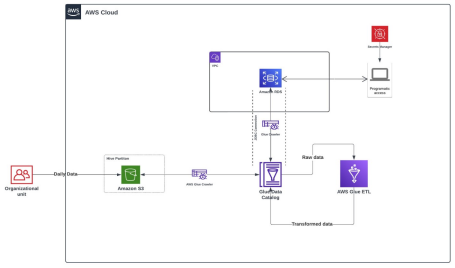
PS/Objective:

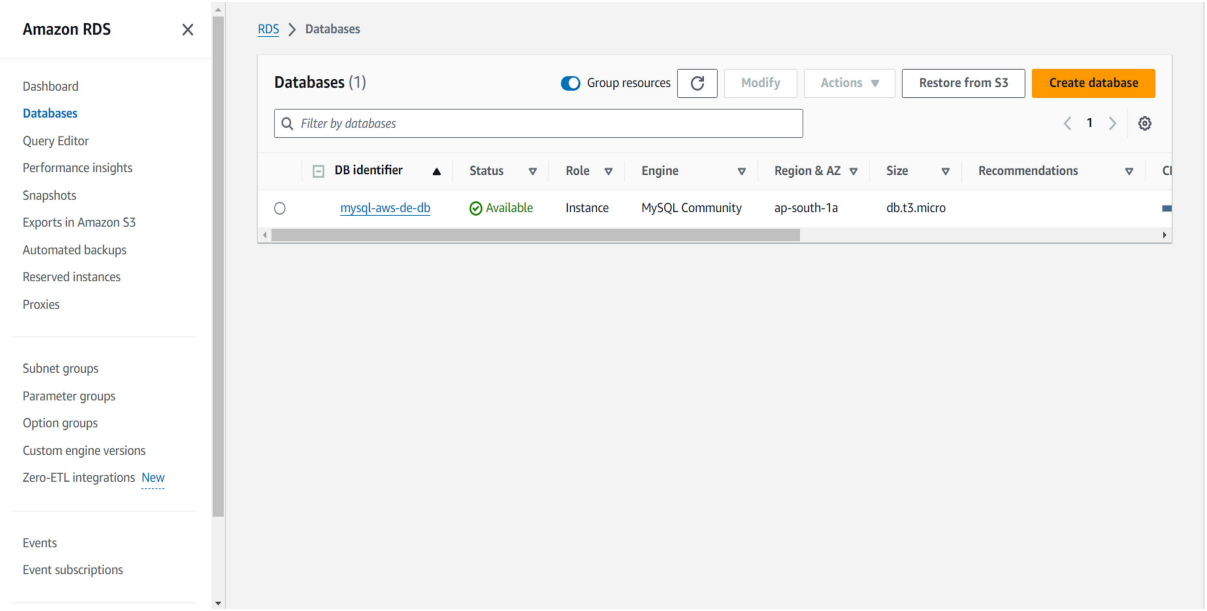
Implement an end-to-end data pipeline on AWS, filtering prepaid telecom data from daily CSV uploads to S3 using Glue ETL with distributed computation and securely ingesting them into RDS. Covering RDS setup, Glue catalog creation, JDBC connection establishment, visual ETL job creation with job bookmarking, and distributed computation using Glue ETL.



Step 1: Create RDS with the Following Parameters

1. Select RDS
2. Click on Databases
3. Create Database
4. Choose a Database Creation Method - Standard create
5. Select Latest Engine Version
6. Choose Free Tier Template (for educational purposes)
7. Provide Your Master Username (DB name)
8. Credentials Management - Choose self-managed (create your own password) and store the credentials in AWS Secrets Manager.
9. Public Access - Yes (strictly not recommended in production cases; since we are connecting to our RDS from outside the network, select yes. Ideally, in production, this should be 'NO').

   For the rest of the settings, keep them as default.

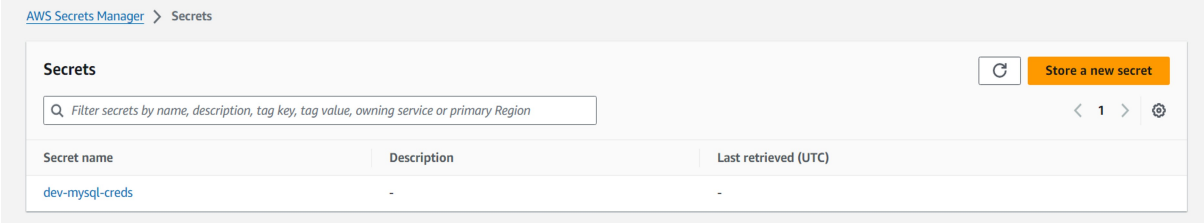


Keep note of the endpoint, port, username, and password - these are the things needed to establish a connection with the database.

Step 2: Store the RDS Credentials in AWS Secrets Manager

To avoid publicly exposing the credentials, securely store the RDS credentials in AWS Secrets Manager.

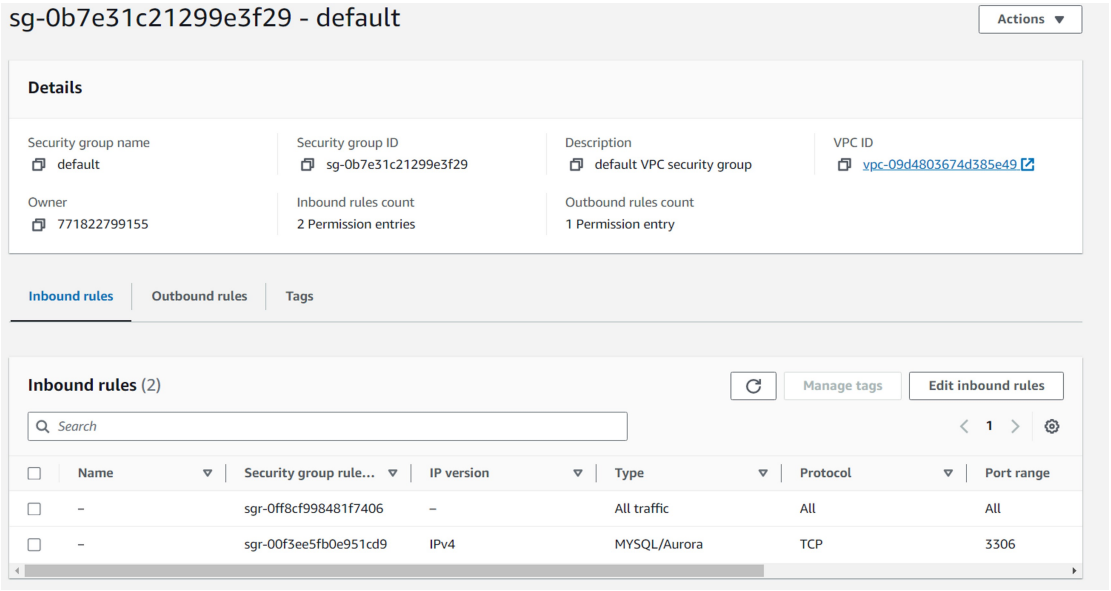
1. Select the AWS Secrets Manager service.
2. Select Store a new secret.
3. For Secret type, choose Credentials for Amazon RDS.
4. Input your credentials.
5. Choose the associated database.
6. Give the secret a name.
7. Click Store.



Step3: Configuring Security Group for Database Access

We need to open the port on which the database service is running to allow inbound traffic. Our RDS is running in a VPC, which has a security group. Within this security group, we need to allow inbound traffic on port 3306, the port on which the database is running.

1. Under the Connectivity & security section of the database instance, click on the VPC link.
2. Click on the security group associated with the VPC.
3. Click Edit inbound rules.
4. Click Add rule.
5. For Type, select MYSQL/Aurora.
6. For Port range, enter 3306.
7. For Source, enter 0.0.0.0/0 (to allow any IP).
8. Click Save rules.



Step4: Connect to the Database and create target db and table

A. Programmatically

In this step, we connect to the RDS instance and run some scripts. We need to install the MySQL connectors(pip install mysql-connector-python) to use functions and methods that will set up the connectivity, and Boto3 to connect to AWS Secrets Manager to retrieve the RDS credentials.

a. Fetch secrets from the Secrets Manager in JSON format (e.g., { "username": "xxxxxx", "password": "xxxxx" }).
b. Establish a connection with the RDS using the MySQL connector.
c. Execute our queries.

Check out the full code [here].

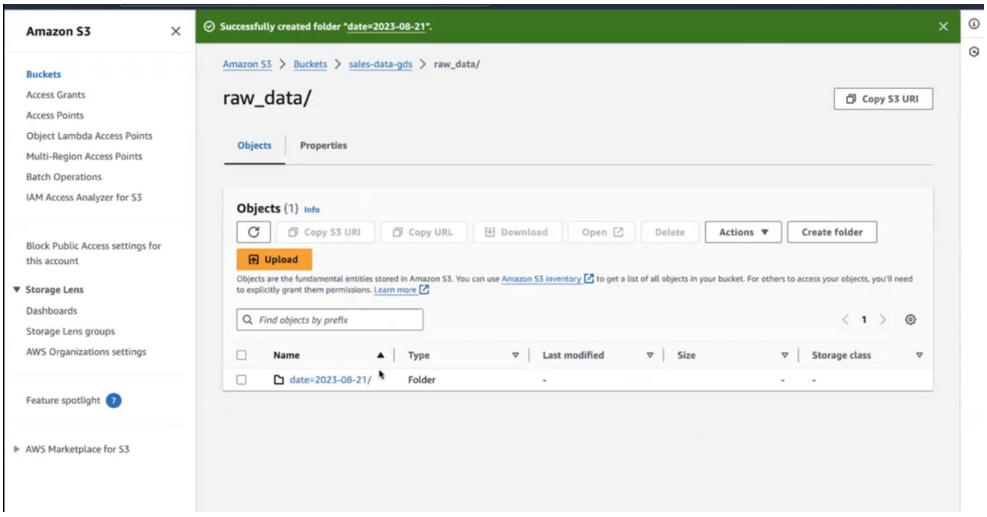
B. Through Terminal

You can also connect to your RDS instance using the terminal. Execute the following command to do so:
mysql -h <host/endpoint> -P <port> -u <username> -p

a. Enter your password when prompted.
b. After connecting, you can execute your SQL queries. [Link to the queries]

Step5: Setup the Data Lake - create an S3 Bucket for Raw Data

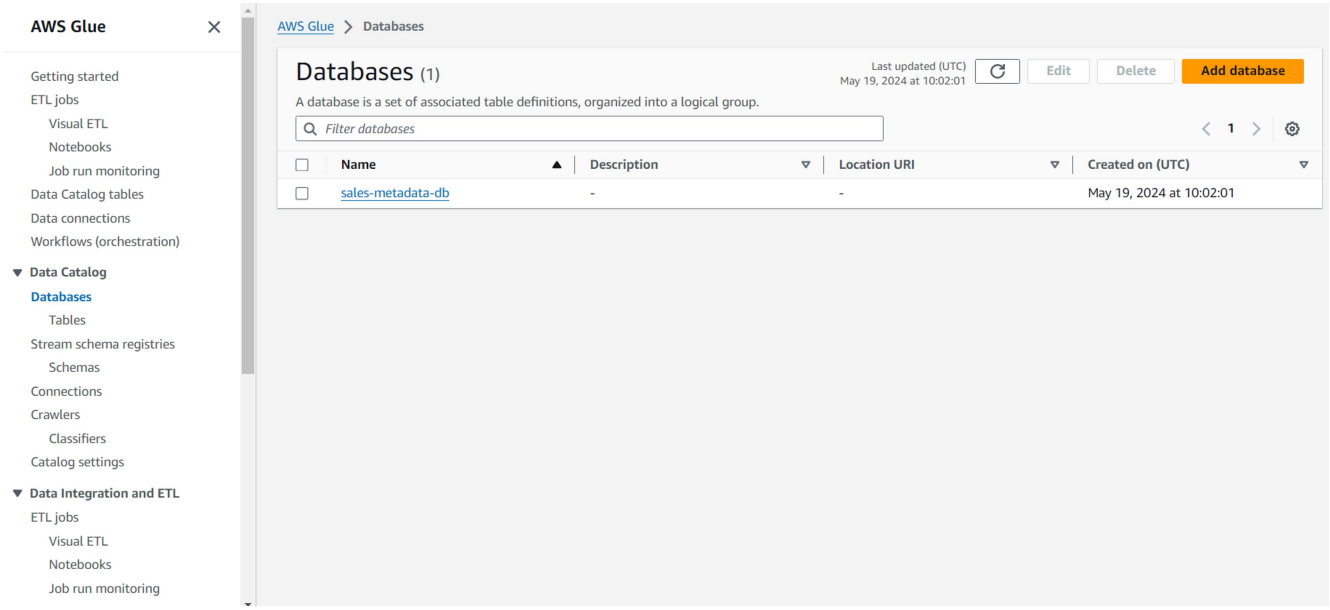
Create an S3 bucket where the client can upload CSV data in Hive partitions (using the date as the partition key).



Step 5: Create a Glue Catalog

Create a Glue catalog to serve as a central repository of metadata.
Since we need metadata for both the source and target, we will create two tables in the catalog.

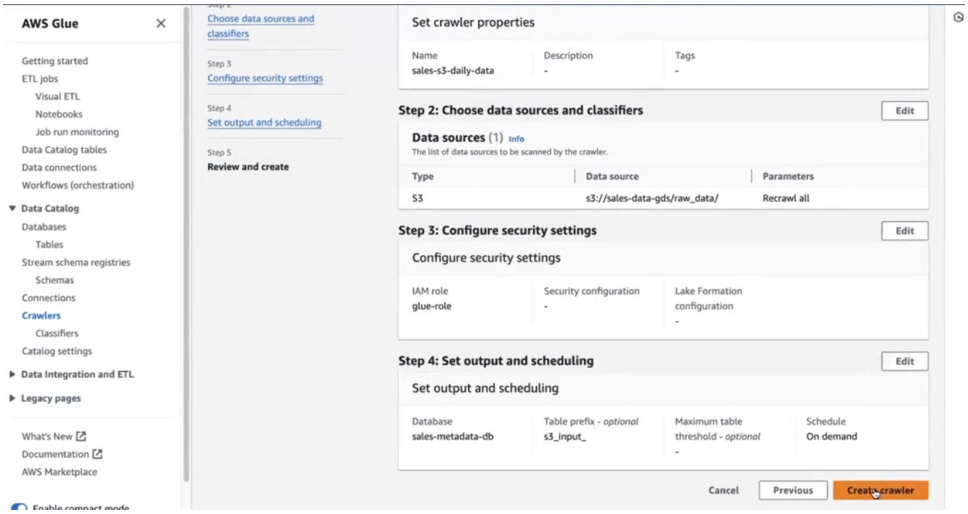
a. Under the Data Catalog section in AWS Glue, select Databases.
b. Give a name to the database.
c. Click on create



Step 6: Create a Crawler for the Source

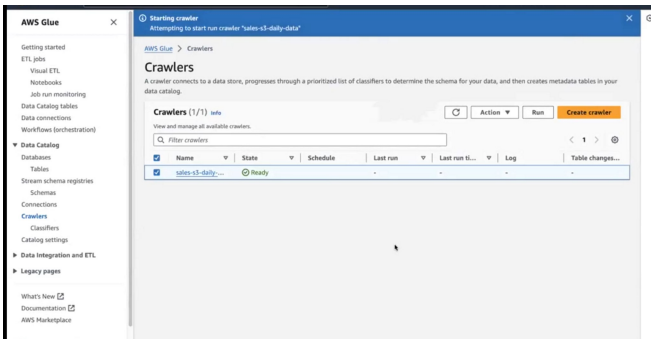
Create a crawler to fetch the metadata of the source/raw data in S3 and store it in a table in the Glue Catalog. For example, Crawler identifies the partition name, columns present in the partition, and their data types. Once we have this metadata, we can read data through the catalog instead of directly from S3 each time.

a. Select Crawlers in AWS Glue.
b. Click Create crawler.
c. Give the crawler a name.
d. For data source configuration, choose Is your data already mapped to Glue tables? - Not yet.
    - Click Add a data source:
    - Choose S3 as the source.
    - Select the source S3 bucket.
    - Select the path of the source folder (choose the top folder, not the file).
    - Select the Crawl all sub-folders option.
e. For IAM Role, since this crawler scans S3, it needs permission to access S3. Create a new role for this glue and add the S3 read access policy.
f. Select the previously created database in the Glue Catalog to store the source S3 metadata as a table.
g. Give the table a name.
h. Choose the On demand option for the crawler schedule. (We can automate this crawler as well, but for better understanding of things lets keep it ondemand)



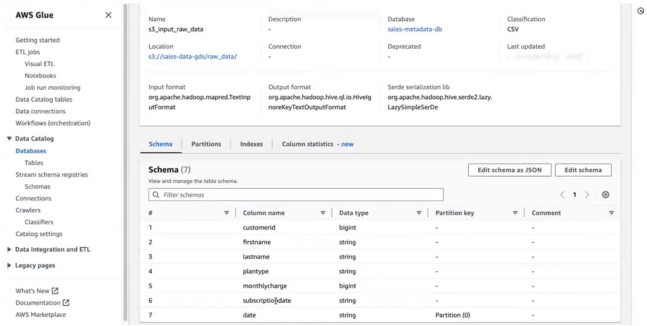
Step 7: Run the Crawler to Create the Schema

Once the crawler is created, select it and run it.



After the crawler runs successfully, follow these steps:

a. Open the Databases under the Data Catalog.
b. Select the database you created.
c. You will see a new table reflected there. Click on it to view the schema that the crawler has created.
d. If you want to modify the schema, use the Edit schema option.



Step 8: Create a JDBC connection to RDS.

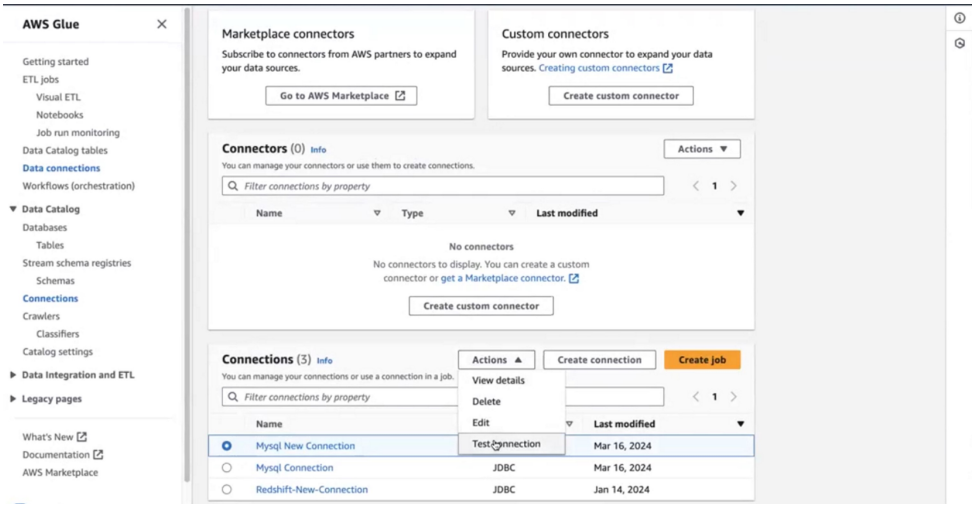
Once our source metadata is set up in the catalog, we need to establish a similar metadata table for the target. This facilitates seamless integration within the Glue ETL pipeline. For our target, which is RDS, we must create a connection to it. This connection will be utilized in our crawler to access the table.

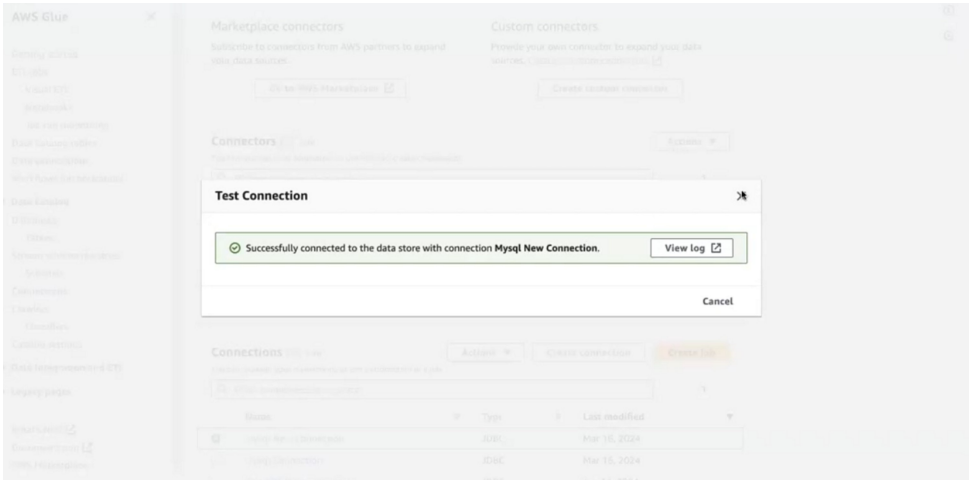
To create the connection:

- Navigate to "Connections" under Data Catalog.
- Choose "Create connection."
- Select "MySQL" as our RDS is MySQL.
- Choose the RDS instance for the connection.
- Specify the database you want to crawl in the RDS.
- Input RDS credentials.
- Provide a name and create the connection.

After creating the connection, it's essential to test it:

- Select the created connection.
- Under actions, click "Test connection."
- Grant permission to connect to RDS by selecting the previous IAM role and adding the RDS policy to it.
- Click "Confirm." If the connection is successful, you will receive confirmation.





Step 9: Endpoint creation

To ensure secure interaction between Glue, RDS, and S3, it's crucial to establish an endpoint. Glue relies on S3 for temporary data storage and metadata exchange over the network. This secure endpoint enables Glue to interact with S3, facilitating proper connection establishment.

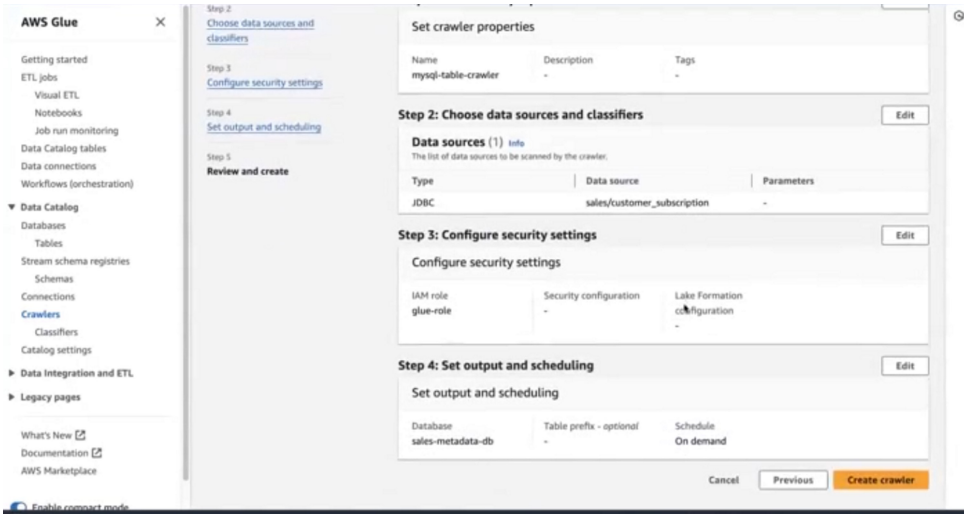
To create the endpoint:

- Navigate to VPC under the networking section in RDS.
- Click on "Endpoints."
- Choose "Create endpoint."
- Provide a name.
- Since we're creating an endpoint to S3, select S3 in the services and choose the one ending with ".s3" (not any other) and select the type as gateway.
- Select the same VPC where our RDS resides.
- Choose the all routing table.
- Apply the "Allow full access" policy.
- Click on "Create endpoint" to finalize the process.

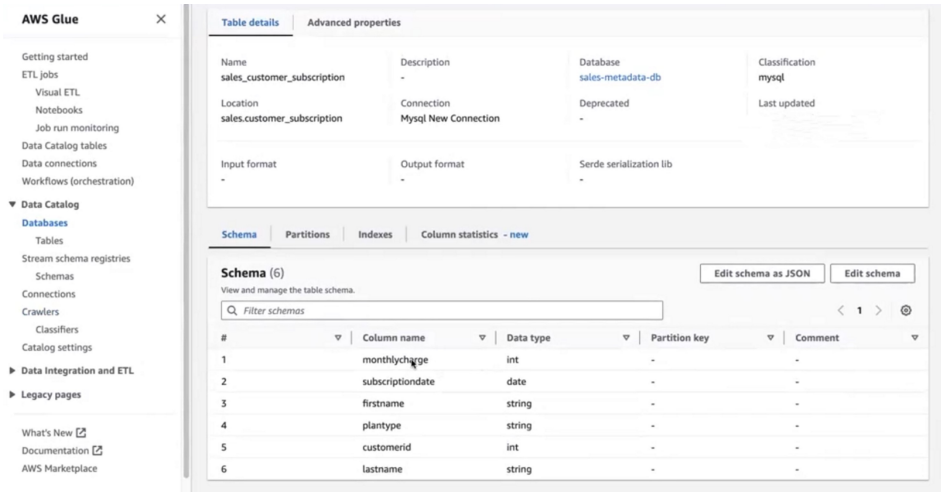
Step10: Create crawler for target table

Now that we've established our JDBC connection and created an endpoint, it's time to set up a crawler to extract data from our target table in RDS.

- Start by creating a new crawler.
- Provide a name for the crawler.
- For the data source, select the JDBC connection we previously set up.
- Specify the table you wish to crawl, ensuring the format is "databasename/tablename."
- Add the JDBC connection by clicking the corresponding button.
- Attach the Glue IAM role we created earlier.
- Choose the "On demand" option.
- Finally, initiate the creation of the crawler.



After creating the crawler, execute it to see the schema on the catalog, similar to what we did for the source crawler. Upon successful execution, the generated schema will be available.



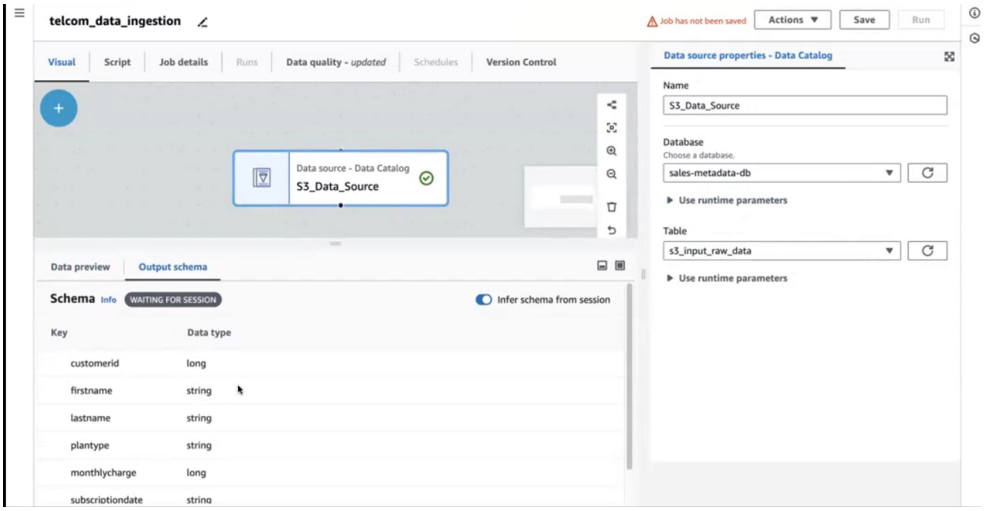
Step 11: Create an ETL Job Using Visual ETL

To handle incremental reads, we'll use Job Bookmarking. This feature keeps track of all the files read so far and looks for newly created objects.

ETL consists of three parts: Extract, Transform, and Load.

Part 1: Extract Data from Source

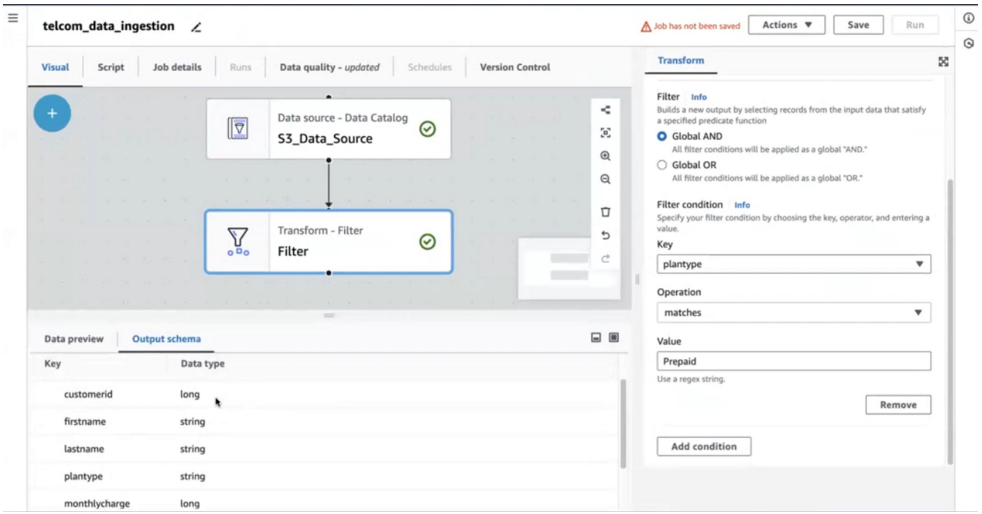
a. After selecting the Visual ETL option, provide a name for the job.
b. Click on 'Add Nodes.'
c. Under Source, choose Glue Data Catalog (this is where our metadata is stored).
d. In the Data Source Properties tab, provide a name for the data source. Choose the table and database from which the metadata will be read, and attach the Glue IAM role.
e. The system will automatically start fetching the data for you.



Part2: Transform The Data

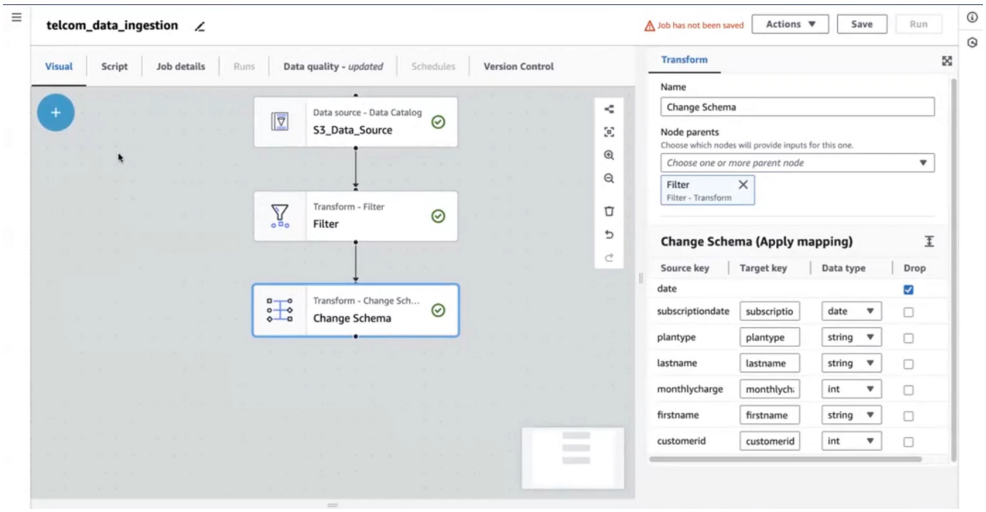
a. Click on 'Add Node' under the Transform section and choose the Filter option.
b. Provide a name for the transformation and add a filter condition:

c. a. Filter condition: Key: Plantype, Operation: matches, Value: Prepaid
d. Once saved, the filtered data schema will be displayed.
e. After the output schema is generated for the filter stage, review the properties and make any necessary edits if there are any mismatches.



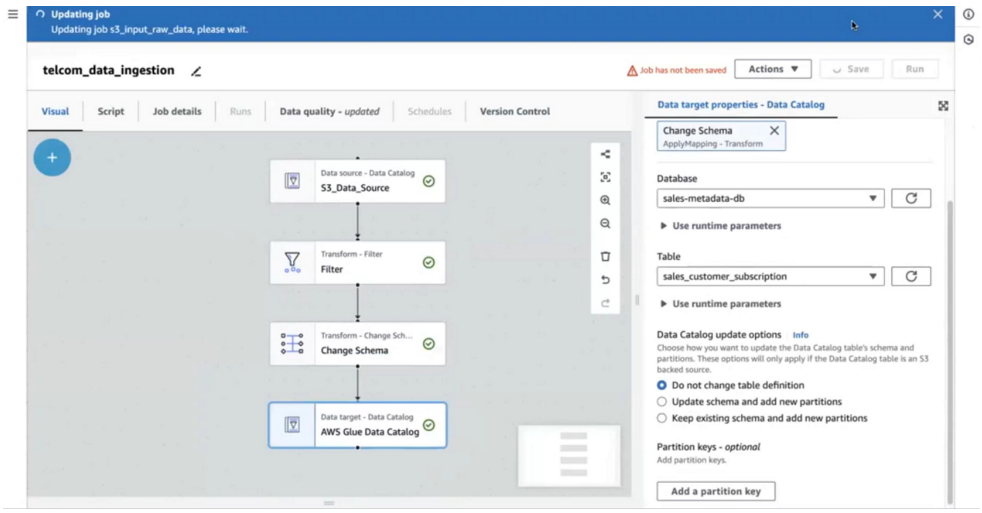
In our case, customerId is read as long instead of int, monthlyCharge as long instead of int, and subscriptionDate as string instead of date. Additionally, an extra column 'date' is present. We can use the change schema property under the Transform section to rectify this.

After selecting, you can change the key name, data type, or drop the column altogether as per requirements. Remember, the final schema of the Transform section should match the schema of the target table.



Part3: Add Target

a. Choose "Data Catalog" as our target is defined in the catalog.
b. Select the database and target table.
c. Click the "Save" button.

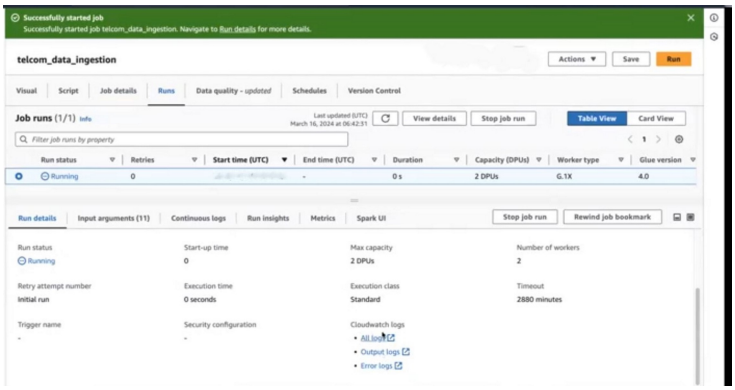


Step12: Define the Glue Infrastructure for Computation

Since Glue performs distributed computation, we need to select the runtime infrastructure. When the ETL job is executed, it will use this infrastructure to perform the computation.

a. Select the "Job details" tab.
b. Choose the Spark 3.3 version.
c. You can either choose the number of worker nodes or use the auto-scaling feature.
d. Enable job bookmarking for incremental reads.
e. Set the number of retries if the ETL job fails at any point, and configure the timeout settings.
f. Under the advanced settings, find the path to the script. This script is distributed to the nodes for computation.
g. Now, run your ETL pipeline by clicking on "Run."
h. Check the "Runs" section to see the execution process and the parameters passed.

This ensures that your Glue job is properly configured for efficient and reliable execution.



Once completed, query the database and check the results.

```
mysql> select * from customer_subscription;
ERROR 2013 (HY000): Lost connection to MySQL server during query
No connection. Trying to reconnect...
Connection id:    58
Current database: sales

+------------+-----------+-----------+----------+---------------+------------------+
| CustomerID | FirstName | LastName  | PlanType | MonthlyCharge | SubscriptionDate |
+------------+-----------+-----------+----------+---------------+------------------+
|       1102 | Abigail   | Johnson   | Prepaid  |            30 | 2022-02-15       |
|       1104 | Addison   | Jones     | Prepaid  |            28 | 2022-05-05       |
|       1106 | Aiden     | Davis     | Prepaid  |            35 | 2022-08-22       |
|       1108 | Alex      | Miller    | Prepaid  |            32 | 2023-02-28       |
|       1110 | Alexander | Rodriguez | Prepaid  |            29 | 2023-05-21       |
|       1112 | Alicia    | Anderson  | Prepaid  |            31 | 2023-07-15       |
|       1114 | Alvin     | Martinez  | Prepaid  |            30 | 2023-08-15       |
|       1116 | Amber     | Lee       | Prepaid  |            28 | 2023-10-05       |
|       1118 | Amy       | Gonzalez  | Prepaid  |            32 | 2023-11-11       |
|       1120 | Andrea    | Lewis     | Prepaid  |            33 | 2023-12-24       |
+------------+-----------+-----------+----------+---------------+------------------+
10 rows in set (2.23 sec)

mysql>
```

The 2023-08-21 CSV file contained 10 prepaid customers, which is now reflected in the database. If you want to check new folders or files, make sure to load the data onto S3, run the crawlers, and then execute the ETL pipeline, as we have configured them as on-demand functions.