Automate CSV to JSON Transformation Between S3 Buckets Using AWS Glue and Lambda.

Problem Statement :-

Automate the process of converting a CSV file uploaded to an S3 bucket into a JSON format and storing it in another S3 bucket. The process should trigger automatically upon CSV file .

Objective :-

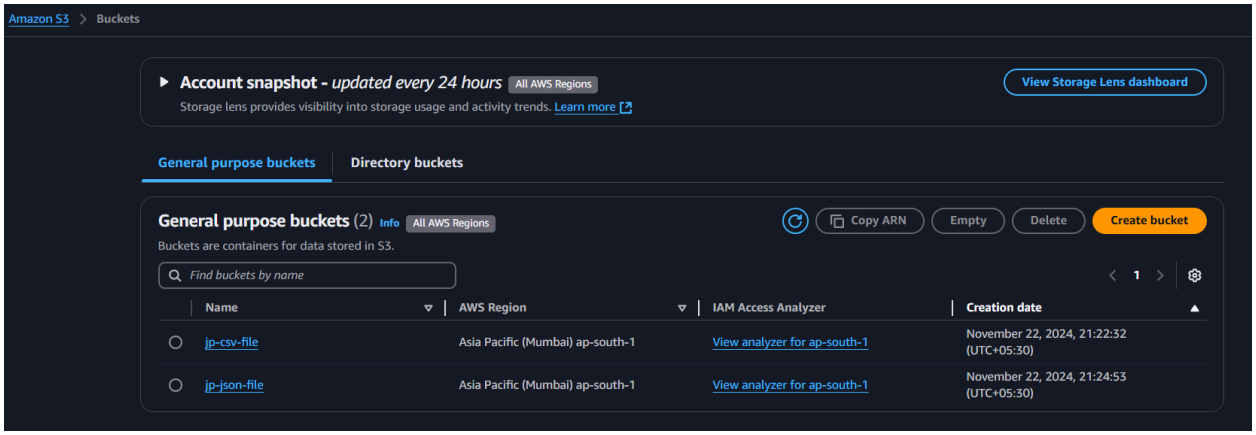
- Set up an AWS Glue job to convert CSV files to JSON format.
- Use AWS Lambda to trigger the process automatically whenever a CSV file is uploaded to the source S3 bucket.
- Ensure all services have the necessary permissions by configuring role

Implementation Steps

Step 1: Create Source and Destination S3 Buckets

1. Create Source S3 Bucket :-

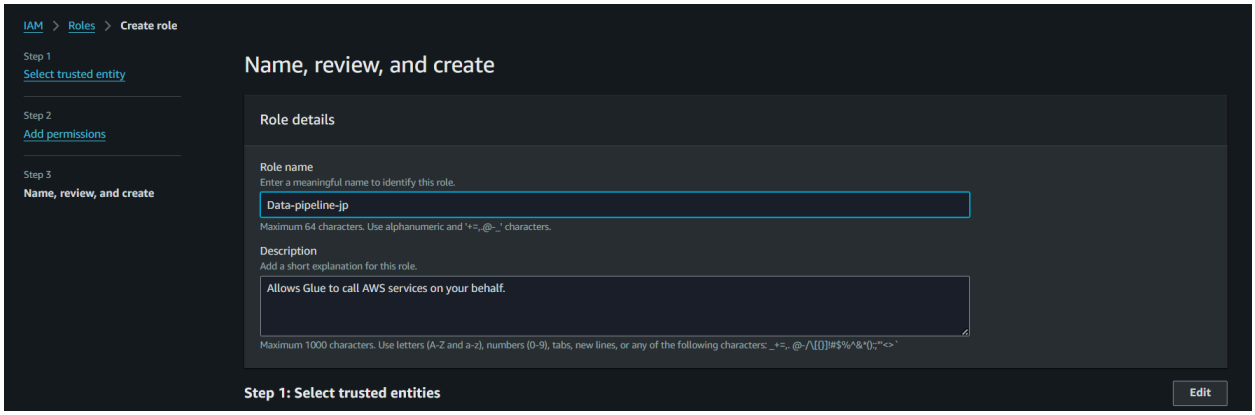
- Go to the **AWS S3 Console**.
- Click **Create Bucket**.
- Name the bucket (e.g., source-csv-bucket), select the region, and configure other settings (e.g., versioning if required)
- Click **Create**.
- Repeat the same steps as above to create another S3 bucket where the JSON files will be stored (e.g., destination-json-bucket).

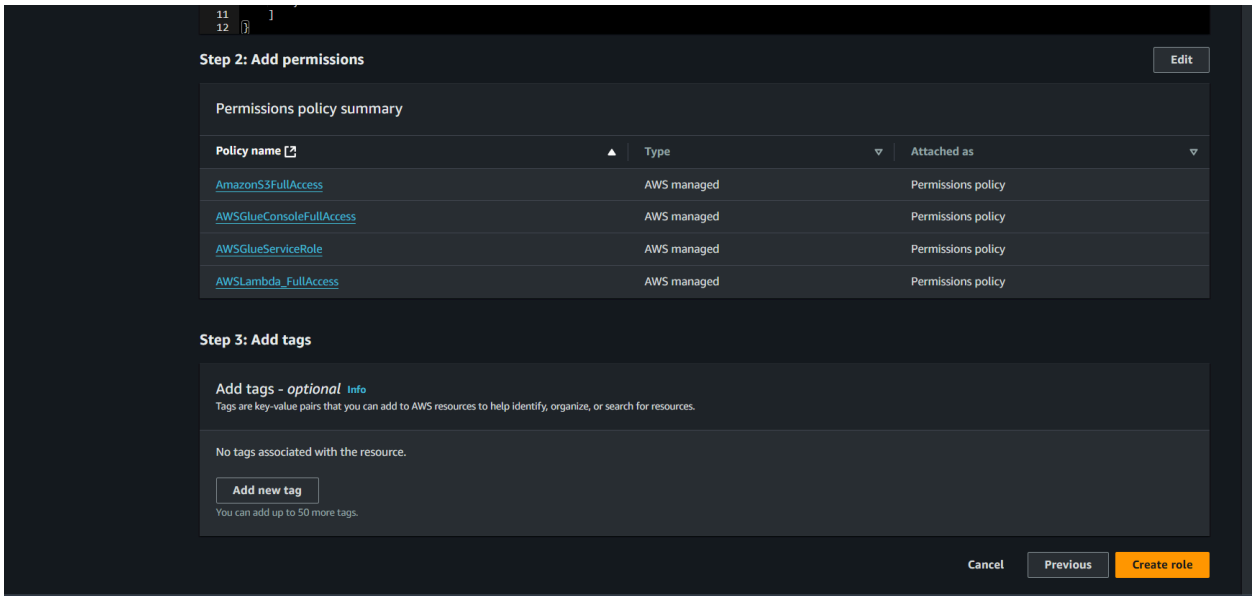


Step 2: Create IAM Role

1.Create IAM Role with Full Access to Lambda, Glue, and S3 :-

- Navigate to the **IAM Console**.
- Click on **Roles** and then **Create Role**.
- Select **AWS Service** as the trusted entity, and then choose **Lambda** as the use case
- Attach the following policies to the role:
  - **AmazonS3FullAccess**
  - **AWSGlueServiceRole**
  - **AWSGlueConsoleFullAccess**
  - **AWSLambda_FullAccess**
- Give the role a name (e.g., Data-pipeline-role) and create it

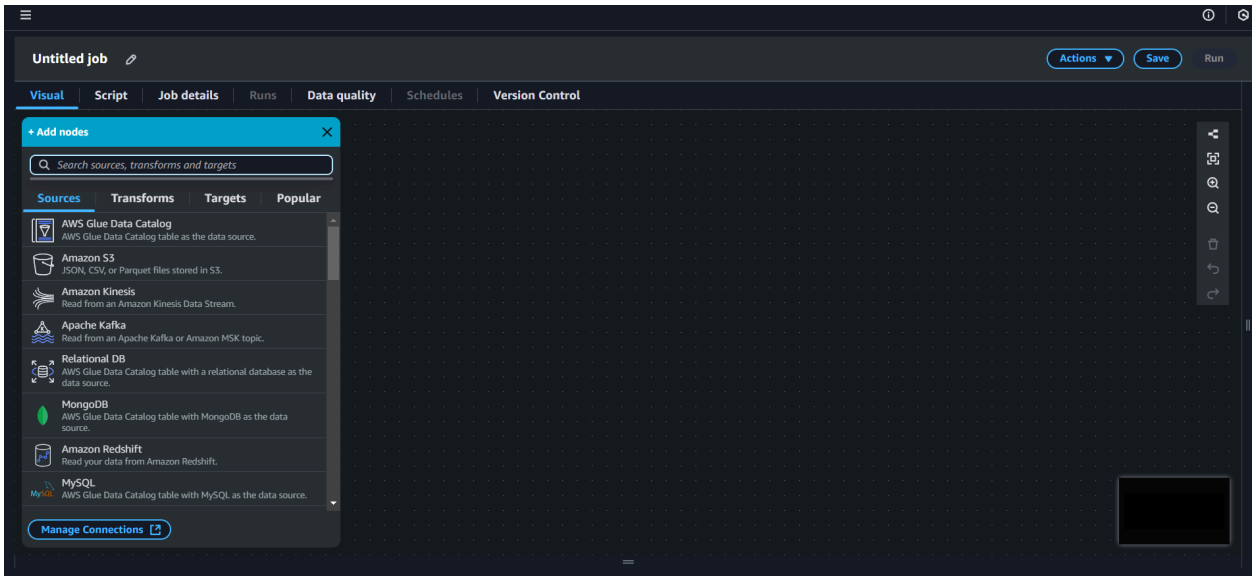




Step 3: Set Up AWS Glue for ETL Job Using Visual Interface

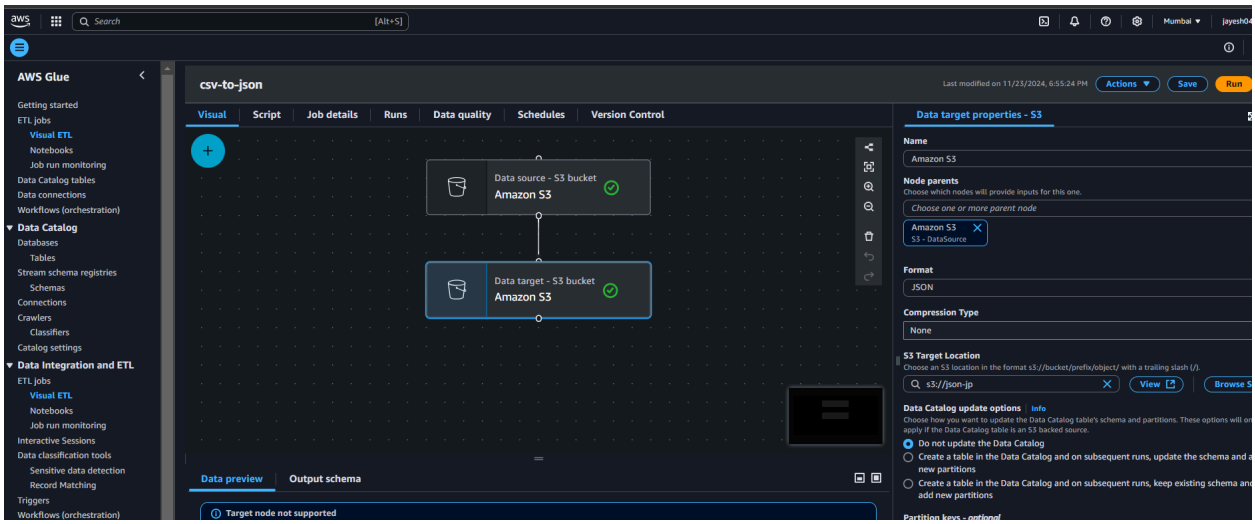
1. Navigate to AWS Glue and Start ETL Job Setup:

- Go to the AWS Glue console from the AWS Management Console.
- On the Glue dashboard, click on ETL Jobs to start setting up a new ETL job.



2. Launch the Visual Interface:

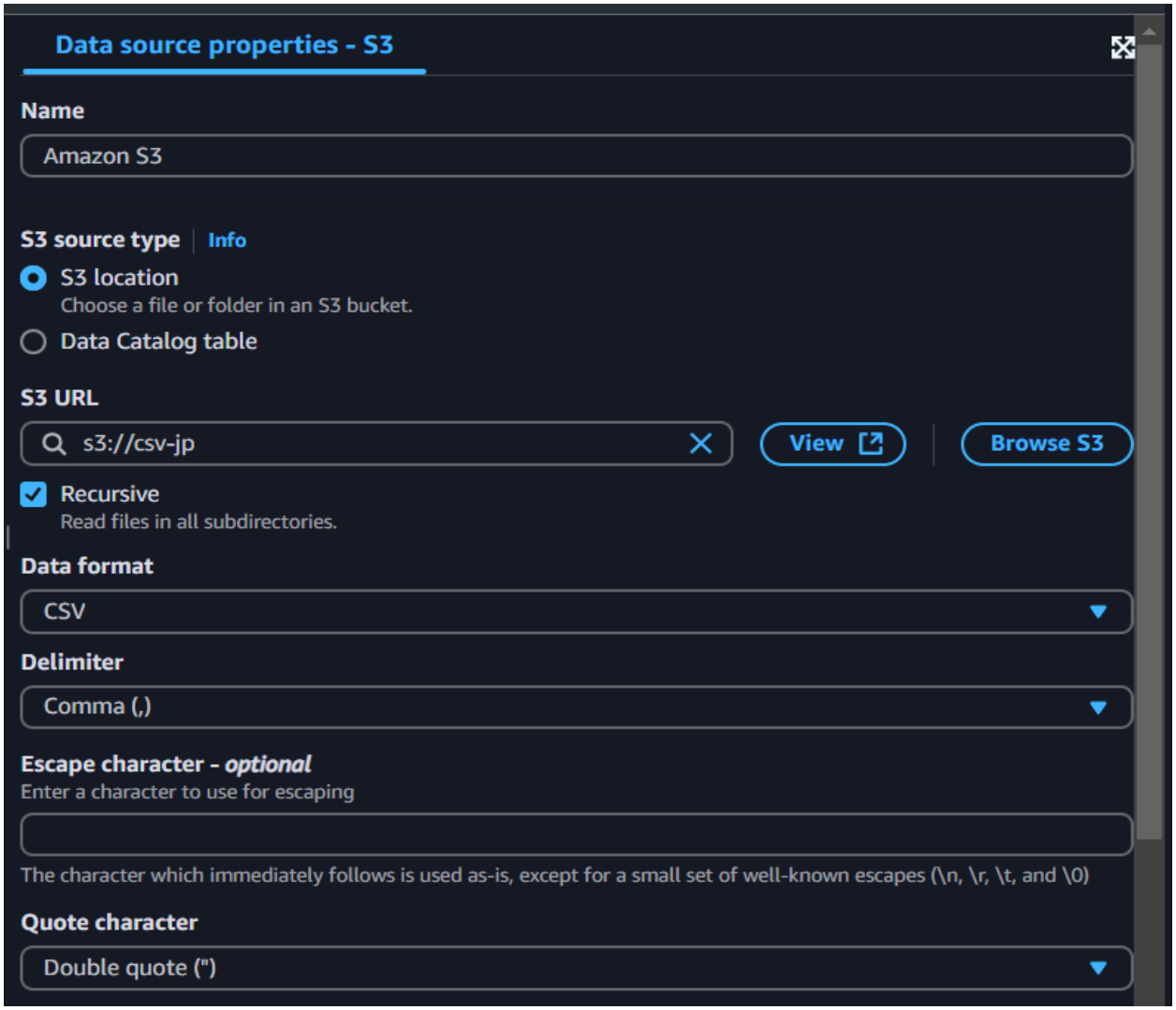
- When the ETL job setup screen appears, select the Visual with source and target option to use the visual editor.



3. Configure the Source (S3 Bucket with CSV Files) :-

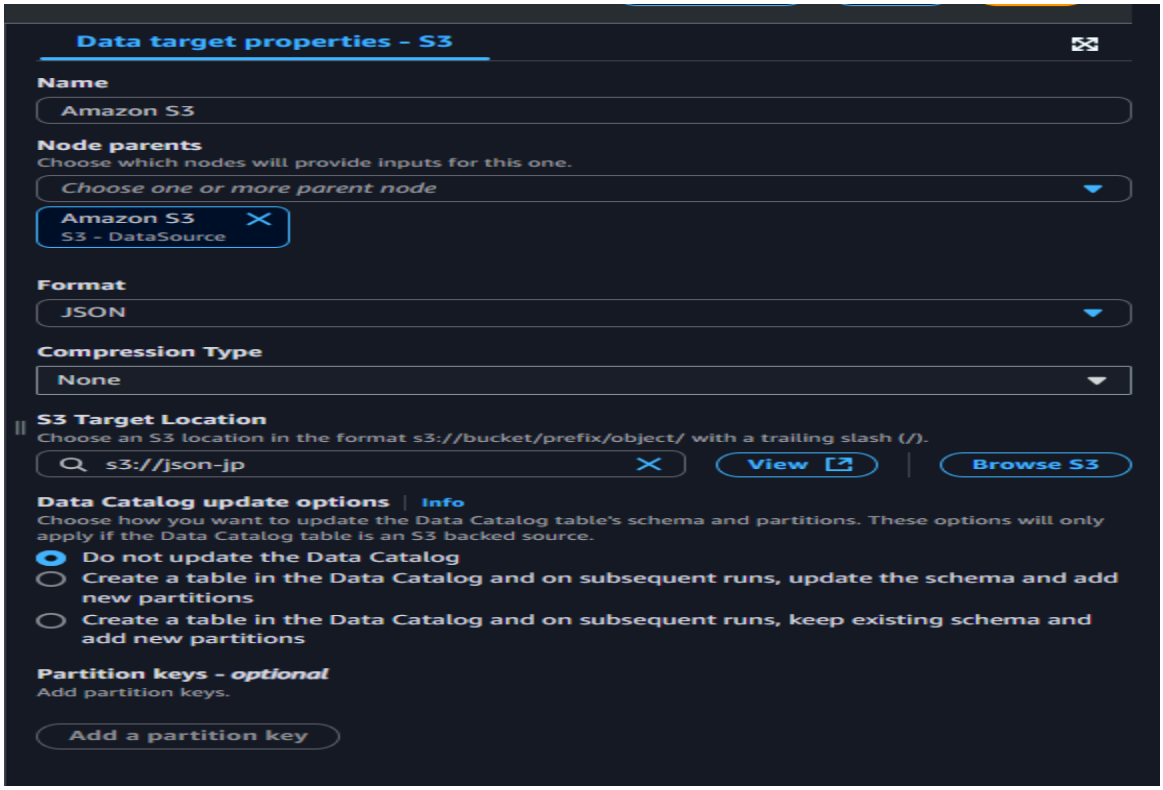
- In the visual interface, click Add Source.
- Choose S3 as the data store.
- Specify the S3 bucket where the CSV files are stored (e.g., source-csv-bucket).

- In the File Format section, select CSV as the format for the source files



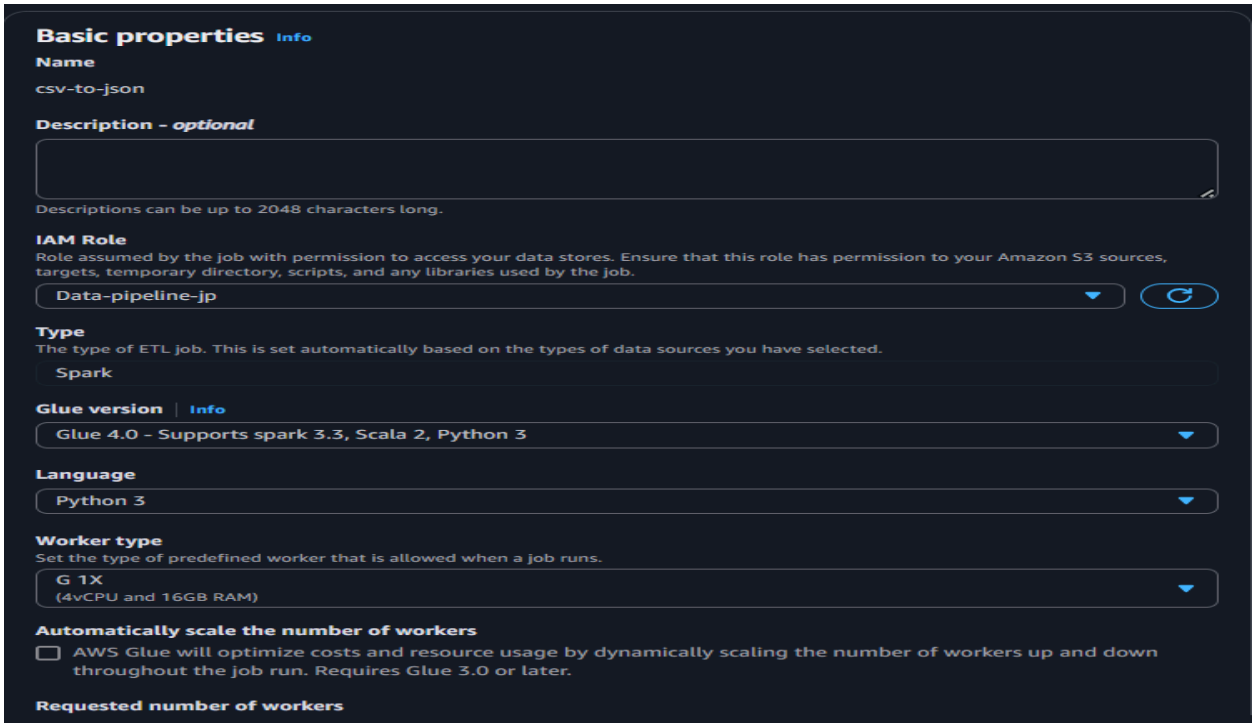
4. Configure the Destination (S3 Bucket for JSON Files):

- Click Add Target to specify the destination for the transformed data.
- Choose S3 as the destination data store.
- Specify the target S3 bucket where the JSON files will be stored (e.g.,destination-json-bucket).
- Set the file format as JSON.



5. Job Details:

- After configuring the source and destination, go to the Job Details section.
- Enter a name for the job (e.g., csv-to-json-etl-job).
- Add the necessary role to allow Glue to access the resources (S3, Lambda).



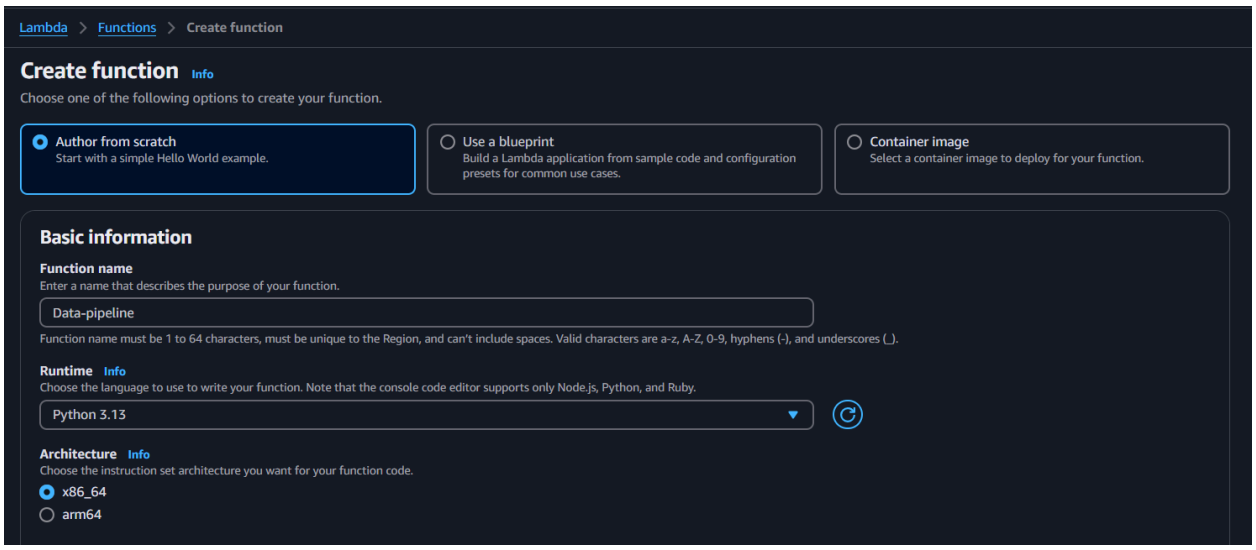
6. Save the Job:

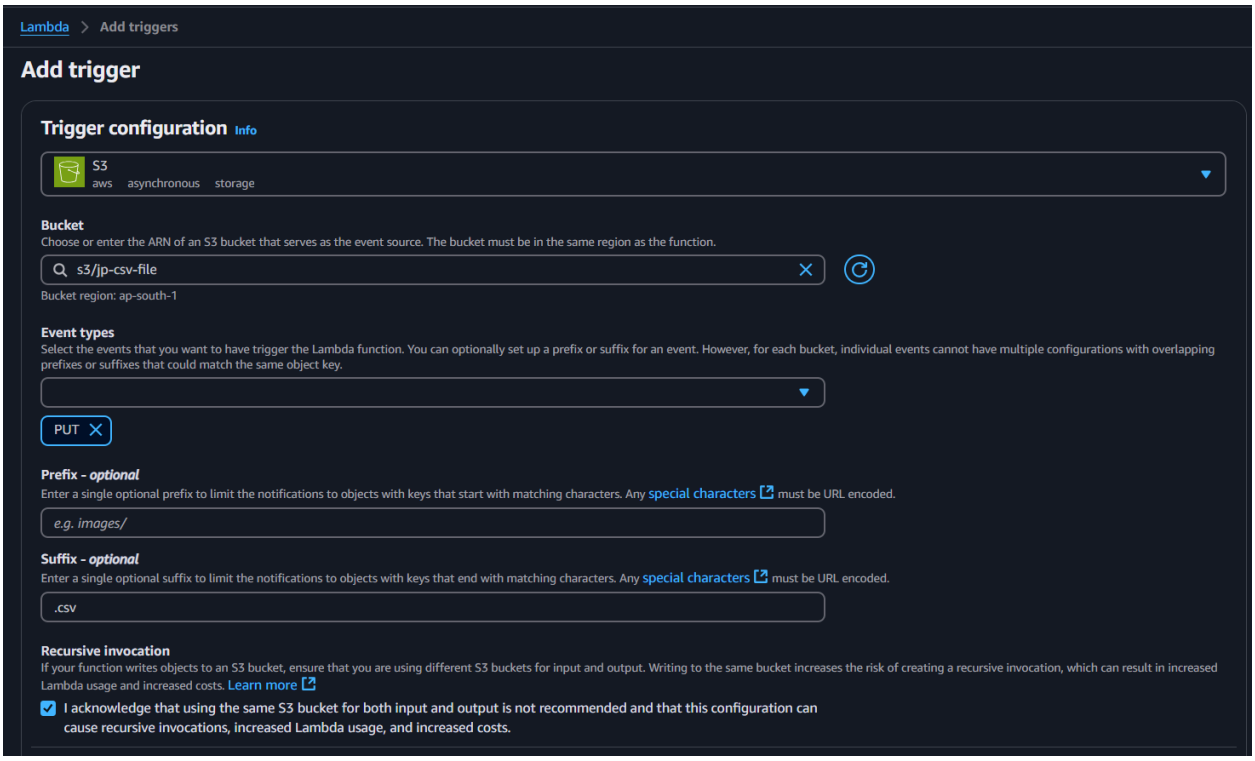
- Click Create to save the job.

Step 4: Set Up AWS Lambda for Automatic Trigger

1. Create Lambda Function:

- Go to the **Lambda Console**
- Click on **Create Function** and choose Author from Scratch.
  - **Name: Give the function a name (e.g., csv-upload-trigger).**
  - **Runtime: Select Python 3.x (or any preferred runtime).**
  - **Permissions: Create a new role with lambda as trusted entity and same permissions as before or you can create role automatically and add permissions later**





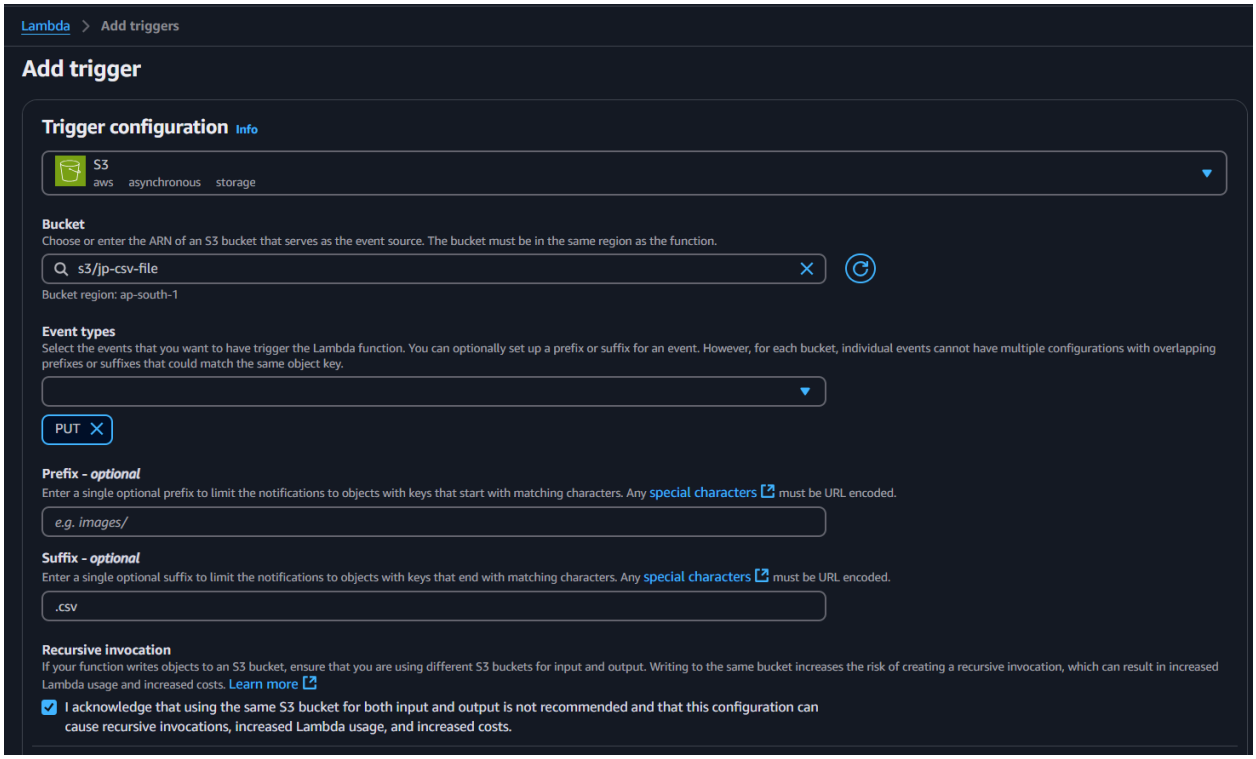
2. Write Lambda Code:

- The Lambda function will trigger the Glue job when a CSV file is uploaded. Use the following Python code as a starting point:

```
import json
import boto3
# Initialize Glue client
glue_client = boto3.client('glue')
def lambda_handler(event, context):
# Start the Glue job
response = glue_client.start_job_run(
JobName='csv-to-json-job') #Replace with your Glue job name
return {
'statusCode': 200,
'body': json.dumps('CSV to JSON Glue job triggered
successfully!')
}
```

3. Set up S3 Trigger for Lambda:

- In the Lambda function console, click on **Add Trigger**.
- Select **S3** as the trigger source.
- Choose the source S3 bucket (source-csv-bucket).
- Set the event type to **PUT** to trigger the Lambda function whenever a CSV file is uploaded.

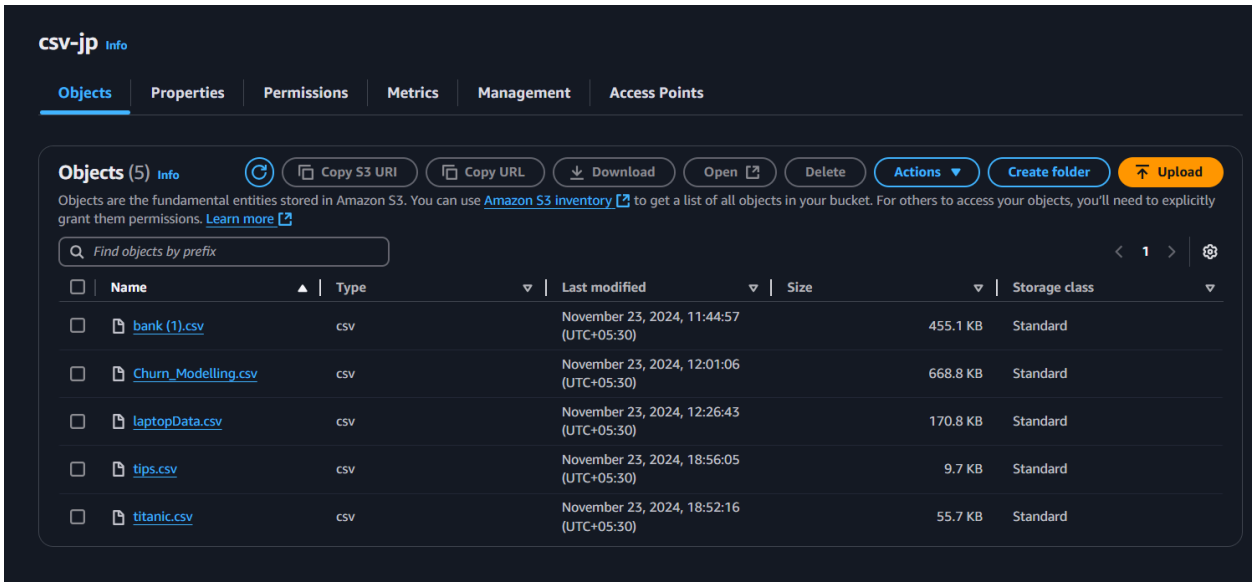


4. Deploy :

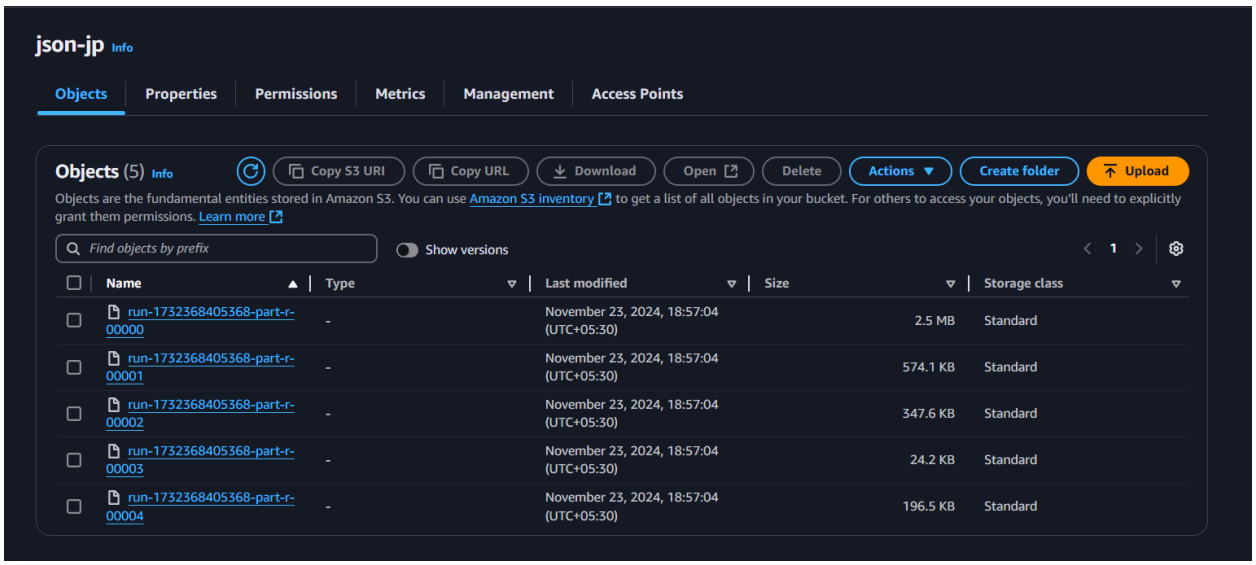
- Deploy the Lambda function.
- Upload a CSV file to the source-csv-bucket and check if the Lambda function triggers the Glue job, converting the CSV to JSON and storing it in the destination-json-bucket.

5.Test :

Input :-



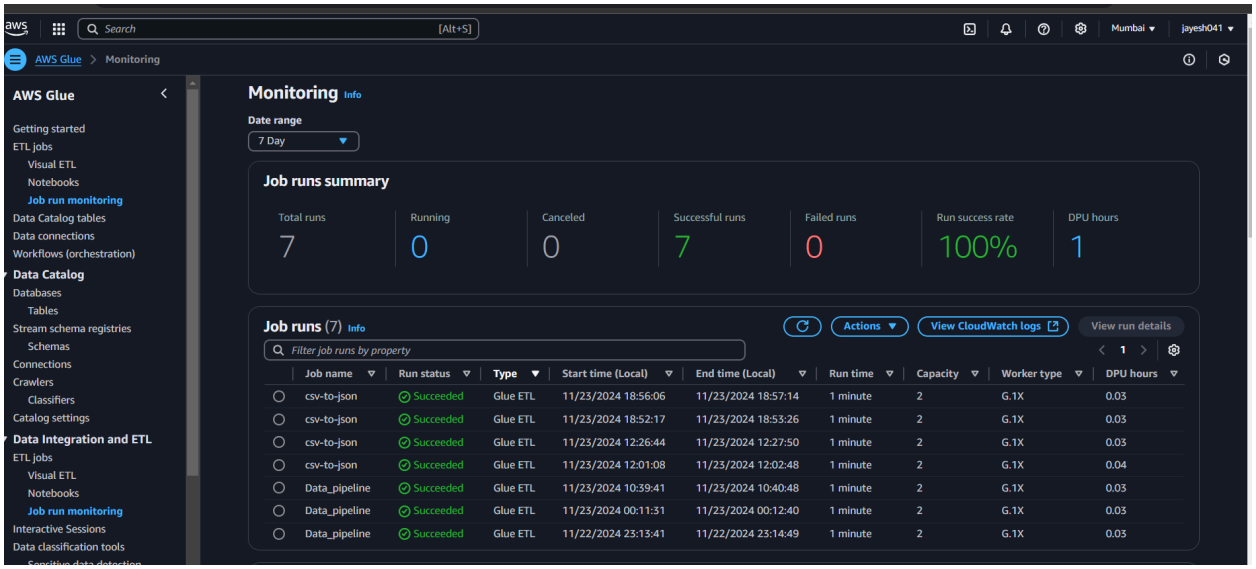
Output :-



Step 5: Monitor and Debug

1. **Monitor with CloudWatch Logs:**
   - In the **CloudWatch Console**, check the logs for both the Lambda function and Glue job.

- Verify if the Glue job was successfully triggered and completed without errors



Conclusion :-

The CSV to JSON transformation and automation setup was successfully completed using AWS Glue for ETL and AWS Lambda for automatic triggers. This setup efficiently converts files between two S3 buckets without manual intervention, saving time and improving data processing workflows.