

# AWS Project-1 (Building the Data Pipeline on AWS)

## Summary

I have created a data pipeline. When any CSV file is stored in S3, through the pipeline, convert it and store it in another S3 bucket in the form of JSON. For this, I have used the AWS glue job to transfer it. To do this job automatically, I have used AWS Lambda with the trigger. So any file comes into the S3 bucket the glue job gets triggered to convert that file into JSON and store it.

## Creating S3 bucket

Give the required configurations as below in the image

The image shows two screenshots of the AWS console 'Create bucket' page. The top screenshot shows the 'General configuration' section where the bucket name 'transformed\_json\_rj' is entered. The bottom screenshot shows the 'Object Ownership' section where 'ACLs enabled' is selected, and the 'Block Public Access settings for this bucket' section where 'Block all public access' is selected.

**General configuration**

AWS Region  
Asia Pacific (Singapore) ap-southeast-1

Bucket name [Info](#)  
transformed\_json\_rj

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - optional  
Only the bucket settings in the following configuration are copied.  
[Choose bucket](#)  
Format: s3://bucket/prefix

**Object Ownership** [Info](#)  
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☒ **ACLs disabled (recommended)**  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☐ **ACLs enabled**  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

**Object Ownership**  
Bucket owner enforced

**Block Public Access settings for this bucket**  
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings.

☒ **ACLs disabled (recommended)**  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☒ **ACLs enabled**  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

**Object Ownership**  
☒ **Bucket owner preferred**  
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

☐ **Object writer**  
The object writer remains the object owner.

**Block Public Access settings for this bucket**  
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings.

aws

Search

[Alt+S]

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

Amazon S3

Buckets

Create bucket

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

Disable

Enable

Tags - optional (0)

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

Add tag

Default encryption [Info](#)

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

Server-side encryption with Amazon S3 managed keys (SSE-S3)

Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

aws

Search

[Alt+S]

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

Amazon S3

Buckets

Create bucket

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

Server-side encryption with Amazon S3 managed keys (SSE-S3)

Server-side encryption with AWS Key Management Service keys (SSE-KMS)

Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

Disable

Enable

Advanced settings

Object Lock

Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. Object Lock works only in versioned buckets. [Learn more](#)

Disable

Enable

Permanently allows objects in this bucket to be locked. Additional Object Lock configuration is required in bucket details after bucket creation to protect objects in this bucket from being deleted or overwritten.

Object Lock works only in versioned buckets. Enabling Object Lock automatically enables Versioning.

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

aws

Search

[Alt+S]

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

Amazon S3

Buckets

Account snapshot - updated every 24 hours [All AWS Regions](#)

[View Storage Lens dashboard](#)

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

General purpose buckets

Directory buckets

General purpose buckets (2) [Info](#) [All AWS Regions](#)

[Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

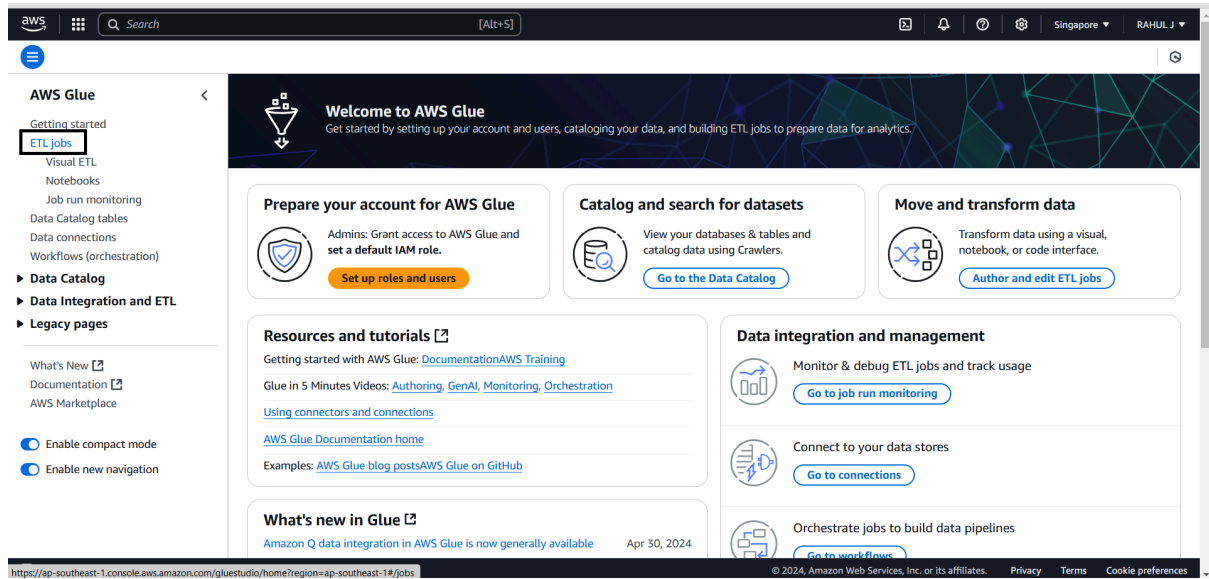
Buckets are containers for data stored in S3.

Find buckets by name

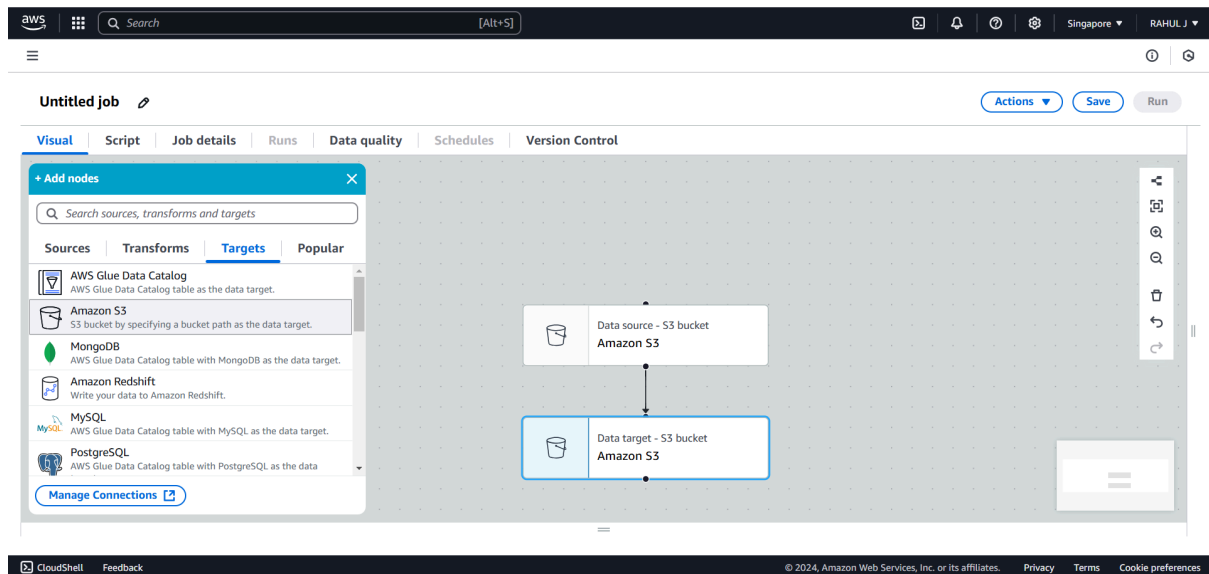
	Name	AWS Region	IAM Access Analyzer	Creation date
<input type="radio"/>	<a href="#">inputcsv-rj</a>	Asia Pacific (Singapore) ap-southeast-1	<a href="#">View analyzer for ap-southeast-1</a>	November 23, 2024, 11:05:18 (UTC+05:30)
<input type="radio"/>	<a href="#">transformed-json-rj</a>	Asia Pacific (Singapore) ap-southeast-1	<a href="#">View analyzer for ap-southeast-1</a>	November 23, 2024, 11:13:51 (UTC+05:30)

# Creating AWS Glue Data Pipeline

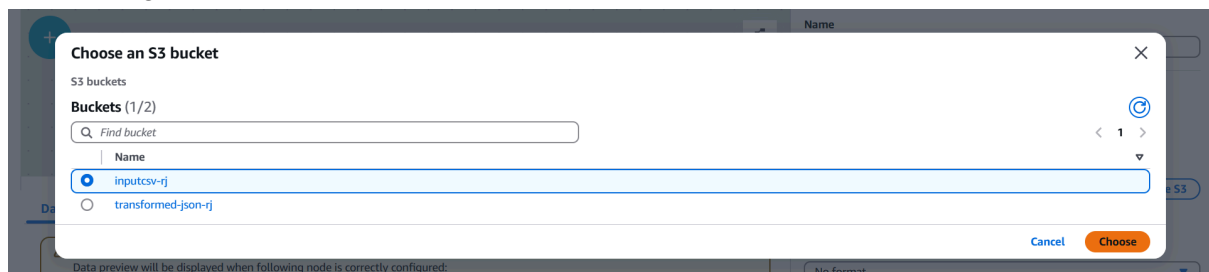
In the search bar search for **AWS Glue** and on the side search for **ETL Jobs**



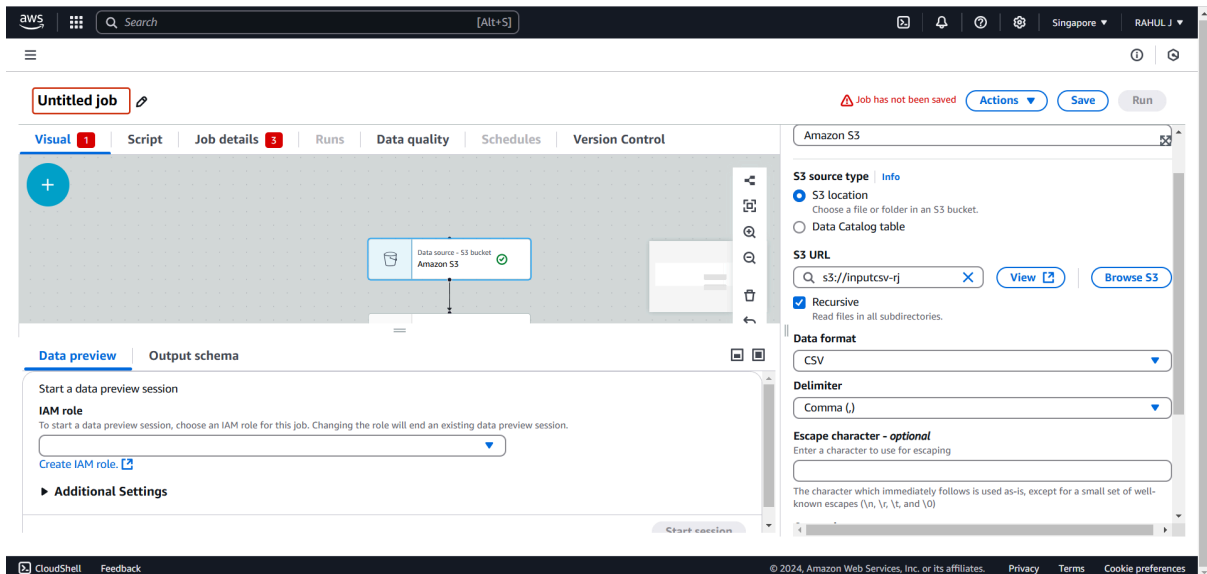
In that click on **visual ETL**, then select the source and the target as the **S3**



Then configure those buckets, first choose the bucket for input one

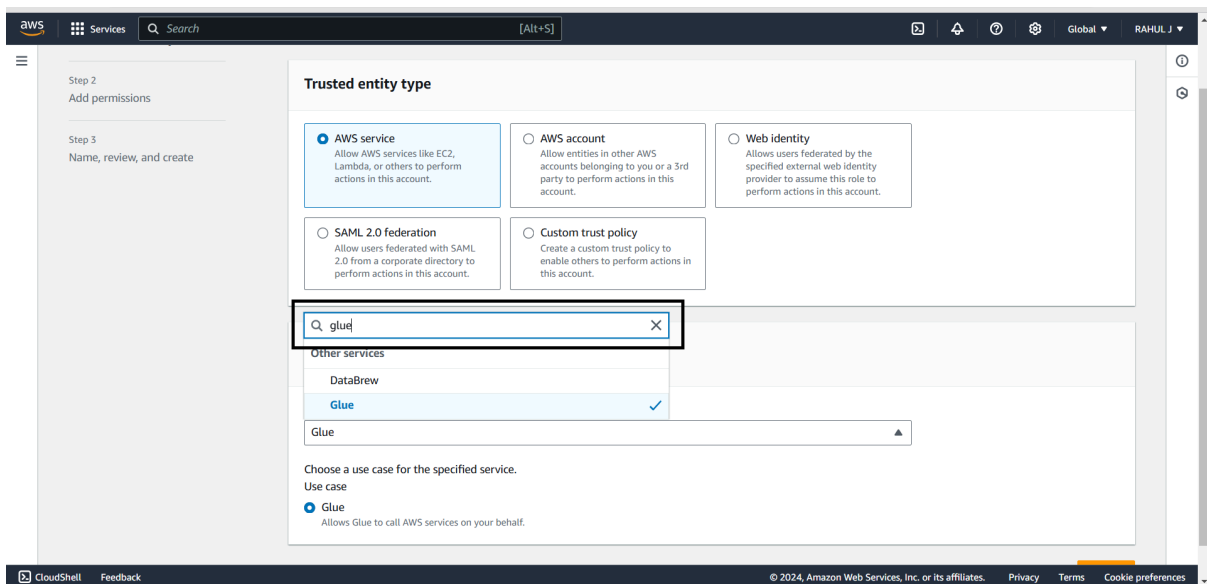


Add the **configuration** for the bucket and click on **save**



## Create an IAM Role

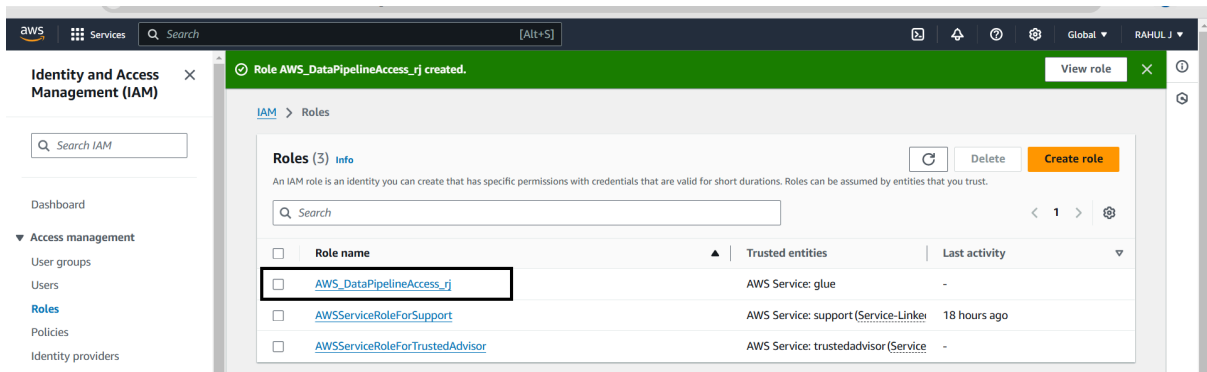
Go to the **IAM service**, in that go to Create Role and select the **USE Case** as **GLUE**



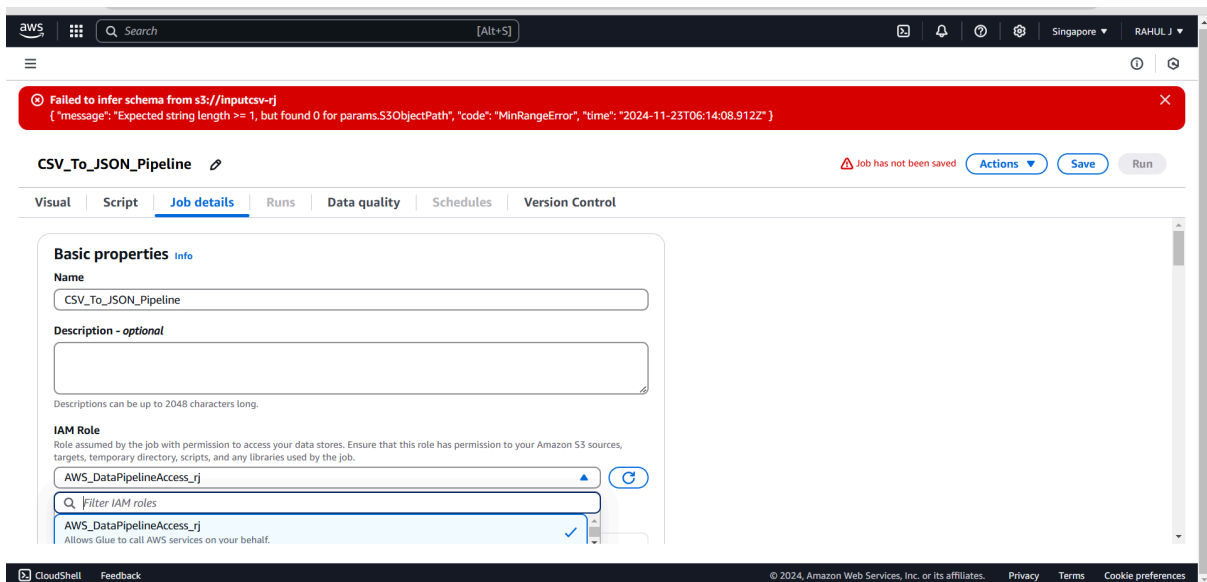
Then click **Next** now choose the policies for the role below in the image

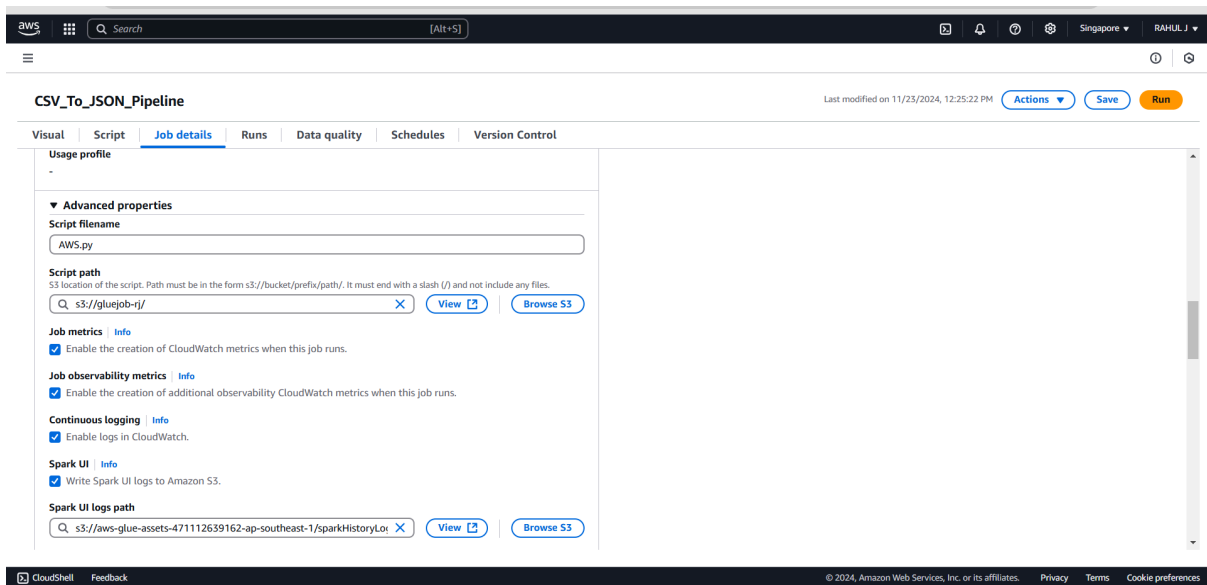
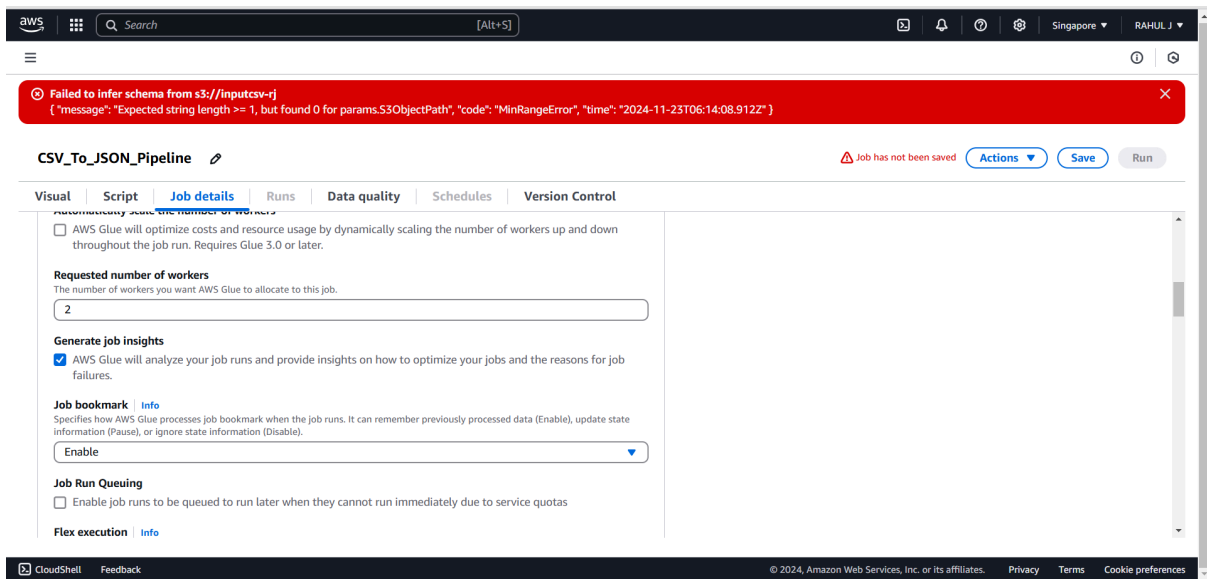
Policy name	Type	Attached as
<a href="#">AmazonS3FullAccess</a>	AWS managed	Permissions policy
<a href="#">AWSGlueConsoleFullAccess</a>	AWS managed	Permissions policy
<a href="#">AWSGlueServiceRole</a>	AWS managed	Permissions policy
<a href="#">AWSLambda_FullAccess</a>	AWS managed	Permissions policy

Then Click next to give the proper name for the role and click on **Create the role**.



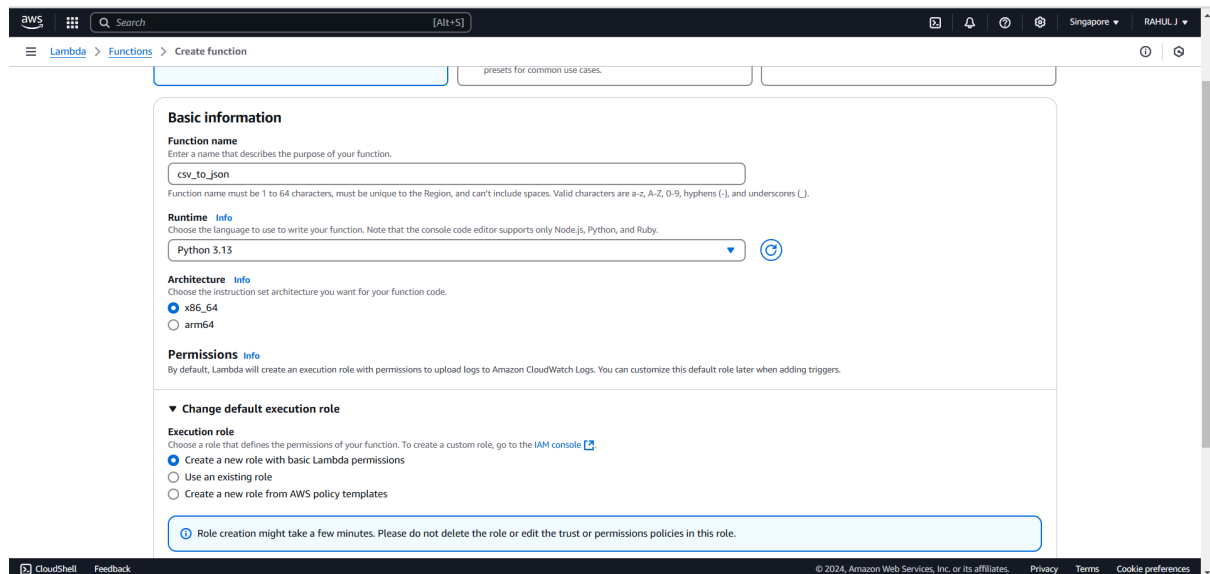
## Configuring the Visual ETL





## Create a Lambda Function

Go to **Lambda service** click on Create **Lambda Function** add the basic configuration to it and click on **Create function**



**Basic information**

**Function name**  
Enter a name that describes the purpose of your function.  
csv\_to\_json  
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (\_).

**Runtime** [Info](#)  
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.  
Python 3.13

**Architecture** [Info](#)  
Choose the instruction set architecture you want for your function code.  
☒ x86\_64  
☐ arm64

**Permissions** [Info](#)  
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

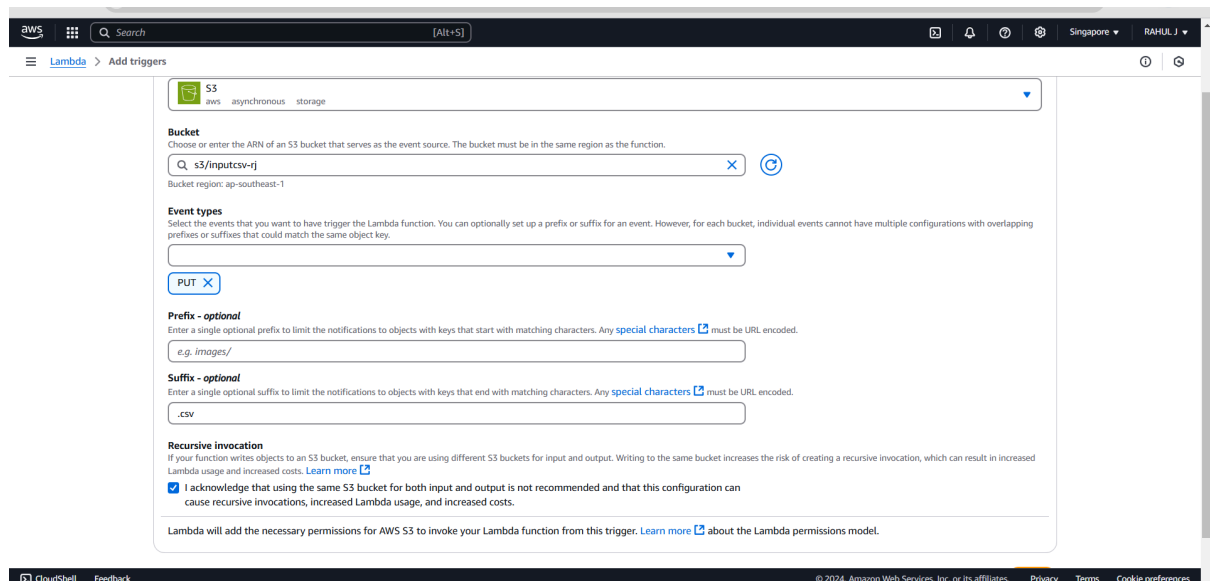
▼ **Change default execution role**

**Execution role**  
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

- ☒ Create a new role with basic Lambda permissions
- ☐ Use an existing role
- ☐ Create a new role from AWS policy templates

ⓘ Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Now click on **Add Trigger** and add the required configuration here we are going to trigger the lambda function when any file is uploading in the **S3 bucket** so we need to choose Event types as **PUT**.



**S3**  
aws asynchronous storage

**Bucket**  
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.  
s3/inputs-csv-rj  
Bucket region: ap-southeast-1

**Event types**  
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.  
PUT

**Prefix - optional**  
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any special characters must be URL encoded.  
e.g. images/

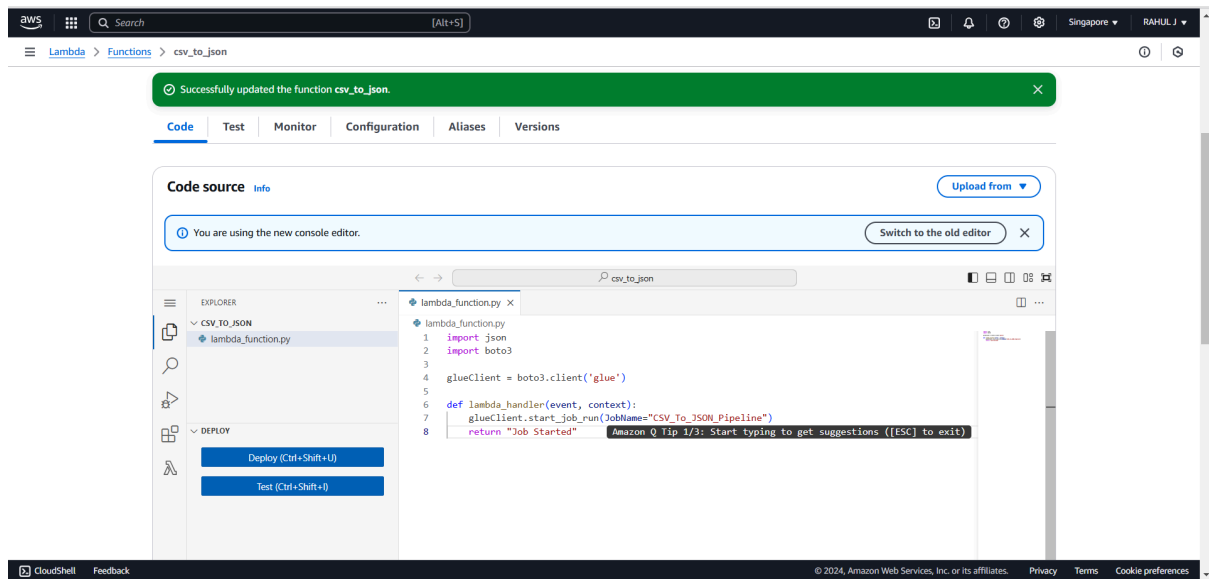
**Suffix - optional**  
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any special characters must be URL encoded.  
.csv

**Recursive invocation**  
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)

☒ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

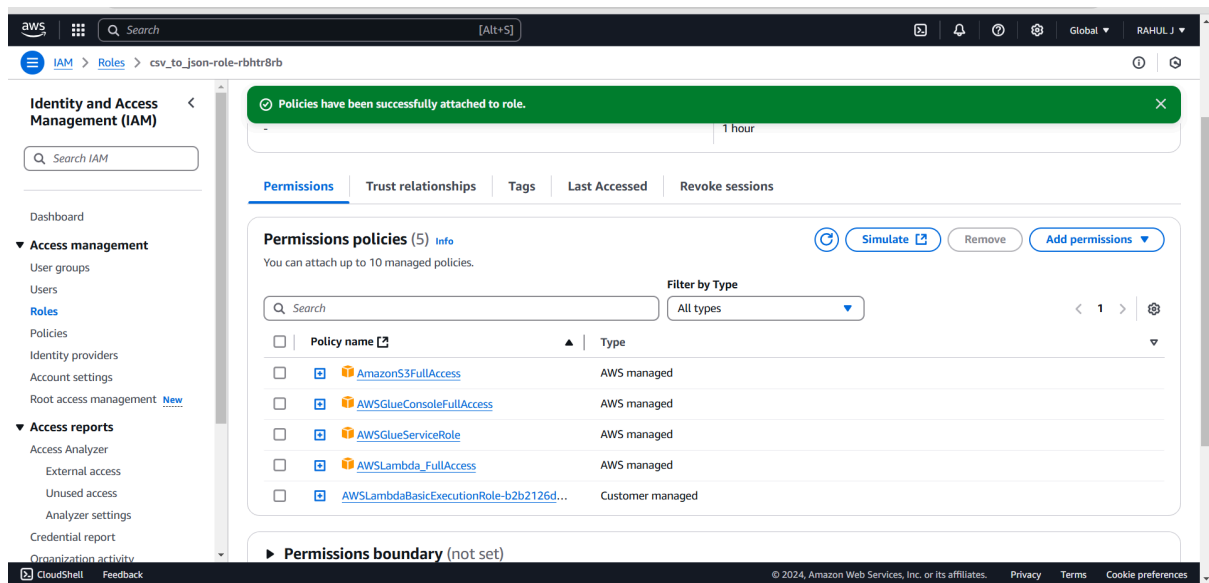
Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Now, we need to add code to trigger the execution of the ETL Jobs after adding the code we need to **deploy the code**



## Attach policies to the IAM Role

We must attach the policies to the **IAM role** to give **access**.



## Checking our services

Uploading the CSV files in the input S3 and checking for the output in the target bucket



Now check for the output in the destined bucket.

To view the file we can download it and see it

```

1 Users > rahu2 > Downloads > C:\run-Processed-Join\input-node1732441052718-5-part-00001
2 ["UID": "16", "iso2": "AS", "iso3": "ASM", "code3": "16", "FIPS": "66-0", "Admin2": "", "Province_State": "American Samoa", "Country_Region": "US", "Lat": "-14.276999999999999", "Long":
3 ["UID": "316", "iso2": "GU", "iso3": "GUM", "code3": "316", "FIPS": "66-0", "Admin2": "", "Province_State": "Guam", "Country_Region": "US", "Lat": "13.44443", "Long": "144.7937", "Combine
4 ["UID": "580", "iso2": "MP", "iso3": "MPM", "code3": "580", "FIPS": "69-0", "Admin2": "", "Province_State": "Northern Mariana Islands", "Country_Region": "US", "Lat": "15.0979", "Long":
5 ["UID": "63072001", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72001-0", "Admin2": "Adjuntas", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.1801170000
6 ["UID": "63072003", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72003-0", "Admin2": "Aguada", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.36025", "Lon
7 ["UID": "63072005", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72005-0", "Admin2": "Aguadilla", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.459681"
8 ["UID": "63072007", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72007-0", "Admin2": "Aguas Buenas", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.251619
9 ["UID": "63072009", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72009-0", "Admin2": "Albionito", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.136361", "L
10 ["UID": "63072011", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72011-0", "Admin2": "Adjuntas", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.1801170000
11 ["UID": "63072013", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72013-0", "Admin2": "Arecibo", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.46631"
12 ["UID": "63072015", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72015-0", "Admin2": "Arroyo", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "17.998457000000
13 ["UID": "63072017", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72017-0", "Admin2": "Barceloneta", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.4455329
14 ["UID": "63072019", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72019-0", "Admin2": "Barraquetas", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.201592
15 ["UID": "63072021", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72021-0", "Admin2": "Bayamon", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.34946", "Lon
16 ["UID": "63072023", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72023-0", "Admin2": "Cabo Rojo", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.040993"
17 ["UID": "63072025", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72025-0", "Admin2": "Caguas", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.21615", "Lon
18 ["UID": "63072027", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72027-0", "Admin2": "Camuy", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.418578", "Lon
19 ["UID": "63072029", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72029-0", "Admin2": "Canovanas", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.328802"
20 ["UID": "63072031", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72031-0", "Admin2": "Carolina", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.374968", "L
21 ["UID": "63072033", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72033-0", "Admin2": "Catano", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.437269", "Lon
22 ["UID": "63072035", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72035-0", "Admin2": "Cayey", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.102851", "Long
23 ["UID": "63072037", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72037-0", "Admin2": "Cayey", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.102851", "Long
24 ["UID": "63072039", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72039-0", "Admin2": "Ciales", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.28885", "Lon
25 ["UID": "63072041", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72041-0", "Admin2": "Cidra", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.173559", "Lon
26 ["UID": "63072043", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72043-0", "Admin2": "Coamo", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.097597", "Long
27 ["UID": "63072045", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72045-0", "Admin2": "Comerio", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.224687", "Lo
28 ["UID": "63072047", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72047-0", "Admin2": "Corozal", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.304264", "Lon
29 ["UID": "63072049", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72049-0", "Admin2": "Culebra", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.31586", "Lon
30 ["UID": "63072051", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72051-0", "Admin2": "Dorado", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.436115", "Lon
31 ["UID": "63072053", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72053-0", "Admin2": "Fajardo", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.318373", "Lo
32 ["UID": "63072054", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72054-0", "Admin2": "Florida", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.373715", "Lo
33 ["UID": "63072055", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72055-0", "Admin2": "Guánica", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "17.982429", "Lo
34 ["UID": "63072057", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72057-0", "Admin2": "Guayama", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.007516", "Lon
35 ["UID": "63072059", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72059-0", "Admin2": "Guayanilla", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.09043"
36 ["UID": "63072061", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72061-0", "Admin2": "Guaynabo", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.3451140000
37 ["UID": "63072063", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72063-0", "Admin2": "Gurabo", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.264641", "Lon
38 ["UID": "63072065", "iso2": "PR", "iso3": "PRI", "code3": "630", "FIPS": "72065-0", "Admin2": "Hatillo", "Province_State": "Puerto Rico", "Country_Region": "US", "Lat": "18.419207", "Lo

```