CREATING AN ETL PIPELINE IN PYTHON FOR YOUTUBE DATA ANALYTICS

Dr. Feng George Yu, Project Coach Youngstown State University, USA
GURRAPU SANJAY BHARGAVA, Youngstown State University, USA
KATA HANUMANTH PUJA, Youngstown State University, USA
SAI KISHORE SUROJU, Youngstown State University, USA

1 INTRODUCTION

One of the most popular websites, YouTube receives over 2.6 billion monthly visits from users worldwide. Imagine that you are in charge of the campaign advertisements for your company's next data-driven marketing initiative. Which media outlet do you believe will be most effective for the advertisement? Nobody else but YouTube! As a result, the business will need to analyze a significant amount of YouTube data using a range of tools and indicators in order to fully comprehend how to market its new campaign on the site. The business might be interested in learning how to classify videos based on statistics and comments.

2 BACKGROUND

YouTube is one of the largest video-sharing platforms globally, hosting billions of videos and attracting billions of users. Analyzing YouTube data can provide valuable insights into user behavior, content popularity, and emerging trends. The project leverages AWS services to build a scalable and robust ETL pipeline for processing YouTube data.

The primary goal of this project is to build an Extract, Transform, and Load (ETL) data pipeline utilizing Python, AWS services (Athena, Glue, Lambda, and S3), and AWS IAM for secure resource access. The administration, simplification, and analysis of structured and semi-structured YouTube video data will be made possible through this pipeline, with a particular emphasis on video categories and trending Metrics.

3 PROBLEM STATEMENT

Analyzing large-scale YouTube data manually is impractical due to its sheer volume. Therefore, a systematic approach is needed to automate the extraction, transformation, and loading of data from YouTube into a format suitable for analysis. The challenge is to design an efficient ETL pipeline that handles data extraction from database, transforms the raw data into a structured format, and loads it into a data warehouse for analytics. So, after analyzing the data we can notice that what type of data most people are watching in different countries. So, people who wants to advertise can put that products in the youtube videos so that they can sell their products

4 METHODOLOGY

We followed different steps and used different AWS to complete this project. They are explained below.

4.1 Languages used

We used SQL, Python for this project.

4.2 Services used

AWS S3, AWS Glue,AWS Lambda, AWS Athena, AWS IAM

4.3 Data Pipeline

When it comes to moving raw data from one system to another, a data pipeline resembles a conveyor belt. The pipeline encompasses the entire process of gathering, storing, and processing data, as well as performing data analytics and converting it into a query-ready format. It controls the ETL process and displays key performance indicators (KPIs).

4.4 Amazon S3

Amazon S3 (AWS Simple Storage Service) is a highly scalable object storage service offered by AWS, which provides secure and durable storage for various data types, such as documents, images, videos, etc. With S3, you can easily store and retrieve data anywhere online. We have used S3 services as the data storage. We uploaded initial Json files into the S3 bucket using the AWS CLI.

4.5 AWS IAM

AWS Identity and Access Management (IAM), a popular service by Amazon Web Services (AWS), enables users to manage access and permissions to their AWS resources securely. We demonstrated how can a organization can create and manage user accounts, assign granular permissions, and control who can access your AWS services and resources. We set up policies to define fine-grained access controls, ensuring that only authorized users or services can interact with AWS environment. It's a crucial tool for maintaining security and ensuring compliance within AWS infrastructure.

4.6 AWS Glue

AWS Glue is a serverless data integration service. This service made us easy to collect, process, and combine data for data analytics . We used Crawlers to clean the data and combine the data, so after running the crawlers our raw data is converted into a clean data. We created ETL using this service. When we run the crawler, crawler will import the data from the S3 bucket and perform theETL using the AWS glue ETL and it will convert the unstructured json files to parquet file using lambda functions so that we can explore the data using AWS Athena.

4.7 AWS Athena

AWS Athena is a serverless query service offered by AWS. In this project we used athena to analyze data stored in Amazon S3 using standard SQL queries and quickly extract insights from our data by querying the data and also used it for data exploration.

4.8 AWS Lambda

We used aws lambda service to automate the data cleaning process when ever we upload the json files in our S3 bucket. We used python language to create a lambda functions.
We data wrangler and pandas for data manipulation. Data is to create a data frame in parquet

4.9 AWS QuickSight

We connected our S3 bucket data and visualized the data and created the dashboards.

5 PROJECT PROCEDURE

5.1. Creating An AWS S3 Bucket For The ETL Pipeline.

The next step in this Python ETL project is to create an AWS S3 bucket following the best practices mentioned in the official AWS documentation and copy the data into the S3 bucket using the AWS CLI (Command Line Interface). Once you have successfully created the bucket, it's time to copy all the data (CSV files containing Youtube video statistics for different regions) into the bucket. Then you will create an AWS Glue Catalog that will act as a central data repository.

5.2. Creating AWS Glue Catalog For The AWS ETL Pipeline.

In this step of the project, you will learn the process of utilizing the AWS Glue service for data management and ETL (Extract, Transform, Load) tasks. The steps involve creating an AWS Glue Catalog, importing data from an S3 bucket using a crawler, and performing ETL using the AWS Glue ETL tool and SQL queries with Athena. Creating an AWS Glue crawler, specifying details like input source and output database further enhances the understanding of data extraction and organization.

You will also use AWS Glue to automatically discover and catalog data stored in your S3 bucket, enabling easier data querying and analysis. You will transform semi-structured JSON data into a more organized format suitable for SQL queries using Athena's SerDe libraries. The following step will give you hands-on experience implementing an ETL process to cleanse and convert the JSON data, preparing it for further analysis and exploration using SQL queries in Athena.

5.3. AWS Lambda Data Pipeline

This ETL (Extract, Transform, and Load) process mainly involves using an AWS Lambda function to extract the JSON data from the raw S3 bucket, convert it into Parquet format, push that data using AWS Data Wrangler into the clean S3 bucket, and automatically catalog it into the Glue Catalog, where you can query it using Athena SQL statements.
You will start by importing essential libraries like data wrangler and Python libraries like Pandas for efficient data manipulation in your ETL process. This step will allow you to understand how to write a Lambda function with event and context parameters. You must also create a dataframe in memory and leverage Pandas to read data from the JSON file into the dataframe, enabling data manipulation and transformation.
You will use a data wrangler to write the dataframe to Parquet in S3 while simultaneously interacting with the Glue Catalog for efficient data cataloging and management. This step also involves creating a new S3 bucket for storing clean and optimized data obtained from the Lambda function, ensuring efficient data organization.

Next, it's time to create a new Lambda function that interacts with the S3 bucket and Data Catalog using Data Wrangler, allowing data extraction and manipulation. You must configure a testevent (s3-json-youtube-put) to simulate and verify the Lambda function's behavior when triggeredby an S3 event. You might also enhance the functionality of your Lambda function by adding Lambda layers. You will further enhance your Lambda function by incorporating AWSDataWrangler, and creat-ing a database in Glue Catalog will allow you to organize and store the cleansed data extracted by the Lambda function, making it easier to query and analyze. You will also gain insights into visual-izing the data in a columnar format (Parquet) using Athena, running SQL queries, and exploring the data further.

Setting up an S3 crawler will allow you to easily automate extracting rawstatistics data from theS3 bucket and populating a table in the Glue Catalog. This step also demonstrates how to transformCSV files within the Glue Catalog table into Parquet format data using Glue Studio, optimizing thedata for efficiency and querying.

5.4. Python ETL Pipeline Project- Data Processing Using AWS Glue.

This step will allow you to leverage AWS Glue's capabilities, particularly the Glue Spark ETL jobs feature, to process and transform data efficiently. You will be able to gain insights into creating a Spark job within AWS Glue, understanding how to transform JSON data to Parquet format, and configuring job properties to suit specific requirements.

Running the Glue Spark job will allow you to process and transform the data imported and stored in the cleansed S3 bucket. This step helps you understand how to leverage the power of the Spark data processing engine. The new S3 crawler will scan the specified path and partitions inthe S3 bucket, extracting data and creating a new table in the Glue Catalog, which will help you automate the process of discovering, cataloging, and extracting data.

Once you have successfully appended the data from the CSV file and JSON files, it's time to addan S3 trigger to the Lambda function generated earlier. You must add an S3 trigger to your Lambda function to automate any further processing of new incoming JSON files, such as video category data, etc., and push that data further into the cleansed layer. You can check the performance of the Lambda function using CloudWatch metrics. Now, it's time to create the final layer in S3, i.e., the Analytics/Reporting Layer, that will further help to visualize the data using QuickSight.

5.5. Data Materialization Using AWS Glue Studio.

The analytics layer aims to provide a materialized view of the cleansed data along with the results of the SQL join query performed earlier to the business users, saving their time and optimizing the cost and performance of the overall business solution.

This step involves creating a new S3 bucket and configuring IAM policies for the necessary privileges. Additionally, you will have to create a new database (db_youtube_analytics) and a Glue Studio job (bigdata-on-youtube-spark-materialize-categories). You will also perform a join operation on Parquet format datasets and store the output in an S3 bucket, which can be viewed using Athena

SQL statements. Finally, you will leverage this view to visualize the data using AWS QuickSight, gaining insights and creating reports.

5.6. Data Visualization Using AWS QuickSight.

Before moving on to the data visualization part, you must create and trigger a new ETL workflow in Glue Studio to automate the new Glue job. This workflow will serve two purposes- transform raw .json data to Parquet format and transform the SQL Join outputs to materialized views in the Analytics database.

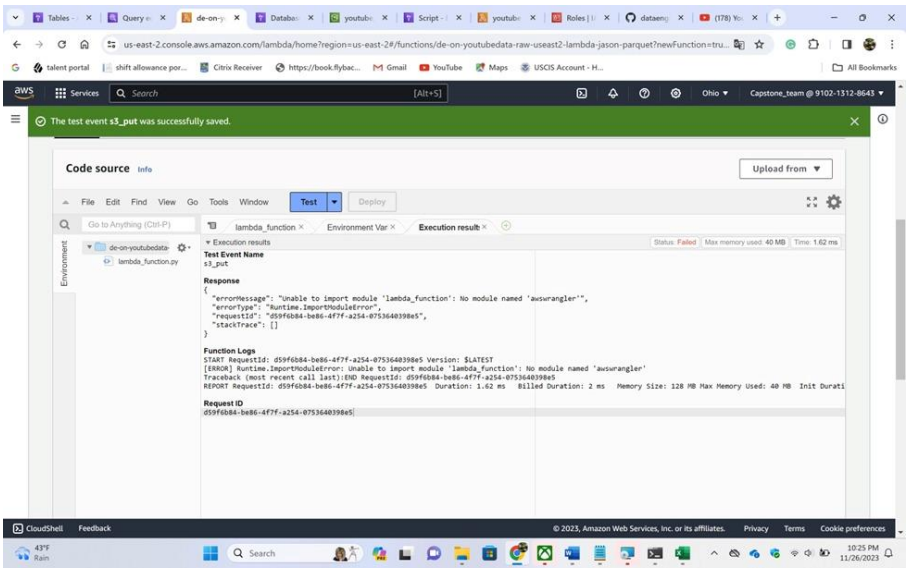
The final step in this data engineering project is to create a new dataset in QuickSight using Athena's new analytics database as the data source. Once you have successfully done that, it's time to start visualizing the data by creating dashboards in QuickSight. You can customize the dashboards and generate as many as you need.

6 EXPERIMENTS

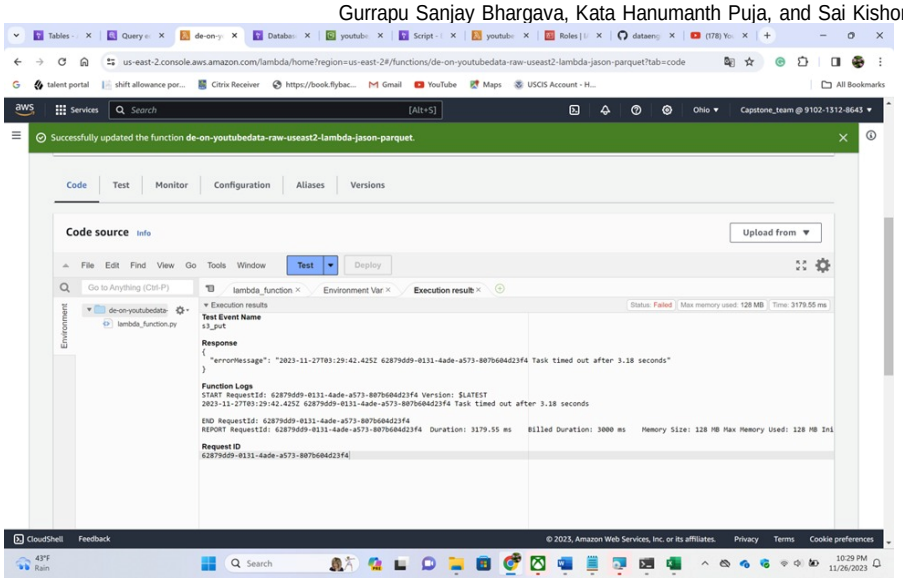
I am discussing the errors we got while working with the Lambda Functions and Testing of Lambda Functions.

6.1 ERRORS

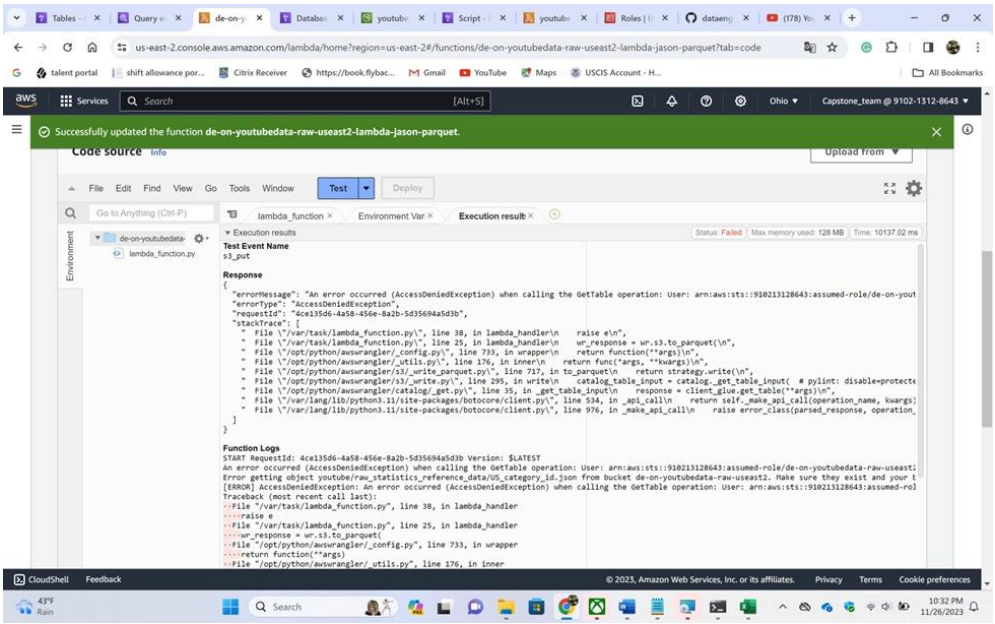
6.1.1 Error 1: While running the test instance in lambda function we faced the error "no module named awswrangler" in order to resolve this error we have imported aswssdkpandas-python 3.11



6.1.2 Error 2: Timeout error- To over come this issue we have set the time to 3 mins 3 secs.



6.1.3 Error 3: We got this error" accessdeniedexception" in order to resolver this error we have to add glue permissions in IAM role. So we have added AWSglueserviceRole and fixed this error.



6.2 Testing

After creating the Lambda function and Aws Glue Catalog, we tested only one dataset to see whether the data is converting from .json to .parquet , it successfully converted the data to .parquet, we have deleted the data sets from our S3 bucket and uploaded all the files to see lambda functions should automatically trigger and convert all the .json files to .parquet and show the data in the Athena. Our lambda function successfully produced the clean data.

7 CONCLUSION

We have used AWS services such as IAM, S3 buckets, lambda Functions using python, Athena, Glue and analyzed the three countries youtube data and visualized the data. From below picture we can say that most people watched Music category, Entertainment and science and technology. So, advertisers can advertise their products in those videos. The third most viewed content is Science and technology, we can expect 80 Percent of them might be the students, so we can advertise the education loans, credit cards, items which has student discounts, uniforms, dressing, cosmetics and many more.

7.1 CONTRIBUTION OF TEAM MEMBERS

We have researched various websites and gained good knowledge on the AWS services before starting this project which helped ourselves to complete this project, apart from group contribution. Data extraction, transformation and AWS service integration by Sanjay Gurrapu.

Sanjay is in charge of creating and carrying out the YouTube data extraction procedure. he focused on setting up the Identity and access management and creation of S3 buckets and also exported the data to s3 bucket using AWS Command line interface and worked on python scripts to create lambda functions. He used Python to play a significant part in the transition phase and Implemented data cleansing, formatting, and structuring for optimal analysis. Actively participated in team meetings, provided valuable insights and updates on progress.

ETL pipeline creation, Error Handling, Optimization, Data visualization, Documentation and Best Practices by Puja Kata.

Puja Implemented robust error handling mechanisms to enhance the pipeline's resilience and contributed to optimization strategies, improving the overall performance of the ETL process. Adding triggers to the lambda function and also transforming the data to cleansed s3 bucket and she visualized the data and created a final dashboard in aws quicksight. She created comprehensive documentation for the ETL pipeline, ensuring clarity for both team members and future users. Advocated for and implemented AWS best practices, promoting efficiency and cost-effectiveness.

Collaboration, Communication, Adaptation and Presentation Preparation by Sai Kishore Suroju.
Sai kishore worked on the adding environmental variables in the lambda functions and worked on jobs created by the lambda functions, used queries in the AWS Athena and made sure that the lambda functions were executing as per our requirements. Actively participated in team meetings, providing valuable insights and updates on progress. he Assisted in the creation of the presentation materials, including slides and visuals for the audience.

8 REFERENCES

1. www.Youtube.com
2. www.Google.com
3. www.kaggle.com

9 APPENDIX

https://github.com/Electron-Sanjay/CSCI-DATA-CAPSTONE-PROJECT
https://www.kaggle.com/datasets/datasnaek/youtube-new