Youtube Data Analysis I End to End DE Project

- **Dataset**: Top Trending YouTube Video Statistics Source: Kaggle
- **Key Features**:
  - Comprehensive daily tracking of YouTube's most popular videos
  - Covers multiple months of trending video data
  - Encompasses 10 regions: USA, UK, Germany, Canada, France, Russia, Mexico, South Korea, Japan, and India
  - Up to 200 trending videos logged per day for each region
- **Data Points Included**:
  - Video title
  - Channel name
  - Publication timestamp
  - Associated tags
  - View count
  - Like and dislike tallies
  - Video description
  - Number of comments
- This dataset offers a rich, multi-regional snapshot of YouTube's trending content, providing insights into video performance and user engagement across various markets.

OVERVIEW:

- This initiative focuses on the efficient handling and in-depth analysis of YouTube's trending video data. The project encompasses:
  - **Secure data management**: Implementing robust security measures for data storage and access.
  - **Streamlined processing**: Developing efficient workflows to handle both structured and semi-structured data from YouTube videos.
  - **Comprehensive analysis**: Conducting thorough examinations of video performance based on:
    - Video categories
    - Trending metrics
  - **Data optimization**: Organizing and structuring the data to facilitate quick and insightful analytics.
- The ultimate goal is to extract meaningful insights from YouTube's trending video landscape, enhancing our understanding of content performance and user engagement patterns across different categories and trending parameters.

PROJECT OBJECTIVES:

- **Data Acquisition System**:
  - Develop a robust mechanism to collect and import data from various sources efficiently.
- **Data Transformation Pipeline**:
  - Create an ETL (Extract, Transform, Load) system to convert raw data into a structured, analysis-ready format.
- **Centralized Data Repository**:

    - Implement a data lake to store diverse datasets from multiple sources in a unified location.
- **Scalable Architecture**:
    - Design a flexible system capable of handling increasing data volumes without performance degradation.
- **Cloud-Based Processing**:
    - Leverage AWS cloud services to process large-scale data that exceeds local computing capabilities.
- **Analytics Dashboard**:
    - Construct an interactive dashboard to visualize key insights and answer critical business questions.
- These objectives aim to create a comprehensive, cloud-based data analytics platform that can efficiently process, store, and analyze large volumes of YouTube trending video data, providing valuable insights through user-friendly visualizations.

AWS SERVICES USED:

- S3
    - Amazon S3 is an object storage service that offers robust scalability, data availability, security, and performance.
- IAM
    - AWS Identity Access and Management allows for secure management of access to AWS services and resources.
- QuickSight
    - Amazon QuickSight is a cloud-based business intelligence (BI) service that is scalable, serverless, embeddable, and powered by machine learning.
- AWS Glue
    - AWS Glue is a fully managed ETL (Extract, Transform, Load) service that simplifies data preparation and integration for analytics, ML, and app development.
    - It automates data discovery, transformation, and loading tasks without requiring server management.
- AWS Lambda
    - Amazon Lambda is a serverless compute service that automatically runs and scales code in response to events, eliminating the need for server management.
    - It enables developers to focus on writing code while AWS handles all the underlying infrastructure provisioning and maintenance.
- AWS Athena
    - Amazon Athena is a serverless query service that allows direct SQL-based analysis of data stored in Amazon S3 without requiring data movement or loading.
    - It offers on-demand, pay-per-query functionality, enabling users to instantly query vast amounts of data directly from S3 using standard SQL, with no infrastructure to manage.

ARCHITECTURE:

WORKFLOW:

This architecture diagram represents a data processing and analytics platform built on AWS.

Source Systems

- These are the origin points of data that need to be ingested into the platform.

Data Ingestion

- Data from the source systems is ingested in bulk using the S3 API.
- This data is stored in an S3 bucket.

Data Platform

- **Data Lake**: This is a central repository where all the data is stored. It is divided into three main areas:
  - **Landing Area**: Raw data is initially stored here in an S3 bucket.
  - **Cleansed / Enriched**: Data is processed and cleaned, and any necessary transformations are applied. The processed data is then stored in another S3 bucket.
  - **Analytics / Reporting**: The final processed data, ready for analytics and reporting, is stored in another S3 bucket.

Data Processing

- **AWS Glue**: This service is used for data processing. It can extract, transform, and load (ETL) data from the landing area to the cleansed/enriched area.
- **AWS Lambda**: This serverless compute service can be used to run code in response to events, such as data processing tasks.

Data Cataloguing & Classification

- **AWS Glue Data Catalog**: This service maintains a catalog of metadata about the data stored in the data lake, making it easier to discover and use.

Data Flow

- **AWS Step Functions**: This service coordinates the workflow of various AWS services involved in the data processing pipeline.

Data Access & Analytics

- **API**: An API is used to provide access to the analytical data.
- **AWS Athena**: This service allows users to query the data stored in S3 using SQL.
- **Redshift (Optional)**: This data warehouse service can be used for complex queries and large-scale data analysis.

Target Systems

- **Monitoring / Alert**: AWS CloudWatch is used for monitoring and alerting.
- **Analytics Tools**: These tools are used for data analysis and visualization:
  - **QuickSight**: Amazon's BI service for creating dashboards and visualizations.
  - **Qlik**: A BI tool for data visualization and discovery.

Identity & Access Management

- This service is used to securely manage access to AWS services and resources, ensuring that only authorized users and applications can access the data and services.

SUMMARY

1. Data is ingested from source systems in bulk and stored in the landing area of the data lake.
2. Data is processed and cleaned using AWS Glue and AWS Lambda, then stored in the cleansed/enriched area.
3. The data catalog is maintained using AWS Glue Data Catalog.
4. AWS Step Functions orchestrate the workflow of data processing.
5. Processed data is accessed via an API for analytical purposes.
6. AWS Athena and optional Redshift are used for querying the data.
7. Data is analyzed and visualized using various analytics tools like QuickSight, Qlik, etc.
8. IAM ensures secure access to all AWS services and resources.

This architecture supports scalable, secure, and efficient data processing and analytics on the AWS cloud.

Appendix:

- Created IAM User -
- Created S3 bucket with 2 folders: Youtube and raw stats, to upload json data into it. A category_id field, which differs by area, is also included in the JSON file linked to the region.
- The data for each region is in its own file.
- Created Glue Crawler on the raw json s3 bucket folder to create metadata.Stored that information into the Glue Database.
- Now to view the data from the Glue Catalog, you need to create an S3 bucket to be used as output for querying and storing the results of the output using Athena.
- When you try to Query JSON data using Athena, it fails and you get this error.
- Formatted JSON file doesn't work with AWS Glue Crawler.
- For this we will convert the JSON data into Column data using Apache Parquet format.
- Created AWS Lambda with IAM role to access S3 bucket.
- Tested the Lambda function
- This helped create a cleaned version of the file into the s3 bucket we created to store the cleaned file.
- Now create and run crawler for all csv raw data. Because of the files that are placed in sub folders region wise, region will be our partition key.
- Next step was to create a Glue Job to process the raw csv data and move it to the cleaned s3 bucket we created for storing the cleansed data.
- Running this ETL Job successfully will help create a PySpark code in the backend.
- Add data frames to this code and partition key.
- Once the job runs successfully you will have cleaned data from different regions of the csv file into your s3 bucket.

- Creating a Lambda trigger to S3 automate new JSON being added and cleaned for a region.
- Build ETL Pipeline and create a Reporting Layer by preprocessing data. Using Glue
- Visualize the data using AWS Quicksight
-

Project Credit: @Darshil Parmar