Scripts Execution

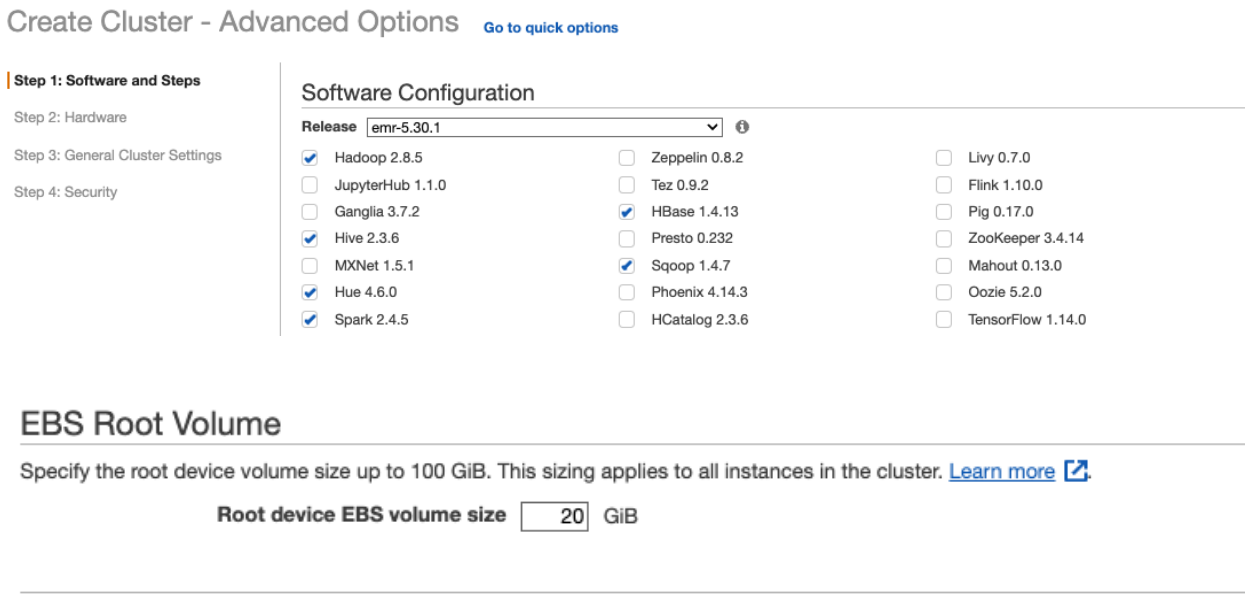
Explanation of the solution to the streaming layer problem

Task 5: Create a streaming data processing framework that ingests real-time POS transaction data from Kafka. The transaction data is then validated based on the three rules' parameters (stored in the NoSQL database) discussed previously.

Task 6: Update the transactions data along with the status (fraud/genuine) in the card_transactions table.

Task 7: Store the 'postcode' and 'transaction_dt' of the current transaction in the look-up table in the NoSQL database if the transaction was classified as genuine.

The below EMR cluster is set up with **HADOOP, SQOOP, HIVE, HBASE AND SPARK** as the software configuration with a **EBS volume of 20 GB.**



© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

1) Login to the EMR Cluster & switch to the root user to run the **pip-install kafka-python.**

```
[root@ip-172-31-67-222 ~]# pip install kafka-python
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip3 install --user` instead.
Collecting kafka-python
  Downloading https://files.pythonhosted.org/packages/75/68/dcb0db055309f680ab2931a3eeb22d865604b638acf8c914bedf4c1a0c8c/kafka_python-2.0.2-py2.py3-none-any.whl (246kB)
    100% |████████████████████████████████| 256kB 4.8MB/s
Installing collected packages: kafka-python
Successfully installed kafka-python-2.0.2
[root@ip-172-31-67-222 ~]#
```

2) Run the following commands in order to Install Happy base and ensure that the thrift server is started.
   a. **sudo yum update**
   b. **sudo yum install python3-devel**
   c. **pip install happybase**

3) Download **db-> dao.py , geomap.py ,rules-> rules.py ,driver.py ,unzipsv.csv** from the resource section of the capstone project

```
[hadoop@ip-172-31-68-201 src]$ ls
db  driver.py  __init__.py  rules  uszipsv.csv
[hadoop@ip-172-31-68-201 src]$
```

4) Updated the public IP of my Instance in **dao.py** file as "**3.95.27.224"**

```
class HBaseDao:
        """
        Dao class for operation on HBase
        """
        __instance = None

        @staticmethod
        def get_instance():
                """ Static access method. """
                if HBaseDao.__instance == None:
                        HBaseDao()
                return HBaseDao.__instance

        def __init__(self):
                if HBaseDao.__instance != None:
                        raise Exception("This class is a singleton!")
                else:
                        HBaseDao.__instance = self
                        self.host = '3.95.27.224'
                        #self.host = 'localhost'
                        for i in range(2):
                                try:
                                        self.pool = happybase.ConnectionPool(size=3, host=self.host, port=9090)
                                        break
                                except:
                                        print("Exception in connecting HBase")
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

5) Updated **rules.py** with following parameters:

lookup_table = 'lookup_data_hbase'

master_table = 'card_transactions_hbase'

6) Python functions created with logic for UDFs **(rules.py)**

```
def verify_ucl_data(card_id, amount):
    try:
        hbasedao = HBaseDao.get_instance()

        card_row = hbasedao.get_data(key=str(card_id), table=lookup_table)
        card_ucl = (card_row[b'card_data:ucl']).decode("utf-8")

        if amount < float(card_ucl):
            return True
        else:
            return False
    except Exception as e:
        raise Exception(e)
```

7) Function to verify the Credit score rule. Credit score of each member should be greater than 200.

```
def verify_credit_score_data(card_id):

    try:
        hbasedao = HBaseDao.get_instance()

        card_row = hbasedao.get_data(key=str(card_id), table=lookup_table)
        card_score = (card_row[b'card_data:score']).decode("utf-8")

        if int(card_score) > 200:
            return True
        else:
            return False
    except Exception as e:
        raise Exception(e)
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

8) Function to verify the Zipcode rules.

```
def verify_postcode_data(card_id, postcode, transaction_dt):

    try:
        hbasedao = HBaseDao.get_instance()
        geo_map = GEO_Map.get_instance()

        card_row = hbasedao.get_data(key=str(card_id), table=lookup_table)
        last_postcode = (card_row[b'card_data:postcode']).decode("utf-8")
        last_transaction_dt = (card_row[b'card_data:transaction_dt']).decode("utf-8")

        current_lat = geo_map.get_lat(str(postcode))
        current_lon = geo_map.get_long(str(postcode))
        previous_lat = geo_map.get_lat(last_postcode)
        previous_lon = geo_map.get_long(last_postcode)

        dist = geo_map.distance(lat1=current_lat, long1=current_lon, lat2=previous_lat, long2=previous_lon)

        speed = calculate_speed(dist, transaction_dt, last_transaction_dt)

        if speed < speed_threshold:
            return True
        else:
            return False

    except Exception as e:
        raise Exception(e)
```

9) Function to calculate the speed from transaction timestamp and the distance.

```
def calculate_speed(dist, transaction_dt1, transaction_dt2):

    transaction_dt1 = datetime.strptime(transaction_dt1, '%d-%m-%Y %H:%M:%S')
    transaction_dt2 = datetime.strptime(transaction_dt2, '%d-%m-%Y %H:%M:%S')

    elapsed_time = transaction_dt1 - transaction_dt2
    elapsed_time = elapsed_time.total_seconds()

    try:
        return dist / elapsed_time
    except ZeroDivisionError:
        return 299792.458
# (Speed of light)
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

10) Functions to verify UCL, Credit score and Speed rules.

```
def verify_rules_status(card_id, member_id, amount, pos_id, postcode, transaction_dt):


    hbasedao = HBaseDao.get_instance()

    # Check if the POS transaction passes all rules.
    # If yes, update the lookup table and insert data in master table as genuine.
    # Else insert the transaction in master table as Fraud.

    rule1 = verify_ucl_data(card_id, amount)
    rule2 = verify_credit_score_data(card_id)
    rule3 = verify_postcode_data(card_id, postcode, transaction_dt)

    if all([rule1, rule2, rule3]):
        status = 'GENUINE'
        hbasedao.write_data(key=str(card_id),
                            row={'card_data:postcode': str(postcode), 'card_data:transaction_dt': str(transaction_dt)},
                            table=lookup_table)
    else:
        status = 'FRAUD'

    new_id = str(uuid.uuid4()).replace('-', '')
    hbasedao.write_data(key=new_id,
                        row={'cardDetail:card_id': str(card_id), 'cardDetail:member_id': str(member_id),
                             'transactionDetail:amount': str(amount), 'transactionDetail:pos_id': str(pos_id),
                             'transactionDetail:postcode': str(postcode), 'transactionDetail:status': str(status),
                             'transactionDetail:transaction_dt': str(transaction_dt)},
                        table=master_table)
```

11) Importing the necessary libraries in the "**driver.py"** file

```
#importing necessary libraries
import os
import sys
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *
from rules.rules import *
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

12) Initializing the Spark session

```
#initialising Spark session
spark = SparkSession  \
    .builder  \
    .appName("CreditCardFraud")  \
    .getOrCreate()
spark.sparkContext.setLogLevel('ERROR')
```

13) Reading input data from Kafka mentioning the details of the Kafka broker, such as bootstrap server, port and topic name

```
# Reading input from Kafka
credit_data = spark.readStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers", "18.211.252.152:9092") \
    .option("startingOffsets","earliest") \
    .option("failOnDataLoss", "false") \
    .option("subscribe", "transactions-topic-verified") \
    .load()
```

14) Defining JSON schema of each transactions

```
# Defining schema for transaction
dataSchema = StructType() \
    .add("card_id", LongType()) \
    .add("member_id", LongType()) \
    .add("amount", DoubleType()) \
    .add("pos_id", LongType()) \
    .add("postcode", IntegerType()) \
    .add("transaction_dt", StringType())
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

15) Reading JSON Data from Kafka as Credit_Data and define UDF to verify the rules

```
# Casting raw data as string and aliasing
credit_data = credit_data.selectExpr("cast(value as string)")
credit_data_stream = credit_data.select(from_json(col="value", schema=dataSchema).alias("credit_data")).select(
    "credit_data.*")

# Define UDF which verifies all the rules for each transaction and updates the lookup and master tables
verify_all_rules = udf(verify_rules_status, StringType())

Final_data = credit_data_stream \
    .withColumn('status', verify_all_rules(credit_data_stream['card_id'],
                                           credit_data_stream['member_id'],
                                           credit_data_stream['amount'],
                                           credit_data_stream['pos_id'],
                                           credit_data_stream['postcode'],
                                           credit_data_stream['transaction_dt']))
```

16) Code to display output in console.

```
# Write output to console as well
output_data = Final_data \
    .select("card_id", "member_id", "amount", "pos_id", "postcode", "transaction_dt") \
    .writeStream \
    .outputMode("append") \
    .format("console") \
    .option("truncate", False) \
    .start()
```

17) Set the Kafka version and run the spark submit command.

```
[hadoop@ip-172-31-68-201 src]$ export SPARK_KAFKA_VERSION=0.10
[hadoop@ip-172-31-68-201 src]$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 driver.py
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved

```
[hadoop@ip-172-31-68-201 src]$ export SPARK_KAFKA_VERSION=0.10
[hadoop@ip-172-31-68-201 src]$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 driver.py
Ivy Default Cache set to: /home/hadoop/.ivy2/cache
The jars for the packages stored in: /home/hadoop/.ivy2/jars
:: loading settings :: url = jar:file:/usr/lib/spark/jars/ivy-2.4.0.jar!/org/apache/ivy/core/settings/ivysettings.xml
org.apache.spark#spark-sql-kafka-0-10_2.11 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-e2d9dbe3-1fd3-4138-b69e-11a77d71e836;1.0
        confs: [default]
        found org.apache.spark#spark-sql-kafka-0-10_2.11;2.4.5 in central
        found org.apache.kafka#kafka-clients;2.0.0 in central
        found org.lz4#lz4-java;1.4.0 in central
        found org.xerial.snappy#snappy-java;1.1.7.3 in central
        found org.slf4j#slf4j-api;1.7.16 in central
        found org.spark-project.spark#unused;1.0.0 in central
:: resolution report :: resolve 502ms :: artifacts dl 12ms
        :: modules in use:
        org.apache.kafka#kafka-clients;2.0.0 from central in [default]
        org.apache.spark#spark-sql-kafka-0-10_2.11;2.4.5 from central in [default]
        org.lz4#lz4-java;1.4.0 from central in [default]
        org.slf4j#slf4j-api;1.7.16 from central in [default]
        org.spark-project.spark#unused;1.0.0 from central in [default]
        org.xerial.snappy#snappy-java;1.1.7.3 from central in [default]
        ---------------------------------------------------------------------
        |                  |            modules            ||   artifacts   |
        |       conf       | number| search|dwnlded|evicted|| number|dwnlded|
        ---------------------------------------------------------------------
        |      default     |   6   |   0   |   0   |   0   ||   6   |   0   |
        ---------------------------------------------------------------------
:: retrieving :: org.apache.spark#spark-submit-parent-e2d9dbe3-1fd3-4138-b69e-11a77d71e836
        confs: [default]
        0 artifacts copied, 6 already retrieved (0kB/11ms)
23/04/01 17:23:22 INFO SparkContext: Running Spark version 2.4.5-amzn-0
23/04/01 17:23:22 INFO SparkContext: Submitted application: CreditCardFraud
23/04/01 17:23:22 INFO SecurityManager: Changing view acls to: hadoop
23/04/01 17:23:22 INFO SecurityManager: Changing modify acls to: hadoop
23/04/01 17:23:22 INFO SecurityManager: Changing view acls groups to:
23/04/01 17:23:22 INFO SecurityManager: Changing modify acls groups to:
23/04/01 17:23:22 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permissions: Set(hadoop); groups with view permissions: Set
(); users  with modify permissions: Set(hadoop); groups with modify permissions: Set()
23/04/01 17:23:22 INFO Utils: Successfully started service 'sparkDriver' on port 38555.
23/04/01 17:23:22 INFO SparkEnv: Registering MapOutputTracker
23/04/01 17:23:23 INFO SparkEnv: Registering BlockManagerMaster
23/04/01 17:23:23 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
23/04/01 17:23:23 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
23/04/01 17:23:23 INFO DiskBlockManager: Created local directory at /mnt/tmp/blockmgr-25501d10-a5fe-43a6-aa18-06a4bc63a6ae
23/04/01 17:23:23 INFO MemoryStore: MemoryStore started with capacity 1028.8 MB
23/04/01 17:23:23 INFO SparkEnv: Registering OutputCommitCoordinator
23/04/01 17:23:23 INFO Utils: Successfully started service 'SparkUI' on port 4040.
```

18) Check the output in Console :

```
atch: 0
-------------------------------------------
+----------------+-------------+-------+---------------+--------+-------------------+--------+
|card_id         |member_id    |amount |pos_id         |postcode|transaction_dt_ts  |status  |
+----------------+-------------+-------+---------------+--------+-------------------+--------+
|348702330256514 |37495066290  |4380912|248063406800722|96774   |2017-12-31 08:24:29|GENUINE|
|348702330256514 |37495066290  |6703385|786562777140812|84758   |2017-12-31 04:15:03|FRAUD  |
|348702330256514 |37495066290  |7454328|466952571393508|93645   |2017-12-31 09:56:42|GENUINE|
|348702330256514 |37495066290  |4013428|45845320330319 |15868   |2017-12-31 05:38:54|GENUINE|
|348702330256514 |37495066290  |5495353|545499621965697|79033   |2017-12-31 21:51:54|GENUINE|
|348702330256514 |37495066290  |3966214|369266342272501|22832   |2017-12-31 03:52:51|GENUINE|
|348702330256514 |37495066290  |1753644|9475029292671  |17923   |2017-12-31 00:11:30|FRAUD  |
|348702330256514 |37495066290  |1692115|27647525195860 |55708   |2017-12-31 17:02:39|GENUINE|
|5189563368503974|1178263015 30|9222134|525701337355194|64002   |2017-12-31 20:22:10|GENUINE|
|5189563368503974|1178263015 30|4133848|1820313834431 15|26346   |2017-12-31 01:52:32|FRAUD  |
|5189563368503974|1178263015 30|8938921|799748246411019|76934   |2017-12-31 05:20:53|FRAUD  |
|5189563368503974|1178263015 30|1786366|131276818071265|63431   |2017-12-31 14:29:38|GENUINE|
|5189563368503974|1178263015 30|9142237|564240259678903|50635   |2017-12-31 19:37:19|GENUINE|
|5407073344486464|1147922084344|6885448|887913906711117|59031   |2017-12-31 07:53:53|FRAUD  |
|5407073344486464|1147922084344|4028209|116266051118182|80118   |2017-12-31 01:06:50|FRAUD  |
|5407073344486464|1147922084344|3858369|896105817613325|53820   |2017-12-31 17:37:26|GENUINE|
|5407073344486464|1147922084344|9307733|729374116016479|14898   |2017-12-31 04:50:16|FRAUD  |
|5407073344486464|1147922084344|4011296|543373367319647|44028   |2017-12-31 13:09:34|GENUINE|
|5407073344486464|1147922084344|9492531|211980095659371|49453   |2017-12-31 14:12:26|GENUINE|
|5407073344486464|1147922084344|7550074|345533088112099|15030   |2017-12-31 02:34:52|FRAUD  |
+----------------+-------------+-------+---------------+--------+-------------------+--------+
nly showing top 20 rows
```

© Copyright 2020. upGrad Education Pvt. Ltd. All rights reserved