Build a Real-Time Spark Streaming Pipeline on AWS using Scala

Business Overview

Analyzing and measuring data as soon as it enters the database is referred to as real-time analytics. Thus, users gain insights or may conclude as soon as data enters their system. Businesses can react quickly using real-time analytics. They can grasp opportunities and avert issues before they occur.

On the other hand, Batch-style analytics might take hours or even days to provide findings. As a result, batch analytical systems frequently produce only static insights based on lagging indications. Real-time analytics insights may help organizations stay ahead of the competition. These pipelines for streaming data generally follow a 3 step process, i.e., Ingest, Analyze and Deliver.

We aim to build a Real-Time Spark Streaming Pipeline using AWS services like AWS S3, Amazon Lambda, Amazon Kinesis Data Streams, Amazon EMR, Amazon Kinesis Firehose, and OpenSearch. We will also use Kibana, a part of OpenSearch, for visualization.

Dataset Description

This GPS trajectory information was gathered over four years by 178 users in the (Microsoft Research) Geolife project (from April 2007 to October 2011). A GPS trajectory in this collection is represented as a series of time-stamped points, each comprising latitude, longitude, and altitude information. This dataset has 17,621 trajectories totaling 1,251,654 kilometers and 48,203 hours in time.

File format:

PLT file fields:

Lines 1...6 are useless in this dataset and can be ignored. Points are described in the following lines, one for each line.

- Field 1: Latitude in decimal degrees.
- Field 2: Longitude in decimal degrees.
- Field 3: All set to 0 for this dataset.
- Field 4: Altitude in feet (-777 if not valid).
- Field 5: Date - number of days (with a fractional part) that have passed since 12/30/1899.
- Field 6: Date as a string.
- Field 7: Time as a string.

Note that fields 5, 6, and 7 represent the same date/time in this dataset. You may use either of them.
Example: 39.906631,116.385564,0,492,40097.5864583333,2009-10-11,14:04:30

Tech Stack

➔

Language: Scala, Python

➔

Services: AWS S3, Amazon Lambda, Amazon Kinesis Data Streams, Amazon EMR, Amazon Kinesis Firehose, Amazon DynamoDB, OpenSearch, Kibana

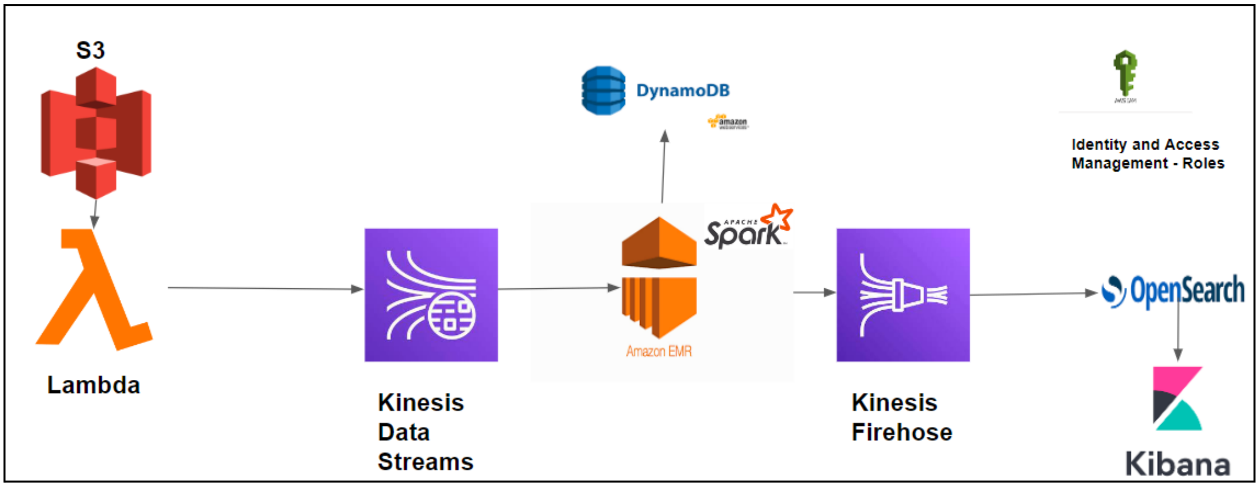
Key Takeaways

- Understanding the GPS Trajectory Dataset
- Understanding each component of the Pipeline in detail
- Creating AWS S3 bucket
- Uploading data to AWS S3 bucket
- Creating the Amazon Kinesis Data Streams
- Understanding the various configurations of Amazon Kinesis Data Streams
- Creating the Lambda function
- Packaging the necessary libraries for the Lambda function
- Understanding the various configurations of the Lambda function
- Add trigger event to the Lambda function
- Understanding the logs created on CloudWatch
- Creating an EC2 key pair for the EMR cluster
- Creating the Amazon EMR cluster
- Understanding the various configurations of the Amazon EMR cluster
- Sending data from AWS S3 to Amazon Kinesis Data Streams using the Lambda function
- Reading data from Amazon Kinesis Data Streams using EMR
- Creating the Amazon Kinesis Firehose delivery stream
- Writing data from Amazon Kinesis Data Streams into Amazon Kinesis Firehose using EMR
- Creating OpenSearch Domain
- Integrating OpenSearch with Amazon Kinesis Firehose delivery stream
- Creating index pattern in OpenSearch
- Creating visualizations in OpenSearch

Approach

1) The Amazon Lambda function will stream log files into Amazon Kinesis Data Streams.
2) EMR will run a spark streaming job to read from Kinesis Data Streams in real-time and load data in the required format in Kinesis Firehose
3) Firehose is used to collect transformed data and write to OpenSearch
4) We use Kibana, which is part of OpenSearch, for visualization

Architecture Diagram:


S3
DynamoDB
amazon
Identity and Access Management - Roles
Lambda
Spark
Amazon EMR
OpenSearch
Kinesis Data Streams
Kinesis Firehose
Kibana