
33
```

SQL Ln 1, Col 1 0 Errors 0 Warnings 0

Output Input

Output streams (2) info Edit destination Connect to destination

Q Search rows

In-application name	AWS service	Resource	ID
error_stream	-	-	-
DESTINATION_SQL_STREAM	Kinesis Firehose Streams	ticker_analytics	4.1

DESTINATION\_SQL\_STREAM - sample results () View sample Pause Resume

CloudShell Feedback Language 34°C Mostly cloudy Search

CloudShell Feedback Language 34°C Mostly cloudy Search ENG US 5/18/2023 3:57 AM Privacy Terms Cookies preferences

Amazon Kinesis Mumbai • Neeraj S G 5 G

Dashboard

Data streams

Data Firehose

Analytics applications

Streaming applications

Studio notebooks

SQL applications (legacy)

Resources

What's new (2)

AWS Streaming Data Solution for Amazon Kinesis (2)

AWS Glue Schema Registry (2)

Customer survey

CloudShell Feedback Language

Search [Alt+G]

CREATE OR REPLACE STREAM DESTINATION\_SQL\_STREAM AS SELECT \* FROM error\_stream [SOURCE\_SQL\_STREAM=001]  
-- performs an aggregate count that is grouped by columns over a 10-second sliding window  
CREATE OR REPLACE PUMP STREAM\_PUMP AS INSERT INTO DESTINATION\_SQL\_STREAM  
-- COUNT([AVG|MAX|SUM|STDEV|POP|STDDEV\_SAMP|VAR\_POP|VAR\_SAMP])  
SELECT STREAM ticker\_symbol, COUNT(\*) OVER TEN\_SECOND\_SLIDING\_WINDOW AS ticker\_symbol\_count  
--  
-- Results partitioned by ticker\_symbol and a 10-second sliding time window  
WINDOW TEN\_SECOND\_SLIDING\_WINDOW AS (  
PARTITION BY ticker\_symbol  
RANGE INTERVAL '10' SECOND PRECEDING);

SQL Ln 35 Col 1 Errors: 0 Warnings: 0

Output Input

Output streams (2) (1)

DESTINATION\_SQL\_STREAM (1) Kinesis Firehose Streams (1) ticker\_analytics (1) ID 4.1

DESTINATION\_SQL\_STREAM - sample results (7)

View sample Pause Resume

ROWID	TICKER_SYMBOL	TICKER_SYMBOL_COUNT
2023-05-...	ALV	1
2023-05-...	BFI	1
2023-05-...	NFS	1
2023-05-...	TGH	1
2023-05-...	WMT	1
2023-05-...	PIN	1
2023-05-...	QWE	1

Back

### **B) View results in S3 buckets**

Navigate to S3 buckets to view the results being saved there:

Amazon S3 ->Buckets -> ticker-123 ->ticker\_analytics/

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

CloudShell Feedback Language

Search

Objects (2)

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

Actions

Upload

Name Type Last modified Size Storage class

- ticker\_analytics/ Folder -
- ticker/ Folder -

CloudShell Feedback Language

34°C Mostly cloudy

CloudShell Feedback Language

ENG US 3:58 AM 5/18/2023

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

CloudShell Feedback Language

Search

Properties

Permissions

Versions

Object overview

Owner

23486598035/7970053c05f9e7731b84764b4695fe1f979510365a04d848dbb

AWS Region

Asia Pacific (Mumbai) ap-south-1

Last modified

May 18, 2023, 03:58:08 (UTC-05:00)

Size

612.0 B

Type

Key

Bucket properties

Bucket Versioning

We recommend that you enable Bucket Versioning to help protect against unintentionally overwriting or deleting objects. Learn more

Enable Bucket Versioning

Object Lock

When enabled, this object will be prevented from being deleted or overwritten until the hold is explicitly removed.

Disabled

Object Lock retention mode

In compliance mode, this object cannot be deleted or overwritten until the hold is explicitly removed. In compliance mode, the object can't be overwritten or deleted.

S3 URI

arn:aws:s3:::ticker-123/ticker\_analytics/2023/05/18/08/ticker\_analytics-1-2023-05-18-08-57-06-b8ce4109-355a-46a7-aab2-ee415f2cb074

Amazon Resource Name (ARN)

Entity tag (Etag)

Object URL

CloudShell Feedback Language

ENG US 3:59 AM 5/18/2023

## Download the results to local system:

Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight

AWS Marketplace for S3

CloudShell Feedback Language

Downloads

This PC (C:) > Users > navee > Downloads

File Edit View

Ln 1, Col 1

Bucket "ticker-123" doesn't have Bucket Versioning enabled

We recommend that you enable Bucket Versioning to help protect against unintentionally overwriting or deleting objects. Learn more

Enable Bucket Versioning

Object Lock

When enabled, this object will be prevented from being deleted or overwritten until the hold is explicitly removed.

Disabled

Object Lock retention mode

In compliance mode, this object cannot be deleted or overwritten until the hold is explicitly removed. In compliance mode, the object can't be overwritten or deleted.

Expiration rule

You can use a lifecycle configuration to define expiration rules to schedule the removal of this object after a pre-defined time period.

Expiration date

The object will be permanently deleted on this date.

CloudShell Feedback Language

ENG US 3:59 AM 5/18/2023

## 6) Configure, create and run AWS Glue crawler

The screenshot shows the AWS Glue Welcome page. On the left, a sidebar lists services like ETL jobs, Data Catalog, and Data integration and ETL. The main area has sections for 'Prepare your account for AWS Glue', 'Catalog and search for datasets', 'Move and transform data', 'Data integration and management', and 'What's new in Glue'. A central box displays recent news items such as 'AWS Glue Crawler now support custom JDBC drivers' and 'AWS Glue large instance types are now generally available'. The bottom of the screen shows a taskbar with browser tabs and system icons.

Navigate to AWS Glue -> Data Catalog -> Crawlers -> Create Crawler

### Crawlers

Crawlers connects to your data stores, examines the data, and creates metadata tables in the AWS Glue Data Catalog. This metadata is then used by AWS Glue for ETL (Extract, Transform, Load) jobs and other data processing tasks.

The screenshot shows the AWS Glue Crawlers page. The sidebar is identical to the previous one. The main content area shows a table titled 'Crawlers (0) info' with a single row indicating 'No resources'. A 'Create crawler' button is located at the top right of the table. The bottom of the screen shows a taskbar with browser tabs and system icons.

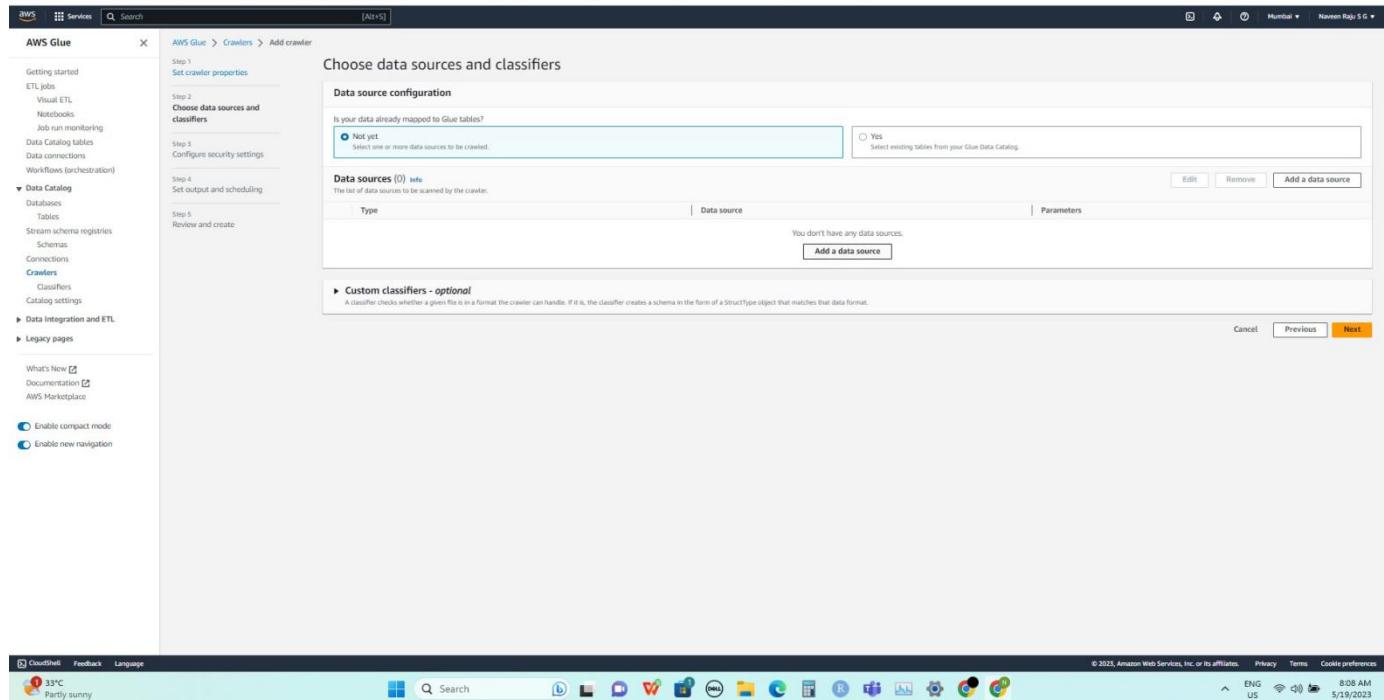
## I) Set Crawler properties

Name : DemoCrawler

## II) Choose data sources and classifiers

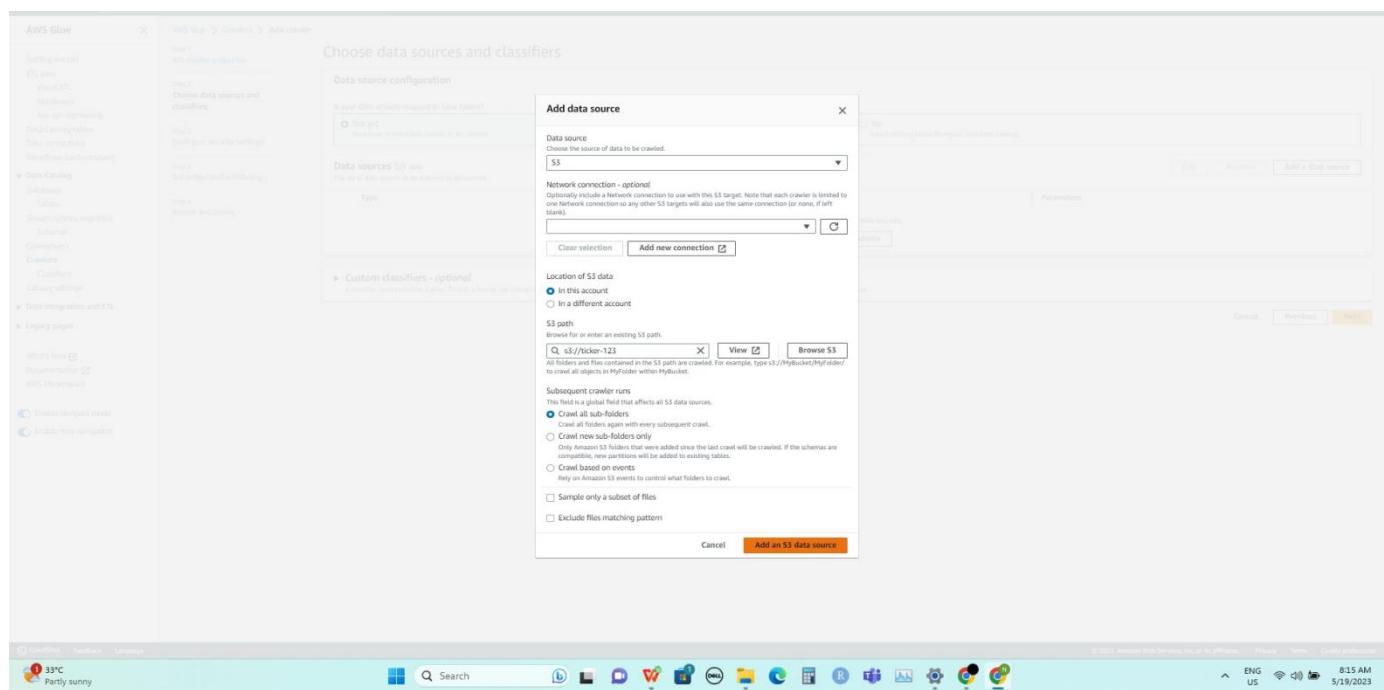
Is your data already mapped to Glue tables : No

Click on Add a Data Source



Choose S3 bucket ticker-123 which was created by AWS Kinesis FireHose and AWS Kinesis Analytics application

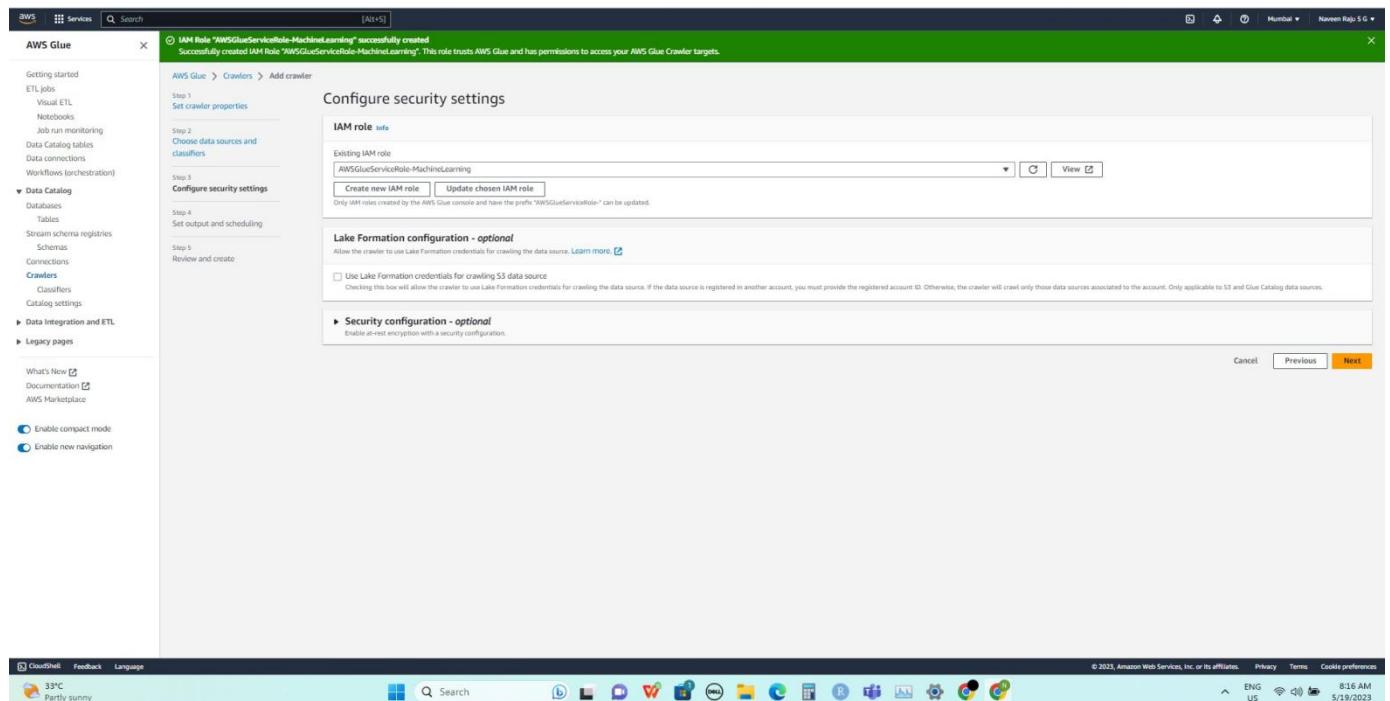
Subsequent crawler runs : Crawl all sub-folders



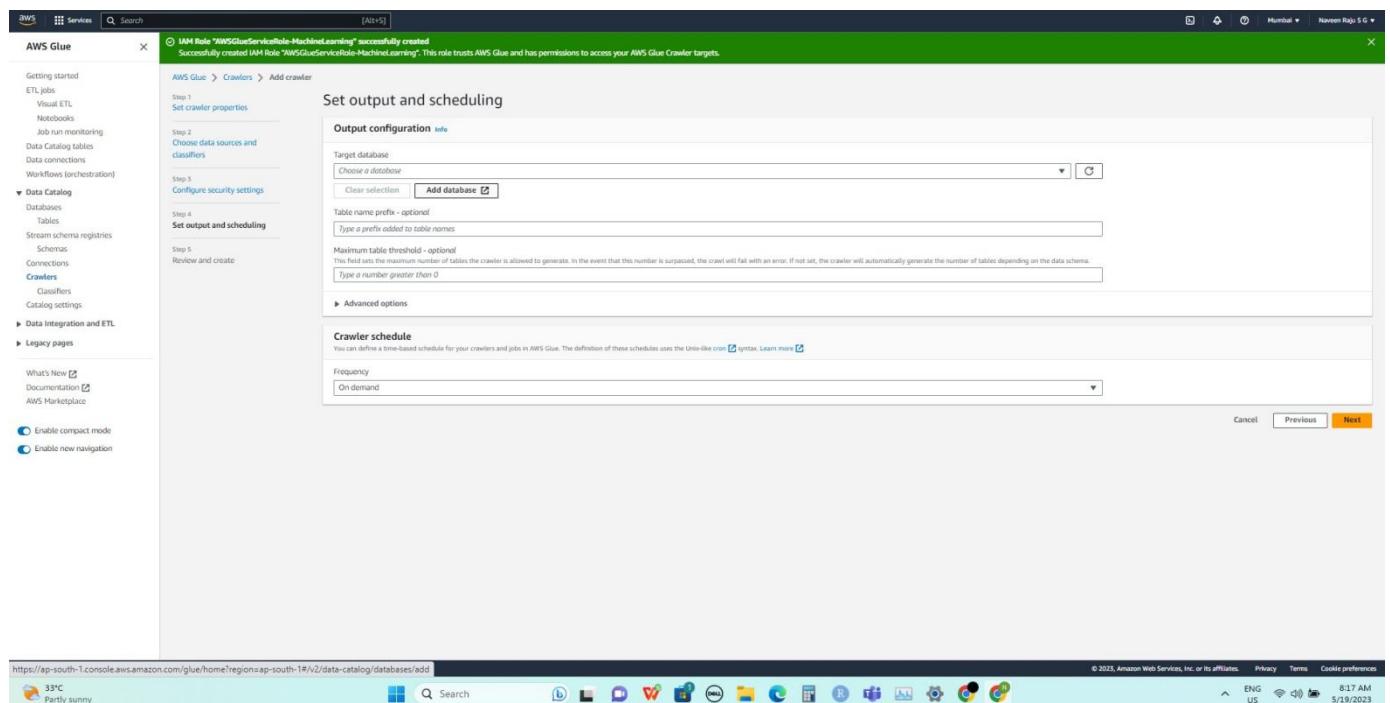
### III) Configure security settings

Create a new IAM role : AWSGlueServiceRole-MachineLearning which has following permissions:

“s3:GetBucketLocation”, “s3>ListBucket”, “s3>ListAllMyBuckets”, “s3:GetBucketAcl”,  
“ec2:DescribeVpcEndpoints”,  
“ec2:DescribeRouteTables”, “ec2>CreateNetworkInterface”,  
“ec2>DeleteNetworkInterface”, “ec2:DescribeNetworkInterfaces”,  
“ec2:DescribeSecurityGroups”, “ec2:DescribeSubnets”, “ec2:DescribeVpcAttribute”,  
“iam>ListRolePolicies”, “iam:GetRole”,  
“iam:GetRolePolicy”, “cloudwatch:PutMetricData”, “s3>CreateBucket”, “s3GetObject”, “s3:PutObject”,  
“s3>DeleteObject”, “logs>CreateLogGroup”, “logs>CreateLogStream”, “logs:PutLogEvents”,  
“ec2>CreateTags”, “ec2:DeleteTags”.



### IV) Set output and Scheduling



## Click on Add Database

The screenshot shows the 'Create a database' page in the AWS Glue Data Catalog. The 'Database details' section contains fields for 'Name' (with placeholder 'Enter a unique database name'), 'Location - optional' (with placeholder 'Set the URL location for use by clients of the Data Catalog.'), and 'Description - optional' (with placeholder 'Enter text'). Below these fields is a note: 'Descriptions can be up to 2048 characters long.' At the bottom right are 'Cancel' and 'Create database' buttons.

The screenshot shows the 'Databases' list in the AWS Glue Data Catalog. It displays one database entry: 'machine\_learning'. The table has columns for 'Name' (checkbox), 'Description' (checkbox), and 'Location URI' (checkbox). A note at the bottom right says 'Created on (UTC) May 19, 2023 at 13:18:40'. At the top right are 'Edit', 'Delete', and 'Add database' buttons.

Continue configuring “Set output and Scheduling”

Target database : machine\_learning (previously created for AWS Kinesis)  
Crawler Schedule : On demand

The screenshot shows the AWS Glue console with the 'Crawlers' section selected. A success message at the top indicates that an IAM role has been created successfully. The main area displays 'Step 4: Set output and scheduling'. Under 'Output configuration', a target database named 'machine\_learning' is selected. The 'Table name prefix - optional' field contains 'Type a prefix added to table names'. Below it, 'Maximum table threshold - optional' is set to 'Type a number greater than 0'. Under 'Advanced options', there's a note about defining a cron schedule. The 'Frequency' dropdown is set to 'On demand'. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

## V)Create and run Crawler

Click on Create Crawler

The screenshot shows the 'Review and create' step of the crawler creation wizard. It summarizes the configurations: Step 1: Set crawler properties (Name: DemoCrawler); Step 2: Choose data sources and classifiers (Data source: s3://ticker-123); Step 3: Configure security settings (IAM role: AWSGlueServiceRole-MachineLearning); Step 4: Set output and scheduling (Database: machine\_learning). The 'Create crawler' button is highlighted at the bottom right. The browser toolbar at the bottom includes CloudShell, Feedback, Language, Search, and various icons.

## Click on Run Crawler

The screenshot shows the AWS Glue Crawler properties page for a crawler named "DemoCrawler". The top banner indicates "One crawler successfully created" and "The following crawler is now created: "DemoCrawler"". The crawler properties section shows the following details:

- Name:** DemoCrawler
- IAM role:** AWSGlueServiceRole-MachineLearning
- Database:** machine\_learning
- State:** READY

The "Crawler runs" tab is selected, showing one run entry:

Start time (UTC)	End time (UTC)	Status	DPU hours	Table changes
May 19, 2023 at 13:22:04	May 19, 2023 at 13:22:46	Completed	41 s	1 table change; 4 partition changes

At the bottom right of the page, there are buttons for "Stop run", "View CloudWatch logs", and "View run details".

The screenshot shows the AWS Glue Crawler properties page for the same "DemoCrawler". The top banner indicates "Crawler successfully starting" and "The following crawler is now starting: "DemoCrawler"". The crawler properties section is identical to the previous screenshot.

The "Crawler runs" tab is selected, showing the same run entry as before:

Start time (UTC)	End time (UTC)	Status	DPU hours	Table changes
May 19, 2023 at 13:22:04	May 19, 2023 at 13:22:46	Completed	41 s	1 table change; 4 partition changes

At the bottom right of the page, there are buttons for "Stop run", "View CloudWatch logs", and "View run details".

## VI) View results in AWS Glue tables

Navigate to AWS Glue -> Tables

We can see a table has been created in the database.

The screenshot shows the AWS Glue Tables interface. On the left, there's a navigation sidebar with various options like Getting started, ETL jobs, Data Catalog, and Data Integration and ETL. The main area displays a table titled 'Tables (1)'. The table has one row for 'ticker\_123'. The columns include Name (ticker\_123), Database (machine\_learning), Location (s3://ticker-123/), Classification (json), and Deprecated (no). There are buttons for Add table, Delete, View data, and Table data. The status bar at the bottom indicates it was last updated on May 19, 2023, at 13:25:08.

Click on the table “ticker\_123” to see:

### Schema

The screenshot shows the AWS Glue Table details page for 'ticker\_123'. It includes sections for Table details (Name: ticker\_123, Location: s3://ticker-123/, etc.) and Schema (0). The Schema section lists nine columns: #, Column name, Data type, Partition key, and Comment. The columns are: 1 (change, double, no), 2 (price, double, no), 3 (ticker\_symbol, string, no), 4 (sector, string, no), 5 (partition\_0, string, Partition (0)), 6 (partition\_1, string, Partition (1)), 7 (partition\_2, string, Partition (2)), 8 (partition\_3, string, Partition (3)), and 9 (partition\_4, string, Partition (4)). There are buttons for Edit schema as JSON and Edit schema.

## Partitions

The screenshot shows the AWS Glue Table Details page for a table named 'ticker\_123'. The 'Partitions' tab is selected, displaying a table with five columns: partition\_0, partition\_1, partition\_2, partition\_3, and partition\_4. The data in the table is as follows:

partition_0	partition_1	partition_2	partition_3	partition_4
ticker	demo2023	05	19	12
tiger_analytics	2023	05	19	12
tiger_analytics	2023	05	19	13
ticker	demo2023	05	19	11

## Partitions Indexes

AWS Glue partition indexes are an important configuration to reduce overall data transfer and processing, and reduce query processing time.

The screenshot shows the AWS Glue Table Details page for a table named 'ticker\_123'. The 'Indexes' tab is selected, displaying a table with two columns: index name and index keys. The data in the table is as follows:

Index name	Index keys
crawler_partition_index	partition_0, partition_1, partition_2, partition_3, partition_4

## On the same window Navigate to Actions->Edit Schema

The screenshot shows the AWS Glue console interface. On the left, there's a navigation sidebar with various options like 'Getting started', 'ETL jobs', 'Data Catalog tables', and 'Data integration and ETL'. The main area displays the 'ticker\_123' table details. At the top right, there are actions like 'Explore', 'View properties', 'Compare versions', 'View data', 'Manage', 'Edit table', 'Edit schema' (which is highlighted in blue), and 'Delete table'. Below the table details, there's a schema editor table with columns: #, Column name, Data type, Partition key, and Comment. The table contains 9 rows of schema information.

#	Column name	Data type	Partition key	Comment
1	change	double	-	-
2	price	double	-	-
3	ticker_symbol	string	-	-
4	sector	string	-	-
5	partition_0	string	Partition (0)	-
6	partition_1	string	Partition (1)	-
7	partition_2	string	Partition (2)	-
8	partition_3	string	Partition (3)	-
9	partition_4	string	Partition (4)	-

This screenshot shows the 'Edit schema' dialog for the 'ticker\_123' table. It has a header 'Edit schema: ticker\_123' and a sub-header 'Schema (9)'. Below is a table with columns: #, Column name, Data type, Partition key, and Comment. The table rows correspond to the schema defined in the previous screenshot. At the bottom of the dialog, there are 'Cancel' and 'Save as new table version' buttons.

#	Column name	Data type	Partition key	Comment
1	change	double	-	-
2	price	double	-	-
3	ticker_symbol	string	-	-
4	sector	string	-	-
5	partition_0	string	Partition (0)	-
6	partition_1	string	Partition (1)	-
7	partition_2	string	Partition (2)	-
8	partition_3	string	Partition (3)	-
9	partition_4	string	Partition (4)	-

Edit names of all partitions as year, month, day, hour because this was the format the files were created in S3 buckets.

The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a sidebar with various navigation options like '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', 'Data Integration and ETL', and 'Legacy pages'. The main area is titled 'Edit schema: ticker\_123' and shows a table with 9 columns. The columns are: #, Column name, Data type, Partition key, and Comment. The 'partition\_4' column is marked as a partition key. At the bottom right of the schema editor, there are 'Cancel' and 'Save as new table version' buttons. The status bar at the bottom indicates it's 8:31 AM on 5/19/2023.

## 7) Configure, create and run AWS Glue studio ETL job

Navigate to “Data Integration and ETL” -> AWS Glue Studio -> ETL Jobs

Now lets convert the JSON format table created in last step to Parquet format table.

Parquet File - is a columnar storage file format widely used in the big data ecosystem.

Three Main Key feature of Parquet File:

Columnar Storage - Parquet stores data in a columnar format rather than the traditional row-based format. This means that the values of each column are stored together, which offers several advantages for analytical processing.

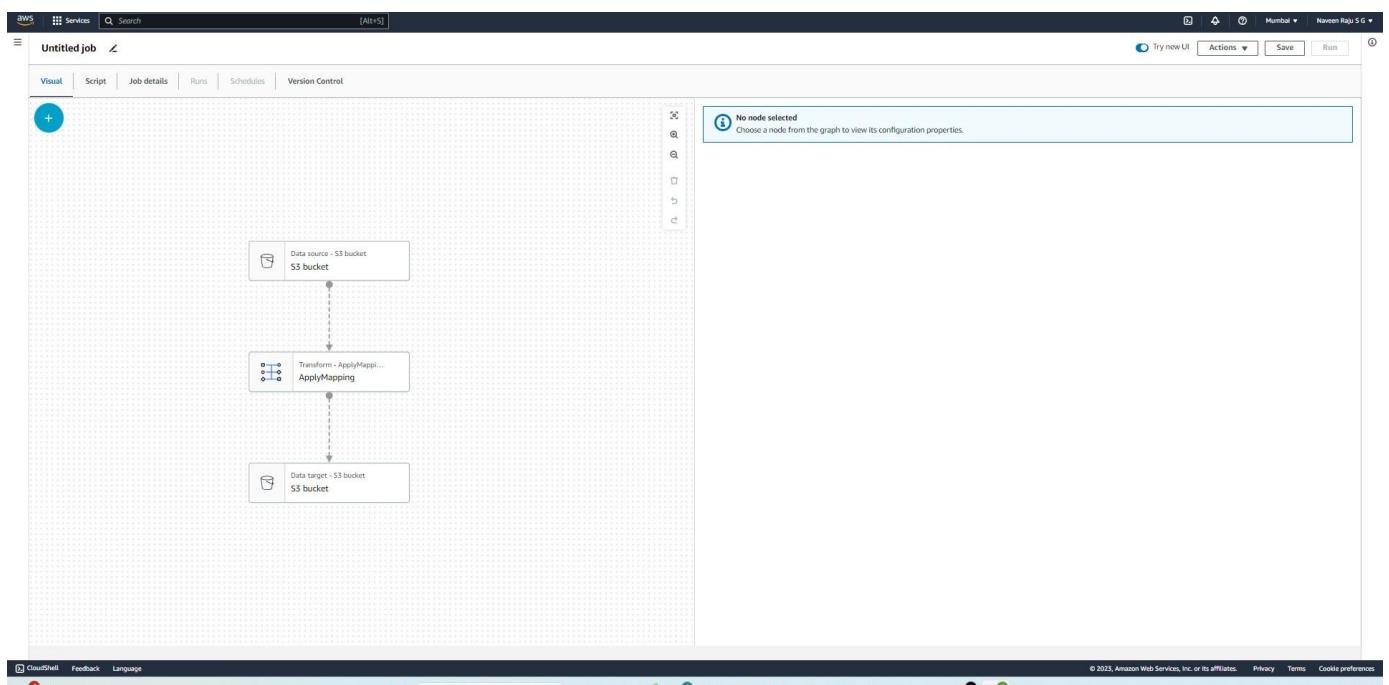
Performance - Due to its columnar nature, Parquet is highly optimized for analytical queries and can significantly speed up query performance.

Schema Evolution: Parquet supports schema evolution, meaning you can add new columns to your data over time without breaking compatibility with existing data. This makes it easier to evolve and manage data schemas in a flexible manner.

Create Job config: Visual with Source and Target

Click on create

Configure 3 parts of the flow chart : Data Source - S3 bucket, Transform - ApplyMapping, Data target - S3 bucket and also configure Job details



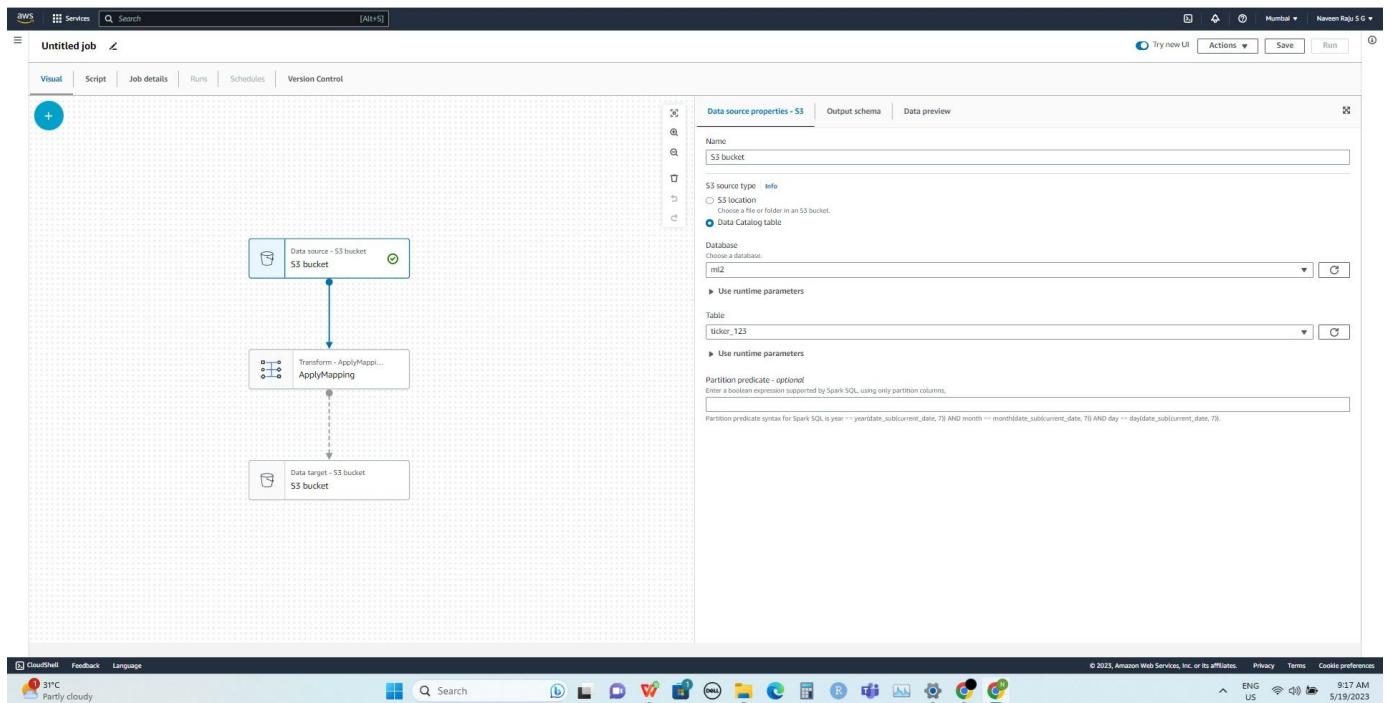
## I) Configure Data Source-S3 bucket

Name : S3 bucket

S3 Source type : Data Catalog table

Database : machine\_learning

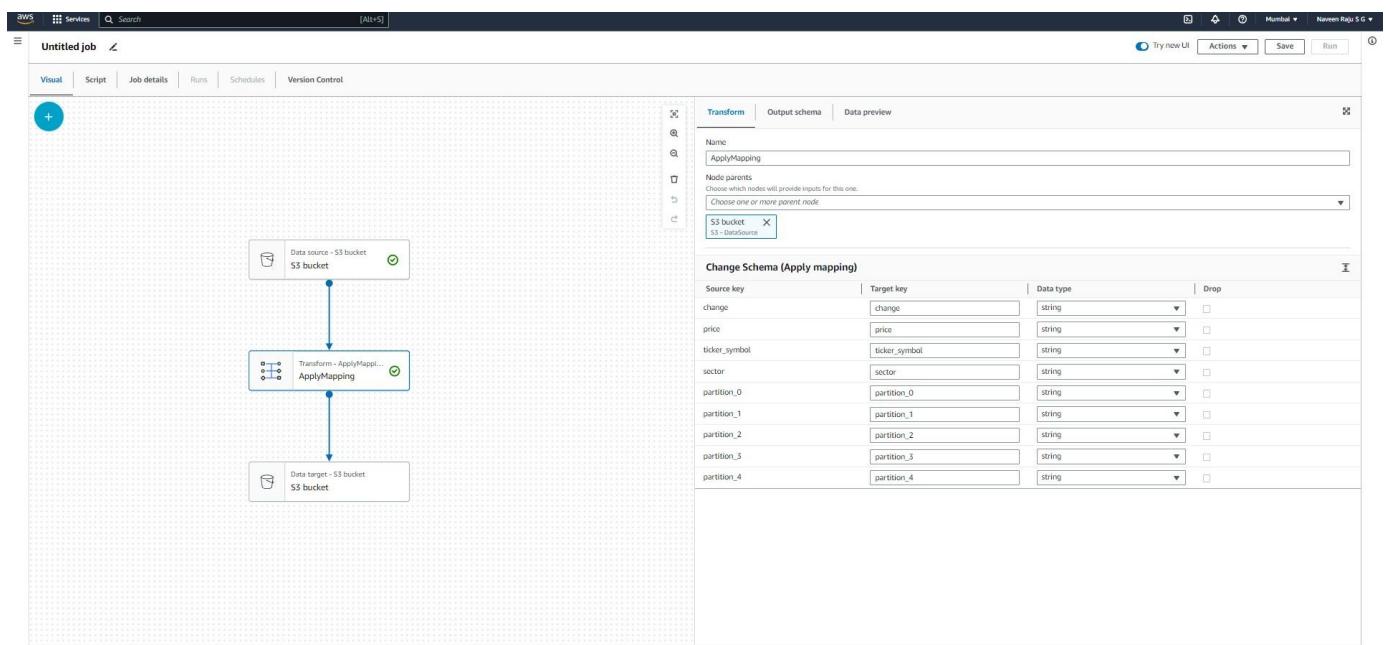
Table : ticker\_123



## II) Configure Transform-Apply Mapping

Name : Apply Mapping

Node parents (which node will provide inputs for this) : S3 bucket



## III) Configure Data target - s3 bucket

Name : S3 bucket

Node parents (Node which will provide input to this one) : ApplyMapping

Format : Parquet

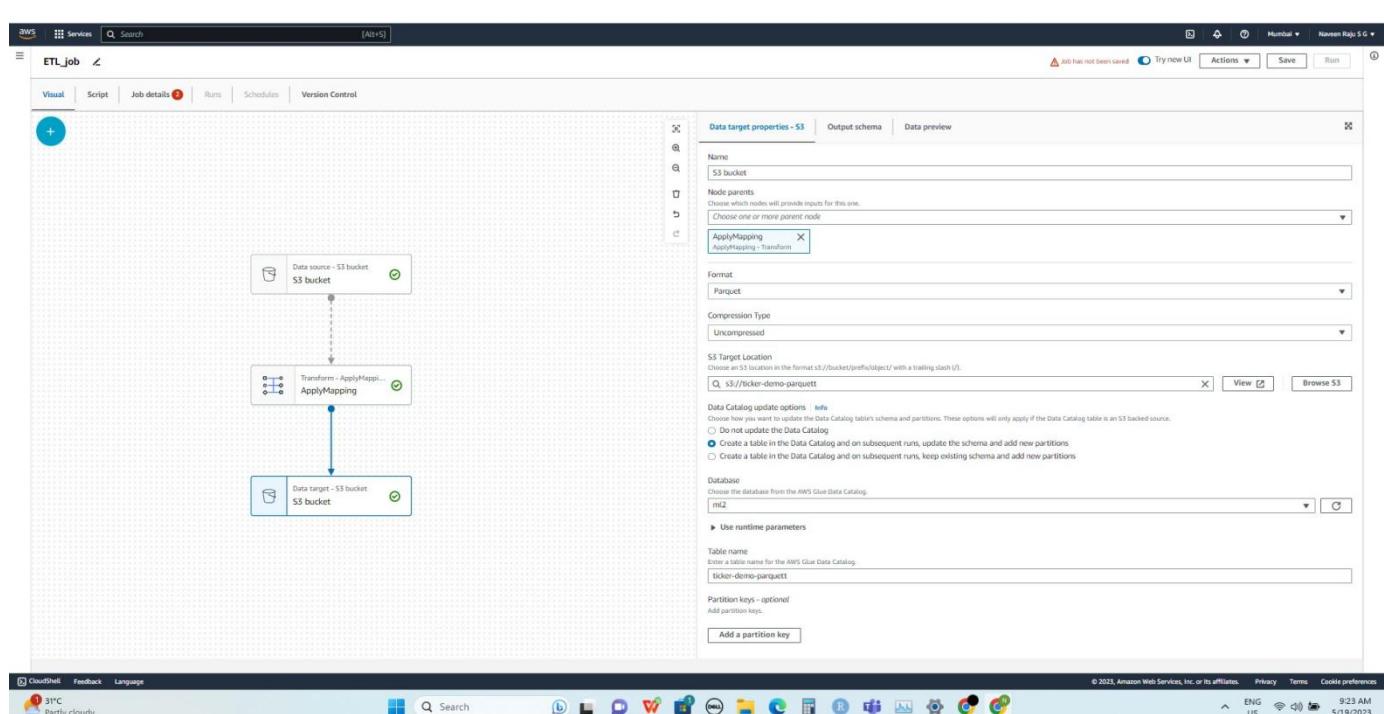
Compression Type : Uncompressed

S3 Target location : s3://ticker-demo-parquett (Create this s3 bucket before giving this location)

Data Catalog update options : Create a table in the Data Catalog and on subsequent runs, update the schema and add new partitions

Database : machine\_learning

Table name : ticker-demo-parquet



## IV) Configure Job details

Create IAM role:

Navigate to Identity and Access Management(IAM) page and create a new role

### Step 1 (Select trusted entity):

Trusted entity type : AWS service

Use case : Glue (Allows Glue service to call AWS services on our behalf)

### Step 2 (Add permissions):

AmazonS3FullAccess

AWSGlueServiceRole

Name of the role : DemoGlueETLRole

Click on create role.

### Job Details config:

Name : Demo job glue ETL

IAM Role : DemoGlueETLRole

Type (The type of ETL job. This is set automatically based on the types of data sources you have selected) : Spark

Glue version : Glue 3.0 - Supports spark 3.1, Scala 2, Python 3

Language : Scala

Worker type(Predefined worker that is allowed when a job runs) : G1X (4VCPU and 16GB RAM)

Requested number of workers (The number of workers we want AWS Glue to allocate to this job) : 2

Job timeout (Max execution time is default 2880 minutes for a Glue ETL job) : 2880

**Demo job glue ETL**

Job has not been saved Try now!

**Actions** Save Run

**Basic properties**

**Name**: Demo job glue ETL

**Description - optional**: Descriptions can be up to 2048 characters long.

**IAM Role**: Role assumed by the job with permission to access your data stores. Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job.

**Type**: The type of ETL job. This is set automatically based on the types of data sources you have selected. Spark

**Glue version**: 3.0

**Language**: Scala

**Worker type**: Set the type of predefined worker that is allowed when a job runs. G 1X (1vCPU and 15GB RAM)

**Automatically scale the number of workers**: AWS Glue will estimate costs and resource usage by dynamically scaling the number of workers up and down throughout the job run. Requires Glue 3.0 or later.

**Requested number of workers**: The number of workers you want AWS Glue to allocate to this job. 10

**Generate job insights**: AWS Glue will analyze your job runs and provide insights on how to optimize your jobs and the reasons for job failures.

**Job bookmark**:  Specifies how AWS Glue processes job bookmark when the job runs. It can remember previously processed data (Enable), update state information (Pause), or ignore state information (Disable).

CloudWatch Feedback Language

32°C

Search

Mumbai Naveen Raju SG V

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookies preferences

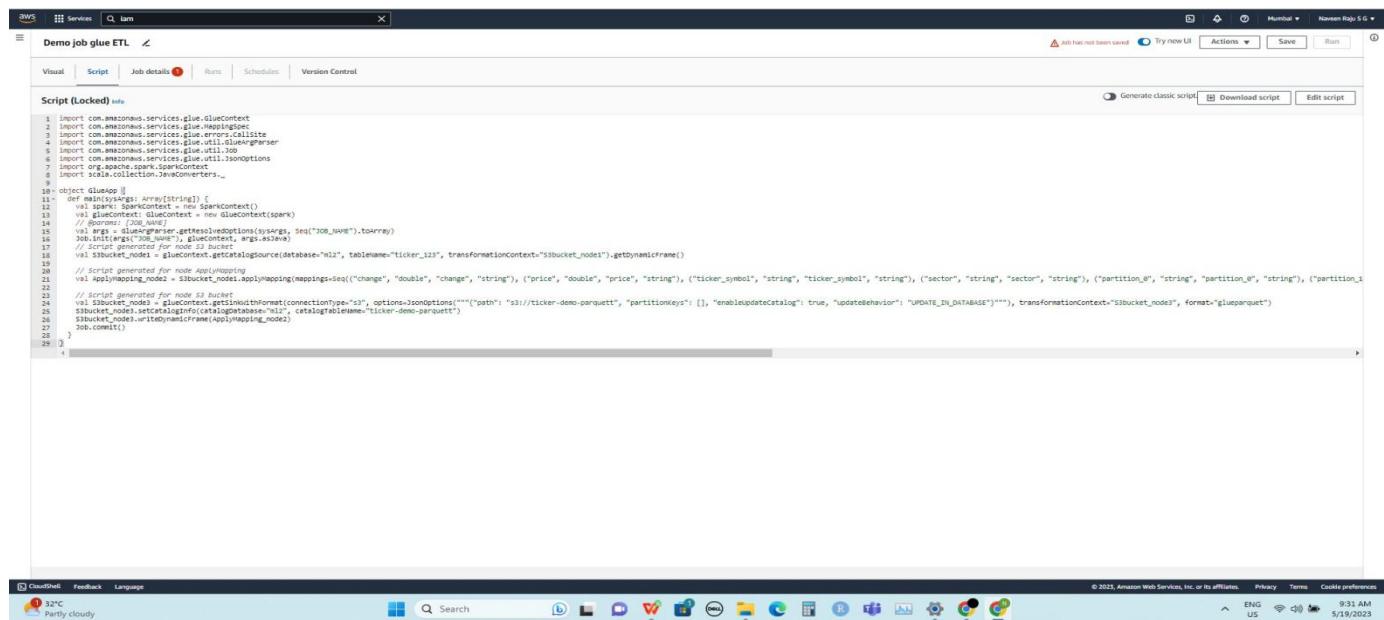
ENGLISH ENG 9:29 AM

The screenshot shows the AWS Glue job configuration interface. The top navigation bar includes links for CloudShell, Feedback, Language, and a search bar. On the right, there are status icons for battery (32°C), signal strength, and time (9:29 AM). The main header displays 'Demo job glue ETL' under the 'aws' service. Below the header, tabs for Visual, Script, Job details (with a red notification dot), Items, Schedules, and Version Control are visible. The 'Job details' tab is active, showing the following configuration:

- Targets**: temporary directory, scripts, and any libraries used by the job. Set to 'DemoGlueETLRole'.
- Type**: The type of ETL job. This is set automatically based on the types of data sources you have selected. Set to 'Spark'.
- Glue version**: Info. Set to 'Glue 3.0 - Supports spark 3.1, Scala 2, Python 3'.
- Language**: Set to 'Scala'.
- Worker type**: Set to 'G 1x' (1xCPU and 16GB RAM).
- Automatically scale the number of workers**: A checkbox that allows AWS Glue to monitor costs and resource usage by dynamically scaling the number of workers up and down throughout the job run. Requires Glue 3.0 or later.
- Requested number of workers**: The number of workers you want AWS Glue to allocate to this job. Set to '2'.
- Generate job insights**: A checkbox that allows AWS Glue to analyze your job runs and provide insights on how to optimize your jobs and the reasons for job failures.
- Job bookmark info**: A section describing how AWS Glue processes job bookmark when the job runs. It can remember previously processed data (Enable), update state information (Present), or ignore state information (Disable). Set to 'Disable'.
- Flex execution info**: A section about running the job on spare capacity, ideal for non-critical workloads that don't require fast job start times or consistent execution times. See recommendations, limitations and pricing in the help panel by clicking on the info link above.
- Number of retries**: Set to '0'.
- Job timeout (minutes)**: Set the execution time. The default is 2,880 minutes (48 hours) for a Glue ETL job. No job timeout is defaulted for a Glue Streaming job. Set to '2880'.

At the bottom, there are 'Actions' (Save, Run), a note about the job not being saved, and links for Try new UI, Save, and Run.

## View the script generated:



```
1 import com.amazonaws.services.glue.GlueContext
2 import com.amazonaws.services.glue.MappingParser
3 import com.amazonaws.services.glue.util.JsonOptions
4 import com.amazonaws.services.glue.util.JobUtil
5 import com.amazonaws.services.glue.util.JsonParser
6 import com.amazonaws.services.glue.util.JsonOptions
7 import org.apache.spark.SparkContext
8 import org.apache.spark.sql.SparkSession
9
10 object GlueJob {
11   def main(args: Array[String]): Unit = {
12     val spark = SparkSession.builder().getOrCreate()
13     val glueContext = new GlueContext(spark)
14     // overrides
15     val sysargs = JobUtil.getARGVOverrides(sysargs, Seq("JOB_NAME").toNSArray)
16     JobUtil.setArgs("JOB_NAME", glueContext, args.asJava)
17     // setup
18     val s3bucket_node1 = glueContext.getcatalogsource(database="ml2", tableName="ticker_123", transformationContext="s3bucket_node1").getByDynamicFrame()
19     // Script generated for node ApplyMapping
20     val applyMapping_node2 = s3bucket_node1.applyMapping(mappings=Seq(("change", "double", "change", "string"), ("price", "double", "price", "string"), ("ticker_symbol", "string", "ticker_symbol", "string"), ("sector", "string", "sector", "string"), ("partition_0", "string", "partition_0", "string"), ("partition_1", "string", "partition_1", "string")))
21     // Script generated for node S3 Bucket
22     val s3node3 = applyMapping_node2.write.format(connectionType="s3", options=JsonOptions("""{"path": "s3://ticker-demo-parquet", "partitionKeys": [], "enableUpdateCatalog": true, "updateBehavior": "UPDATE_IN_DATABASE"}"""), transformationContext="s3bucket_node3", format="glueparquet")
23     s3bucket_node3.setCatalogInfo(catalogDatabase="ml2", catalogTableName="ticker_demo-parquet")
24     s3bucket_node3.updateDynamicFrame(applyMapping_node2)
25     s3bucket_node3.commit()
26   }
27 }
```

## Continue editing JobConfig ->Advance Properties:

Script filename : scalaETL

Script Path (S3 path) : keep the default path which would have auto generated

Select the check box of the following:

Job metrics (Enable the creation of Job Metrics when the job runs)

Continuous logging (Enable logs in Cloud watch)

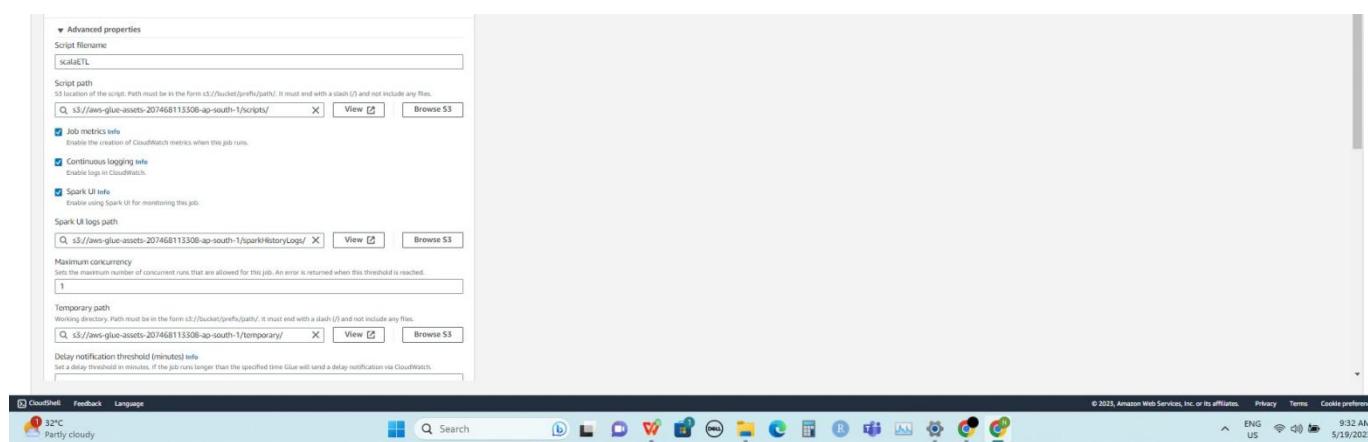
Spark UI (Enable using Spark UI for monitoring this job)

Spark UI logs path : keep the default path which would have auto generated

Maximum concurrency (Maximum concurrent runs for this jobs that are allowed) : 1

Temporary path(working directory) : keep the default path which would have auto generated

Security Configuration (encryption) : None



Advanced properties

Script filename: scalaETL

Script path: s3://aws-glue-assets-207468113508-ap-south-1/scripts/

Job metrics info:  Enable the creation of CloudWatch metrics when this job runs.

Continuous logging info:  Enable logs in CloudWatch.

Spark UI info:  Enable using Spark UI for monitoring this job.

Spark UI logs path: s3://aws-glue-assets-207468113508-ap-south-1/sparkHistoryLogs/

Maximum concurrency: 1

Temporary path: s3://aws-glue-assets-207468113508-ap-south-1/temporary/

Delay notification threshold (minutes) info: Set a delay (threshold in minutes). If the job runs longer than the specified time Glue will send a delay notifications via CloudWatch.



## Save the config and Run the job:

The screenshot shows the AWS Glue Job configuration page for a job named "Demo job glue ETL". Under "Basic properties", the "Name" is set to "Demo job glue ETL". The "IAM Role" dropdown is set to "DemoJobRoleETLRole". The "Type" is set to "Spark". The "Glue version" is "Glue 3.0 - Supports spark 3.1, Scala 2, Python 3". The "Language" is "Scala". The "Worker type" is "G 1X (4vCPU and 15GiB RAM)". The "Requested number of workers" is set to 2. The "Generate job insights" checkbox is checked.

The screenshot shows the AWS Glue Job runs page. It displays one job run under "Job runs (1/1) Info". The run status is "Running" and it started at "05/19/2023 09:32:26". The "Table View" tab is selected. The log stream shows the following output:

```

CloudWatch continuous logs
Driver logs
23/05/19 14:33:16 INFO DAGScheduler: Job 0 finished: runJob at GlueParquetHadoopWriter.scala:176, Took 12.360520 s
23/05/19 14:33:16 INFO DAGScheduler: Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
23/05/19 14:33:16 INFO DAGScheduler: ResultStage 0 (runJob at GlueParquetHadoopWriter.scala:176) finished in 12.155 s
23/05/19 14:33:04 INFO DAGScheduler: Final stage: ResultStage 0 (runJob at GlueParquetHadoopWriter.scala:176)
23/05/19 14:33:04 INFO DAGScheduler: Got job 0 (runJob at GlueParquetHadoopWriter.scala:176) with 00 output partitions
23/05/19 14:33:04 INFO DAGScheduler: Final stage: ResultStage 0 (runJob at GlueParquetHadoopWriter.scala:176) with 00 output partitions
23/05/19 14:33:04 INFO DAGScheduler: In action for final frontier: Partition 0 (3) is com.amazonaws.services.glue.sinks.HadoopDataSink
23/05/19 14:33:00 INFO GlueContext: Glue secret manager integration: secretId is not provided
23/05/19 14:32:59 INFO GlueContext: The parameter in action 1 com.amazonaws.services.glue.hadoopDataSource
23/05/19 14:32:59 INFO GlueContext: Glue secret manager integration: secretId is not provided.
23/05/19 14:32:59 INFO GlueContext: location s3://ticker-123/
23/05/19 14:32:59 INFO GlueContext: No partitions from catalog are 4. consider catalogPartitionPreditc to reduce the number of partitions to scan through
23/05/19 14:32:59 INFO GlueContext: getExternalSource: transactionId: <not-specified> noTime: <not-specified> catalogPartitionIndexPreditc: <not-specified>
23/05/19 14:32:58 INFO GlueContext: classification: spot
23/05/19 14:32:58 INFO GlueContext: getCatalogSource: catalogId: null, namespace: n12, tableName: ticker_123, isRegisteredWithLF: false, isGoverned: false, isRowFilterEnabled: false, useAdvancedFiltering: false, isTableFromSchemaitity: false
23/05/19 14:32:57 INFO GlueContext: Gluemetrics configured and enabled
23/05/19 14:32:55 INFO Utils: Successfully started service 'sparkDriver' on port 38379.
  
```

Below the log, the "Input arguments (10)" section shows the command-line arguments used to run the job:

Key	Value
--class	GlueApp
--enable-metrics	true
--enable-spark-ui	true
--spark-event-log-path	s3://aws-glue-assets-207468113308-ap-south-1/sparkHistoryLogs/
--enable-job-insights	true
--enable-glue-datacatalog	true

## V) View results in S3 buckets

The screenshot shows the AWS S3 console interface. On the left, the navigation pane is open with 'Amazon S3' selected. Under 'Buckets', 'ticker-demo-parquet' is listed. The main content area shows a table of objects with 30 entries. Each entry includes a checkbox, the object name (e.g., run-1684506784011-part-block-0-r-00029-snappy.parquet), the type (parquet), the last modified date (May 19, 2023, 09:33:17 UTC-05:00), size (1.8 KB), and storage class (Standard). The table has headers for Name, Type, Last modified, Size, and Storage class. At the top of the table, there are buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. A search bar at the top says 'Find objects by prefix'. The bottom of the page shows the browser's address bar with the URL <https://s3.console.aws.amazon.com/s3/object/ticker-demo-parquet?region=ap-south-1&prefix=run-1684506784011-part-block-0-r-00029-snappy.parquet>. The status bar at the bottom right shows '32°C Partly cloudy', 'ENG US', '9:34 AM', and '5/19/2023'.

Download the files to view

The screenshot shows the AWS S3 object details page for 'run-1684506784011-part-block-0-r-00029-snappy.parquet'. The left sidebar is identical to the previous screenshot. The main content area is titled 'Properties' for this specific object. It shows the following details:

- Object overview:**
  - Owner: 2a486d9803579700b3c05f9e7731b84764b45695fe1fd979510365a04d848db
  - AWS Region: Asia Pacific (Mumbai) ap-south-1
  - Last modified: May 19, 2023, 09:33:17 (UTC-05:00)
  - Size: 1.8 KB
  - Type: parquet
  - Key: run-1684506784011-part-block-0-r-00029-snappy.parquet
- Object management overview:** The following bucket properties and object management configurations impact the behavior of this object.
- Bucket properties:** Bucket Versioning.
- Management configurations:** Replication status.

The top right of the page shows a download button and a preview of the file content. The preview shows two rows of data:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

## **8) Configure, create and run AWS Athena interactive query service**



### **I) Launch query editor and edit settings**

Let us query the data created by AWS Glue ETL jobs in previous step

The screenshot shows the AWS Athena console interface. On the left, there's a sidebar with "Analytics" and a search bar. The main content area has a dark header with "Amazon Athena" and "Start querying data instantly." Below this, there's a "How it works" diagram and several sections: "Get started" (with radio buttons for "Query your data" and "Analyze your data"), "Pricing" (listing "Query editor" at \$5 per TB), "Getting started" (with a link to "Learn more"), and "More resources" (links to Documentation, API reference, FAQs, Support forums, and Release notes). At the bottom, there's a navigation bar with links for "Privacy", "Terms", and "Cookie preferences".

Before we run our first query we need to first set up a query result location in S3.

The screenshot shows the Amazon Athena Query editor interface. At the top, there are two notifications: "New Athena query engine available" and "Workgroup query engine upgrade complete". The main area is titled "Query 1" and contains a "Data" section where "AwsDataCatalog" is selected as the Data source and "machine\_learning" as the Database. Below this, under "Tables and views", there is a "Tables" section with one item listed. The SQL tab shows a single line of code: "Run Explain Cancel Clear Create". The "Query results" tab is active, displaying a search bar and a message: "No results Run a query to view results". A "Results" section is also present. On the right side, there are buttons for "Edit settings" and "Workgroup primary". The bottom of the window shows a toolbar with icons for Run, Explain, Cancel, Clear, and Create.



Create a S3 bucket named “aethena\_results”

The screenshot shows the "Manage settings" page for Amazon Athena. Under "Query result location and encryption", the "Location of query result - optional" field contains "s3://aethena-results". There is a note: "You can create and manage lifecycle rules for this bucket. Use Amazon S3 lifecycle rules to store your query results and metadata cost effectively or to delete them after a period of time." Below this, there are sections for "Expected bucket owner - optional" (with a note about specifying the AWS account ID), "Assign bucket owner full control over query results" (with a note about granting full control to another account), and "Encrypt query results". At the bottom, there are "Cancel" and "Save" buttons.



## II) Config editor

Data source : AWSDataCatalog

Database : machine\_learning

The screenshot shows the Amazon Athena Query Editor interface. On the left, there's a sidebar with navigation links like 'Amazon Athena', 'Jobs', 'Workflows', 'Administration', and 'Turn on compact mode'. The main area has tabs for 'Editor', 'Recent queries', 'Saved queries', and 'Settings'. Under 'Editor', the 'Data' tab is selected, showing 'Data source: AwsDataCatalog' and 'Database: ml2'. Below this, there's a 'Tables and views' section with a 'Create' button and a search bar. A 'Tables (2)' section lists 'ticker-demo-parquett' and 'ticker\_123'. A 'Views (0)' section is shown. On the right, there's a large text input field for writing SQL queries, with a placeholder 'Run query to view results'. Below the input field, there are buttons for 'Run Query', 'Preview Table', 'Generate table DDL', 'Insert', 'Insert into editor', 'Manage', 'Delete table', 'View properties', and 'View in Glue'. At the bottom right, there are 'Copy' and 'Download results' buttons.

## III) Write queries

Run the query by writing it in query editor:

SELECT \* FROM "ml2"."ticker-demo-parquett" limit 10; (Show first 10 records)

This screenshot shows the same Amazon Athena Query Editor interface as the previous one, but now with a second query named 'Query 2' visible in the list. The 'Data' tab is still selected, and the 'Tables (2)' section shows 'ticker-demo-parquett' and 'ticker\_123'. The main query editor area contains the SQL command: 'SELECT \* FROM "ml2"."ticker-demo-parquett" limit 10;'. Below the editor, there are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. The 'Query results' section is expanded, showing a table titled 'Completed' with 10 rows of data. The columns are labeled '#', 'change', 'price', 'ticker\_symbol', 'sector', 'partition\_0', 'partition\_1', 'partition\_2', 'partition\_3', and 'partition\_4'. The data includes entries for companies like MAB, HUV, VVS, and WSB across different sectors and partitions. At the bottom right of the results table, there are 'Copy' and 'Download results' buttons. The status bar at the bottom indicates 'Time in queue: 101 ms', 'Run time: 585 ms', and 'Data scanned: 2.31 KB'.

SELECT \* FROM "ml2"."ticker\_123" limit 10; (Show first 10 records)

The screenshot shows the Amazon Athena Query Editor interface. On the left, the navigation sidebar includes 'Jobs', 'Workflows' (Powered by Step Functions), 'Administration' (Workgroups, Data sources), and a 'Turn on compact mode' toggle. The main area has tabs for 'Editor', 'Recent queries', 'Saved queries', and 'Settings'. The 'Editor' tab is active, displaying three tabs: 'Query 1' (selected), 'Query 2', and 'Query 3'. Query 1 contains the SQL command: 'SELECT \* FROM "ml2"."ticker\_123" limit 10;'. Below this, the 'Data' section shows 'Data source: AvenDataCatalog' and 'Database: ml2'. Under 'Tables and views', there are two tables: 'ticker-demo-parquet' and 'ticker\_123'. The 'ticker\_123' table is selected and is described as 'Partitioned'. The 'Views' section shows zero views. At the bottom of the editor, there are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', and 'Create'. The 'Query results' tab is selected, showing the status 'Completed'. The results table has 10 rows and includes columns: #, change, price, ticker\_symbol, sector, partition\_0, partition\_1, partition\_2, partition\_3, and partition\_4. The results are as follows:

#	change	price	ticker_symbol	sector	partition_0	partition_1	partition_2	partition_3	partition_4
1	1.62	25.21	MAB	ENERGY	tiger_analytics	2023	05	19	12
2	0.08	5.23	KIN	ENERGY	ticker	demo2023	05	19	12
3	1.63	177.79	BNM	TECHNOLOGY	ticker	demo2023	05	19	12
4	-0.09	4.8	HJK	TECHNOLOGY	tiger_analytics	2023	05	19	13
5	-13.34	191.66	HJV	ENERGY	ticker	demo2023	05	19	11
6	-1.9	99.74	AAPL	TECHNOLOGY	ticker	demo2023	05	19	11
7	-19.22	189.18	HJV	ENERGY	tiger_analytics	2023	05	19	13
8	0.79	33.63	CJB	TECHNOLOGY	ticker	demo2023	05	19	12

At the bottom of the results table, there are buttons for 'Copy' and 'Download results'. The status bar at the bottom right indicates: Time in queue: 106 ms, Run time: 811 ms, Data scanned: 11.70 KB, © 2023, Amazon Web Services, Inc. or its affiliates, Privacy, Terms, Cookies preferences, ENG US, 9:41 AM, 5/19/2023.

## **9) Deleting resources:**

### **A) Stop the application created**

The screenshot shows the AWS Kinesis Data Analytics console. In the center, the 'ticker-analytics' application details are displayed. At the top right of this section, there is a 'Actions' dropdown menu with options: 'Run', 'View CloudWatch dashboard', 'Stop', and 'Delete'. A modal window titled 'Stop application ticker-analytics?' is open in the foreground, containing the message: 'This will stop processing data from the source.' with 'Cancel' and 'Stop' buttons.

### **B) Delete delivery stream (stream that delivers results to S3 buckets)**

#### **Delete the delivery stream tiger\_analytics**

The screenshot shows the AWS Kinesis Data Firehose console. In the center, the 'tiger\_analytics' delivery stream details are displayed. At the top right of this section, there is a 'Delete delivery stream' button. A modal window titled 'Delete delivery stream?' is open in the foreground, containing the message: 'Are you sure you want to delete this delivery stream?' with 'Cancel' and 'Delete' buttons.

The screenshot shows the AWS Kinesis Data Firehose console. A success message at the top states "ticker\_analytics was successfully created." Below it, a note says "Deleting delivery stream ticker\_analytics It can take up to a minute before the status is updated." A blue banner at the top right says "Introducing the new Kinesis Data Firehose console experience" with a note about easier access to information. The main area shows a table of "Delivery streams (2)". The first row, "ml-raju", is active and points to "ticker-123". The second row, "ticker\_analytics", is deleting and points to "ticker-123".

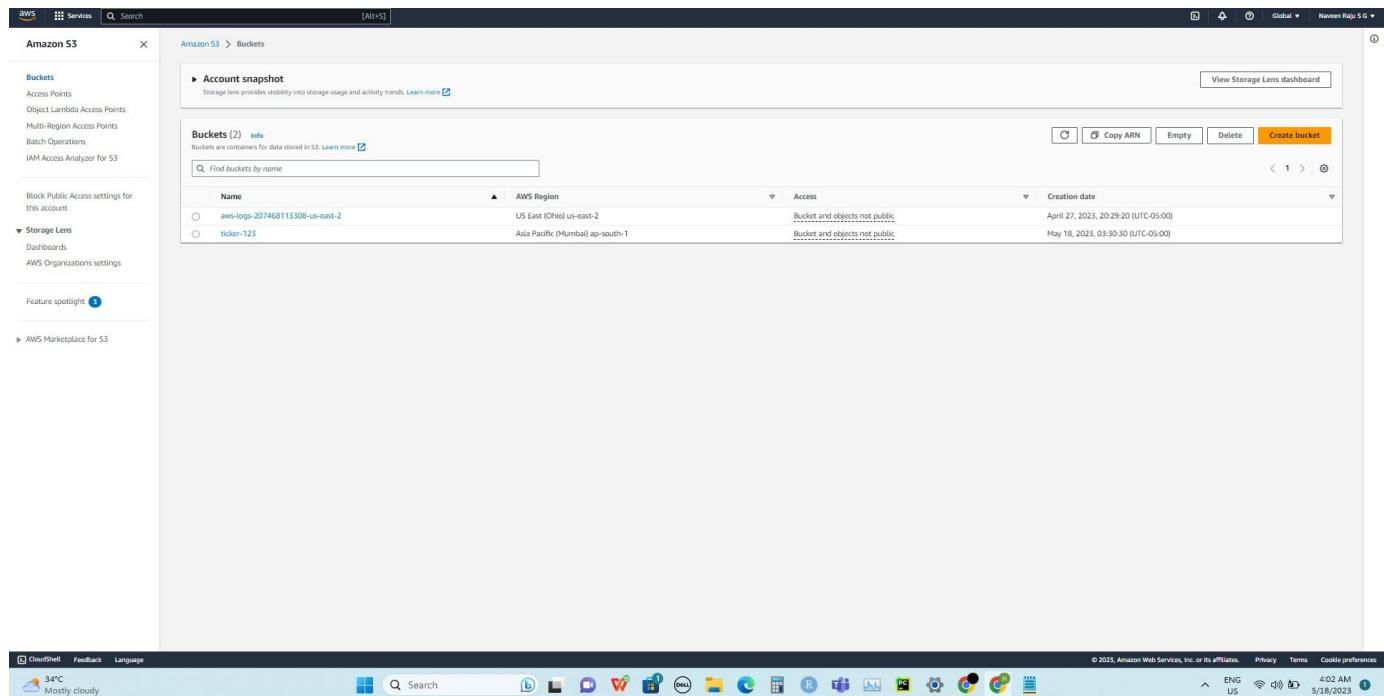
## C) Delete delivery stream (ingest data from producers that are configured to send data to Kinesis Data FireHose)

Delete the delivery stream ml\_raju:

The screenshot shows the AWS Kinesis Data Firehose console. A note at the top says "Deleting delivery stream ml\_raju It can take up to a minute before the status is updated." The main area shows a table of "Delivery streams (1)". The single row, "ml\_raju", is in a "Deleting" state and points to "ticker-123".

## **D) Delete S3 buckets**

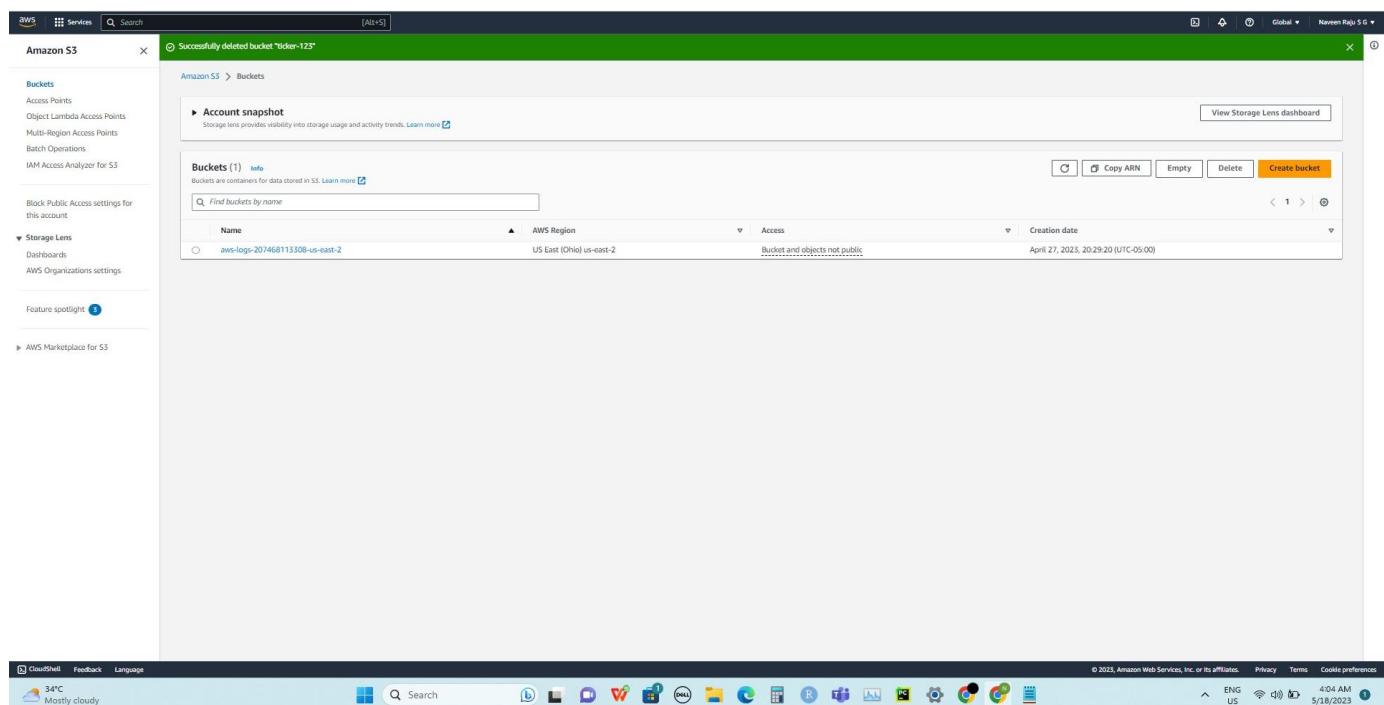
**Delete all S3 bucket ticker-123**



The screenshot shows the AWS S3 Buckets page. On the left, there's a sidebar with links like Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens, Dashboards, AWS Organizations settings, Feature spotlight, and AWS Marketplace for S3. The main area has a heading "Amazon S3 > Buckets". Below it is an "Account snapshot" section with a link to "View Storage Lens dashboard". The "Buckets (2)" section shows two entries:

Name	AWS Region	Access	Creation date
aws-logs-207468113308-us-east-2	US East (Ohio) us-east-2	Bucket and objects not public	April 27, 2023, 20:29:20 (UTC-05:00)
ticker-123	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	May 18, 2023, 03:30:30 (UTC-05:00)

At the bottom right of the main area, there are buttons for "Copy ARN", "Empty", "Delete", and "Create bucket". There are also navigation arrows and a refresh icon.



The screenshot shows the AWS S3 Buckets page after a deletion. A green banner at the top says "Successfully deleted bucket 'ticker-123'". The main area is identical to the previous screenshot, showing the two remaining buckets: aws-logs-207468113308-us-east-2 and ticker-123. The "Delete" button is now grayed out for the ticker-123 bucket.

## **E) Delete SQL application (Analytics applications)**

**Delete tiger-analytics application:**

The screenshot shows the AWS Kinesis service dashboard. On the left, there's a sidebar with various options like Dashboard, Data streams, Data Firehose, Analytics applications, Streaming applications, Studio notebooks, and SQL applications (legacy). Under Resources, it lists What's new, AWS Streaming Data Solution for Amazon Kinesis, and AWS Glue Schema Registry. A Customer survey link is also present. The main content area is titled "Kinesis Data Analytics Blueprints for Apache Flink" and "Deleting application ticker-analytics." It shows a message: "For new applications, we recommend that you use Kinesis Data Analytics Studio instead of Kinesis Data Analytics for legacy SQL applications for running SQL queries. Kinesis Data Analytics Studio provides advanced analytical capabilities, enabling you to build sophisticated stream processing applications in minutes." Below this is a table for "SQL applications (legacy)" with one entry: "ticker-analytics" (Status: Deleting). At the bottom right of the table are buttons for Run, Stop, Delete, and Create SQL application (legacy).

This screenshot is from the same AWS session as the previous one, but it shows the result of the deletion. The main content area now displays a green success message: "Application ticker-analytics has been successfully deleted." The rest of the interface is identical to the previous screenshot, including the sidebar and the table showing the now-empty list of SQL applications.

## F) Delete Crawler

The screenshot shows the AWS Glue interface with the 'Crawlers' section selected. A single crawler, 'DemoCrawler', is listed in the table. The 'Actions' menu for this crawler includes options like 'Edit crawler', 'Duplicate crawler', 'Delete crawler', 'View details', 'Resume schedule', 'Pause schedule', and 'Stop run'. The 'Delete crawler' option is highlighted with a blue border.

## G) Delete AWS Glue Studio Job

The screenshot shows the AWS Glue interface with the 'Jobs' section selected. A single job, 'Demo job glue ETL', is listed in the table. The 'Actions' menu for this job includes options like 'Edit job', 'Clone job', 'Schedule job', 'Delete job(s)', and 'Reset job bookmark'. The 'Delete job(s)' option is highlighted with a blue border.

## H) Delete AWS Glue Databases

The screenshot shows the AWS Glue Studio interface. At the top, there are four job creation options: 'Visual with a source and target' (selected), 'Visual with a blank canvas', 'Spark script editor', and 'Python Shell script editor'. Below these, a 'Source' section shows 'Amazon S3' selected, with a note about JSON, CSV, or Parquet files. A 'Target' section shows 'Amazon S3' selected, with a note about specifying a bucket path. A table titled 'Your jobs (1)' lists one job: 'Demo job glue ETL' (Type: Glue ETL, Last modified: 5/19/2023, 9:33:00 AM, AWS Glue version: 3.0). The 'Actions' menu for this job includes 'Edit job', 'Clone job', 'Schedule job', 'Delete job(s)', and 'Reset job bookmark'.



## I)Delete AWS Glue Databases

The screenshot shows the AWS Glue console under the 'Data Catalog' section. The left sidebar includes 'Getting started', 'ETL jobs', 'Visual ETL', 'Notebooks', 'Job run monitoring', 'Data Catalog tables', 'Data connections', 'Workflows (orchestration)', 'Databases' (selected), 'Tables', 'Stream schema registries', 'Schemas', 'Connections', 'Crawlers', 'Classifiers', 'Catalog settings', 'Data integration and ETL', 'AWS Glue Studio', 'Jobs', 'Interactive Sessions', 'Notebooks', 'Data classification tools', 'Sensitive data detection', 'Record Matching', 'Triggers', 'Workflows (orchestration)', 'Blueprints', and 'Security configurations'. Under 'Legacy pages', there are links for 'What's New', 'Documentation', and 'AWS Marketplace'. At the bottom, there are 'Enable compact mode' and 'Enable new navigation' checkboxes. The main area displays a table titled 'Databases (3/3)' with three entries: 'machine\_learning' (Created on (UTC): May 19, 2023 at 13:18:40), 'ml' (Created on (UTC): May 19, 2023 at 13:53:53), and 'ml2' (Created on (UTC): May 19, 2023 at 14:01:38). The 'Actions' column for each entry includes 'Edit', 'Delete', and 'Add database'.

## J) Delete all S3 buckets that were created