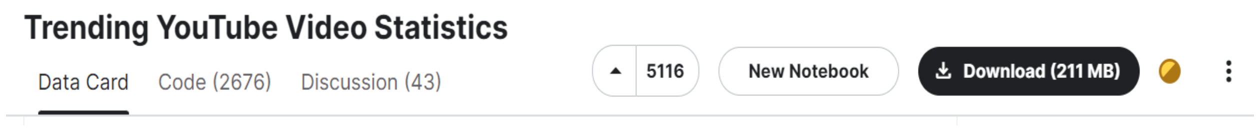
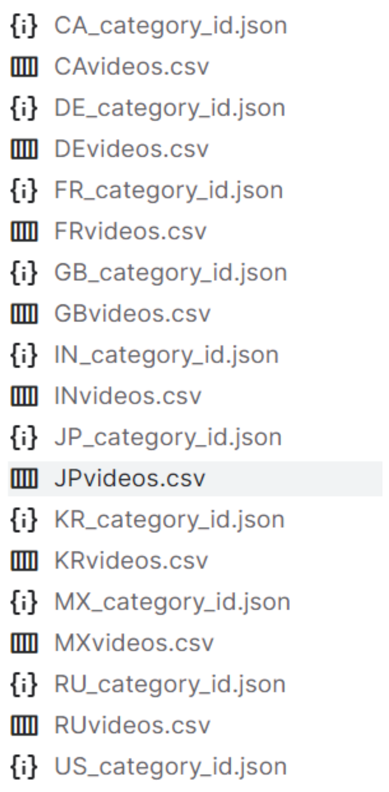
Youtube Analytics:

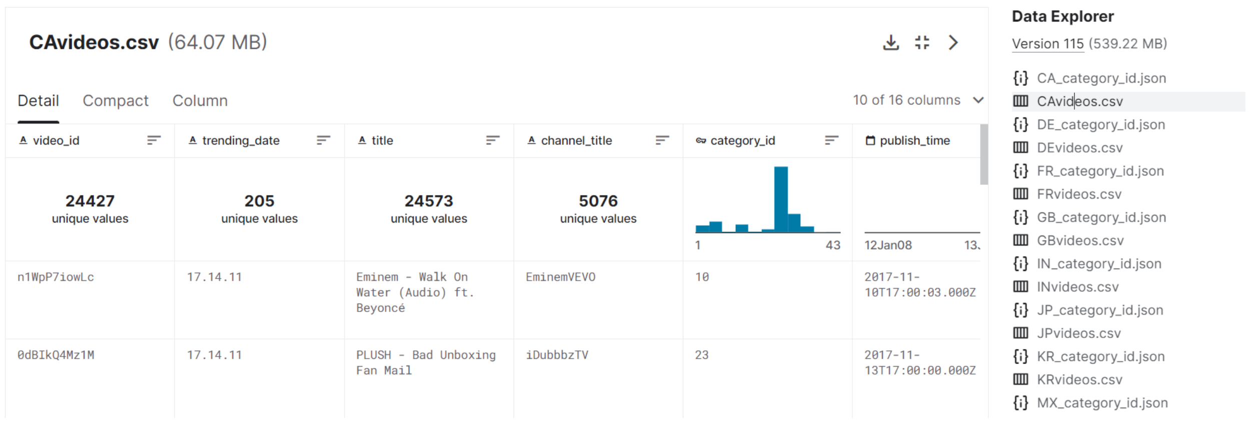
Data:


Trending YouTube Video Statistics

https://www.kaggle.com/datasets/datasnaek/youtube-new?select=JPvideos.csv

The Dataset contains information of trending videos on the most popular video streaming services, YouTube. The data was collected through YouTube API and contains regional information about trending videos fōrom around 2017-18. There are 10 regions in total where data was fetched from. For each region, we have two files- one JSON and CSV.


CA_category_id.json
CAvideos.csv
DE_category_id.json
DEvideos.csv
FR_category_id.json
FRvideos.csv
GB_category_id.json
GBvideos.csv
IN_category_id.json
INvideos.csv
JP_category_id.json
JPvideos.csv
KR_category_id.json
KRvideos.csv
MX_category_id.json
MXvideos.csv
RU_category_id.json
RUvideos.csv
US_category_id.json

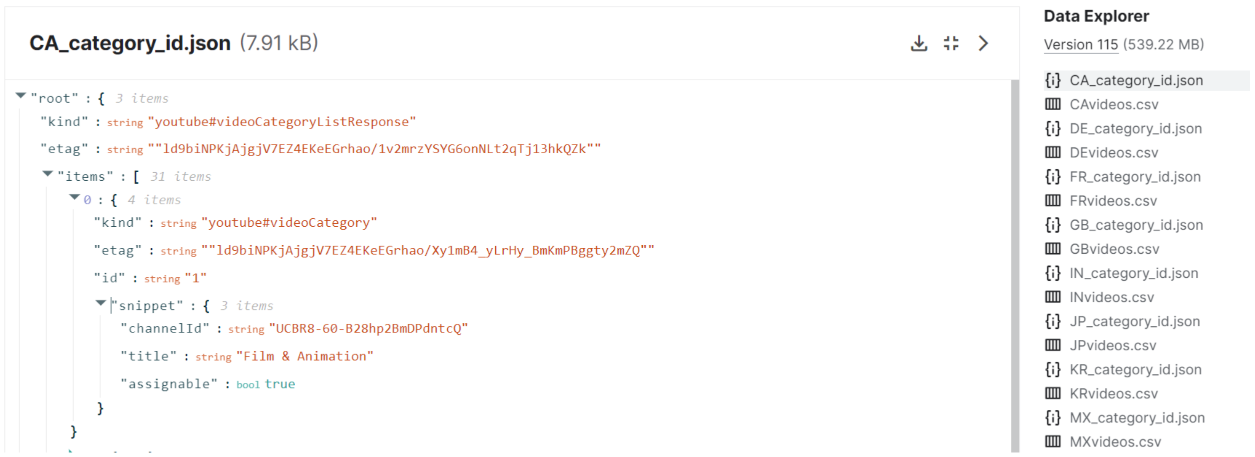

CAvideos.csv (64.07 MB)

Detail Compact Column — 10 of 16 columns

video_id	trending_date	title	channel_title	category_id	publish_time
24427 unique values	205 unique values	24573 unique values	5076 unique values	1 – 43	12Jan08 – 13.
n1WpP7iowLc	17.14.11	Eminem - Walk On Water (Audio) ft. Beyoncé	EminemVEVO	10	2017-11-10T17:00:03.000Z
0dBIkQ4Mz1M	17.14.11	PLUSH - Bad Unboxing Fan Mail	iDubbbzTV	23	2017-11-13T17:00:00.000Z

Data Explorer
Version 115 (539.22 MB)

- CA_category_id.json
- CAvideos.csv
- DE_category_id.json
- DEvideos.csv
- FR_category_id.json
- FRvideos.csv
- GB_category_id.json
- GBvideos.csv
- IN_category_id.json
- INvideos.csv
- JP_category_id.json
- JPvideos.csv
- KR_category_id.json
- KRvideos.csv
- MX_category_id.json


CA_category_id.json (7.91 kB)

```
"root" : { 3 items
  "kind" : string "youtube#videoCategoryListResponse"
  "etag" : string ""ld9biNPKjAjgjV7EZ4EKeEGrhao/1v2mrzYSYG6onNLt2qTj13hkQZk""
  "items" : [ 31 items
    0 : { 4 items
      "kind" : string "youtube#videoCategory"
      "etag" : string ""ld9biNPKjAjgjV7EZ4EKeEGrhao/Xy1mB4_yLrHy_BmKmPBggty2mZQ""
      "id" : string "1"
      "snippet" : { 3 items
        "channelId" : string "UCBR8-60-B28hp2BmDPdntcQ"
        "title" : string "Film & Animation"
        "assignable" : bool true
      }
    }
```

Data Explorer
Version 115 (539.22 MB)

- CA_category_id.json
- CAvideos.csv
- DE_category_id.json
- DEvideos.csv
- FR_category_id.json
- FRvideos.csv
- GB_category_id.json
- GBvideos.csv
- IN_category_id.json
- INvideos.csv
- JP_category_id.json
- JPvideos.csv
- KR_category_id.json
- KRvideos.csv
- MX_category_id.json
- MXvideos.csv

Because of the difference in file format/structure, I have to combine the csv and json files for each region to make a final analysis model.

Approach:
For this I utilized AWS Cloud Services. AWS offers several advantages over on premises infrastructure including Scalability, Cost Effectiveness, Serverless architecture, Services made only for Data Analytics, BDA.

For my data storage, I needed tools to save structured and semi-structured data which is why I thought of S3 Data Lake formation. S3-based data lakes offer cost advantages compared to traditional data warehouses.

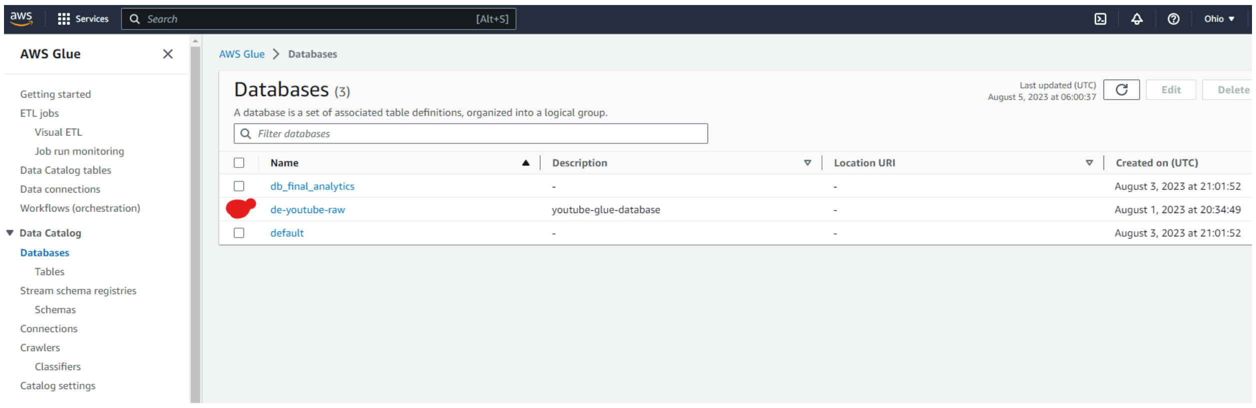
Data lakes are well-suited for storing and managing diverse data types, including structured, semi-structured (like JSON), and unstructured data (like CSV). Since the YouTube dataset contains both JSON and CSV files, a data lake provides the flexibility to store and process this mixed data without the need to convert it to a rigid schema upfront.

Pipeline 1:

Processing JSON files:

After downloading files from Kaggle, I uploaded all json files to my S3 raw data bucket through AWS CLI. Before doing this, I created an IAM user with my root account and created an S3 raw Bucket in it which will contain my raw data (JSON in this case).

My data is now uploaded to S3, to understand the schema and do further analysis in Athena, I need to create a Glue crawler that will crawl through my data and store metadata in the Glue catalog.



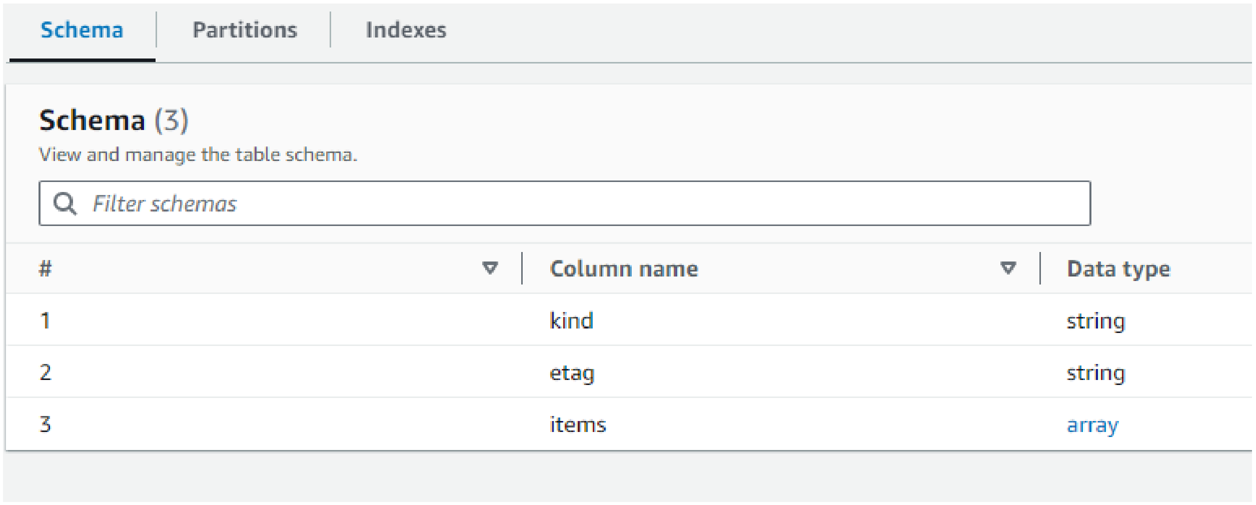
Databases (3)

A database is a set of associated table definitions, organized into a logical group.

Last updated (UTC) August 5, 2023 at 06:00:37

Name	Description	Location URI	Created on (UTC)
db_final_analytics	-	-	August 3, 2023 at 21:01:52
de-youtube-raw	youtube-glue-database	-	August 1, 2023 at 20:34:49
default	-	-	August 3, 2023 at 21:01:52

Here is how the schema looked after crawling through my raw s3:



Schema (3)

View and manage the table schema.

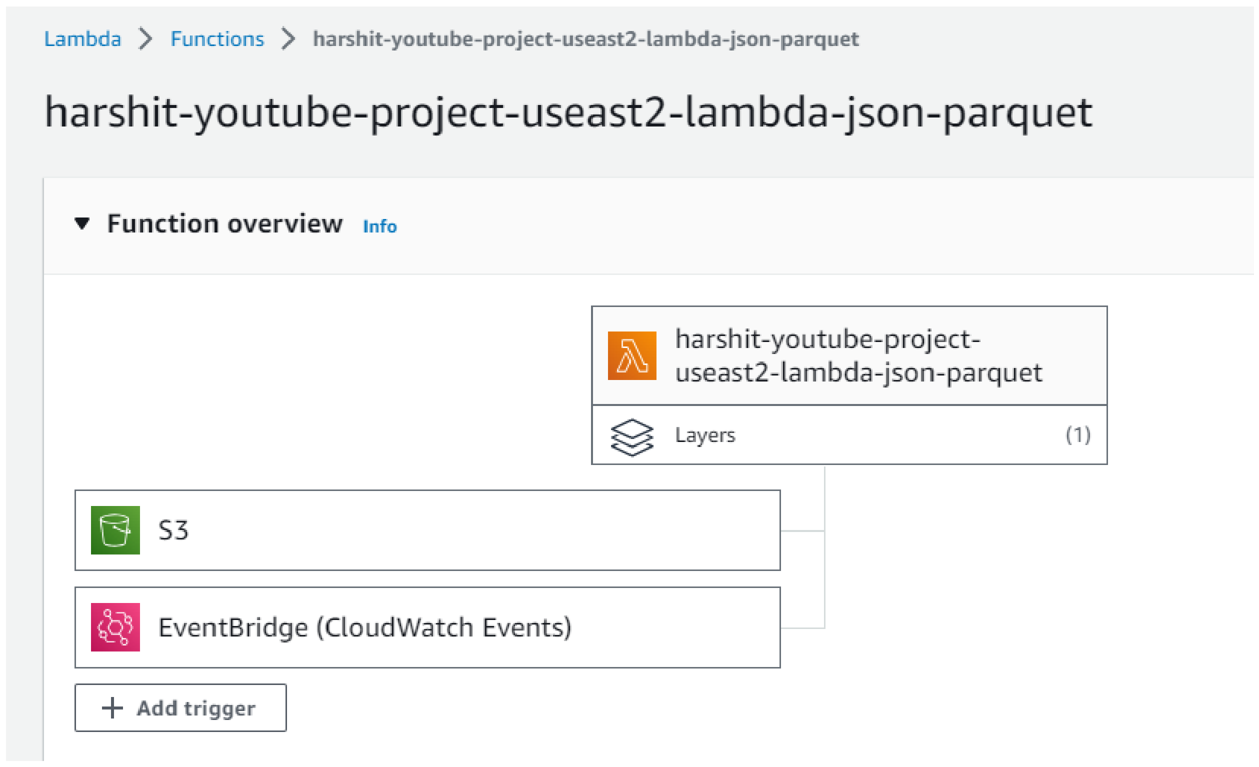
#	Column name	Data type
1	kind	string
2	etag	string
3	items	array

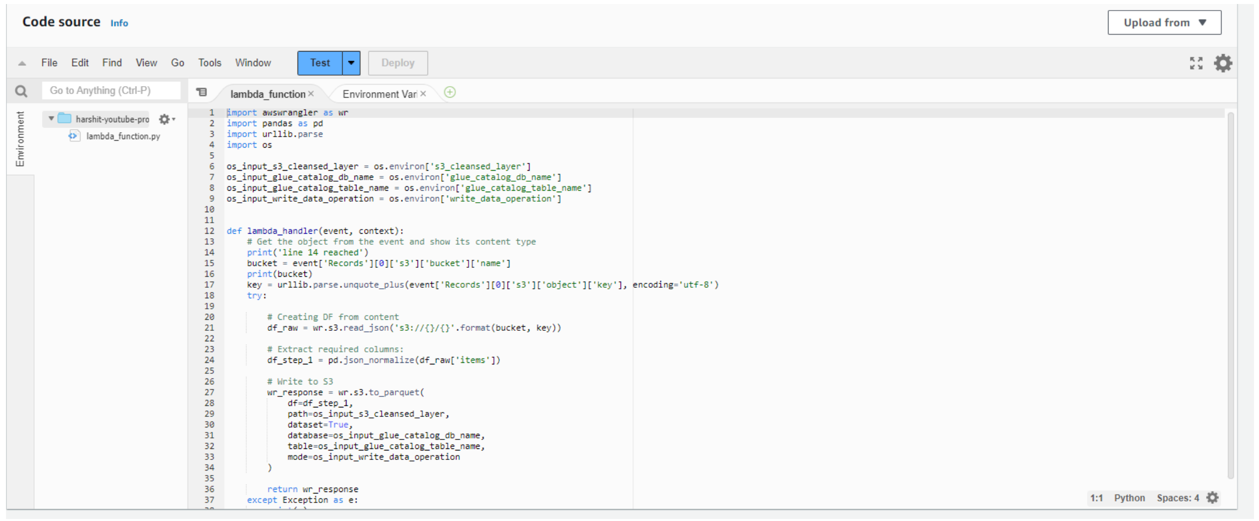
<u>Roles & Permissions:</u> I gave my glue crawler an IAM role that permits it to access S3 and Glue Service Role.

Now this catalog can be referred to by Athena for querying data in s3. But when I ran Athena to acces contents of s3, I received an error which required me to convert my multi line json format data to Parquet format, a format which makes Athena process queries efficiently.

For this a ETL pipeline is made with following architecture:

Setting up Lambda Function:


Lambda > Functions > harshit-youtube-project-useast2-lambda-json-parquet
harshit-youtube-project-useast2-lambda-json-parquet
Function overview Info
harshit-youtube-project-useast2-lambda-json-parquet
Layers (1)
S3
EventBridge (CloudWatch Events)
+ Add trigger



```python
import awswrangler as wr
import pandas as pd
import urllib.parse
import os

os_input_s3_cleansed_layer = os.environ['s3_cleansed_layer']
os_input_glue_catalog_db_name = os.environ['glue_catalog_db_name']
os_input_glue_catalog_table_name = os.environ['glue_catalog_table_name']
os_input_write_data_operation = os.environ['write_data_operation']


def lambda_handler(event, context):
    # Get the object from the event and show its content type
    print('line 14 reached')
    bucket = event['Records'][0]['s3']['bucket']['name']
    print(bucket)
    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
    try:

        # Creating DF from content
        df_raw = wr.s3.read_json('s3://{}/{}'.format(bucket, key))

        # Extract required columns:
        df_step_1 = pd.json_normalize(df_raw['items'])

        # Write to S3
        wr_response = wr.s3.to_parquet(
            df=df_step_1,
            path=os_input_s3_cleansed_layer,
            dataset=True,
            database=os_input_glue_catalog_db_name,
            table=os_input_glue_catalog_table_name,
            mode=os_input_write_data_operation
        )

        return wr_response
    except Exception as e:
```

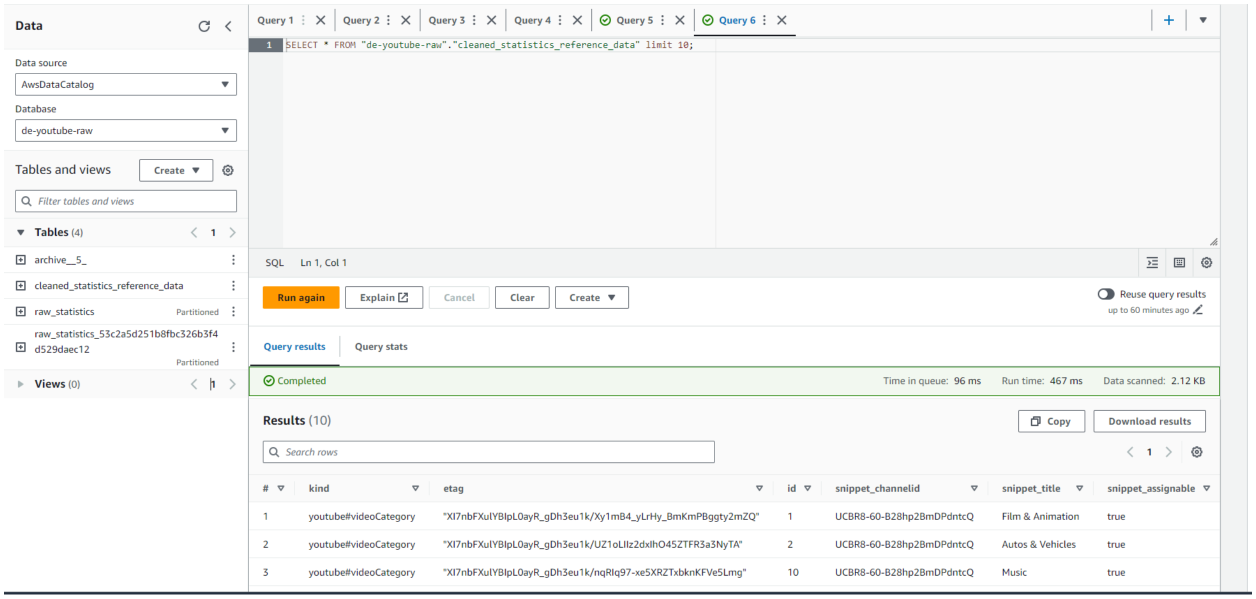
Roles: I gave my Lambda Function same IAM role that permits it to access S3 and Glue .

Here Lambda function needs to be triggered whenever an object is put on s3 raw. The code in run time environment will be executed which will convert the multi line JSON into Parquet format with help of environment variables (info about destination bucket, catalog and table). Additionally, the lambda function will write to the glue data catalog which stores schema of json converted parquet data in s3. I used Pandas and AWS Wrangler library in python for reading, converting to parquet and loading the file in the target destination. For this I had to upload a zip file of these libraries to set up the lambda layer. The transformed data will be stored in a new s3 named S3 Cleaned.

Lambda Trigger: I created an S3 trigger that triggers my lambda function whenever an object is inserted in the bucket. This will remove manual involvement and will automatically trigger the function, convert the files and store in the new S3 (also create glue catalog).

Using CloudWatch logs to solve trigger problem for lambda:
Problems faced: I couldn't invoke my lambda function whenever objects were inserted. So I created cloud watch event to check status of my function when invoked. It turned out that CloudWatch event were not triggered when new objects were inserted, this was due to incorrect bucket prefix I configured my lambda function with. To solve this I simply removed the prefix preference from trigger.

Now my S3 cleaned bucket contains converted parquet files which can be accessed by Athena. I'm now able to access files in S3 cleaned buckets with help of the catalog that I got from lambda. Here is how it looks after I ran query on S3 cleaned Bucket:



```
SELECT * FROM "de-youtube-raw"."cleaned_statistics_reference_data" limit 10;
```

Completed — Time in queue: 96 ms Run time: 467 ms Data scanned: 2.12 KB

Results (10)

#	kind	etag	id	snippet_channelid	snippet_title	snippet_assignable
1	youtube#videoCategory	"XI7nbFXulYBIpL0ayR_gDh3eu1k/Xy1mB4_yLrHy_BmKmPBggty2mZQ"	1	UCBR8-60-B28hp2BmDPdntcQ	Film & Animation	true
2	youtube#videoCategory	"XI7nbFXulYBIpL0ayR_gDh3eu1k/UZ1oLIIz2dxIhO45ZTFR3a3NyTA"	2	UCBR8-60-B28hp2BmDPdntcQ	Autos & Vehicles	true
3	youtube#videoCategory	"XI7nbFXulYBIpL0ayR_gDh3eu1k/nqRIq97-xe5XRZTxbknKFVe5Lmg"	10	UCBR8-60-B28hp2BmDPdntcQ	Music	true

Pipeline 2:

Processing CSV data:

Now that my JSON file is converted, I also need to do some transformation in my csv files: change data type for columns like likes, comments, views etc. In the process I will also try to convert csv to parquet so that my final bucket contains both json converted parquet files from pipeline 1 and csv to parquet files from pipeline 2.

When uploading csv files to my S3 raw, I want information about the schema of files. So I set up a crawler that will store catalog in Glue Catalog that will crawl through csv files in S3 raw.

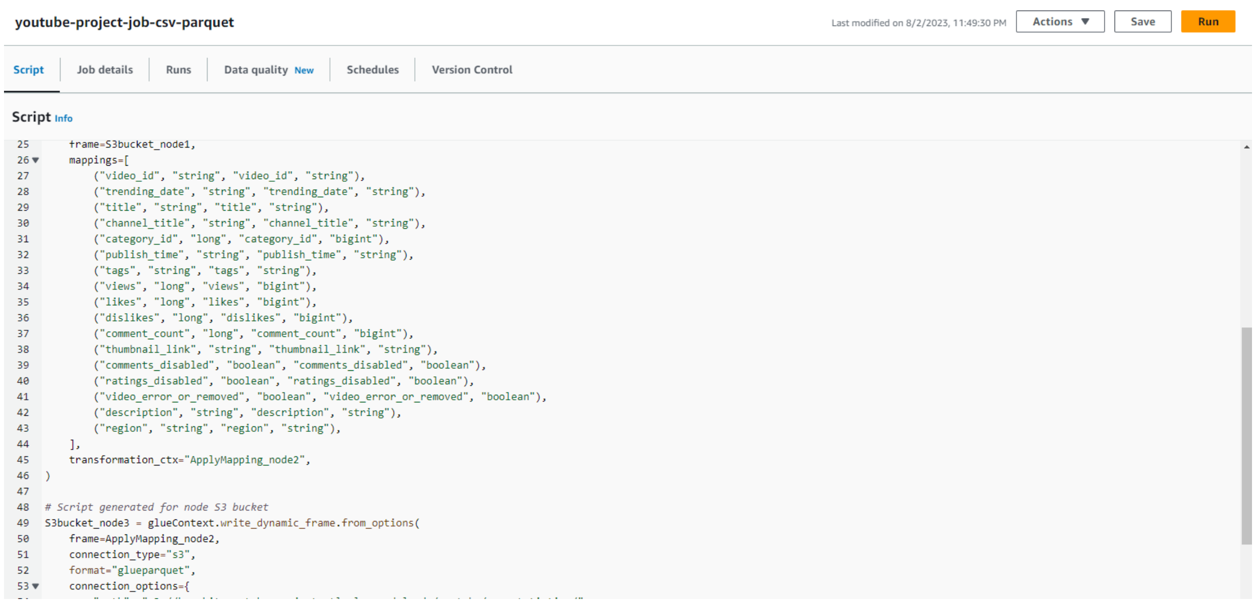
This data catalog, combined with the one in pipeline 1 should give Athena enough metadata to combine the csv files and the json turned parquet files from pipeline 1. But after running a join query between two schemas, I got an error stating that the data type for the join was not

compatible. Therefore I need to change the datatype for numeric columns which have string data type instead. I will also try to convert csv files to parquet for easy processing with Athena.
<u>Setting up Glue ETL Job:</u>
Next I needed to set up a ETL job which can be done with AWS Glue. By giving source bucket, IAM role, Transformation via visual job creator (converting data type), choosing destination target bucket where format will be Parquet format.

Here in the Glue ETL script I modified the automatically generated code so that the destination files are categorized with regions and then stored in the destination S3 cleaned bucket.

After doing the above changes, I was receiving an error because I couldn't convert the non english words in title columns from non english native regions.

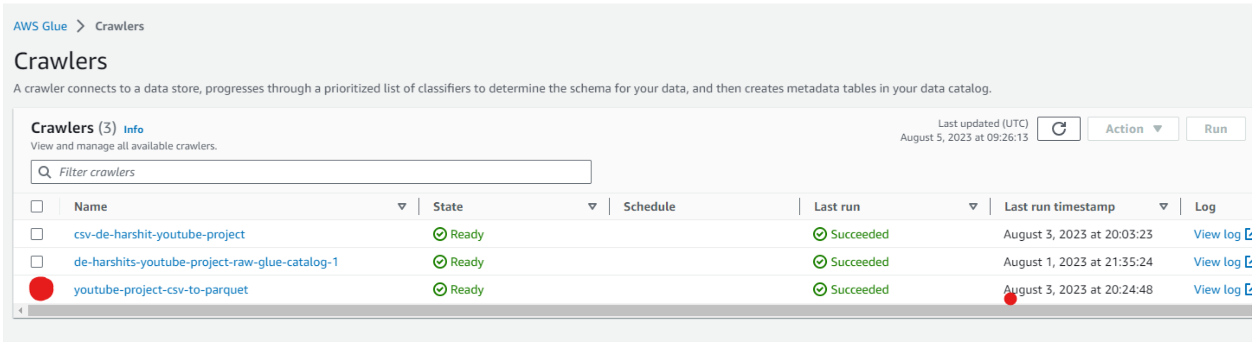
I utilized Bucket versioning in raw S3 bucket and soft deleted every files except ca(Canada),us (United States) and gb (Great Britain) which are english dominated region.



```
    frame=S3bucket_node1,
    mappings=[
        ("video_id", "string", "video_id", "string"),
        ("trending_date", "string", "trending_date", "string"),
        ("title", "string", "title", "string"),
        ("channel_title", "string", "channel_title", "string"),
        ("category_id", "long", "category_id", "bigint"),
        ("publish_time", "string", "publish_time", "string"),
        ("tags", "string", "tags", "string"),
        ("views", "long", "views", "bigint"),
        ("likes", "long", "likes", "bigint"),
        ("dislikes", "long", "dislikes", "bigint"),
        ("comment_count", "long", "comment_count", "bigint"),
        ("thumbnail_link", "string", "thumbnail_link", "string"),
        ("comments_disabled", "boolean", "comments_disabled", "boolean"),
        ("ratings_disabled", "boolean", "ratings_disabled", "boolean"),
        ("video_error_or_removed", "boolean", "video_error_or_removed", "boolean"),
        ("description", "string", "description", "string"),
        ("region", "string", "region", "string"),
    ],
    transformation_ctx="ApplyMapping_node2",
)

# Script generated for node S3 bucket
S3bucket_node3 = glueContext.write_dynamic_frame.from_options(
    frame=ApplyMapping_node2,
    connection_type="s3",
    format="glueparquet",
    connection_options={
```

Now after running this job successfully, my S3 cleaned bucket contains both json converted parquet from pipeline 1 and csv converted parquet with necessary data type transformations. To get a schema for this csv converted parquet files, I need to set up a crawler which will store schema of these files in data catalog.


AWS Glue > Crawlers

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 (3) Info

View and manage all available crawlers.

Last updated (UTC) August 5, 2023 at 09:26:13

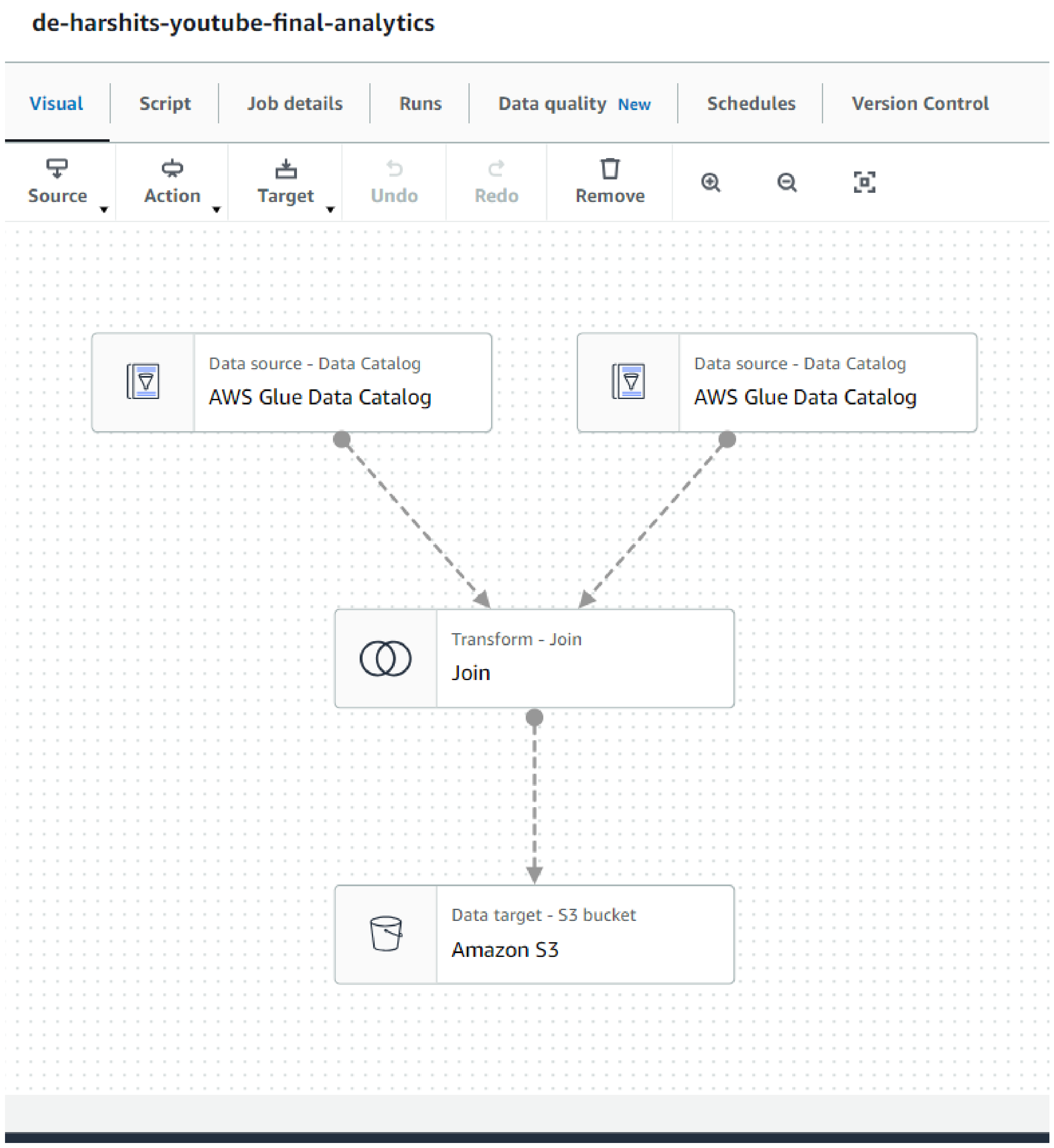
Name	State	Schedule	Last run	Last run timestamp	Log
csv-de-harshit-youtube-project	Ready		Succeeded	August 3, 2023 at 20:03:23	View log
de-harshits-youtube-project-raw-glue-catalog-1	Ready		Succeeded	August 1, 2023 at 21:35:24	View log
youtube-project-csv-to-parquet	Ready		Succeeded	August 3, 2023 at 20:24:48	View log

Pipeline 3:

Combining both parquet files and performing data transformation to categorize data with region then category_id.

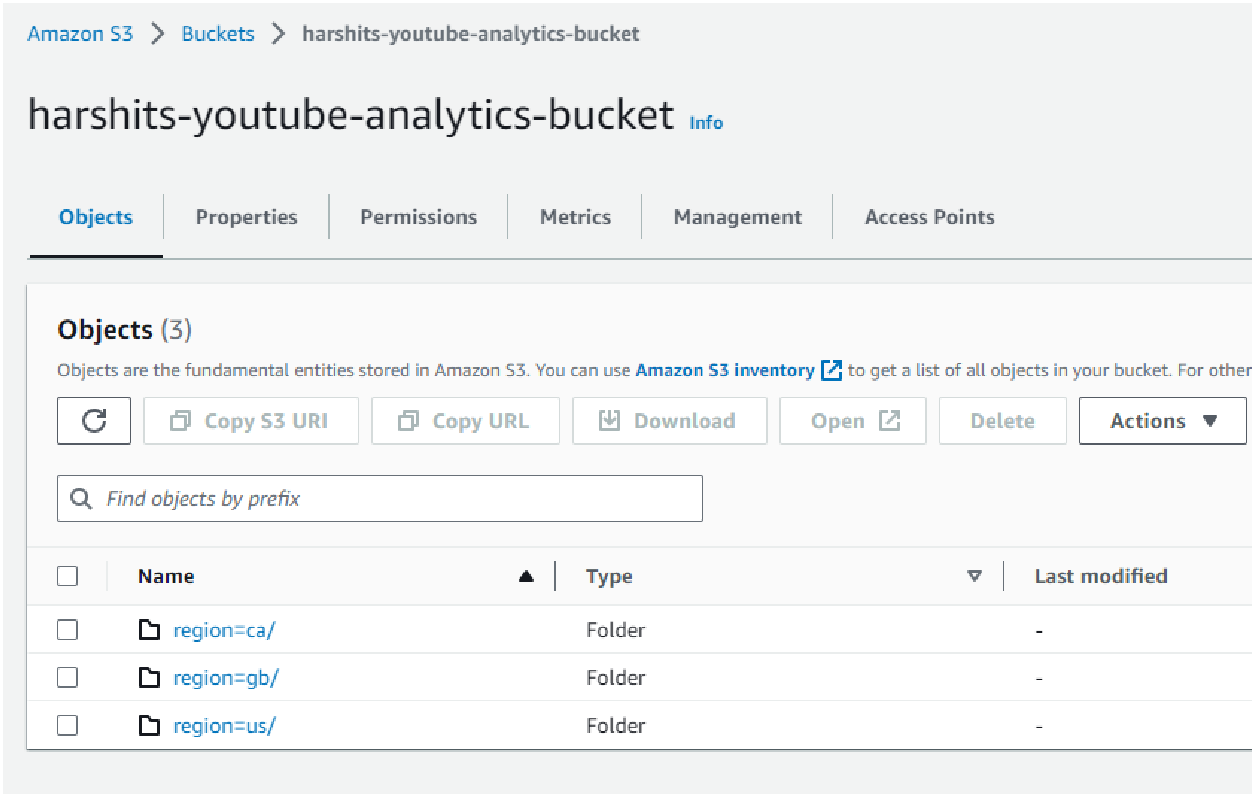
Now my S3 cleaned data contains both parquet files. I will need to combine these both and store in new S3 final bucket which categorizes files based on region and category_id. I will also include “add to catalog option” that will automatically create a data schema for my Final S3 bucket.

I will again use Glue ETL job to perform extract transform loading of data.
The visual from Glue ETL looks like this:



There are two catalogs one for each parquet files (csv and json converted). The transformation includes joining both schemas on category_id and id.

The data inserted in my New Final S3 bucket will look like this:

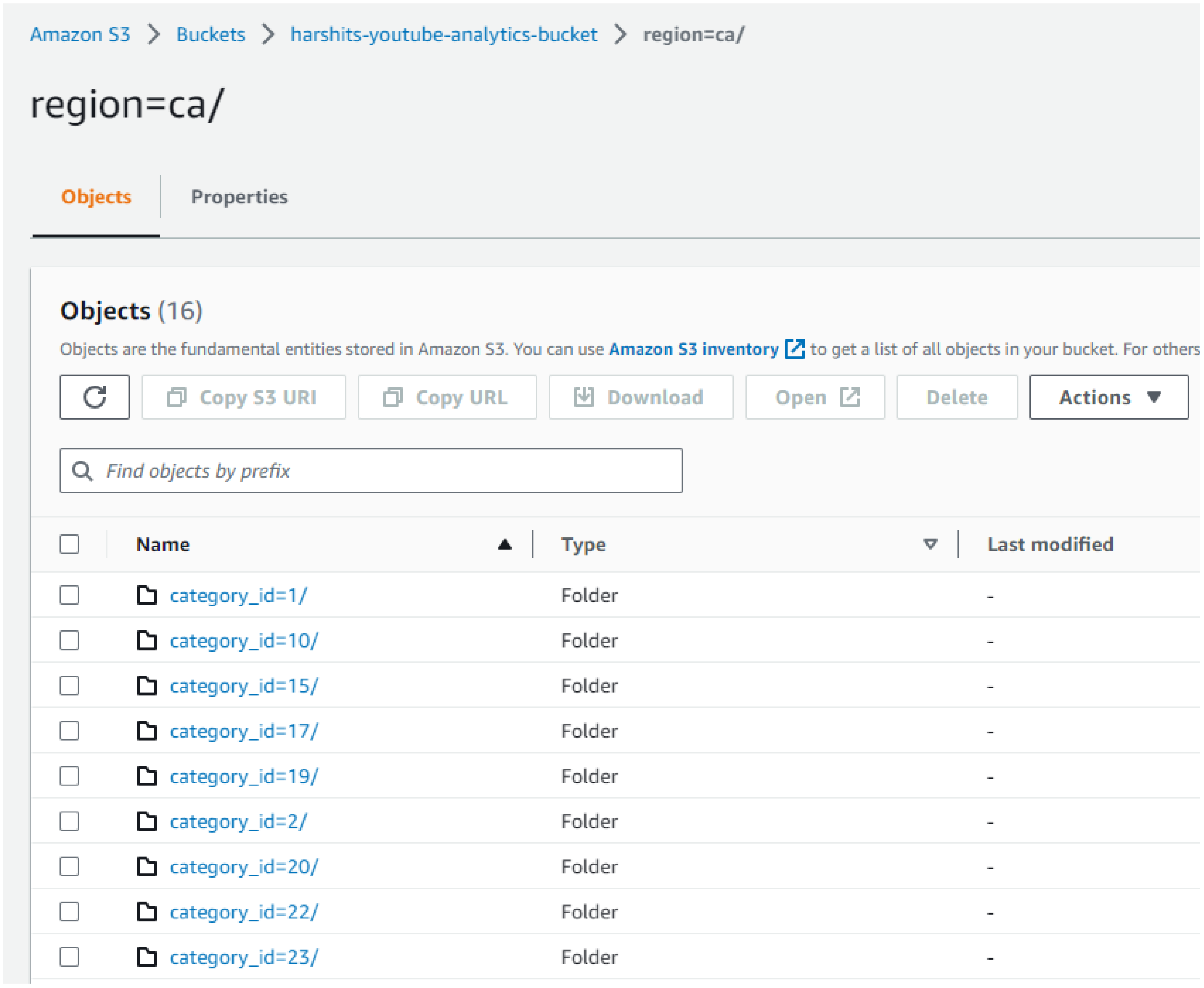


harshits-youtube-analytics-bucket Info

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For other

Name	Type	Last modified
region=ca/	Folder	-
region=gb/	Folder	-
region=us/	Folder	-


Amazon S3 > Buckets > harshits-youtube-analytics-bucket > region=ca/

region=ca/

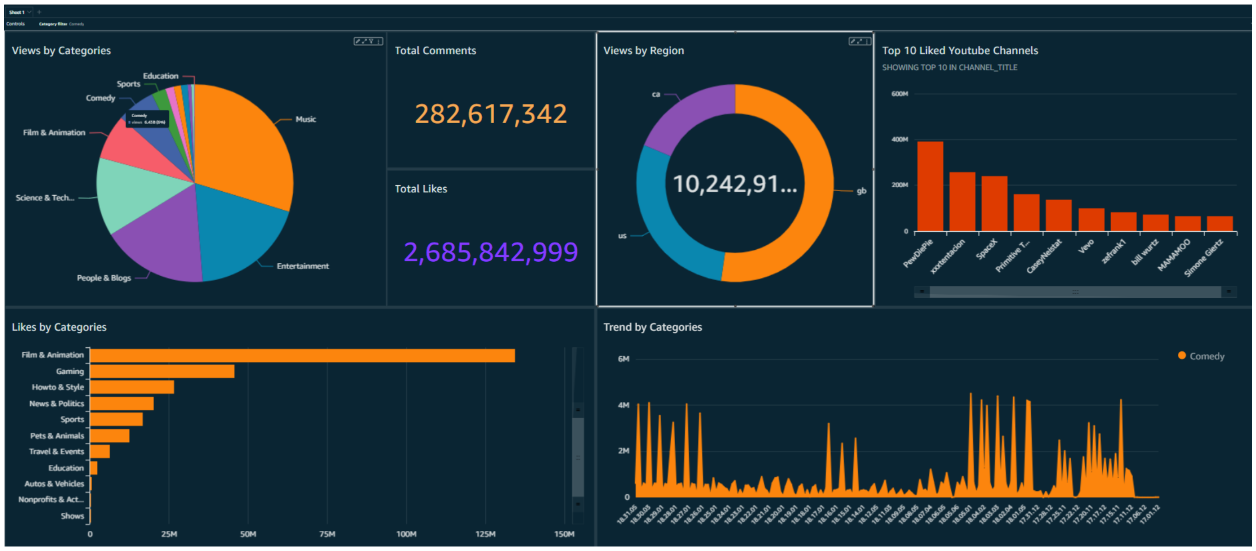
Objects (16)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others

Name	Type	Last modified
category_id=1/	Folder	-
category_id=10/	Folder	-
category_id=15/	Folder	-
category_id=17/	Folder	-
category_id=19/	Folder	-
category_id=2/	Folder	-
category_id=20/	Folder	-
category_id=22/	Folder	-
category_id=23/	Folder	-

Visualizing the data in Final S3 Bucket:
Using AWS Quicksight, I built a dashboard which helps deliver business insights and give visual representation. I connect Athena as my data source and now Quicksight can automatically query in Athena which has catalog on my final youtube analytics data.

Dashboard:

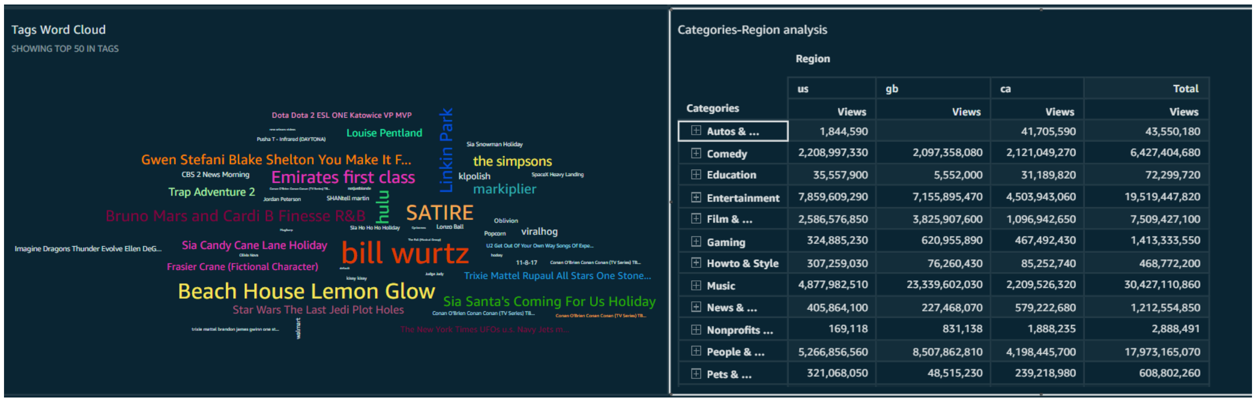


Total Comments

282,617,342

Total Likes

2,685,842,999


Tags Word Cloud
SHOWING TOP 50 IN TAGS
Dota Dota 2 ESL ONE Katowice VP MVP
Louise Pentland
Linkin Park
Gwen Stefani Blake Shelton You Make It F...
the simpsons
Emirates first class
markiplier
Trap Adventure 2
Bruno Mars and Cardi B Finesse R&B
hulu
SATIRE
viralhog
Imagine Dragons Thunder Evolve Ellen DeG...
Sia Candy Cane Lane Holiday
bill wurtz
Frasier Crane (Fictional Character)
Trixie Mattel Rupaul All Stars One Stone...
Beach House Lemon Glow
Sia Santa's Coming For Us Holiday
Star Wars The Last Jedi Plot Holes
The New York Times UFOs u.s. Navy jets m...

Categories-Region analysis

Categories	us	gb	ca	Total
	Views	Views	Views	Views
Autos & ...	1,844,590		41,705,590	43,550,180
Comedy	2,208,997,330	2,097,358,080	2,121,049,270	6,427,404,680
Education	35,557,900	5,552,000	31,189,820	72,299,720
Entertainment	7,859,609,290	7,155,895,470	4,503,943,060	19,519,447,820
Film & ...	2,586,576,850	3,825,907,600	1,096,942,650	7,509,427,100
Gaming	324,885,230	620,955,890	467,492,430	1,413,333,550
Howto & Style	307,259,030	76,260,430	85,252,740	468,772,200
Music	4,877,982,510	23,339,602,030	2,209,526,320	30,427,110,860
News & ...	405,864,100	227,468,070	579,222,680	1,212,554,850
Nonprofits ...	169,118	831,138	1,888,235	2,888,491
People & ...	5,266,856,560	8,507,862,810	4,198,445,700	17,973,165,070
Pets & ...	321,068,050	48,515,230	239,218,980	608,802,260