TWITTER ETL PIPELINE

This is not a tutorial but a brief description of the project.

Some details may not be mentioned clearly for the above-mentioned reason.

I would've wished to keep this pipeline running 24x7 but the incurring costs will be unaffordable for me hence, I will be turning off the instance to avoid computing costs.

This project was created with the help of an online tutorial by Darshil Parmar. He is a wonderful engineer and a marvellous teacher. Do check out his channel at https://www.youtube.com/@DarshilParmar

This is a basic ETL implementation using the following technologies.

1: Python

2: Pandas

3: AWS(S3 and EC2)

4: Apache Airflow

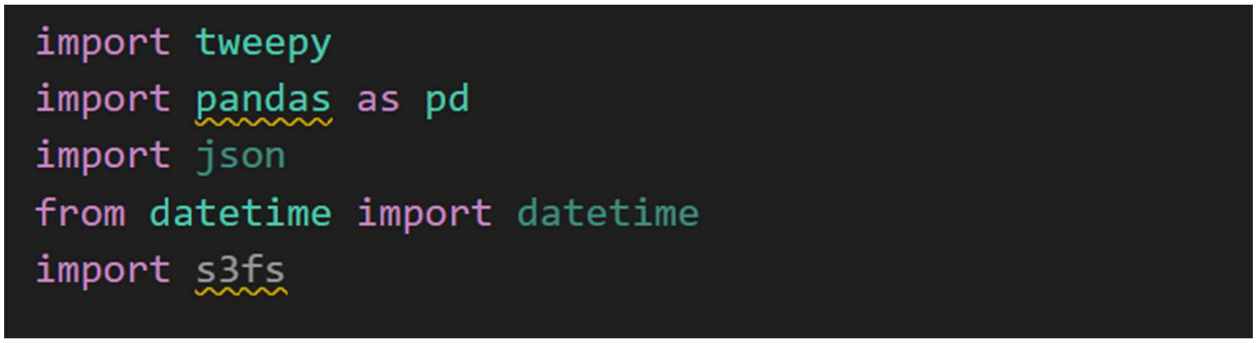
5: API(Twitter API)

6: Linux Terminal commands

Step 1: Extracting the data from Twitter API

I have attached all files along with this document but it may not run always as the AWS account credentials may differ.

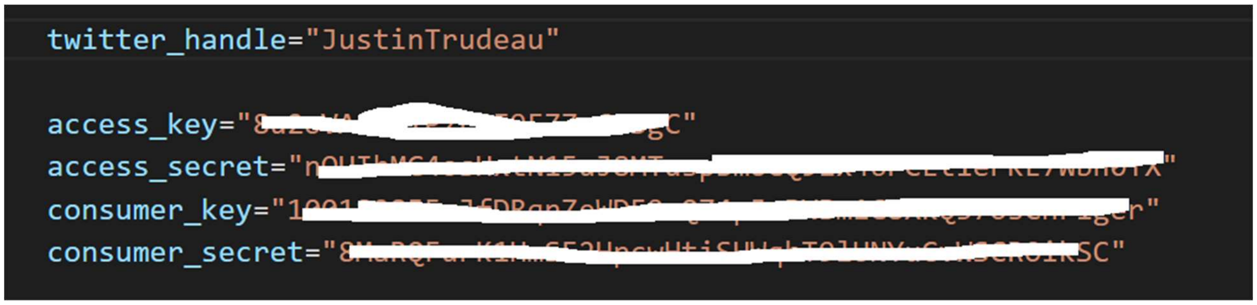
We will need the following packages:



```python
import tweepy
import pandas as pd
import json
from datetime import datetime
import s3fs
```

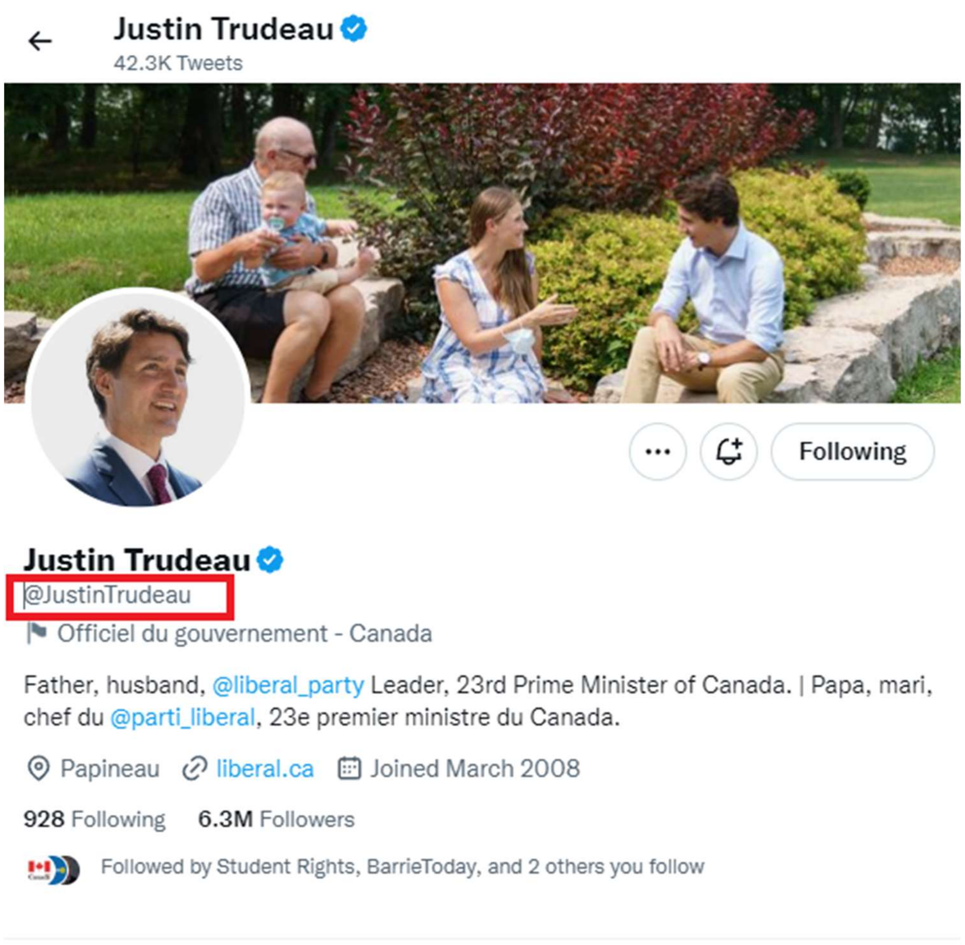
We first start with creating a developer account for getting the credentials for the Twitter API and storing them in the respective variables. You can learn more about Twitter API in their documentation.

https://developer.twitter.com/en/docs/twitter-api



```python
twitter_handle="JustinTrudeau"

access_key="[redacted]"
access_secret="[redacted]"
consumer_key="[redacted]"
consumer_secret="[redacted]"
```

In the twitter_handle variable enter the twitter handle as a string. In this case we will be scraping President Justin Trudeau's latest 200 tweets with the help of the twitter API.



Justin Trudeau
42.3K Tweets

Following

Justin Trudeau
@JustinTrudeau
Officiel du gouvernement - Canada

Father, husband, @liberal_party Leader, 23rd Prime Minister of Canada. | Papa, mari, chef du @parti_liberal, 23e premier ministre du Canada.

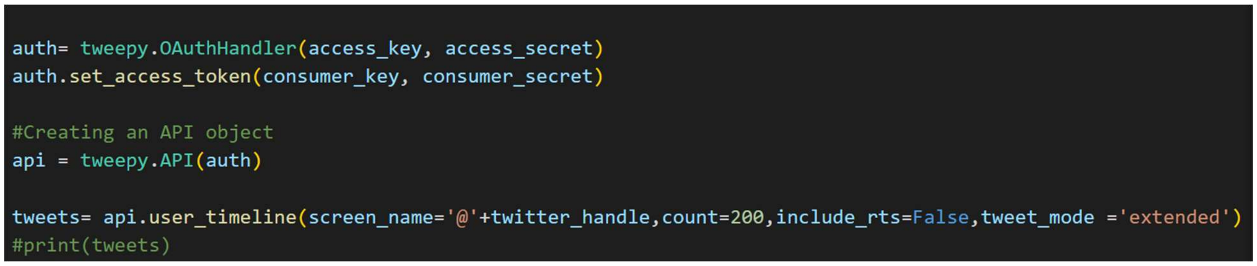
Papineau liberal.ca Joined March 2008

928 Following 6.3M Followers

Followed by Student Rights, BarrieToday, and 2 others you follow

The Twitter handles can be found on the profile page as mentioned above.
Please do not enter the '@' character before the handle name.

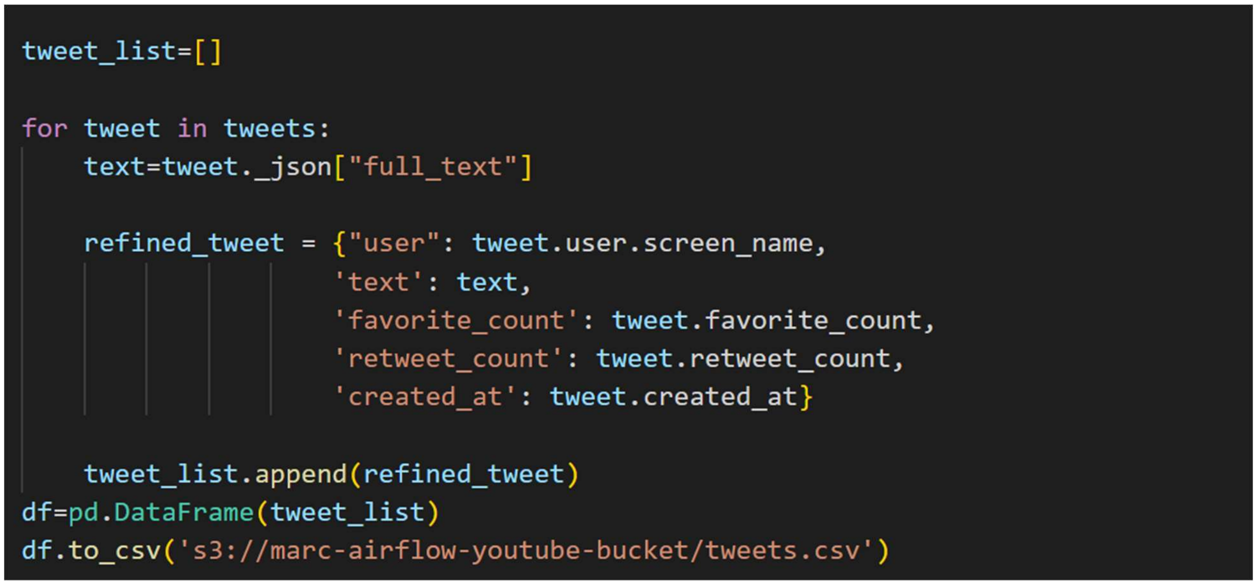
After the authentication of the API requests is done, we will be requesting data.



```
auth= tweepy.OAuthHandler(access_key, access_secret)
auth.set_access_token(consumer_key, consumer_secret)

#Creating an API object
api = tweepy.API(auth)

tweets= api.user_timeline(screen_name='@'+twitter_handle,count=200,include_rts=False,tweet_mode ='extended')
#print(tweets)
```

Once the API is setup, we will be extracting the data as a JSON file and converting it into a Python list using the json library.

We will then convert this list into a Pandas Data Frame for better organization and then convert it into a '.csv' file and store it in an Amazon S3 bucket.



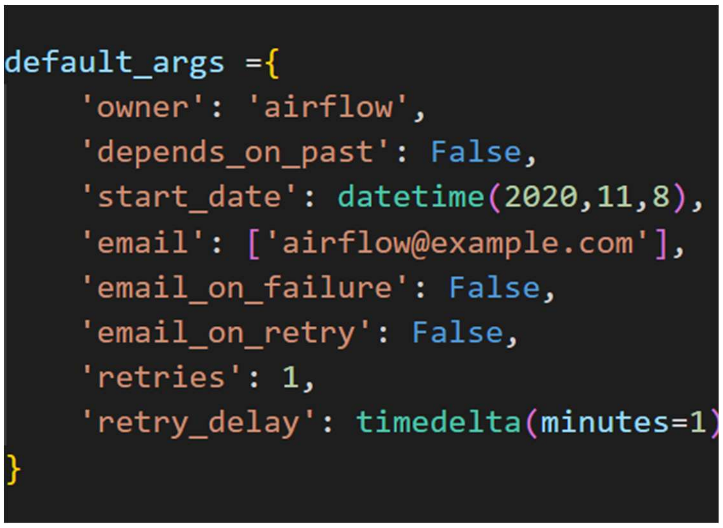
```python
tweet_list=[]

for tweet in tweets:
    text=tweet._json["full_text"]

    refined_tweet = {"user": tweet.user.screen_name,
                    'text': text,
                    'favorite_count': tweet.favorite_count,
                    'retweet_count': tweet.retweet_count,
                    'created_at': tweet.created_at}

    tweet_list.append(refined_tweet)
df=pd.DataFrame(tweet_list)
df.to_csv('s3://marc-airflow-youtube-bucket/tweets.csv')
```

STEP 2: Writing DAGs

DAGs are short for Directed Acyclic Graphs and we will require the following libraries for this process.

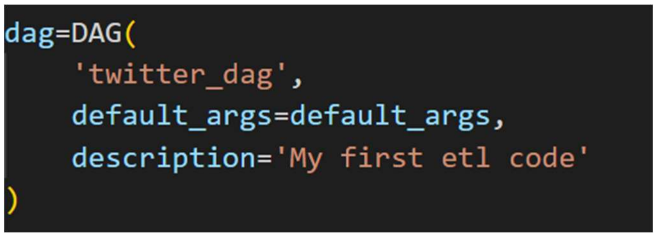
We will also convert our Step 1 code into a function called run_twitter_etl and import it here.

We will have to define the default_args as follows. For more information, visit the Apache Airflow Documentation. https://airflow.apache.org/docs/apache-airflow/stable/index.html



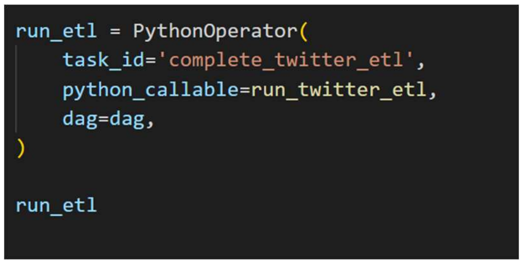
```python
default_args ={
    'owner': 'airflow',
    'depends_on_past': False,
    'start_date': datetime(2020,11,8),
    'email': ['airflow@example.com'],
    'email_on_failure': False,
    'email_on_retry': False,
    'retries': 1,
    'retry_delay': timedelta(minutes=1)
}
```

Here we are writing a DAG.



```python
dag=DAG(
    'twitter_dag',
    default_args=default_args,
    description='My first etl code'
)
```

We will use the following function called ‘PythonOperator’ from the Airflow library and provide the following parameters and call it.

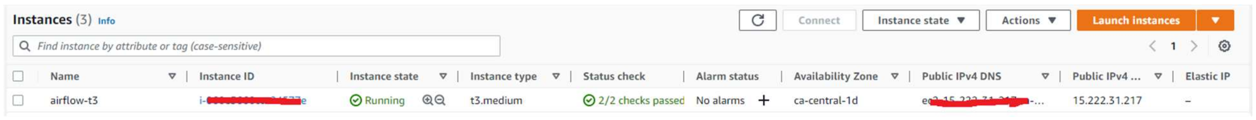


```python
run_etl = PythonOperator(
    task_id='complete_twitter_etl',
    python_callable=run_twitter_etl,
    dag=dag,
)

run_etl
```

Step 3: Setting up an AWS EC2 Instance

Create an account on AWS. I created a free-tier account but paid for the t3.medium instance as Apache Airflow has higher system requirements. The t2.micro free tier instance will not work. Trust me, I wasted approximately 6 hours trying to run it on a t2.micro but I just wasn't able to put it together.

Once the instance is created, we will install all the required packages similar to our python scripts. This instance works on Ubuntu.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
airflow-t3	i-[redacted]e	Running	t3.medium	2/2 checks passed	No alarms	ca-central-1d	ec[redacted]-...	15.222.31.217	–

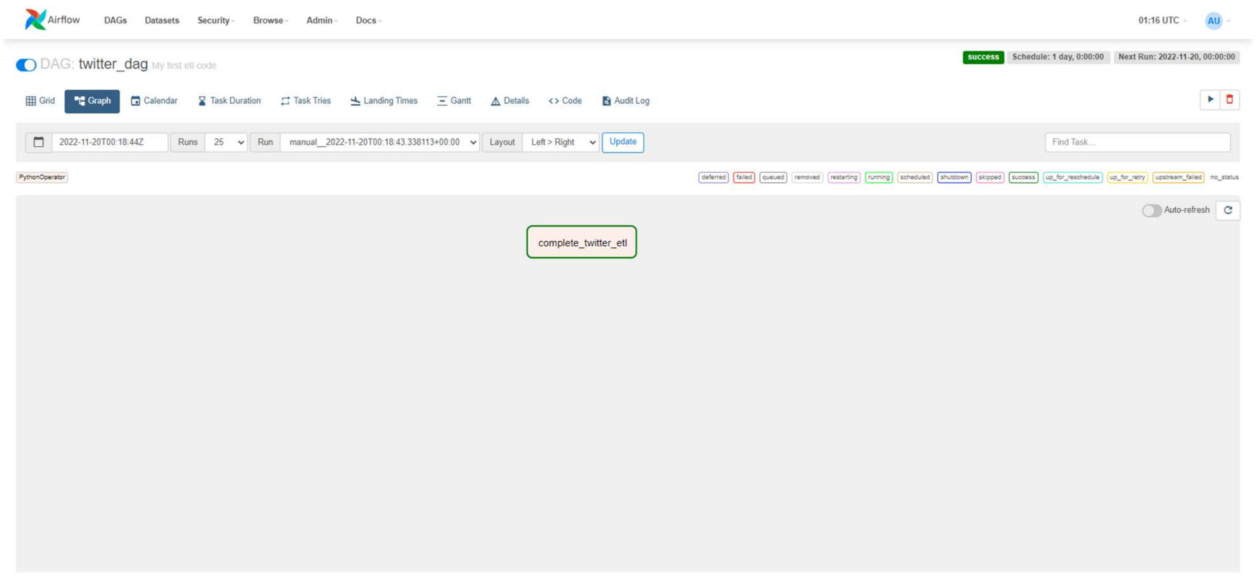
Step 4: Setting up an S3 bucket to collect the data from the pipeline.

This is a fairly simple process. Please refer to the documentation for additional help.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/creating-bucket.html

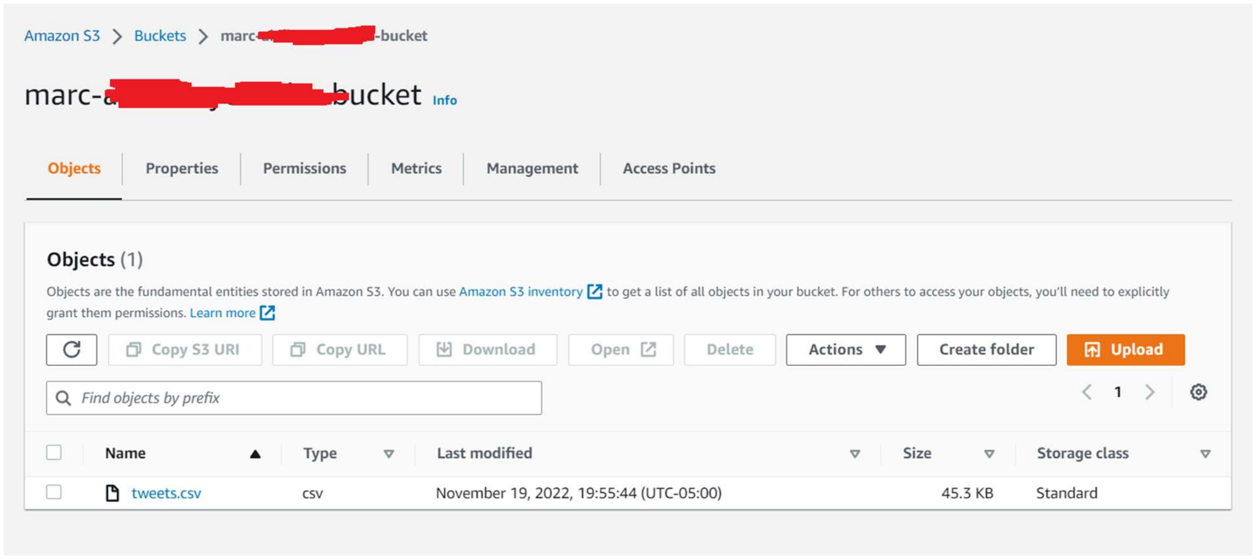
Step 5: Setting up Apache Airflow

After setting up the instance, we will run the standalone version of airflow and import our python script and the DAG into Airflow and run it.



The green bordered box signifies that the pipeline is up and running.

The final output is stored in an S3 bucket as a '.csv' file.



marc-[redacted]bucket Info

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
tweets.csv	csv	November 19, 2022, 19:55:44 (UTC-05:00)	45.3 KB	Standard

Limitations: Since this is just a project that sets up the following pipeline, I have not added many complications to the pipeline. We can add more transformations and analyses to the same code and make it better in the following ways.

1: Update every 1 hour.

2: Add/drop columns

3: Take inputs from the analyst and transform the data accordingly to perform analysis.