



# ZUBAIR

S O F T W A R E   E N G I N E E R



## YOUTUBE DATA ANALYSIS – END TO END DATA ENGINEERING PROJECT

---

e n g r z u b a i r k h a t t i @ g m a i l . c o m

---

L I N K E D I N

---

G I T H U B

# Table of Contents

<b>Create IAM User.....</b>	<b>3</b>
Steps for project.....	4
1) Create IAM User.....	4
<b>Download AWS CLI .....</b>	<b>8</b>
2) Download AWS CLI.....	9
<b>S3 Bucket.....</b>	<b>11</b>
3) Create S3 Bucket .....	12
4) Copy data to S3.....	12
<b>AWS Glue.....</b>	<b>13</b>
5) Create Glue Crawler.....	14
<b>AWS Athena.....</b>	<b>18</b>
6) Run Crawler and View Data Using Athena .....	19
<b>AWS Lamda .....</b>	<b>21</b>
7) Create the Lambda Function .....	22
8) Adding preprocessing code and assigning environment variables .....	23
<b>Preprocess The Data.....</b>	<b>29</b>
9) Create Crawler for CSV files.....	30
10) Run SQL query over athena .....	31
11) Resolve issue after changing schema .....	33
<b>AWS Glue Studio .....</b>	<b>34</b>
12) Create a new Job in AWS Glue .....	35
13) Create another crawler .....	37
<b>Lamda Trigger.....</b>	<b>38</b>
14) Update lambda function.....	39
15) Update the query in athena (optional).....	40
<b>Build ETL.....</b>	<b>41</b>
16) Build ETL for combined data of both buckets with ETL Job .....	42
<b>AWS Quicksight.....</b>	<b>47</b>
17) Setup QuickSight Account.....	48
18) Create Dataset .....	48
19) Create new analysis .....	48

»» STEP 01

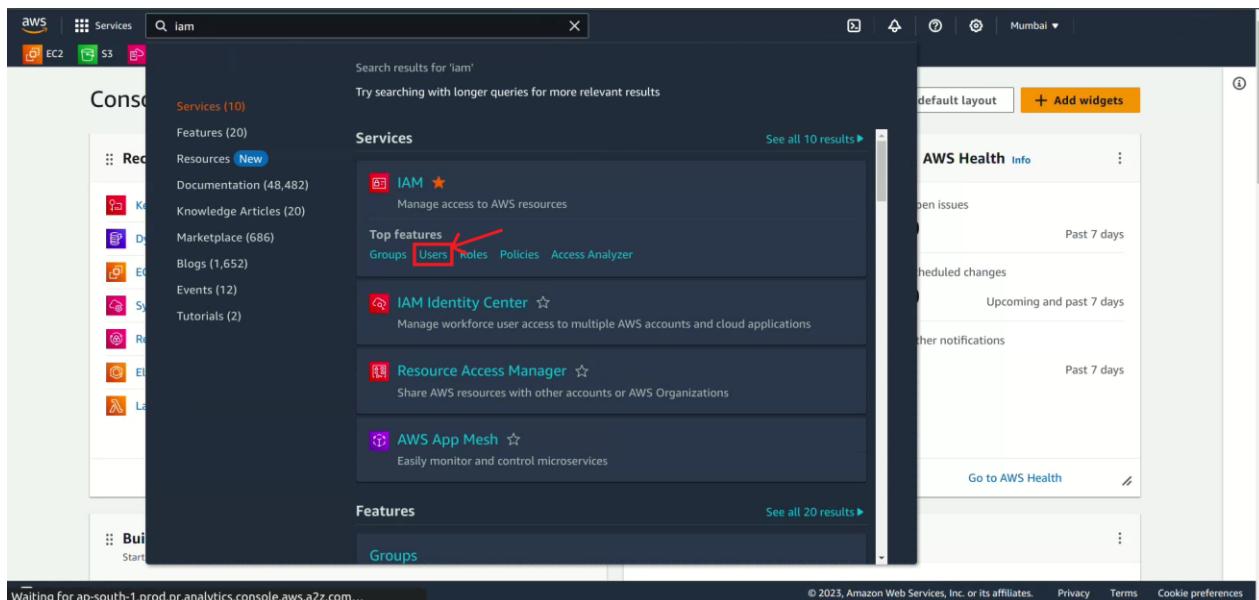
# CREATE IAM USER



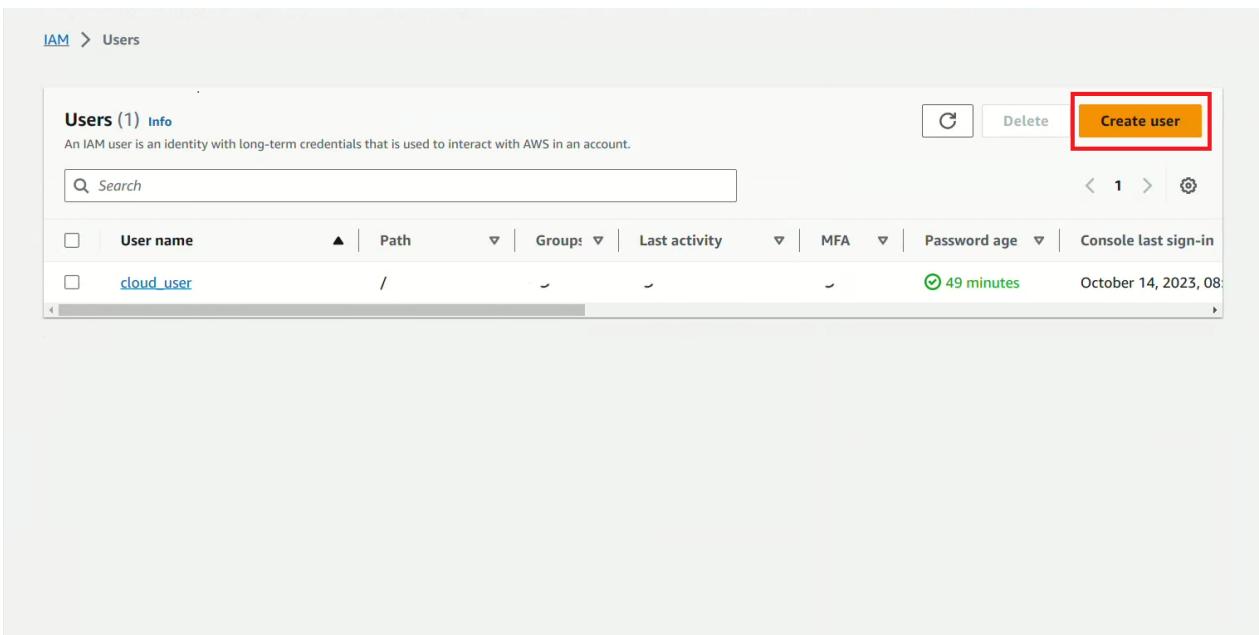
## Steps for project

### 1) Create IAM User

- Search IAM then click on user



- Click on create user.



- Give username and give custom password

IAM > Users > Create user

Step 1 Specify user details

Step 2 Set permissions

Step 3 Review and create

Step 4 Retrieve password

### Specify user details

**User details**

User name: zubair\_1  
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)

Provide user access to the AWS Management Console - optional  
If you're providing console access to a person, it's a best practice [to manage their access in IAM Identity Center.](#)

Console password:

- Autogenerated password  
You can view the password after you create the user.
- Custom password  
Enter a custom password for the user.

Show password

Users must create a new password at next sign-in - Recommended  
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

### User details

User name: zubair\_1  
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)

Provide user access to the AWS Management Console - optional  
If you're providing console access to a person, it's a best practice [to manage their access in IAM Identity Center.](#)

Console password:

- Autogenerated password  
You can view the password after you create the user.
- Custom password  
Enter a custom password for the user.

\*\*\*\*\*  
 Show password

Users must create a new password at next sign-in - Recommended  
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

[Cancel](#) [Next](#)

- Choose Attach Policy Directly and search

IAM > Users > Create user

Step 1 Specify user details

Step 2 Set permissions

Step 3 Review and create

### Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

#### Permissions options

- Add user to group  
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions  
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly  
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

#### Permissions policies (1134)

Choose one or more policies to attach to your new user.

Filter by Type: All types

Policy name	Type	Attached entities
AccessAnalyzerServiceRolePolicy	AWS managed	0

Filter by Type  
All types

 AdministratorAccess AWS managed - job function

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

User created successfully  
You can view and download the user's password and email instructions for signing in to the AWS Management Console.  
[View user](#)

IAM > Users > Create user

Step 1 [Specify user details](#)  
Step 2 [Set permissions](#)  
Step 3 [Review and create](#)  
Step 4 **Retrieve password**

Console sign-in details  
Console sign-in URL: <https://322270801872.signin.aws.amazon.com/console>  
User name: zubair\_1  
Console password:  

[Email sign-in instructions](#) [Download .csv file](#) [Return to users list](#)

**zubair\_1** [Info](#) [Delete](#)

**Summary**

ARN  arn:aws:iam::322270801872:user/zubair_1	Console access  Enabled without MFA	Access key 1 <a href="#">Create access key</a>
Created October 14, 2023, 08:37 (UTC+05:00)	Last console sign-in  Today	

IAM > Users > zubair\_1 > Create access key

Step 1 **Access key best practices & alternatives** [Info](#)  
Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and best practices:

Step 2 - optional  
Set description tag

Step 3  
Retrieve access keys

**Use case**

- Command Line Interface (CLI)**  
You plan to use this access key to enable the AWS CLI to access your AWS account.
- Local code**  
You plan to use this access key to enable application code in a local development environment to access your AWS account.
- Application running on an AWS compute service**  
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

Confirmation

I understand the above recommendation and want to proceed to create an access key.

Cancel **Next**

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

>Create access key

### Set description tag - optional Info

The description for this access key will be attached to this user as a tag and shown alongside the access key.

#### Description tag value

Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

access-key

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: \_ . : / = + - @

Cancel

Previous

**Create access key**

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

### Retrieve access keys Info

#### Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

#### Access key

#### Secret access key

AKIAUWCGWF7IFBMMKX4K

\*\*\*\*\* [Show](#)

#### Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

[Download .csv file](#)

**Done**

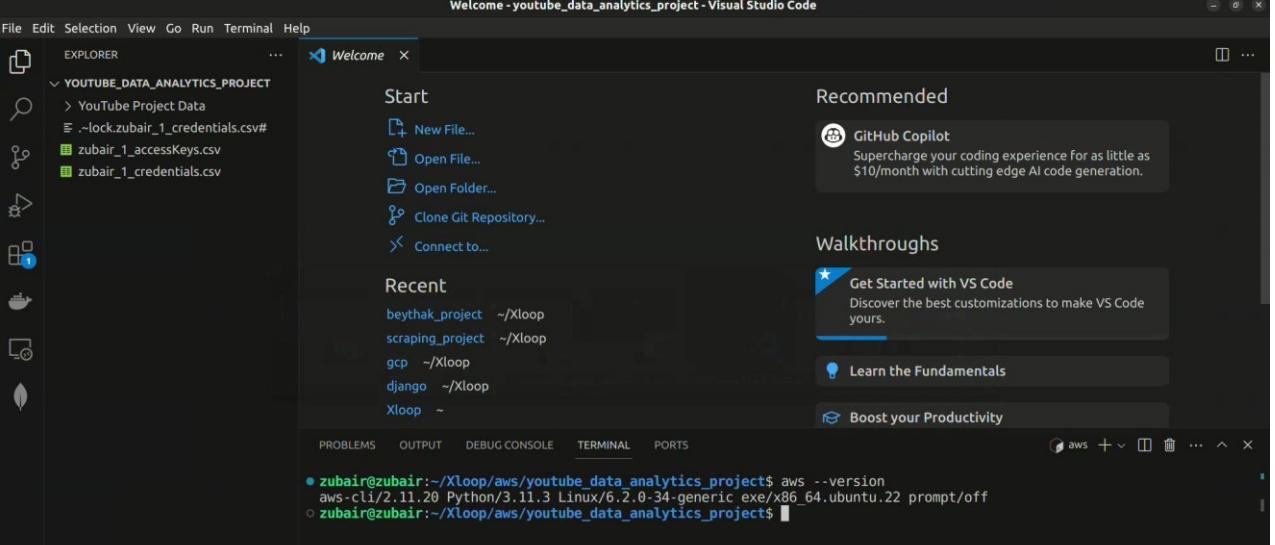
»» STEP 02

## DOWNLOAD AWS CLI



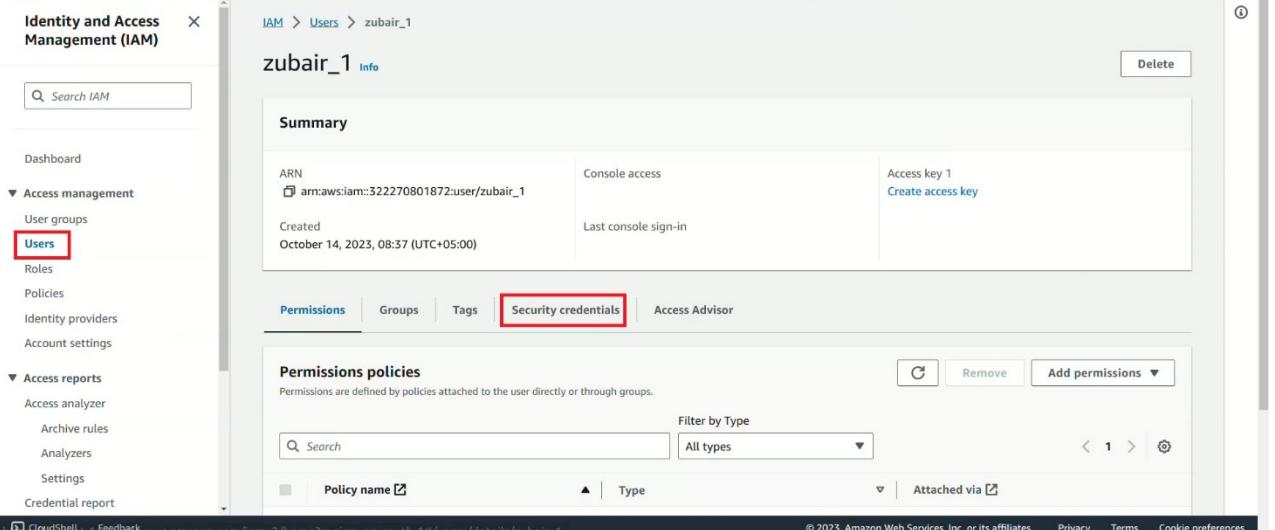
## 2) Download AWS CLI

- 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 --version” in terminal to check CLI is installed successfully.



The screenshot shows the Visual Studio Code interface with the title bar "Welcome - youtube\_data\_analytics\_project - Visual Studio Code". The left sidebar shows an "EXPLORER" view with a folder named "YOUTUBE\_DATA\_ANALYTICS\_PROJECT" containing files like ".lock.zubair\_1\_credentials.csv#", "zubair\_1\_accessKeys.csv", and "zubair\_1\_credentials.csv". The main area has sections for "Start" (New File..., Open File..., Open Folder..., Clone Git Repository..., Connect to...) and "Recommended" (GitHub Copilot). Below that is a "Recent" section listing projects: "beythak\_project", "scraping\_project", "gcp", "django", and "Xloop". On the right, there are "Walkthroughs" for "Get Started with VS Code", "Learn the Fundamentals", and "Boost your Productivity". At the bottom, the terminal tab is active with the command "aws --version" run, showing the output: "aws-cli/2.11.20 Python/3.11.3 Linux/6.2.0-34-generic exe/x86\_64 ubuntu.22 prompt/off".

- Again, redirect to newly created user menu and click on “Security credentials”



The screenshot shows the AWS IAM "Users" page. The left sidebar has a "Users" section highlighted with a red box. The main content shows the "zubair\_1" user details under the "Summary" tab. It includes ARN (arn:aws:iam::322270801872:user/zubair\_1), Console access (Last console sign-in: October 14, 2023, 08:37 (UTC+05:00)), and Access key 1 (Create access key). Below this, tabs for "Permissions", "Groups", "Tags", and "Security credentials" are present, with "Security credentials" highlighted with a red box. The "Permissions policies" section lists attached policies, with a search bar and filter options. The bottom of the page shows CloudShell, Feedback, and copyright information from 2023.

- Copy the console login link

The screenshot shows the AWS IAM Security credentials page. The top navigation bar includes tabs for Permissions, Groups, Tags, Security credentials (which is the active tab), and Access Advisor. Below the tabs, the heading "Console sign-in" is displayed. Under this heading, there is a "Console sign-in link" with a copy icon and the URL <https://322270801872signin.aws.amazon.com/console>. This URL is highlighted with a red box.

Console sign-in link

<https://322270801872signin.aws.amazon.com/console>

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

The screenshot shows a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is the active tab), and PORTS. The terminal output shows the AWS CLI version and a configuration command:

- zubair@zubair:~/Xloop/aws/youtube\_data\_analytics\_project\$ aws --version  
aws-cli/2.11.20 Python/3.11.3 Linux/6.2.0-34-generic exe/x86\_64.ubuntu.22 prompt/off
- zubair@zubair:~/Xloop/aws/youtube\_data\_analytics\_project\$ aws configure

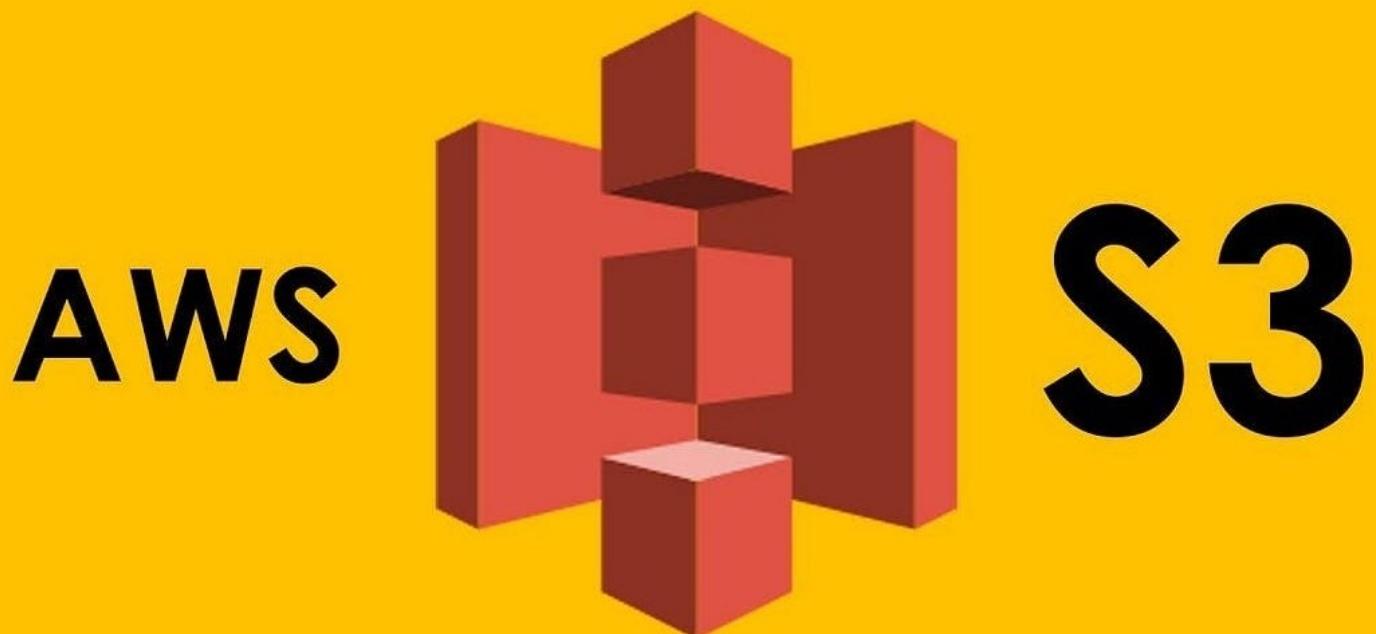
- Put “**access key**” credentials through terminal

The screenshot shows a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is the active tab), and PORTS. The terminal output shows the AWS configuration process:

- zubair@zubair:~/Xloop/aws/youtube\_data\_analytics\_project\$ aws --version  
aws-cli/2.11.20 Python/3.11.3 Linux/6.2.0-34-generic exe/x86\_64.ubuntu.22 prompt/off
- zubair@zubair:~/Xloop/aws/youtube\_data\_analytics\_project\$ aws config  
AWS Access Key ID [\*\*\*\*\*BEMY]: KX4K  
AWS Secret Access Key [\*\*\*\*\*ctRv]: /Iap04zUs  
Default region name [ap-south-1]:  
Default output format [None]:
- zubair@zubair:~/Xloop/aws/youtube\_data\_analytics\_project\$

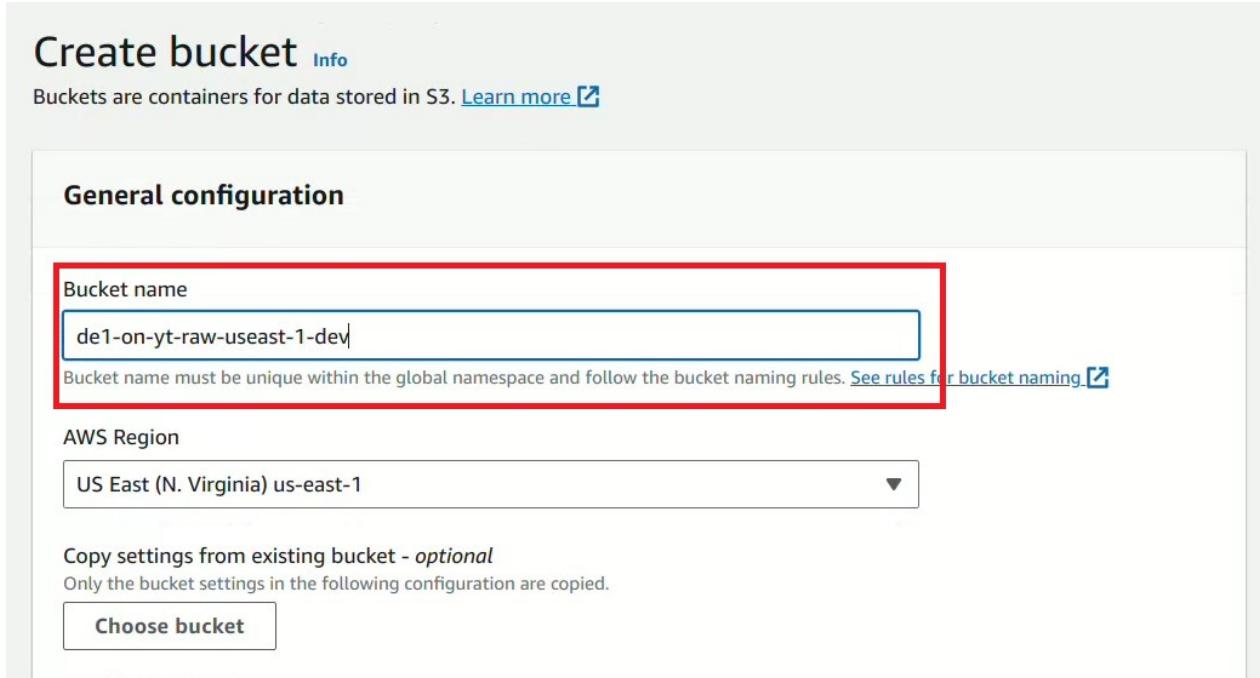
»» STEP 03

# 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

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● zubair@zubair:~/Xloop/aws/youtube_data_analytics_project$ cd YouTube\ Project\ Data/
○ zubair@zubair:~/Xloop/aws/youtube_data_analytics_project/YouTube Project Data$ aws s3 cp . s3://del-on-yt-raw-useast-1-dev/youtube/raw_statistics_reference_data/ --recursive --exclude "*" --include "*.json"
upload: ./JP_category_id.json to s3://del-on-yt-raw-useast-1-dev/youtube/raw_statistics_reference_data/JP_category_id.json
upload: ./GB_category_id.json to s3://del-on-yt-raw-useast-1-dev/youtube/raw_statistics_reference_data/GB_category_id.json
upload: ./RU_category_id.json to s3://del-on-yt-raw-useast-1-dev/youtube/raw_statistics_reference_data/RU_category_id.json
upload: ./CA_category_id.json to s3://del-on-yt-raw-useast-1-dev/youtube/raw_statistics_reference_data/CA_category_id.json
Completed 31.8 KiB/79.7 KiB (13.2 KiB/s) with 6 file(s) remaining
```

»» STEP 04

# AWS GLUE



## 5) Create Glue Crawler

- Search AWS Glue

The screenshot shows the AWS CloudSearch interface. At the top, there is a search bar with the text 'aws glue'. Below the search bar, there are several service categories: 'Amazon S3', 'Buckets', 'Services (156)', 'Features (349)', and 'Loading'. On the right side, there is a detailed result for 'AWS Glue' with a star icon. Below the result, there are links for 'Top features', 'AWS Glue Studio', 'Data Catalog', 'Crawlers' (which is highlighted with a red box), and 'Workflows'.

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

The screenshot shows the 'Add crawler' wizard at Step 2: 'Choose data sources and classifiers'. The left sidebar lists steps: Step 1 (Set crawler properties), Step 2 (Choose data sources and classifiers), Step 3 (Configure security settings), Step 4 (Set output and scheduling), and Step 5 (Review and create). The main area is titled 'Choose data sources and classifiers' and contains a 'Data source configuration' section. It asks if data is already mapped to Glue tables, with 'Not yet' selected. It also shows a 'Data sources (0)' section with an 'Add a data source' button (highlighted with a red box). A note below says 'You don't have any data sources.'

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

### S3 path

Browse for or enter an existing S3 path.

[View](#)[Browse S3](#)

All folders and files contained in the S3 path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

### S3 path

Browse for or enter an existing S3 path

 le1-on-yt-raw-useast-1-dev/youtube/ X[View](#)[Browse S3](#)

All folders and files contained in the S3 path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

### Subsequent crawler runs

This field is a global field that affects all S3 data sources.

**Crawl all sub-folders**

Crawl all folders again with every subsequent crawl.

**Crawl new sub-folders only**

Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are compatible, new partitions will be added to existing tables.

**Crawl based on events**

Rely on Amazon S3 events to control what folders to crawl.

**Sample only a subset of files**

**Exclude files matching pattern**

[Cancel](#)[Add an S3 data source](#)

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

### IAM resources

Resources in this AWS Account



User groups	Users	Roles	Policies	Identity providers
0	1	46	2	0

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

Select trusted entity SELECT TRUSTED ENTITY INFO

Step 2  
Add permissions

Step 3  
Name, review, and create

**Trusted entity type**

- AWS service**  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account**  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity**  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

**Use case**  
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case Glue

Choose a use case for the specified service.  
Use case  
 **Glue**

LoudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

**Add permissions INFO**

**Permissions policies (1/884) INFO**  
Choose one or more policies to attach to your new role.

Filter by Type All types 7 matches

Policy name	Type	Description
AmazonSageMakerServiceCatalogProd...	AWS managed	Service role policy used by the AWS Gl...
AWSGlueServiceNotebookRole	AWS managed	Policy for AWS Glue service role which ...
<input checked="" type="checkbox"/> <b>AWSGlueServiceRole</b>	AWS managed	Policy for AWS Glue service role which ...
AwsGlueSessionUserRestrictedNoteboo...	AWS managed	Provides permissions that allows users...

- 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**

AWS Glue > Crawlers > Add crawler

Step 1  
Set crawler properties

Step 2  
Choose data sources and classifiers

Step 3  
Configure security settings

Step 4  
Set output and scheduling

**Configure security settings**

**IAM role INFO**

Existing IAM role Choose an IAM role C

admin
<b>de-on-yt-raw-glue-s3-role</b>

Allows Glue to call AWS services on your behalf.

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

AWS Glue > Crawlers > Add crawler

Step 1 Set crawler properties

Step 2 Choose data sources and classifiers

Step 3 Configure security settings

Step 4

## Set output and scheduling

**Output configuration** Info

Target database

Choose a database Add database

Clear selection

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

AWS Glue > Crawlers > Add crawler

Step 1 Set crawler properties

Step 2 Choose data sources and classifiers

Step 3 Configure security settings

Step 4

## Set output and scheduling

**Output configuration** Info

Target database

Choose a database C

de-on-yt-raw

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

**One crawler successfully created**  
The following crawler is now created: "de-on-yt-raw-glue-catalog-1"

AWS Glue > Crawlers > de-on-yt-raw-glue-catalog-1

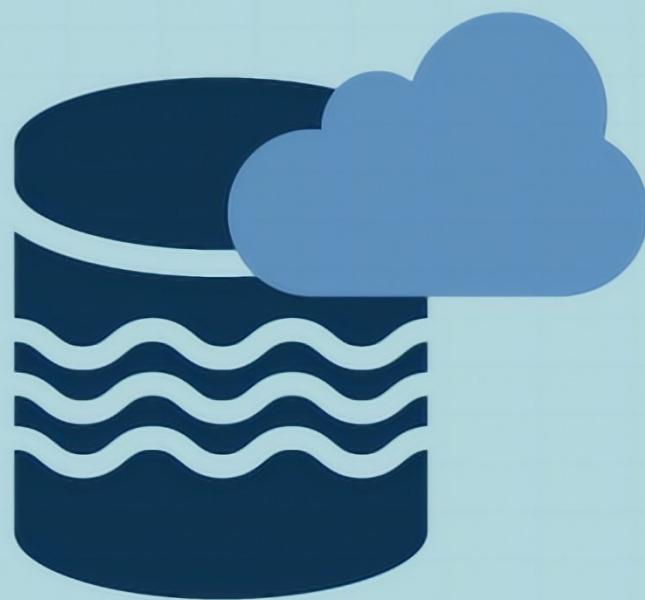
de-on-yt-raw-glue-catalog-1

Last updated (UTC) October 14, 2023 at 04:51:39 Run crawler Edit

Crawler properties

»» STEP 05

# AWS ATHENA



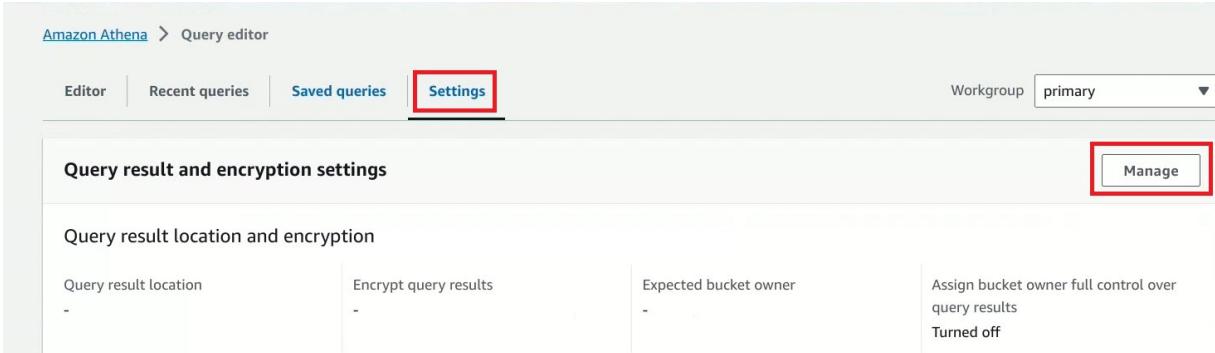
## 6) Run Crawler and View Data Using Athena

- After running crawler, search Athena



The screenshot shows the Amazon Athena start page. At the top left, there's a link to 'Analytics'. The main title 'Amazon Athena' is displayed in large, bold letters, followed by the tagline 'Start querying data instantly.' Below this, a brief description states: 'Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 and other federated data sources using standard SQL.' To the right, there's a 'Get started' section with two options: 'Query your data with Trino SQL' (selected, indicated by a blue dot) and 'Analyze your data using PySpark and Spark SQL'. A red box highlights the 'Launch query editor' button.

- Click on settings tab



The screenshot shows the 'Query result and encryption settings' page in the Amazon Athena settings. At the top, there are tabs for 'Editor', 'Recent queries', 'Saved queries' (highlighted with a red box), and 'Settings'. A 'Workgroup' dropdown is set to 'primary'. In the 'Query result and encryption settings' section, there are four fields: 'Query result location' (set to '-'), 'Encrypt query results' (set to '-'), 'Expected bucket owner' (set to '-'), and a note about assigning bucket owner control. A 'Manage' button is highlighted with a red box. The 'Query result location and encryption' section below contains a text input field with 's3://de1-on-xt-raw-useast-1-dev-athena-job' (highlighted with a red box), a 'View' button, and a 'Browse S3' button (both highlighted with red boxes). A callout box at the bottom left provides information about lifecycle rules for the bucket, with a 'Learn more' link.

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

### Query result location and encryption

Location of query result - *optional*

Enter an S3 prefix in the current region where the query result will be saved as an object.

X

View

Browse S3



You can create and manage lifecycle rules for this bucket

Use Amazon S3 lifecycle rules to store your query results and metadata cost effectively or to delete them after a period of time.

[Learn more](#)

Lifecycle configuration

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

The screenshot shows the AWS Athena console interface. On the left, there's a sidebar with 'Data source' set to 'AwsDataCatalog' and 'Database' set to 'de-on-yt-raw'. Below that is a section titled 'Tables and views' with a 'Create' button. Under 'Tables', there are two entries: 'raw\_statistics' (Partitioned) and 'raw\_statistics\_reference\_data'. To the right of the table list is a context menu with options like 'Run Query', 'Preview Table' (which is highlighted with a red box), 'Generate table DDL', 'Insert', 'Insert into editor', 'Manage', 'Delete table', 'View properties', and 'View in Glue' (with a red box around it). At the bottom of the menu are 'Run' and 'Explain' buttons. A red box also highlights the three-dot menu icon located between the table list and the menu options.

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

The screenshot shows the 'Query results' tab of the AWS Athena console. It displays a failed query with the following details:

- Status:** Failed
- Time in queue:** 154 ms
- Run time:** 310 ms
- Data scanned:** -

The error message is:

```

    ✘ HIVE_CURSOR_ERROR: Row is not a valid JSON Object - JSONException: A JSONObject text must end with '}' at 2 [character 3 line 1]

    This query ran against the "de-on-yt-raw" database, unless qualified by the query. Please post the error
    message on our forum or contact customer support with Query Id: 479a24ce-ec8b-4629-afc5-
    8125ec38cb0a
  
```

A red box highlights the entire error message area.

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

»» STEP 06

# AWS LAMDA



## 7) Create the Lambda Function

- Search lambda

The screenshot shows the AWS Lambda Functions list page. At the top right, there is a large orange button labeled "Create function". Below it, the page displays a table with one row of data. The columns are: "Function name" (with a checkbox), "Description" (dropdown), "Package type" (dropdown), "Runtime" (dropdown), and "Last modified" (timestamp). The single listed function is "cfst-1449-1508b3b79c1796bdf596318df28-InitFunction-VfRT9jtD8mPU". A search bar at the top is partially visible.

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

The screenshot shows the "Create function" wizard. In the "Basic information" step, the "Function name" field contains "de-on-yt-raw-useast-1-lambda-json-parquet". The "Runtime" dropdown is set to "Python 3.8". Under the "Execution role" section, the "Use an existing role" option is selected. Other options like "Create a new role with basic Lambda permissions" and "Create a new role from AWS policy templates" are also present but not selected.

- Create new role for lambda

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

### Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

Lambda

Choose a use case for the specified service.

#### Use case

Lambda

Allows Lambda functions to call AWS services on your behalf.

- Then give naming convention “**de-on-yt-raw-useast-1-s3-lambda-role**”
  - Choose existing role from lambda menu
- Use an existing role
- Create a new role from AWS policy templates

#### Existing role

Choose an existing role that you've created to be used with this Lambda function. The role

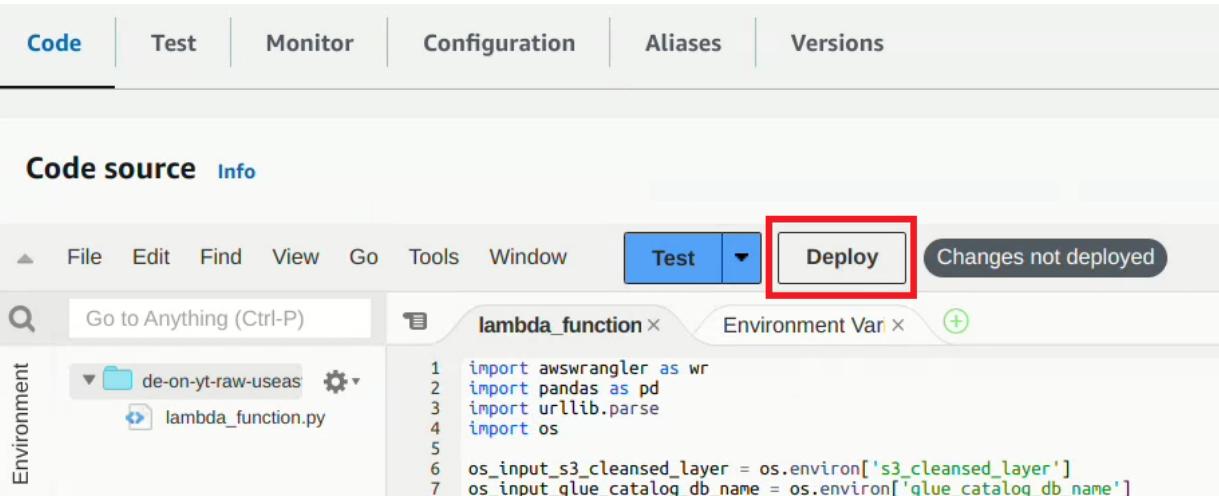
de-on-yt-raw-useast-1-s3-lambda-role

[View the de-on-yt-raw-useast-1-s3-lambda-role role](#)  on the IAM console.

- Create function

## 8) Adding preprocessing code and assigning environment variables

- Copy code from [GitHub](#) and paste in “**code tab**” of lambda function



Code    Test    Monitor    Configuration    Aliases    Versions

Code source [Info](#)

File Edit Find View Go Tools Window Test Deploy Changes not deployed

Go to Anything (Ctrl-P)

Environment **lambda\_function** Environment Var +

```

1 import awswrangler as wr
2 import pandas as pd
3 import urllib.parse
4 import os
5
6 os_input_s3_cleansed_layer = os.environ['s3_cleansed_layer']
7 os_input_glue_catalog_db_name = os.environ['glue_catalog_db_name']

```

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

The screenshot shows the AWS Lambda function configuration interface. The top navigation bar has tabs: Code, Test, Monitor, Configuration (which is highlighted with a red box), Aliases, and Versions. On the left, a sidebar lists General configuration, Triggers, Permissions, Destinations, Function URL, and Environment variables (which is also highlighted with a red box). The main content area shows General configuration details: Description (empty), Timeout (0 min 3 sec). The Environment variables section is empty.

- 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\_cleaned\_layer, s3://de-on-**yt**-cleaned-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**-cleaned-useast-1-dev**”

### Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

Key	Value	Remove
glue_catalog_db_name	de-on- <b>yt</b> -cleaned	<b>Remove</b>
glue_catalog_table_name	cleaned_statistics_reference_data	<b>Remove</b>
s3_cleaned_layer	s3://de-on- <b>yt</b> -cleaned-useast-1-dev/yo	<b>Remove</b>
write_data_operation	append	<b>Remove</b>

**Add environment variable**

- 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**

```

File Edit Find View Go Tools Window Test Deploy
Go to Anything (Ctrl-P) lambda_function Configure test event Ctrl-Shift-C
Environment de-on-yt-raw-useas lambda_function.py
9 os_input_write_data_operation = os.environ['WRITE_DATA_OPERATION']
10
11
12 def lambda_handler(event, context):
13     # Get the object from the event and show its content type
14     bucket = event['Records'][0]['s3']['bucket']['name']
15     key = urllib.parse.unquote_plus(event['Records'][0]['s3']['key'])
16     try:
17         # Creating DF from content
18

```

- 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.

The screenshot shows the AWS Lambda Test interface. The top navigation bar includes 'Go', 'Tools', 'Window', a red-bordered 'Test' button, and 'Deploy'. Below the navigation is a tab bar with 'lambda\_function.x' (selected), 'Environment Var x', and 'Execution result: x'. The 'Execution result' tab is expanded, showing a 'Test Event Name' of 's3\_bucket'. The 'Response' section contains an error message: 
 

```
{
  "errorMessage": "Unable to import module 'lambda_function': No module named 'awswrangler'",
  "errorType": "Runtime.ImportModuleError",
  "stackTrace": []
}
```

 A red box highlights this error message. The 'Function Logs' section shows the following log entries:
 

```
START RequestId: f74d746e-6ce4-4928-a39f-bd311c943c1a Version: $LATEST
[ERROR] Runtime.ImportModuleError: Unable to import module 'lambda_function': No module named 'awswrangler'
Traceback (most recent call last):END RequestId: f74d746e-6ce4-4928-a39f-bd311c943c1a
```

- 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

The screenshot shows the 'Layers' configuration page. At the top right are 'Edit' and 'Add a layer' buttons, with 'Add a layer' highlighted by a red box. Below is a table with columns: Merge order, Name, Layer version, Compatible runtimes, Compatible architectures, and Version ARN. A note says 'There is no data to display.'

## Function runtime settings

Runtime	Architecture
Python 3.8	x86_64

## Choose a layer

### Layer source [Info](#)

Choose from layers with a compatible runtime and instruction set architecture or specify the Amazon Resource Name (ARN) of a layer version. You can also [create a new layer](#).

AWS layers

Choose a layer from a list of layers provided by AWS.

Custom layers

Choose a layer from a list of layers created by your AWS account or organization.

Specify an ARN

Specify a layer by providing the ARN.

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.

#### Specify an ARN

Specify a layer by providing the Amazon Resource Name (ARN).

arn:aws:lambda:us-east-1:336392948345:layer:AWSDataWrangler-Python38:1

Verify

#### Description

AWS Data Wrangler Lambda Layer - 2.12.0 (Python 3.8)

#### Compatible runtimes

Python 3.8

#### Compatible architectures

-

Cancel

Add

Configuration Aliases Versions

#### General configuration

Edit

##### Description

-

##### Memory

128 MB

##### Ephemeral storage

512 MB

##### Timeout

3 min 0 sec

Edit

##### Execution role

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

Use an existing role

Create a new role from AWS policy templates

##### Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

de-on-yt-raw-useast-1-s3-lambda-role



View the de-on-yt-raw-useast-1-s3-lambda-role role [on the IAM console](#).

Cancel

Save

- Run test and response should be:

```
{  
  "paths": [  
    "s3://de1-on-yt-cleanse-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

Crawlers

A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawlers (1) <a href="#">Info</a>		Last updated (UTC)	Action	Run	<a href="#">Create crawler</a>
		October 14, 2023 at 05:54:36			
<input type="checkbox"/> Name		State	Schedule	Last run	Last run timestamp
<input type="checkbox"/> de-on-yt-raw-glue...					October 14, 2023 ...
				<a href="#">View log</a>	1 updated
<a href="#">Filter crawlers</a> < 1 >					

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

Location of S3 data

In this account  
 In a different account

S3 path

Browse for or enter an existing S3 path.

All folders and files contained in the S3 path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Subsequent crawler runs

This field is a global field that affects all S3 data sources.

- 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**.

**Step 2: Choose data sources and classifiers**

Type	Data source	Parameters
S3	s3://de1-on-yt-raw-useast-1-dev/youtu...	Recrawl all

**Step 3: Configure security settings**

IAM role	Security configuration	Lake Formation configuration
de-on-yt-raw-glue-s3-role	-	-

**Step 4: Set output and scheduling**

Database	Table prefix - optional	Maximum table threshold - optional	Schedule
de-on-yt-raw	-	-	On demand

Cancel Previous **Create 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);`"

The screenshot shows the AWS Athena results page. At the top, there are buttons for 'Run again', 'Explain', 'Cancel', 'Clear', 'Create', and a 'Reuse query results' checkbox. Below this, tabs for 'Query results' and 'Query stats' are visible, with 'Query results' being active. A status bar indicates the query is 'Completed' with a time in queue of 160 ms, a run time of 7.55 sec, and data scanned of 514.17 MB. The results section displays 326,873 rows. A search bar at the top of the results table allows for filtering. The table has columns: '#', 'video\_id', 'trending\_date', and 'title'. The first few rows of data are:

#	video_id	trending_date	title
1	5ugKfHgsmYw	18.07.02	陸自ヘリ、垂直に落下＝路上の車が撮影
2	ohObafdd34Y	18.07.02	イッテQ お祭り男宮川×手越 巨大ブランコ②
3	aBr2kKAHN6M	18.07.02	Live Views of Starman
4	5wNnwChvmsQ	18.07.02	東京ディズニーリゾートの元キャストが暴露した秘密5選
5	B7J47qFvdsk	18.07.02	榮倉奈々、衝撃の死んだふり！映画『家に帰ると妻が必ず死ん
6	QINPfGEdpRo	18.07.02	右翼さつきさんの死因を元手・和田君が全員前に進み、一同涙

- 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.

The screenshot shows the AWS Glue Data Catalog Tables interface. On the left sidebar, under 'Data Catalog tables', 'Tables' is selected. The main area displays a table named 'cleaned\_statistics\_refer' with the following details:

Name	Database	Location	Classification
cleaned_statistics_refer	de-on-yt-cleaned	s3://de1-on-yt-cleanse	Parquet
raw_statistics	de-on-yt-raw	s3://de1-on-yt-raw-use	CSV
raw_statistics_reference	de-on-yt-raw	s3://de1-on-yt-raw-use	JSON

Below this, the 'Schema (6)' section shows the table columns:

#	Column name	Data type	Partition key	Comment
1	kind	string	-	-
2	etag	string	-	-
3	id	string	-	-
4	snippet_channel_id	string	-	-
5	snippet_title	string	-	-
6	snippet_assignable	boolean	-	-

A red box highlights the 'Edit schema as JSON' button. Below the table, a JSON editor shows the schema definition:

```

1 [ 
2 { 
3   "Name": "kind",
4   "Type": "string",
5   "Comment": ""
6 },
7 { 
8   "Name": "etag",
9   "Type": "string",
10  "Comment": ""
11 },
12 { 
13   "Name": "id",
14   "Type": "bigint", // This line is highlighted with a red box
15   "Comment": ""
16 },
17 ]

```

## 11) Resolve issue after changing schema

- You might face this error after changing schema, “**TYPE\_MISMATCH**”.

The screenshot shows the 'Query results' tab selected in the top navigation bar. Below it, a red box highlights the error message: 'TYPE\_MISMATCH: Unable to read parquet data. This is most likely caused by a mismatch between the parquet and metastore schema'. The message also states: 'This query ran against the "de-on-xt-cleaned" database, unless qualified by the query. Please post the error message on our [forum](#) or contact [customer support](#) with Query Id: d25fd9ed-968a-4e64-abf5-77c9834d119e'.

- 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-xt-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.

The screenshot shows the 'Code' tab of the AWS Lambda function configuration interface. It displays the following code:

```
1 SELECT a.title, a.category_id, a.region FROM "de-on-xt-raw"."raw_statistics" a
2 INNER JOIN "de-on-xt-cleaned"."cleaned_statistics_reference_data" b ON a.category_id=b.id
3 where a.region = 'ca';
```

»» STEP 08

# AWS GLUE STUDIO



## 12) Create a new Job in AWS Glue

- Search **Glue job**

The screenshot shows the AWS search results for 'glue job'. On the left, there's a sidebar with categories like Services (10), Features (35), Resources (New), Blogs (986), Documentation (35,054), Knowledge Articles (20), Tutorials (7), Events (19), and Marketplace (93). The main area displays search results for 'Services'. The first result is 'Athena', followed by 'AWS Glue'. The 'AWS Glue' card is detailed with its description: 'AWS Glue is a serverless data integration service.' Below it, under 'Top features', are tabs for 'AWS Glue Studio' (which is highlighted with a red box), Data Catalog, Crawlers, and Workflows. At the bottom of the page, there's a breadcrumb navigation 'AWS Glue > Jobs' and a title 'AWS Glue Studio' with an 'Info' link. The 'Create' button at the top right of the 'AWS Glue Studio' section is also highlighted with a red box.

- 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"
28 push_down_predicate = predicate_pushdown)
29 ## @type: ApplyMapping
30 ## @args: [mapping = [("video_id", "string", "video_id", "string"), ("trending_date", "string", "trending_date", "string"),
("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-cleaned-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”**”

**de-on-yt-cleaned-csv-to-parquet**

Script    **Job details**    Runs    Data quality [New](#)    Schedules    Version Control

Role assumed by the job: **de-on-yt-raw-glue-s3-role**

Type: Spark

Glue version: **Glue 2.0 - Supports spark 2.4, Scala 2, Python 3**

Language: Python 3

Worker type:

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

**Job bookmark** [Info](#)

Specifies how AWS Glue processes job bookmark when the job runs. It can remember previously processed data (Enable), update state information (Pause), or ignore state information (Disable).

**Enable**

**Flex execution** [Info](#)

Reduce costs by running this job on spare capacity. Ideal for non-urgent workloads that don't require fast jobs start times or consistent execution times. See recommendations, limitations and pricing in the help panel by clicking on the Info link above.

Job metrics | [Info](#)

Enable the creation of CloudWatch metrics when this job runs.

Continuous logging | [Info](#)

Enable logs in CloudWatch.

Spark UI | [Info](#)

Enable using Spark UI for monitoring this job.

**Maximum concurrency**  
Sets the maximum number of concurrent runs that are allowed for this job. An error is returned when this threshold is reached.

1
---

Temporary path

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

**de-on-yt-cleaned-csv-to-parquet**

Last modified on 10/15/2023, 8:30:12 PM Actions ▾

Successfully started job  
Successfully started job de-on-yt-cleaned-csv-to-parquet. Navigate to [Run details](#) for more details.

Script | Job details | **Runs** | Data quality [New](#) | Schedules | Version Control

Job runs (1/1) [Info](#) Last updated (UTC) October 15, 2023 at 15:32:27

Run status	Retries	Start time	End time	Duration	Capacity (DPUs)	Worker type	Glue version
<input checked="" type="radio"/> <a href="#">Succeeded</a>	0	10/15/2023 20:30:17	10/15/2023 20:32:13	1 m 49 s	10 DPUs	G.1X	4.0

### 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

de-on-yt-raw-useast-1-lambda-json-parquet

Function overview [Info](#)

de-on-yt-raw-useast-1-lambda-json-parquet

Layers (1)

+ Add trigger      + Add destination

- 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**.

### Bucket

Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.

 X C

Bucket region: us-east-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. Each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same key.

All object create events X

### Prefix - optional

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

youtube/raw\_statistics\_reference\_data/

### Suffix - optional

Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

.json

- 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;`”

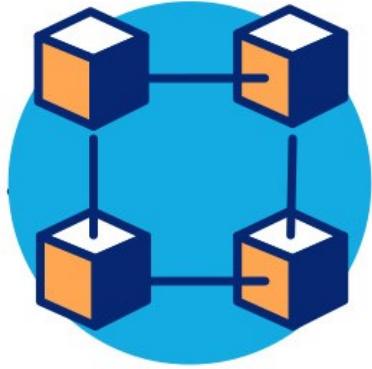
»» STEP 10

# BUILD ETL

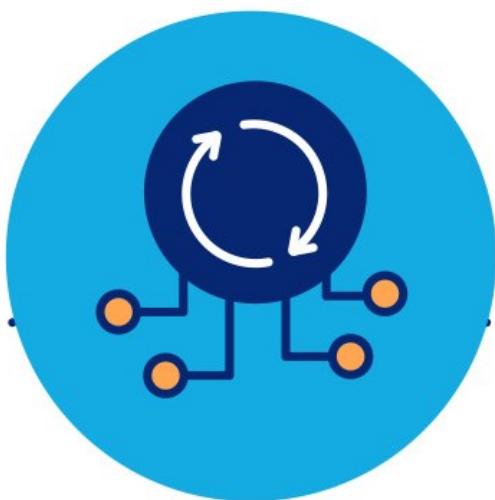
Extract



Load



Transform



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

- Create ETL job “de-on-yt-parquet-analytics-version”

AWS Glue Studio [Info](#)

Create job [Info](#) Create

Visual with a source and target  
Start with a source, ApplyMapping transform, and target.

Visual with a blank canvas  
Author using an interactive visual interface.

Spark script editor  
Write or upload your own Spark code.

Python Shell script editor  
Write or upload your own Python shell script.

Jupyter Notebook  
Write your own code in a Jupyter Notebook for interactive development.

Ray script editor [New](#)  
Write your own code to run on Ray.

Source Amazon S3  
JSON, CSV, or Parquet files stored in S3. → Target Amazon S3  
S3 bucket by specifying a bucket path as the data target.

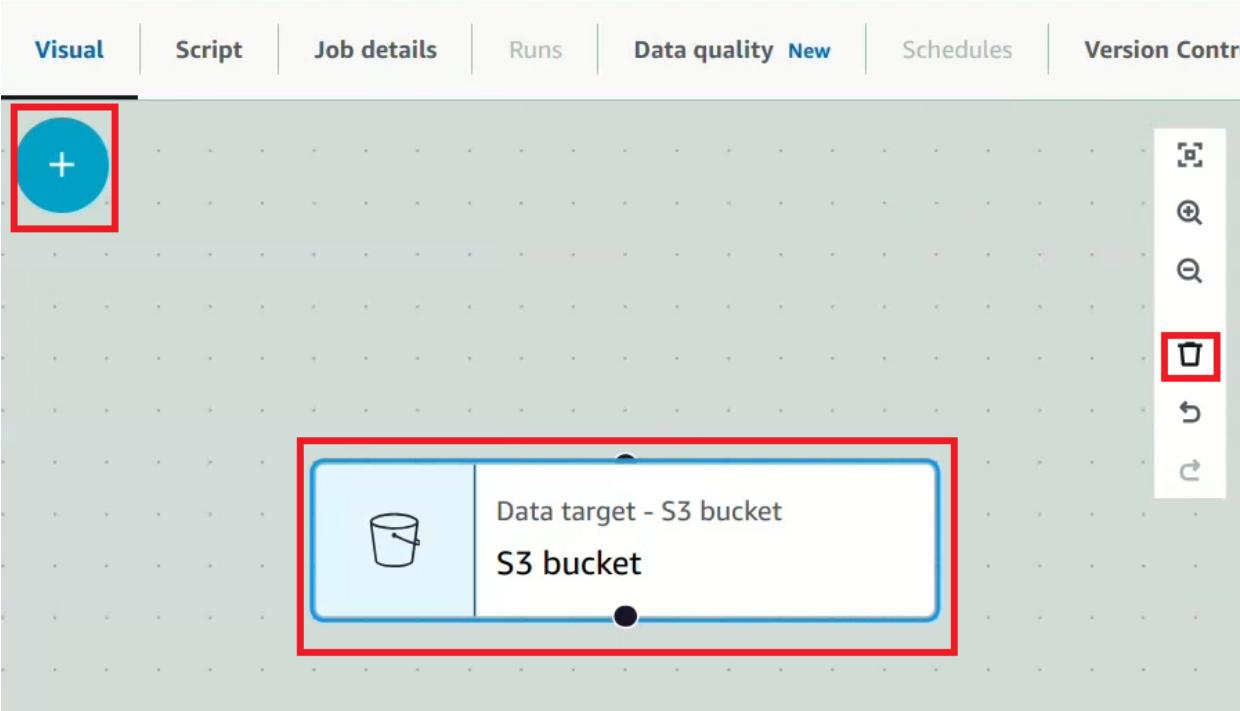
• Select S3 bucket and delete them from UI and click on + sign then Set two **AWSDataCatalog** de-on-yt-parquet-analytics-version Λ

Visual Script Job details Runs Data quality [New](#) Schedules Version Control

+

Data target - S3 bucket  
S3 bucket

Λ Λ Λ Λ Λ Λ



Visual | Script | Job details | Runs | Data quality New

+ Add nodes X

Search sources, transforms and targets

Sources | Transforms | Targets | Popular

AWS Glue Data Catalog AWS Glue Data Catalog table as the data source.

Amazon S3 JSON, CSV, or Parquet files stored in S3.

- 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**

Name

AWS Glue Data Catalog

Database

Choose a database.

de-on-yt-cleaned

► Use runtime parameters

Table

cleaned\_statistics\_reference\_data

Name

Database

Choose a database.

 C

**► Use runtime parameters**

Table

 C

- Add join and set node parents as both **AWSDataCatalog**
- Set join condition as **category\_id = id**

Node parents

Choose which nodes will provide inputs for this one.

Join type

Select the type of join to perform.

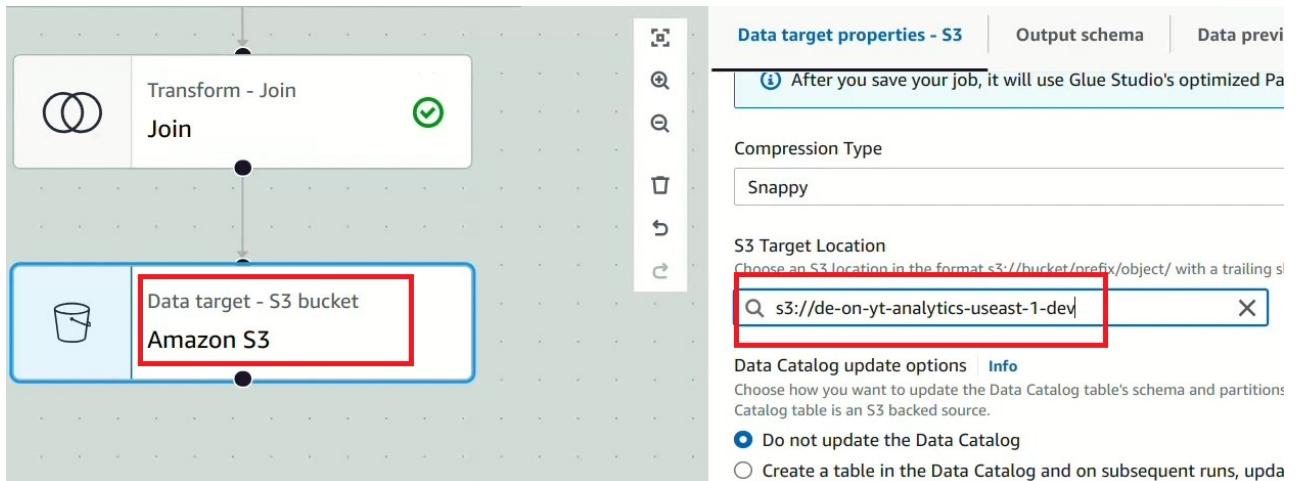
**Inner join**  
Select all rows from both datasets that meet the join condition.

Join conditions

Select a field from each parent node for the join condition.

<input type="text" value="AWS Glue Data Catalog category_id"/> <span style="border: 1px solid red; padding: 2px;">▼</span>	<input type="text" value="AWS Glue Data Catalog id"/> <span style="border: 1px solid red; padding: 2px;">▼</span> = <span style="border: 1px solid red; padding: 2px;">C</span>
----------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- 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**

The screenshot shows the AWS Glue Data Catalog interface. On the left, the navigation menu has 'Databases' selected and highlighted with a red box. In the main area, there is a table titled 'Databases (2)'. The table has columns for Name, Description, Location URI, and Created on (UTC). Two databases are listed: 'de-on-yt-cleaned' and 'de-on-yt-raw'. At the top right, there is an 'Add database' button, which is also highlighted with a red box. Below the table, there is a section titled 'Data Integration and ETL' with three radio button options: 'Do not update the Data Catalog', 'Create a table in the Data Catalog and on subsequent runs, update the schema and add new partitions' (which is selected and highlighted with a red box), and 'Create a table in the Data Catalog and on subsequent runs, keep existing schema and add new partitions'. Further down, there is a 'Database' section where 'db\_yt\_analytics' is selected from a dropdown menu, which is highlighted with a red box. There is also a 'Use runtime parameters' link. At the bottom, there is a 'Table name' section where 'final\_analytics' is entered into a text input field, which is also highlighted with a red box.

- Set partition keys as **region** and **category\_id**

Partition keys - *optional*

Add partition keys.

Partition (0)

region



Partition (1)

category\_id



[Add a partition key](#)

- 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



Amazon 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](mailto:engrzubairkhatti@gmail.com)

[LinkedIn](#)