Prerequisite Concepts:

AWS

S3 bucket

Why IAM account is necessary?

**Data lake**: centralized repository, all data will be stored in one repository such as structured/unstructured data, audio files, video files, etc.

**AWS Glue Catalog**: Data catalog is like metadata of data we are storing. (How many columns, types of columns, kind of data in columns, etc). Glue catalog discovers the data coming from various sources, extracts the metadata, builds a glue catalog where you can do ETL.

Steps:

1. Created AWS account and created IAM User.

2. Installed AWS CLI to access AWS console programmatically.

3. Downloaded dataset from Kaggle: <https://www.kaggle.com/datasets/datasnaek/youtube-new?resource=download>

4. Created S3 bucket and uploaded dataset onto the bucket using AWS CLI.

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

A screen shot of a computer

Description automatically generated

5. Created different folder for different region using below commands,

aws s3 cp CAvideos.csv s3://yt-raw-useast1-dev-de/youtube/raw\_statistics/region=ca/

aws s3 cp DEvideos.csv s3://yt-raw-useast1-dev-de/youtube/raw\_statistics/region=de/

aws s3 cp FRvideos.csv s3://yt-raw-useast1-dev-de/youtube/raw\_statistics/region=fr/

aws s3 cp GBvideos.csv s3://yt-raw-useast1-dev-de/youtube/raw\_statistics/region=gb/

aws s3 cp INvideos.csv s3://yt-raw-useast1-dev-de/youtube/raw\_statistics/region=in/

aws s3 cp JPvideos.csv s3://yt-raw-useast1-dev-de/youtube/raw\_statistics/region=jp/

aws s3 cp KRvideos.csv s3://yt-raw-useast1-dev-de/youtube/raw\_statistics/region=kr/

aws s3 cp MXvideos.csv s3://yt-raw-useast1-dev-de/youtube/raw\_statistics/region=mx/

aws s3 cp RUvideos.csv s3://yt-raw-useast1-dev-de/youtube/raw\_statistics/region=ru/

aws s3 cp USvideos.csv s3://yt-raw-useast1-dev-de/youtube/raw\_statistics/region=us/

A screenshot of a computer

Description automatically generated

6. Create new crawler yt-raw-glue-catalog-1, provided path ‘’s3//yt-raw-useast1-dev-de/youtube/raw\_statistics\_reference\_data’ to thejson files.

7. Created Role ‘yt-glue-s3-role’ for a service to access another service

(Example when a service such as Glue wants to access data from S3, it doesn’t have direct permission, hence we create this Role and gave AmazonS3Full access permission policy)

Need to give one more permission, click on Add permission -> attach policy, add ‘AWSGlueServiceRole’.

8. Created Database ‘yt-raw-de’.

9. In crawler add this role ‘yt-glue-s3-role’. Once completed, click on ‘Run Crawler’, this crawler will go to S3 bucket and the data that we stored in folders and understand what this data contain and build a catalog.

10. Click on tables and view tables, it will redirect to Athena page, but it requires one more bucket to store output. So created another bucket with name ‘yt-raw-useast1-de-athena-job’. Clicked on Manage settings and add the Athena bucket.

11. Tried to run the command but it threw an error ‘HIVE\_CURSOR\_ERROR: Row is not a valid JSON Object - JSONException: A JSONObject text must end with '}' at 2 [character 3 line 1]’. Because AWS has some specific rules for json. Hence, we convert JSON into table format i.e., Apache Parquet format.

12. Create Lambda and add role (AmazonS3Full access) like step 7. Added environment variables.

Key: glue\_catalog\_db\_name Value: db\_youtube\_cleaned

Key: glue\_catalog\_table\_name Value:cleaned\_statistics\_reference\_data

Key: s3\_cleansed\_layer Value: s3://yt-cleansed-useast1-dev-de/youtube

Key: write\_data\_operation Value: append

A screenshot of a computer

Description automatically generated

13. Created new bucket to store the cleansed data ‘yt-cleansed-useast1-dev-de’. Add to envi variables ‘s3://yt-cleansed-useast1-dev-de/youtube’. Once added envi variables, test the code.

14. Use S3 PUT event to run the code. Edit respective bucket name and key and hit Run.

15. Got error

Unable to import module 'lambda\_function': No module named 'awswrangler'".

Reason: Whenever lambda runs it creates new compute environment for every run it creates and does not have all the packages it needs to run that code. Hence, we have lambda layers which is attached to compute environment and has all different packages the code requires to run successfully.

Solution: Add layer ‘AWSSDKPandas-Python312’ on lambda.

16. Error 2

"2024-01-24T01:56:11.237Z 41497ce5-a76f-4f17-b20b-c0c25afeb08e Task timed out after 3.20 seconds."

Solution: Configuration -> General settings -> increase timeout to 3 seconds.

17. Error 3

An error occurred (AccessDeniedException) when calling the GetTable operation: User: arn:aws:sts::905418370190:assumed-role/yt-raw-useast1-s3-lambda-role/yt-raw-useast1-lambda-json-parquet is not authorized to perform: glue:GetTable on resource: arn:aws:glue:us-east-1:905418370190:catalog because no identity-based policy allows the glue:GetTable action",

Code is for creating Glue catalog and glue table. We need to give permission of Glue to Lambda as well. In step 12 we only gave access to S3 and not Glue. So, repeat same steps and give permission ‘AWSGlueServiceRole’ to Lambda.

18. Successful output

Response

{

"paths": [

"s3://yt-cleansed-useast1-dev-de/youtube/d177f7375cfc4c488f86d2aab7294e2b.snappy.parquet"

],

"partitions\_values": {}

}

It means it had created a file under our S3 bucket.

A screenshot of a web page

Description automatically generated

In code, we define database name, table name of Glue, which writes data to S3 bucket, creating Glue catalog and table on Glue.

Click on table and View data in Athena and we can see the structured version of the JSON data.

A screenshot of a computer

Description automatically generated

19. Created new crawler ‘yt-raw-csv-crawler-01’

Path: ‘s3://yt-raw-useast1-dev-de/youtube/raw\_statistics/’

Database: ‘yt-raw-de’

IAM role: ‘yt-raw-glue-s3-role’

Go to Athena, select database ‘yt-raw-de’.

A screenshot of a computer

Description automatically generated

We did partition table for region wise so that we can select data from particular region.

SELECT \* FROM "yt-raw-de"."raw\_statistics" where region='ca';

20. JOIN both tables raw\_statistics and raw\_statistics\_reference\_data

*SELECT \* FROM "yt-raw-de"."raw\_statistics" a*

*INNER JOIN "db\_youtube\_cleaned"."cleaned\_statistics\_reference\_data" b ON a.category\_id=b.id;*

Error:

TYPE\_MISMATCH: line 2:87: Cannot apply operator: bigint = varchar.

In raw\_statistics table the id is bigint but in cleaned table it is String.

*SELECT \* FROM "yt-raw-de"."raw\_statistics" a*

*INNER JOIN "db\_youtube\_cleaned"."cleaned\_statistics\_reference\_data" b ON a.category\_id=cast(b.id as int);*

21. Further edit the query to select only title, id from raw table and snippet title from cleaned table where region is CA.

*SELECT a.title, a.category\_id, b.snippet\_title FROM "yt-raw-de"."raw\_statistics" a*

*INNER JOIN "db\_youtube\_cleaned"."cleaned\_statistics\_reference\_data" b ON a.category\_id=cast(b.id as int)*

*where a.region='ca';*

A screenshot of a computer

Description automatically generated

22. Usually you should not cast datatype rather convert datatype before running query.

So if we see schema of cleaned table we see datatype of id as string, we will edit schema and change to bigint.

A screenshot of a computer

Description automatically generated

23.

*SELECT a.title, a.category\_id, b.snippet\_title FROM "yt-raw-de"."raw\_statistics" a*

*INNER JOIN "db\_youtube\_cleaned"."cleaned\_statistics\_reference\_data" b ON* ***a.category\_id=b.id***

*where a.region='ca';*

Error: TYPE\_MISMATCH: Unable to read parquet data. This is most likely caused by a mismatch between the parquet and metastore schema.

Reason: Parquet file comes with metadata. All columns have their own schema attached to it. When we created parquet file, id column had string datatype attached to it, when we convert datatype inside AWS catalog, it doesn’t change that datatype inside parquet file.

Solution:

-Go to S3 bucket and select cleaned version and delete the file. And run code again, it will create new object in cleansed table.

Now when we run query in Athena, it gives output.

*SELECT a.title, a.category\_id, b.snippet\_title FROM "yt-raw-de"."raw\_statistics" a*

*INNER JOIN "db\_youtube\_cleaned"."cleaned\_statistics\_reference\_data" b ON* ***a.category\_id=b.id****where a.region='ca';*

A screenshot of a computer

Description automatically generated

A screenshot of a computer

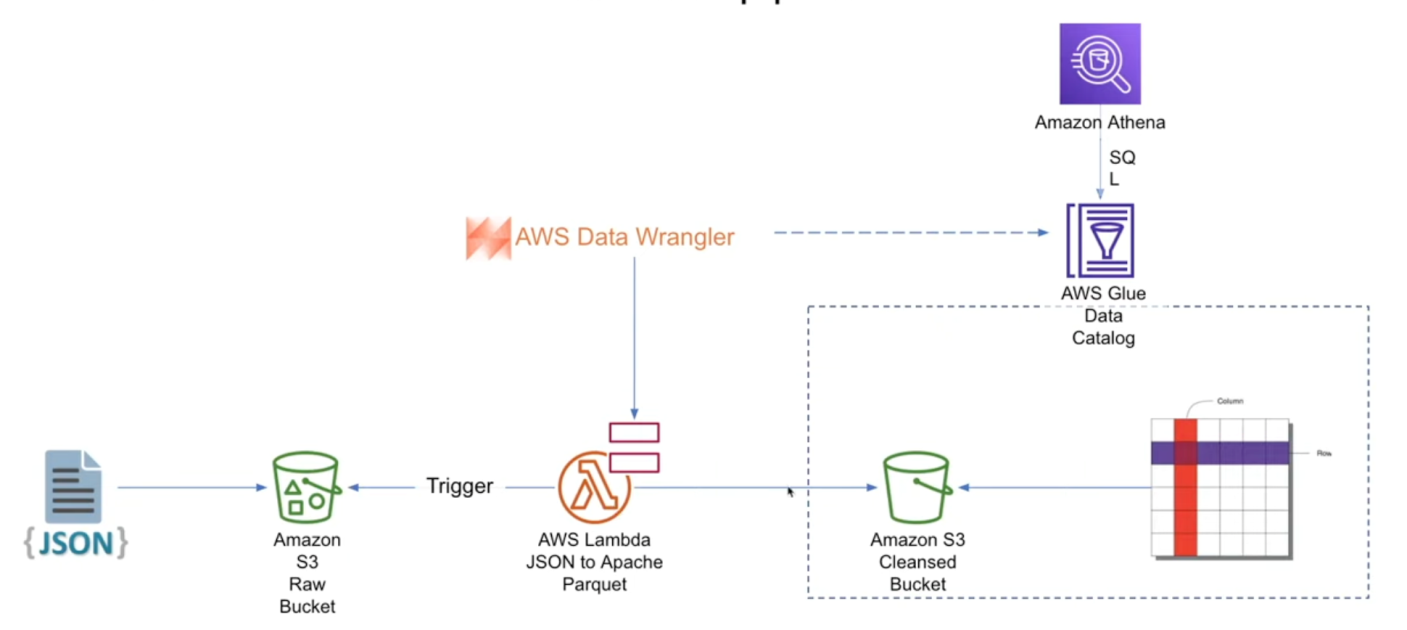
Description automatically generated

24. Move raw region data into cleaner data using glue ETL

Create JOB from Glue with name ‘yt-cleansed-csv-to-parquet-de’

Added new node with S3 bucket and

The second part, when it clicks next, you have to go to the visual (first item on the same tab you clicked on job details), add a node, first you choose the S3 bucket to choose the source, than next you add a new node from schema, and than a third node on the target tab



A screenshot of a computer

Description automatically generated

Bulk: Data coming from different sources (Kaggle)

Landing area: Where we uploaded raw data.

Cleansed: Cleaned data is stored in second bucket. Used AWS Lambda to process and store the data in this new bucket.

A screenshot of a computer

Description automatically generated

A screenshot of a facebook page

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Notes:

Uploaded data to S3 bucket. Why? S3 is basically used for storage (called as object storage).

Naming conventions should be followed.