UDACITY

‹ Return to "Data Streaming Nanodegree" in the classroom

DISCUSS ON STUDENT HUB

SF Crime Statistics with Spark Streaming

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Awesome work 🎉 ✨ 🎊

Great job !!! You have successfully completed this project. Kudos to you and keep up the good work !!! 👏👏👏👏👏👏👏👏

Please leave this review a good rating if you liked it.

Implement Apache Kafka Server

These two files - `kafka/config/server.properties` and `kafka/config/zookeeper.properties` - should have the correct port for the topics, as evidenced by matching ports in the 2 files `server.properties` and `producer_server.py`. (E.g., if the student used 9092 for the topic ports in the `producer_server.py`, these files should have changed). Additionally, the host for kafka/config/server.properties should be changed to localhost.

Awesome

You have used the correct port and have changed the host to localhost. Good job !

The consumer-console logs from the terminal imply that the student has set up the Kafka/Zookeeper server and Kafka topic correctly (Kafka host/port number matches in the code and the configuration files, topic name is written out in the `producer_server.py`, and the same topic is created for Kafka producer). The logs show the byte stream from the Kafka server is being consumed by the consumer if the proper consumer is given.

Student has implemented a Kafka consumer in `consumer_server.py` and demonstrated that the consumer is able to consume data. The screenshot of the kafka-consumer-console output demonstrates that the students were able to send out a byte stream from Kafka Producer, and the rows should be in json (or python dict) format.

Extra Resources

Thorough Introduction to Apache Kafka
Sneak Peek into Apache Zookeeper

Implement Apache Spark Structured Streaming

Use various configs in SparkSession, set up for local mode (using local[*]) and Spark readstream configurations. The student should be using at least format ('kafka'), 'kafka.bootstrap.servers', and 'subscribe'. Other configuration properties such as 'startingOffsets' and 'maxRatePerPartition' are optional. This should act as a broker internally tied with Spark, and the student should be able to consume the data that they generated from the local Kafka server to the Spark Structured Streaming.

Student demonstrates in their answer to the first question that they understand how changing values of the SparkSession property parameters affects the throughput and latency of their data.

The screenshot of the Spark UI on the Streaming tab demonstrates that the student used proper properties to render the Spark UI and monitor their jobs.

Awesome

The Apache Spark structured streaming is properly implemented, the data generated from the local Kafka server is correctly consumed to the Spark structured streaming.

Extra Resources

Structured Streaming Programming Guide
Structured streaming in a flash

From the data that is coming in from the Kafka producer, the student should be investigating the schema of the data (key, value, topic, partition, offset, timestamp, timestampType). The student should successfully select the relevant column (value) from the data and investigate the JSON object (key-value pair).

Using this information about the JSON object, the student should create a proper schema for Apache Spark Structured Streaming. This schema will check data types, and will be used with Spark Catalyst Optimizers.

Awesome

The schema of the data is investigated properly and the relevant column (value) is selected correctly. An appropriate schema is created for Apache Spark Structured Streaming.

Extra Resource

Deep Dive into Spark Catalyst Optimizer

Aggregator/Join logic should show different types of crimes occurred in certain time frames (using window functions, group by original_crime_time), and how many calls occurred in certain time frames (group by on certain time frame, but the student will need to create a user defined function internally for Spark to change time format first to do this operation). Any aggregator or join logic should be performed with Watermark API. The students can play around with the time frame - this is part of their analytical exercises and data exploration.

Student demonstrates in their response to question 2 that they understand, given a resource and system, how to find out the most optimal configuration for their application.

The screenshot of the progress reporter demonstrates that the student correctly ingested the data in a micro-batch fashion and was able to monitor each micro-batch through the console.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review