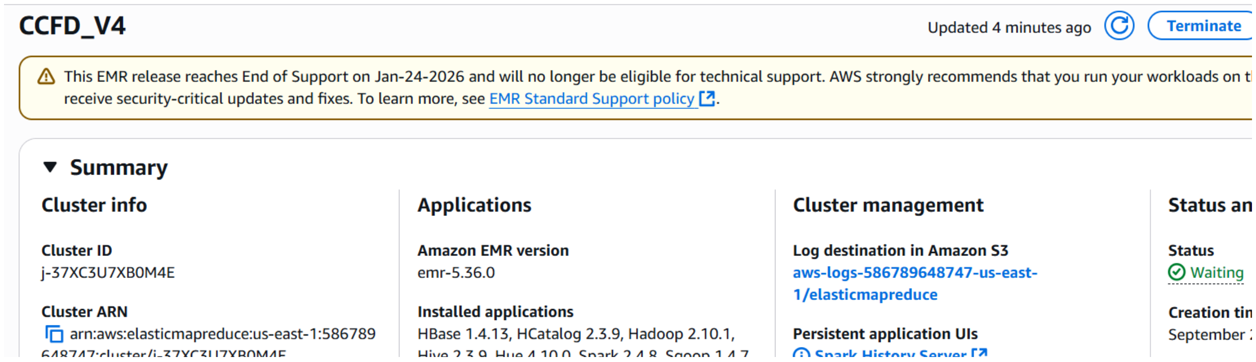
Scripts Execution

Explanation of the solution to the streaming layer problem

1. Created cluster.



CCFD_V4

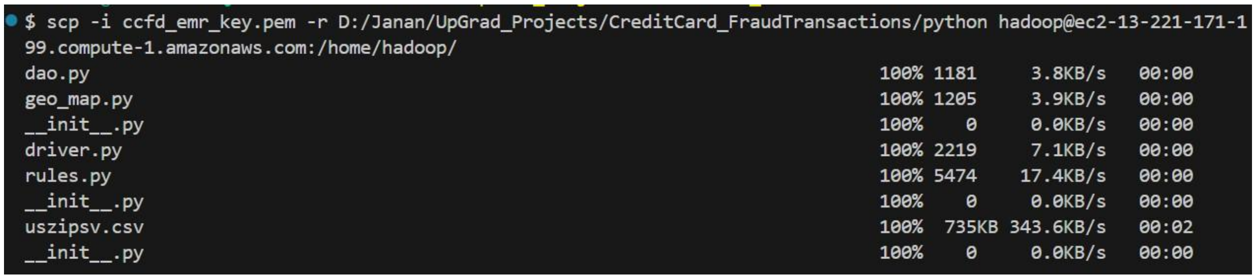
Updated 4 minutes ago

This EMR release reaches End of Support on Jan-24-2026 and will no longer be eligible for technical support. AWS strongly recommends that you run your workloads on t
receive security-critical updates and fixes. To learn more, see EMR Standard Support policy.

Summary

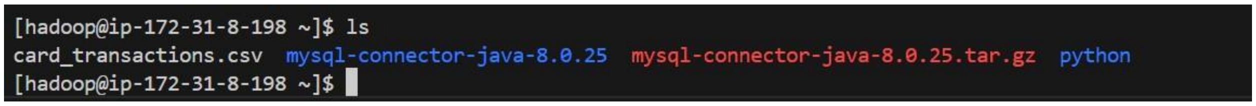
Cluster info	Applications	Cluster management	Status an
Cluster ID j-37XC3U7XB0M4E	**Amazon EMR version** emr-5.36.0	**Log destination in Amazon S3** aws-logs-586789648747-us-east-1/elasticmapreduce	**Status** Waiting
Cluster ARN arn:aws:elasticmapreduce:us-east-1:586789648747:cluster/j-37XC3U7XB0M4E	**Installed applications** HBase 1.4.13, HCatalog 2.3.9, Hadoop 2.10.1, Hive 2.3.9, Hue 4.10.0, Spark 2.4.8, Sqoop 1.4.7	**Persistent application UIs** Spark History Server	**Creation tin** September

Console Screenshot 1: Created EMR with Hadoop, Hive, HBase, HCatalog, Spark, ZooKeeper, Hue and Sqoop.

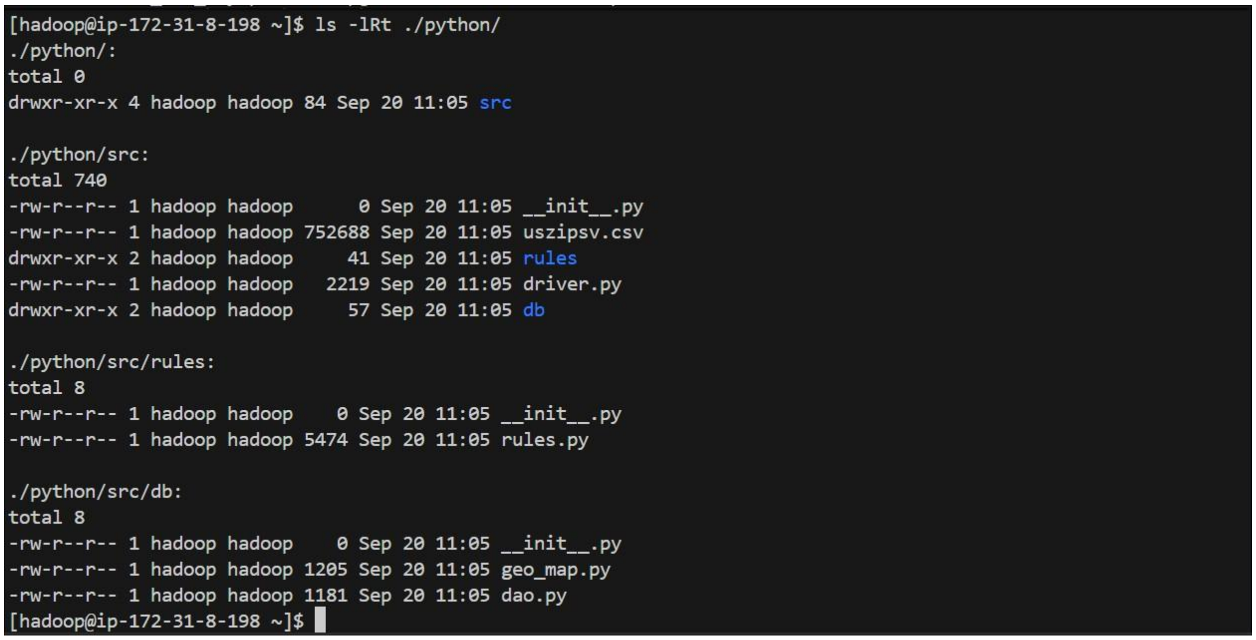


```
$ scp -i ccfd_emr_key.pem -r D:/Janan/UpGrad_Projects/CreditCard_FraudTransactions/python hadoop@ec2-13-221-171-199.compute-1.amazonaws.com:/home/hadoop/
dao.py                                                  100% 1181     3.8KB/s   00:00
geo_map.py                                              100% 1205     3.9KB/s   00:00
__init__.py                                             100%    0     0.0KB/s   00:00
driver.py                                               100% 2219     7.1KB/s   00:00
rules.py                                                100% 5474    17.4KB/s   00:00
__init__.py                                             100%    0     0.0KB/s   00:00
uszipsv.csv                                             100%  735KB 343.6KB/s   00:02
__init__.py                                             100%    0     0.0KB/s   00:00
```

2. Transferring Sample data to the cluster

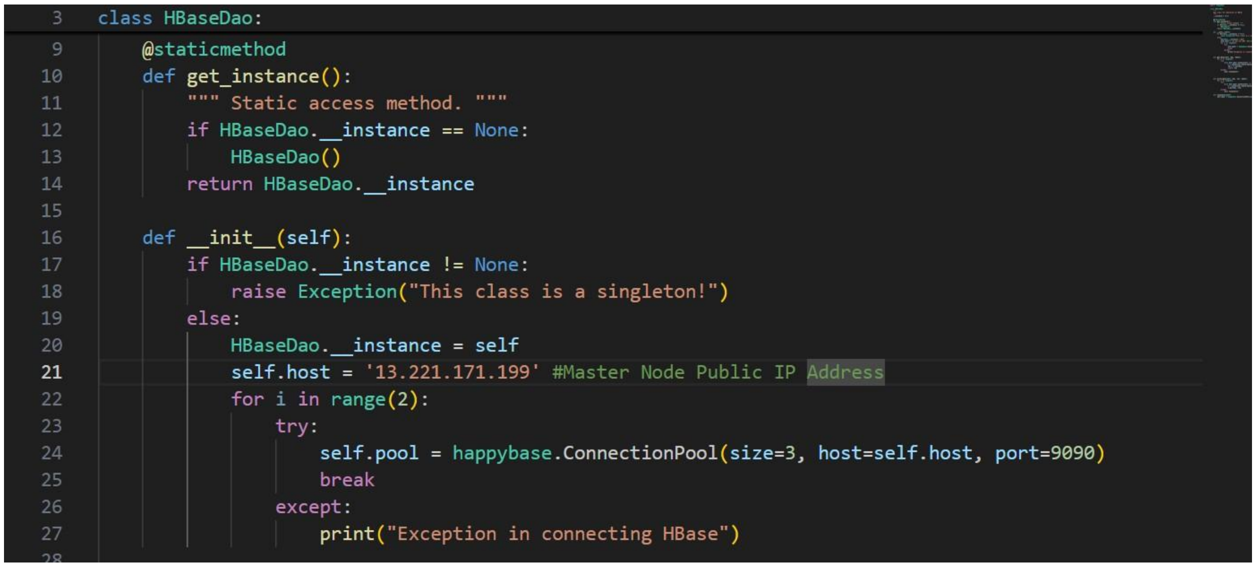


```
[hadoop@ip-172-31-8-198 ~]$ ls
card_transactions.csv  mysql-connector-java-8.0.25  mysql-connector-java-8.0.25.tar.gz  python
[hadoop@ip-172-31-8-198 ~]$
```



```
[hadoop@ip-172-31-8-198 ~]$ ls -lRt ./python/
./python/:
total 0
drwxr-xr-x 4 hadoop hadoop 84 Sep 20 11:05 src

./python/src:
total 740
-rw-r--r-- 1 hadoop hadoop      0 Sep 20 11:05 __init__.py
-rw-r--r-- 1 hadoop hadoop 752688 Sep 20 11:05 uszipsv.csv
drwxr-xr-x 2 hadoop hadoop     41 Sep 20 11:05 rules
-rw-r--r-- 1 hadoop hadoop   2219 Sep 20 11:05 driver.py
drwxr-xr-x 2 hadoop hadoop     57 Sep 20 11:05 db

./python/src/rules:
total 8
-rw-r--r-- 1 hadoop hadoop    0 Sep 20 11:05 __init__.py
-rw-r--r-- 1 hadoop hadoop 5474 Sep 20 11:05 rules.py

./python/src/db:
total 8
-rw-r--r-- 1 hadoop hadoop    0 Sep 20 11:05 __init__.py
-rw-r--r-- 1 hadoop hadoop 1205 Sep 20 11:05 geo_map.py
-rw-r--r-- 1 hadoop hadoop 1181 Sep 20 11:05 dao.py
[hadoop@ip-172-31-8-198 ~]$
```

3. Replacing self.host with Public IP address of our Master node.

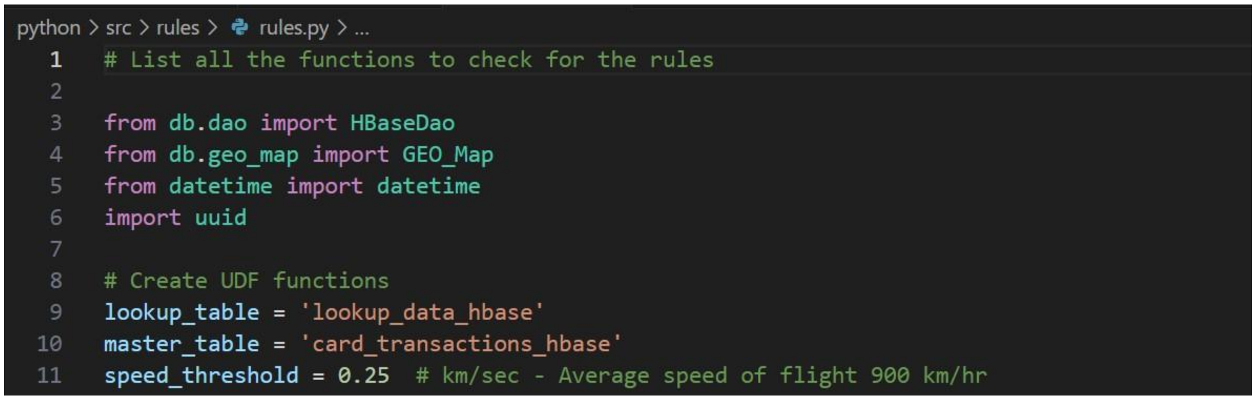


```
class HBaseDao:
    @staticmethod
    def get_instance():
        """ Static access method. """
        if HBaseDao.__instance == None:
            HBaseDao()
        return HBaseDao.__instance

    def __init__(self):
        if HBaseDao.__instance != None:
            raise Exception("This class is a singleton!")
        else:
            HBaseDao.__instance = self
            self.host = '13.221.171.199' #Master Node Public IP Address
            for i in range(2):
                try:
                    self.pool = happybase.ConnectionPool(size=3, host=self.host, port=9090)
                    break
                except:
                    print("Exception in connecting HBase")
```

4. Updating rules.py with:

lookup_table = 'lookup_data_hbase'

master_table = 'card_transactions_hbase'

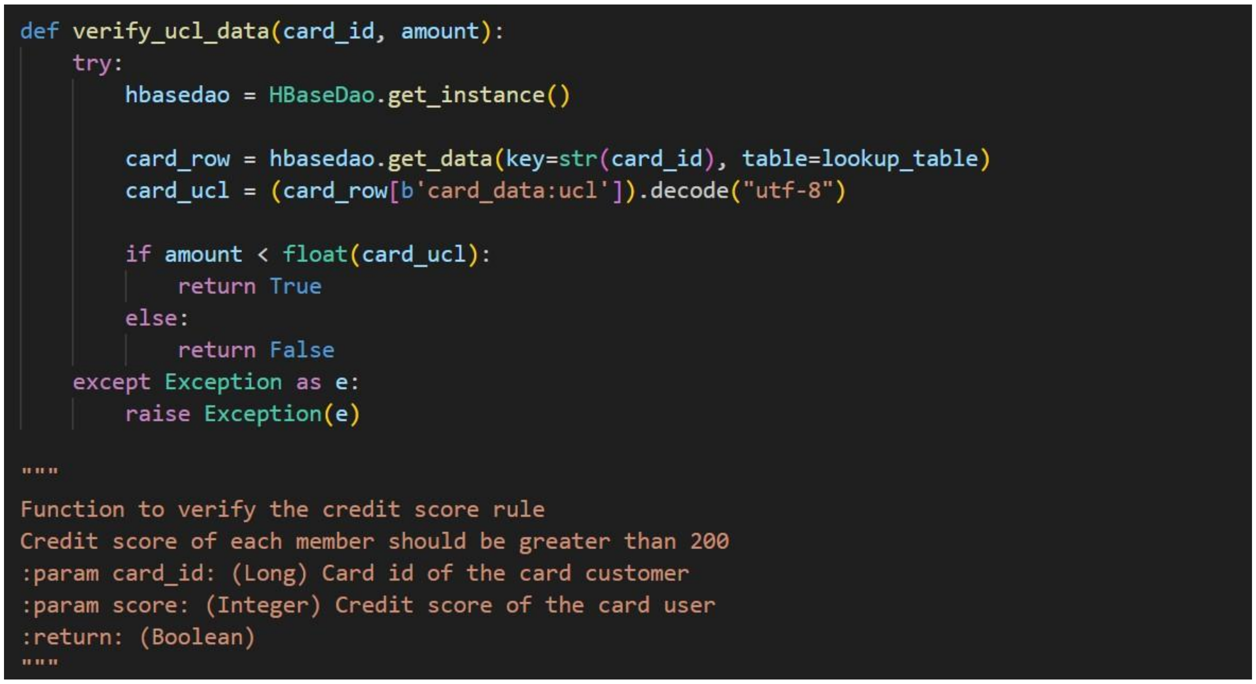


python > src > rules > rules.py > ...

```python
# List all the functions to check for the rules

from db.dao import HBaseDao
from db.geo_map import GEO_Map
from datetime import datetime
import uuid

# Create UDF functions
lookup_table = 'lookup_data_hbase'
master_table = 'card_transactions_hbase'
speed_threshold = 0.25  # km/sec - Average speed of flight 900 km/hr
```

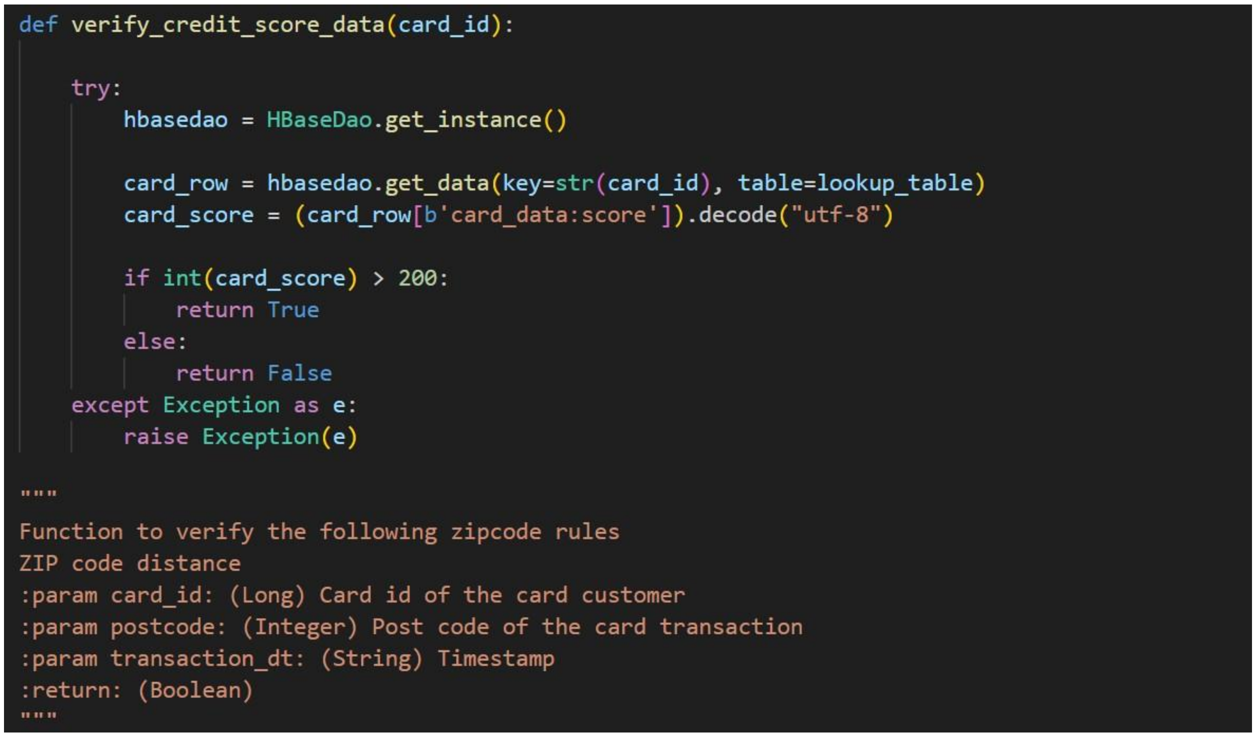
SS: Created UDF function

5. Created Python functions, containing the logic for the UDFs (rules.py) verify_ucl_data : Function to verify the UCL rule Transaction amount should be less than Upper control limit (UCL)



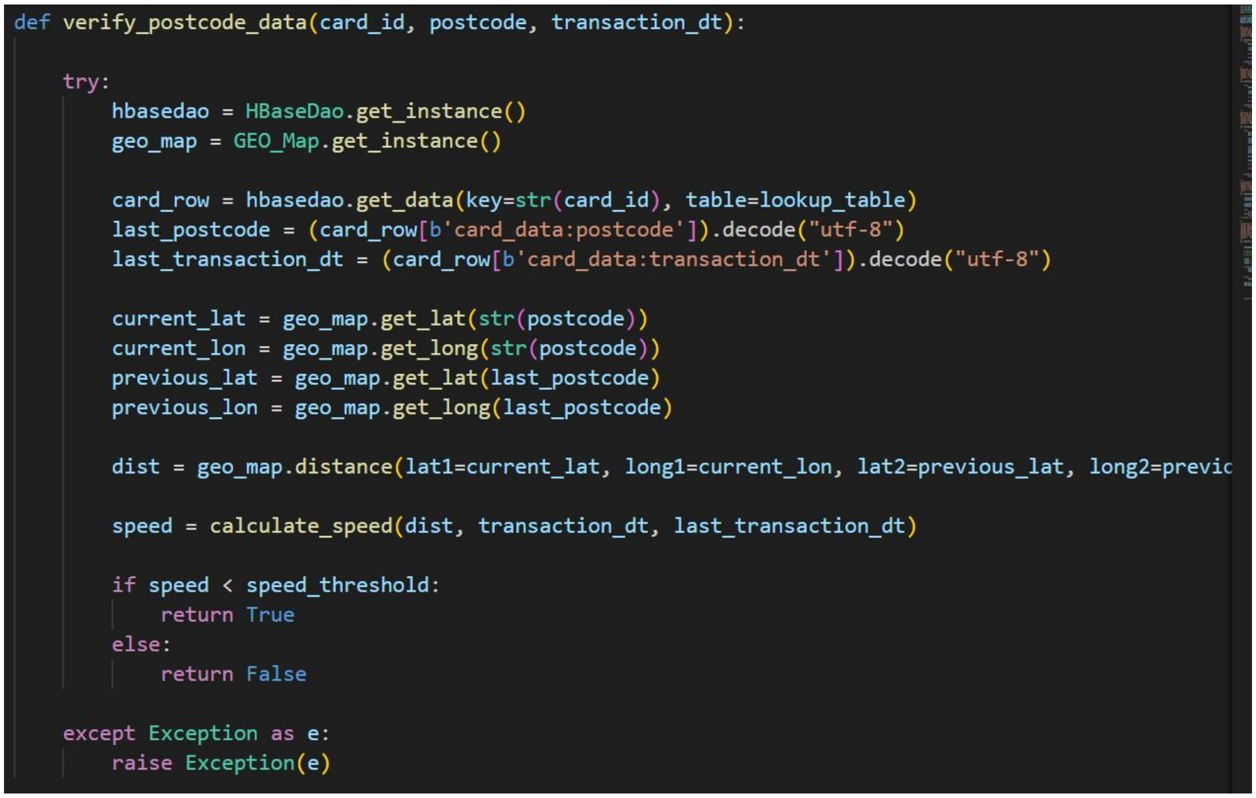
```python
def verify_ucl_data(card_id, amount):
    try:
        hbasedao = HBaseDao.get_instance()

        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)

"""
Function to verify the credit score rule
Credit score of each member should be greater than 200
:param card_id: (Long) Card id of the card customer
:param score: (Integer) Credit score of the card user
:return: (Boolean)
"""
```

6. verify_credit_score_data: Function to verify the credit score rule .Credit score of each member should be greater than 200.



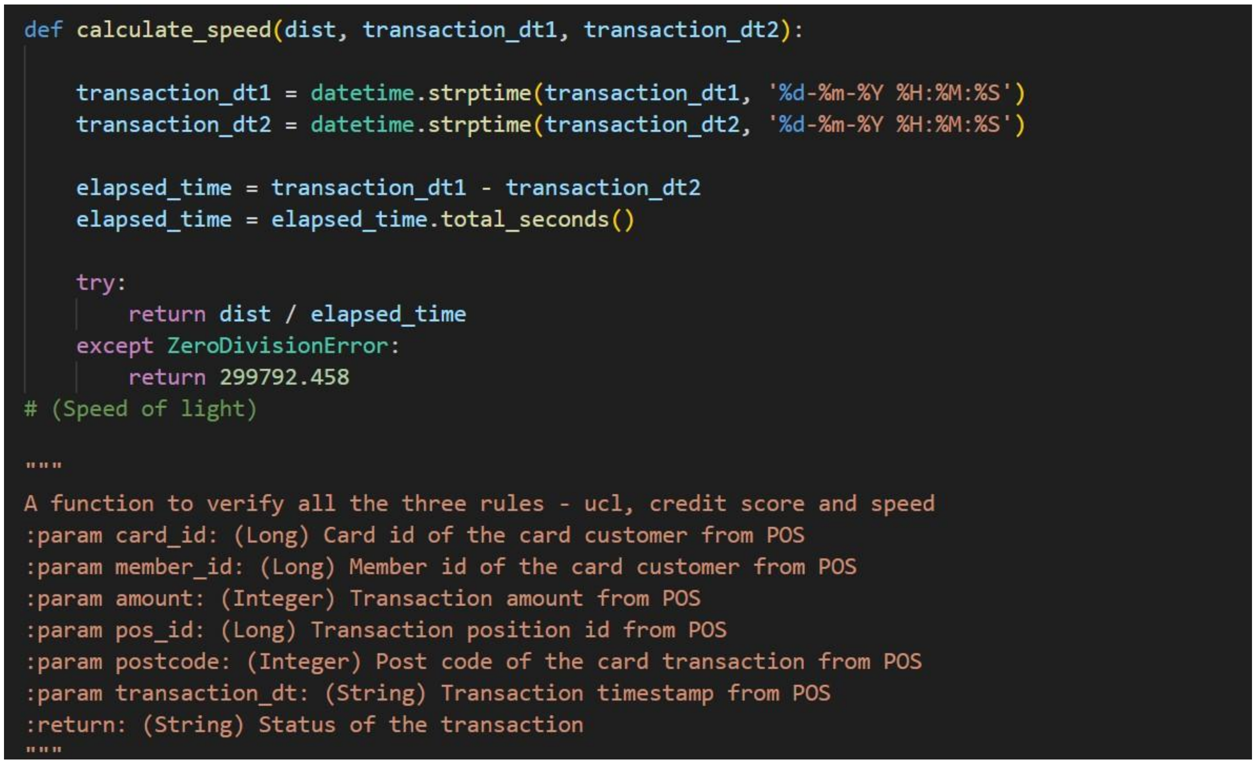
```python
def verify_credit_score_data(card_id):

    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)

"""
Function to verify the following zipcode rules
ZIP code distance
:param card_id: (Long) Card id of the card customer
:param postcode: (Integer) Post code of the card transaction
:param transaction_dt: (String) Timestamp
:return: (Boolean)
"""
```

7. verify_postcode_data: Function to verify the following zipcode rules. ZIP code distance.



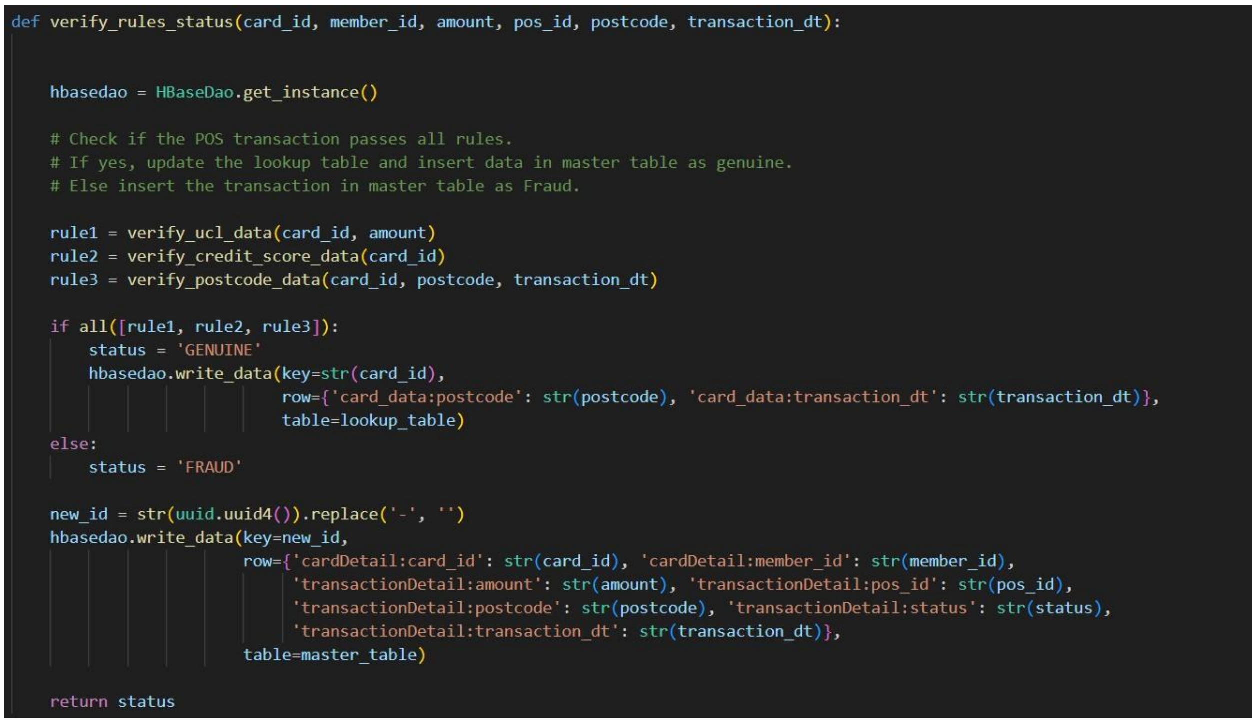
```python
def verify_postcode_data(card_id, postcode, transaction_dt):

    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=previc

        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)
```

8. calculate_speed: A function to calculate the speed from distance and transaction timestamp differentials.

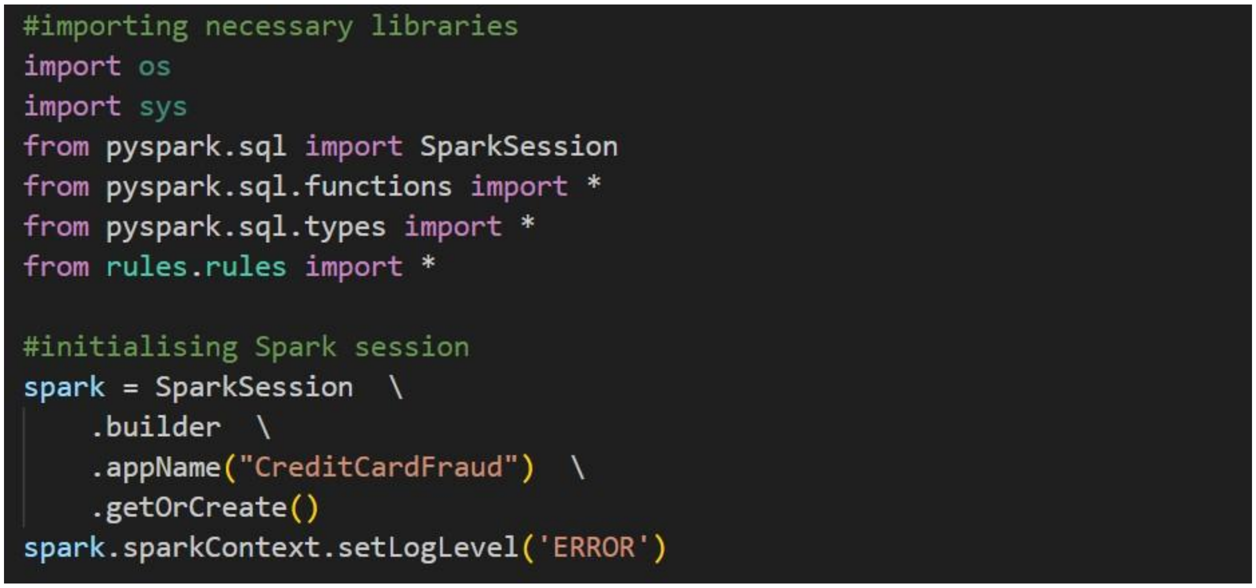


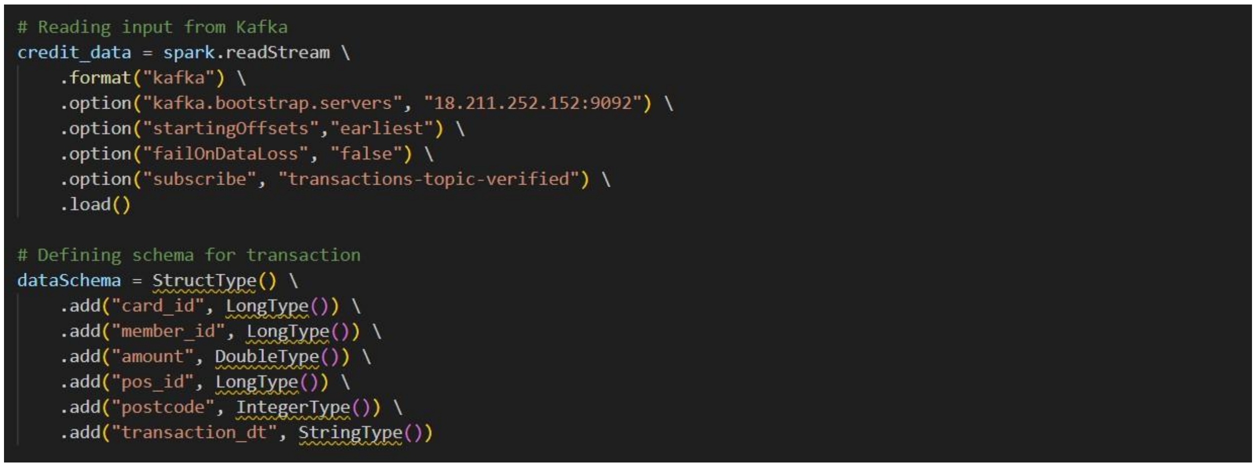
```python
def calculate_speed(dist, transaction_dt1, transaction_dt2):

    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)

"""
A function to verify all the three rules - ucl, credit score and speed
:param card_id: (Long) Card id of the card customer from POS
:param member_id: (Long) Member id of the card customer from POS
:param amount: (Integer) Transaction amount from POS
:param pos_id: (Long) Transaction position id from POS
:param postcode: (Integer) Post code of the card transaction from POS
:param transaction_dt: (String) Transaction timestamp from POS
:return: (String) Status of the transaction
"""
```

9. verify_rules_status: A function to verify all the three rules - ucl, credit score and speed.

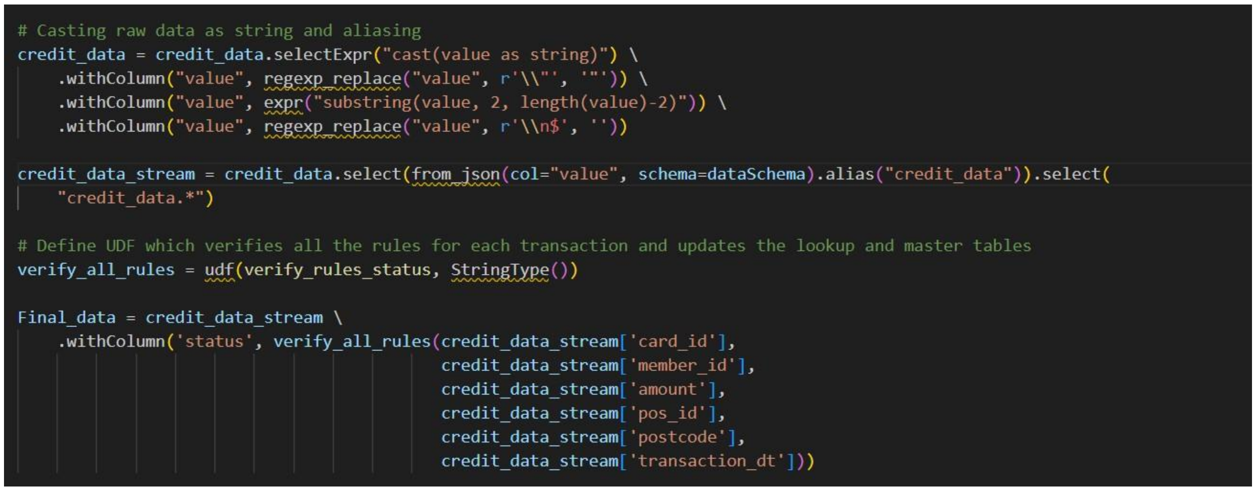

```python
def verify_rules_status(card_id, member_id, amount, pos_id, postcode, transaction_dt):


    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)

    return status
```

10. Now we are updating the driver.py

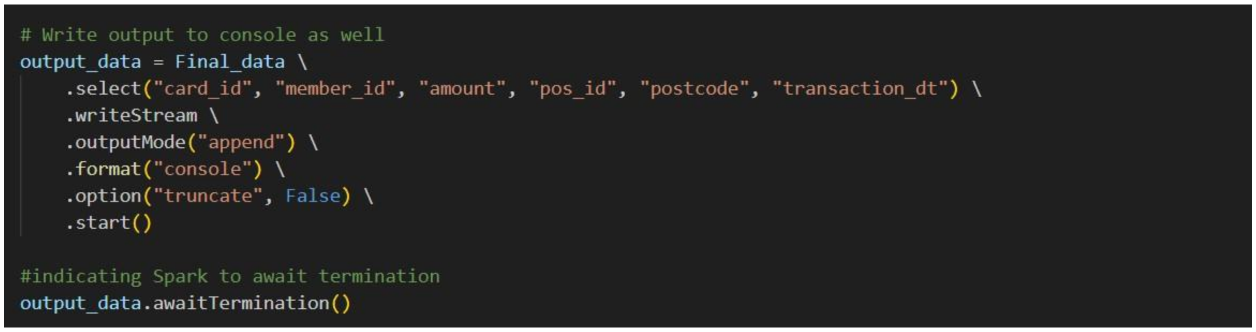

```python
#importing necessary libraries
import os
import sys
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *
from rules.rules import *

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



```
# 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()

# 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())
```

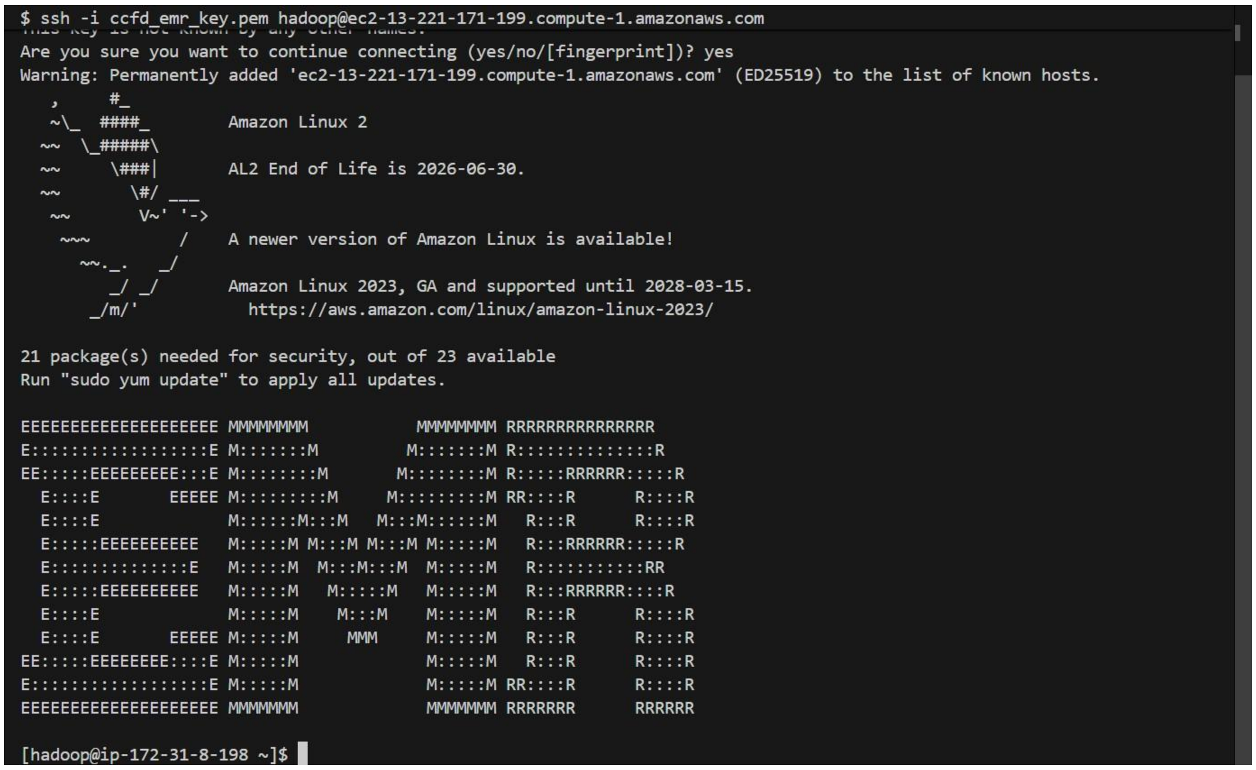


```
# Casting raw data as string and aliasing
credit_data = credit_data.selectExpr("cast(value as string)") \
    .withColumn("value", regexp_replace("value", r'\\"', '"')) \
    .withColumn("value", expr("substring(value, 2, length(value)-2)")) \
    .withColumn("value", regexp_replace("value", r'\\n$', ''))

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']))
```

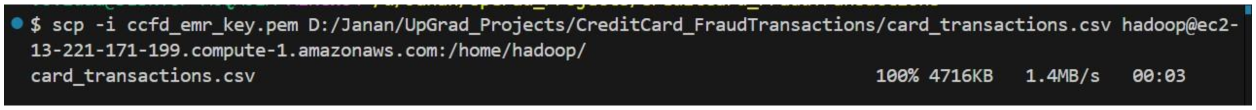


```
# Write output to console as well
output_data = Final_data \
    .select("card_id", "member_id", "amount", "pos_id", "postcode", "transaction_dt") \
    .writeStream \
    .outputMode("append") \
    .format("console") \
    .option("truncate", False) \
    .start()

#indicating Spark to await termination
output_data.awaitTermination()
```

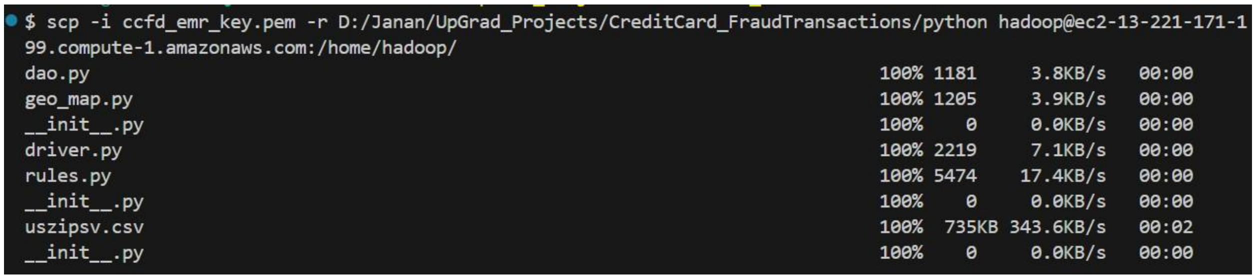
a. Importing dependencies and setting Kafka consumer.
b. Connecting to Kafka use the following details:
Bootstrap-server: **18.211.252.152**
Port Number: **9092**
Topic: **transactions-topic-verified**
c. Reading Input from Kafka



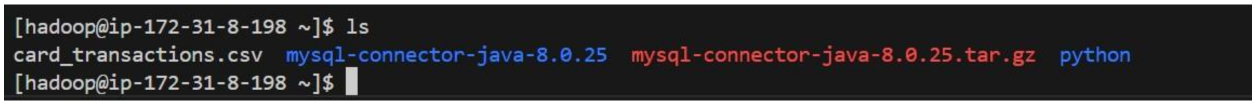
```
$ ssh -i ccfd_emr_key.pem hadoop@ec2-13-221-171-199.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-221-171-199.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

21 package(s) needed for security, out of 23 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-8-198 ~]$
```



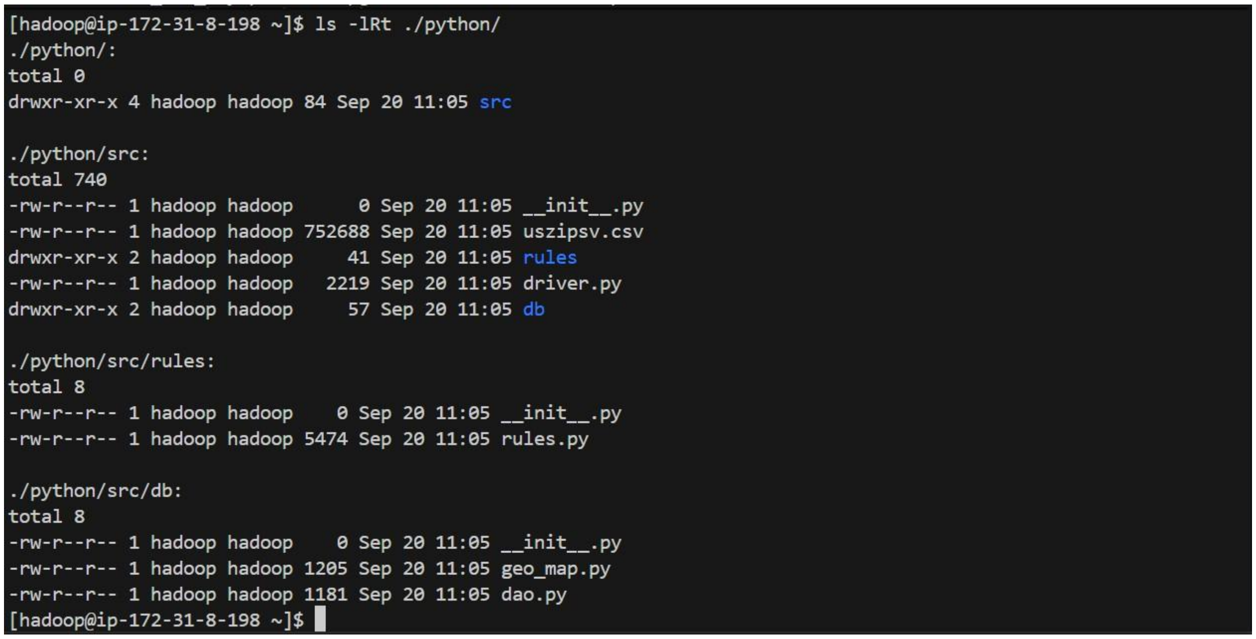
```
$ scp -i ccfd_emr_key.pem D:/Janan/UpGrad_Projects/CreditCard_FraudTransactions/card_transactions.csv hadoop@ec2-
13-221-171-199.compute-1.amazonaws.com:/home/hadoop/
card_transactions.csv                                                 100% 4716KB   1.4MB/s   00:03
```



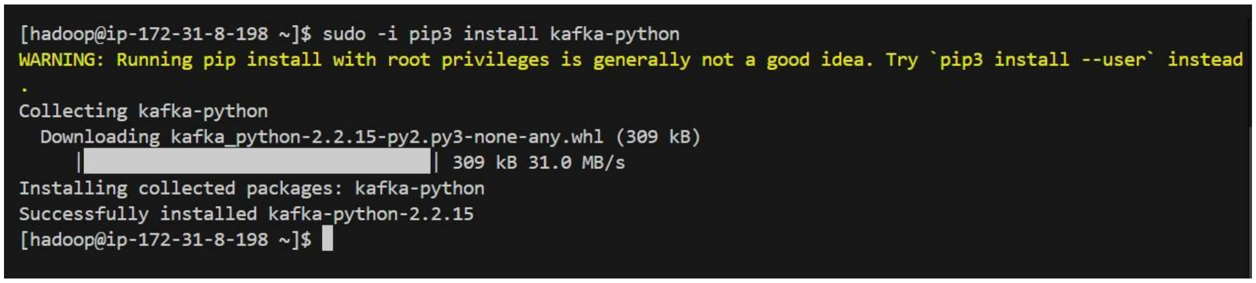
```
$ scp -i ccfd_emr_key.pem -r D:/Janan/UpGrad_Projects/CreditCard_FraudTransactions/python hadoop@ec2-13-221-171-1
99.compute-1.amazonaws.com:/home/hadoop/
dao.py                                                                100% 1181     3.8KB/s   00:00
geo_map.py                                                            100% 1205     3.9KB/s   00:00
__init__.py                                                           100%    0     0.0KB/s   00:00
driver.py                                                             100% 2219     7.1KB/s   00:00
rules.py                                                              100% 5474    17.4KB/s   00:00
__init__.py                                                           100%    0     0.0KB/s   00:00
uszipsv.csv                                                           100%  735KB 343.6KB/s   00:02
__init__.py                                                           100%    0     0.0KB/s   00:00
```



```
[hadoop@ip-172-31-8-198 ~]$ ls
card_transactions.csv  mysql-connector-java-8.0.25  mysql-connector-java-8.0.25.tar.gz  python
[hadoop@ip-172-31-8-198 ~]$
```

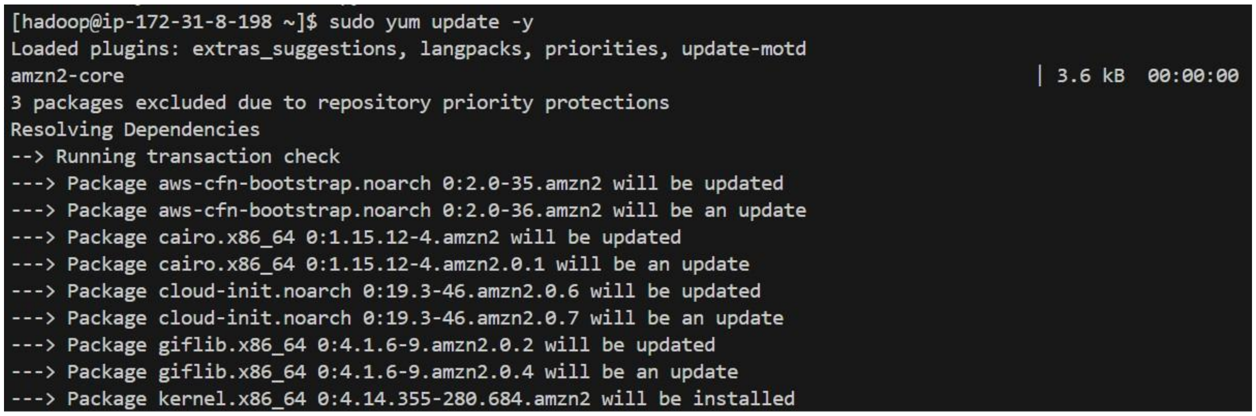


```
[hadoop@ip-172-31-8-198 ~]$ ls -lRt ./python/
./python/:
total 0
drwxr-xr-x 4 hadoop hadoop 84 Sep 20 11:05 src

./python/src:
total 740
-rw-r--r-- 1 hadoop hadoop      0 Sep 20 11:05 __init__.py
-rw-r--r-- 1 hadoop hadoop 752688 Sep 20 11:05 uszipsv.csv
drwxr-xr-x 2 hadoop hadoop     41 Sep 20 11:05 rules
-rw-r--r-- 1 hadoop hadoop   2219 Sep 20 11:05 driver.py
drwxr-xr-x 2 hadoop hadoop     57 Sep 20 11:05 db

./python/src/rules:
total 8
-rw-r--r-- 1 hadoop hadoop    0 Sep 20 11:05 __init__.py
-rw-r--r-- 1 hadoop hadoop 5474 Sep 20 11:05 rules.py

./python/src/db:
total 8
-rw-r--r-- 1 hadoop hadoop    0 Sep 20 11:05 __init__.py
-rw-r--r-- 1 hadoop hadoop 1205 Sep 20 11:05 geo_map.py
-rw-r--r-- 1 hadoop hadoop 1181 Sep 20 11:05 dao.py
[hadoop@ip-172-31-8-198 ~]$
```



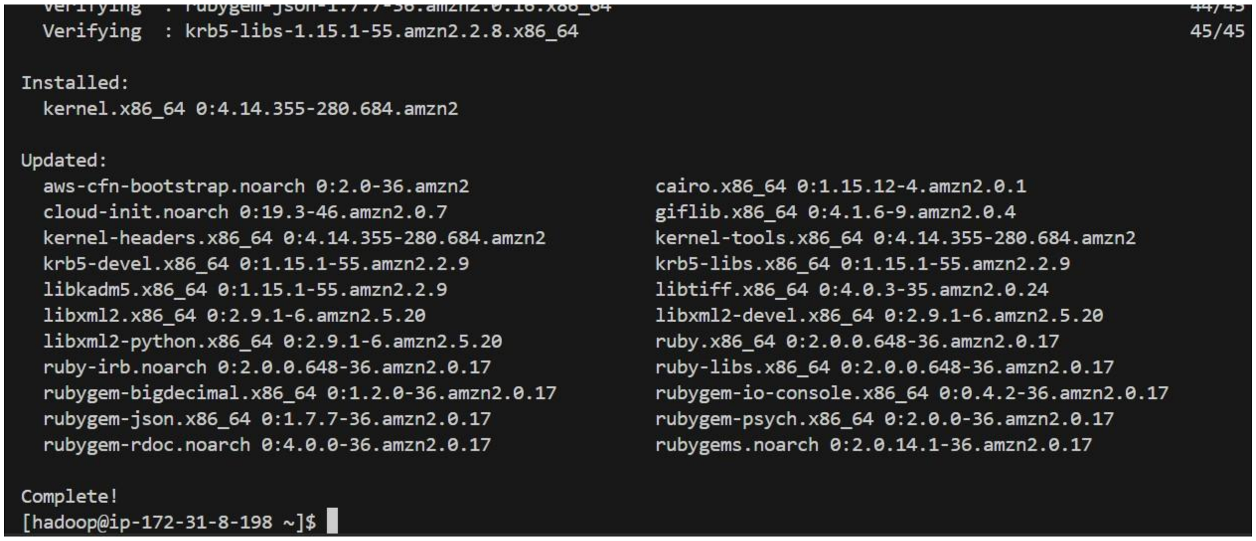
```
[hadoop@ip-172-31-8-198 ~]$ sudo -i pip3 install kafka-python
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip3 install --user` instead
.
Collecting kafka-python
  Downloading kafka_python-2.2.15-py2.py3-none-any.whl (309 kB)
     |████████████████████████████████| 309 kB 31.0 MB/s
Installing collected packages: kafka-python
Successfully installed kafka-python-2.2.15
[hadoop@ip-172-31-8-198 ~]$
```

11. Installing Kafka Python using Root privileges Sudo -i pip install kafka-python

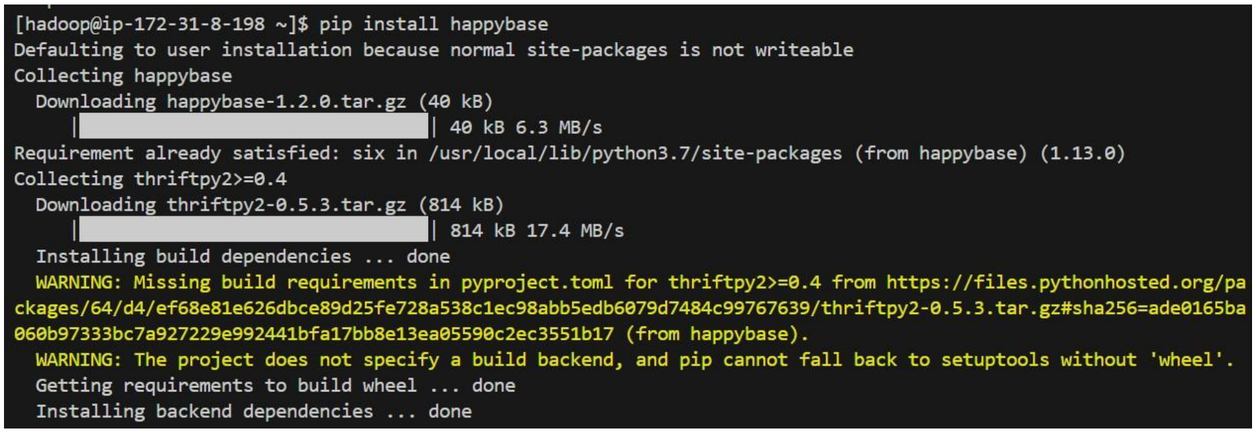


```
[hadoop@ip-172-31-8-198 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                                    | 3.6 kB  00:00:00
3 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
---> Package aws-cfn-bootstrap.noarch 0:2.0-35.amzn2 will be updated
---> Package aws-cfn-bootstrap.noarch 0:2.0-36.amzn2 will be an update
---> Package cairo.x86_64 0:1.15.12-4.amzn2 will be updated
---> Package cairo.x86_64 0:1.15.12-4.amzn2.0.1 will be an update
---> Package cloud-init.noarch 0:19.3-46.amzn2.0.6 will be updated
---> Package cloud-init.noarch 0:19.3-46.amzn2.0.7 will be an update
---> Package giflib.x86_64 0:4.1.6-9.amzn2.0.2 will be updated
---> Package giflib.x86_64 0:4.1.6-9.amzn2.0.4 will be an update
---> Package kernel.x86_64 0:4.14.355-280.684.amzn2 will be installed
```

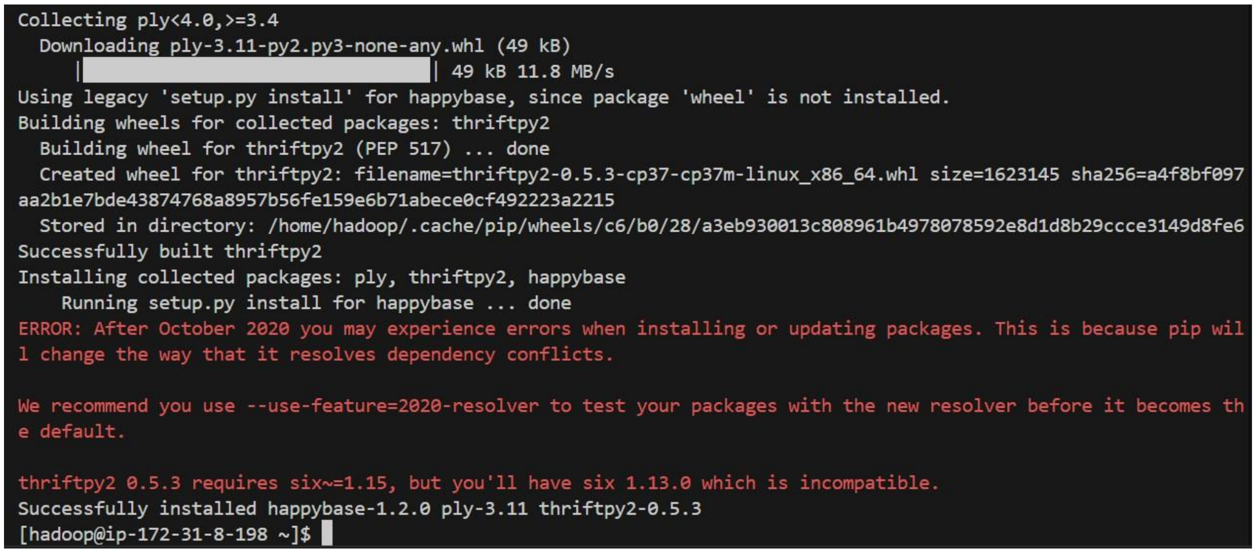
12. Updating all the Linux packages sudo yum update -y



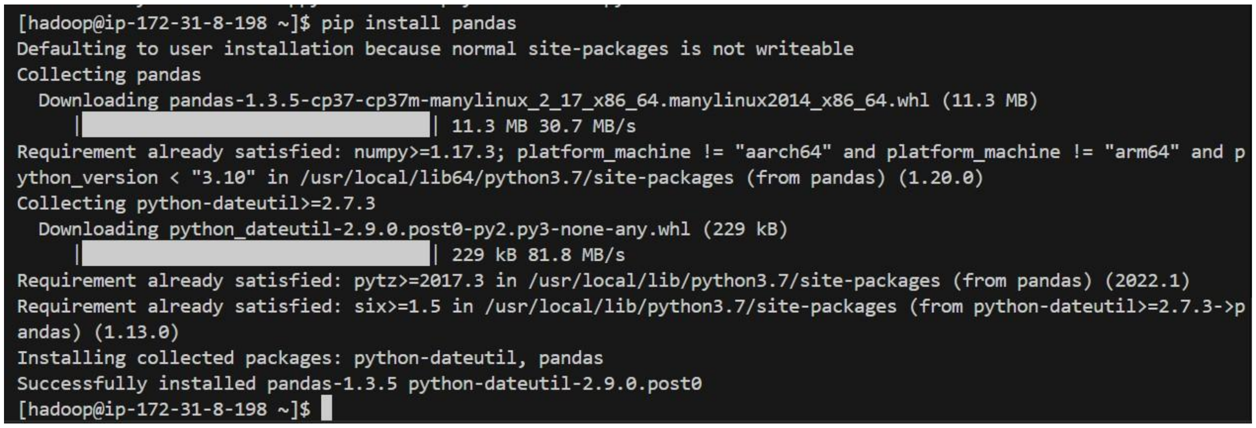
```
  Verifying  : rubygem-json-1.7.7-36.amzn2.0.16.x86_64                                          44/45
  Verifying  : krb5-libs-1.15.1-55.amzn2.2.8.x86_64                                             45/45

Installed:
  kernel.x86_64 0:4.14.355-280.684.amzn2

Updated:
  aws-cfn-bootstrap.noarch 0:2.0-36.amzn2                cairo.x86_64 0:1.15.12-4.amzn2.0.1
  cloud-init.noarch 0:19.3-46.amzn2.0.7                  giflib.x86_64 0:4.1.6-9.amzn2.0.4
  kernel-headers.x86_64 0:4.14.355-280.684.amzn2         kernel-tools.x86_64 0:4.14.355-280.684.amzn2
  krb5-devel.x86_64 0:1.15.1-55.amzn2.2.9                krb5-libs.x86_64 0:1.15.1-55.amzn2.2.9
  libkadm5.x86_64 0:1.15.1-55.amzn2.2.9                  libtiff.x86_64 0:4.0.3-35.amzn2.0.24
  libxml2.x86_64 0:2.9.1-6.amzn2.5.20                    libxml2-devel.x86_64 0:2.9.1-6.amzn2.5.20
  libxml2-python.x86_64 0:2.9.1-6.amzn2.5.20             ruby.x86_64 0:2.0.0.648-36.amzn2.0.17
  ruby-irb.noarch 0:2.0.0.648-36.amzn2.0.17              ruby-libs.x86_64 0:2.0.0.648-36.amzn2.0.17
  rubygem-bigdecimal.x86_64 0:1.2.0-36.amzn2.0.17        rubygem-io-console.x86_64 0:0.4.2-36.amzn2.0.17
  rubygem-json.x86_64 0:1.7.7-36.amzn2.0.17              rubygem-psych.x86_64 0:2.0.0-36.amzn2.0.17
  rubygem-rdoc.noarch 0:4.0.0-36.amzn2.0.17              rubygems.noarch 0:2.0.14.1-36.amzn2.0.17

Complete!
[hadoop@ip-172-31-8-198 ~]$ 
```



```
[hadoop@ip-172-31-8-198 ~]$ pip install happybase
Defaulting to user installation because normal site-packages is not writeable
Collecting happybase
  Downloading happybase-1.2.0.tar.gz (40 kB)
     |████████████████████████████████| 40 kB 6.3 MB/s
Requirement already satisfied: six in /usr/local/lib/python3.7/site-packages (from happybase) (1.13.0)
Collecting thriftpy2>=0.4
  Downloading thriftpy2-0.5.3.tar.gz (814 kB)
     |████████████████████████████████| 814 kB 17.4 MB/s
  Installing build dependencies ... done
  WARNING: Missing build requirements in pyproject.toml for thriftpy2>=0.4 from https://files.pythonhosted.org/pa
ckages/64/d4/ef68e81e626dbce89d25fe728a538c1ec98abb5edb6079d7484c99767639/thriftpy2-0.5.3.tar.gz#sha256=ade0165ba
060b97333bc7a927229e992441bfa17bb8e13ea05590c2ec3551b17 (from happybase).
  WARNING: The project does not specify a build backend, and pip cannot fall back to setuptools without 'wheel'.
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
```



```
Collecting ply<4.0,>=3.4
  Downloading ply-3.11-py2.py3-none-any.whl (49 kB)
     |████████████████████████████████| 49 kB 11.8 MB/s
Using legacy 'setup.py install' for happybase, since package 'wheel' is not installed.
Building wheels for collected packages: thriftpy2
  Building wheel for thriftpy2 (PEP 517) ... done
  Created wheel for thriftpy2: filename=thriftpy2-0.5.3-cp37-cp37m-linux_x86_64.whl size=1623145 sha256=a4f8bf097
aa2b1e7bde43874768a8957b56fe159e6b71abece0cf492223a2215
  Stored in directory: /home/hadoop/.cache/pip/wheels/c6/b0/28/a3eb930013c808961b4978078592e8d1d8b29ccce3149d8fe6
Successfully built thriftpy2
Installing collected packages: ply, thriftpy2, happybase
    Running setup.py install for happybase ... done
ERROR: After October 2020 you may experience errors when installing or updating packages. This is because pip wil
l change the way that it resolves dependency conflicts.

We recommend you use --use-feature=2020-resolver to test your packages with the new resolver before it becomes th
e default.

thriftpy2 0.5.3 requires six~=1.15, but you'll have six 1.13.0 which is incompatible.
Successfully installed happybase-1.2.0 ply-3.11 thriftpy2-0.5.3
[hadoop@ip-172-31-8-198 ~]$ 
```

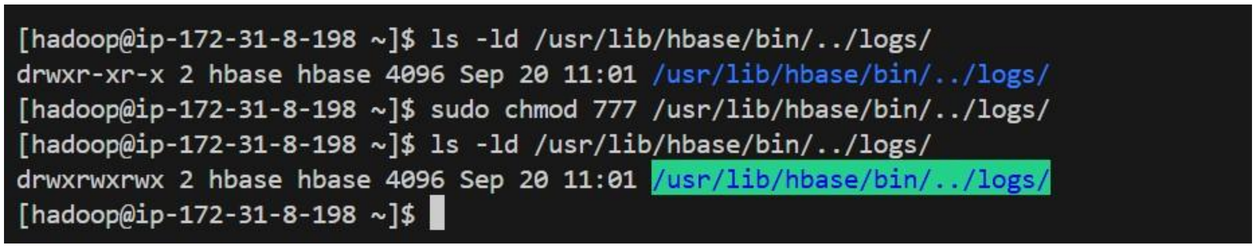


```
[hadoop@ip-172-31-8-198 ~]$ pip install pandas
Defaulting to user installation because normal site-packages is not writeable
Collecting pandas
  Downloading pandas-1.3.5-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.3 MB)
     |████████████████████████████████| 11.3 MB 30.7 MB/s
Requirement already satisfied: numpy>=1.17.3; platform_machine != "aarch64" and platform_machine != "arm64" and p
ython_version < "3.10" in /usr/local/lib64/python3.7/site-packages (from pandas) (1.20.0)
Collecting python-dateutil>=2.7.3
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
     |████████████████████████████████| 229 kB 81.8 MB/s
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/site-packages (from pandas) (2022.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/site-packages (from python-dateutil>=2.7.3->p
andas) (1.13.0)
Installing collected packages: python-dateutil, pandas
Successfully installed pandas-1.3.5 python-dateutil-2.9.0.post0
[hadoop@ip-172-31-8-198 ~]$
```

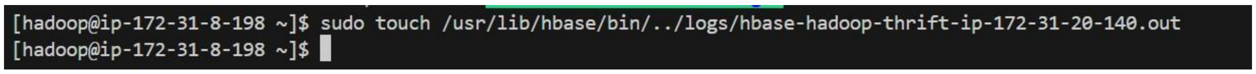
13. Installing Happybase & Pandas sudo yum install python3-devel -y pip install happybase pip install pandas

14. Making sure the permissions to directory for thrift services and starting the thrift service.

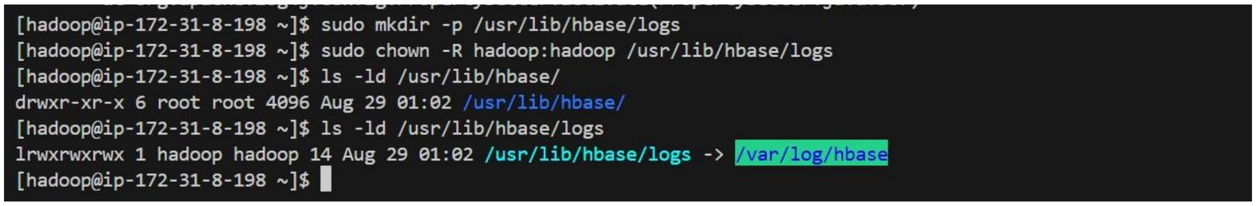
```
ls -ld /usr/lib/hbase/bin/../logs/
sudo chmod 777 /usr/lib/hbase/bin/../logs/
sudo touch /usr/lib/hbase/bin/../logs/hbase-hadoop-thrift-ip-172-31-20-140.out
/usr/lib/hbase/bin/hbase-daemon.sh start thrift -p 9090
```



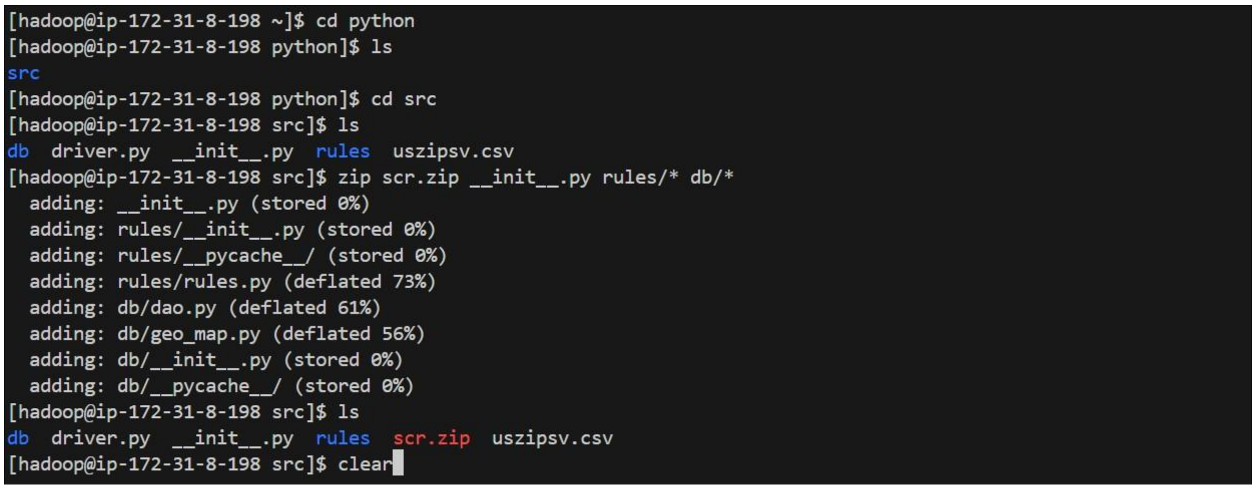
```
[hadoop@ip-172-31-8-198 ~]$ ls -ld /usr/lib/hbase/bin/../logs/
drwxr-xr-x 2 hbase hbase 4096 Sep 20 11:01 /usr/lib/hbase/bin/../logs/
[hadoop@ip-172-31-8-198 ~]$ sudo chmod 777 /usr/lib/hbase/bin/../logs/
[hadoop@ip-172-31-8-198 ~]$ ls -ld /usr/lib/hbase/bin/../logs/
drwxrwxrwx 2 hbase hbase 4096 Sep 20 11:01 /usr/lib/hbase/bin/../logs/
[hadoop@ip-172-31-8-198 ~]$
```



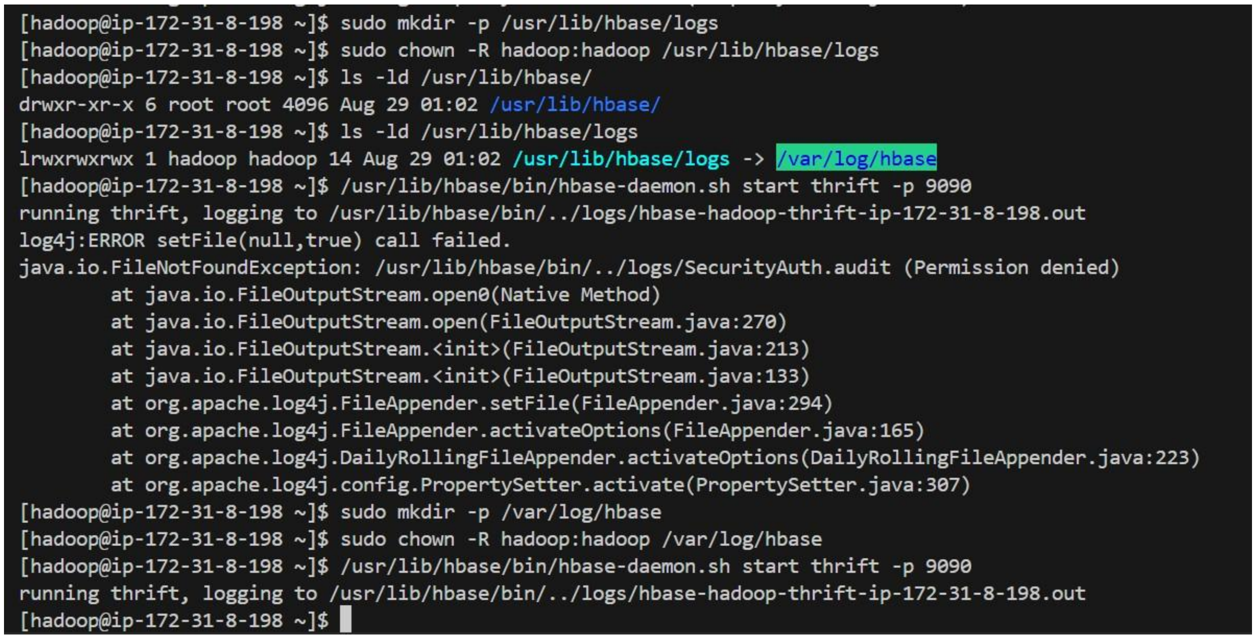
```
[hadoop@ip-172-31-8-198 ~]$ sudo touch /usr/lib/hbase/bin/../logs/hbase-hadoop-thrift-ip-172-31-20-140.out
[hadoop@ip-172-31-8-198 ~]$
```



```
[hadoop@ip-172-31-8-198 ~]$ sudo mkdir -p /usr/lib/hbase/logs
[hadoop@ip-172-31-8-198 ~]$ sudo chown -R hadoop:hadoop /usr/lib/hbase/logs
[hadoop@ip-172-31-8-198 ~]$ ls -ld /usr/lib/hbase/
drwxr-xr-x 6 root root 4096 Aug 29 01:02 /usr/lib/hbase/
[hadoop@ip-172-31-8-198 ~]$ ls -ld /usr/lib/hbase/logs
lrwxrwxrwx 1 hadoop hadoop 14 Aug 29 01:02 /usr/lib/hbase/logs -> /var/log/hbase
[hadoop@ip-172-31-8-198 ~]$
```

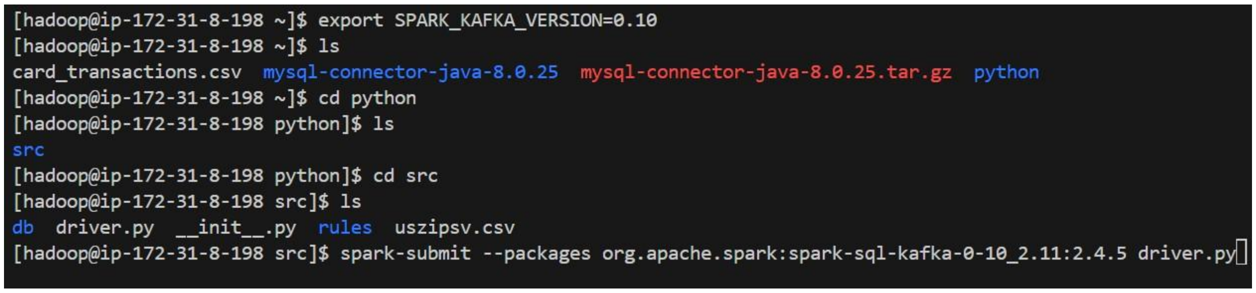


```
[hadoop@ip-172-31-8-198 ~]$ cd python
[hadoop@ip-172-31-8-198 python]$ ls
src
[hadoop@ip-172-31-8-198 python]$ cd src
[hadoop@ip-172-31-8-198 src]$ ls
db  driver.py  __init__.py  rules  uszipsv.csv
[hadoop@ip-172-31-8-198 src]$ zip scr.zip __init__.py rules/* db/*
  adding: __init__.py (stored 0%)
  adding: rules/__init__.py (stored 0%)
  adding: rules/__pycache__/ (stored 0%)
  adding: rules/rules.py (deflated 73%)
  adding: db/dao.py (deflated 61%)
  adding: db/geo_map.py (deflated 56%)
  adding: db/__init__.py (stored 0%)
  adding: db/__pycache__/ (stored 0%)
[hadoop@ip-172-31-8-198 src]$ ls
db  driver.py  __init__.py  rules  scr.zip  uszipsv.csv
[hadoop@ip-172-31-8-198 src]$ clear
```



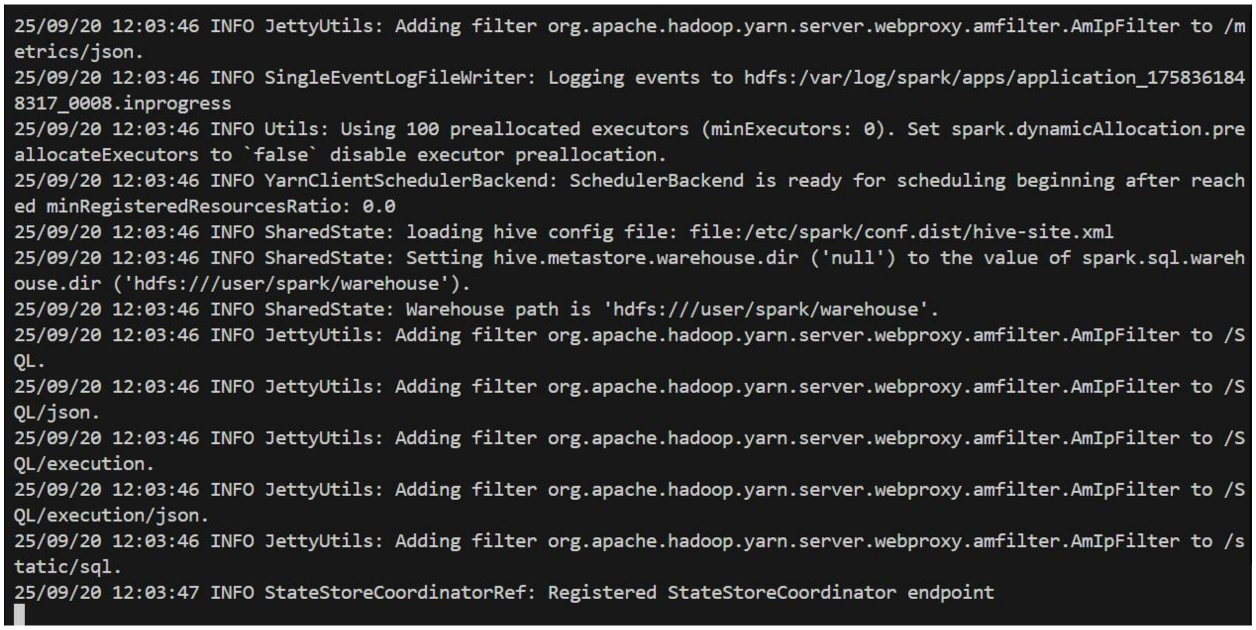
```
[hadoop@ip-172-31-8-198 ~]$ sudo mkdir -p /usr/lib/hbase/logs
[hadoop@ip-172-31-8-198 ~]$ sudo chown -R hadoop:hadoop /usr/lib/hbase/logs
[hadoop@ip-172-31-8-198 ~]$ ls -ld /usr/lib/hbase/
drwxr-xr-x 6 root root 4096 Aug 29 01:02 /usr/lib/hbase/
[hadoop@ip-172-31-8-198 ~]$ ls -ld /usr/lib/hbase/logs
lrwxrwxrwx 1 hadoop hadoop 14 Aug 29 01:02 /usr/lib/hbase/logs -> /var/log/hbase
[hadoop@ip-172-31-8-198 ~]$ /usr/lib/hbase/bin/hbase-daemon.sh start thrift -p 9090
running thrift, logging to /usr/lib/hbase/bin/../logs/hbase-hadoop-thrift-ip-172-31-8-198.out
log4j:ERROR setFile(null,true) call failed.
java.io.FileNotFoundException: /usr/lib/hbase/bin/../logs/SecurityAuth.audit (Permission denied)
        at java.io.FileOutputStream.open0(Native Method)
        at java.io.FileOutputStream.open(FileOutputStream.java:270)
        at java.io.FileOutputStream.<init>(FileOutputStream.java:213)
        at java.io.FileOutputStream.<init>(FileOutputStream.java:133)
        at org.apache.log4j.FileAppender.setFile(FileAppender.java:294)
        at org.apache.log4j.FileAppender.activateOptions(FileAppender.java:165)
        at org.apache.log4j.DailyRollingFileAppender.activateOptions(DailyRollingFileAppender.java:223)
        at org.apache.log4j.config.PropertySetter.activate(PropertySetter.java:307)
[hadoop@ip-172-31-8-198 ~]$ sudo mkdir -p /var/log/hbase
[hadoop@ip-172-31-8-198 ~]$ sudo chown -R hadoop:hadoop /var/log/hbase
[hadoop@ip-172-31-8-198 ~]$ /usr/lib/hbase/bin/hbase-daemon.sh start thrift -p 9090
running thrift, logging to /usr/lib/hbase/bin/../logs/hbase-hadoop-thrift-ip-172-31-8-198.out
[hadoop@ip-172-31-8-198 ~]$
```

19. Setting Kafka version and running spark submit command. export
SPARK_KAFKA_VERSION=0.10 spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 driver.py



```
[hadoop@ip-172-31-8-198 ~]$ export SPARK_KAFKA_VERSION=0.10
[hadoop@ip-172-31-8-198 ~]$ ls
card_transactions.csv  mysql-connector-java-8.0.25  mysql-connector-java-8.0.25.tar.gz  python
[hadoop@ip-172-31-8-198 ~]$ cd python
[hadoop@ip-172-31-8-198 python]$ ls
src
[hadoop@ip-172-31-8-198 python]$ cd src
[hadoop@ip-172-31-8-198 src]$ ls
db  driver.py  __init__.py  rules  uszipsv.csv
[hadoop@ip-172-31-8-198 src]$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 driver.py
```

20. Count Data in Hbase: count 'lookup_data_hive’



```
25/09/20 12:03:46 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /m
etrics/json.
25/09/20 12:03:46 INFO SingleEventLogFileWriter: Logging events to hdfs:/var/log/spark/apps/application_175836184
8317_0008.inprogress
25/09/20 12:03:46 INFO Utils: Using 100 preallocated executors (minExecutors: 0). Set spark.dynamicAllocation.pre
allocateExecutors to `false` disable executor preallocation.
25/09/20 12:03:46 INFO YarnClientSchedulerBackend: SchedulerBackend is ready for scheduling beginning after reach
ed minRegisteredResourcesRatio: 0.0
25/09/20 12:03:46 INFO SharedState: loading hive config file: file:/etc/spark/conf.dist/hive-site.xml
25/09/20 12:03:46 INFO SharedState: Setting hive.metastore.warehouse.dir ('null') to the value of spark.sql.wareh
ouse.dir ('hdfs:///user/spark/warehouse').
25/09/20 12:03:46 INFO SharedState: Warehouse path is 'hdfs:///user/spark/warehouse'.
25/09/20 12:03:46 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /S
QL.
25/09/20 12:03:46 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /S
QL/json.
25/09/20 12:03:46 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /S
QL/execution.
25/09/20 12:03:46 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /S
QL/execution/json.
25/09/20 12:03:46 INFO JettyUtils: Adding filter org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter to /s
tatic/sql.
25/09/20 12:03:47 INFO StateStoreCoordinatorRef: Registered StateStoreCoordinator endpoint
```

Please Note:

1. Due to an issue with the Kafka server, I was unable to retrieve any data. However, once Batch 0 outputs data to the console, we should be able to verify the number of entries.
2. We also attempted to use the static dataset provided by UpGrad in text format. To bypass Kafka, we tried importing it directly into Spark Streaming from local storage. Despite multiple attempts, Spark could not locate or access the file. We verified read/write permissions, but the error persisted indicating that the file does not exist.
   (The corresponding code was later replaced with Kafka Consumer details.)

```
# Reading input from a text file

credit_data = spark.readStream \

        .option("inferSchema", "true") \

        .text("/home/hadoop/python/src/transactions-topic-verified.txt")
```