YouTube ETL Pipeline:

Utilizing Python, and Apache Airflow on EC2 Instance

—

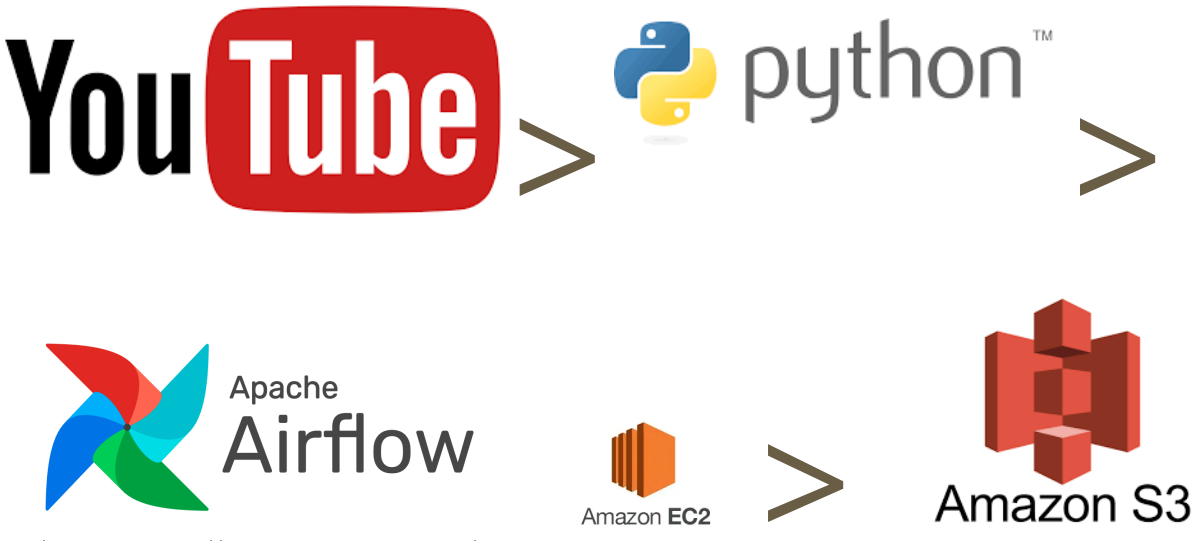
Chirag Madhukar
chirag.madh09@gmail.com
(732) 822-1514

Project Overview

Introduction:

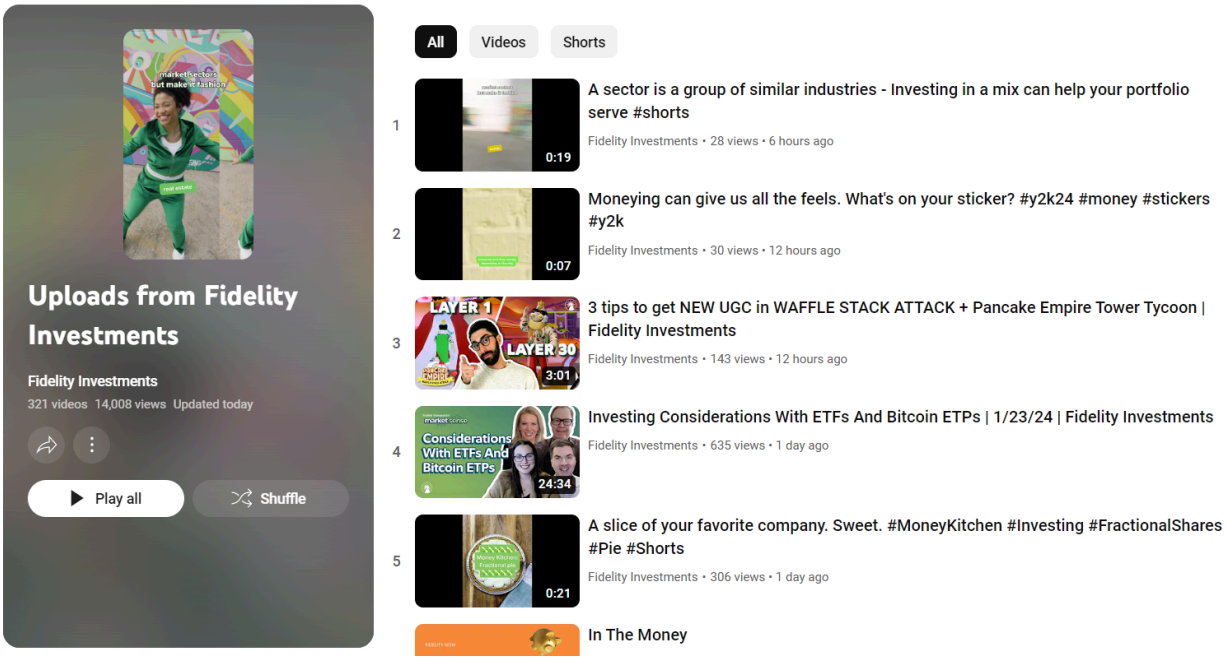
The YouTube ETL pipeline project aims to extract, transform, and load data from the Fidelity YouTube channel using the YouTube API. The primary objective is to gather information about the videos, including titles, likes, comments, upload dates, and other relevant metrics. The processed data is then stored in an S3 bucket using Apache Airflow on AWS EC2 instance for workflow management.

Data Pipeline is shown below



Data Collection

We are trying to pull this below 321 videos data into a dataframe which contains Video Title, Published Date, Views, and Likes count.



To access the YouTube API and collect data, an API credential key is required.

1. Step 1: Create a Google Cloud Platform (GCP) Project Go to Google Cloud Console. Create a new project (e.g., YouTubeETLProject).
2. Step 2: Enable YouTube Data API v3 In GCP Console, navigate to "APIs & Services" > "Dashboard." Enable "YouTube Data API v3."
3. Step 3: Create API Credentials In GCP Console, go to "APIs & Services" > "Credentials." Create an API Key.
4. Step 4: Use API Key in YouTube ETL Script Copy the generated API key and paste it into the api_key variable in YouTube_ETL.py.

Data Processing

YouTube_ETL.py File: Attached in the Email

The YouTube_ETL.py script extracts data from the Fidelity YouTube channel using the obtained API key. It performs the following steps:

1. Fetches channel statistics (subscribers, views, total videos).
2. Retrieves the playlist ID for Fidelity channel.
3. Extracts video IDs from the playlist.
4. Gathers video details (title, published date, views, likes) using the video IDs.
5. Creates a Pandas DataFrame with the collected data. Saves the processed data frame into an S3 bucket.

The DataFrame is displayed, and the subsequent step involves constructing a data pipeline within Apache Airflow to channel the data into an S3 bucket.. This data can be accessed on various cloud big data platforms for analysis.

	Title	Published_date	Views	Likes
0	A sector is a group of similar industries - In...	2024-01-25T21:30:03Z	28	1
1	Moneying can give us all the feels. What's on ...	2024-01-25T16:03:13Z	30	1
2	3 tips to get NEW UGC in WAFFLE STACK ATTACK +...	2024-01-25T15:35:31Z	143	5
3	Investing Considerations With ETFs And Bitcoin...	2024-01-24T22:07:03Z	635	41
4	A slice of your favorite company. Sweet. #Mone...	2024-01-24T16:37:58Z	306	3
..	...	...	...	...
316	How to Pay Off Debt | Fidelity Investments	2015-05-13T15:36:43Z	38430	173
317	3 Things You Need to Know about an Emergency F...	2015-04-22T15:29:42Z	350587	1135
318	401(k) Contribution Challenge - Investing Basi...	2014-09-19T18:27:20Z	70716	261
319	What is a Sector? - Investing Basics | Fidelit...	2014-09-19T18:17:09Z	25885	116
320	What are Municipal Bonds? | Fidelity Investments	2013-09-17T15:54:51Z	167926	1470

[321 rows x 4 columns]

For instance, we can seamlessly transfer this dataset from the S3 bucket into Databricks, where it can be transformed into a PySpark DataFrame using the Parquet format. Subsequently, this processed data can be loaded into Snowflake for in-depth analysis. Furthermore, the data can be directly connected from Snowflake to Tableau, facilitating data visualization, insights, and comprehensive analysis.

Worflow Management

Apache Airflow Setup:

Setting up Apache Airflow on an EC2 instance involves several steps. Here's a simplified walkthrough:

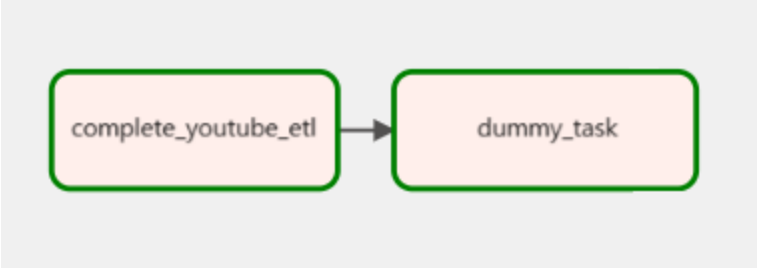
1. Update and install necessary packages.
    a. sudo apt update
    b. sudo apt install python3-pip
    c. sudo apt install sqlite3
    d. sudo apt install python3.10-venv
    e. sudo apt-get install libpq-dev
2. Create and activate a virtual environment.
    a. python3 -m venv venv
    b. source venv/bin/activate
3. Install Apache Airflow.
    a. pip install "apache-airflow[postgres]==2.5.0" --constraint "https://raw.githubusercontent.com/apache/airflow/constraints-2.5.0/constraints-3.7.txt"
4. Initialize Airflow database.
    a. airflow db init
5. Install and configure PostgreSQL.
    a. sudo apt-get install postgresql postgresql-contrib
    b. sudo -i -u postgres
    c. Psql
    d. CREATE DATABASE airflow;
    e. CREATE USER airflow WITH PASSWORD 'airflow';
    f. GRANT ALL PRIVILEGES ON DATABASE airflow TO airflow;
    g. Exit PostGreSQL shell by pressing Ctrl+D two times.
6. Configure Airflow for PostgreSQL.
    a. sed -i 's#sqlite:////home/ubuntu/airflow/airflow.db#postgresql+psycopg2://airflow:airflow@localhost/airflow#g' airflow.cfg
    b. sed -i 's#SequentialExecutor#LocalExecutor#g' airflow.cfg
7. Initialize Airflow database again.

   a. airflow db init
8. Create an admin user.
   a. airflow users create -u airflow -f airflow -l airflow -r Admin -e airflow@gmail.com
9. Start Airflow webserver and scheduler.
   a. airflow webserver &
   b. airflow scheduler

Now, you should be able to access the Airflow web interface by navigating to your EC2 instance's public IP or DNS on port 8080 (e.g., http://your-ec2-public-ip:8080). Adjust security groups and firewall rules to allow traffic on port 8080 for the Airflow web server.

YouTube_DAG.py File: Attached in the Email

The YouTube_DAG.py file schedules and orchestrates the ETL process. It includes tasks for running the YouTube_ETL.py script and a dummy task.



Note: Ensure that the DAG folder structure in Visual Studio Code (VSCode) is consistent with what is specified in the .cfg file and matches the actual project structure. This alignment ensures seamless execution of Airflow DAGs.

Data Storage

The processed data frame from the ETL process is saved into an S3 bucket. The filename can be later accessed through big data and cloud platform tools like Databricks in the following way:

- ***df = spark.read.csv("s3://bucket_name/path_to_data.csv")***

Next Steps and Recommendation

Data Warehousing and Advanced Analytics:

Leveraging Cloud Big Data Platforms:

The processed data, currently stored in the S3 bucket, can be seamlessly integrated into various cloud big data platforms for comprehensive analysis and advanced analytics. One exemplary pathway involves utilizing Databricks and Snowflake for enhanced capabilities.

Loading Data into Databricks:

Databricks, a unified analytics platform, allows for efficient data processing and analytics. The data from the S3 bucket can be loaded into Databricks, transforming it into a PySpark DataFrame using the CSV or Parquet format for optimized performance and scalability.

```
# Sample PySpark code to load data from S3 into PySpark DataFrame
df = spark.read.csv("s3://bucket_name/path_to_data.csv")
df = spark.read.csv("s3://bucket_name/path_to_data.parquet")
```

Data Analysis in Snowflake:

Snowflake, a cloud-based data warehousing platform, provides a robust environment for analytical queries and processing. The PySpark DataFrame from Databricks can be seamlessly loaded into Snowflake for further analysis.

Integration with Tableau:

Once the data is stored in Snowflake, it can be directly fed into Tableau for powerful data visualization, insights, and analysis. Tableau's integration with Snowflake ensures real-time access to the latest data, providing stakeholders with a dynamic and interactive reporting environment.

In summary, transitioning the processed data through Databricks and Snowflake opens avenues for advanced analytics, scalable processing, and seamless integration with Tableau. This approach ensures that stakeholders can derive meaningful insights from the enriched dataset, fostering data-driven decision-making.

Data Pipeline for Next Steps:

Conclusion

In conclusion, the YouTube ETL pipeline successfully extracts and processes data from the Fidelity YouTube channel. The combination of YouTube API, Pandas, and Apache Airflow provides a robust foundation for scalable and automated data processing.

This can achieved in Informatica in the following way as well. To upload data from a Pandas DataFrame to Amazon S3 using Informatica, start by configuring an S3 connection and designing the target table structure. Utilize Informatica's PowerExchange for Amazon S3 for enhanced connectivity. Create a mapping within the Informatica workflow, setting the Pandas DataFrame as the source and the S3 bucket table as the target. Configure necessary data transformations, then execute the workflow. Informatica's scalability and monitoring features ensure a smooth data transfer, effectively moving the Pandas DataFrame to the designated location on Amazon S3. Adjustments can be made based on project requirements and the Informatica version in use.

Technical Details

Connecting VSCode to EC2 Instance:

To connect Visual Studio Code (VSCode) to an EC2 instance:

1. Install the "Remote - SSH" extension in VSCode.
2. Open the command palette (Ctrl + Shift + P) and select "Remote-SSH: Connect to Host."
3. Enter the SSH key, IP address and EC2 instance details.
4. Connect to the EC2 instance.

Tech Stack:

- YouTube Data Extraction: YouTube API, Python
- Data Processing: Pandas, Google API Client
- Workflow Management: Apache Airflow
- Data Storage: Amazon S3
- IDE/Editor: Visual Studio Code

- Cloud Platforms: Google Cloud Platform, AWS EC2

For Next Steps

- Data Warehousing: Snowflake, Databricks
- Data Visualization: Tableau.
- Integration and Automation: Informatica PowerCenter

This project leverages a diverse tech stack to ensure efficient data extraction, processing, and visualization.

Thank you!