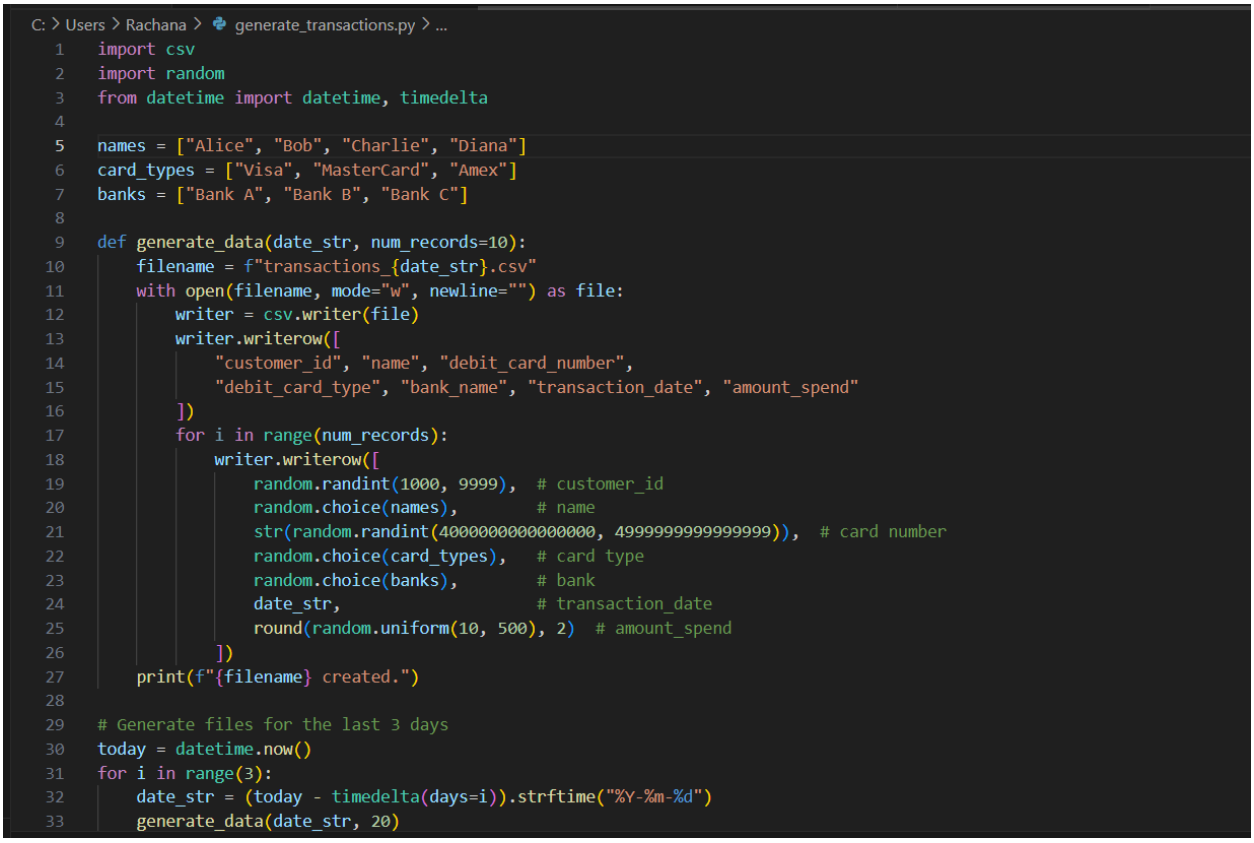
Data pipeline using S3,RDS,Glue

The goal is to aggregate customer debit card purchases on a daily basis and update the aggregated data in a MySQL table hosted on Amazon RDS. I used AWS S3 for data storage, AWS Glue for data processing, and Amazon RDS for data persistence.

Step 1: Generate data and store in S3 using Hive style partition

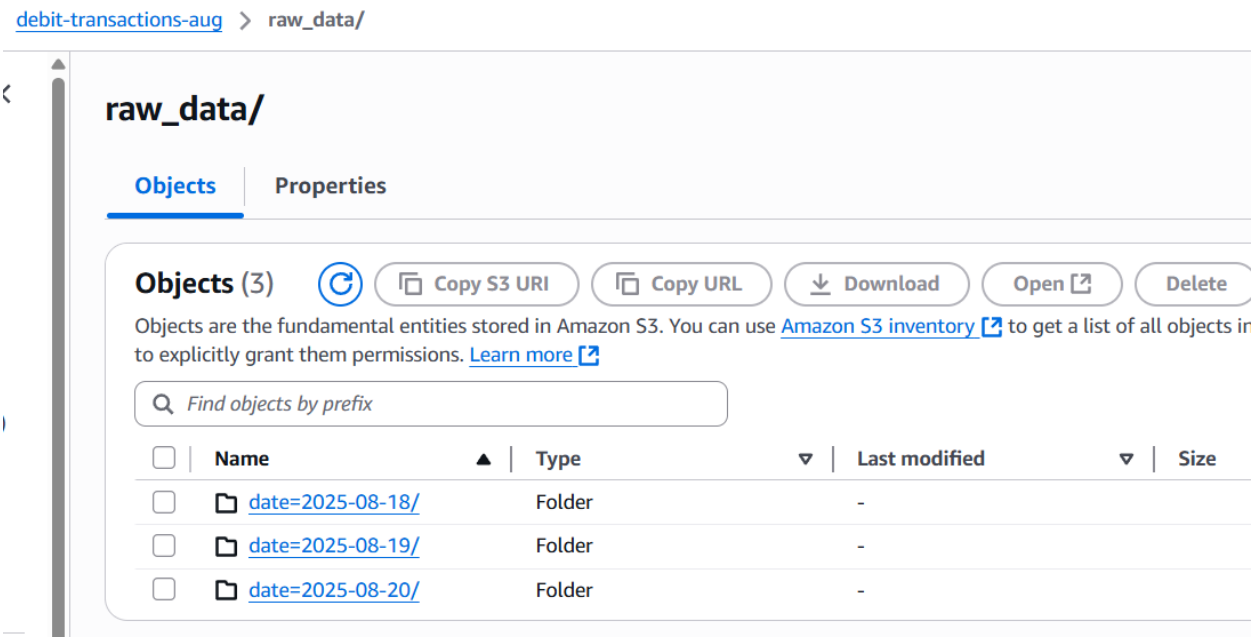
1. Generate mock daily transaction data and store it in CSV files.

2. Upload daily transaction CSV files to an AWS S3 bucket using a Hive-style partition.



C: > Users > Rachana > generate_transactions.py > ...

```python
import csv
import random
from datetime import datetime, timedelta

names = ["Alice", "Bob", "Charlie", "Diana"]
card_types = ["Visa", "MasterCard", "Amex"]
banks = ["Bank A", "Bank B", "Bank C"]

def generate_data(date_str, num_records=10):
    filename = f"transactions_{date_str}.csv"
    with open(filename, mode="w", newline="") as file:
        writer = csv.writer(file)
        writer.writerow([
            "customer_id", "name", "debit_card_number",
            "debit_card_type", "bank_name", "transaction_date", "amount_spend"
        ])
        for i in range(num_records):
            writer.writerow([
                random.randint(1000, 9999),  # customer_id
                random.choice(names),        # name
                str(random.randint(4000000000000000, 4999999999999999)),  # card number
                random.choice(card_types),   # card type
                random.choice(banks),        # bank
                date_str,                    # transaction_date
                round(random.uniform(10, 500), 2)  # amount_spend
            ])
    print(f"{filename} created.")

# Generate files for the last 3 days
today = datetime.now()
for i in range(3):
    date_str = (today - timedelta(days=i)).strftime("%Y-%m-%d")
    generate_data(date_str, 20)
```



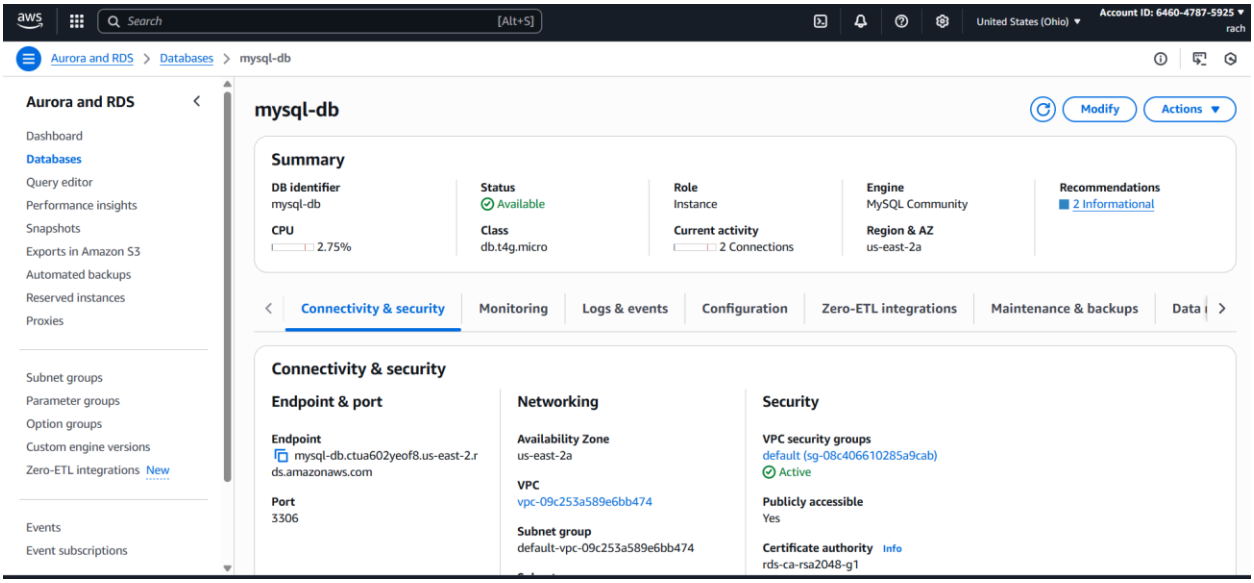
Step 2: Set up a MySQL table in Amazon RDS to store aggregated transaction data.

1. Go to AWS Console → RDS → Create database.
2. Choose:
    - Engine: MySQL
    - Create username/password.
3. Once created, note the endpoint (e.g., mydb.xxxxx.us-east-1.rds.amazonaws.com).
4. Connect via MySQL Workbench and create the table:

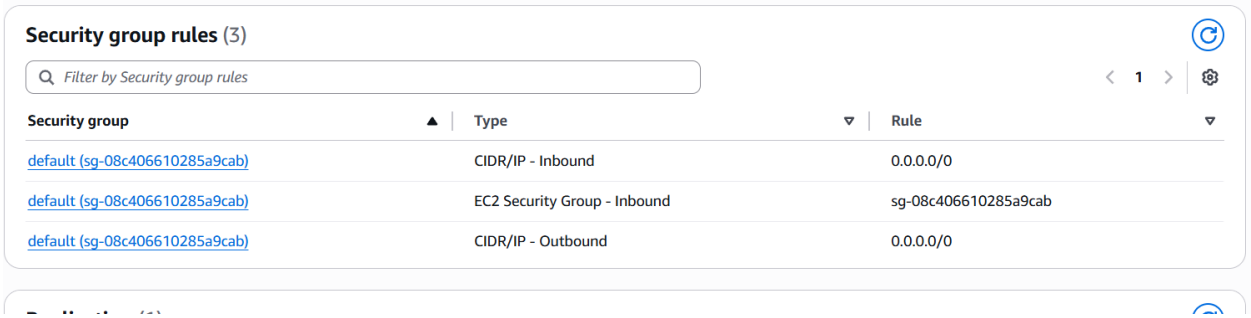
```
CREATE DATABASE transactions_db;

USE transactions_db;


CREATE TABLE aggregated_transactions (
  customer_id INT,
  debit_card_number VARCHAR(20),
  bank_name VARCHAR(50),
  total_amount_spend DECIMAL(10,2),
```

PRIMARY KEY (customer_id, debit_card_number)

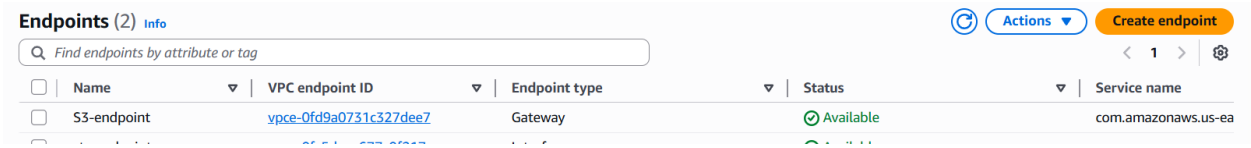
);



5. Make sure security group of rds has port 3306 open for inbound requests

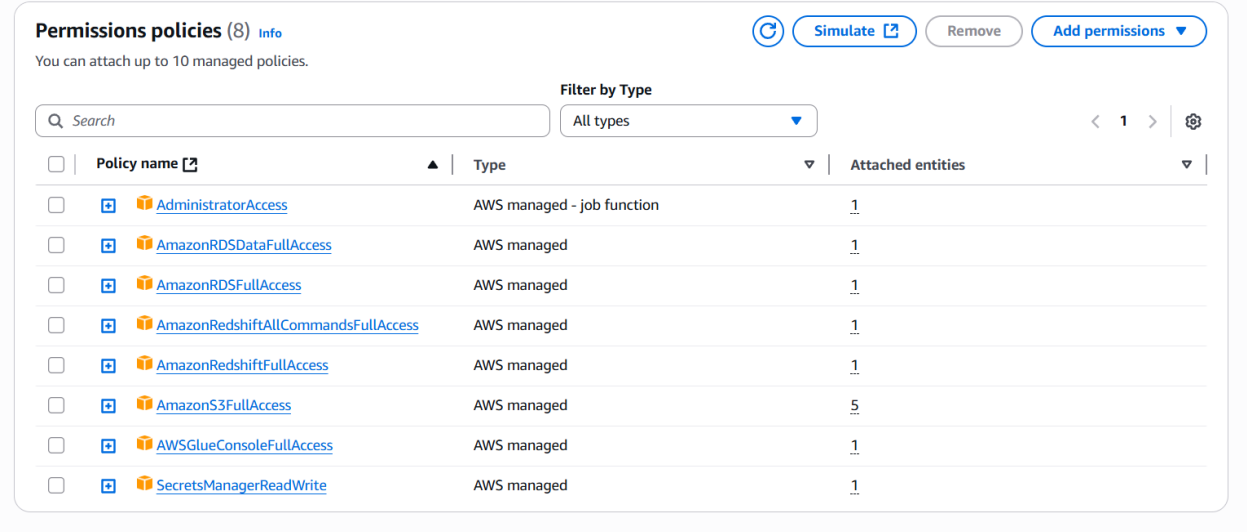


6. Also create an endpoint for S3 in RDS VPC

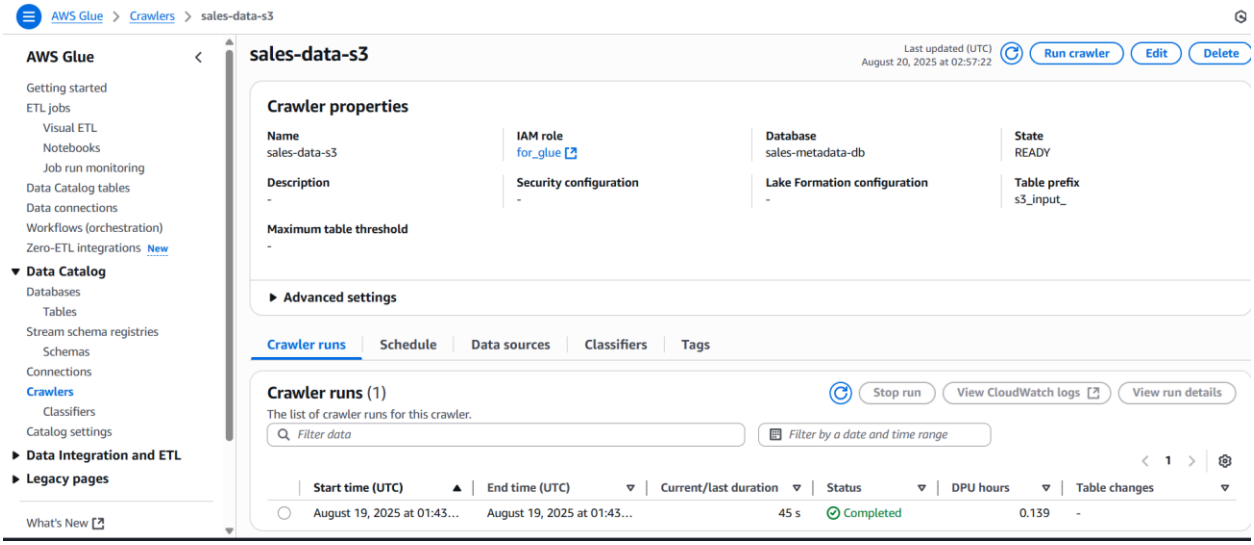


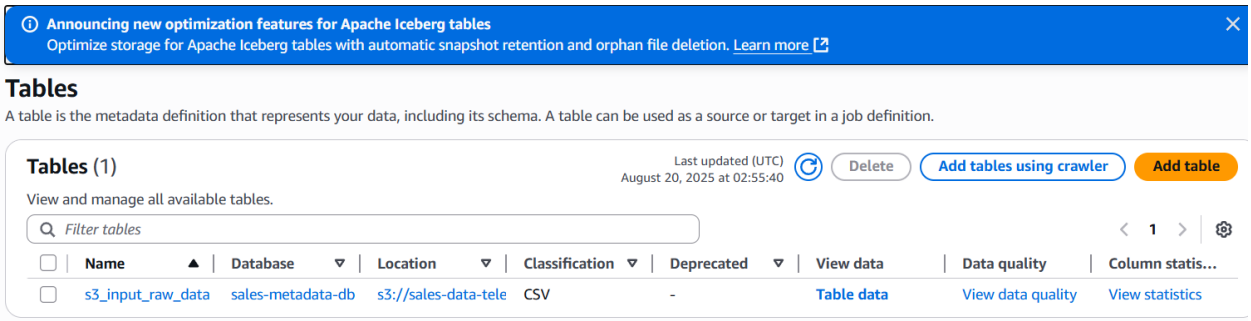
Step 3: Write an AWS Glue job to process daily transactions from S3, aggregate them, and update the RDS MySQL table

1. **Set up AWS Glue**:
    - First make sure glue has all necessary IAM permissions

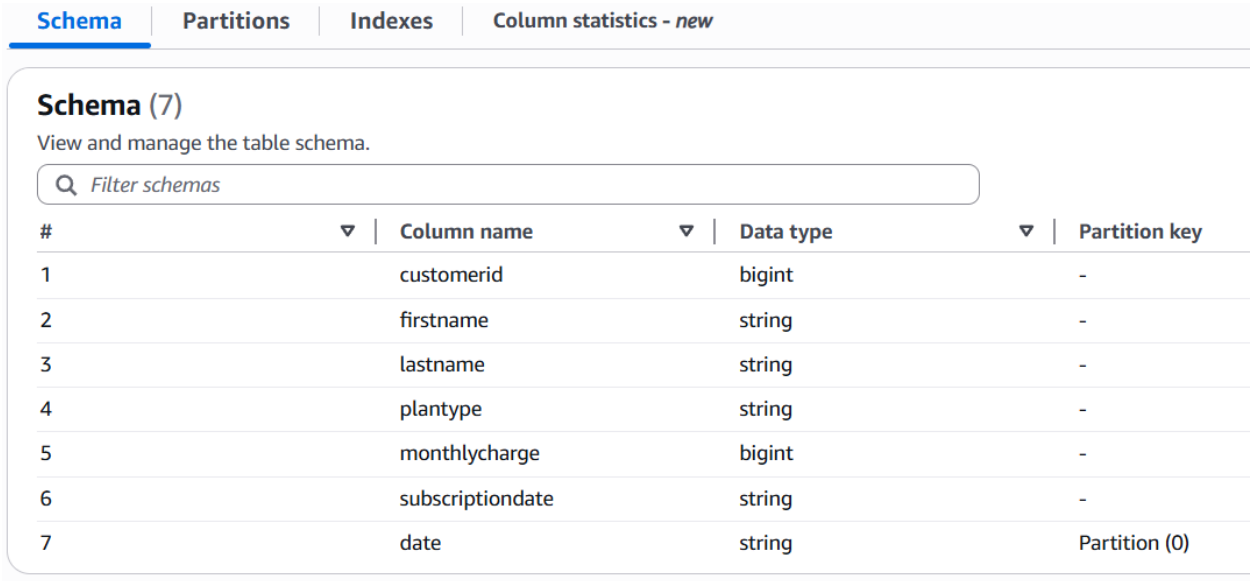


2. Go to Glue → Create Crawler (point to S3 bucket).
3. Let it create a Glue Data Catalog table.





Data Catalog showing schema from glue crawler



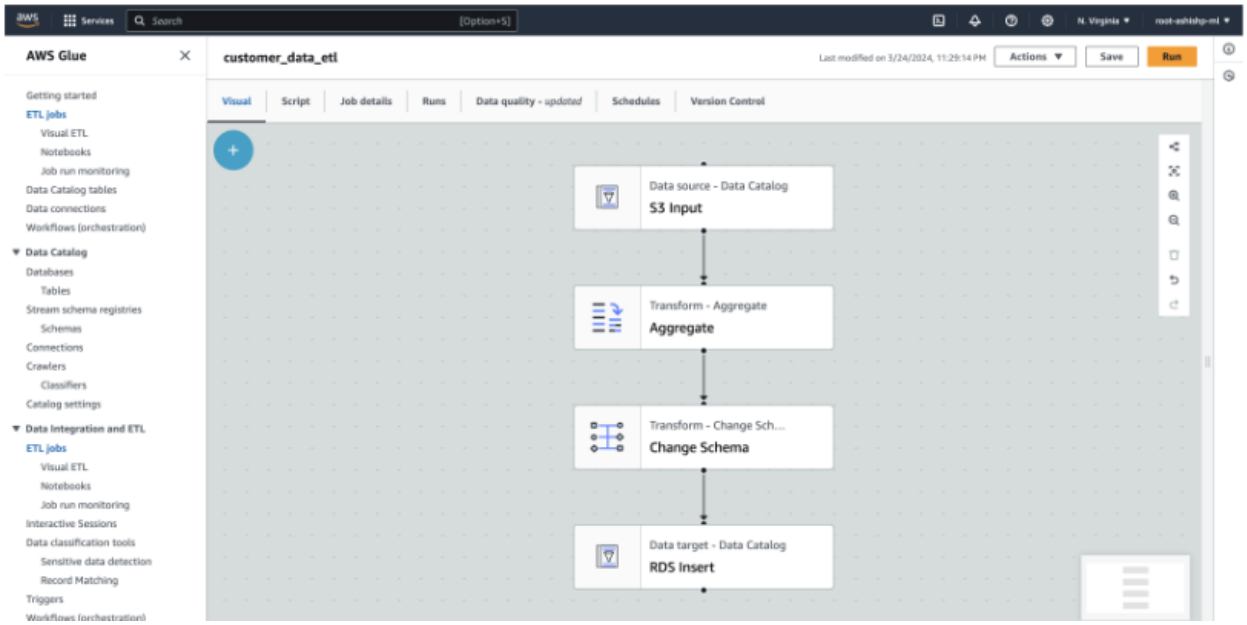
#	Column name	Data type	Partition key
1	customerid	bigint	-
2	firstname	string	-
3	lastname	string	-
4	plantype	string	-
5	monthlycharge	bigint	-
6	subscriptiondate	string	-
7	date	string	Partition (0)

Step 4: Create a Glue Connection to RDS:

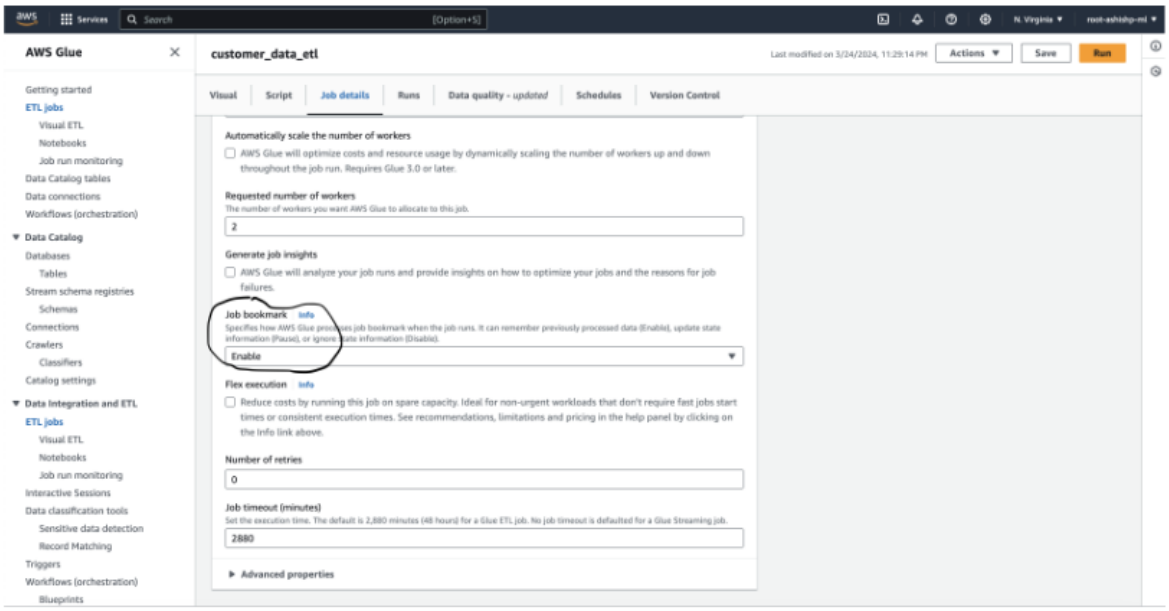
1. Go to Glue → Connections → Add connection.
2. Type: JDBC → MySQL → Enter RDS endpoint, DB name, username, password.
3. Store credentials in **AWS Secrets Manager** for security.

Step 5: Create a Glue Job to aggregate data from S3 and push to RDS

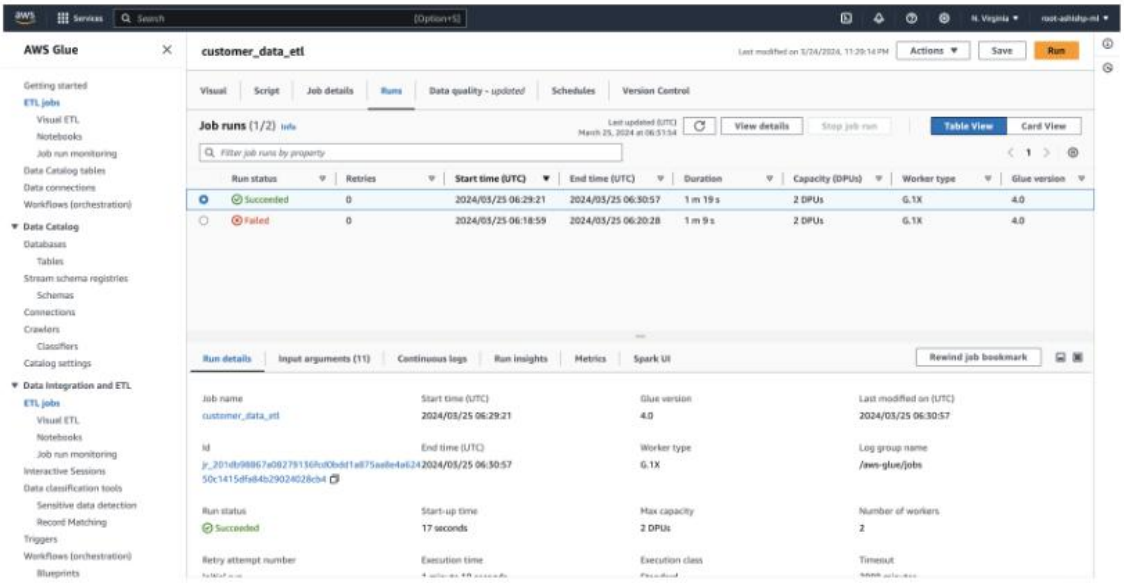
AWS Glue



Incremental Load



Glue Job Run


AWS Glue
customer_data_etl
Actions
Save
Run
Visual
Script
Job details
Runs
Data quality - updated
Schedules
Version Control
Job runs (1/2)
View details
Stop job run
Table View
Card View
Run status
Retries
Start time (UTC)
End time (UTC)
Duration
Capacity (DPUs)
Worker type
Glue version
Succeeded
0
2024/03/25 06:29:21
2024/03/25 06:30:57
1 m 19 s
2 DPUs
G.1X
4.0
Failed
0
2024/03/25 06:18:59
2024/03/25 06:20:28
1 m 9 s
2 DPUs
G.1X
4.0
Run details
Input arguments (11)
Continuous logs
Run insights
Metrics
Spark UI
Rewind job bookmark
Job name
customer_data_etl
Start time (UTC)
2024/03/25 06:29:21
Glue version
4.0
Last modified on (UTC)
2024/03/25 06:30:57
End time (UTC)
2024/03/25 06:30:57
Worker type
G.1X
Log group name
/aws-glue/jobs
Run status
Succeeded
Start-up time
17 seconds
Max capacity
2 DPUs
Number of workers
2