Twitter Data Pipeline using Airflow

The goal of this project is to extract Twitter data using the Twitter API, transform it using Python, deploy the code on Airflow/EC2, and store the final result on Amazon S3.

Phase 1 involves extracting data from Twitter by obtaining API credentials and generating an access token for the etl_airflow_project app. The code is written in Python and stored in a file called twitter_etl.py. The code is tested and then converted into a function called run_twitter_etl() to be imported through twitter_dag.py on the Airflow Server.

The Airflow phase includes creating an EC2 instance and installing Airflow, pandas, s3fs, and tweepy. Allow all HTTP traffic

Deploy Airflow on EC2 instance.

Bootstrapping Phase: Please see the file Bootstraping

The Airflow phase includes creating an EC2 instance and installing Airflow, pandas, s3fs, and tweepy. A folder called twitter_dag is created, and the dag_folder is modified to /home/ubuntu/airflow/twitter_dag, where the Python files are stored. The Twitter ETL code is split into two files: twitter_etl.py and twitter_dag.py, and they are both copied and pasted into this folder.

The Airflow Server is accessed through a web browser using the EC2 instance's DNS and port 8080**.**

To ensure security, access to the Airflow Server is restricted, and a security group is set up to allow access only from specific IP addresses. A unique S3 bucket is created to store the data created by this DAG, and access is granted to the EC2 instance to write to it.