

Trending Youtube Video Analysis With AWS

TABLE OF CONTENTS

I.	Introduction.....	3
II.	Extracting data from source	3
III.	Transform the current data.....	6
1.	Moving csv data to a cleaner database.....	6
2.	Build a crawler to crawl the cleansed csv data from S3 bucket to AWS Glue database	7
3.	Add Trigger event in Lambda.....	7
IV.	Build a pipeline to get data for analytics.....	7
1.	Create an analytic job using visual ETL in AWS Glue	8
2.	Use the newly data to analyze with Quicksight on AWS	8
V.	Conclusion	9
VI.	REFERENCES	9

I. Introduction

YouTube (the world-famous video sharing website) maintains a list of the top trending videos on the platform. According to Variety magazine, “To determine the year’s top-trending videos, YouTube uses a combination of factors including measuring users interactions (number of views, shares, comments and likes).

This dataset is a daily record of the top trending YouTube videos. This dataset includes several months (and counting) of data on daily trending YouTube videos. Data is included for the US, GB, DE, CA, and FR regions (USA, Great Britain, Germany, Canada, and France, respectively), with up to 200 listed trending videos per day.

II. Extracting data from source

1. Upload data to S3 Bucket using AWS CLI

```
aws s3 cp . s3://youtube-deprojet/youtube/raw_statistics_reference_data/ --recursive --exclude "*" --include "*.json"
```

Then upload the csv file of each region to respective folders:

```
aws s3 cp CAvideos.csv s3://youtube-deprojet/youtube/raw_statistics/region=ca/
```

```
aws s3 cp DEvideos.csv s3://youtube-deprojet/youtube/raw_statistics/region=de/
```

```
aws s3 cp FRvideos.csv s3://youtube-deprojet/youtube/raw_statistics/region=fr/
```

```
aws s3 cp GBvideos.csv s3://youtube-deprojet/youtube/raw_statistics/region=gb/
```

```
aws s3 cp INvideos.csv s3://youtube-deprojet/youtube/raw_statistics/region=in/
```

```
aws s3 cp JPvideos.csv s3://youtube-deprojet/youtube/raw_statistics/region=jp/
```

```
aws s3 cp KRvideos.csv s3://youtube-deprojet/youtube/raw_statistics/region=kr/
```

```
aws s3 cp MXvideos.csv s3://youtube-deprojet/youtube/raw_statistics/region=mx/
```

```
aws s3 cp RUvideos.csv s3://youtube-deprojet/youtube/raw_statistics/region=ru/
```

```
aws s3 cp USvideos.csv s3://youtube-deprojet/youtube/raw_statistics/region=us/
```

2. Build crawler in AWS Glue

I created the crawler to crawl the data from S3 to a database in Glue.

youtube-deproject-glue-catalog

Last updated (UTC)
November 10, 2024 at 10:39:43

↻

Run crawler

Edit

Delete

Crawler properties

Name

youtube-deproject-glue-catalog

Description

-

Maximum table threshold

-

IAM role

youtube-deproject-s3-glue-role [↗](#)

Security configuration

-

Database

youtube_db

Lake Formation configuration

-

State

READY

Table prefix

-

▶ Advanced settings

Crawler runs

Schedule

Data sources

Classifiers

Tags

Crawler runs (1)

The list of crawler runs for this crawler.

🔍 Filter data

📅 Filter by a date and time range

< 1 > ⚙️

	Start time (UTC) ▲	End time (UTC) ▼	Current/last duration ▼	Status ▼	DPU hours ▼	Table changes ▼
○	November 6, 2024 at 14:18:4	November 6, 2024 at 14:19:4	01 min	✅ Completed	0.074	1 table change, 0 partition changes

3. Preprocessing the data using AWS Lambda

I created a function that will only take the third key in the json data (because that is the only thing we need).

youtube-deproject-lambda-json-parser

Throttle

📄 Copy ARN

Actions ▼

▼ Function overview

Info

Export to Application Composer

Download ▼

Diagram

Template

youtube-deproject-lambda-json-parser

Layers (1)

S3

+ Add trigger

+ Add destination

Description

-

Last modified

3 days ago

Function ARN

arn:aws:lambda:ap-southeast-2:571600839766:function:youtube-deproject-lambda-json-parser

Function URL

Info

After writing code, I configured the testing event so it will take the data from the data source we want.

```

    },
    "s3": {
      "s3SchemaVersion": "1.0",
      "configurationId": "testConfigRule",
      "bucket": {
        "name": "youtube-deprojet",
        "ownerIdentity": {
          "principalId": "EXAMPLE"
        },
        "arn": "arn:aws:s3:::youtube-deprojet"
      },
      "object": {
        "key": "youtube/raw_statistics_reference_data/CA_category_id.json",
        "size": 1024,
        "eTag": "0123456789abcdef0123456789abcdef",
        "sequencer": "0A1B2C3D4E5F678901"
      }
    }
  }
}

```

We also need to configure the variables inside the code so it can perform accordingly.

Environment variables (4)

The environment variables below are encrypted at rest with the default Lambda service key.

Key	Value
glue_catalog_db_name	youtube_cleaned_db
glue_catalog_table_name	cleaned_statistics_reference_data
s3_cleansed_layer	s3://youtube-de-project-cleansed/youtube
write_data_operation	append

After running, we will have a new cleansed database in our Glue.

[AWS Glue](#) > [Databases](#) > youtube_cleaned_db

youtube_cleaned_db

Last updated (UTC)
November 10, 2024 at 10:54:20

Edit

Delete

Database properties

Name youtube_cleaned_db	Description -	Location -	Created on (UTC) November 7, 2024 at 10:11:38
----------------------------	------------------	---------------	--

Tables (2)

Last updated (UTC)
November 10, 2024 at 10:54:21

Delete

Add tables using crawler

Add table

View and manage all available tables.

<

1

>

<input type="checkbox"/>	Name	Database	Location	Classification	Deprecated	View data	Data quality
<input type="checkbox"/>	cleaned_statistics_ref	youtube_cleaned_db	s3://youtube-de-proje	Parquet	-	Table data	View data quality
<input type="checkbox"/>	raw_statistics	youtube_cleaned_db	s3://youtube-de-proje	Parquet	-	Table data	View data quality

4. Load csv data to the table by using Crawler

youtube-deproject-crawler-csv

Last updated (UTC)
November 10, 2024 at 10:56:48

Run crawler

Edit

Delete

Crawler properties

Name

youtube-deproject-crawler-csv

IAM role

youtube-deproject-s3-glue-role

Database

youtube_db

State

READY

Description

-

Security configuration

-

Lake Formation configuration

-

Table prefix

-

Maximum table threshold

-

► Advanced settings

Crawler runs

Schedule

Data sources

Classifiers

Tags

Crawler runs (1)

The list of crawler runs for this crawler.

Filter data

Filter by a date and time range

< 1 >

Start time (UTC)

▲

End time (UTC)

▼

Current/last duration

▼

Status

▼

DPU hours

▼

Table changes

▼

○

November 7, 2024 at 15:28:3

November 7, 2024 at 15:29:2

53 s

Completed

0.085

1 table change, 10 partition changes

III. Transform the current data

1. Moving csv data to a cleaner database

Add a job in AWS Glue.

youtube-deproject-csv-to-parquet

Last modified on 11/8/2024, 4:05:30 PM

Actions ▼

Save

Run

Script

Job details

Runs

Data quality

Schedules

Version Control

Script

Info

```

1 import sys
2 from aws glue.transforms import *
3 from aws glue.utils import getResolvedOptions
4 from pyspark.context import SparkContext
5 from aws glue.context import GlueContext
6 from aws glue.job import Job
7
8 from aws glue.dynamicframe import DynamicFrame
9
10 ## @params: [JOB_NAME]
11 args = getResolvedOptions(sys.argv, ['JOB_NAME'])
12
13 sc = SparkContext()
14 glueContext = GlueContext(sc)
15 spark = glueContext.spark_session
16 job = Job(glueContext)
17 job.init(args['JOB_NAME'], args)
18 ## @type: DataSource
19 ## @args: [database = "db_youtube_raw", table_name = "raw_statistics", transformation_ctx = "datasource0"]
20 ## @return: datasource0
21 ## @inputs: []
22 predicate_pushdown = "region in ('ca','gb','us')"
23

```

This job will cover through all the data, apply data type based on the schema we specified and store the preprocessed data in a S3 bucket

2. Build a crawler to crawl the cleansed csv data from S3 bucket to AWS Glue database

youtube-deproject-csv-to-parquet

Last updated (UTC)
November 10, 2024 at 11:04:48

Run crawler

Edit

Delete

Crawler properties

Name

youtube-deproject-csv-to-parquet

Description

-

Maximum table threshold

-

IAM role

youtube-deproject-s3-glue-role

Security configuration

-

Database

youtube_cleaned_db

Lake Formation configuration

-

State

READY

Table prefix

-

Advanced settings

Crawler runs

Schedule

Data sources

Classifiers

Tags

Crawler runs (1)

The list of crawler runs for this crawler.

Filter data

Filter by a date and time range

< 1 >

Start time (UTC)

End time (UTC)

Current/last duration

Status

DPU hours

Table changes

November 8, 2024 at 09:25:00

November 8, 2024 at 09:25:00

53 s

Completed

0.075

1 table change, 3 partition changes

3. Add Trigger event in Lambda

A trigger is created to make this pipeline automated. When a new file is added to the original S3 bucket, the lambda function will run right after for that data.

Triggers (1)

Info

Fix errors

Edit

Delete

Add trigger

Find triggers

< 1 >

Trigger

S3: youtube-deproject

arn:aws:s3:::youtube-deproject

Details

Bucket arn: arn:aws:s3:::youtube-deproject

Event types: s3:ObjectCreated:*

isComplexStatement: No

Notification name: 6bb9fbf9-3ca0-4990-b4c1-c90478f476e7

Prefix: youtube/raw_statistics_reference_data/

Service principal: s3.amazonaws.com

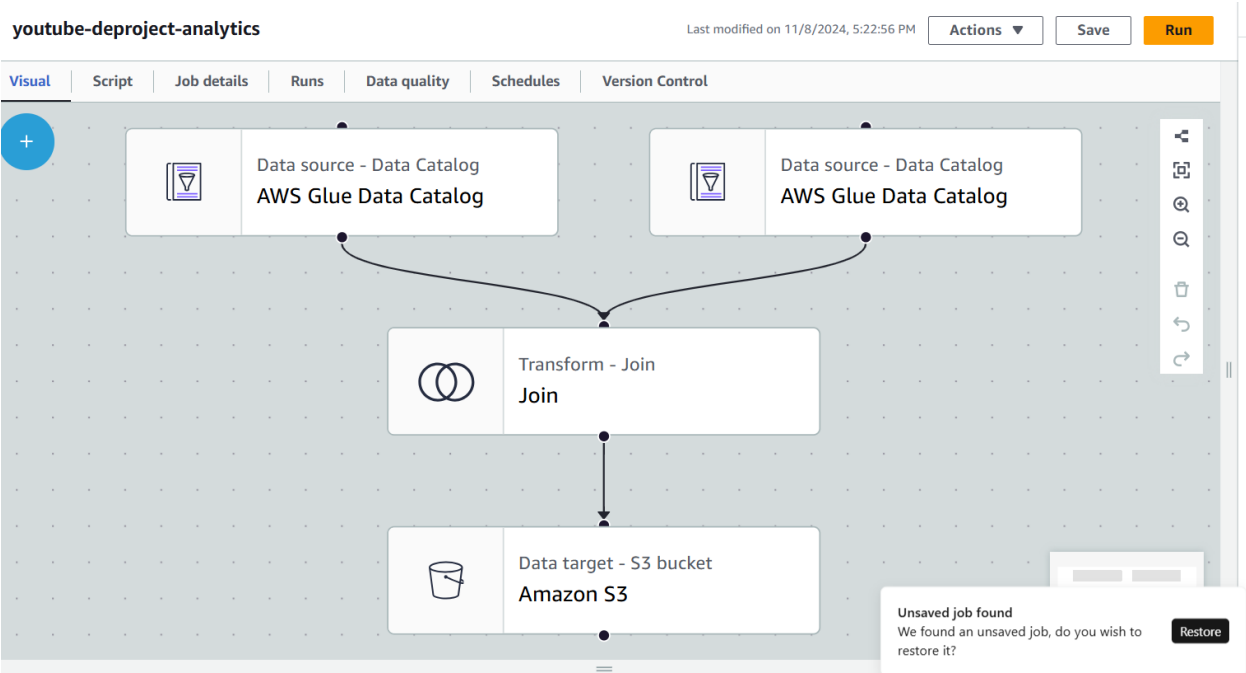
Source account: 571600839766

Statement ID: lambda-c3ddb44a-0f0d-4022-9baf-a32915fe76cf

Suffix: .json

IV. Build a pipeline to get data for analytics

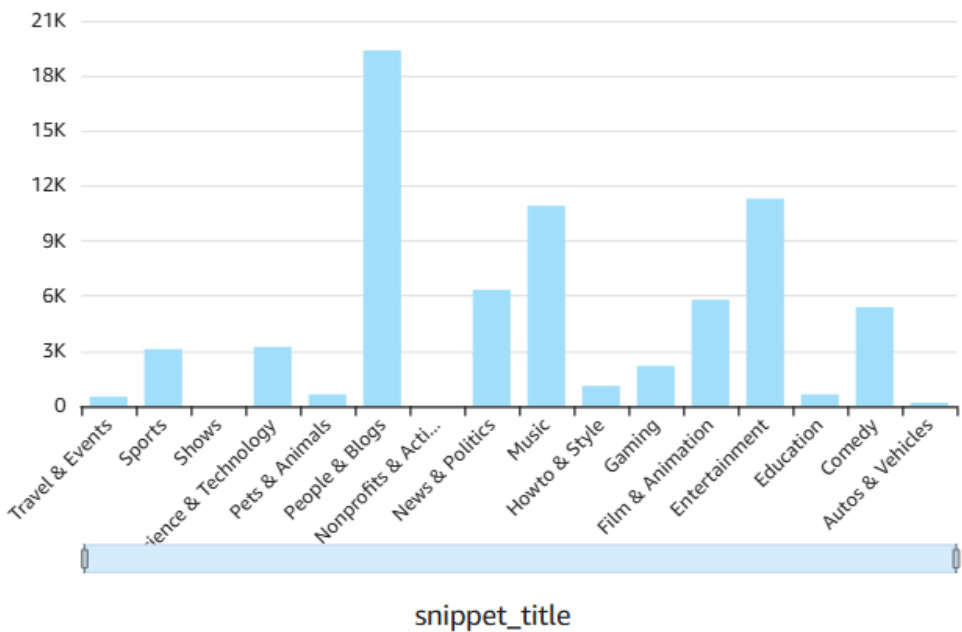
1. Create an analytic job using visual ETL in AWS Glue



2. Use the newly data to analyze with Quicksight on AWS

Build a simple visual to look through the key insight inside the data.

Number of dislike based on snippet title



V. Conclusion

- Challenges:
 - Handling inconsistencies in the text data, such as special characters and stopwords, was challenging.
 - Learning the architecture of AWS and the services it offers.
 - Visualize and contextualize the overall work pipeline.
- Summary:
 - This project is to implement a simple data pipeline using AWS that help contribute to my understanding the real word data process.
 - Learned effective techniques for implementing and optimizing the services.

VI. References

<https://www.kaggle.com/datasets/datasnaek/youtube-new>

<https://www.youtube.com/watch?v=yZKJFKu49Dk&list=PLBJe2dFI4sgvQTNNkI3ETYJgNPR4CBpFd>