

AWS Data Analytics Speciality

Capstone Project (Solution)

About CloudThat

- CloudThat is the first company in India to Cloud Training & Consulting services for mid-market & enterprise clients around the world. With expertise in major Cloud platforms including Microsoft Azure, Amazon Web Services (AWS) and Google Cloud Platform (GCP). CloudThat is uniquely positioned to be the single technology source for organizations looking to utilize the flexibility and power Cloud Computing provides.
- CloudThat is focused on quickly empowering IT professionals and organizations with leveraging Cloud, Big Data & IoT. Founded by Bhavesh Goswami, an ex-Microsoft and ex-Amazonian who was part of the Microsoft and AWS product development teams.
- Till date we have trained more than 200,000 IT professionals and conducted corporate training for some of the fortune 500 companies which include Accenture, Infosys, Fidelity, HCL, Intuit, GE, TCS, HP, SAP, Oracle, Western Union, Philips, Flipkart, L&T and Samsung, just to name a few.
- We have presence in Bengaluru, USA & UK, but offer on-site and pre-scheduled public batches in different IT centric cities of India and Overseas.
- CloudThat is a Microsoft Gold Partner, Advanced AWS Consulting partner, Google Consulting Partner, Red Hat Certified Training Partner, MongoDB Ready Partner, and part of Pearson Testing Network.
- Our current course offerings are on Azure, Dynamics 365, Microsoft Security Suite, AI & Machine Learning, Cloud Security, Analytics, Red Hat, IoT, DevOps, Chef, Docker, Ansible, Kubernetes, Blockchain, Big Data, etc. We are constantly adding more courses and more consulting offerings.

Phase-1: Ingestion and Storage

1. Create an S3 bucket to load the raw datasets. Give the name of the bucket as **tlc-trip-record-data-lake-<yourname-random number>** as shown in the figure below.

Amazon S3 > Buckets > Create bucket

Create bucket [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

tlc-trip-record-data-lake-110

Bucket name must be globally unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

US East (N. Virginia) us-east-1

2. Create three folders within the bucket namely **raw-layer**, **processed-layer** and **reference-layer**.

Amazon S3 > Buckets > tlc-trip-record-data-lake-110

tlc-trip-record-data-lake-110 [Info](#)

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects and their permissions. [Learn more](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#)

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	processed-layer/	Folder	-
<input type="checkbox"/>	raw-layer/	Folder	-
<input type="checkbox"/>	reference-layer/	Folder	-

3. Load **raw data files for Q2-2020** in the **raw-layer** with the following hierarchy as **raw/2020/Q2** and **lookup file** in **reference-layer**, respectively.

Amazon S3 > Buckets > tlc-trip-record-data-lake-mi-110 > raw-layer/ > 2020/ > Q2/

Q2/ Copy S3 URI

Objects Properties

Objects (3)

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](#)

Refresh Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	yellow_tripdata_2020-04.csv	csv		20.7 MB	Standard
<input type="checkbox"/>	yellow_tripdata_2020-05.csv	csv		30.2 MB	Standard
<input type="checkbox"/>	yellow_tripdata_2020-06.csv	csv		47.9 MB	Standard

Amazon S3 > Buckets > tlc-trip-record-data-lake-110 > reference-layer/

reference-layer/ Copy S3 URI

Objects Properties

Objects (1)

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](#)

Refresh Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	taxi_zone_lookup.csv	csv		12.0 KB	Standard

Phase-2: Cataloging

- Now, let's create a crawler for raw-layer as **crawler-raw-layer-<yourname>**.
Go to **AWS Glue>>Crawler>>Create Crawler**

The screenshot shows the AWS Glue console interface for creating a new crawler. The left sidebar contains navigation links for Data Catalog, Data Integration and ETL, and other services. The main area is titled 'Add new crawler' and shows a progress bar with five steps. Step 1, 'Set crawler properties', is the current step. It includes a 'Crawler details' section with a 'Name' field (containing 'crawler-raw-layer') and a 'Description' field (with a placeholder 'Enter a description'). Below these is a 'Tags' section with a note to use tags to organize resources. At the bottom right are 'Cancel' and 'Next' buttons.

- Select S3 as data source and choose path as **s3://tlc-trip-record-data-lake-<yourname-random number>/raw-layer/2020/Q2/**

The screenshot shows the 'Add data source' dialog box. It has a 'Data source' dropdown menu set to 'S3'. Below it is a 'Network connection' section with a dropdown menu and a refresh button. There are 'Clear selection' and 'Add new connection' buttons. The 'Location of S3 data' section has two radio buttons: 'In this account' (selected) and 'In a different account'. The 'S3 path' section has a text input field containing 's3://tlc-trip-record-data-lake-110/raw-l...', a 'View' button, and a 'Browse' button. At the bottom, there is a 'Subsequent crawler runs' section with a note and a radio button for 'Crawl all sub-folders'.

6. Select **GlueCapstoneRole** as IAM role.

The screenshot shows the 'Configure security settings' step in the AWS Glue console. On the left, a sidebar lists four steps: Step 1 (Set crawler properties), Step 2 (Choose data sources and classifiers), Step 3 (Configure security settings, which is the active step), and Step 4. The main content area is titled 'Configure security settings' and features an 'IAM role' section. It includes a dropdown menu for 'Existing IAM role' with 'GlueCapstoneRole' selected, a 'Create new IAM role' button, and an 'Update chosen IAM role' button. A note states: 'Only IAM roles created by the AWS Glue console and have the prefix "AWSGlueServiceRole-" can be updated.' There are also 'Info' and 'View' links.

7. Click on Add database.

The screenshot shows the 'Set output and scheduling' step in the AWS Glue console. The sidebar on the left shows Step 3 (Configure security settings) as the active step. The main content area is titled 'Set output and scheduling' and contains an 'Output configuration' section. It features a 'Target database' dropdown menu with 'Choose a database' selected, a 'Clear selection' button, and an 'Add database' button. There are also 'Info' and refresh icons.

8. Name: **tlc-trip-record-data-db-<yourname>** and click on create database.

The screenshot shows the 'Create a database' form in the AWS Glue console. The title is 'Create a database' with a subtitle 'Create a database in the AWS Glue Data Catalog.' The form is divided into 'Database details' and includes fields for 'Name' (with the value 'tlc-trip-record-data-db'), 'Location - optional', and 'Description - optional'. A note at the bottom states: 'Descriptions can be up to 2048 characters long.' At the bottom right, there are 'Cancel' and 'Create database' buttons.

The screenshot shows the 'Databases' list in the AWS Glue console. The title is 'Databases (1)' with a subtitle 'A database is a set of associated table definitions, organized into a logical group.' Below the title, there is a list of databases with columns for 'Name', 'Description', and 'Location URI'. The first database listed is 'tlc-trip-record-data-db'. There are buttons for 'Add database', 'Edit', and 'Delete' for each database entry. A search bar labeled 'Filter databases' is also present.

9. Now add the newly created database as Target database.

The screenshot shows the 'Add new crawler' wizard in the AWS Glue console. The 'Set output and scheduling' step is active. On the left, a sidebar lists four steps: 'Set crawler properties', 'Choose data sources and classifiers', 'Configure security settings', and 'Set output and scheduling'. The main area is titled 'Set output and scheduling' and contains an 'Output configuration' section. In this section, the 'Target database' dropdown is set to 'tlc-trip-record-data-db'. Below it are 'Clear selection' and 'Add database' buttons. The 'Table name prefix - optional' field is set to 'raw_'. A refresh button is located to the right of the database dropdown.

10. Keep crawler schedule frequency as On demand and create the crawler.

The screenshot shows the 'Crawler schedule' section of the AWS Glue console. It includes a description: 'You can define a time-based schedule for your crawlers and jobs in AWS Glue. The definition of these schedules uses the Unix-like cron syntax. Learn more.' Below this, the 'Frequency' dropdown menu is set to 'On demand'.

11. Once the crawler is created, run the crawler.

The screenshot shows the 'Crawler runs' section of the AWS Glue console. It displays a table of crawler runs for a specific crawler. The table has columns for 'Start time (UTC)', 'End time (UTC)', 'Current/last duration', 'Status', and 'DPU hours'. One run is shown with a status of 'Running' and a duration of '17 s'.

Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours
[Redacted]	[Redacted]	17 s	Running	-

The screenshot shows the 'Crawler runs' section of the AWS Glue console, displaying a table of crawler runs. The table has columns for 'Start time (UTC)', 'End time (UTC)', 'Current/last duration', 'Status', and 'DPU hours'. One run is shown with a status of 'Completed' and a duration of '01 min 02 s'.

Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours
[Redacted]	[Redacted]	01 min 02 s	Completed	-

12. See the table got created for raw-layer and explore the metadata.

AWS Glue

Data Catalog

Databases

Tables

Stream schema registries

Schemas

Connections

Crawlers

Classifiers

Catalog settings

Data Integration and ETL

AWS Glue Studio

Jobs

Crawler successfully starting

The following crawler is now starting: "crawler-raw-layer"

AWS Glue > Tables

Tables

A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Tables (1/1)

View and manage all available tables.

Refresh

Delete

Data quality

Add tables using crawler

Add table

Filter tables

Name

Database

Location

Classification

Dep

raw_q2

tlc-trip-record-data-db

s3://tlc-trip-record-data-lake-mi-110/raw-layer/2020/Q2/

csv

-

Schema

Partitions

Indexes

Schema (18)

View and manage the table schema.

Filter schemas

Edit schema as JSON

Edit schema

#	Column name	Data type	Partition key	Comment
1	vendorid	bigint	-	-
2	tpcp_pickup_datetime	string	-	-
3	tpcp_dropoff_datetime	string	-	-
4	passenger_count	bigint	-	-
5	trip_distance	double	-	-
6	ratecodeid	bigint	-	-
7	store_and_fwd_flag	string	-	-
8	pulocationid	bigint	-	-
9	dolocationid	bigint	-	-
10	payment_type	bigint	-	-
11	fare_amount	double	-	-
12	extra	double	-	-
13	mta_tax	double	-	-
14	tip_amount	double	-	-
15	tolls_amount	double	-	-
16	improvement_surcharge	double	-	-
17	total_amount	double	-	-
18	congestion_surcharge	double	-	-

13. Now let's create the crawler for reference layer. Give the name to crawler as **crawler-reference-layer-<yourname>**



AWS Glue > Crawlers > Add new crawler

Step 1
Set crawler properties

Step 2
Choose data sources and classifiers

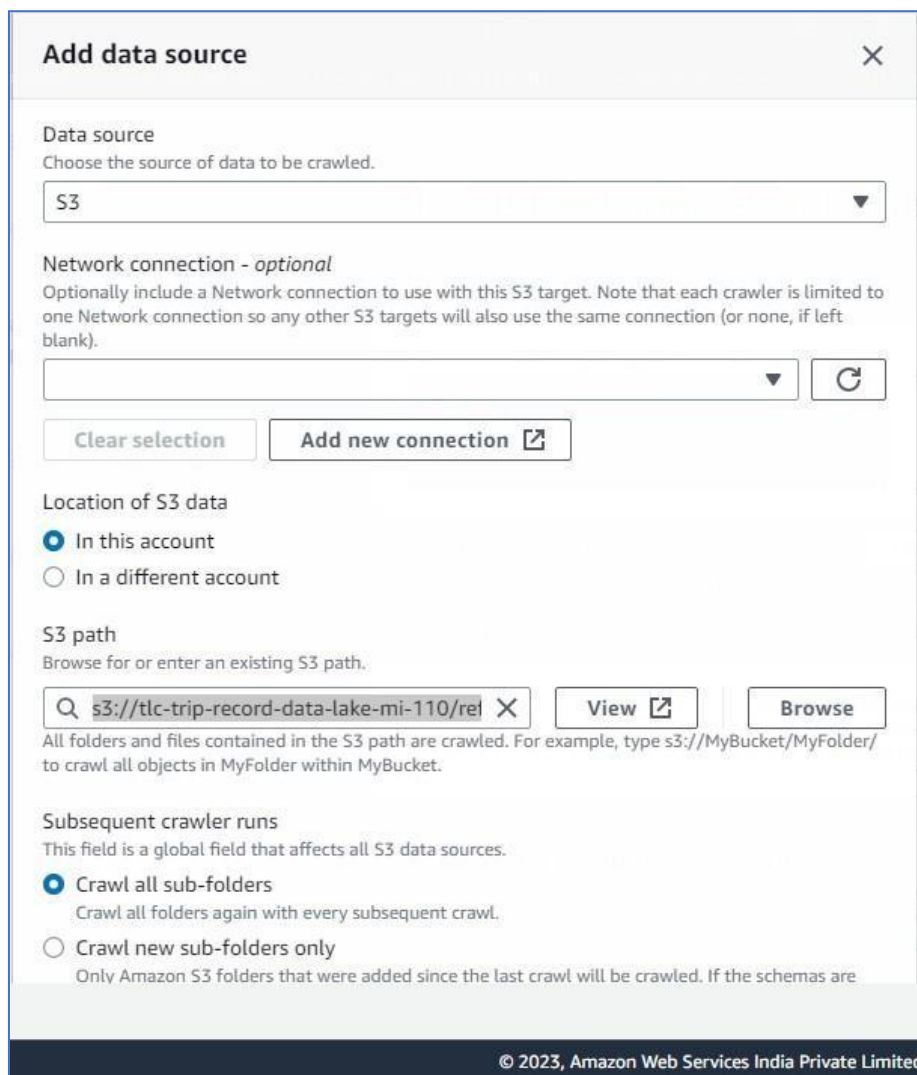
Step 3

Set crawler properties

Crawler details [Info](#)

Name
crawler-reference-layer

14. Choose S3 as data source and give path as **s3://tlc-trip-record-data-lake-<yourname-random number>/reference-layer/**



Add data source

Data source
Choose the source of data to be crawled.
S3

Network connection - optional
Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any other S3 targets will also use the same connection (or none, if left blank).

Clear selection Add new connection

Location of S3 data
☒ In this account
☐ In a different account

S3 path
Browse for or enter an existing S3 path.
s3://tlc-trip-record-data-lake-mi-110/re View Browse
All folders and files contained in the S3 path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Subsequent crawler runs
This field is a global field that affects all S3 data sources.
☒ Crawl all sub-folders
Crawl all folders again with every subsequent crawl.
☐ Crawl new sub-folders only
Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are

© 2023, Amazon Web Services India Private Limited

15. Choose **GlueCapstoneRole** as an IAM Role.

The screenshot shows the 'Configure security settings' step in the AWS Glue console. On the left, a sidebar lists four steps: Step 1 (Set crawler properties), Step 2 (Choose data sources and classifiers), Step 3 (Configure security settings), and Step 4 (Set output and scheduling). The main area is titled 'Configure security settings' and contains an 'IAM role' section. It features a dropdown menu for 'Existing IAM role' with 'GlueCapstoneRole' selected, a 'Create new IAM role' button, and an 'Update chosen IAM role' button. A note states: 'Only IAM roles created by the AWS Glue console and have the prefix 'AWSGlueServiceRole-' can be updated.'

16. Select an existing database **tlc-trip-record-data-db-<yourname>** as target database.

The screenshot shows the 'Set output and scheduling' step in the AWS Glue console. The sidebar on the left shows Step 4 as the active step. The main area is titled 'Set output and scheduling' and contains an 'Output configuration' section. It includes a 'Target database' dropdown menu with 'tlc-trip-record-data-db' selected, a 'Clear selection' button, and an 'Add database' button. Below this is a 'Table name prefix - optional' text input field with the placeholder text 'Type a prefix added to table names'.

17. Keep crawler schedule frequency as On-Demand and create the crawler.

The screenshot shows the 'Crawler schedule' section in the AWS Glue console. It includes a description: 'You can define a time-based schedule for your crawlers and jobs in AWS Glue. The definition of these schedules uses the Unix-like cron syntax. Learn more.' Below this is a 'Frequency' dropdown menu with 'On demand' selected.

18. Once the crawler for reference layer is created, run it and check one table for reference layer is being created and explore the metadata.

The screenshot shows the 'Crawler runs' table in the AWS Glue console. The table has columns for Start time (UTC), End time (UTC), Current/last duration, Status, and DPU hours. A single row is visible, showing a crawler run that is 'Running' and has a duration of '10 s'. The table is titled 'Crawler runs (1)' and includes a search bar and a filter by date and time range.

Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours
[REDACTED]	[REDACTED]	10 s	Running	-

Crawler runsScheduleData sourcesClassifiersTags

Crawler runs (1)

The list of crawler runs for this crawler.

Filter data

Filter by a date and time range

< 1 > ⚙

Start time (UTC) ▲

End time (UTC) ▼

Current/last duration ▼

Status ▼

DPU hours ▼

○

53 s

✔ Completed

-

AWS Glue

×

▼ Data Catalog

Databases

Tables

Stream schema registries

Schemas

Connections ↗

Crawlers

Classifiers

Catalog settings New

▼ Data Integration and ETL

AWS Glue Studio

AWS Glue > Tables

Tables

A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Tables (1/2)

View and manage all available tables.

↻

Delete

Data quality

Add tables using crawler

Add table

Filter tables

< 1 > ⚙

	Name	Database	Location	Classification	De
<input type="checkbox"/>	raw_q2	tlc-trip-record-data-db	s3://tlc-trip-record-data-lake-mi-110/raw-layer/2020/Q2/	csv	-
<input checked="" type="checkbox"/>	reference_layer	tlc-trip-record-data-db	s3://tlc-trip-record-data-lake-mi-110/reference-layer/	csv	-

SchemaPartitionsIndexes

Schema (4)

View and manage the table schema.

Filter schemas

< 1 > ⚙

Edit schema as JSON

Edit schema

#	Column name	Data type	Partition key	Comment
1	locationid	bigint	-	-
2	borough	string	-	-
3	zone	string	-	-
4	service_zone	string	-	-

Phase-3: Ad-Hoc Exploration – SQL Analytics

19. Go to your S3 bucket and create a new folder named as **results**.

Amazon S3 > Buckets > tlc-trip-record-data-lake-mi-110 > Create folder

Create folder Info

Use folders to group objects in buckets. When you create a folder, S3 creates an object using the name that you specify followed by a slash (/). This object then appears as folder on the console. [Learn more](#)

Your bucket policy might block folder creation

If your bucket policy prevents uploading objects without specific tags, metadata, or access control list (ACL) grantees, you will not be able to create a folder using this configuration. Instead, you can use the [upload](#) configuration to upload an empty folder and specify the appropriate settings.

Folder

Folder name

 /

Folder names can't contain "/" or ". See rules for naming

20. Go to **Athena>>Query Editor>>Settings>> Manage** and set the **Query Result location and encryption** as **s3://tlc-trip-record-data-<yourname-randomnumber>/results** and save.

Amazon Athena > Query editor

Editor | Recent queries | Saved queries | **Settings**

Workgroup primary

Query result and encryption settings

[Manage](#)

Choose S3 data set

S3 buckets > tlc-trip-record-data-lake-mi-110

Objects (1/4)

 [Refresh](#)

	Key	Last modified	Size
<input type="radio"/>	processed-layer	-	-
<input type="radio"/>	raw-layer	-	-
<input type="radio"/>	reference-layer	-	-
<input checked="" type="radio"/>	results	-	-

Cancel [Choose](#)

Amazon Athena > Query editor > Manage settings

Manage settings

Query result location and encryption

Location of query result - *optional*
Enter an S3 prefix in the current region where the query result will be saved as an object.

[View](#) [Browse S3](#)

21. Now perform ad-hoc SQL Analytics on raw table and reference table.

Data

Query 4

1 SELECT * FROM "tlc-trip-record-data-db"."raw_q2" limit 10;

Data source

AwsDataCatalog

Database

tlc-trip-record-data-db

Tables and views

Create

Filter tables and views

▼ Tables (2)

raw_q2

reference_layer

SQL Ln 1, Col 59

Run again

Explain

Cancel

Clear

Create

Reuse query results
*Athena engine version 3 only

Query results

Query stats

Completed

Time in queue: 151 ms

Run time: 536 ms

Data scanned: 988.55 KB

Results (10)

Copy

Download results

Search rows

< 1 >

#	vendorid	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	ratecodeid	store_atg
1	1	2020-06-01 00:31:23	2020-06-01 00:49:58	1	3.6	1	N
2	1	2020-06-01 00:42:50	2020-06-01 01:04:33	1	5.6	1	N
3	1	2020-06-01 00:39:51	2020-06-01 00:49:09	1	2.3	1	N
4	1	2020-06-01 00:56:13	2020-06-01 01:11:38	1	5.3	1	N

Query 5

1 SELECT * FROM "tlc-trip-record-data-db"."reference_layer" limit 10;

Data source

AwsDataCatalog

Database

tlc-trip-record-data-db

Tables and views

Create

Filter tables and views

▼ Tables (2)

raw_q2

reference_layer

SQL Ln 1, Col 1

Run again

Explain

Cancel

Clear

Create

Reuse query results
*Athena engine version 3 only

Query results

Query stats

✔ Completed

Time in queue: 171 ms

Run time: 594 ms

Data scanned: 12.03 KB

Results (10)

Copy

Download results

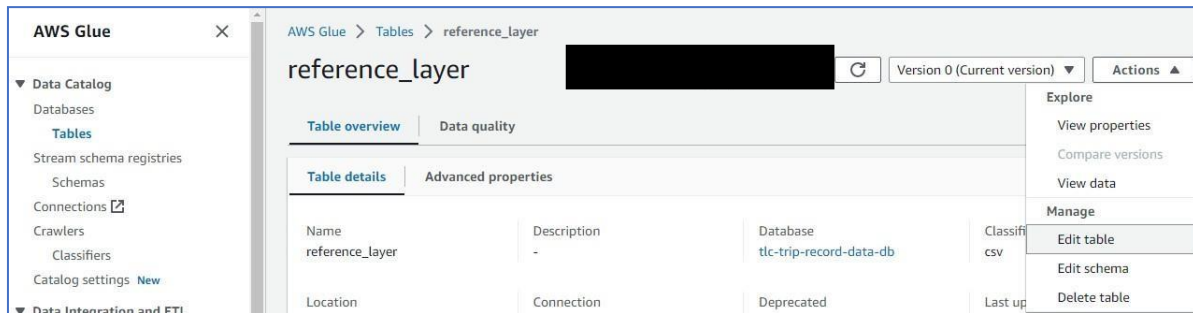
Q Search rows

< 1 >

⚙

#	locationid	borough	zone	service_zone
1	1	"EWR"	"Newark Airport"	"EWR"
2	2	"Queens"	"Jamaica Bay"	"Boro Zone"
3	3	"Bronx"	"Allerton/Pelham Gardens"	"Boro Zone"
4	4	"Manhattan"	"Alphabet City"	"Yellow Zone"

22. Now let's get rid of these double quotes you saw in last query execution for reference data. Go to **AWS Glue >> Tables >> your reference layer table >> Actions >> Edit table >>** Under **Serialization lib**: replace the value with **org.apache.hadoop.hive.serde2.OpenCSVSerde**

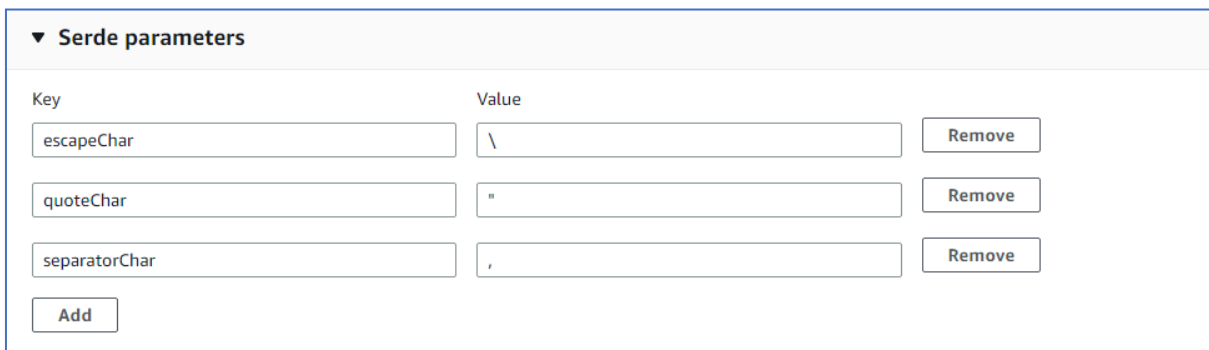


The screenshot shows the AWS Glue console interface. On the left, the 'Data Catalog' sidebar is visible. The main area displays the 'reference_layer' table details. The 'Serialization lib' field is highlighted, and the 'Edit table' action is selected from the 'Actions' menu.

Serialization lib

org.apache.hadoop.hive.serde2.OpenCSVSerde

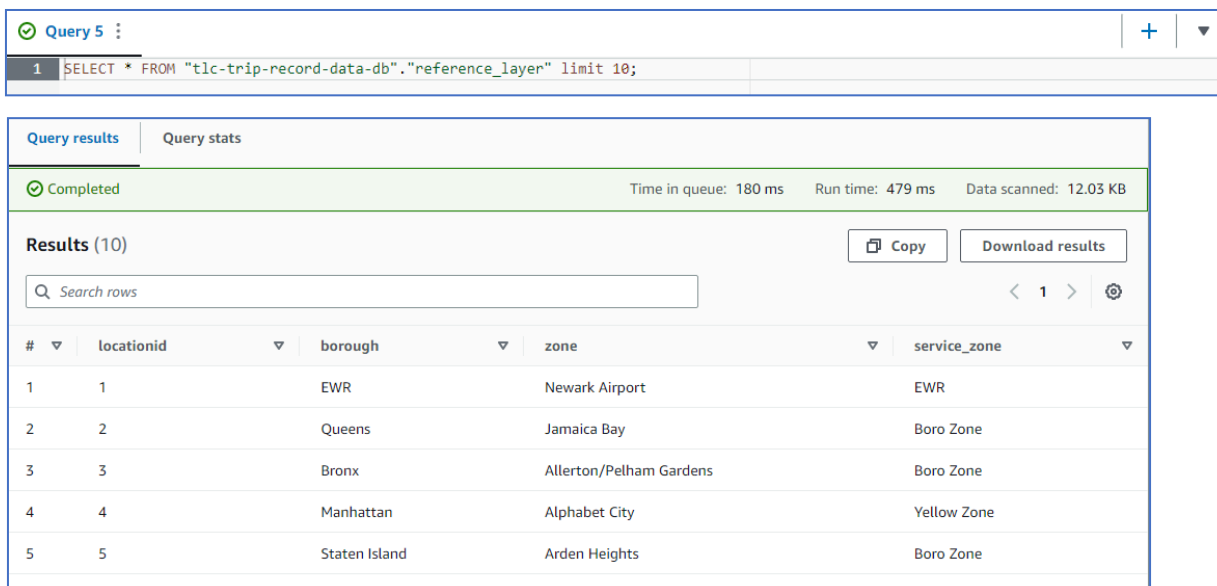
23. And update the **Serde parameters** as follows:



The screenshot shows the 'Serde parameters' section in the AWS Glue console. It displays a table with columns 'Key' and 'Value'. The parameters are: escapeChar (backslash), quoteChar (double quote), and separatorChar (comma). There is an 'Add' button and 'Remove' buttons for each parameter.

Key	Value
escapeChar	\
quoteChar	"
separatorChar	,

24. Now go back to **Athena** and query your reference table and see the results.



The screenshot shows the AWS Athena console. The top section displays the query execution status: 'Query 5' is 'Completed'. The query text is: `SELECT * FROM 'tlc-trip-record-data-db'.'reference_layer' limit 10;`. Below the query, the 'Query results' tab is selected, showing the results of the query. The results are displayed in a table with columns: #, locationid, borough, zone, and service_zone. The results show 10 rows of data.

Query 5 : `SELECT * FROM 'tlc-trip-record-data-db'.'reference_layer' limit 10;`

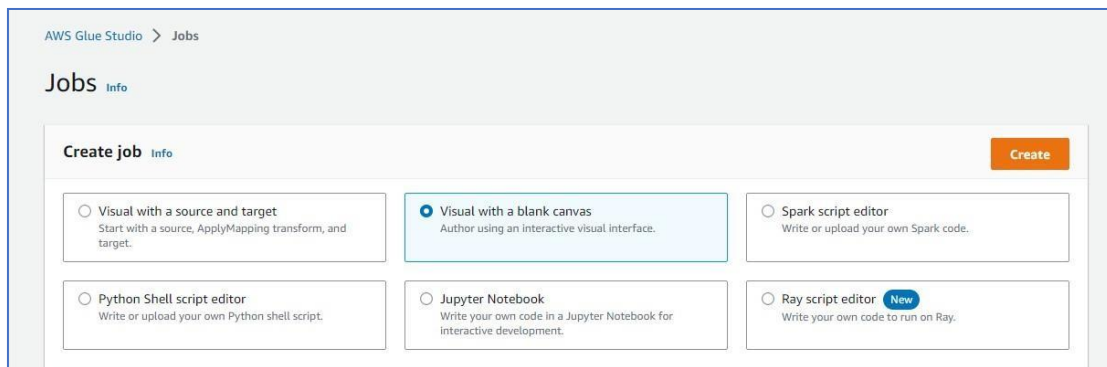
Completed Time in queue: 180 ms Run time: 479 ms Data scanned: 12.03 KB

Results (10)

#	locationid	borough	zone	service_zone
1	1	EWR	Newark Airport	EWR
2	2	Queens	Jamaica Bay	Boro Zone
3	3	Bronx	Allerton/Pelham Gardens	Boro Zone
4	4	Manhattan	Alphabet City	Yellow Zone
5	5	Staten Island	Arden Heights	Boro Zone
6	6	Staten Island	Arden Heights	Boro Zone
7	7	Staten Island	Arden Heights	Boro Zone
8	8	Staten Island	Arden Heights	Boro Zone
9	9	Staten Island	Arden Heights	Boro Zone
10	10	Staten Island	Arden Heights	Boro Zone

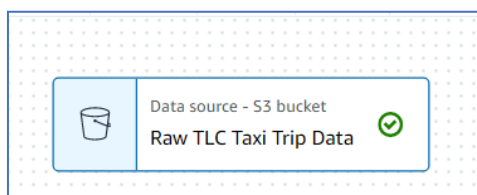
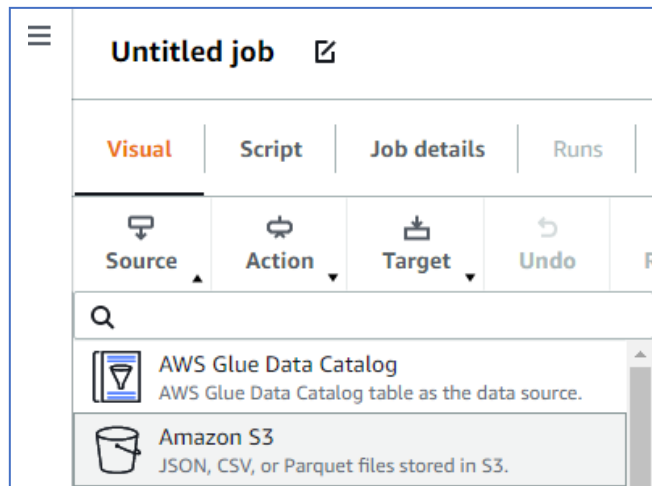
Phase-4: Processing - Let's Transform Taxi Ride Data

25. Go to **AWS Glue Studio >> Jobs >> Visual with a blank canvas >> Create**



26. Now, let's start building our ETL workflow.

Select **Source>>S3**.



Node properties >> Name > Raw TLC Taxi Trip Data

Node properties | Data source properties - S3 | Output schema

Data preview

Name
Raw TLC Taxi Trip Data

Node type
Choose which type of node to add to the job.
Amazon S3
JSON, CSV, or Parquet files stored in S3.

Data Source Properties – S3: Select **Data Catalog table** as **S3 source type**. Select **your database** that you created earlier in **Glue Data catalog**. Select **raw table** you got earlier.

Node properties | **Data source properties - S3** | Output schema

Data preview

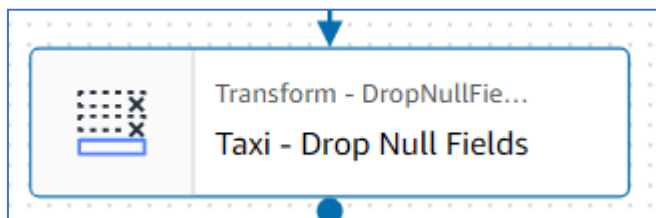
S3 source type [Info](#)
☒ Data Catalog table
☐ S3 location
Choose a file or folder in an S3 bucket.

Database
Choose a database.
tlc-trip-record-data-db

► Use runtime parameters

Table
raw_q2

27. Now, let's add transformation after source node. From **actions >> choose DropNullFields**



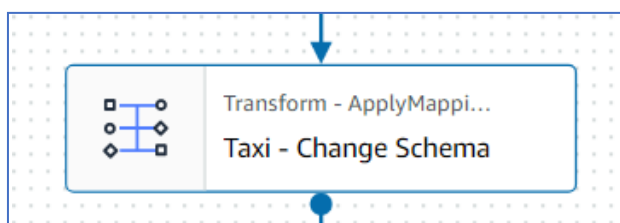
Node properties: Name>>Taxi – Drop Null Fields

The screenshot shows the 'Node properties' tab for a 'Drop Null Fields' node. The 'Name' field is set to 'Taxi - Drop Null Fields'. The 'Node type' is 'Drop Null Fields' with a description: 'Remove columns that have only empty/null values.' The 'Node parents' section shows 'Raw TLC Taxi Trip Data' as the parent, with 'S3 - DataSource' listed below it.

Transform: Select Empty String ("" or "") and "null" String

The screenshot shows the 'Transform' tab for the 'DropNullFields' node. The description is 'Remove fields or columns where all the values are the null objects.' Under 'Choose additional values that represent a null value below.', the checkboxes for 'Empty String ("" or "")' and '"null" String' are checked, while '-1 Integer' is unchecked. There is a section for 'Add custom null values' with a description 'Specify custom null values by entering the value and choosing the datatype.' and an 'Add new value' button.

28. Add next transform, **actions >> ApplyMapping**.



Node properties: Taxi – Change Schema

Node properties	Transform	Output schema	Data preview
Name Taxi - Change Schema			
Node type Choose which type of node to add to the job. Change Schema (Apply Mapping) Change field names & data types and drop fields.			
Node parents Choose which nodes will provide inputs for this one. Select parents			
Taxi - Drop Null Fields DropNullFields - Transform			

Transform:

Rename the columns as:

vendor_id, pickup_datetime, dropoff_datetime

Node properties	Transform	Output schema	Data preview
Apply mapping			
Source key	Target key	Data type	Drop
vendorid	vendor_id	long	<input type="checkbox"/>
tpcp_pickup_datetime	pickup_datet	string	<input type="checkbox"/>
tpcp_dropoff_datetime	dropoff_date	string	<input type="checkbox"/>

Drop ratecodeid and store_and_fwd_flag columns.

ratecodeid	<input checked="" type="checkbox"/>
store_and_fwd_flag	<input checked="" type="checkbox"/>

Rename: pickup_locationid and drop_off_locationid.

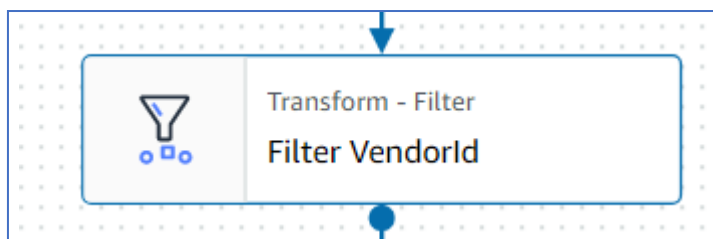
pulocationid	pickup_locati	long	<input type="checkbox"/>
dolocationid	drop_off_loci	long	<input type="checkbox"/>

Drop the following columns.

fare_amount	<input checked="" type="checkbox"/>
extra	<input checked="" type="checkbox"/>
mta_tax	<input checked="" type="checkbox"/>
tip_amount	<input checked="" type="checkbox"/>
tolls_amount	<input checked="" type="checkbox"/>
improvement_surcharge	<input checked="" type="checkbox"/>

congestion_surcharge	<input checked="" type="checkbox"/>
----------------------	-------------------------------------

29. Let's add next transformation. **Actions>>Filter.**



Node properties: Filter VendorId

Node properties

Transform

Output schema

Data preview

Name

Filter VendorId

Node type

Choose which type of node to add to the job.

Filter

Filter data based on conditions.

Node parents

Choose which nodes will provide inputs for this one.

Select parents

Taxi - Change Schema

ApplyMapping - Transform

Taxi - Change Schema X
ApplyMapping - Transform

Transform: Choose Global OR and in filter condition add filter condition as mentioned.

Node properties

Transform

Output schema

Data preview

Filter

Info

Builds a new output by selecting records from the input data that satisfy a specified predicate function

☐

Global AND

All filter conditions will be applied as a global "AND."

☒

Global OR

All filter conditions will be applied as a global "OR."

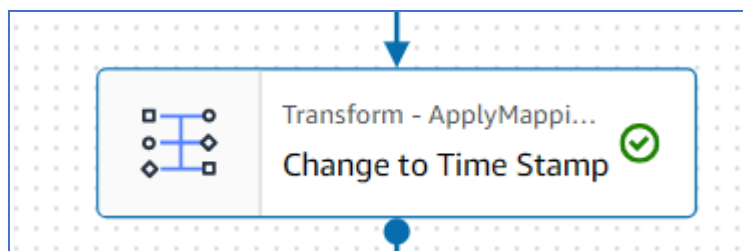
Filter condition

Info

Specify your filter condition by choosing the key, operator, and entering a value.

Key	Operation	Value	
vendor_id ▼	= ▼	1	
vendor_id ▼	= ▼	2	

30. Add next transform. Actions >> ApplyMapping.



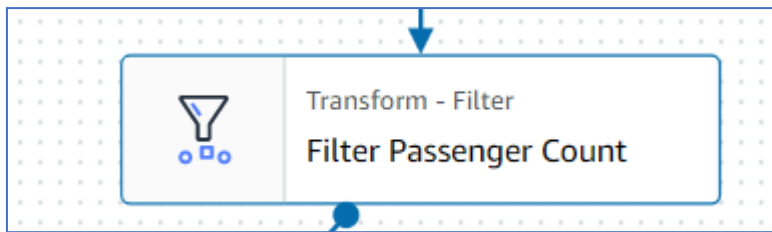
Node properties: Name as Change to Time Stamp

Transform: Change the data type of the two columns mentioned below to timestamp.





pickup_datetime	pickup_datet	times... ▼	<input type="checkbox"/>
dropoff_datetime	dropoff_date	times... ▼	<input type="checkbox"/>

31. Add next transform. **Actions >> Filter.**

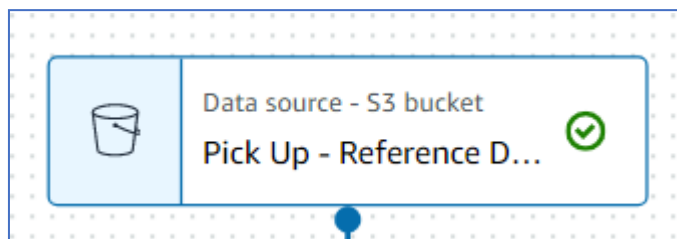
Node properties: Name as Filter Passenger Count



Transform: Select Global AND and add filter condition as shown below.

Node properties	Transform	Output schema	Data preview 								
<p>Filter Info</p> <p>Builds a new output by selecting records from the input data that satisfy a specified predicate function</p> <p><input checked="" type="radio"/> Global AND All filter conditions will be applied as a global "AND."</p> <p><input type="radio"/> Global OR All filter conditions will be applied as a global "OR."</p> <p>Filter condition Info</p> <p>Specify your filter condition by choosing the key, operator, and entering a value.</p> <table><thead><tr><th>Key</th><th>Operation</th><th>Value</th><th></th></tr></thead><tbody><tr><td>passenger_count ▼</td><td>> ▼</td><td>0</td><td></td></tr></tbody></table>				Key	Operation	Value		passenger_count ▼	> ▼	0	
Key	Operation	Value									
passenger_count ▼	> ▼	0									

32. Now click in the blank space on the canvas and add another S3 source.



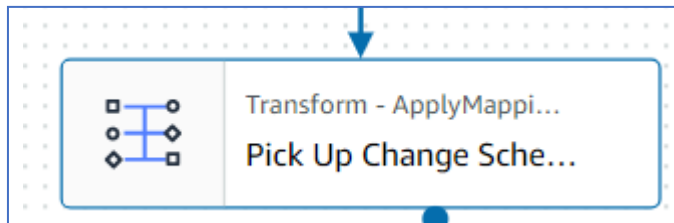
Node properties: Name as Pick Up - Reference Data

Node properties	Data source properties - S3	Output schema	
Data preview			
Name			
<input type="text" value="Pick Up - Reference Data"/>			
Node type			
Choose which type of node to add to the job.			
<div>Amazon S3 JSON, CSV, or Parquet files stored in S3.</div>			

Data source properties: Choose Data Catalog table as S3 source. Choose your Glue Database you created earlier. Select reference table, this time.

Node properties	Data source properties - S3	Output schema	
Data preview			
S3 source type Info			
<input checked="" type="radio"/> Data Catalog table			
<input type="radio"/> S3 location Choose a file or folder in an S3 bucket.			
Database			
Choose a database.			
<input type="text" value="tlc-trip-record-data-db"/>			
<input type="button" value="↻"/>			
▶ Use runtime parameters			
Table			
<input type="text" value="reference_layer"/>			
<input type="button" value="↻"/>			

33. Add an action next to your node "Pick Up – Reference Data" as Action >> ApplyMapping.



Node properties: Pick Up Change Schema

Node properties | Transform | Output schema | Data preview

Name
Pick Up Change Schema

Node type
Choose which type of node to add to the job.
Change Schema (Apply Mapping)
Change field names & data types and drop fields.

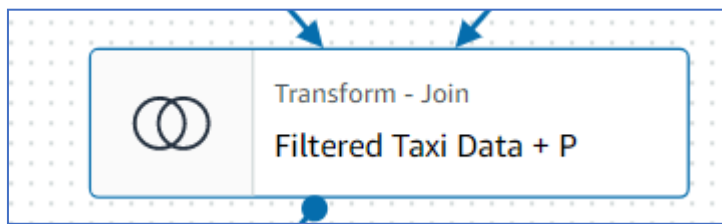
Node parents
Choose which nodes will provide inputs for this one.
Select parents

Pick Up - Reference Data X
S3 - DataSource

Transform: Rename the columns as follows: Append pick_ before every name in Target key.

Node properties	Transform	Output schema	Data preview
Apply mapping			
Source key	Target key	Data type	Drop
locationid	pick_location	long	<input type="checkbox"/>
borough	pick_borough	string	<input type="checkbox"/>
zone	pick_zone	string	<input type="checkbox"/>
service_zone	pick_service_	string	<input type="checkbox"/>

34. Click in the blank space on canvas. Add Join transform.



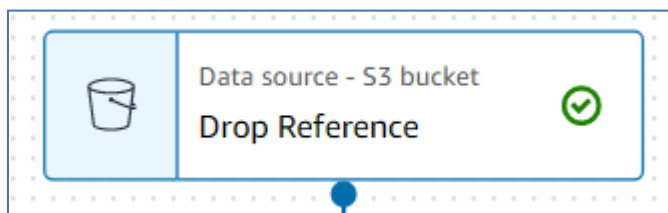
Node properties: Name as Filtered Taxi Data + P and select Pick Up Change Schema and Filter Passenger Count as node parents.

Node properties	Transform	Output schema	Data preview
<p>Name</p> <p>Filtered Taxi Data + P</p>			
<p>Node type</p> <p>Choose which type of node to add to the job.</p> <p>Join Combine records from two datasets based on a set of conditions.</p>			
<p>Node parents</p> <p>Choose which nodes will provide inputs for this one.</p> <p>Select parents</p> <div><div>Pick Up Change Schema ApplyMapping - Transform</div><div>Filter Passenger Count Filter - Transform</div></div>			

Transform: Apply join condition as follows.

Node properties	Transform	Output schema	Data preview
<p>Join type</p> <p>Select the type of join to perform.</p> <p>Inner join Select all rows from both datasets that meet the join condition.</p>			
<p>Join conditions</p> <p>Select a field from each parent node for the join condition.</p> <div><div>Pick Up Change Schema pickup_locationid</div><div>=</div><div>Filter Passenger Count pickup_location_id</div><div>✕</div></div> <p>Add condition</p>			

35. Click in blank space on canvas and one more S3 source.



Node properties: Name as Drop Reference

Node properties | Data source properties - S3 | Output schema |

Data preview

Name

Node type
Choose which type of node to add to the job.

Amazon S3
JSON, CSV, or Parquet files stored in S3.

Data source: Select Data Catalog table as S3 source type. Select Glue Database created earlier and select reference table again.

Node properties | **Data source properties - S3** | Output schema |

Data preview

S3 source type [Info](#)
☒ Data Catalog table
☐ S3 location
Choose a file or folder in an S3 bucket.

Database
Choose a database.

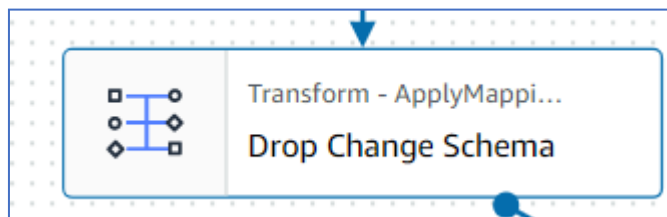
tlc-trip-record-data-db

► Use runtime parameters

Table

reference_layer

36. Add ApplyMapping transform to this new source.



Node properties: Name as Drop Change Schema

Node properties | Transform | Output schema | Data preview

Name

Node type
Choose which type of node to add to the job.

Change Schema (Apply Mapping)

Change field names & data types and drop fields.

Node parents
Choose which nodes will provide inputs for this one.

Select parents

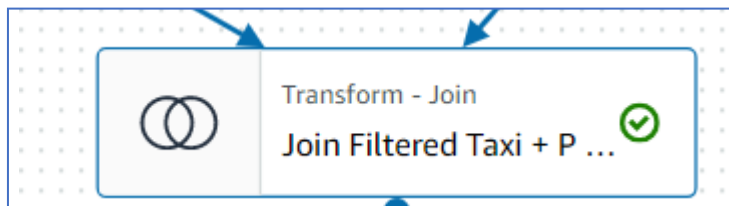
Drop Reference

S3 - DataSource

Transform: Simply append drop_ for each column in target key as shown below.

Node properties	Transform	Output schema	Data preview
Apply mapping			
Source key	Target key	Data type	Drop
locationid	<input type="text" value="drop_locationid"/>	<input type="text" value="long"/>	<input type="checkbox"/>
borough	<input type="text" value="drop_borough"/>	<input type="text" value="string"/>	<input type="checkbox"/>
zone	<input type="text" value="drop_zone"/>	<input type="text" value="string"/>	<input type="checkbox"/>
service_zone	<input type="text" value="drop_service_zone"/>	<input type="text" value="string"/>	<input type="checkbox"/>

37. Click in the blank space on the canvas. Add one more Join transform.



Node properties: Name as Join Filtered Taxi + P + D and select Drop Change Schema and Filtered Taxi Data + P as node parents.

Node properties

Transform

Output schema

Data preview

Name

Join Filtered Taxi + P + D

Node type

Choose which type of node to add to the job.

Join

Combine records from two datasets based on a set of conditions.

Node parents

Choose which nodes will provide inputs for this one.

Select parents

Drop Change Schema

Filtered Taxi Data + P

ApplyMapping - Transform

Join - Transform

Transform: Use the join condition as shown below.

Node properties

Transform

Output schema

Data preview

Join type

Select the type of join to perform.

Inner join

Select all rows from both datasets that meet the join condition.

Join conditions

Select a field from each parent node for the join condition.

Filtered Taxi Data + P

Drop Change Schema

drop_off_location_id

=

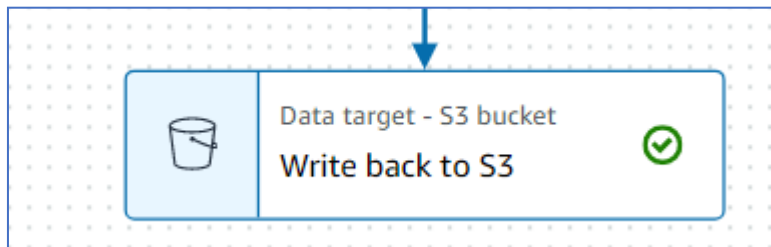
drop_locationid

Add condition

38. Now after final joined node i.e. Join Filtered Taxi + P + D, add ApplyMapping and drop these two columns.

pickup_location_id	<input checked="" type="checkbox"/>
drop_off_location_id	<input checked="" type="checkbox"/>

39. Last step, add target. Select S3.



Node properties: Name as Write back to S3

Node properties

Data target properties - S3

Output schema

Data preview

Name

Write back to S3

Node type

Choose which type of node to add to the job.

Amazon S3

Output data directly in an S3 bucket.

Node parents

Choose which nodes will provide inputs for this one.

Select parents

Change Schema (Apply Mapping)

×

ApplyMapping - Transform

Data target: Format: Parquet, Compression: Snappy and target path as
s3://tlc-trip-record- data-lake-mi-<yourname-randownumber>/processed-layer/

Node properties

Data target properties - S3

Output schema

Data preview

Format

Parquet

Compression Type

Snappy

S3 Target Location

Choose an S3 location in the format s3://bucket/prefix/object/ with a trailing slash (/).

Q

s3://tlc-trip-record-data-lake

X

View

Browse S3

Data Catalog update options

Info

Choose how you want to update the Data Catalog table's schema and partitions. These options will only apply if the Data Catalog table is an S3 backed source.

☒ Do not update the Data Catalog

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

☐ Create a table in the Data Catalog and on subsequent runs, keep existing schema and add new partitions

Partition keys - optional

Add partition keys.

40.Go to Job Details Tab: Name as Transforming Taxi Ride Data and select GlueCapstoneRole

Transforming Taxi Ride Data

Visual

Script

Job details

Runs

Data quality

Schedules

Version Control

Basic properties

Info

Name

Transforming Taxi Ride Data

Description - optional

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.

GlueCapstoneRole

Make sure you go with the following configuration as shown below:

Type

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

Spark

Glue version [Info](#)

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

Language

Python 3

Worker type

Set the type of predefined worker that is allowed when a job runs.

G 1X
(4vCPU and 16GB RAM)

☐ Automatically scale the number of workers

AWS Glue will optimize 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.

2

☐ 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 [Info](#)

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).

Disable

Give a unique name to your script. <unique name>.py

▼ Advanced properties

Script filename

demoscript.py

Script path

☐ Job metrics [Info](#)

Enable the creation of CloudWatch metrics when

☐ Continuous logging [Info](#)

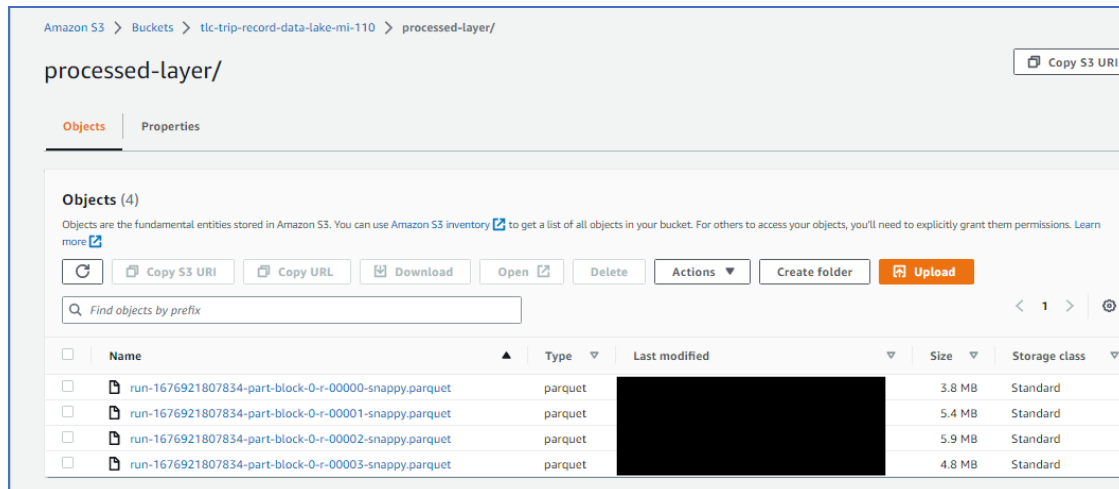
Enable logs in CloudWatch.

☐ Spark UI [Info](#)

Enable using Spark UI for monitoring this job.

41. Save the job and run.

42. Watch for the output in processed layer in S3.

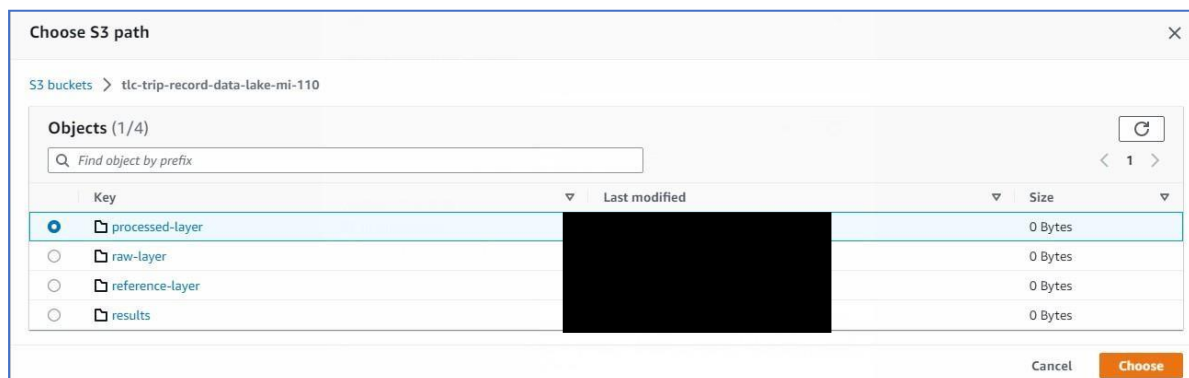


43. Go AWS Glue >> **Create a crawler for processed layer.** Name as **crawler-processed-layer-<yourname>**



Choose S3 as data source.

Provide s3://tlc-trip-record-data-lake-<yourname-randomnumber>/processed-layer/ as S3 path



Add data source

Data source

Choose the source of data to be crawled.

S3

Network connection - optional

Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any other S3 targets will also use the same connection (or none, if left blank).

Clear selection
Add new connection

Location of S3 data

☒ In this account
☐ In a different account

S3 path

Browse for or enter an existing S3 path.

View
Browse

All folders and files contained in the S3 path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Subsequent crawler runs

This field is a global field that affects all S3 data sources.

☒ Crawl all sub-folders
☐ Crawl new sub-folders only

Crawl all folders again with every subsequent crawl.
Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are

© 2023, Amazon Web Services India Private Limited

44. Select **GlueCapstone** as an IAM role.

AWS Glue > Crawlers > Add new crawler

Step 1
Set crawler properties

Step 2
Choose data sources and classifiers

Step 3
Configure security settings

Step 4

Configure security settings

IAM role Info

Existing IAM role

GlueCapstoneRole

Create new IAM role
Update chosen IAM role

Only IAM roles created by the AWS Glue console and have the prefix "AWSGlueServiceRole-" can be updated.

45. Select your existing Glue database as target database and create the crawler. Run it.

AWS Glue > Crawlers > Add new crawler

Step 1
Set crawler properties

Step 2
Choose data sources and classifiers

Step 3
Configure security settings

Set output and scheduling

Output configuration Info

Target database
tlc-trip-record-data-db

Clear selection Add database

46. You will be able to see metadata table for processed layer as well.

AWS Glue

✓ Crawler successfully starting
The following crawler is now starting: "crawler-processed-layer"

AWS Glue > Tables

Tables

A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Tables (1/3)

Filter tables

Name	Database	Location	Classification
processed_layer	tlc-trip-record-data-db	s3://tlc-trip-record-data-lake-mi-110/processed-layer/	parquet

47. Now, let's perform SQL Analytics using Athena on processed layer.

Data

Data source: AwsDataCatalog

Database: tlc-trip-record-data-db

Tables and views: Create

Filter tables and views

Tables (3): processed_layer, raw_q2, reference_layer

Query 5: SELECT * FROM "tlc-trip-record-data-db"."processed_layer" limit 10;

SQL Ln 1, Col 68

Run again Explain Cancel Clear Create

Reuse query results
*Athena engine version 3 only

Results (10)

Search rows

#	pick_locationid	pick_borough	pick_zone	pick_service_zone	vendor_id	pickup_datetime	dropoff_datetime
1	132	Queens	JFK Airport	Airports	1	2020-04-10 08:13:33.000	2020-04-10 08:13:33.000
2	132	Queens	JFK Airport	Airports	2	2020-04-13 23:46:04.000	2020-04-13 23:46:04.000
3	132	Queens	JFK Airport	Airports	1	2020-04-26 12:37:45.000	2020-04-26 12:37:45.000
4	132	Queens	JFK Airport	Airports	1	2020-05-18 14:06:56.000	2020-05-18 14:06:56.000
5	132	Queens	JFK Airport	Airports	2	2020-05-22 17:30:56.000	2020-05-22 17:30:56.000
6	132	Queens	JFK Airport	Airports	2	2020-05-24 12:47:28.000	2020-05-24 12:47:28.000

48. Do some more analysis.

Amazon Athena > Query editor

Editor | Recent queries | Saved queries | Settings

Workgroup: primary

Data

Data source: AwsDataCatalog

Database: tlc-trip-record-data-db

Tables and views:

▼ Tables (3) < 1 >

Query 5: Query 6:

```
1 SELECT payment_type, pick_borough, SUM(total_amount) "Total Revenue" from processed_layer
2 GROUP BY payment_type, pick_borough
3 ORDER BY payment_type, pick_borough ASC;
```

SQL Ln 3, Col 36

#	payment_type	pick_borough	Total Revenue
2	1	Brooklyn	249289.99000000206
3	1	EWR	2669.5
4	1	Manhattan	9386629.220000826
5	1	Queens	1096478.5399999865
6	1	Staten Island	12408.75000000002
7	1	Unknown	132535.98000000115
8	2	Bronx	49637.86999999938
9	2	Brooklyn	68562.0299999986
10	2	EWR	439.3499999999997
11	2	Manhattan	4229139.020006562
12	2	Queens	480157.6900000231

Stage-5: Visualisation (Optional)

49. Sign up for Amazon QuickSight Account.

Create your QuickSight account

Enterprise + QBack

Authentication method

☒ Use IAM federated identities & QuickSight-managed users
Authenticate with single sign-on (SAML or OpenID Connect), AWS IAM credentials, or QuickSight credentials

☐ Use IAM federated identities only
Authenticate with single sign-on (SAML or OpenID Connect) or AWS IAM credentials

☐ Use Active Directory
Authenticate with Active Directory credentials

QuickSight region

You can still use QuickSight in a region without Q, or select a Q-supported region.

Select a region ⓘ
US East (N. Virginia) ▼

Account info

QuickSight account name

You will need this for you and others to sign in ⓘ

Notification email address

For QuickSight to send important notifications

QuickSight access to AWS services

Make your existing AWS data and users available in QuickSight. [Learn more](#)

IAM Role

☒ Use QuickSight-managed role (default)

☐ Use an existing role

Allow access and autodiscovery for these resources

☐ Amazon Redshift

☐ Amazon RDS

☐ IAM

☒ Amazon S3 (1 buckets selected)
[Select S3 buckets](#)

☒ Amazon Athena
Make sure you've chosen the right Amazon S3 buckets for QuickSight access

☐ Amazon S3 Storage Analytics

☐ AWS IoT Analytics

☐ Amazon OpenSearch Service

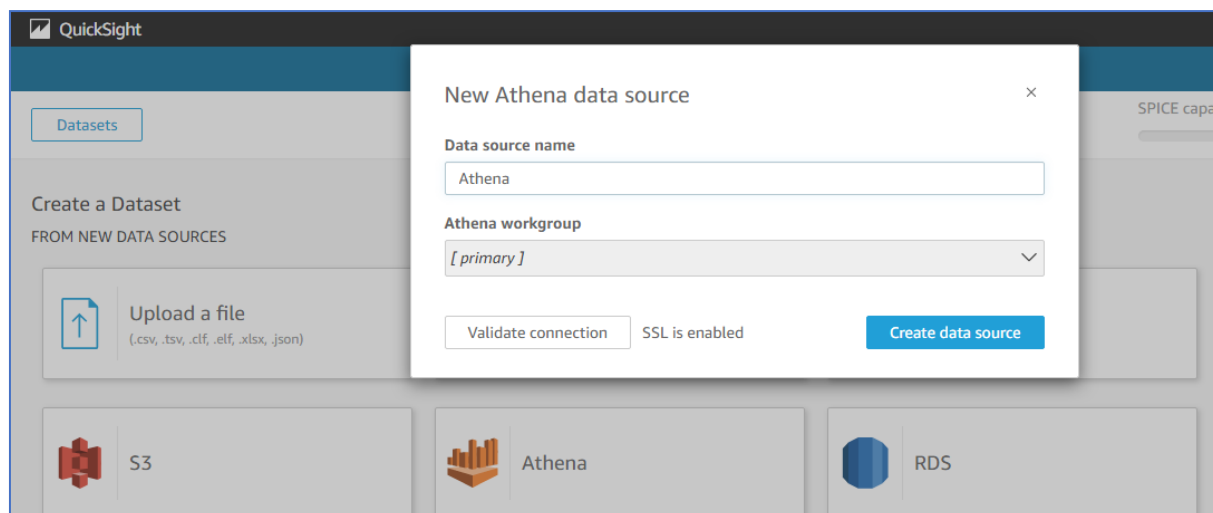
☐ Amazon SageMaker

☐ Amazon Timestream

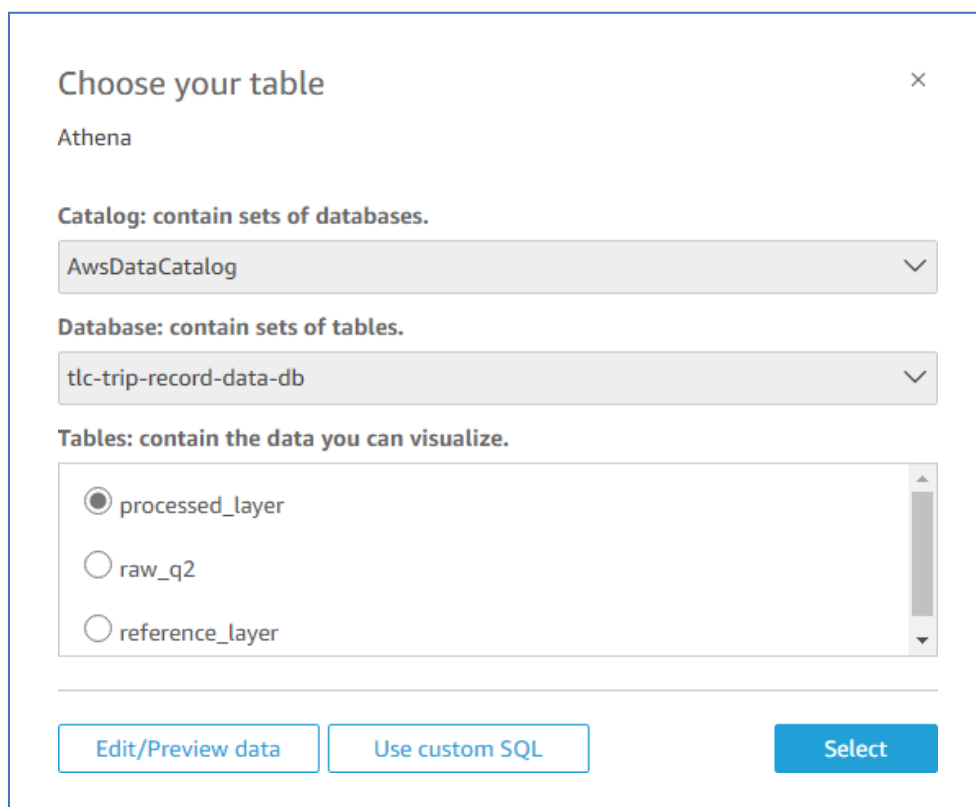
☐ AWS SecretsManager
[Select secrets](#)

Finish

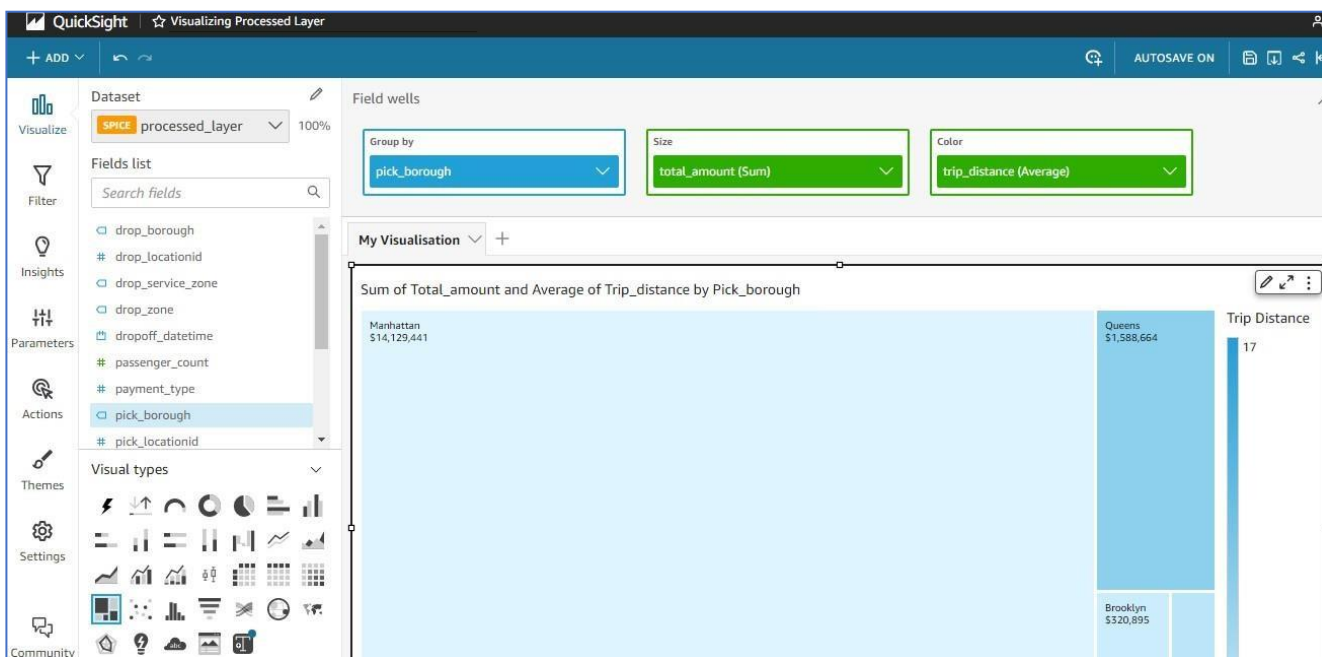
50. Once we will have an Amazon QuickSight Account, let's start adding new dataset. Choose **Athena as dataset source. Under Data Source name >. Athena. Create data source.**



51. Under **Choose your table >> Select your Glue Database. Table>> processed table and click select.**



52. Once the dataset is imported into SPICE, create visualization as required. Here is one sample visualization for you.



53. Save it and publish as a Dashboard as following:

Publish a dashboard [X]

☒ Publish new dashboard as
TLC Taxi Ride Analysis

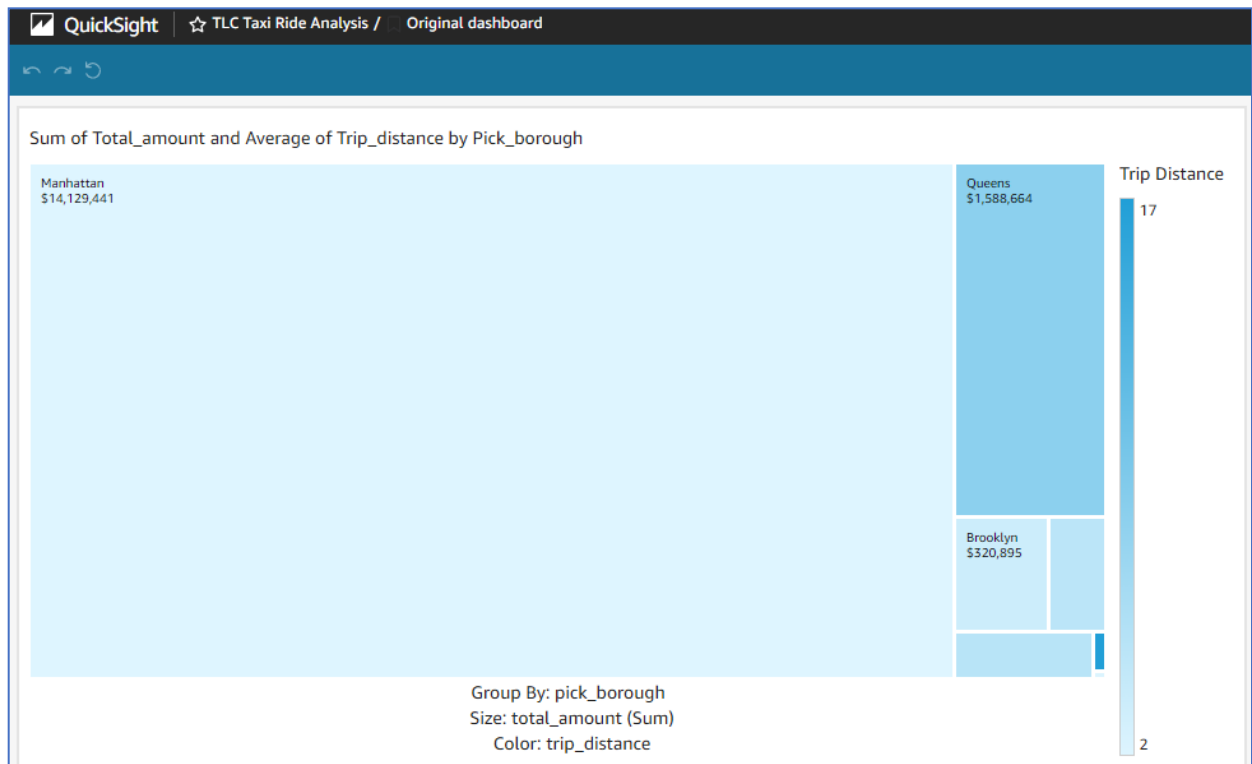
☐ Replace an existing dashboard

ALL SHEETS SELECTED [v]

☐ Enable topic for analysis ⓘ [Learn more about Q topic](#)

Advanced publish options [v]

Publish dashboard



54. Once done, terminate your Amazon QuickSight Account.

Account termination

QuickSight account name [REDACTED]

Account termination protection ⓘ
Account termination protection is an extra safe-guard to help prevent accidental deletion of accounts.

☐ Account termination protection is off.

Delete account
Deleting this account can't be undone and will permanently delete all users, dashboards, analyses, along with other related data.

Type "confirm" to delete this account