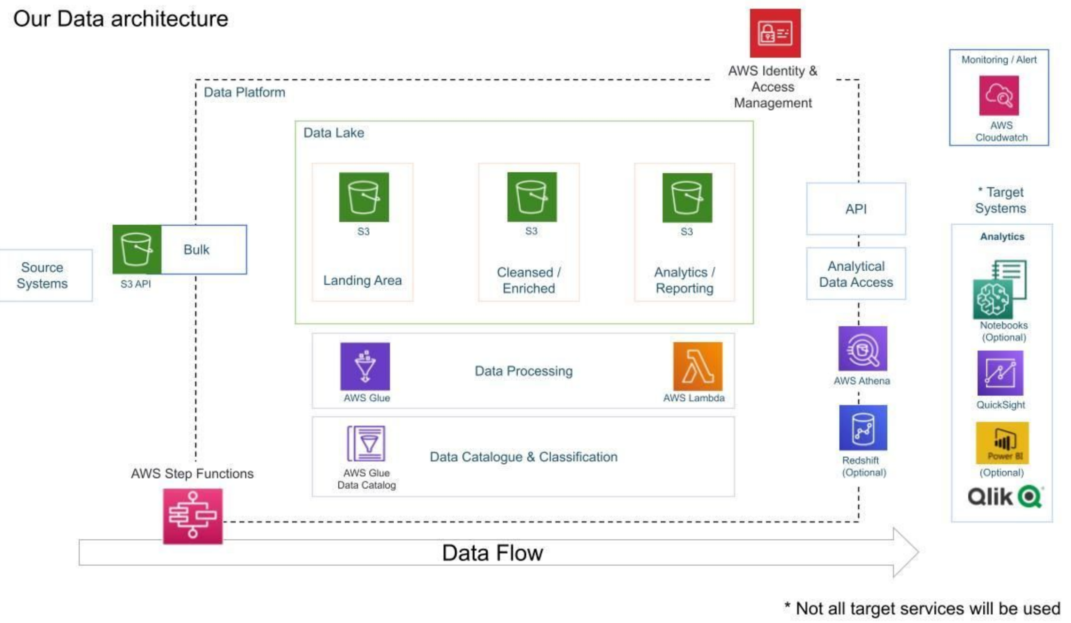
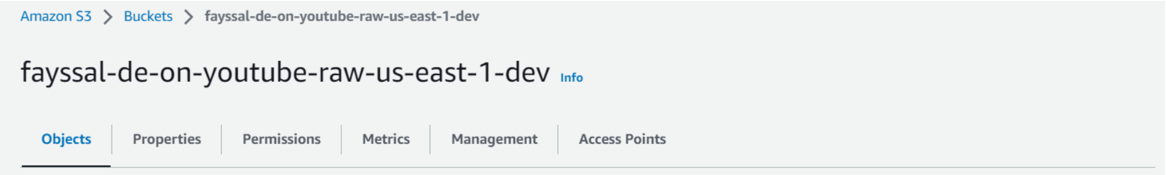
📌

YouTube Data Analysis



1. Get Our Data

- Download from kaggle.com/datasnaek/youtube-new
- Create an Amazon S3 bucket, for our landing bucket



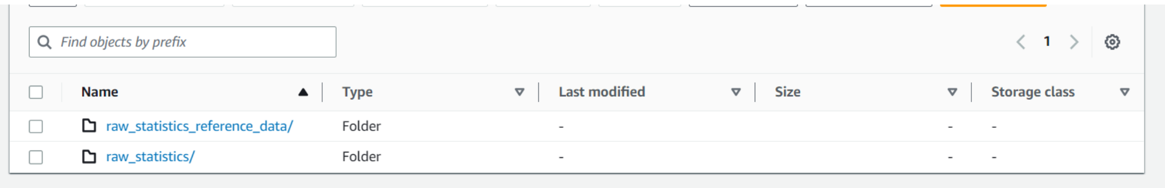
Amazon S3 > Buckets > fayssal-de-on-youtube-raw-us-east-1-dev

fayssal-de-on-youtube-raw-us-east-1-dev Info

Objects | Properties | Permissions | Metrics | Management | Access Points

- Copy the data to S3, using our AWS CL

```
# To copy all JSON Reference data to same location:
aws s3 cp . s3://fayssal-de-on-youtube-raw-us-east-1-dev/youtube/raw_statistics_reference_data/ --recursive --exclude "*" --incluc

# To copy all data files to its own location, following Hive-style patterns:
aws s3 cp CAvideos.csv s3://fayssal-de-on-youtube-raw-us-east-1-dev/youtube/raw_statistics/region=ca/
aws s3 cp DEvideos.csv s3://fayssal-de-on-youtube-raw-us-east-1-dev/youtube/raw_statistics/region=de/
aws s3 cp FRvideos.csv s3://fayssal-de-on-youtube-raw-us-east-1-dev/youtube/raw_statistics/region=fr/
aws s3 cp GBvideos.csv s3://fayssal-de-on-youtube-raw-us-east-1-dev/youtube/raw_statistics/region=gb/
aws s3 cp INvideos.csv s3://fayssal-de-on-youtube-raw-us-east-1-dev/youtube/raw_statistics/region=in/
aws s3 cp JPvideos.csv s3://fayssal-de-on-youtube-raw-us-east-1-dev/youtube/raw_statistics/region=jp/
aws s3 cp KRvideos.csv s3://fayssal-de-on-youtube-raw-us-east-1-dev/youtube/raw_statistics/region=kr/
aws s3 cp MXvideos.csv s3://fayssal-de-on-youtube-raw-us-east-1-dev/youtube/raw_statistics/region=mx/
aws s3 cp RUvideos.csv s3://fayssal-de-on-youtube-raw-us-east-1-dev/youtube/raw_statistics/region=ru/
aws s3 cp USvideos.csv s3://fayssal-de-on-youtube-raw-us-east-1-dev/youtube/raw_statistics/region=us/
```

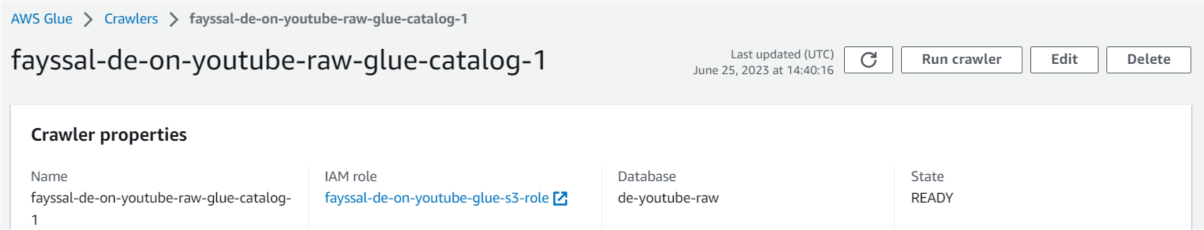


□	Name	Type	Last modified	Size	Storage class
□	raw_statistics_reference_data/	Folder	-	-	-
□	raw_statistics/	Folder	-	-	-

2. Build Glue Crawler and Catalog for a JSON file

1. Create The Crawler

We create the **crawler** with the data source as `s3://fayssal-de-on-youtube-raw-us-east-1-dev/youtube/raw_statistics_reference_data`, and then proceed to extract insights from the collected data.


AWS Glue > Crawlers > fayssal-de-on-youtube-raw-glue-catalog-1
Run crawler
Edit
Delete

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

Last updated (UTC)
June 25, 2023 at 14:40:16

Crawler properties

Name: fayssal-de-on-youtube-raw-glue-catalog-1

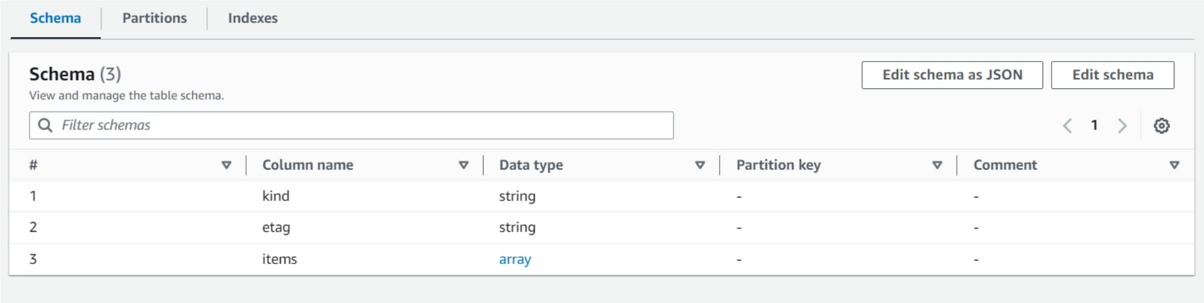
IAM role: fayssal-de-on-youtube-glue-s3-role

Database: de-youtube-raw

State: READY

2. The Catalog table

After the crawler runs, it creates a table in the catalog that contains metadata about the target S3 bucket.



Schema (3)

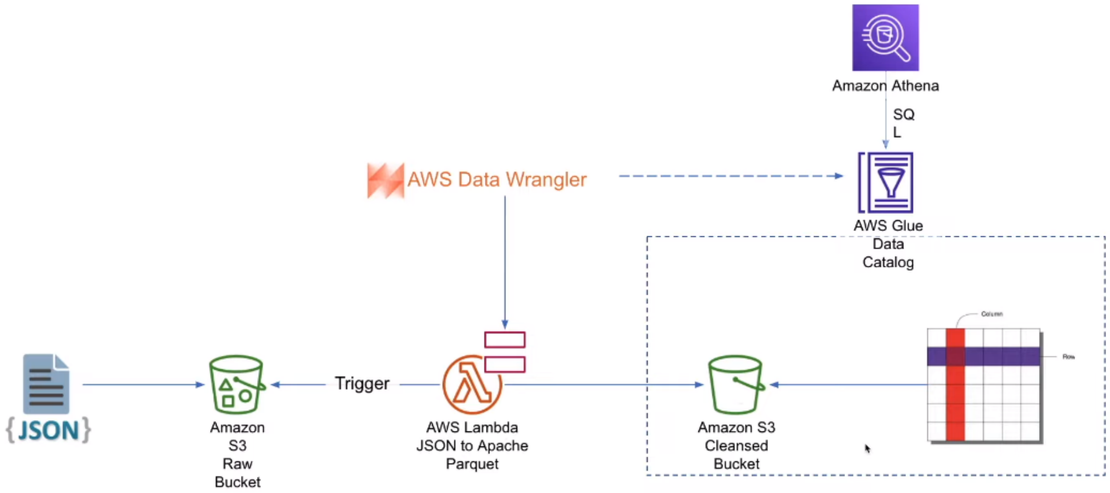
View and manage the table schema.

#	Column name	Data type	Partition key	Comment
1	kind	string	-	-
2	etag	string	-	-
3	items	array	-	-

But Athena will not be able to query this table as it doesn't have the actual column we want to query. Therefore, we need to perform some preprocessing on the JSON files before actually creating the crawler.

3. Preprocessing Data : Data Cleansing

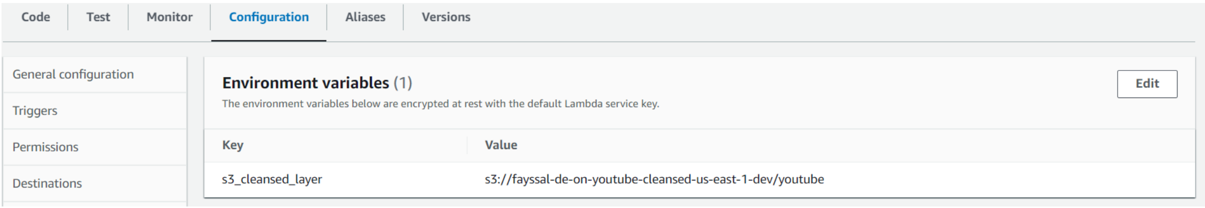
So our goal is to extract the necessary column from JSON files and transform the files from JSON format to Parquet format.



1. Writing ETL Job In Lambda and Cleaning Data

```
import json

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

# Temporary hard-coded AWS Settings; i.e. to be set as OS variable in Lambda
os_input_s3_cleansed_layer = os.environ['s3_cleansed_layer']

def lambda_handler(event, context):
    # Get the object from the event and show its content type
    bucket = event['Records'][0]['s3']['bucket']['name']
    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
        )

        return wr_response
    except Exception as e:
        print(e)
        print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as this function.'.f
        raise e
```

For the **Environment variables** :

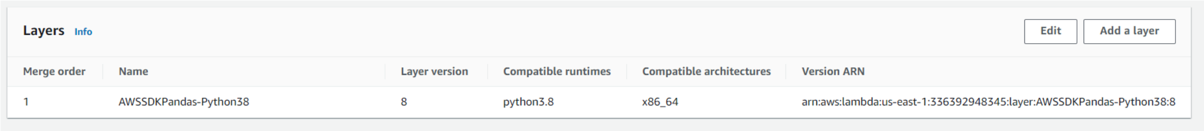


Environment variables (1)

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

Key	Value
s3_cleansed_layer	s3://fayssal-de-on-youtube-cleansed-us-east-1-dev/youtube

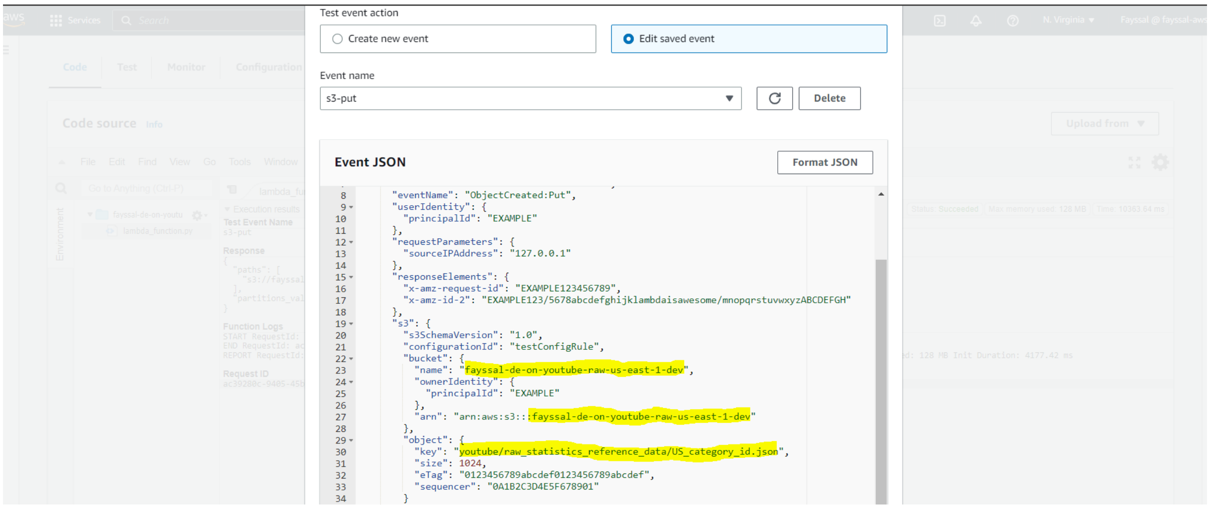
Adding a **Layer** to AWS Lambda :



Layers Info

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
1	AWSSDKPandas-Python38	8	python3.8	x86_64	arn:aws:lambda:us-east-1:336392948345:layer:AWSSDKPandas-Python38:8

Testing the Lambda function :



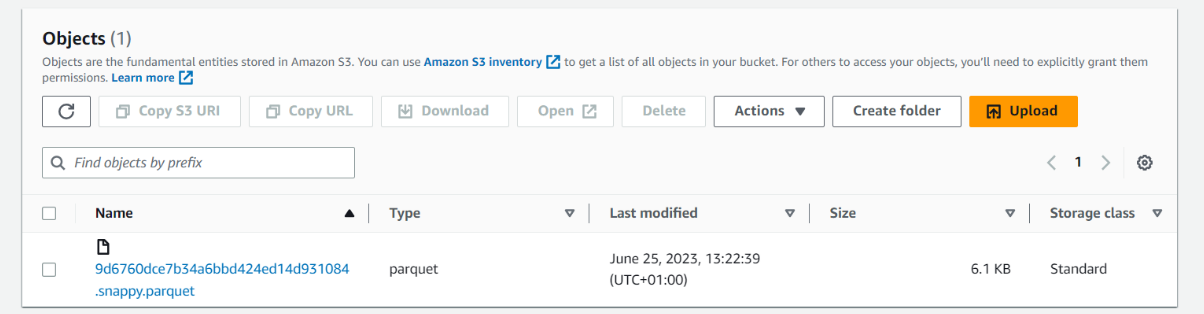
Test event action

○ Create new event ● Edit saved event

Event name: s3-put

```
      "eventName": "ObjectCreated:Put",
      "userIdentity": {
        "principalId": "EXAMPLE"
      },
      "requestParameters": {
        "sourceIPAddress": "127.0.0.1"
      },
      "responseElements": {
        "x-amz-request-id": "EXAMPLE123456789",
        "x-amz-id-2": "EXAMPLE123/5678abcdefghijklambdaisawesome/mnopqrstuvwxyzABCDEFGH"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "fayssal-de-on-youtube-raw-us-east-1-dev",
          "ownerIdentity": {
            "principalId": "EXAMPLE"
          },
          "arn": "arn:aws:s3:::fayssal-de-on-youtube-raw-us-east-1-dev"
        },
        "object": {
          "key": "youtube/raw_statistics_reference_data/US_category_id.json",
          "size": 1024,
          "eTag": "0123456789abcdef0123456789abcdef",
          "sequencer": "0A1B2C3D4E5F678901"
        }
```

The result :

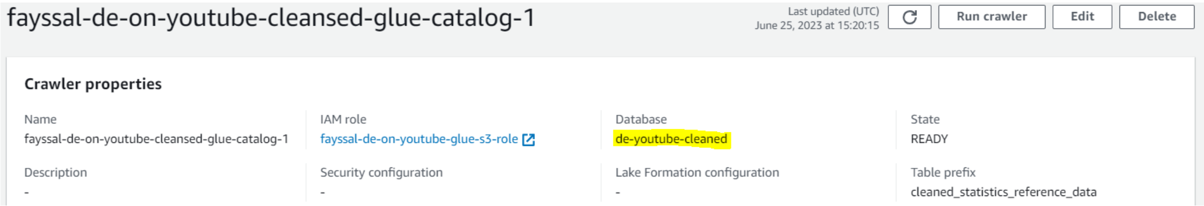


Objects (1)

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 to access your objects, you'll need to explicitly grant them permissions. **Learn more**

□	Name	Type	Last modified	Size	Storage class
□	9d6760dce7b34a6bbd424ed14d931084.snappy.parquet	parquet	June 25, 2023, 13:22:39 (UTC+01:00)	6.1 KB	Standard

2. Creating a Crawler for the cleansed data



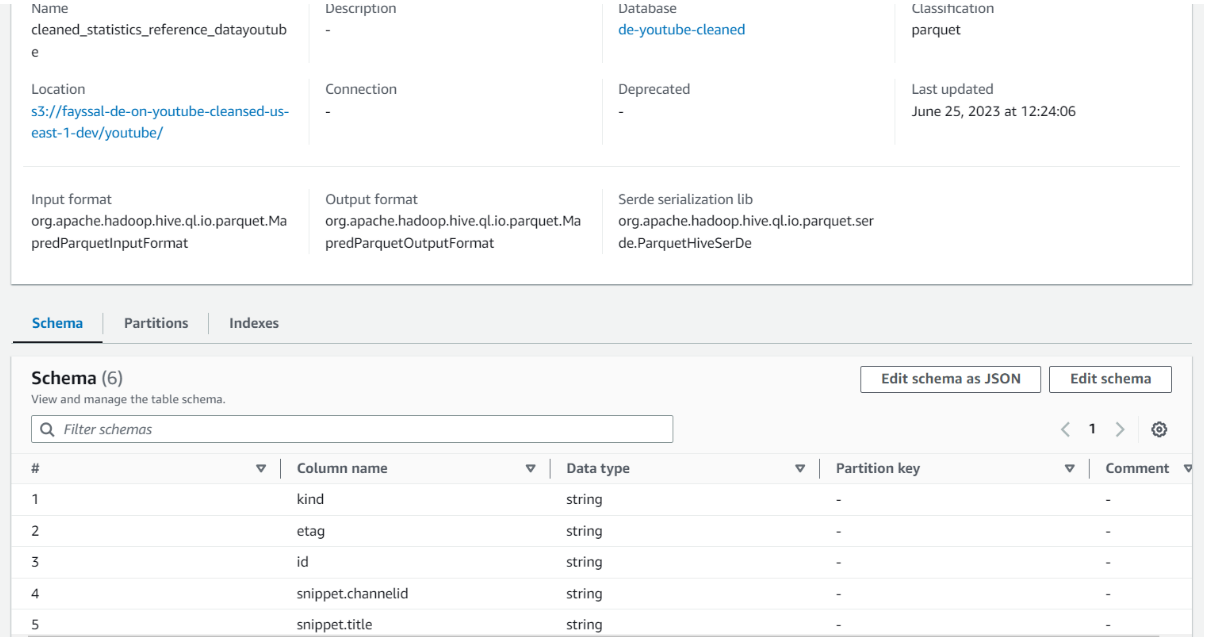
fayssal-de-on-youtube-cleansed-glue-catalog-1

Last updated (UTC)
June 25, 2023 at 15:20:15

Crawler properties

Name	IAM role	Database	State
fayssal-de-on-youtube-cleansed-glue-catalog-1	fayssal-de-on-youtube-glue-s3-role	de-youtube-cleaned	READY
Description	**Security configuration**	**Lake Formation configuration**	**Table prefix**
-	-	-	cleaned_statistics_reference_data

And this is the result :

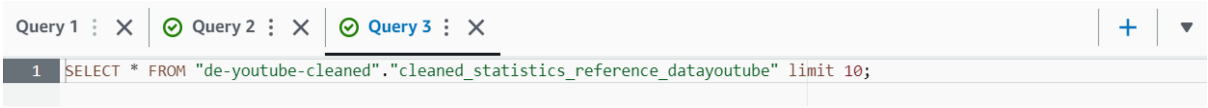


Name cleaned_statistics_reference_datayoutube	Description -	Database de-youtube-cleaned	Classification parquet
Location s3://fayssal-de-on-youtube-cleansed-us-east-1-dev/youtube/	Connection -	Deprecated -	Last updated June 25, 2023 at 12:24:06
Input format org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat	Output format org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat	Serde serialization lib org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe	

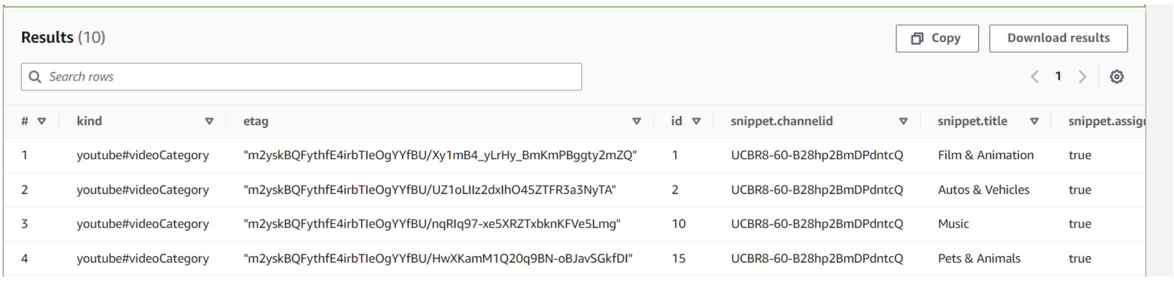
Schema (6)
View and manage the table schema.

#	Column name	Data type	Partition key	Comment
1	kind	string	-	-
2	etag	string	-	-
3	id	string	-	-
4	snippet.channelid	string	-	-
5	snippet.title	string	-	-

3. Run queries against it in Athena



```
SELECT * FROM "de-youtube-cleaned"."cleaned_statistics_reference_datayoutube" limit 10;
```

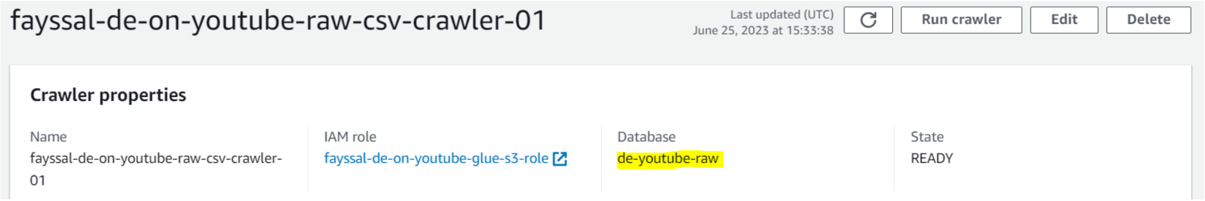


Results (10)

#	kind	etag	id	snippet.channelid	snippet.title	snippet.assig
1	youtube#videoCategory	"m2yskBQFythfE4irbTIeOgYYfBU/Xy1mB4_yLrHy_BmKmPBggty2mZQ"	1	UCBR8-60-B28hp2BmDPdntcQ	Film & Animation	true
2	youtube#videoCategory	"m2yskBQFythfE4irbTIeOgYYfBU/UZ1oLIIz2dxIhO45ZTFR3a3NyTA"	2	UCBR8-60-B28hp2BmDPdntcQ	Autos & Vehicles	true
3	youtube#videoCategory	"m2yskBQFythfE4irbTIeOgYYfBU/nqRIq97-xe5XRZTxbknKFVe5Lmg"	10	UCBR8-60-B28hp2BmDPdntcQ	Music	true
4	youtube#videoCategory	"m2yskBQFythfE4irbTIeOgYYfBU/HwXKamM1Q20q9BN-oBJavSGkfDI"	15	UCBR8-60-B28hp2BmDPdntcQ	Pets & Animals	true

4. Build Glue Crawler and Catalog for a CSV files

1. Create The Crawler



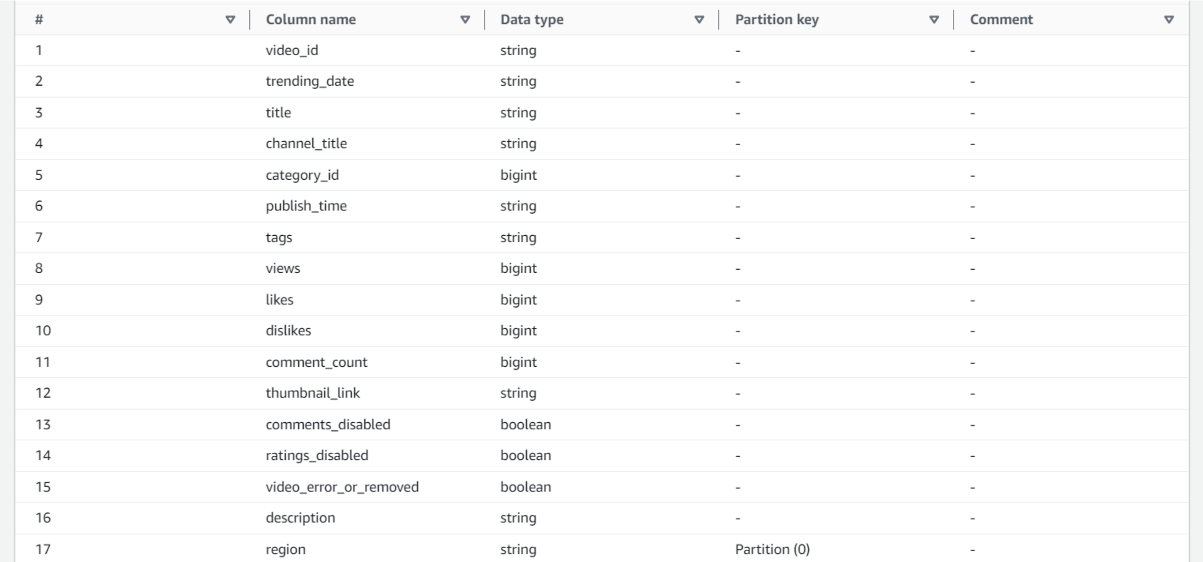
fayssal-de-on-youtube-raw-csv-crawler-01

Last updated (UTC)
June 25, 2023 at 15:33:38

Crawler properties

Name fayssal-de-on-youtube-raw-csv-crawler-01	IAM role fayssal-de-on-youtube-glue-s3-role	Database de-youtube-raw	State READY

2. The Catalog table



#	Column name	Data type	Partition key	Comment
1	video_id	string	-	-
2	trending_date	string	-	-
3	title	string	-	-
4	channel_title	string	-	-
5	category_id	bigint	-	-
6	publish_time	string	-	-
7	tags	string	-	-
8	views	bigint	-	-
9	likes	bigint	-	-
10	dislikes	bigint	-	-
11	comment_count	bigint	-	-
12	thumbnail_link	string	-	-
13	comments_disabled	boolean	-	-
14	ratings_disabled	boolean	-	-
15	video_error_or_removed	boolean	-	-
16	description	string	-	-
17	region	string	Partition (0)	-

3. Joining the cleaned table and the raw statistics table in Athena

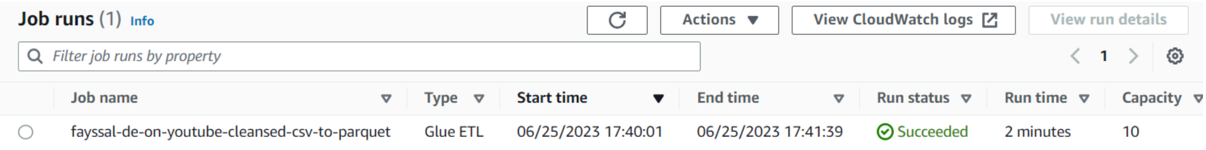
```
SELECT * FROM "de-youtube-raw"."raw_statistics" a
INNER JOIN "de-youtube-cleaned"."cleaned_statistics_reference_datayoutube" b on a.category_id=b.id ;
```

5. Schema Change using a ETL Job : CSV → Parquet

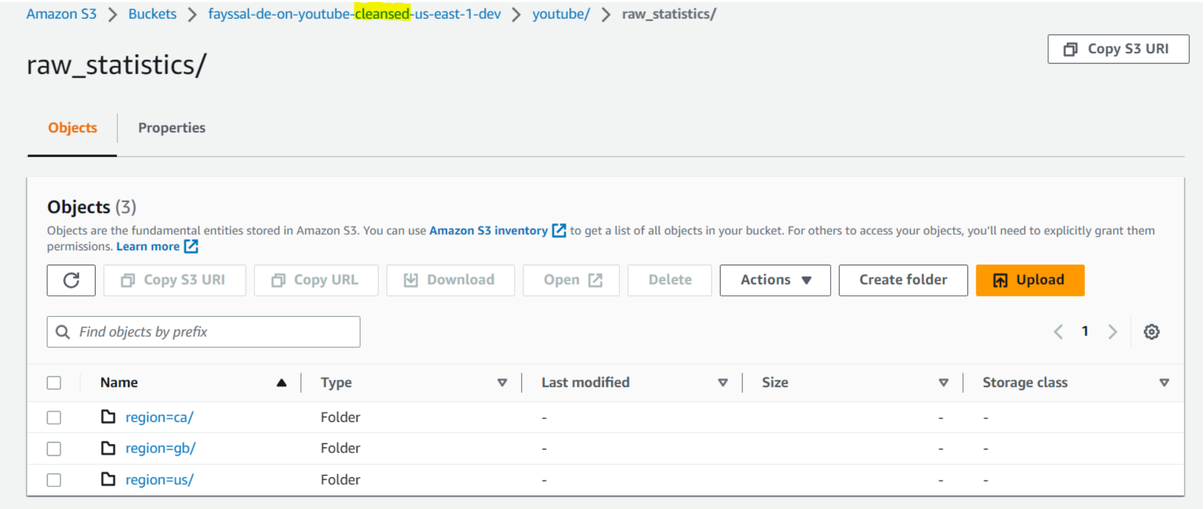
This job takes the CSV data from the catalog table, transforms it into the Parquet format, and puts it in an S3 bucket.

Job Script :

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

args = getResolvedOptions(sys.argv, ["JOB_NAME"])
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args["JOB_NAME"], args)

predicate_pushdown = "region in ('ca','gb','us')"

# Script generated for node raw_statistics_table
raw_statistics_table_node1 = glueContext.create_dynamic_frame.from_catalog(
    database="de-youtube-raw",
    table_name="raw_statistics",
    transformation_ctx="raw_statistics_table_node1",
    push_down_predicate = predicate_pushdown,
)

# Script generated for node ApplyMapping
ApplyMapping_node2 = ApplyMapping.apply(
    frame=raw_statistics_table_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={
        "path": "s3://fayssal-de-on-youtube-cleansed-us-east-1-dev/youtube/raw_statistics/",
        "partitionKeys": ["region"],
    },
    transformation_ctx="S3bucket_node3",
)

job.commit()
```



	Job name	Type	Start time	End time	Run status	Run time	Capacity
○	fayssal-de-on-youtube-cleansed-csv-to-parquet	Glue ETL	06/25/2023 17:40:01	06/25/2023 17:41:39	✓ Succeeded	2 minutes	10



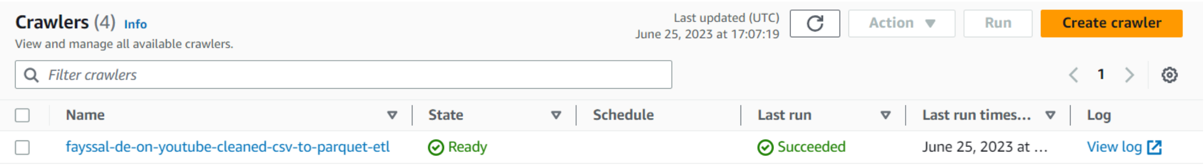
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	region=ca/	Folder	-	-	-
□	region=gb/	Folder	-	-	-
□	region=us/	Folder	-	-	-

Create a Glue Crawl

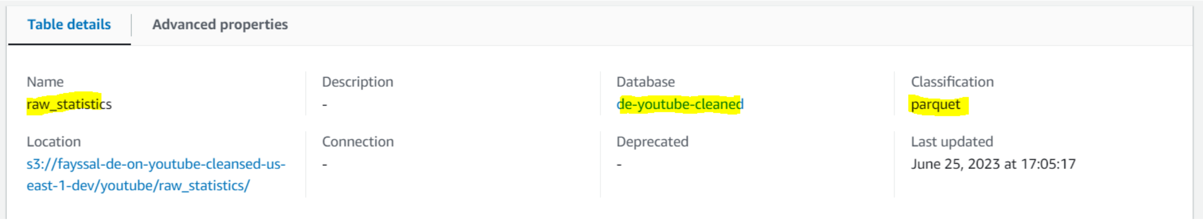
So now let's create a glue crawl on top of this bucket

> 💡 This process enables easier data exploration, querying, and integration with other AWS services. Once the crawler has completed its task, the **metadata tables in the AWS Glue Data Catalog** can be used for querying and analysis using services like **AWS Athena** or **AWS Glue ETL jobs**.



View and manage all available crawlers.

□	Name	State	Schedule	Last run	Last run times...	Log
□	fayssal-de-on-youtube-cleaned-csv-to-parquet-etl	✓ Ready		✓ Succeeded	June 25, 2023 at ...	View log



Name
raw_statistics

Description
-

Database
de-youtube-cleaned

Classification
parquet

Location
s3://fayssal-de-on-youtube-cleansed-us-east-1-dev/youtube/raw_statistics/

Connection
-

Deprecated
-

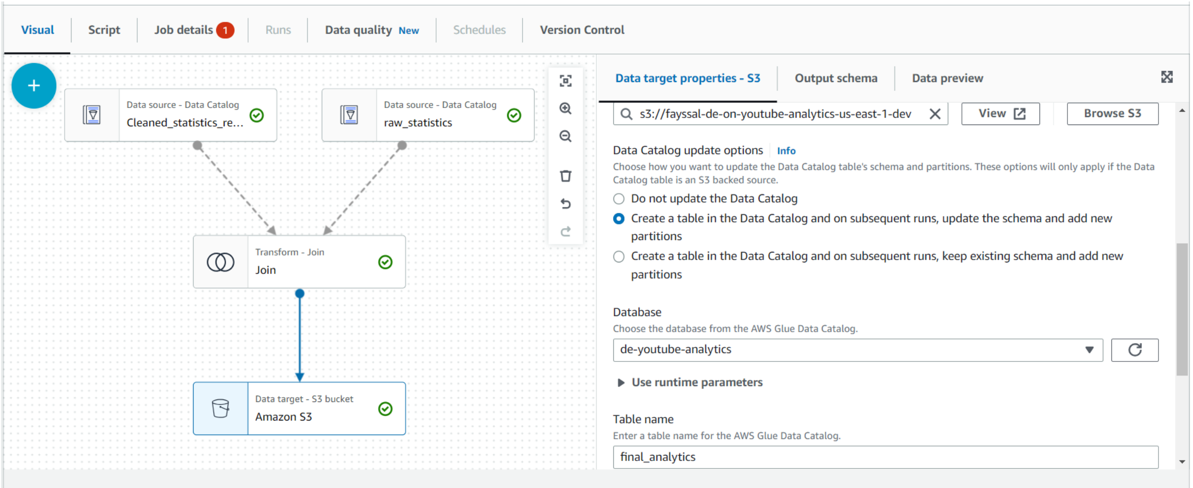
Last updated
June 25, 2023 at 17:05:17

6. Building ETL Pipeline

Instead of joining tables like this every time:

```
select * from "de-youtube-cleaned"."raw_statistics" a
INNER JOIN "de-youtube-cleaned"."cleaned_statistics_reference_datayoutube" b a.category_id=b.id;
```

We will create an ETL pipeline that consolidates them into a single S3 bucket.



s3://fayssal-de-on-youtube-analytics-us-east-1-dev

Data Catalog update options Info

Choose how you want to update the Data Catalog table's schema and partitions. These options will only apply if the Data Catalog table is an S3 backed source.

○ Do not update the Data Catalog

● Create a table in the Data Catalog and on subsequent runs, update the schema and add new partitions

○ Create a table in the Data Catalog and on subsequent runs, keep existing schema and add new partitions

Database

Choose the database from the AWS Glue Data Catalog.

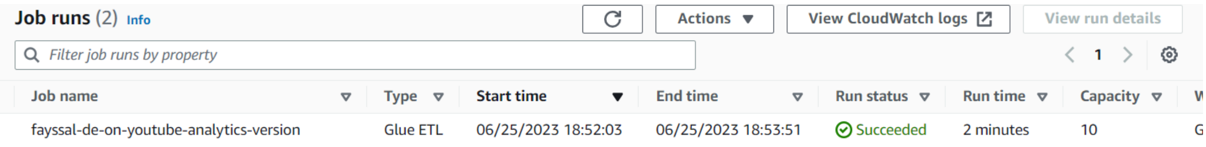
de-youtube-analytics

▶ Use runtime parameters

Table name

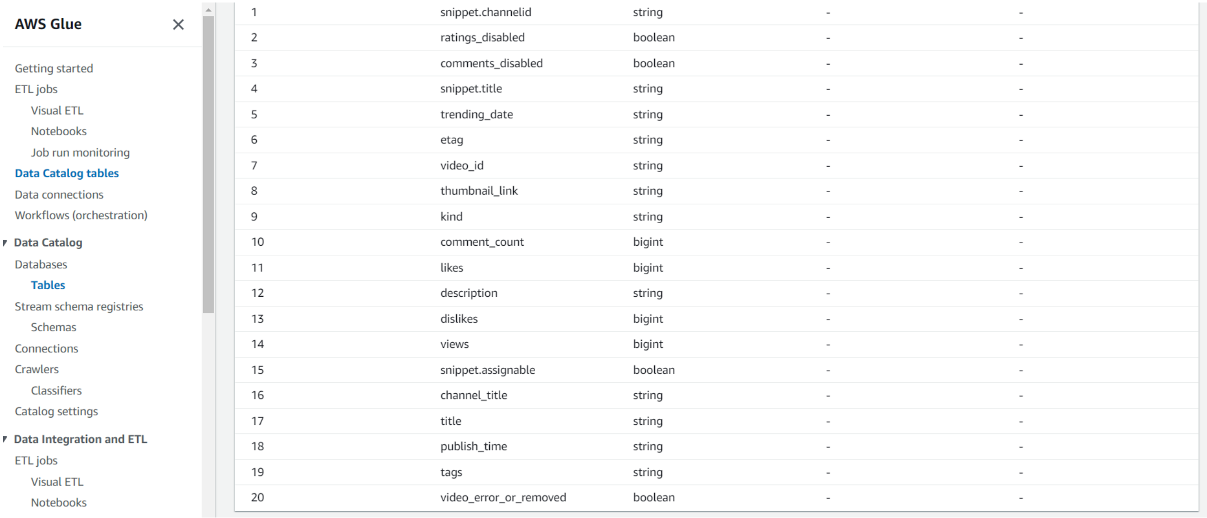
Enter a table name for the AWS Glue Data Catalog.

final_analytics



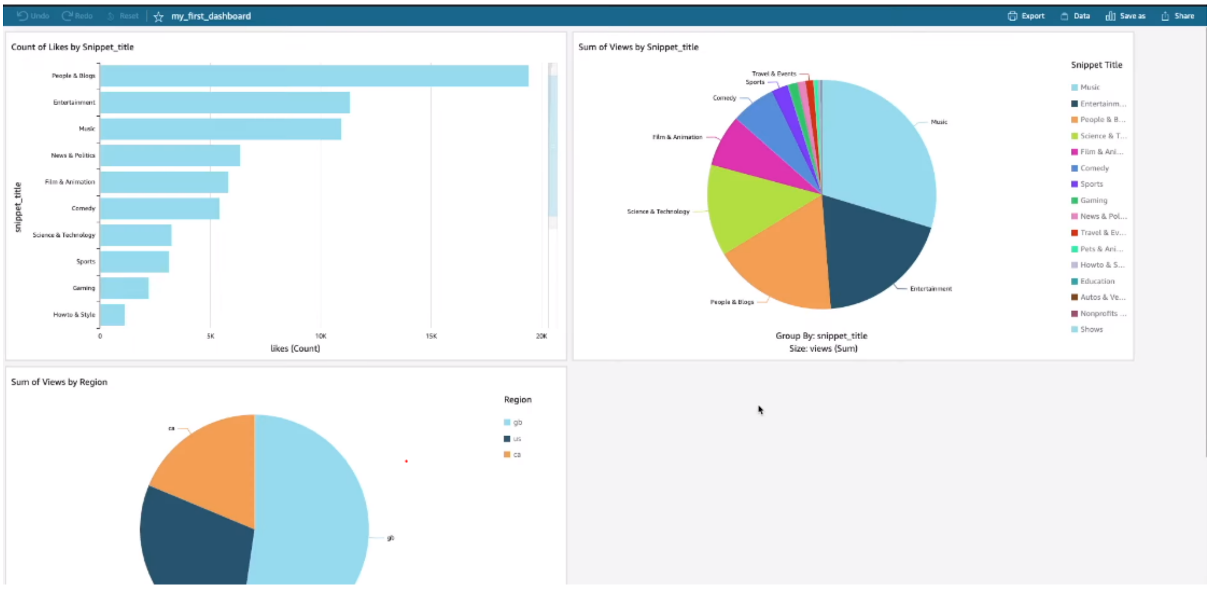
Job name	Type	Start time	End time	Run status	Run time	Capacity	W
fayssal-de-on-youtube-analytics-version	Glue ETL	06/25/2023 18:52:03	06/25/2023 18:53:51	✓ Succeeded	2 minutes	10	G

The final Result :



1	snippet.channelid	string	-	-
2	ratings_disabled	boolean	-	-
3	comments_disabled	boolean	-	-
4	snippet.title	string	-	-
5	trending_date	string	-	-
6	etag	string	-	-
7	video_id	string	-	-
8	thumbnail_link	string	-	-
9	kind	string	-	-
10	comment_count	bigint	-	-
11	likes	bigint	-	-
12	description	string	-	-
13	dislikes	bigint	-	-
14	views	bigint	-	-
15	snippet.assignable	boolean	-	-
16	channel_title	string	-	-
17	title	string	-	-
18	publish_time	string	-	-
19	tags	string	-	-
20	video_error_or_removed	boolean	-	-

7. Aws Quicksight


Undo
Redo
Reset
my_first_dashboard
Export
Data
Save as
Share
Count of Likes by Snippet_title
People & Blogs
Entertainment
Music
News & Politics
Film & Animation
Comedy
Science & Technology
Sports
Gaming
Howto & Style
snippet_title
0
5K
10K
15K
20K
likes (Count)
Sum of Views by Snippet_title
Travel & Events
Sports
Comedy
Film & Animation
Science & Technology
People & Blogs
Entertainment
Music
Group By: snippet_title
Size: views (Sum)
Snippet Title
Music
Entertainm...
People & B...
Science & T...
Film & Ani...
Comedy
Sports
Gaming
News & Pol...
Travel & Ev...
Pets & Ani...
Howto & S...
Education
Autos & Ve...
Nonprofits ...
Shows
Sum of Views by Region
ca
gb
Region
gb
us
ca