Requirement from client:

XYZ company wants to launch a new data driven campaign, and they are looking at YouTube as the main advertising channel.

XYZ’s initial questions:

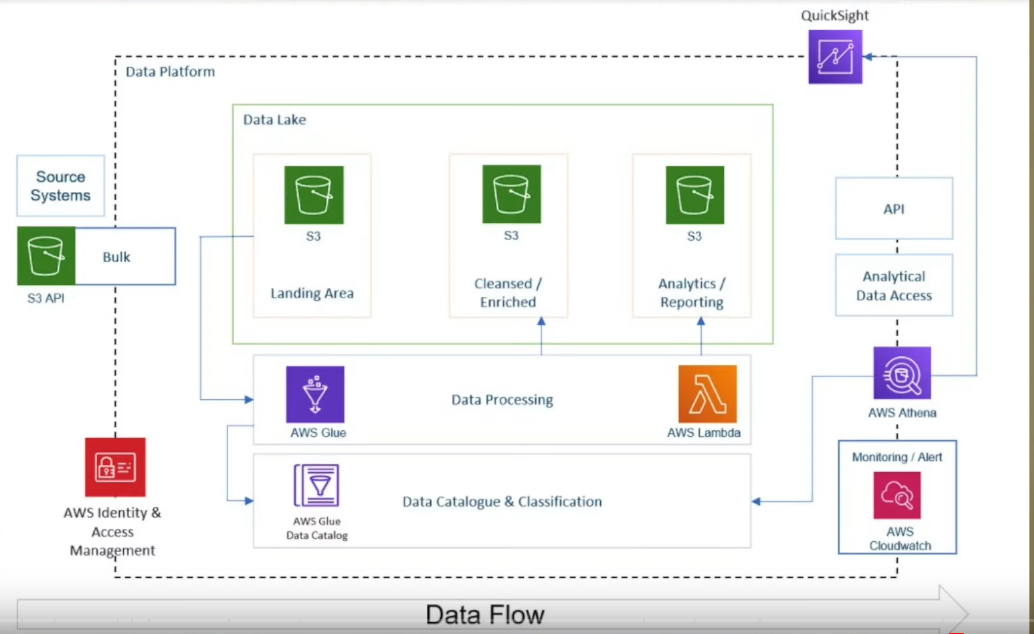
1. **How to categorize videos, based on their comments and statistics.**
2. **What factors affect how popular a YouTube video will be.**

**Goals and success Criteria for the project.**

1. Data Ingestion
2. ETL Design
3. Data Lake
4. Scalability
5. AWS Cloud
6. Reporting/Dashboard.

=====================================================================

Project Architecture.



=====================================================================

**Data Centers:**

On-Premise Data center: Company has its own hardware.

Could data center : These deliver service over the internet. You can rent the storage resource.

Dataset: **Trending YouTube Video Statistics**

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

SETUP

**Then I am creating an IAM user in AWS. Don’t use the root account as it has more permission and if anything happens you might lose all the data. Always use IAM account as it has less exposure. [You get CSV file with username, link ..]**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

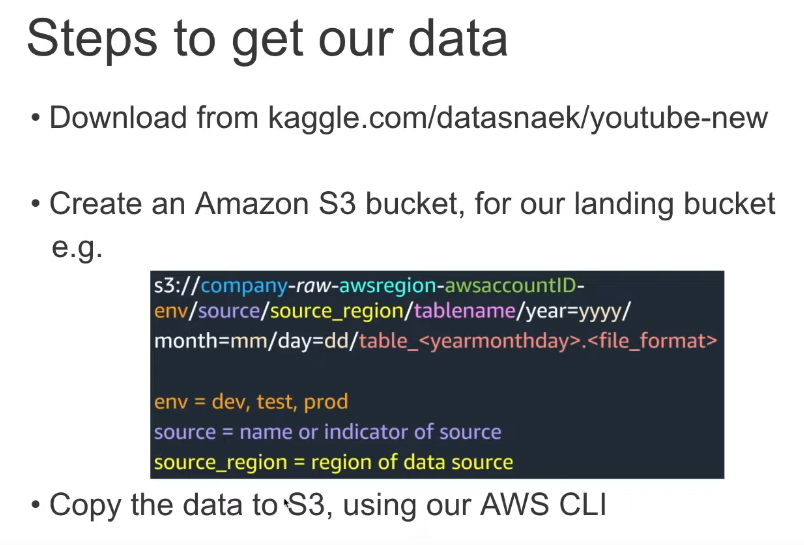
Then you can interact with AWS env in two ways AWS Console, Programmatically(AWS CLI, SDK)

We will use AWS CLI to upload the data to s3 bucket.

Then you have to download AWS CLI and then configure with credential data you got from above CSV file.

+++++++++++++++++++++++++++++++++++++++++++++++++++

STEP 1: Upload data into S3



Then create a **s3** bucket on AWS - **de-on-youtube-raw-useast1-prafuldev**

Upload the data into S3 bucket.

In terminal, navigate the folder where you have data.

# To copy all JSON Reference data to same location:

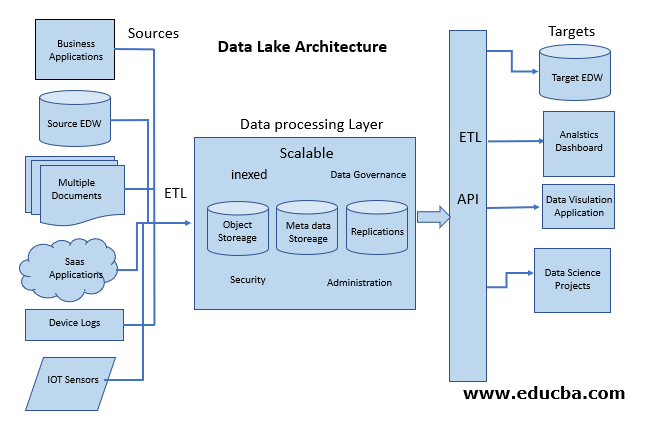
aws s3 cp . s3://de-on-youtube-raw-useast1-prafuldev/youtube/raw\_statistics\_reference\_data/ --recursive --exclude "\*" --include "\*.json"

| # To copy all data files to its own location, following Hive-style patterns: |  |  |
| --- | --- | --- |
| aws s3 cp CAvideos.csv s3://de-on-youtube-raw-useast1-dev/youtube/raw\_statistics/region=ca/ |  |  |

**=====================================================================**

**Data Lake House**

A data lake is a system or repository of data stored in its **natural/raw** format that may be structured or unstructured, from different source like relational database, non-relational database, machine learning, log analytics etc. usually in object blobs or files.

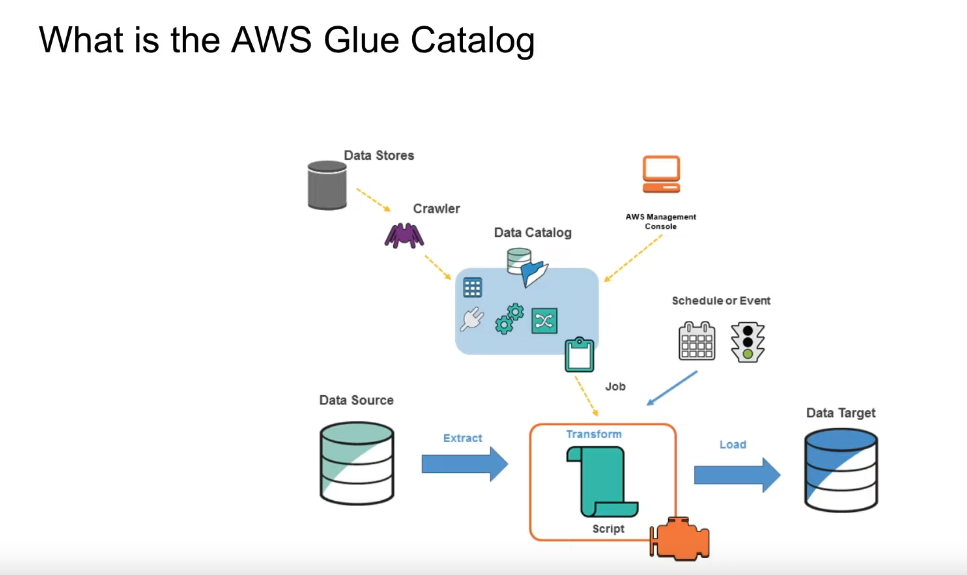


+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

STEP 2

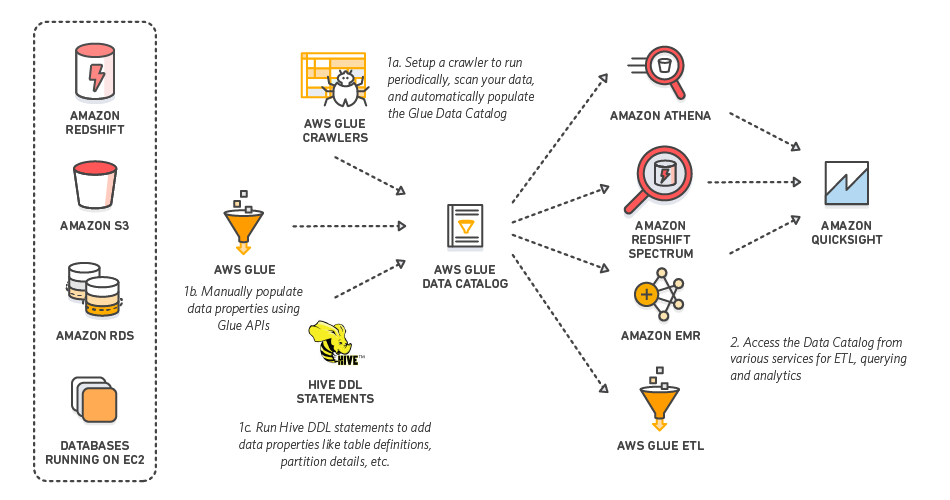
**AWS Glue Catalog**

Data catalog basically contain all the information of the data like number of column, data types etc.



The AWS Glue Data Catalog is a **metadata store**. It is a managed service that lets us to store, annotate, and share metadata in the AWS Cloud in the same way we would do in an

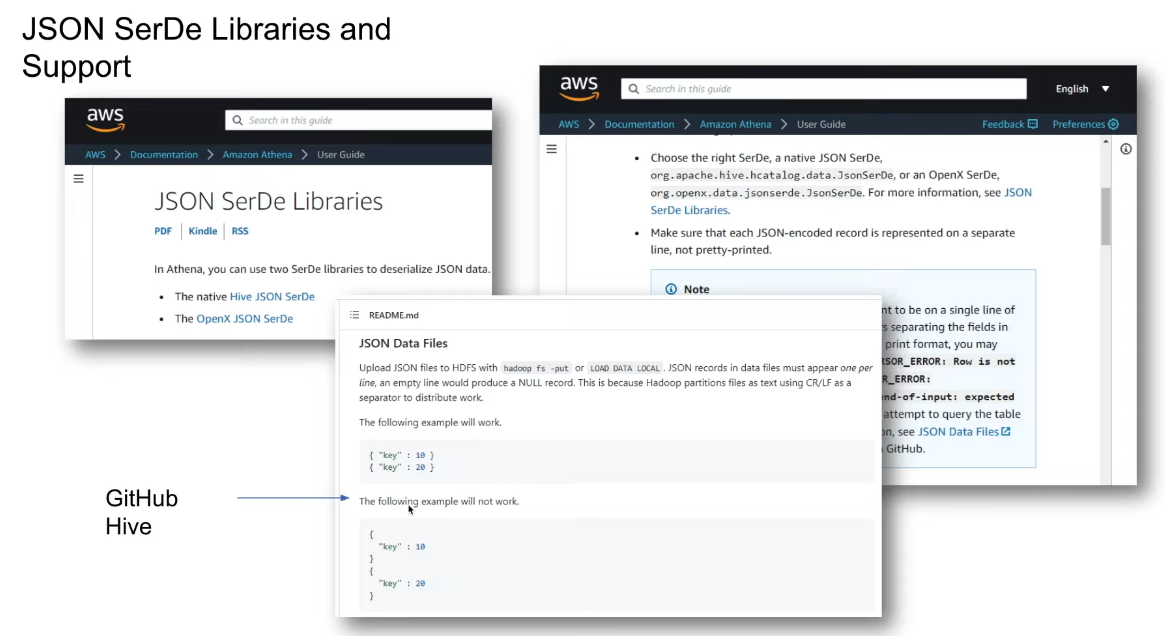
Apache Hive metastore. **Using AWS glue we can crawl the data from source and store in data catalog.**



AWS Glue can’t read certain json file structure, For glue we should provide every thing in one line not in like the below image it can’t understand. So now we have converting into tabular format.

It is called Apache Parquet format.

**Using AWS Glue we basically crawl the data and store in a catalog that is in metadata format. Here you can understand the column, data types, how the data is built etc.**



For that we need to use this schema, as above picture.

Now we are creating the crawler using, AWS glue

**de-on-youtube-raw-glue-catalog-1**

If glue wants to access data from s3, It doesn’t have permission. So we need to create ROLES for it. IN IAM you can create role.

**de-on-youtube-glue-s3-role** [Permissions s3 and glue service ]

Now the catalog(**de-on-youtube-raw-glue-catalog-1**) is created using glue. You can see the table also.

------------------------------------------------------------------------------------------------------------------

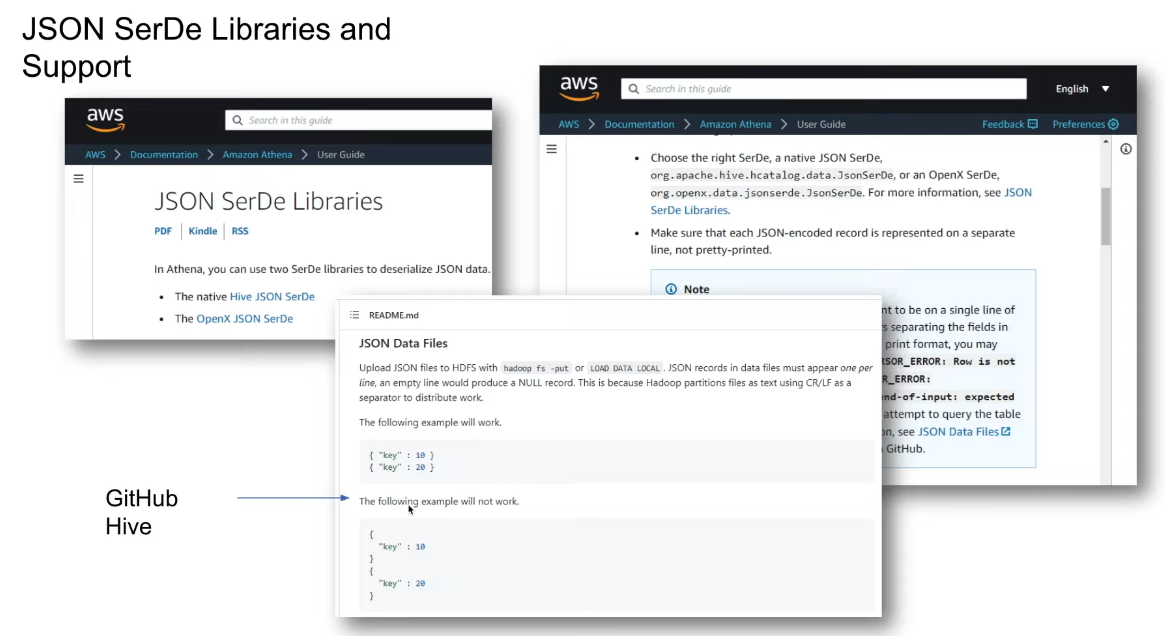
STEP 3

Then you can use AWS Athena, which ad hoc query toll. Using which we can query the data of what we have.

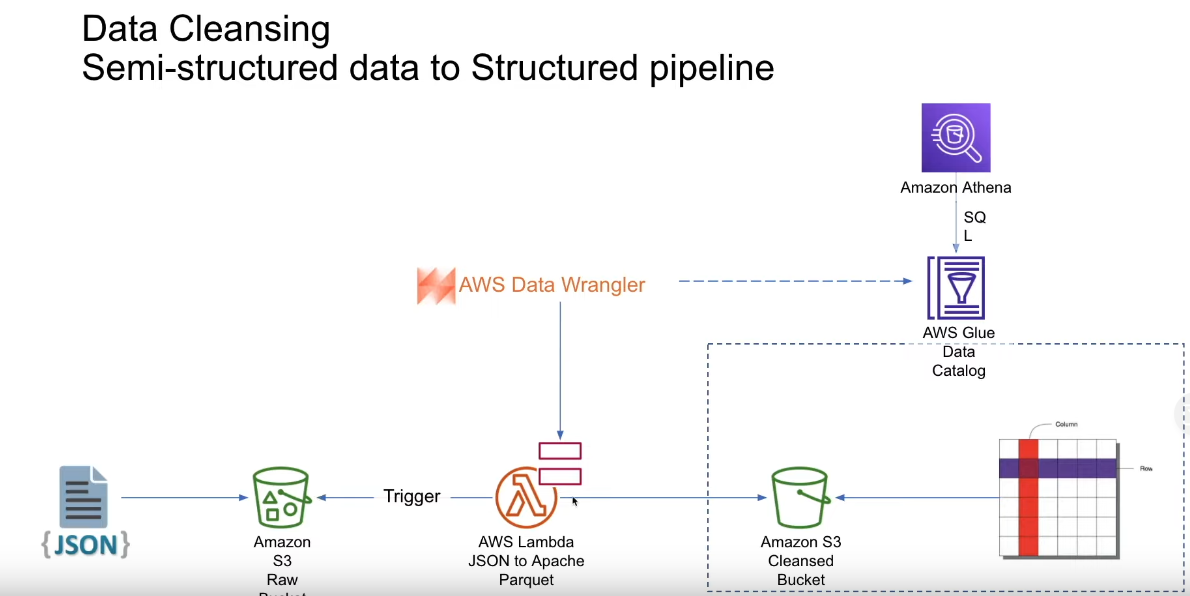
To use the AWS Athena, you have to provide output location that is use can create one more s3 bucket.

**De-on-youtube-raw-ueast1-athena-job (output bucket)**

**The data should be in this format for AWS Glue crawler to understand the data**



**So now we use json data and convert into tabular data. For that we will use this pipeline**



So now use AWS service which is called AWS Lambda to convert json into Apache parquet. Basically with AWS Lambda we can do some transformation and store the cleaned data back in s3 bucket. The you can query in AWS Athena.

====================================================================

STEP 4

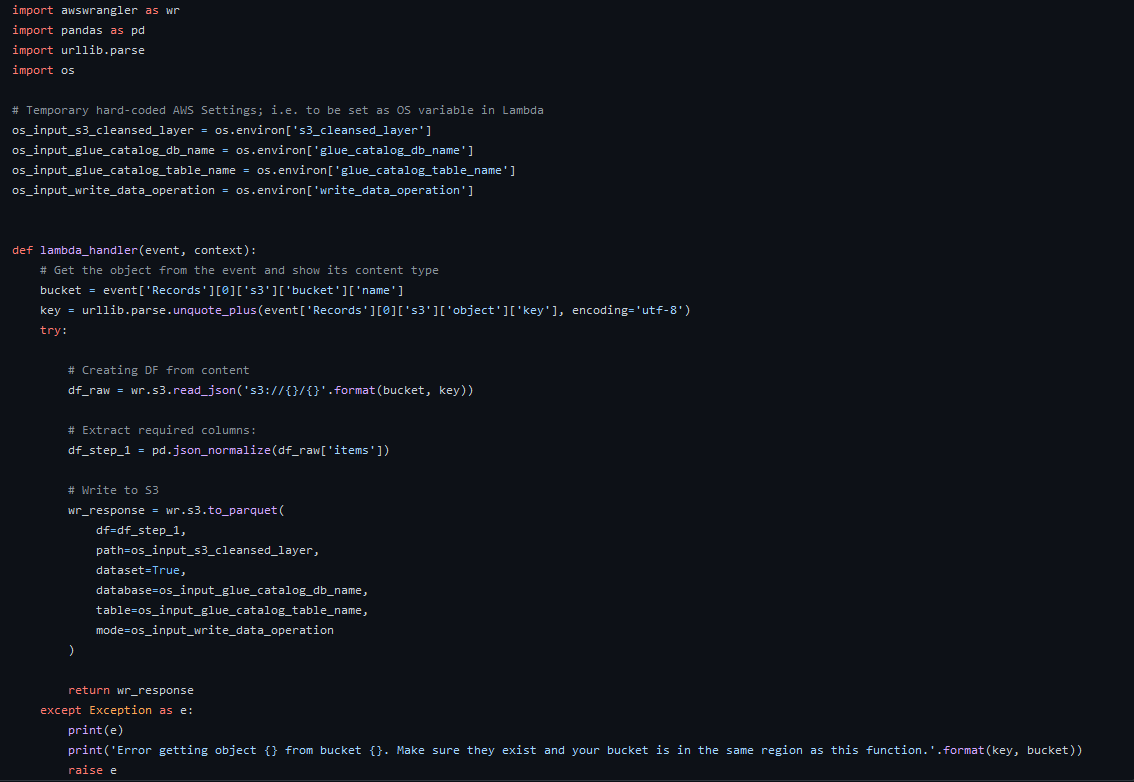
Creating a AWS Lambda

**de-on-youtube-raw-useast1-lambda-json-parquet**

For execution as AWS Glue we have to create one more role for AWS Lambda.

**de-on-youtube-raw-useast1-s3-lambda-role**

Now the Lambda function is created. Here you can write our python code for extracting data, do ETL , cleaning(basically we read what we want and then write the data into s3 bucket) and then convert to tabular data, Again store the cleaned data in s3.

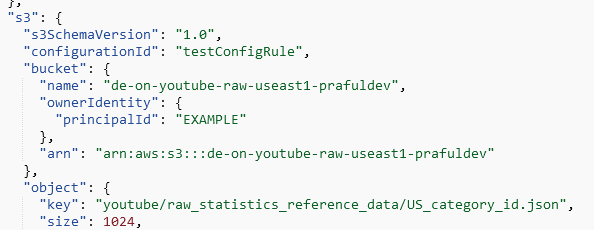
****

We have to create one more s3 bucket for data storage,

**de-on-youtube-cleansed-useast1-prafuldev**

**In configuration tab, add all the variables. Like db name, table name, s3 bucket, append.**

Then we run **s3\_put event**, where we have mention bucket name, and key name(basically what ever after that bucket name

****

Then if you run a program an table will be created. Remember to create a db in AWS glue with name what you have mentioned in ENV variable.

Now you can check the data, go to glue→ table then click on view table. There you have successfully converted unstructured json file data into structured data(tabular form). **We have used read\_json function to convert json into tabular.**

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

STEP -5

Now we are going to crawl all the raw data in CSV files according to region, Which make our table a partition table with region.

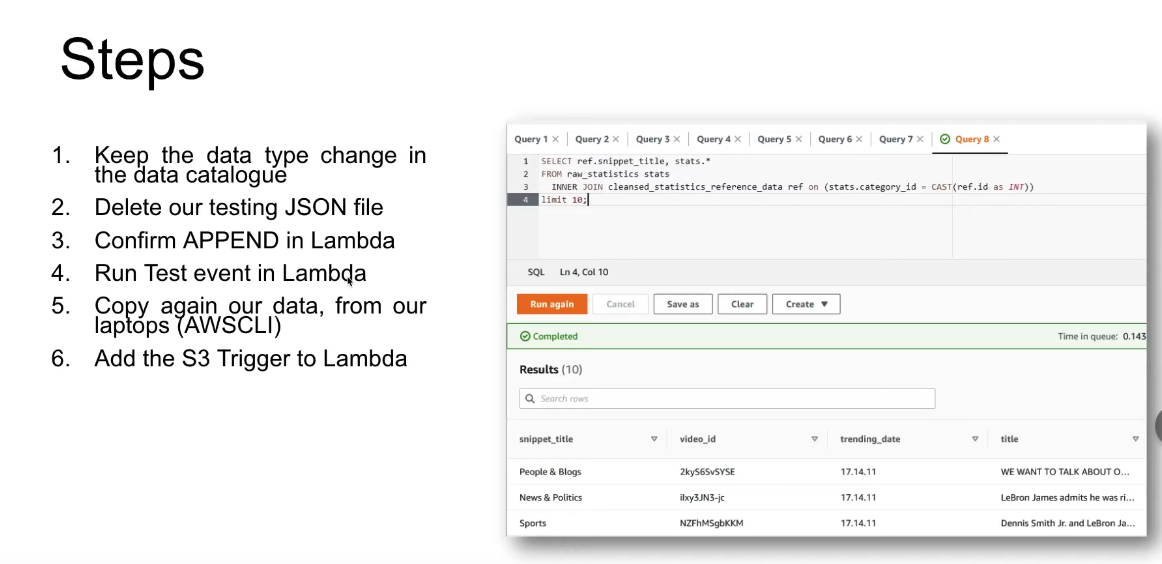
Create a new crawler **de-on-youtube-raw-csv-crawler-01 (**give data file path, and glue role**)**

**One thing you should notice here, When store data in s3 buckler we stored all the raw data files in for region=ca/ like this, this is to have the partition table with partition key as region.**

**Now in de\_youtube\_raw database, we have two tables (raw\_statistics - csv files) and )raw\_statistics\_reference \_data - json file)**

In Athena syntax to sql query - SELECT \* FROM "de\_youtube\_raw"."raw\_statistics" limit 10;

**When we tried to join table from two different database(json and csv table), We had issue with id begining string, we want int. We tried to convert to big int in glue schema itself, but did not work.**

****

**What we did was we changed the id into bigint inside the Glue catelog that we will keep as it is., then deleted the old data in cleaned database table. Then again we created new s3 put event in the AWS lambda by giving data file path, bucket name, key name and all. Then we run the test event in lambda. That solved big int issue.**

**==================================================================**

STEP 7

Now we are using AWS Glue ETL, →jobs(legacy) to store the cleaned data file of raw\_statitics (csv files) into cleaned database from raw database. Here we're also converting **csv files**(semi structured) into parquet file format.

AWS Glue will create a entire **PySpark script,** we have written code in such way it stores the data as per partition key in the s3 bucket which is cleaned database. Here we have to mention partionkeys, filter data for countries, in **PySpark script.**

Then we create a catalog on cleaned version data buckets,

**de-on-youtube-cleaned-csv-to-parquet-etl**

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**STEP 8**

Then we have to create a **trigger(some kind of automation mechanism)** so that when there will be new data file into bucket, the lambda function should convert the **json file into cleaned parquet** format data.

Add trigger→bucket name→all objects(event type) → prefix(key) → suffix(.json)

We delete the data in s3 bucket to test - all the json file and again uploaded the data to s3 bucket from AWS CLI .

Now in db\_youtube\_cleaned database we have two cleaned tables of json files and csv files in parquet format(tabular format).

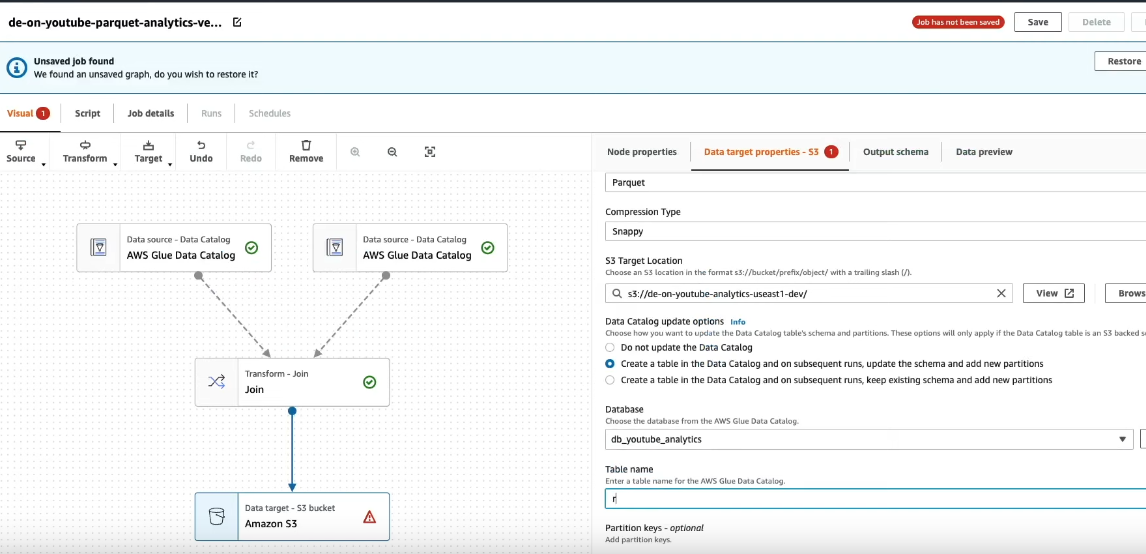
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**STEP 9**

**Now using Glue Studio we us it for graphical interface that makes it easy to create, run, and monitor extract, transform, and load (ETL) jobs in AWS Glue.**

**We need to join the two tables→ then store that data in another analytical bucket,**

**In this ETL pipeline we can store based on different partition keys, like region, id etc.**

****

**Why analytical layer is important, as data scientist he can’t write query all the time, so after joining the data using AWS glue studio it is easy to query the data.**

**+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**Dash Board for data visulization**

**Using QuickSight we can do the data visualization according to requirement based on the question asked my clients.**

**==============================================================**