Zubair

Software Engineer

YouTube Data Analysis – End To End Data Engineering Project

engrzubairkhatti@gmail.com

LinkedIn

GitHub

Table of Contents

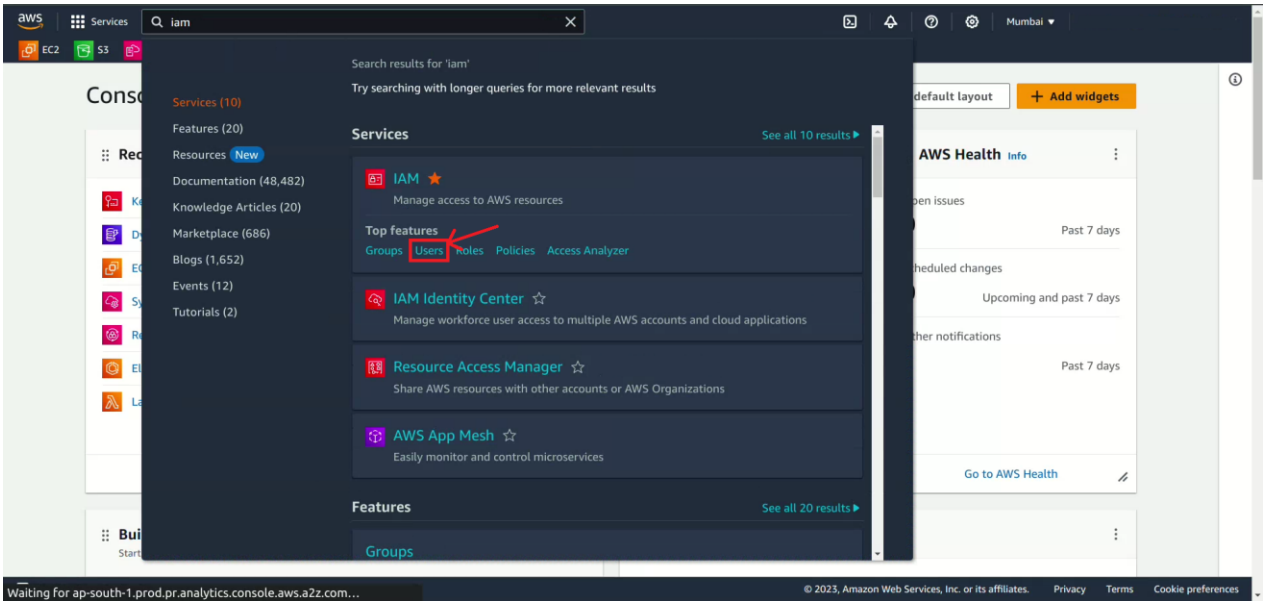
STEP 01

CREATE IAM USER

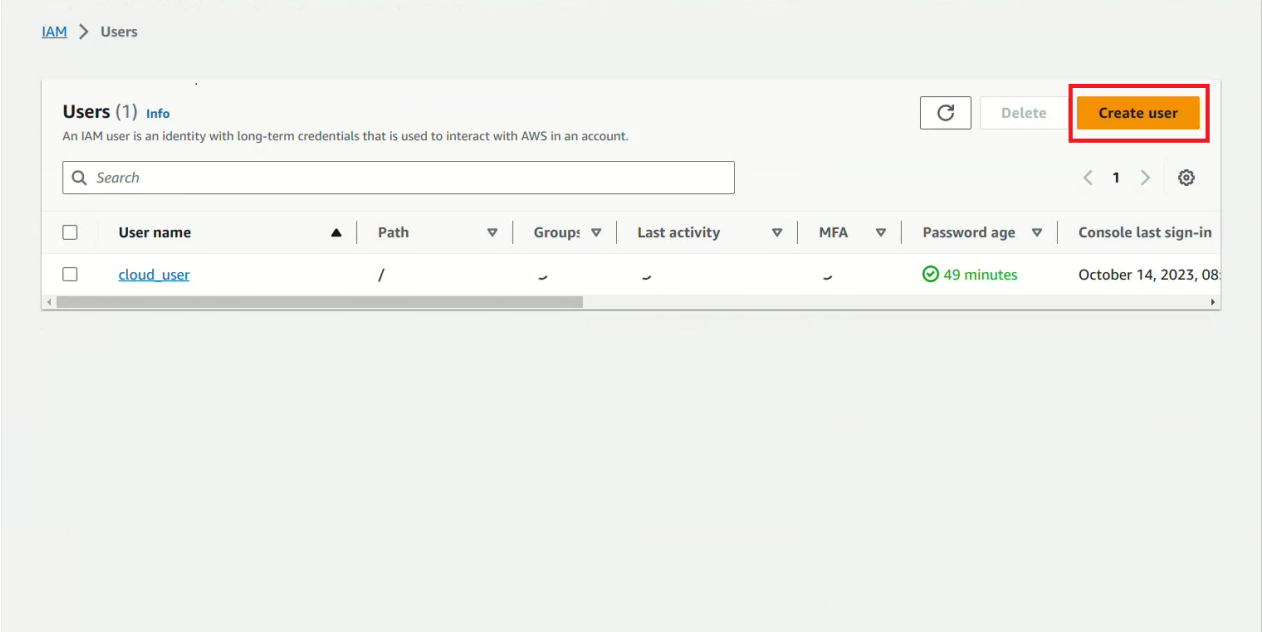
Steps for project

1) Create IAM User

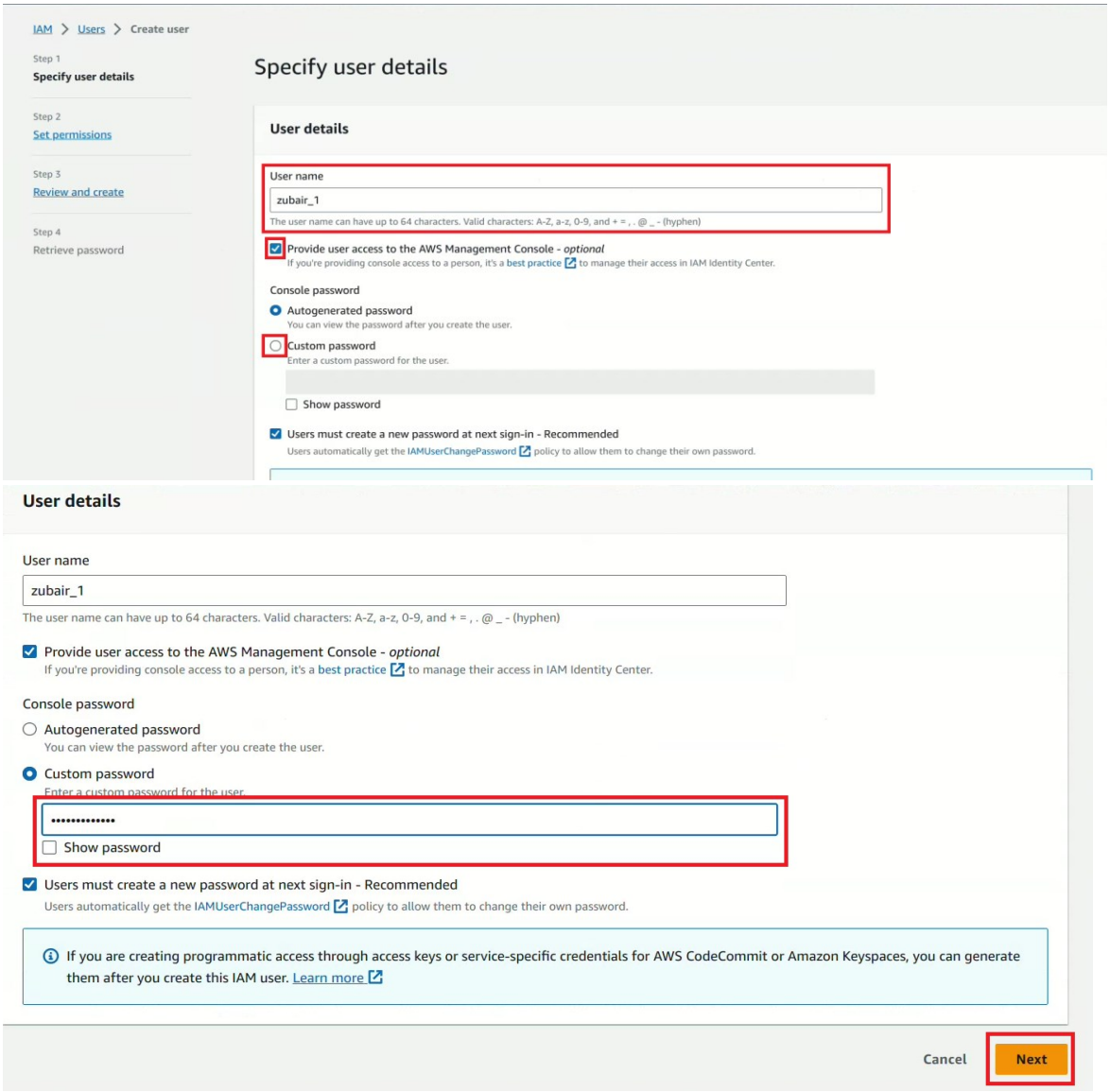
- Search IAM then click on user



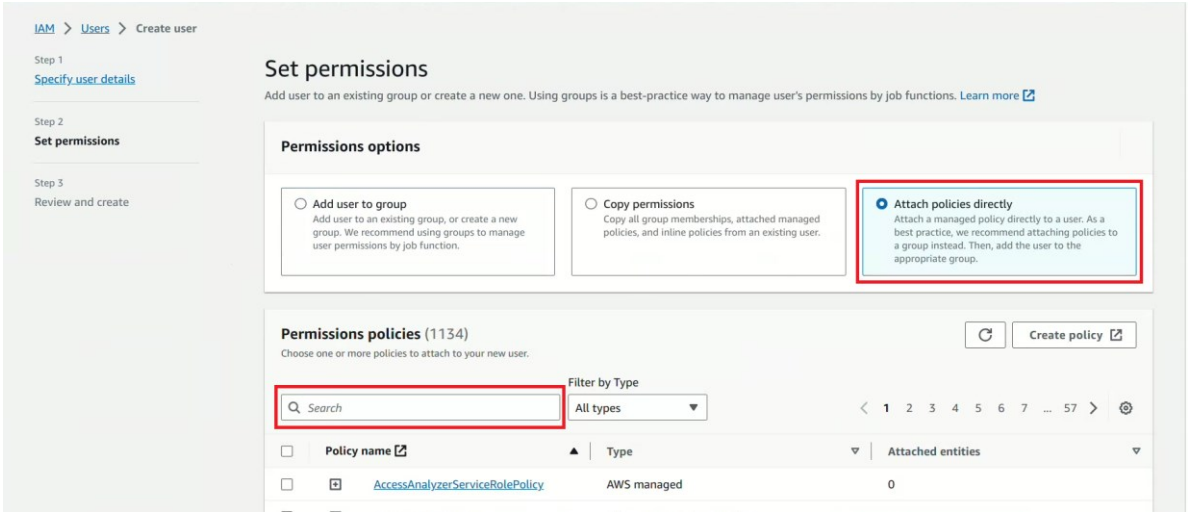
- Click on create user.

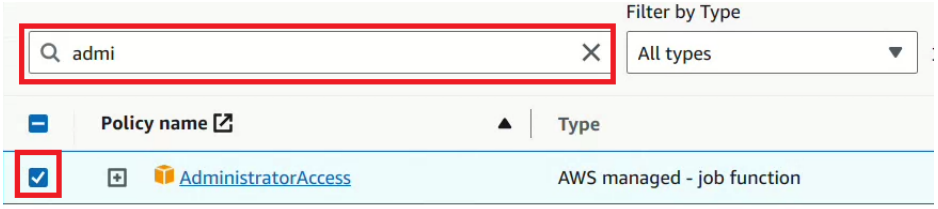


- Give username and give custom password

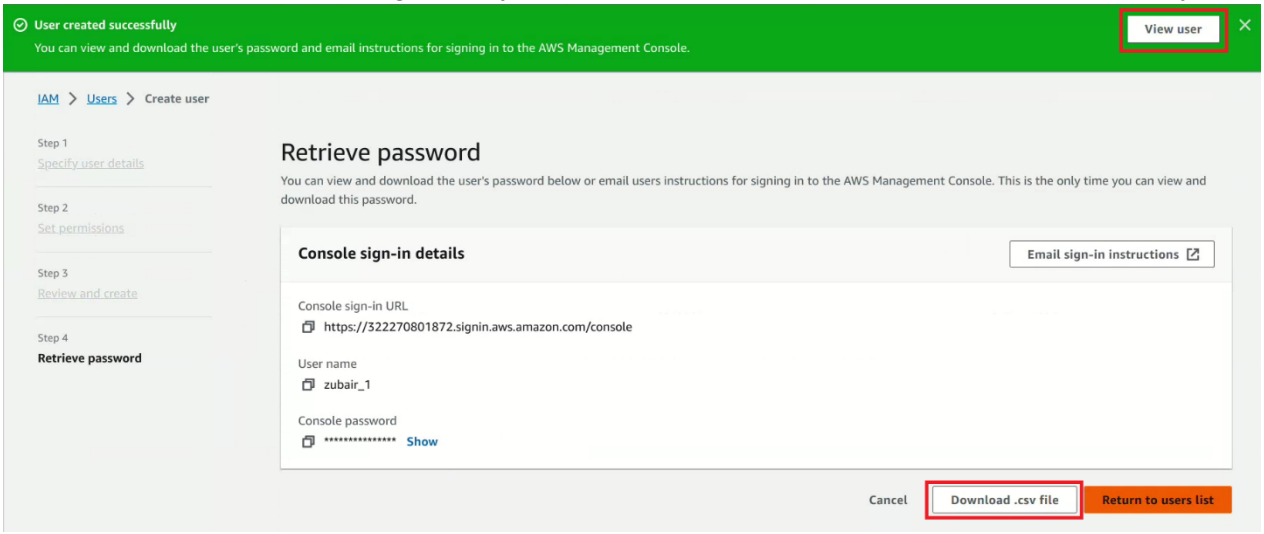


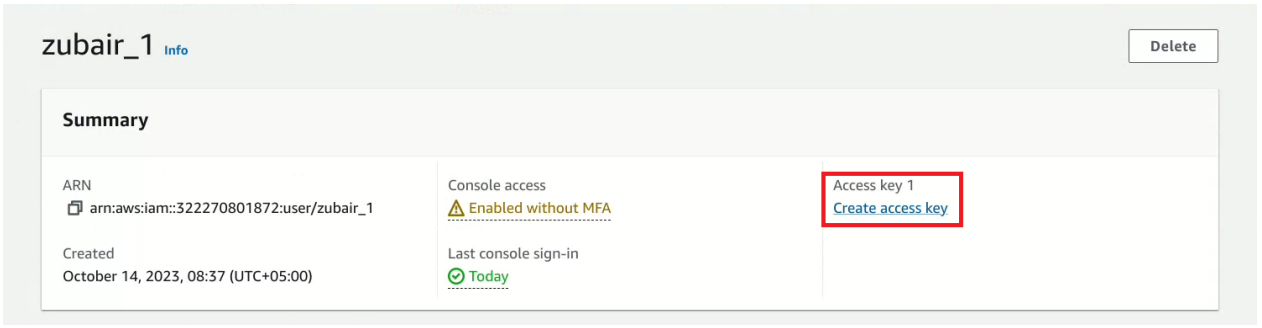
- Choose Attach Policy Directly and search

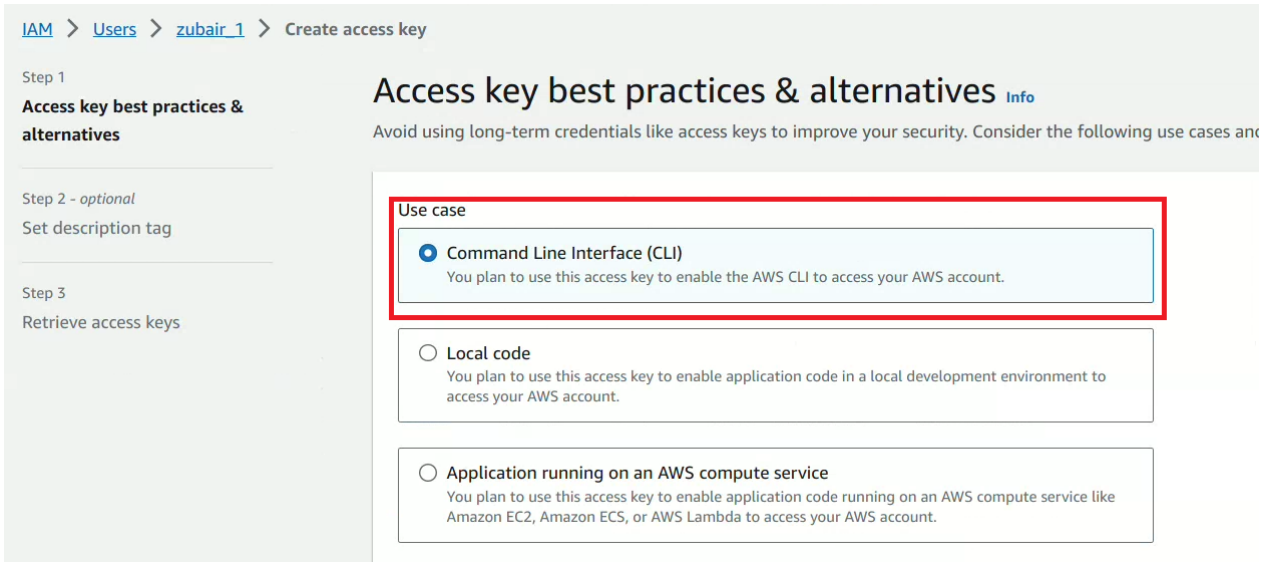


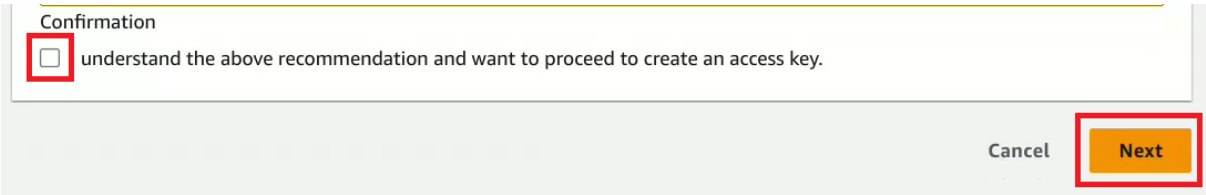


- Review your inputs and create user.
- Download csv file to login with your credentials. Click view user to create access key.

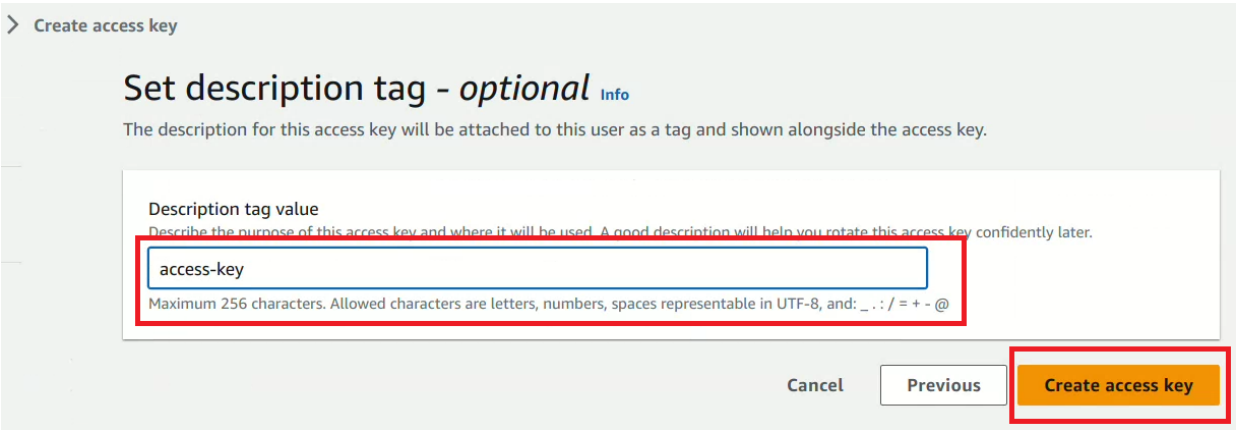




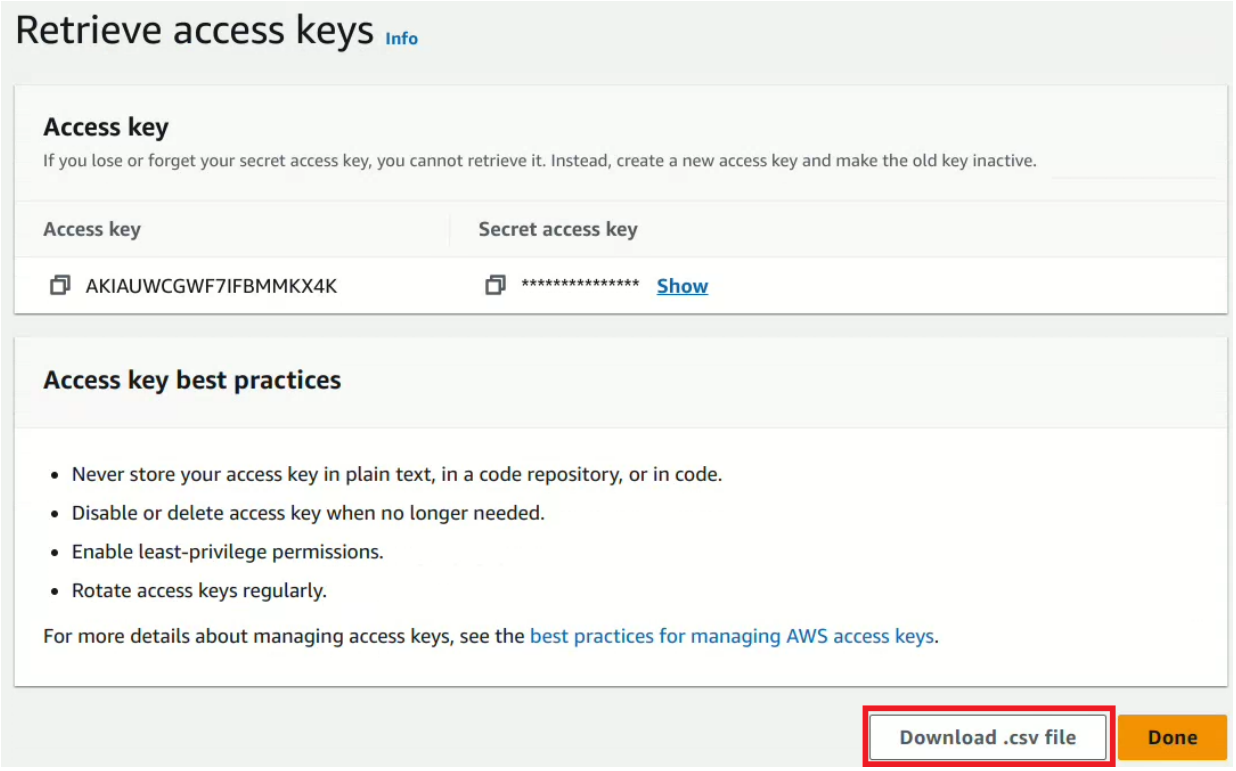




- Give description for your access key to find specific key easily.



- Download csv file of access key credentials to access from AWS CLI

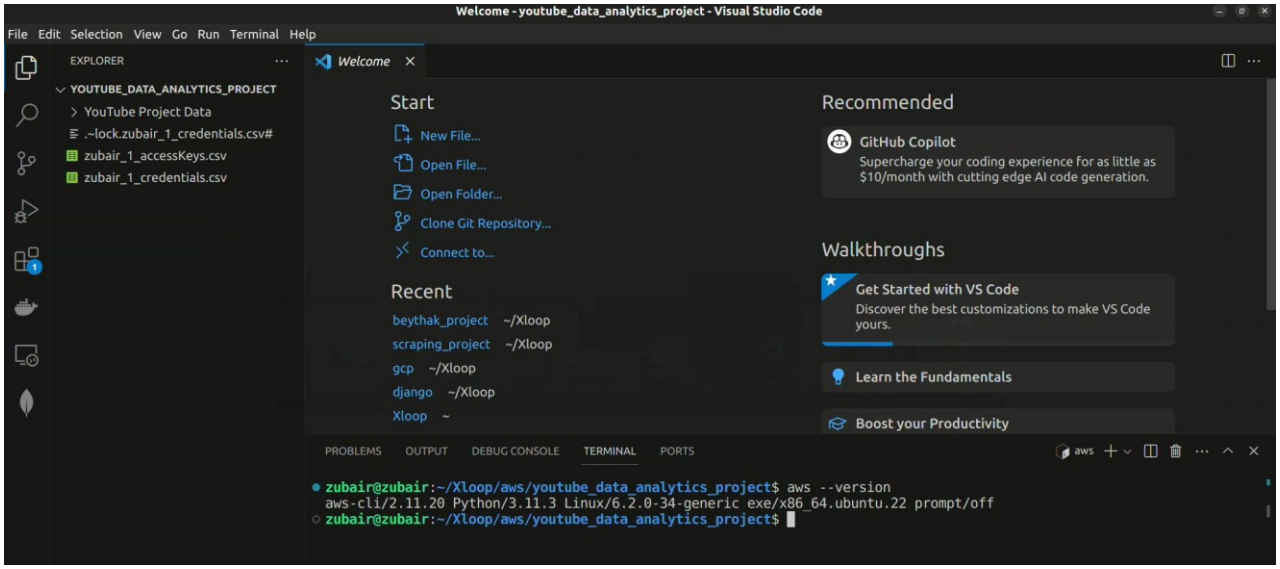


STEP 02

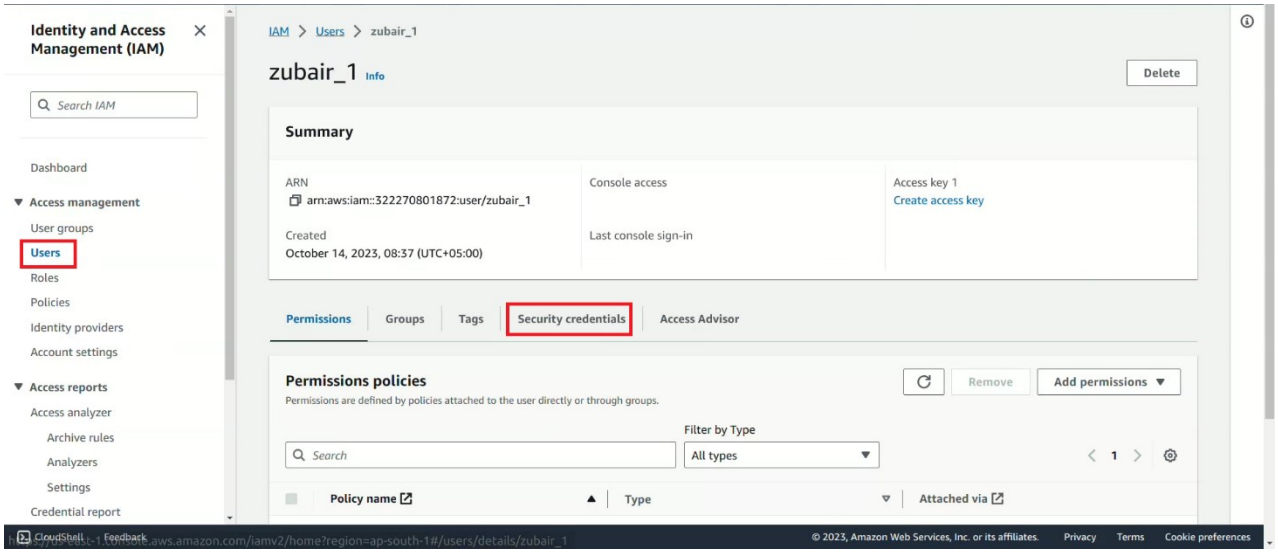
DOWNLOAD AWS CLI

2) Download AWS CLI

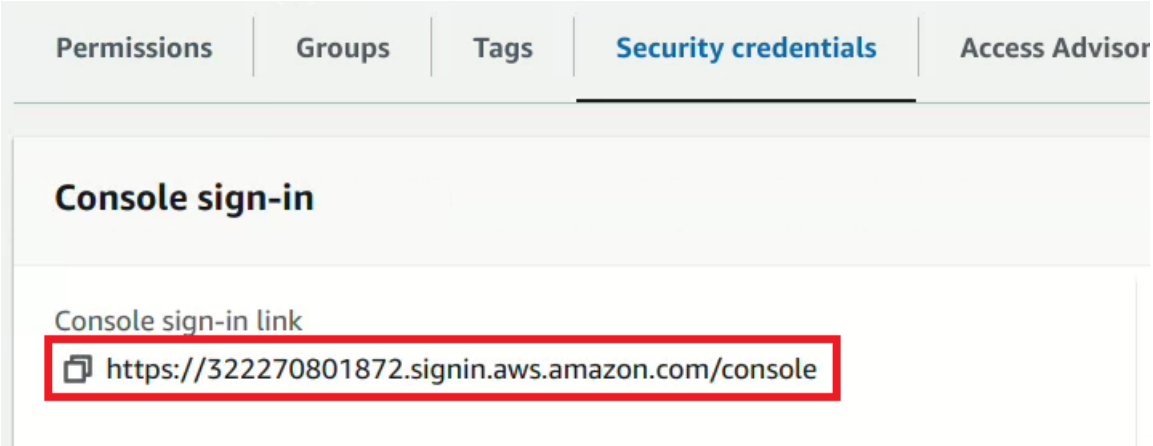
- Download **AWS CLI**, I have installed for Linux
- Create directory with name "**youtube_data_analytics_project**" and open vs code inside that directory, then open terminal to check **AWS CLI** is installed, type "**aws --verison**" in terminal to check **CLI** is installed successfully.



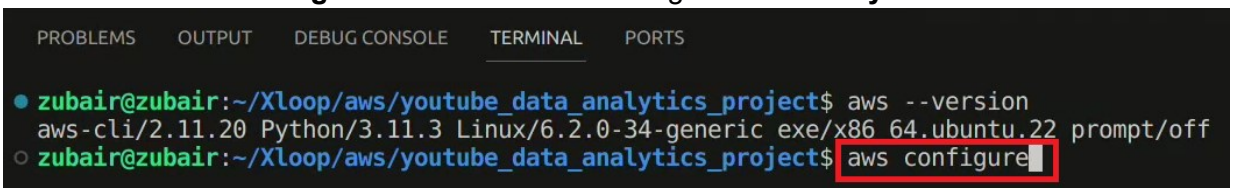
- Again, redirect to newly created user menu and click on "**Security credentials**"



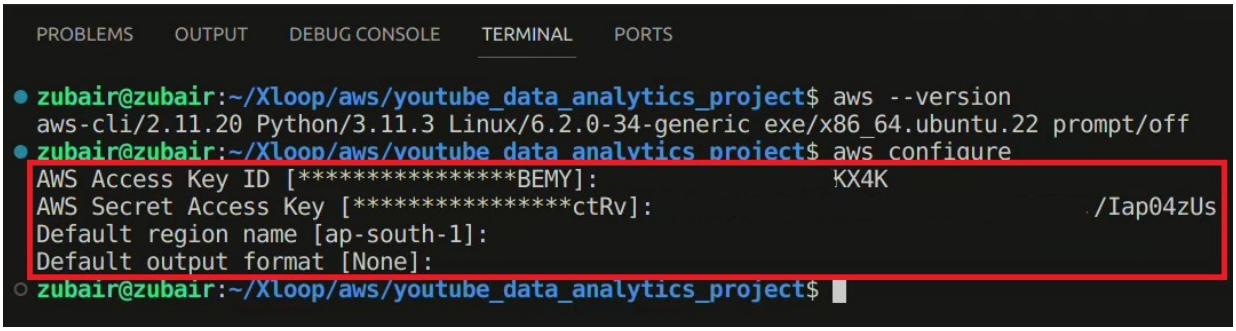
- Copy the console login link



- Logout and signin with IAM user credentials which are downloaded earlier
- Write **aws configure** in vs code terminal to give **access key**



- Put "**access key**" credentials through terminal

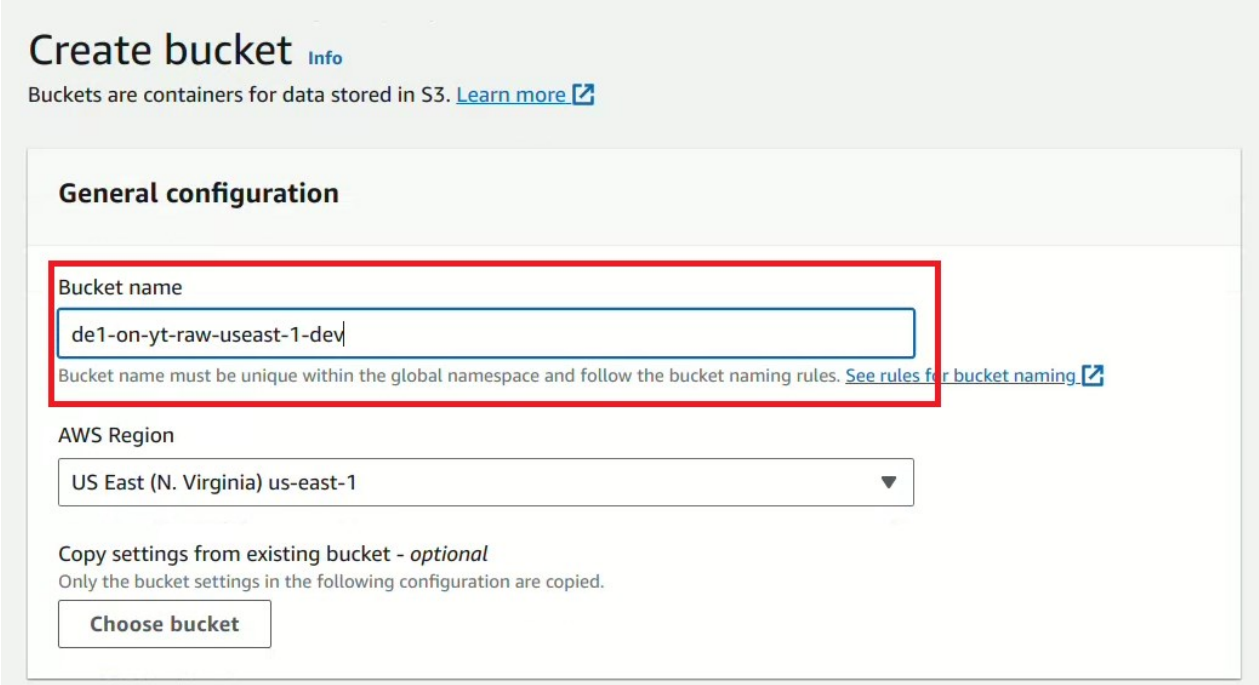


STEP 03

S3 BUCKET

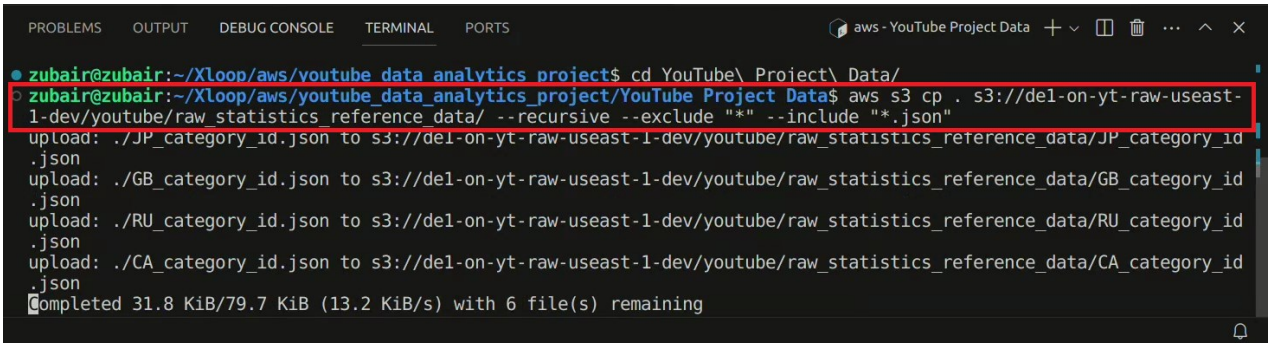
3) Create S3 Bucket

- Search **S3** and click on "**Create Bucket**"
- Follow the naming convention "**`projectname-raw-region-dev`**" and set all thing default then create bucket



4) Copy data to S3

- First download data from **GitHub**
- Unzip the downloaded data
- Change folder to where data is stored using terminal and copy json file command from **GitHub** to copy all the json data using AWS CLI into S3



- Also use more command from **GitHub** to copy CSV files

```
zubair@zubair:~/Xloop/aws/youtube_data_analytics_project/YouTube Project Data$ aws s3 cp CAvideos.csv s3://de1-on-yt-raw-useast-1-dev/youtube/raw_statistics/region=ca/
```

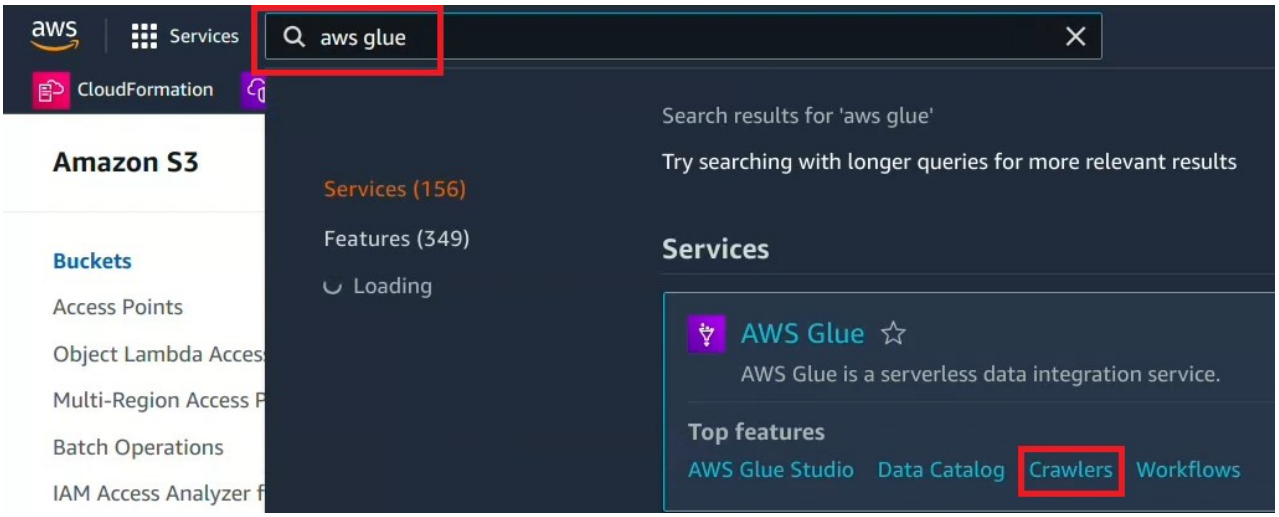
- Change bucket name to you own bucket name

Step 04

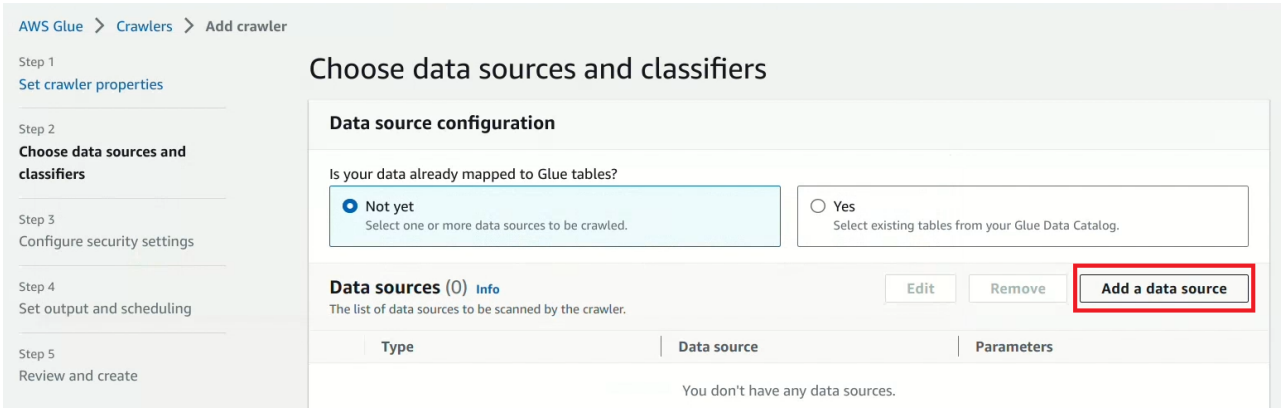
AWS Glue

5) Create Glue Crawler

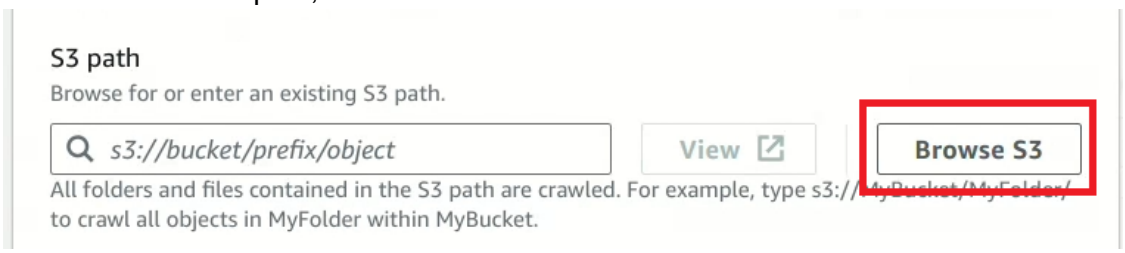
- Search AWS Glue

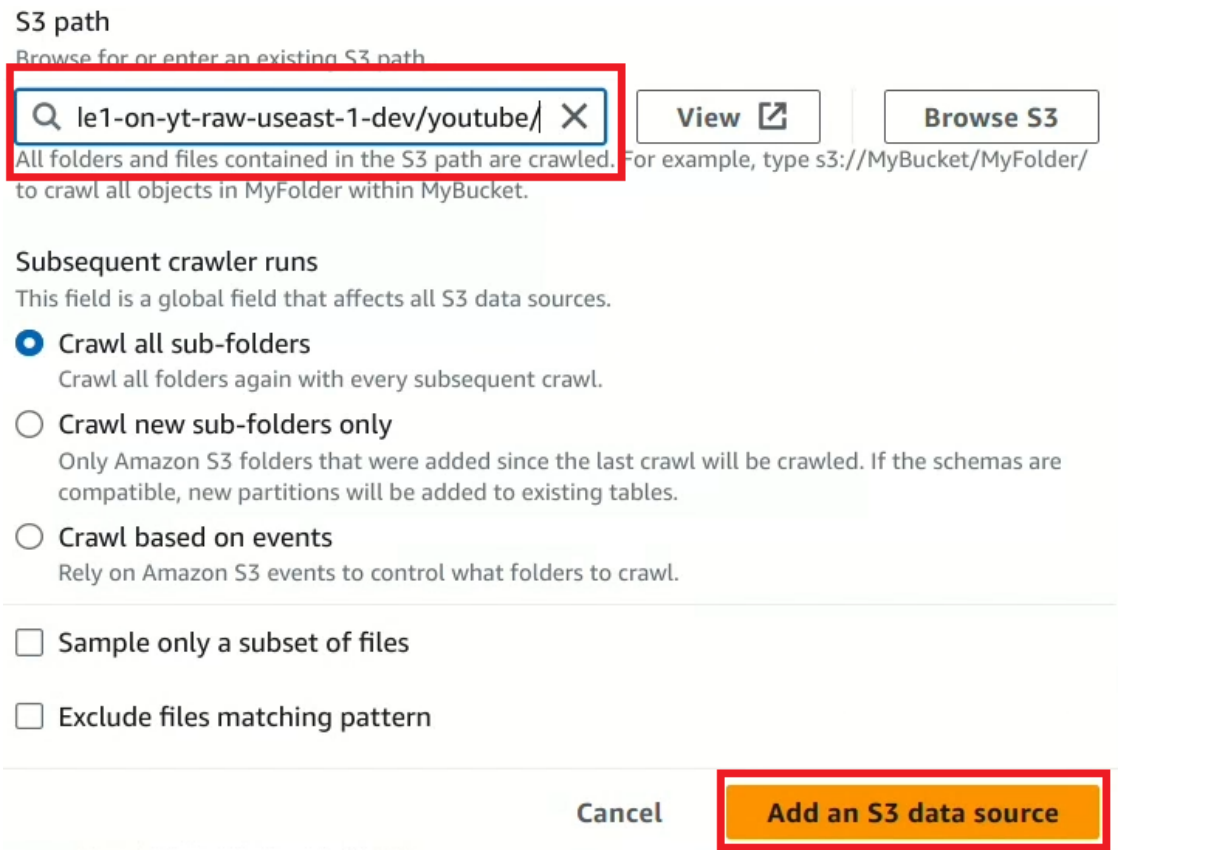


- Give naming convention **de-on-yt-raw-glue-catalog-1** and click **Next**
- Add data source

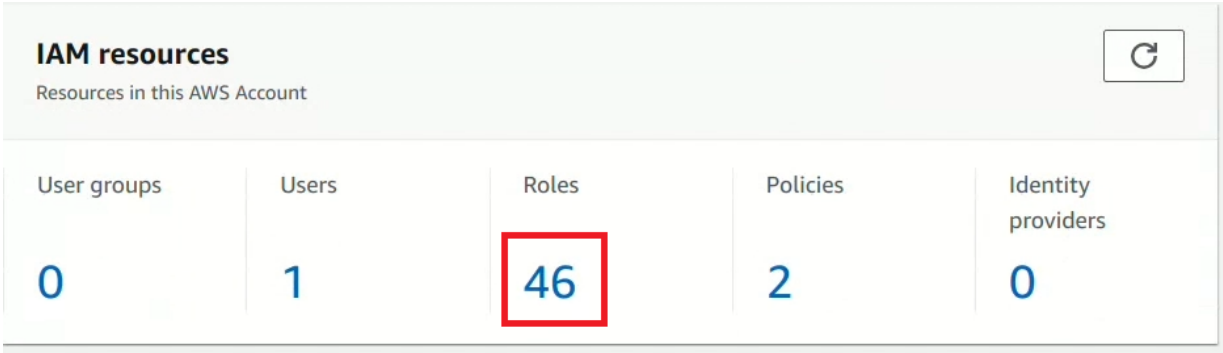


- Choose S3 path, then Click on **Next**

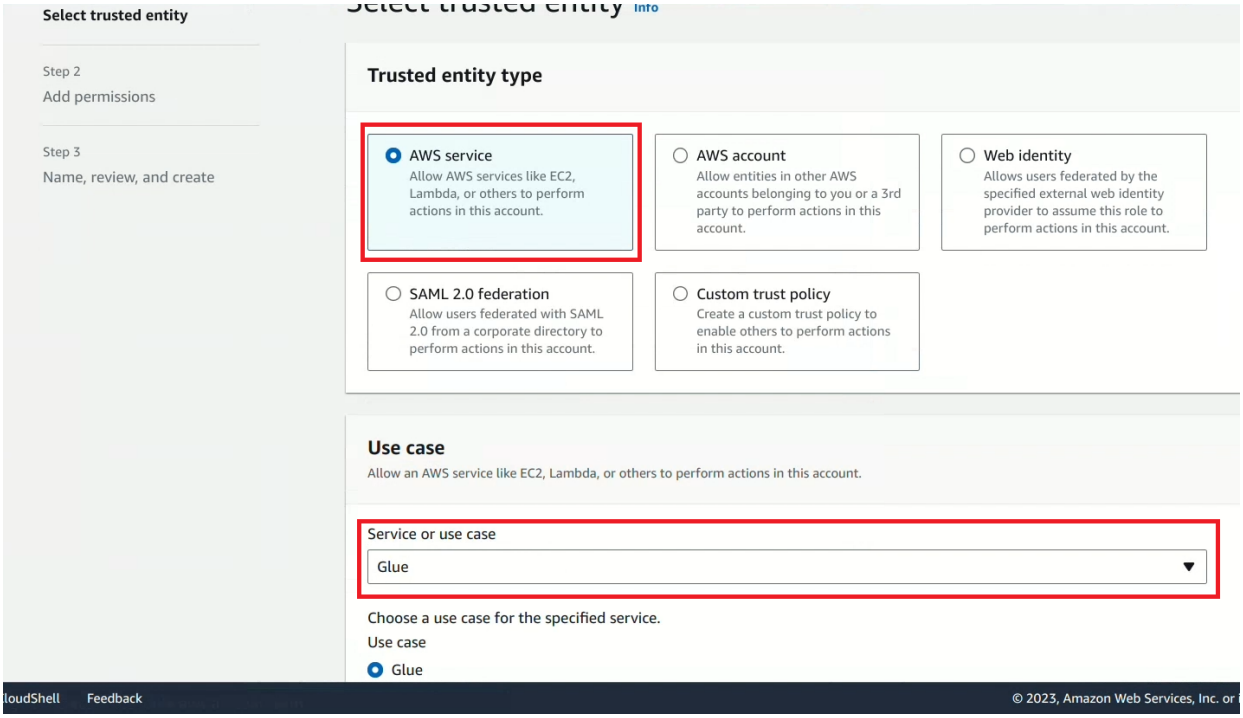




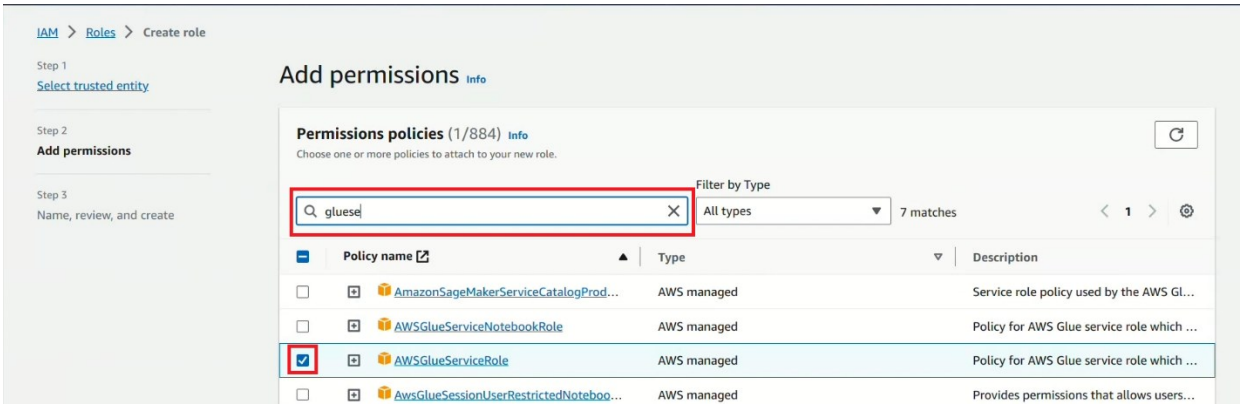
- Create IAM role on existing IAM User, search IAM
- Click on roles, create roles



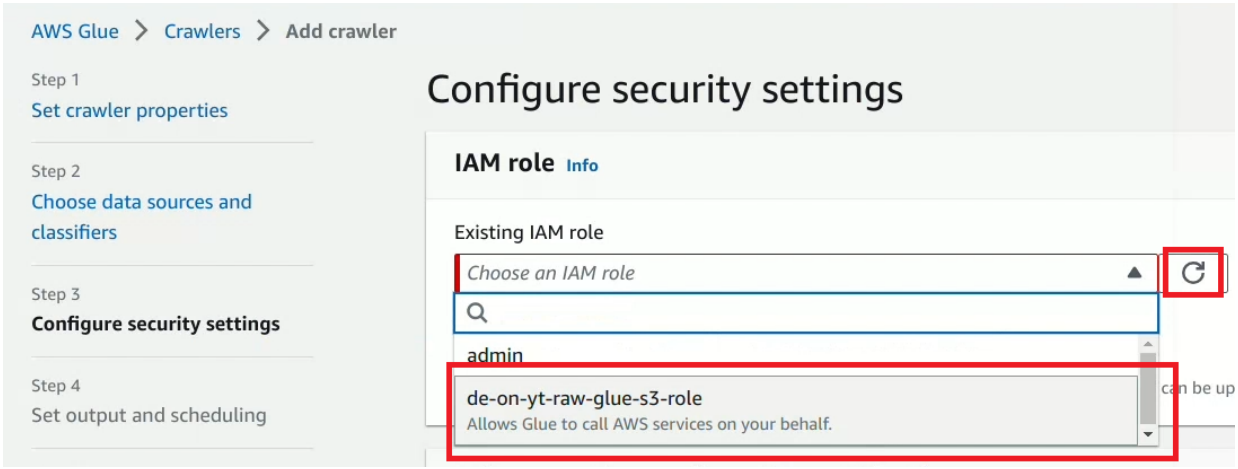
- Select Glue, choose Glue and Click **Next**



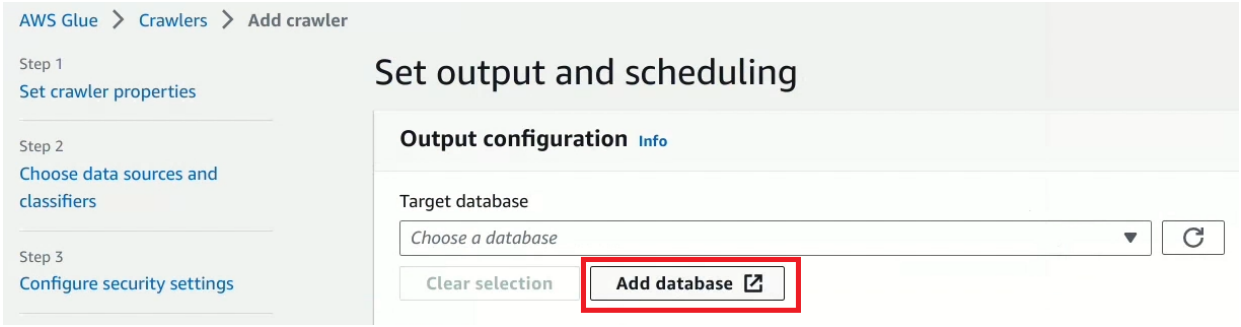
- Choose permission **AmazonS3FullAccess** and **AWSGlueServiceRole** and click **Next**



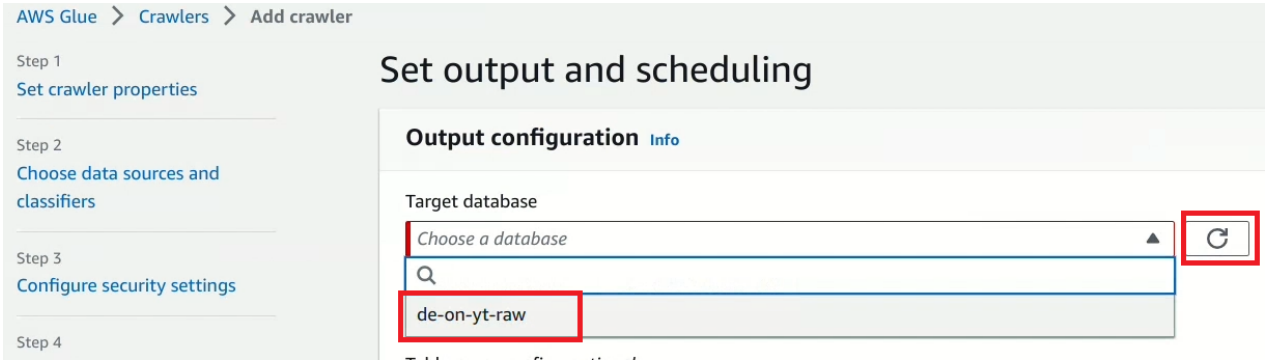
- Give role name with naming convention **de-on-yt-raw-glue-s3-role** and "**Create role**"
- Choose IAM role back into Crawler menu and click **Next**



- Create new database, with naming convention **de-on-yt-raw** and click "**Create database**"



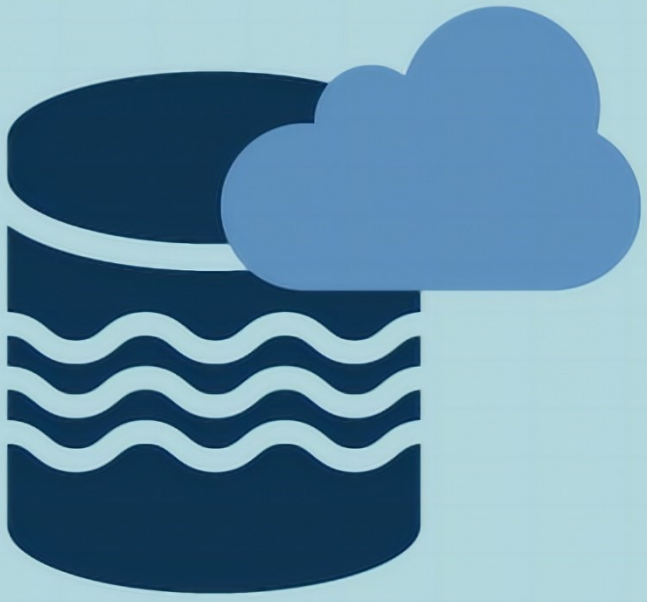
- Choose newly created database and click **Next**



- Then review the all things and "**Create crawler**"
- Run Crawler to build table into the selected database

Step 05

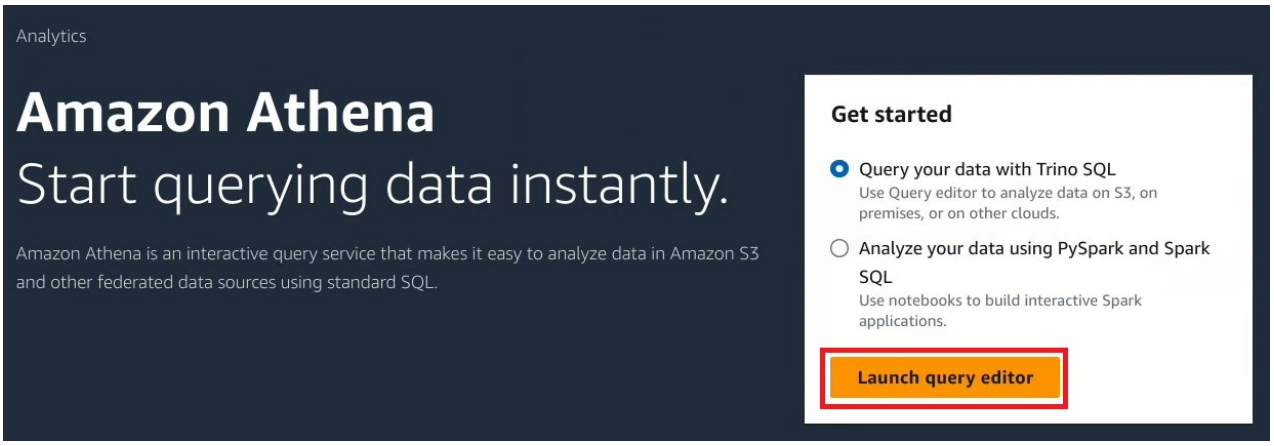
AWS Athena



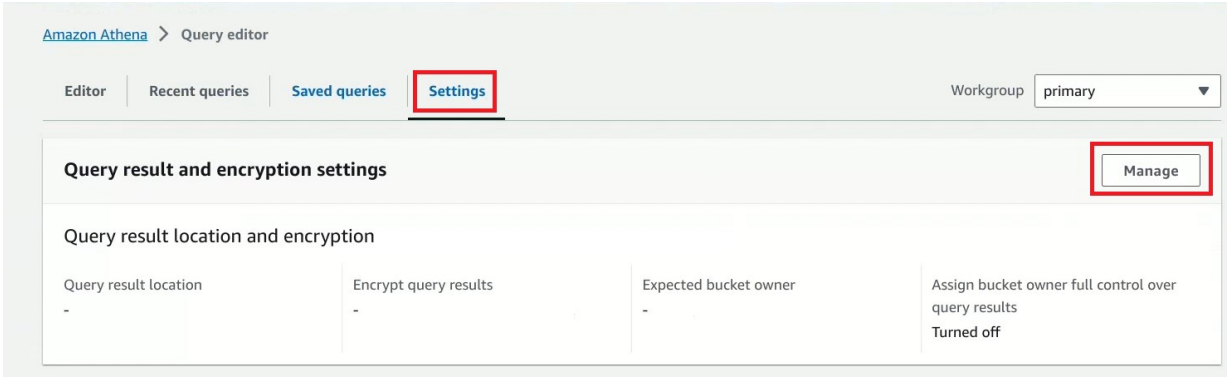
amazon
ATHENA

6) Run Crawler and View Data Using Athena

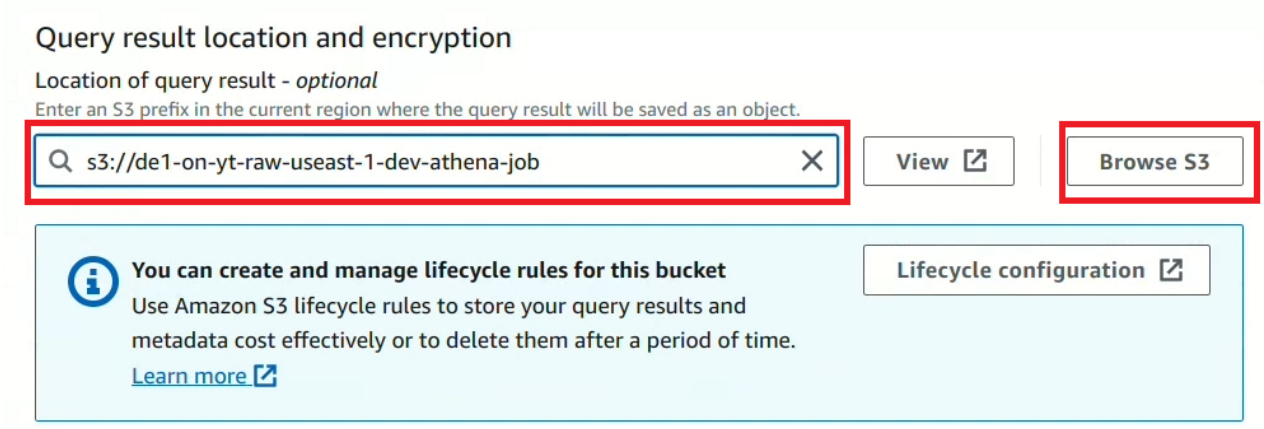
- After running crawler, search Athena



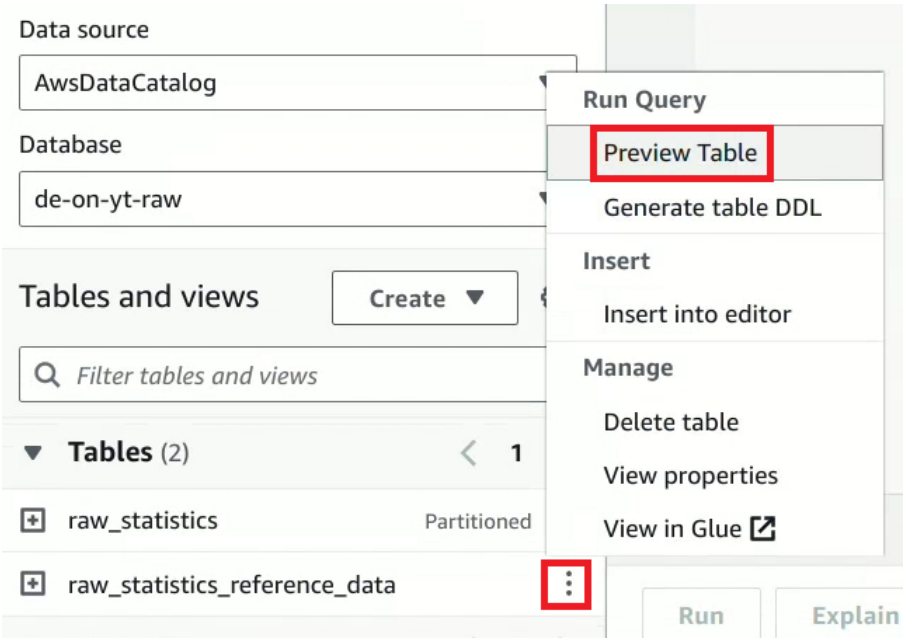
- Click on settings tab



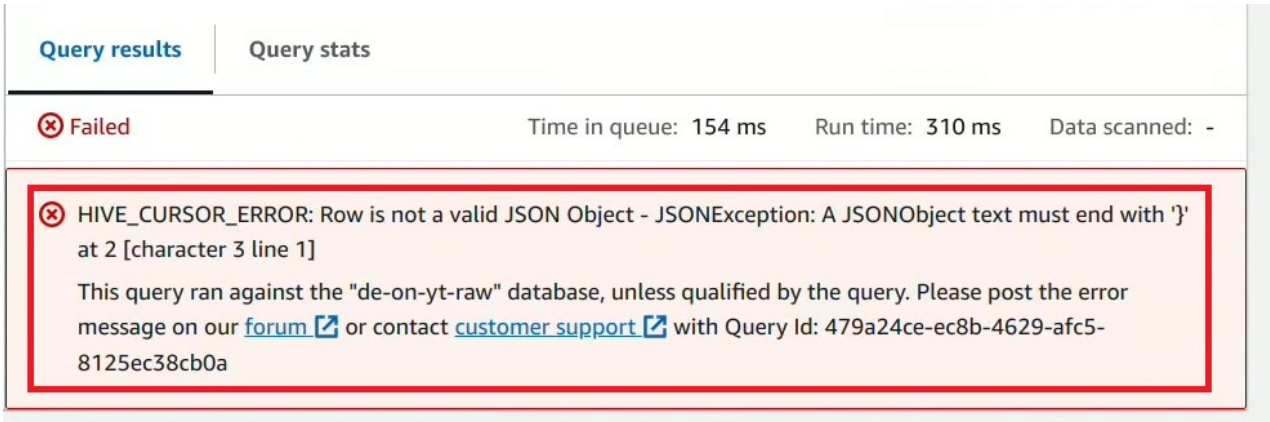
- Then again create new bucket in S3 with the convention "**de-on-yt-raw-useast-1-dev-athena-job**"
- Choose edit setting in athena menu and select newly created bucket for SQL queries and click **Save**.



- Select Data source as **AwsDataCatalog** and **Database** as "***de-on-yt-raw***"



- There will be the json format issue if we run the athena query



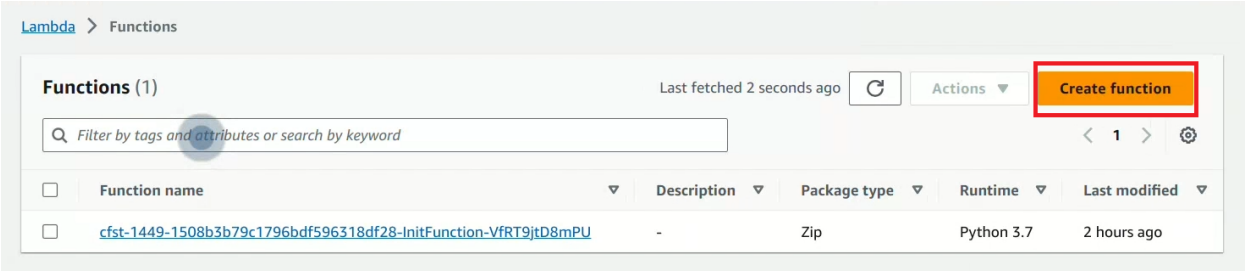
- To resolve this, create a lambda function to preprocess the data

Step 06

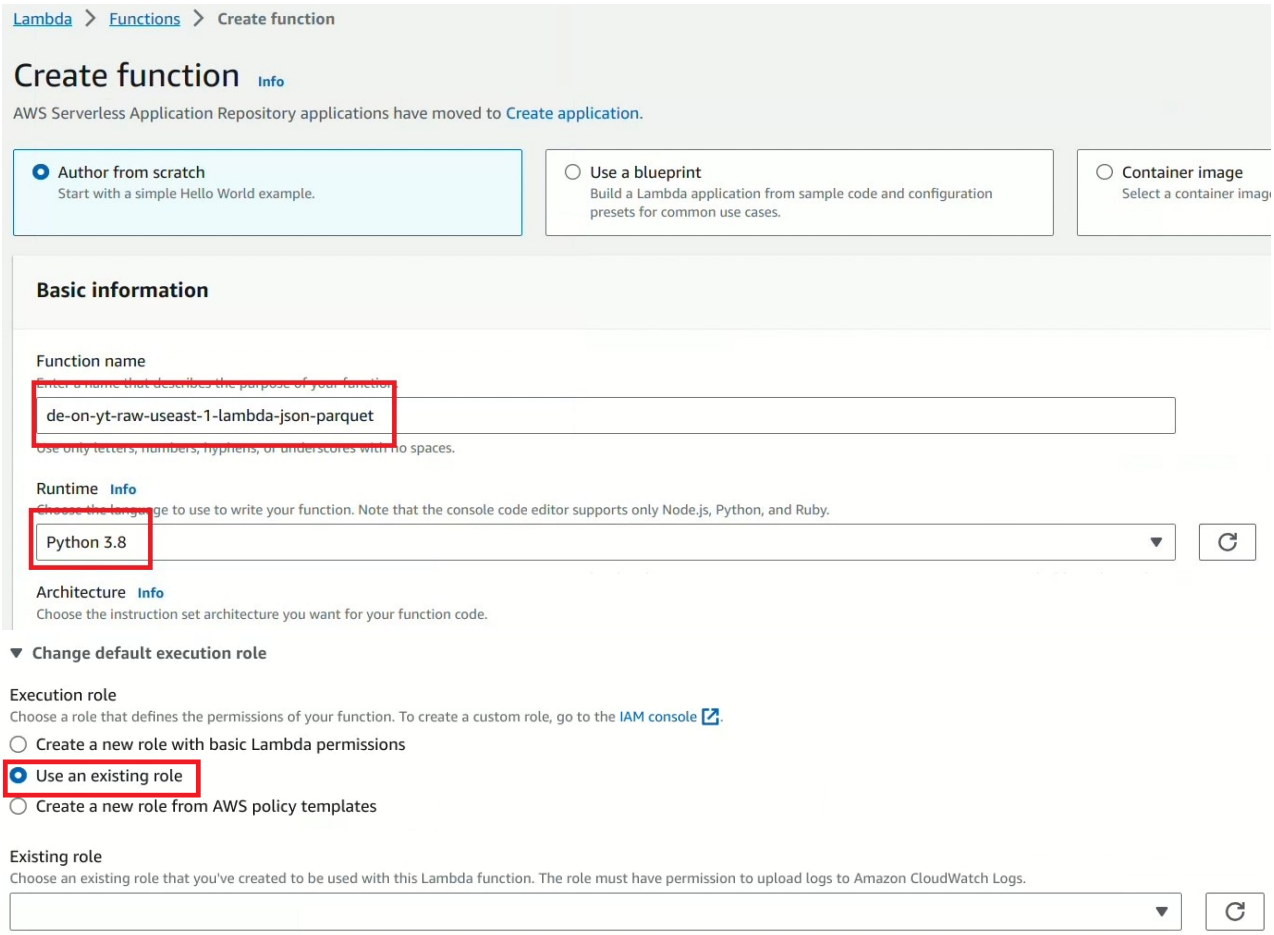
AWS Lamda

7) Create the Lambda Function

- Search lambda

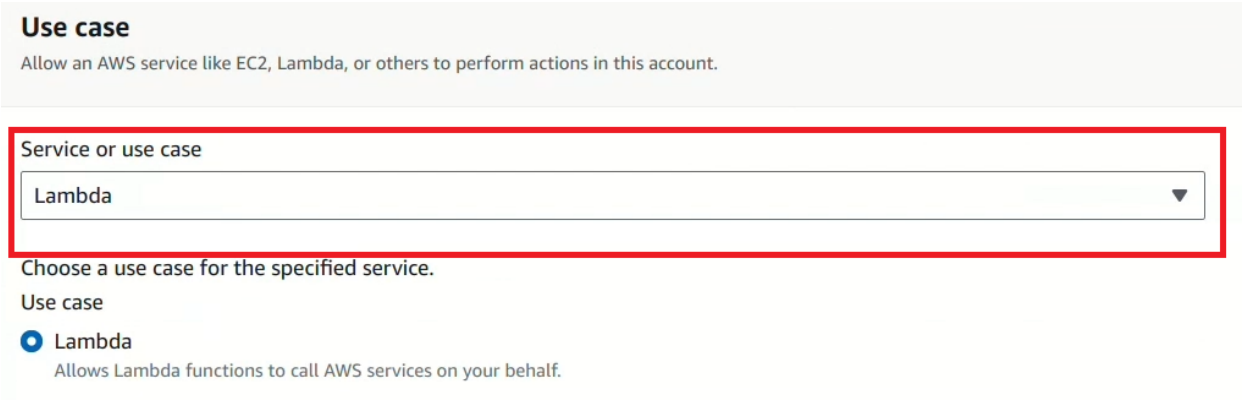


- Give naming convention " **de-on-yt-raw-useast-1-lambda-json-parquet** ", choose runtime python 3.8 and Click on "**Change default execution role**"

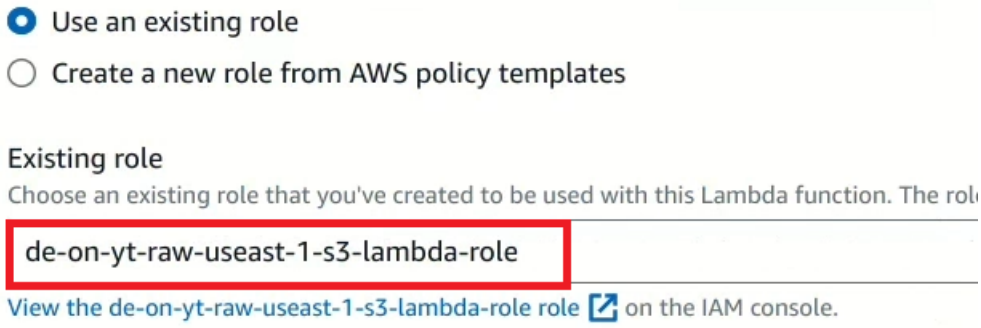


- Create new role for lambda

- Select lambda and click next, then select permission policy **“AmazonS3FullAccess”** and **“AWSGlueServiceRole”**



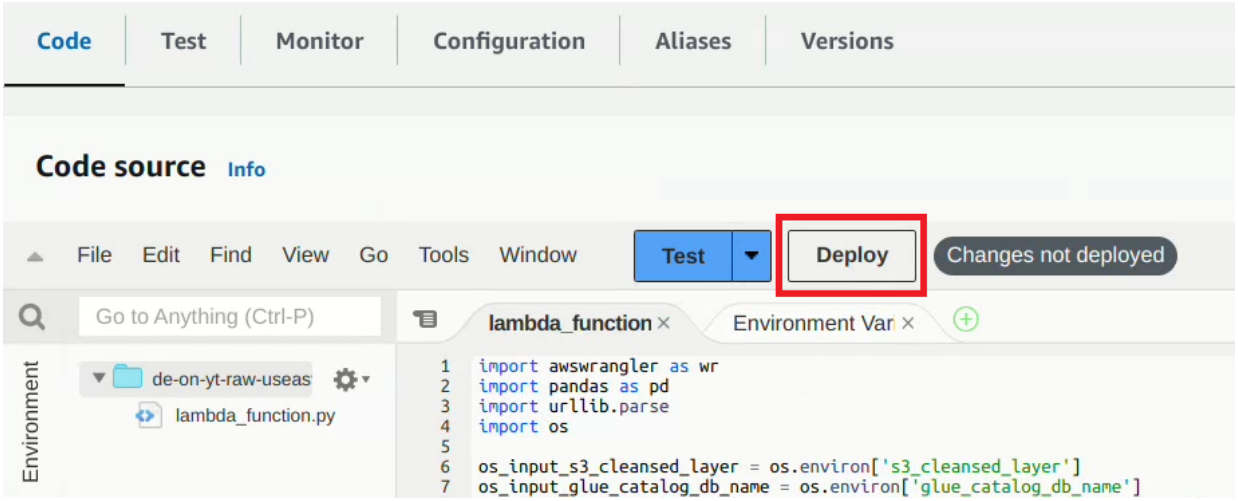
- Then give naming convention “**de-on-yt-raw-useast-1-s3-lambda-role**”
- Choose existing role from lambda menu



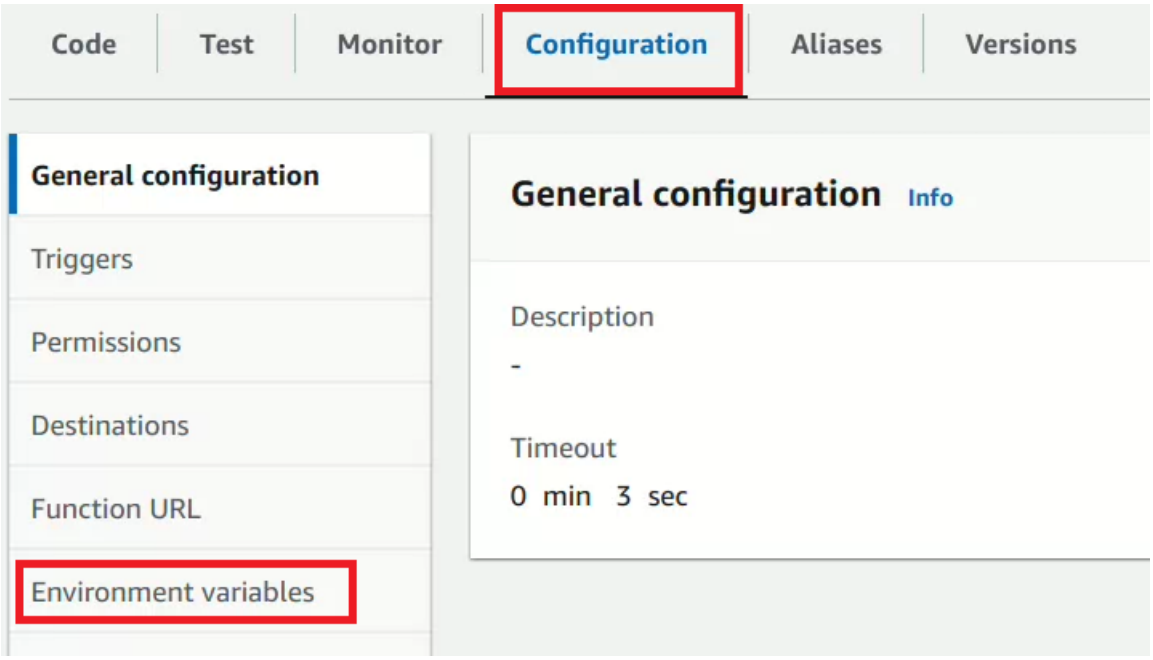
- Create function

8) Adding preprocessing code and assigning environment variables

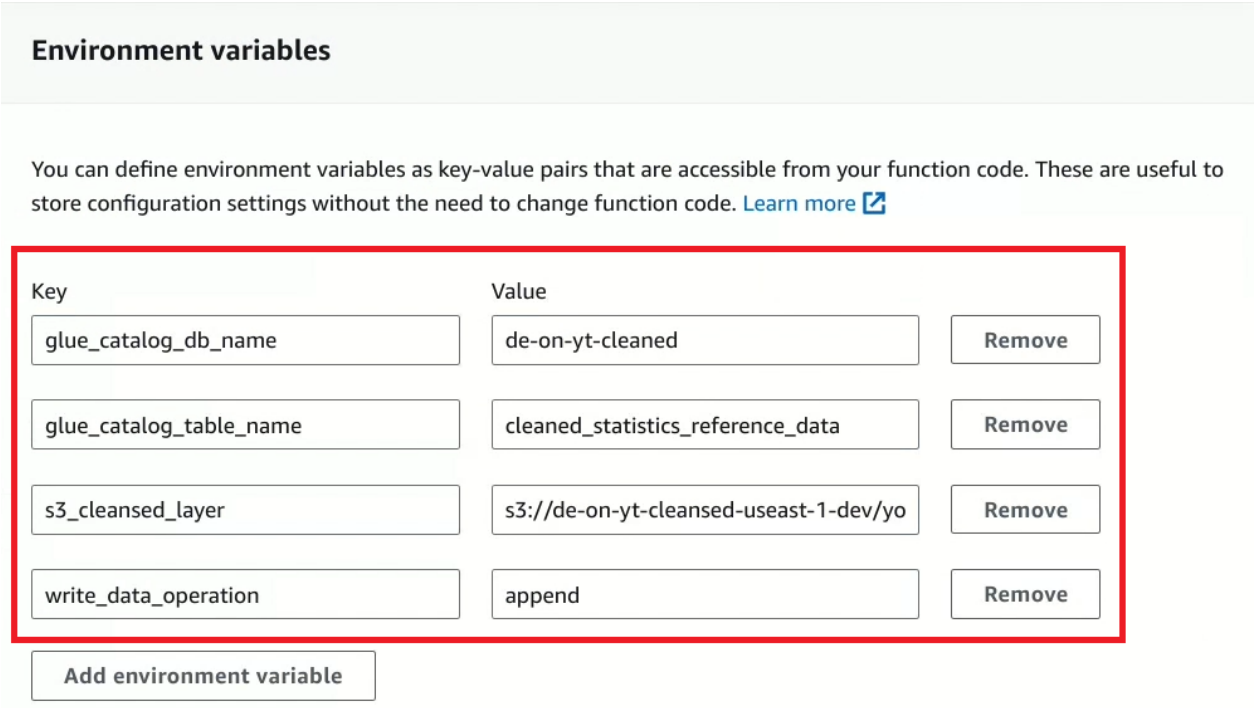
- Copy code from **GitHub** and paste in “***code tab***” of lambda function



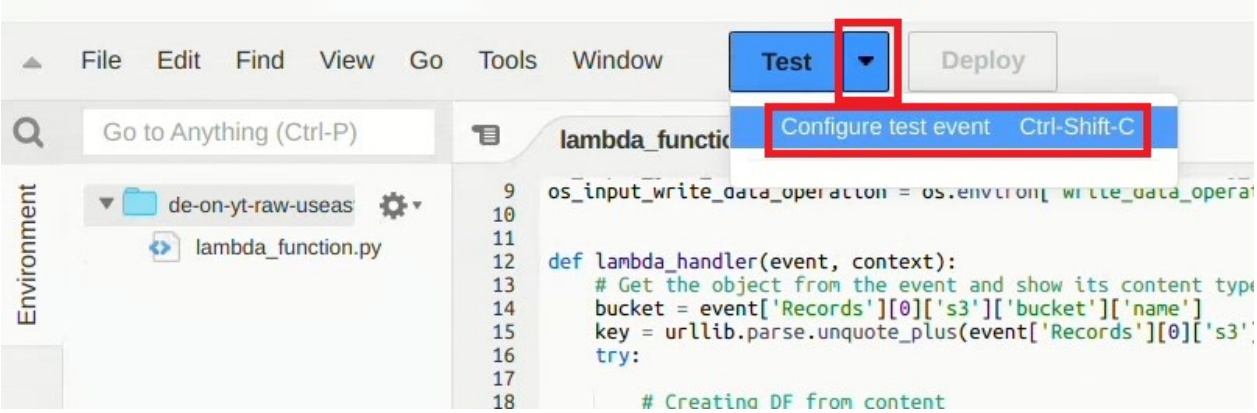
- Select **configuration** tab, and choose **Environment Variables**



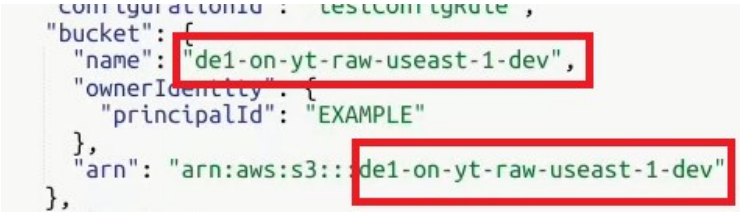
- Assign environment variable,
    - glue_catalog_db_name, de-on-yt-cleaned(*you can create any database of your choice, here we have to create 2 databases, 1 for raw and 1 for cleaned*)
    - glue_catalog_table_name, cleaned_statistics_reference_data
    - s3_cleansed_layer, s3://de-on-yt-cleansed-useast-1-dev/youtube
    - write_data_operation, append
- then click **Save** button and create another s3 bucket for environment variable with naming convention "**de-on-yt-cleansed-useast-1-dev**"



- these environment variables of lambda function will create a data base with name **“de-on-yt-cleaned”** and create a table inside the database with name **“cleaned_statistics_reference_data”** and store all data into the bucket to query later using athena
- After creating **S3 bucket**, choose **test** form **code** and select **configure test event**



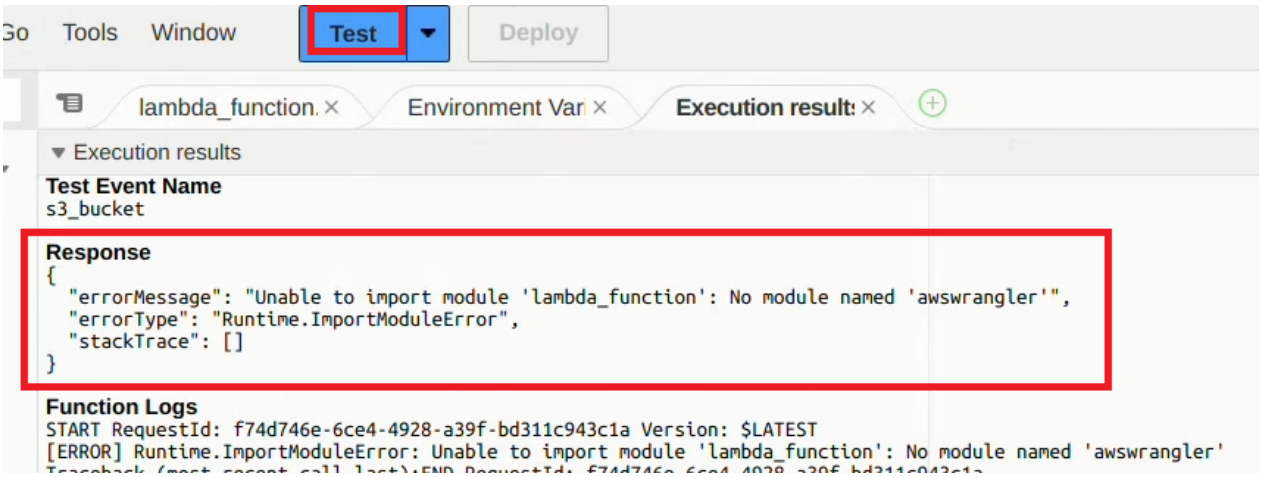
- Give any name which relate to template and select template **s3-put** and change example-bucket with ***bucket name***, edit **arn** example-bucket with ***bucket name*** and **key** with **s3 uri** of ***CA_category_id.json***



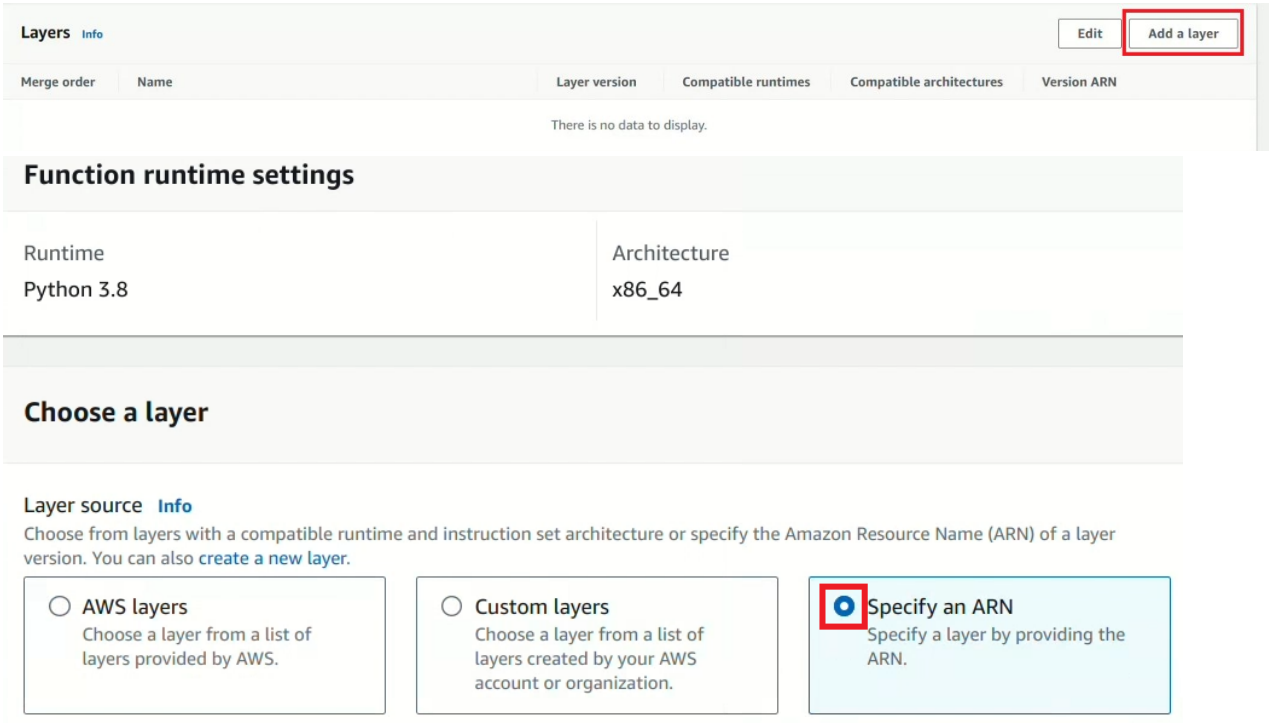
- Edit uri by removing *s3://de-on-yt-raw-useast-1-dev/*

```
    "arn": "arn:aws:s3:::de1-on-yt-raw-useast-1-dev"
  },
  "object": {
    "key": "youtube/raw_statistics_reference_data/CA_category_id.json",
    "size": 1024,
    "eTag": "0123456789abcdef0123456789abcdef",
    "sequencer": "0A1B2C3D4E5F678901"
  }
}
```

- Then click on **save** button and run test and solve the error.

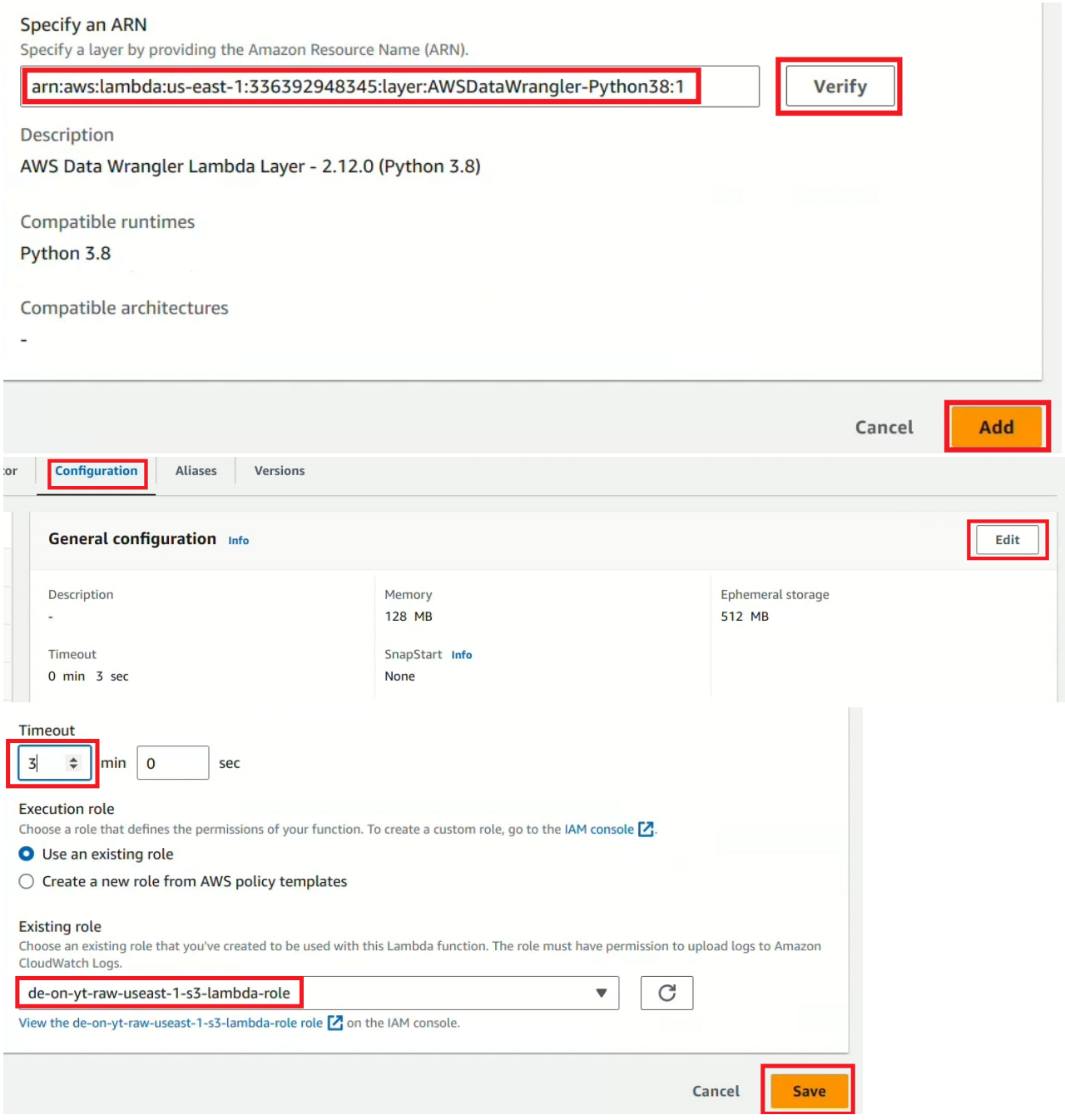


- Click on layers and set layer to "**AWSDataWrangler-Python38**" through **Specify ARN** from source, then change timeout to 3 minute in general configuration of **configuration** tab



AWS Data Wrangler Version	Python Version	Layer ARN
2.12.0	3.7	arn:aws:lambda:<region>:336392948345:layer:AWSDataWrangler-Python37:1
2.12.0	3.8	arn:aws:lambda:<region>:336392948345:layer:AWSDataWrangler-Python38:1
2.13.0	3.7	arn:aws:lambda:<region>:336392948345:layer:AWSDataWrangler-Python37:2
2.13.0	3.8	arn:aws:lambda:<region>:336392948345:layer:AWSDataWrangler-Python38:2
2.13.0	3.9	arn:aws:lambda:<region>:336392948345:layer:AWSDataWrangler-Python39:1

- Change the region to your current region where s3 bucket is stored or the region which you are using from the very start.



- Run test and response should be:

```
{
  "paths": [
    "s3://de1-on-yt-cleansed-useast-1-
dev/youtube/52e1092eda44402d93cfbf55305c3f4d.snappy.parquet"
  ],
  "partitions_values": {}
}
```

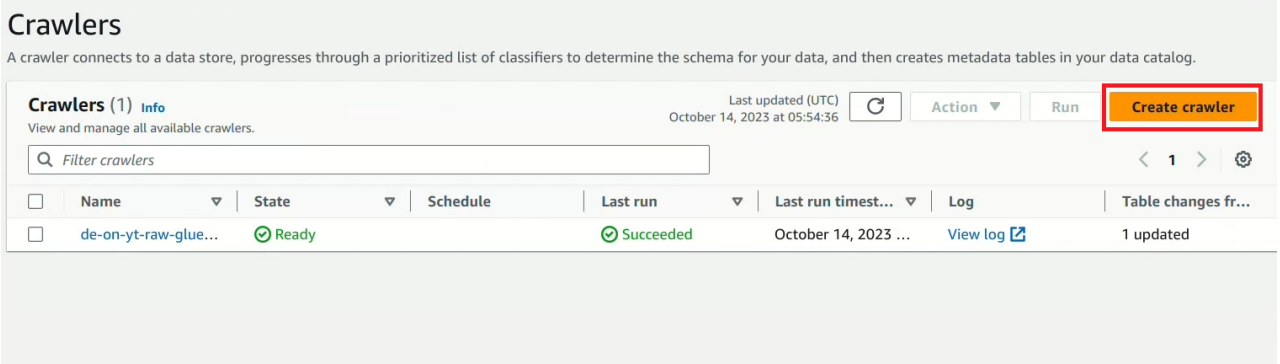
- After all successful process, second we have to access all json data and then CSV data into athena catalog.

STEP 07

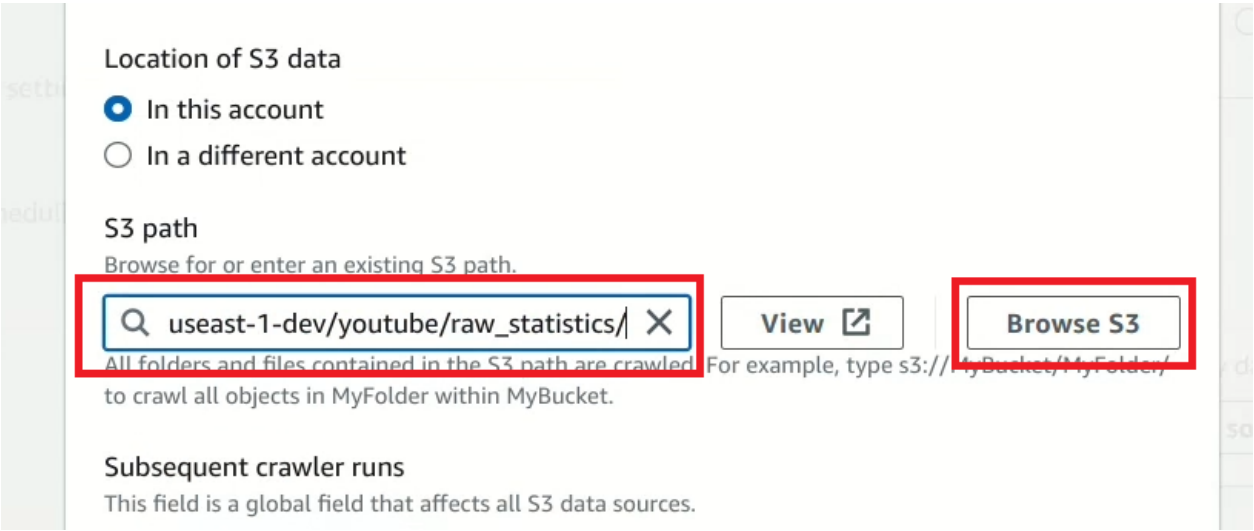
PREPROCESS THE DATA

9) Create Crawler for CSV files

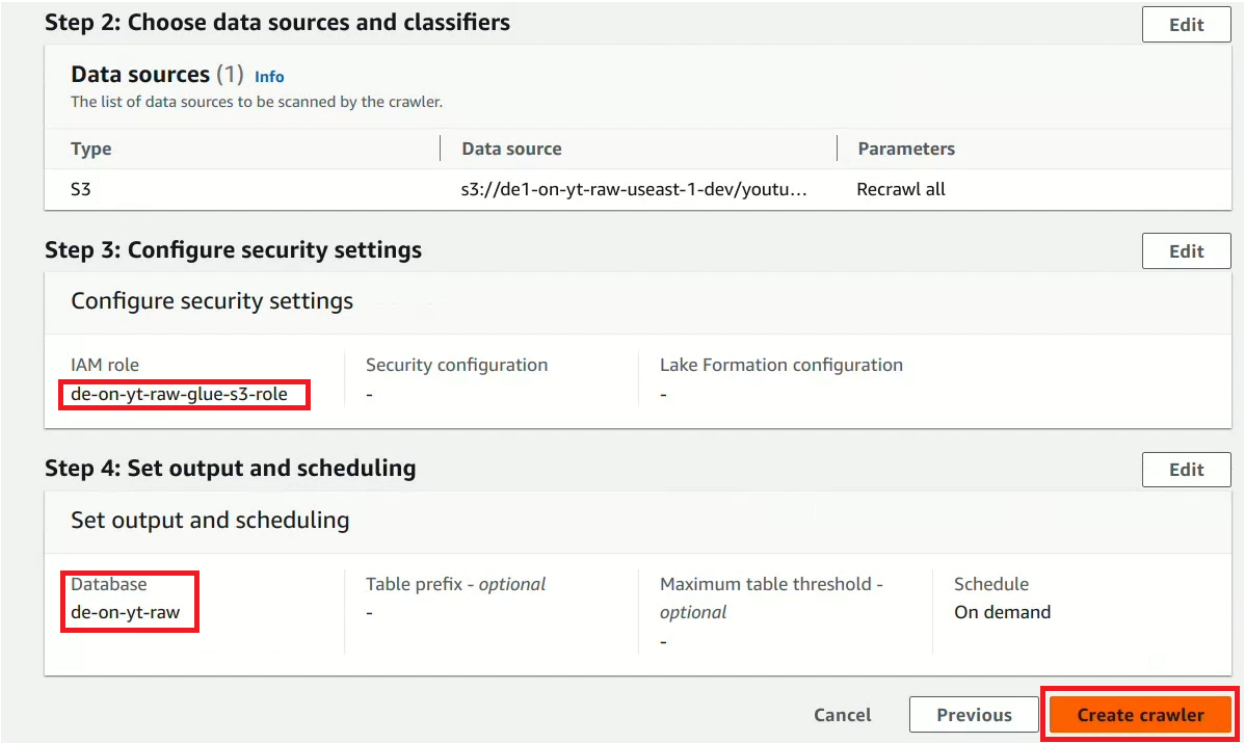
- Move to **AWS Glue** service
- Open **crawler** tab and create **crawler**



- Give naming convention "**de-on-yt-raw-glue-csv-crawler-01**", then set **S3 bucket** path for **raw_statistics**

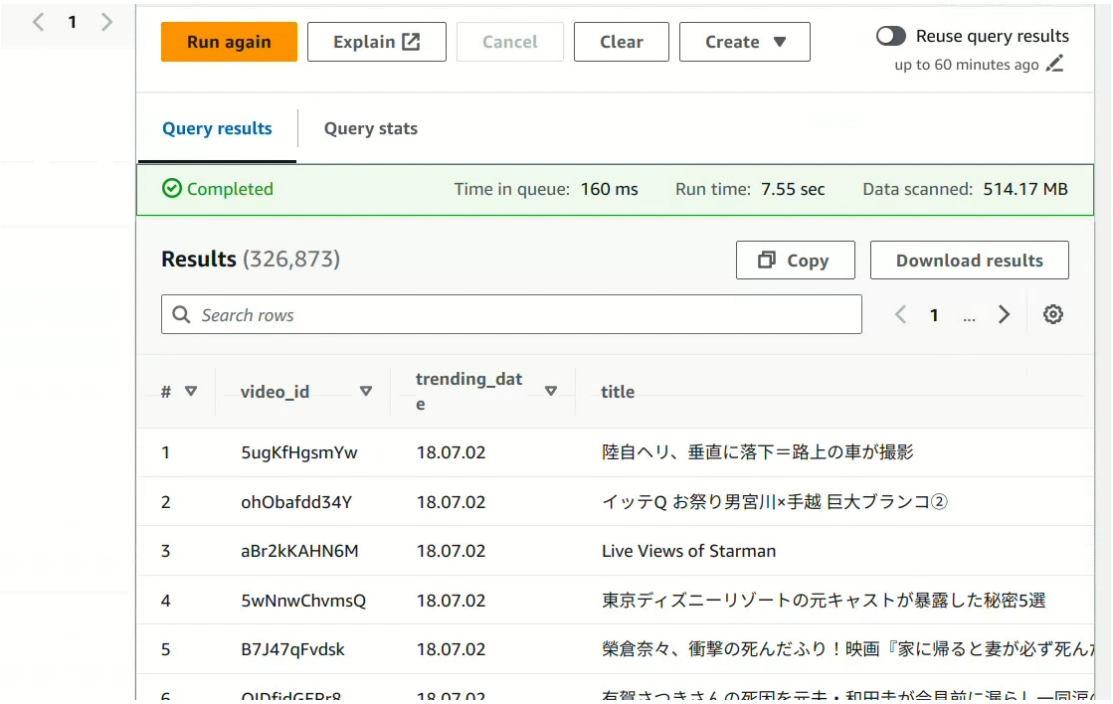


- Choose existing IAM role with name "**de-on-yt-raw-glue-s3-role**"
- Then choose the database for raw named "**de-on-yt-raw**"
- After creating the crawler, **run crawler**.

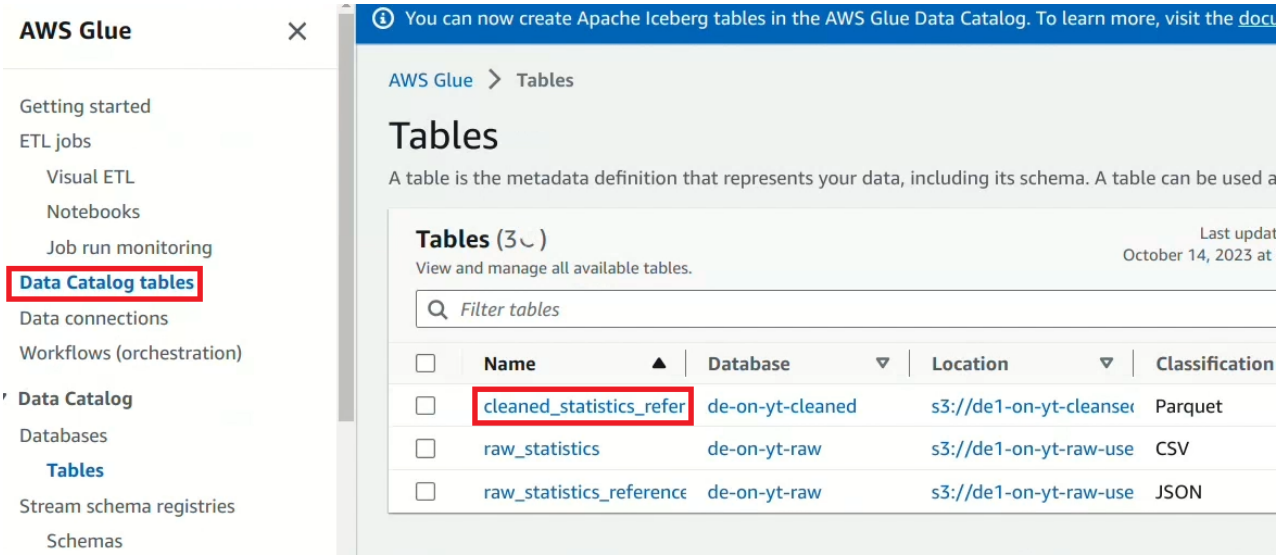


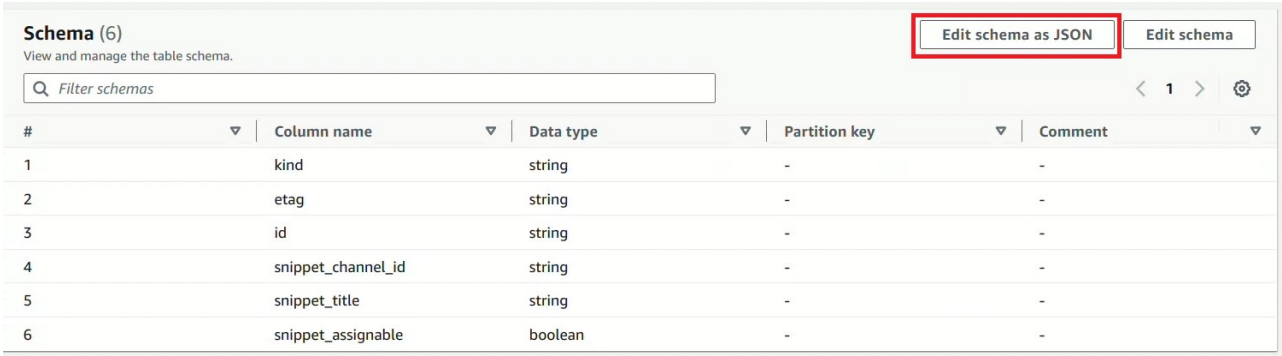
10) Run SQL query over athena

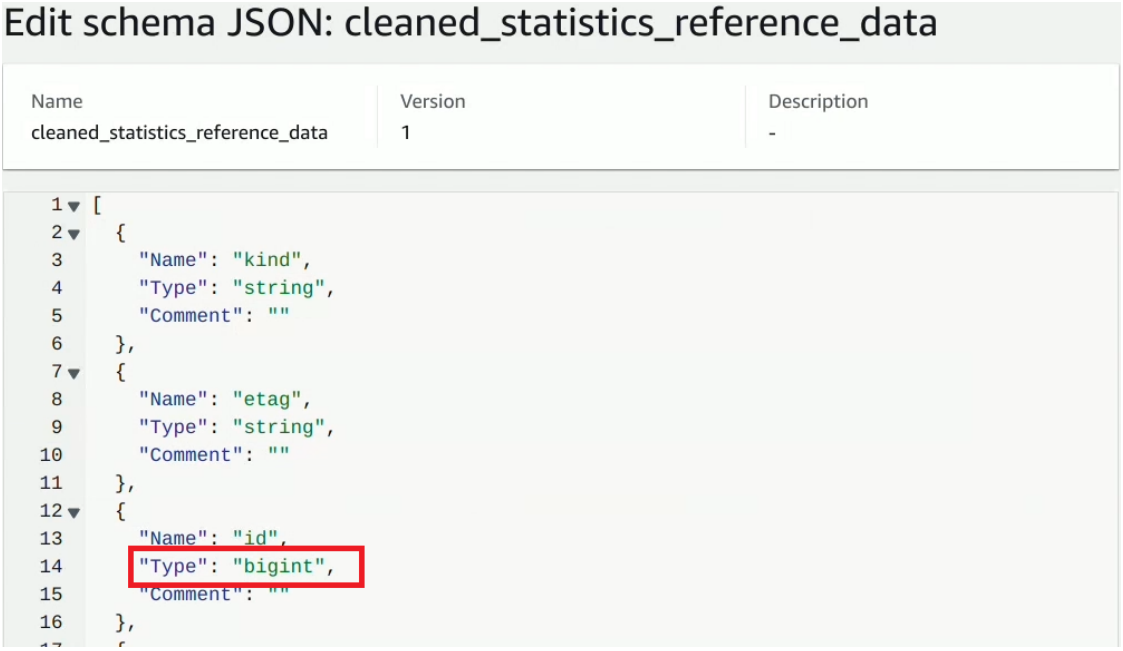
- Select **de-on-yt-cleaned** database and run to check for errors
- Join **de-on-yt-cleaned** with **de-on-yt-raw** with query "***SELECT * FROM "de-on-yt-raw"."raw_statistics" a INNER JOIN "de-on-yt-cleaned"."cleaned_statistics_reference_data" b ON a.category_id=cast(b.id as int);***"



- In above query, type cast the id with int but we can change the schema from database table manually to avoid type casting for every new query
- Move to AWS Glue menu, then tables and select "**cleaned_statistics_reference_data**" table and edit schema for the ***id*** field from ***string*** type to ***bigint*** and remember after changing schema you must have to **run crawler** again.

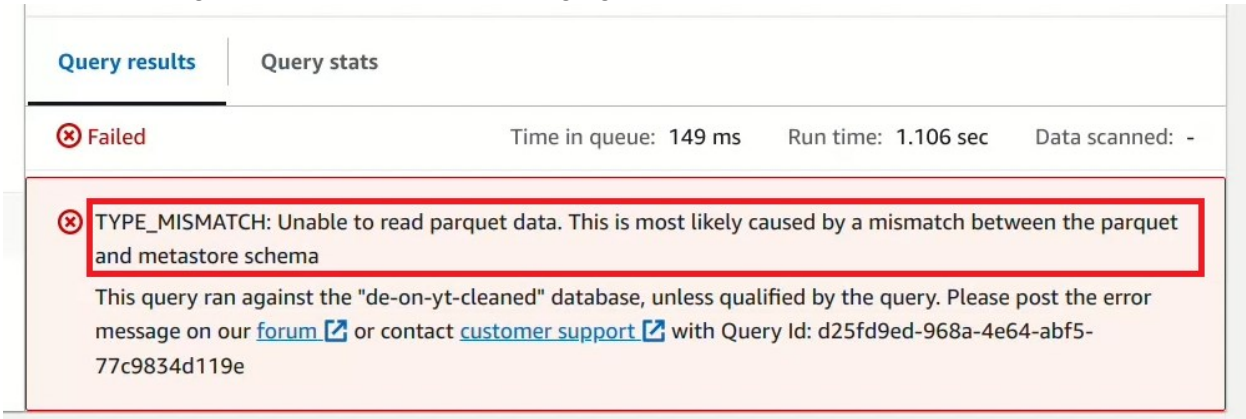






11) Resolve issue after changing schema

- You might face this error after changing schema, "**TYPE_MISMATCH**".



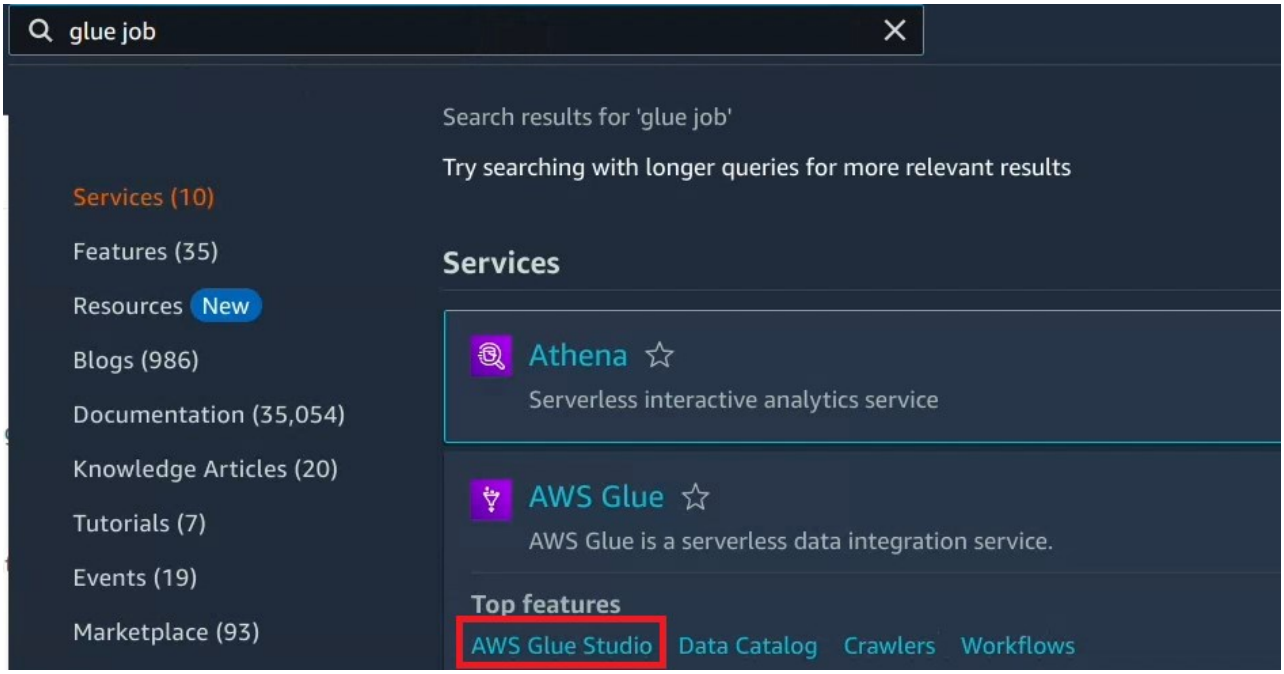
- The reason behind the error is **parquet** file already created from previous **lambda function** testing execution.
- To resolve this issue:
  - delete all parquet files stored in S3 buckets.
  - delete ***"cleaned_statistics_reference_data"*** table.
  - run lambda function ***test*** again if this don't resolve your issue the create another **testing event** and repeat the same steps as in **step 8**, and change the JSON file "**CA_category_id.json**"
  - run ***"de-on-yt-raw-useast-1-dev-athena-job"*** crawler again.
- You can select a **specific column** and then filter data for a **specific region**.
- Remove the *type casting* from query and run again.

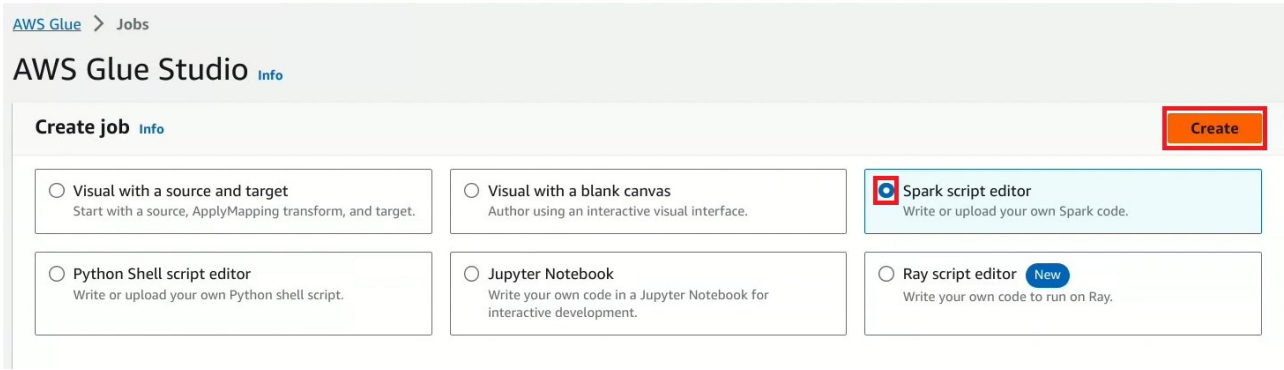
Step 08

AWS Glue Studio

12) Create a new Job in AWS Glue

- Search **Glue job**



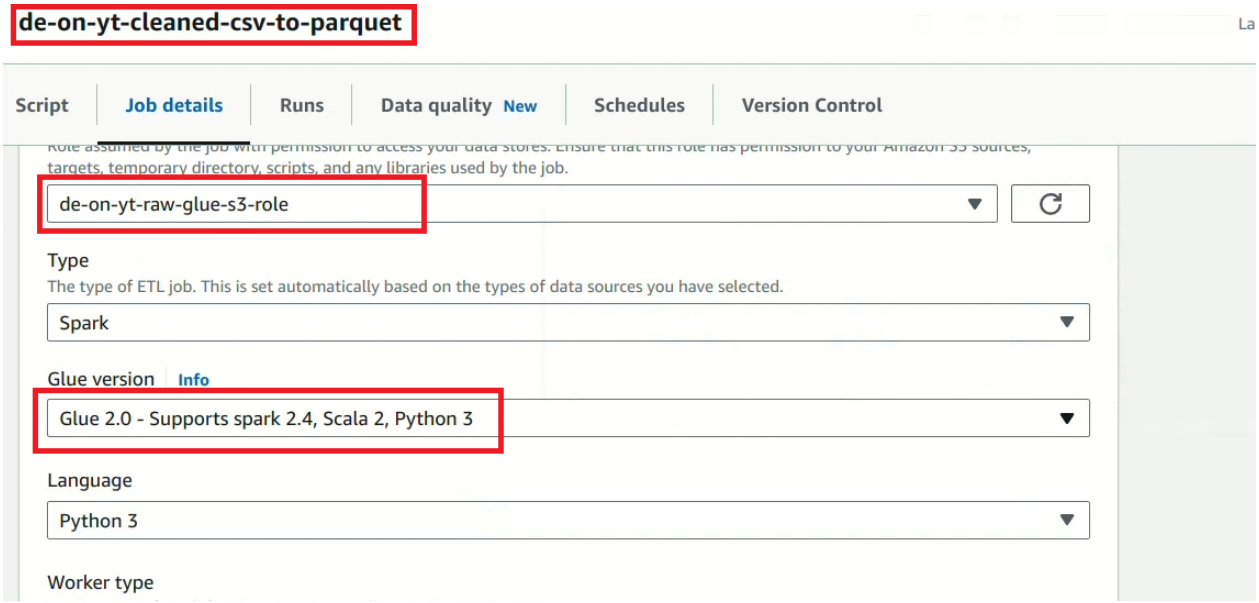


- Then copy the code from **GitHub** and paste into script editor
- Set source database at line no. **27** "**de-on-yt-raw**" and table "**raw_statistics**"

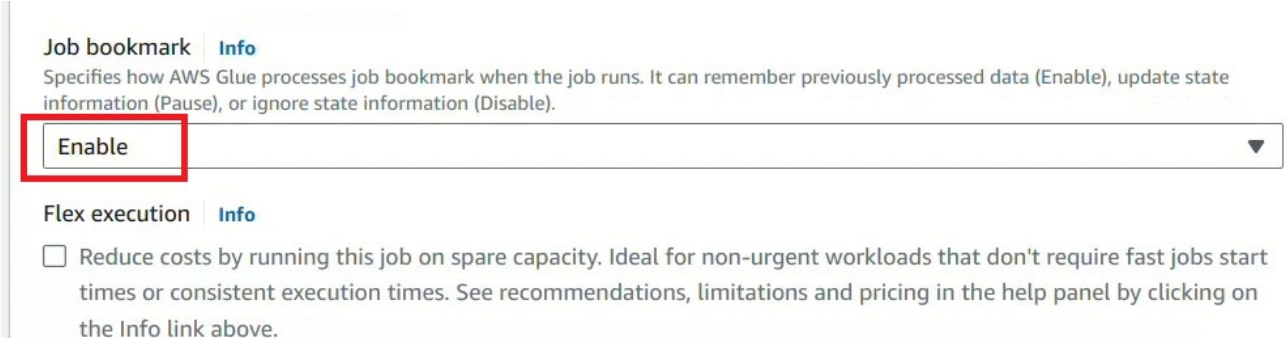
```
25  predicate_pushdown = "region in ('ca','gb','us')"
26
27  datasource0 = glueContext.create_dynamic_frame.from_catalog(database = "db_youtube_raw", table_name = "raw_statistics"
        push_down_predicate = predicate_pushdown)
28
29  ## @type: ApplyMapping
30  ## @args: [mapping = [("video_id", "string", "video_id", "string"), ("trending_date", "string", "trending_date", "stri
        ("channel_title", "string", "channel_title", "string"), ("category_id", "long", "category_id", "long"), ("publish_
        ("tags", "string", "tags", "string"), ("views", "long", "views", "long"), ("likes", "long", "likes", "long"), ("di
        ("comment_count", "long", "comment_count", "long"), ("thumbnail_link", "string", "thumbnail_link", "string"), ("co
        "comments_disabled", "boolean"), ("ratings_disabled", "boolean", "ratings_disabled", "boolean"), ("video_error_or_
```

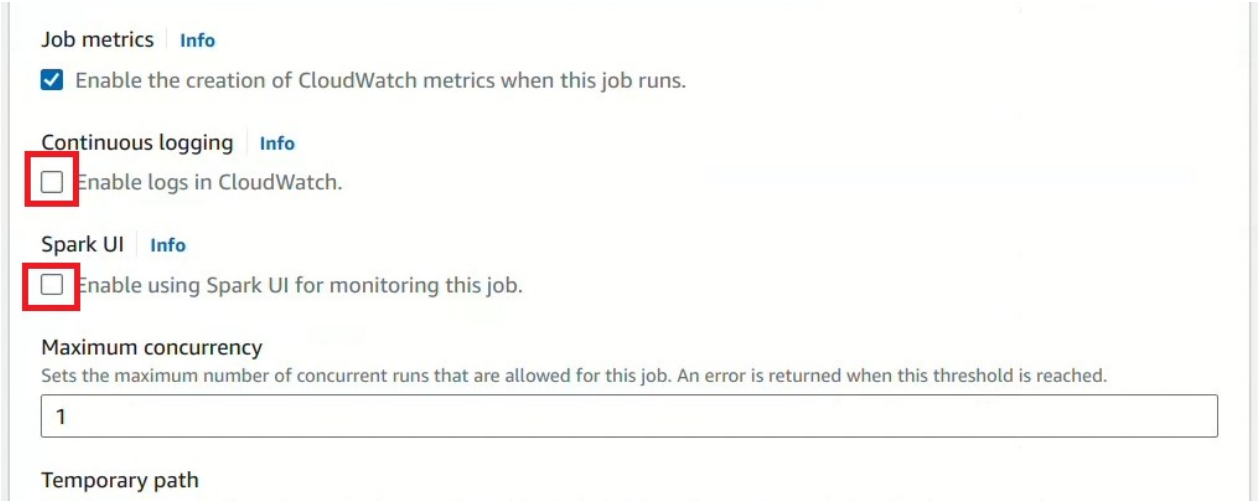
- Set target S3 bucket with path "**s3://de-on-yt-cleansed-useast-1-dev/youtube/raw_statistics/**" at line no. **54**

- Reason behind copying code, unavailability of same UI as in video and some transformation is also not available
- Give job name as ***"de-on-yt-cleaned-csv-to-parquet"***

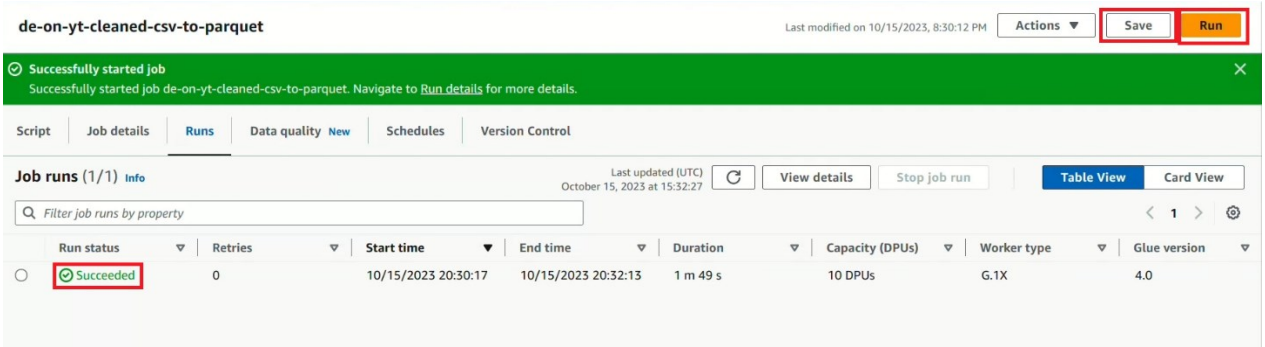


- Enable Job metrics and uncheck continuous logging and Spark UI.





- Save your job and run it and wait for the **succeeded** status.



13) Create another crawler

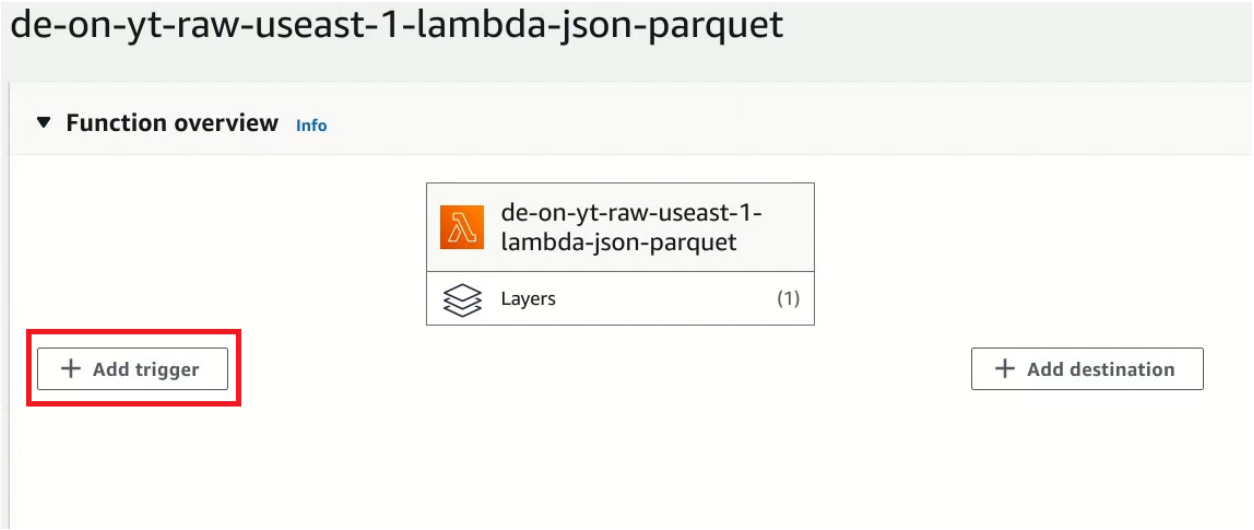
- Give naming convention "**de-on-yt-cleaned-csv-to-parquet-etl**"
- Select database source as newly created **"raw_statistics"** folder in **"de-on-yt-cleansed-useast-1-dev"** bucket
- Select IAM role "**de-on-yt-raw-glue-s3-role**"
- Select database "**de-on-yt-cleaned**"

STEP 09

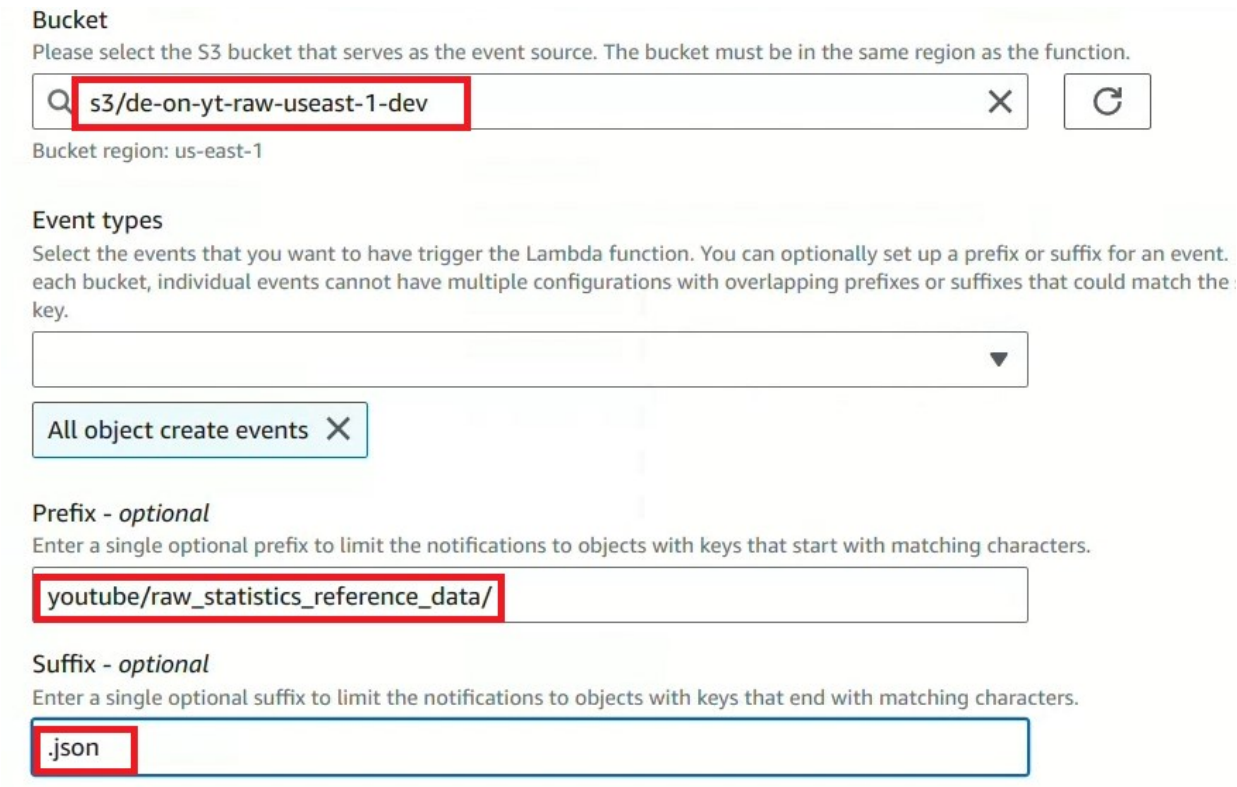
LAMDA TRIGGER

14) Update lambda function

- Add S3 trigger



- Set bucket to "**de-on-yt-raw-useast-1-dev**"
- Set prefix to "**youtube/raw_statistics_reference_data/**"
- Set suffix to **.json** then acknowledge terms and click **Add**.



- By add trigger every time new file upload in "**de-on-yt-raw-useast-1-dev/ youtube/raw_statistics_reference_data/**" bucket folder will automatically create "**.parquet**" file in ***"de-on-yt-cleansed-useast-1-dev/youtube/cleaned_statistics_reference_data"*** bucket folder.
- Delete all json files from the bucket, also parquet file generated for this json file in cleaned bucket
- Reupload all json files and rerun the newly created crawler to show parquet files in the cleaned bucket

15) Update the query in athena (optional)

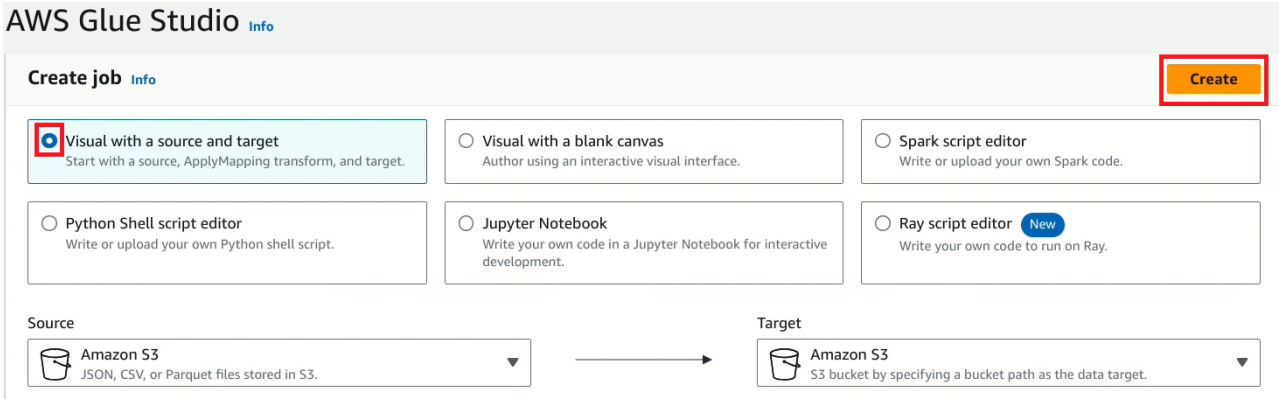
- Edit the query which is "**`SELECT * FROM "de-on-yt-cleaned"."raw_statistics" a INNER JOIN "de-on-yt-cleaned"."cleaned_statistics_reference_data" b ON a.category_id=b.id;`**"

STEP 10

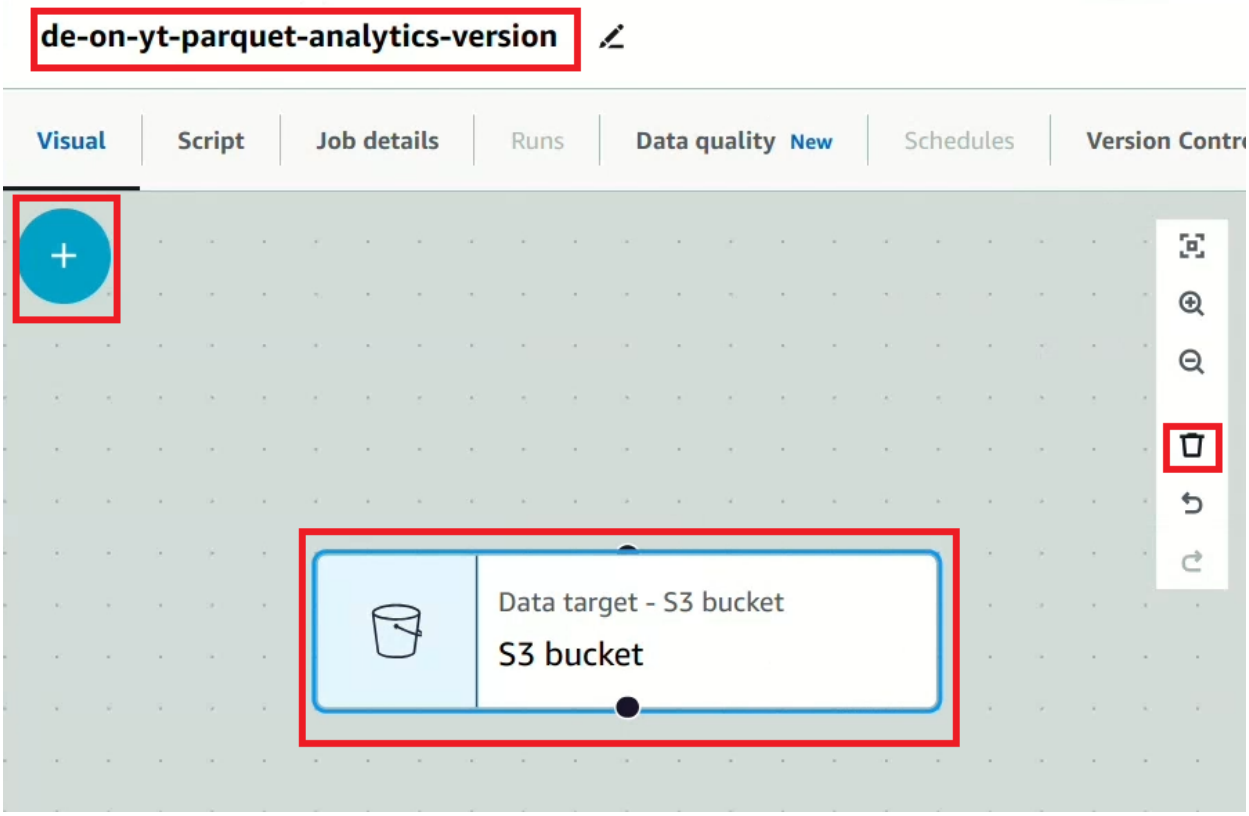
BUILD ETL

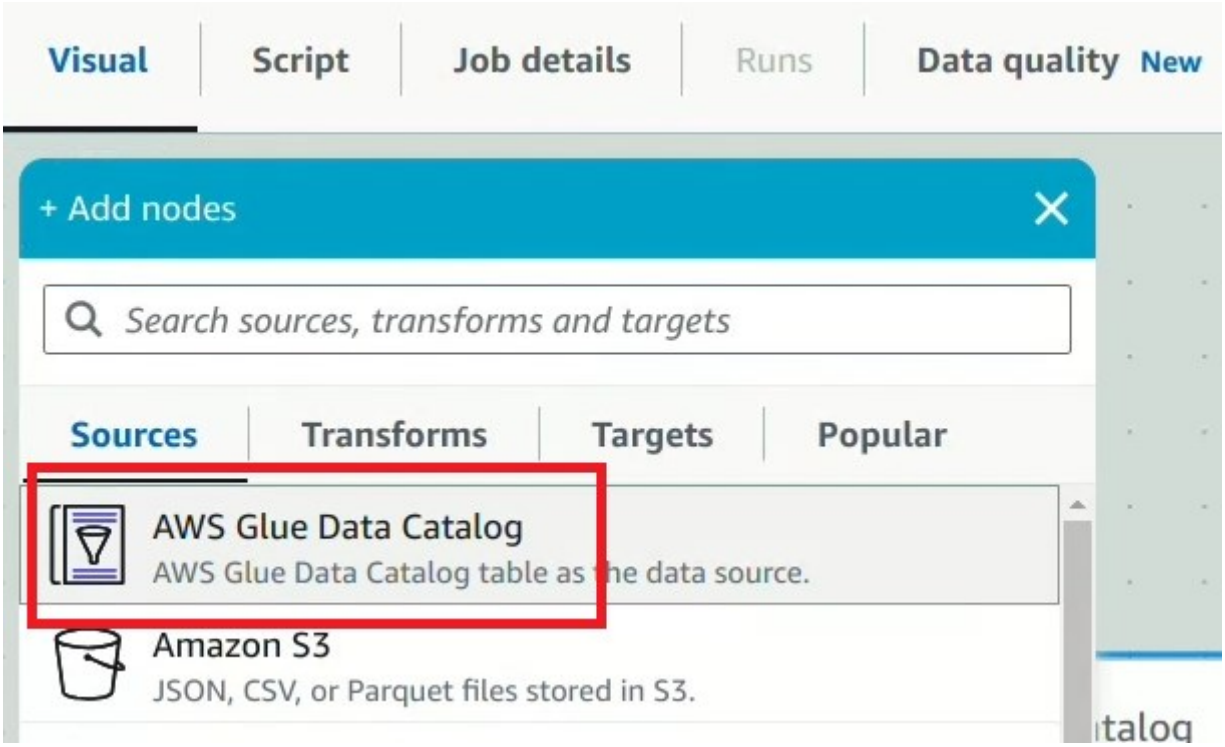
16) Build ETL for combined data of both buckets with ETL Job

- Create ETL job "**de-on-yt-parquet-analytics-version**"

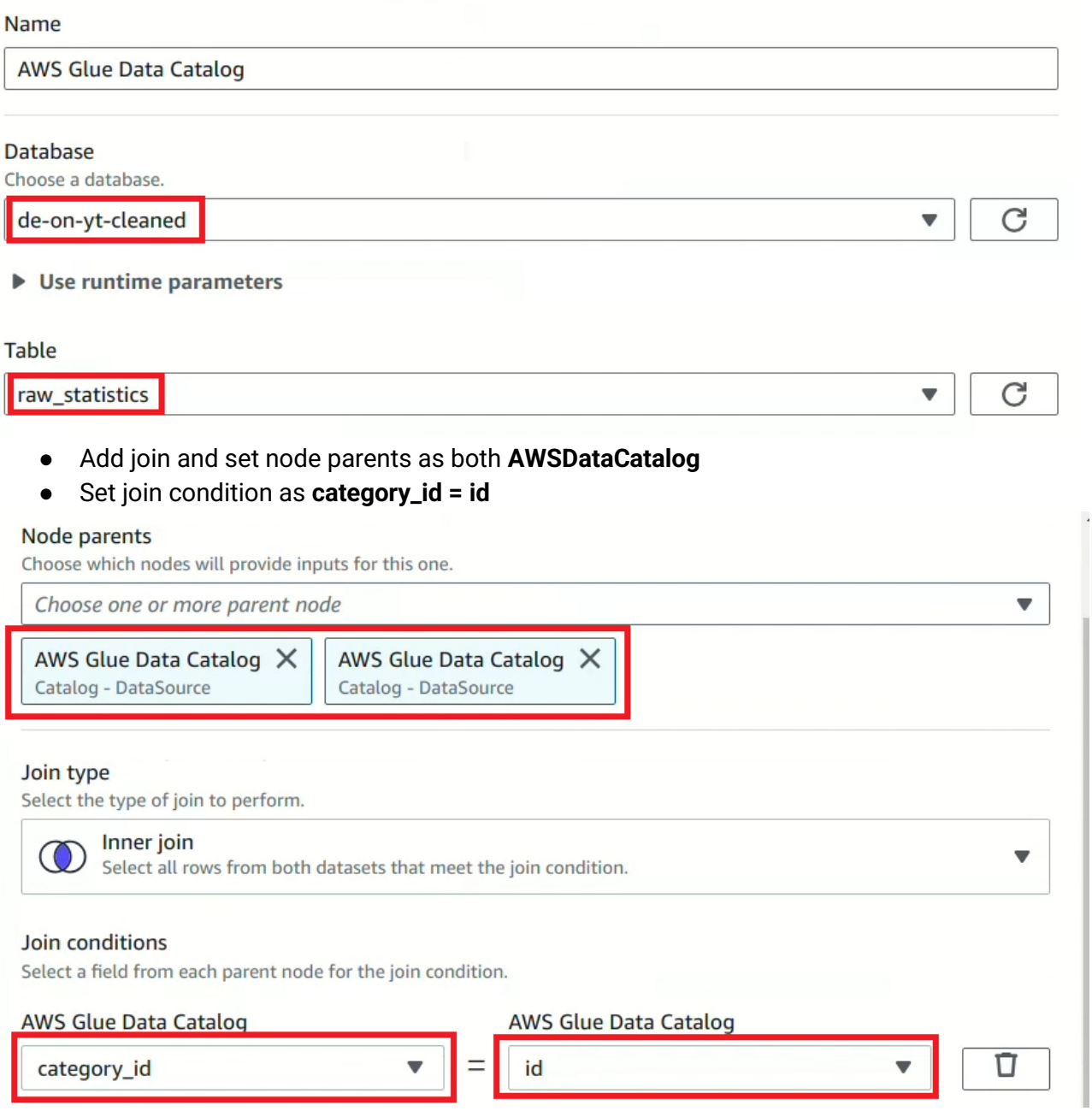


- Select S3 bucket and delete them from UI and click on + sign then Set two **AWSDataCatalog**



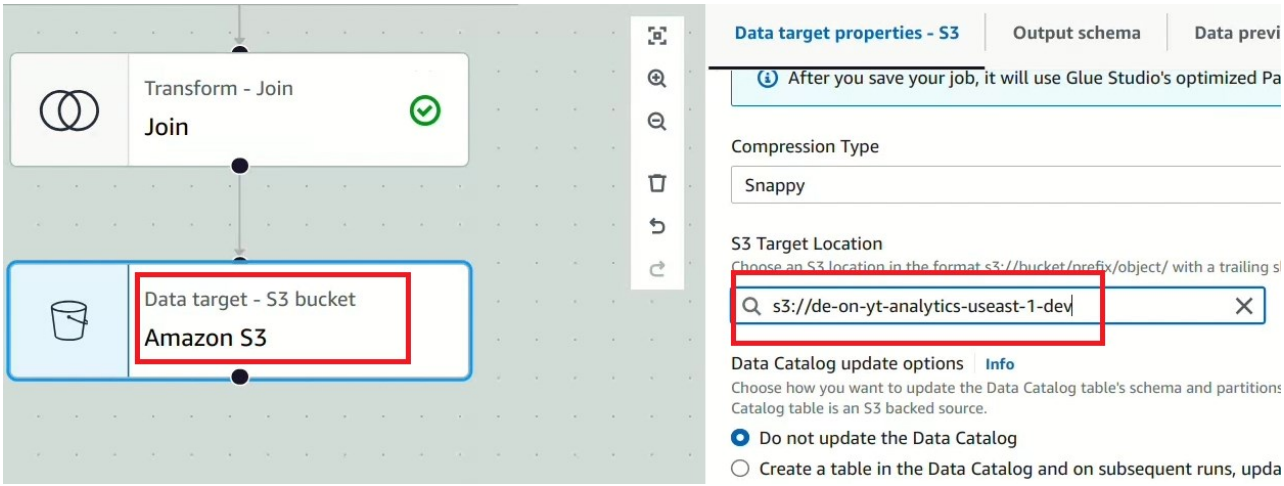


- Select database "**de-on-yt-cleaned**" for both catalog
- Change tables for both catalog, 1 for **cleaned_statistics_reference_data** and 2 for **raw_statistics**

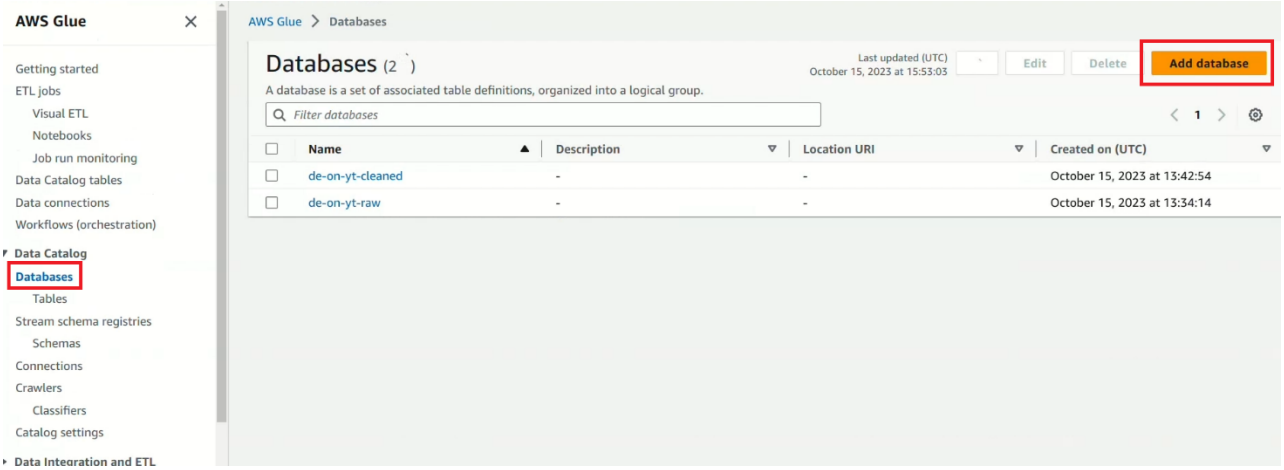


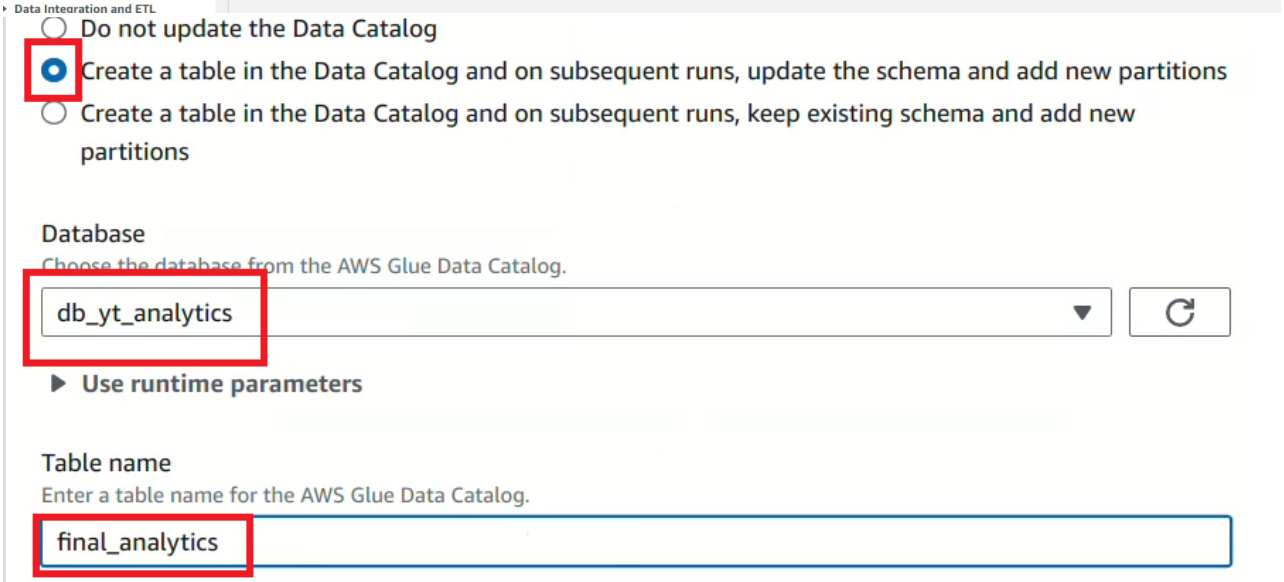
- Add join and set node parents as both **AWSDataCatalog**
- Set join condition as **category_id = id**

- Create another bucket “**de-on-yt-analytics-useast-1-dev**”
- Set target S3 bucket, choose path with was created in recently the **analytics** one, choose format as **parquet** and compression type **Snappy**

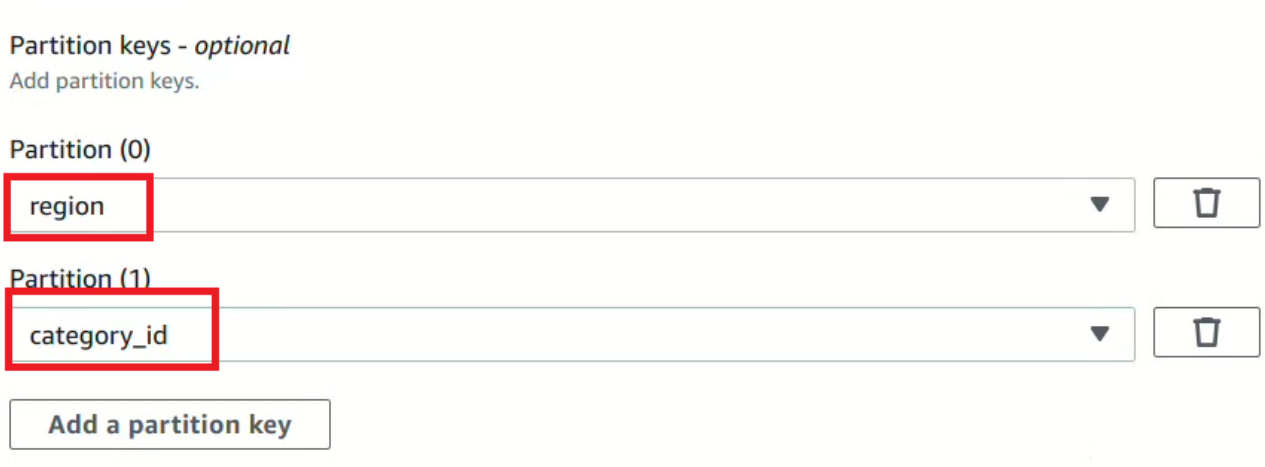


- Also create a new database with name **db_yt_analytics,** then choose from drop down and table name **final_analytics**





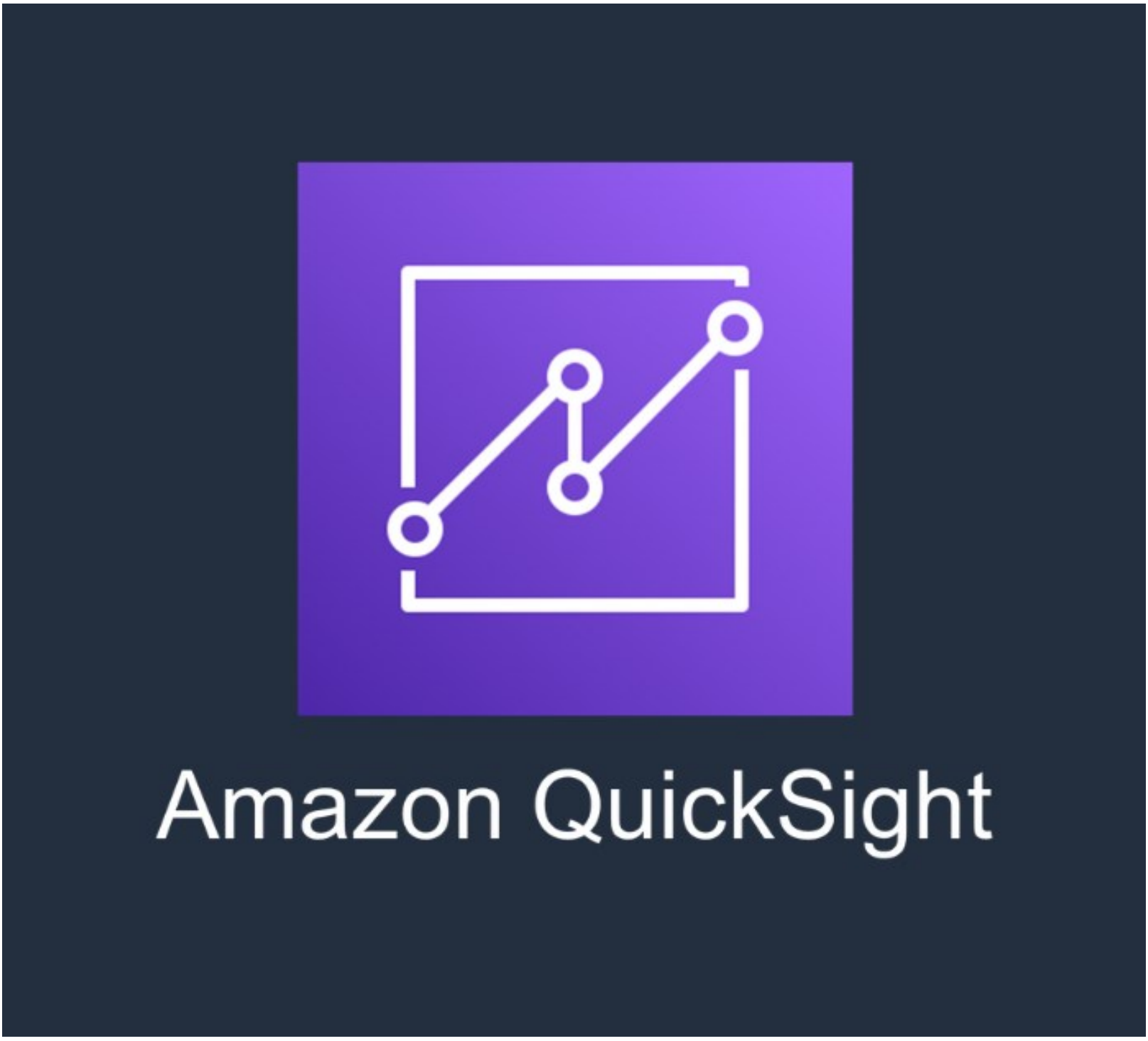
- Set partition keys as **region** and **category_id**



- Set job details with **IAM role** to ***"de-on-yt-raw-glue-s3-role"*** then enable **Job metrics** and uncheck **continuous logging** and **Spark UI,** save your job and run it and wait for the **succeeded** status.
- After run it successfully the parquet files are created in the **analytics** bucket

Step 11

AWS QuickSight



17) Setup QuickSight Account

- Click on signup
- Choose standard from bottom
- Give unique username and email
- Select the services S3 and more services if needed or you can set it up by manage the quicksight after setup the account later
- Go to Amazon QuickSight after finishing the account setup

18) Create Dataset

- Choose dataset from menu
- New dataset
- Choose athena, give name "**yt_analytics_dashboard**" and **validate connection**
- Create data source, choose database "**db_yt_analytics**" inside this choose table "**final_analytics**"

19) Create new analysis

- Create a bar chart horizontally for snippet title with the sum of likes
- Create a pie chart for snippet title with sum of views
- Create a pie chart for region with the num of users in each region
- Create KPI for average no of comments and total no of likes
- Can create dashboard publish link
- Can export in the form of pdf
- Here is the output pdf

THANK YOU

FOR FOLLOW UP

Keep pushing your skills and passion for data engineering. More projects await, leading you to success in your career!

ZUBAIR

To Connect

engrzubairkhatti@gmail.com

LinkedIn