Katherine Gallagher

Topic: Amazon Kinesis Firehose
Project Summary

Project Description: I will be using Amazon Kinesis Firehose to stream filtered tweets in (near) real-time to Amazon S3 and Amazon Redshift.

Problem: Our increasingly connected world generates massive amounts of data. Amazon Kinesis originally sought to provide an integrated solution to data collection and analysis by providing a platform on which to build custom applications that capture, process, and store streaming data. While this proved useful for technologically literate users, Amazon realized that others would find a simple data delivery stream preferable. Amazon Kinesis Firehose, "the easiest way to load streaming data into AWS", was the result of this realization.

Description: Introduced in 2013, Amazon Kinesis Firehose is a pay-as-you-go autoscaling delivery stream capable of loading gigabytes of streaming data per second from hundreds of thousands of sources into Amazon S3 or Amazon Redshift in near real-time (within 60 seconds of sending). While Firehose allows considerable flexibility in detail specification, allowing users to set batch size, interval, compression, encryption, etc., it is a fully-managed service, and self-administers the resources required for data loading.

Benefits: The primary benefit to Firehose is ease of use; a delivery stream can be up and running with a few clicks in the AWS Kinesis console and requires no further administration. Rather than requiring a minimum spend, users pay per GB of data ingested, so Firehose is equally appealing to small and large data consumers. As the transport data unit is opaque to the delivery stream, Firehose does not discriminate between data formats.

Drawbacks: While the S3 integration appears seamless, I found Redshift less flexible to work with programmatically. That said, this is more a complaint with Redshift than Firehose, and can be easily circumvented by setting options in the AWS console or deploying from SQL.

Data Set: Filtered tweets captured from the Twitter Public Stream (Resource URL: https://stream.twitter.com/1.1/statuses/filter.json)

Operating System: Mac OS X Yosemite - Version 10.10.5

Software:
- SQL Workbench/J
- Eclipse Java EE IDE for Web Developers - Version Mars.1 Release (4.5.1)

Overview of steps:
- Set up Twitter account and Twitter application to obtain credentials for Twitter API calls
- Create S3 bucket, Redshift cluster, and Redshift table to hold incoming data
- Programmatically create a Firehose delivery stream connected to S3 and Redshift
- Programmatically connect to Twitter Public Stream and send tweets via Firehose

Links to YouTube videos:
- Short: https://youtu.be/hBHbgOtiy18
- Long: https://youtu.be/WyO30uaz7Fw