Real-Time Twitter feed dashboard in Snowflake using Azure

Business Overview
Companies miss opportunities and are exposed to risk as a result of delays in company operations and decision-making. Organizations can move rapidly based on real-time data since it reveals issues and opportunities. Data that is gathered, processed, and evaluated in real-time is referred to as real-time data, and it comprises data that is ready to utilize as soon as it is created. A snapshot of historical data is what near real-time data is. When speed is critical, near real-time processing is preferred, although processing time in minutes rather than seconds is acceptable. Batch data that has been previously stored is considerably slower, and by the time it is ready to use, it might be days old.

In this project, we ingest generated Twitter feeds to Snowflake in near real-time that will power an in-built dashboard utility in Snowflake to obtain popularity feeds reports.

Data Pipeline
A data pipeline is a technique for transferring data from one system to another. The data may or may not be updated, and it may be handled in real-time (or streaming) rather than in batches. The data pipeline encompasses everything from harvesting or acquiring data using various methods to storing raw data, cleaning, validating, and transforming data into a query-worthy format, displaying KPIs, and managing the above process.

Dataset Description
We will use the Twitter API and fetch tweets and their metadata(re-tweets, comments, likes) using Python.

Tech Stack
➔Language: Python
➔Services: Azure Storage Account, Azure Queue, Snowpipe, Snowflake, Azure Resource Group

Snowflake
Snowflake is a data storage, processing, and analytics platform that blends a unique SQL query engine with a cloud-native architecture. Snowflake delivers all the features of an enterprise analytic database to the user. Snowflake components include:
- Warehouse/Virtual Warehouse
- Database and Schema
- Table
- View
- Stored procedure
- Snowpipe
- Stream
- Task

Azure Queue
Azure Queue Storage allows application components to communicate in the cloud. Application components are frequently separated when developing for scale so that they may scale independently. Queue Storage enables asynchronous communications between application components operating in the cloud, on the desktop, on an on-premises server, or a mobile device. Queue Storage also allows you to manage asynchronous activities and create process workflows.

Approach

- We write API calls to fetch Twitter insights in real-time via Python and this code can be run in a local machine every day once
- We create a Snowpipe component in Snowflakes by using Azure IAM integration(cross-account access) as Snowflakes hosted on the Azure account is different from the Azure account we own. This in turn uses Azure EventGrid and Function App in the backend to automate the file load
- As soon as the script generates files in Azure Blob storage, Snowpipe recognizes the file arrival and loads the snowflakes table with file data automatically
- We create a dashboard in Snowflakes that is scheduled to refresh every 30 mins to show actual feed data from Twitter, Eg. No of likes and comments/feed to understand popular feed and their sentiment.

Key Takeaways

- Understanding the project overview
- Creating Snowflake Account
- Setup Twitter developer account
- Deep Dive into Azure Storage Account
- Understanding different components of Azure Storage Account
- Azure Storage Account creation
- Creating Storage Queue
- Integrating Azure Queue and Snowflake
- Creating Snowpipe Objects
- Creating Snowflake Dashboard

Architecture