Code Logic - Retail Data Analysis

Logic for Python Script 'spark-streaming.py'

Initial we are importing necessary Spark libraries to work with

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *
```

Now, we have built some UDFs to calculate different items.

- Total Cost UDF - To calculate the total income from every invoice we needed to calculate the income from sale of each product, so we multiplied the unit price of the product with the quantity of the product purchased. The sum of this cost across the products in that invoice gives us the total cost of the order. We also made sure that if the transaction is a return transaction, the total cost is negative.

```
def find_total_order_cost(items, trn_type):
    if items is not None:
        total_cost = 0
        item_price = 0
        for item in items:
            item_price = (item['quantity'] * item['unit_price'])
            total_cost = total_cost + item_price
            item_price = 0

        if trn_type == "RETURN":
            return total_cost * -1
        else:
            return total_cost
```

- Total Items UDF - To calculate the number of products in every invoice we added the quantity ordered of each product in that invoice

```
def find_total_item_count(items):
    if items is not None:
        total_count = 0
        for item in items:
            total_count = total_count + item['quantity']
        return total_count
```

- Is Order UDF - To determine if invoice is for an order or not, we used an if-else statement

```
def flag_isOrder(trn_type):
    if trn_type == "ORDER":
        return(1)
    else:
        return(0)
```

- Is Return UDF - To determine if invoice is for a return or not, we used an if-else statement

```
def flag_isReturn(trn_type):
    if trn_type == "RETURN":
        return(1)
    else:
        return(0)
```

Initializing the Spark session and setting the log level to error as a good practice

```
spark = SparkSession  \
    .builder  \
    .appName("spark-streaming")  \
    .getOrCreate()
spark.sparkContext.setLogLevel('ERROR')
```

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

```
orderRawData = spark.readStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers", "18.211.252.152:9092") \
    .option("startingOffsets","earliest") \
    .option("failOnDataLoss", "false") \
    .option("subscribe", "real-time-project") \
    .load()
```

Defining JSON schema of each order, using appropriate datatypes and StrucField in the case of the item attributes

```
jsonSchema = StructType() \
    .add("invoice_no", LongType()) \
    .add("country", StringType()) \
    .add("timestamp", TimestampType()) \
    .add("type", StringType()) \
    .add("items", ArrayType(StructType([
    StructField("SKU", StringType()),
    StructField("title", StringType()),
    StructField("unit_price", FloatType()),
    StructField("quantity", IntegerType()),
])))
```

Reading the raw JSON data from Kafka as ‘order stream’ by casting it to string and storing it into the alias ‘data’

```
orderStream = orderRawData.select(from_json(col("value").cast("string"),
jsonSchema).alias("data")).select("data.*")
```

Defining the UDFs by Converting the Python functions we defined earlier, and assigning the appropriate return datatype

```
sum_total_order_cost = udf(find_total_order_cost, FloatType())
sum_total_item_count = udf(find_total_item_count, IntegerType())
sum_isOrder = udf(flag_isOrder, IntegerType())
sum_isReturn = udf (flag_isReturn, IntegerType())
```

Calculating the additional columns according to the required input values

```
expandedOrderStream = orderStream \
    .withColumn("total_cost", sum_total_order_cost(orderStream.items,
orderStream.type)) \
    .withColumn("total_items", sum_total_item_count(orderStream.items)) \
    .withColumn("is_order", sum_isOrder(orderStream.type)) \
    .withColumn("is_return", sum_isReturn(orderStream.type))
```

Writing the summarised input values to console, using 'append' output method and applying truncate as false and setting the processing time to 1 minute

```
extendedOrderQuery = expandedOrderStream \
    .select("invoice_no", "country", "timestamp", "total_cost",
"total_items", "is_order", "is_return") \
    .writeStream \
    .outputMode("append") \
    .format("console") \
    .option("truncate", "false") \
    .trigger(processingTime = "1 minute") \
    .start()
```

Calculating time-based KPIs (Total sale volume, OPM, Rate of return, Average transaction size) having tumbling window of one minute and watermark of one minute.

```
aggStreamByTime = expandedOrderStream \
    .withWatermark("timestamp","1 minute") \
    .groupBy(window("timestamp", "1 minute", "1 minute")) \
    .agg(sum("total_cost").alias("total_sale_volume"),
         count("invoice_no").alias("OPM"),
         avg("is_return").alias("rate_of_return"),
         avg("total_cost").alias("average_transaction_size")
        ) \
    .select("window", "OPM", "total_sale_volume", "average_transaction_size",
"rate_of_return" )
```

Writing the time-based KPIs data to HDFS - HDFS into JSON files for each one-minute window, using 'append' output mode, setting truncate as false, and specifying the HDFS output path for both the KPI files and for their checkpoints. We have taken sixteen batches at the interval of 1 minute.

```
queryByTime = aggStreamByTime.writeStream \
    .format("json") \
    .outputMode("append") \
    .option("truncate","false") \
    .option("path","/user/ec2-user/time_kpi") \
    .option("checkpointLocation","/user/ec2-user/time_kpi_checkpoints") \
    .trigger(processingTime="1 minute") \
    .start()
```

Calculating time-and-country-based KPIs (Total sale volume, OPM, Rate of return) having tumbling window of one minute and watermark of one minute. Here we grouped by window and country both.

```
aggStreamByCountry = expandedOrderStream \
    .withWatermark("timestamp", "1 minute") \
    .groupBy(window("timestamp", "1 minute", "1 minute"), "country") \
```

```
    .agg(sum("total_cost").alias("total_sale_volume"),
         count("invoice_no").alias("OPM"),
         avg("is_return").alias("rate_of_return")) \
    .select("window", "country", "OPM", "total_sale_volume", "rate_of_return"
)
```

Writing the the time-and-country-based KPIs data to HDFS into JSON files for each one-minute window, using 'append' output mode, setting truncate as false, and specifying the HDFS output path for both the KPI files and for their checkpoints.

```
queryByCountry = aggStreamByCountry.writeStream \
    .format("json") \
    .outputMode("append") \
    .option("truncate","false") \
    .option("path","/user/ec2-user/country_kpi") \
    .option("checkpointLocation","/user/ec2-user/country_kpi_checkpoints") \
    .trigger(processingTime="1 minute") \
    .start()
```

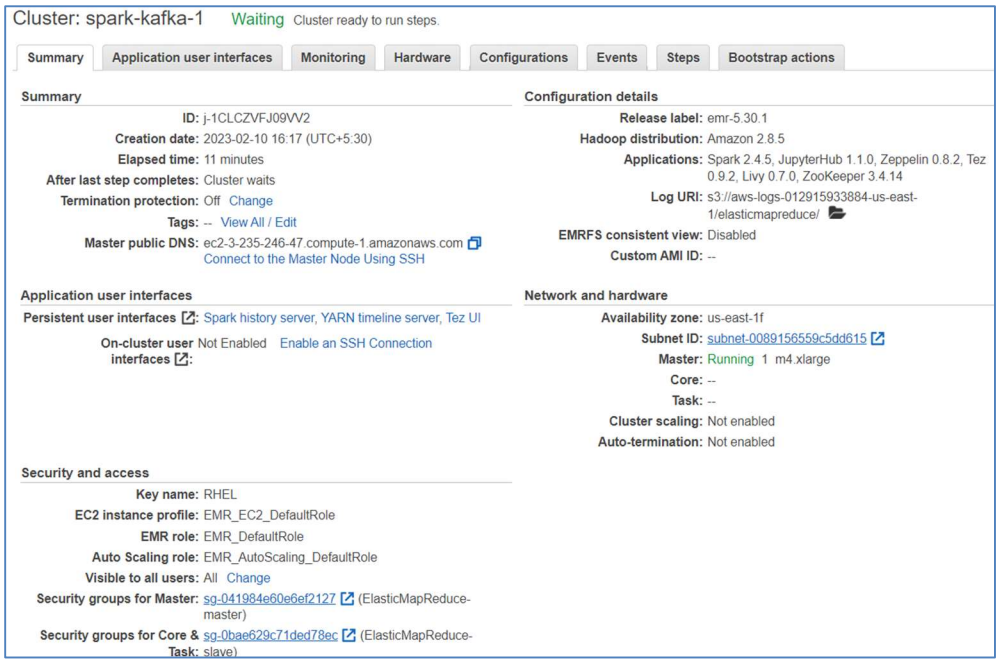
Indicating Spark to await termination

```
extendedOrderQuery.awaitTermination()
queryByCountry.awaitTermination()
queryByTime.awaitTermination()
```

So, above is the total Code logic which we have built, which is available in 'spark-streaming.py'. And same we will use in console to run through Spark Submit command.

EMR Cluster Creation:

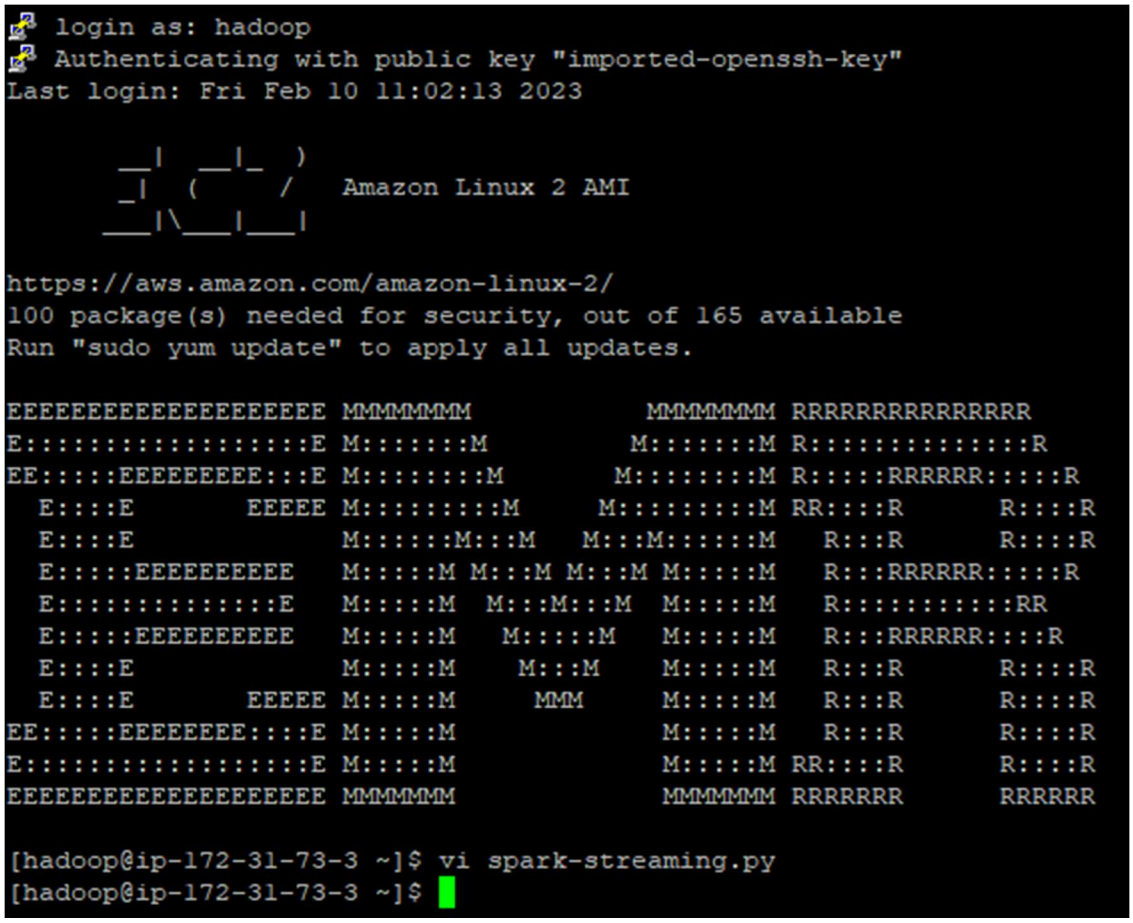
We have created EMR cluster in AWS with required application and same we will use for further activity. Here is the cluster snapshot for reference.



Console Commands:

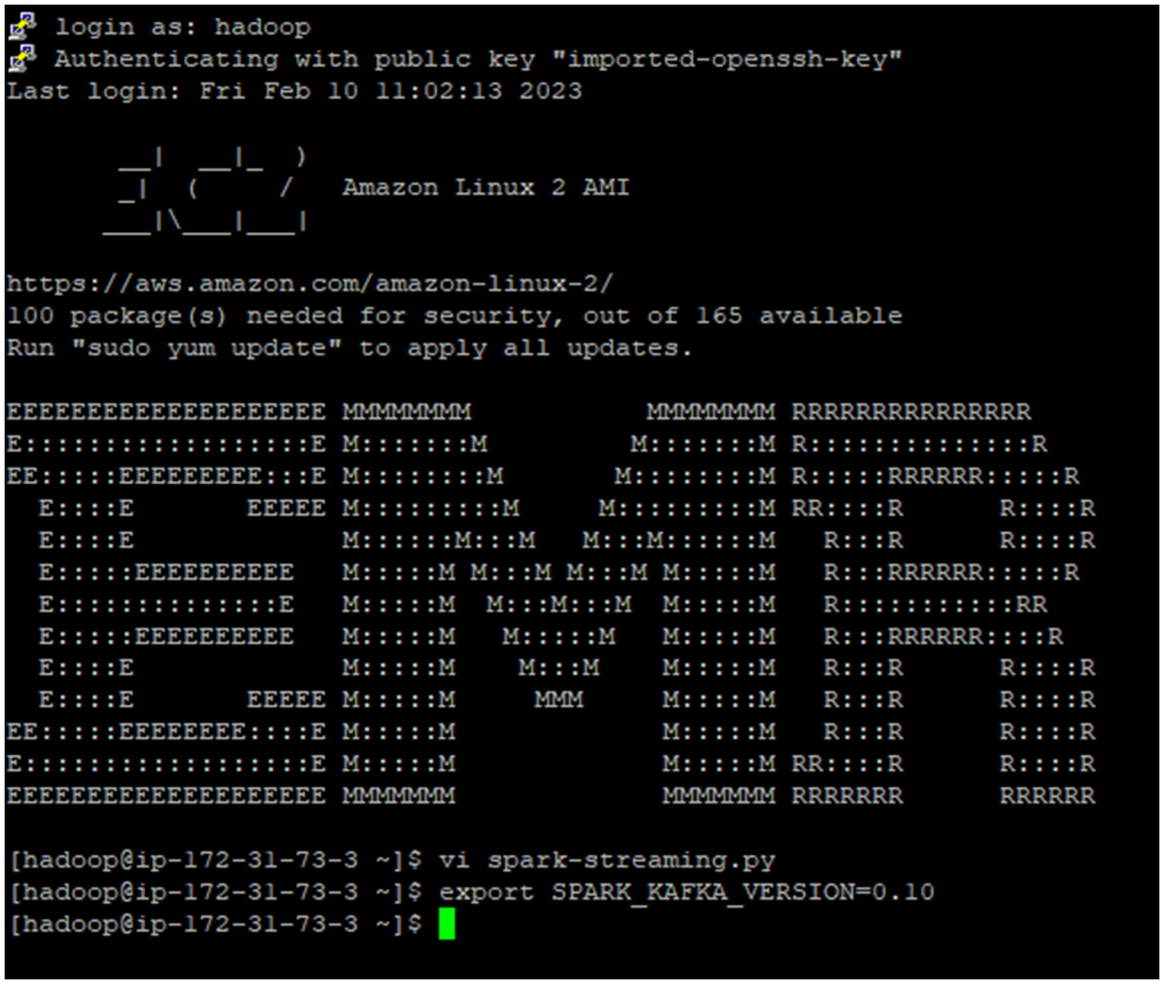
We started by logging into EMR cluster with the help of Putty.
And we created ‘spark-streaming.py’ through below command with code which we discussed above.

vi spark-streaming.py



Next, we set the Kafka Version using the following command

export SPARK_KAFKA_VERSION=0.10



And now, we ran the below Spark Submit command with details of python file created above and Spark-Kafka details.

spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 spark-streaming.py

```
[hadoop@ip-172-31-73-3 ~]$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 spark-streaming.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-2ea7ba82-649e-4da7-9d51-7810bc8828fb;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
downloading https://repo1.maven.org/maven2/org/apache/spark/spark-sql-kafka-0-10_2.11/2.4.5/spark-sql-kafka-0-10_2.11-2.4.5.jar ...
        [SUCCESSFUL ] org.apache.spark#spark-sql-kafka-0-10_2.11;2.4.5!spark-sql-kafka-0-10_2.11.jar (21ms)
downloading https://repo1.maven.org/maven2/org/apache/kafka/kafka-clients/2.0.0/kafka-clients-2.0.0.jar ...
        [SUCCESSFUL ] org.apache.kafka#kafka-clients;2.0.0!kafka-clients.jar (71ms)
downloading https://repo1.maven.org/maven2/org/spark-project/spark/unused/1.0.0/unused-1.0.0.jar ...
        [SUCCESSFUL ] org.spark-project.spark#unused;1.0.0!unused.jar (3ms)
downloading https://repo1.maven.org/maven2/org/lz4/lz4-java/1.4.0/lz4-java-1.4.0.jar ...
        [SUCCESSFUL ] org.lz4#lz4-java;1.4.0!lz4-java.jar (13ms)
downloading https://repo1.maven.org/maven2/org/xerial/snappy/snappy-java/1.1.7.3/snappy-java-1.1.7.3.jar ...
        [SUCCESSFUL ] org.xerial.snappy#snappy-java;1.1.7.3!snappy-java.jar(bundle) (65ms)
downloading https://repo1.maven.org/maven2/org/slf4j/slf4j-api/1.7.16/slf4j-api-1.7.16.jar ...
        [SUCCESSFUL ] org.slf4j#slf4j-api;1.7.16!slf4j-api.jar (5ms)
:: resolution report :: resolve 1538ms :: artifacts dl 184ms
        :: 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   |   6   |   6   |   0   ||   6   |   6   |
        ---------------------------------------------------------------------
:: retrieving :: org.apache.spark#spark-submit-parent-2ea7ba82-649e-4da7-9d51-7810bc8828fb
        confs: [default]
        6 artifacts copied, 0 already retrieved (4749kB/18ms)
```

And then we can see that final summarized batch outputs

```
-------------------------------------------
Batch: 0
-------------------------------------------
+--------------+--------------+-------------------+----------+-----------+--------+----------+
|invoice_no    |country       |timestamp          |total_cost|total_items|is_order|is_return|
+--------------+--------------+-------------------+----------+-----------+--------+----------+
|154132552816301|United Kingdom|2023-01-17 16:14:39|16.95     |1          |1       |0         |
|154132552816302|United Kingdom|2023-01-17 16:14:55|32.699997 |12         |1       |0         |
|154132552816303|United Kingdom|2023-01-17 16:15:01|17.7      |6          |1       |0         |
|154132552816304|United Kingdom|2023-01-17 16:15:05|252.59    |109        |1       |0         |
|154132552816305|United Kingdom|2023-01-17 16:15:15|55.38     |24         |1       |0         |
|154132552816306|United Kingdom|2023-01-17 16:15:17|72.47     |30         |1       |0         |
|154132552816307|United Kingdom|2023-01-17 16:15:23|34.45     |23         |1       |0         |
|154132552816308|United Kingdom|2023-01-17 16:15:26|6.98      |4          |1       |0         |
|154132552816309|France        |2023-01-17 16:15:33|8.639999  |14         |1       |0         |
|154132552816310|United Kingdom|2023-01-17 16:15:37|19.92     |2          |1       |0         |
|154132552816311|United Kingdom|2023-01-17 16:15:40|20.869999 |14         |1       |0         |
|154132552816312|United Kingdom|2023-01-17 16:15:47|76.57     |45         |1       |0         |
|154132552816313|United Kingdom|2023-01-17 16:15:50|14.58     |13         |1       |0         |
|154132552816314|United Kingdom|2023-01-17 16:15:56|35.4      |12         |1       |0         |
|154132552816315|United Kingdom|2023-01-17 16:16:08|10.79     |1          |1       |0         |
|154132552816316|United Kingdom|2023-01-17 16:16:10|459.55    |84         |1       |0         |
|154132552816317|United Kingdom|2023-01-17 16:16:11|1.25      |1          |1       |0         |
|154132552816318|United Kingdom|2023-01-17 16:16:13|187.47    |350        |1       |0         |
|154132552816319|United Kingdom|2023-01-17 16:16:34|37.35     |23         |1       |0         |
|154132552816320|United Kingdom|2023-01-17 16:16:35|70.8      |24         |1       |0         |
+--------------+--------------+-------------------+----------+-----------+--------+----------+
only showing top 20 rows
```

```
-------------------------------------------
Batch: 1
-------------------------------------------
+--------------+--------------+-------------------+----------+-----------+--------+----------+
|invoice_no    |country       |timestamp          |total_cost|total_items|is_order|is_return|
+--------------+--------------+-------------------+----------+-----------+--------+----------+
|154132553164538|United Kingdom|2023-02-10 11:20:32|3.9       |2          |1       |0         |
|154132553164539|United Kingdom|2023-02-10 11:20:33|10.2      |4          |1       |0         |
|154132553164540|United Kingdom|2023-02-10 11:20:37|23.55     |13         |1       |0         |
|154132553164541|United Kingdom|2023-02-10 11:20:38|71.11     |62         |1       |0         |
|154132553164542|United Kingdom|2023-02-10 11:20:42|237.59999 |48         |1       |0         |
|154132553164543|United Kingdom|2023-02-10 11:20:46|89.91     |32         |1       |0         |
|154132553164544|United Kingdom|2023-02-10 11:20:55|186.9     |15         |1       |0         |
|154132553164545|United Kingdom|2023-02-10 11:20:56|65.17     |57         |1       |0         |
|154132553164546|United Kingdom|2023-02-10 11:21:06|19.859999 |14         |1       |0         |
|154132553164547|United Kingdom|2023-02-10 11:21:07|18.95     |7          |1       |0         |
|154132553164548|United Kingdom|2023-02-10 11:21:17|12.5      |2          |1       |0         |
|154132553164549|United Kingdom|2023-02-10 11:21:17|20.76     |17         |1       |0         |
|154132553164550|United Kingdom|2023-02-10 11:21:18|76.52     |38         |1       |0         |
|154132553164551|United Kingdom|2023-02-10 11:21:21|32.27     |5          |1       |0         |
|154132553164552|United Kingdom|2023-02-10 11:21:21|39.8      |4          |1       |0         |
|154132553164553|United Kingdom|2023-02-10 11:21:30|1.66      |2          |1       |0         |
|154132553164554|United Kingdom|2023-02-10 11:21:38|10.150001 |7          |1       |0         |
|154132553164555|United Kingdom|2023-02-10 11:21:47|25.279999 |12         |1       |0         |
|154132553164556|United Kingdom|2023-02-10 11:21:54|31.54     |10         |1       |0         |
|154132553164557|United Kingdom|2023-02-10 11:22:20|1.5       |2          |1       |0         |
+--------------+--------------+-------------------+----------+-----------+--------+----------+
```

we kept it running the same till next 15-16 batches logged the session and terminated the session.

Now, we checked EC2 location to make sure that KPI files generated are saved in specified location.

hadoop fs -ls /user/ec2-user

```
[hadoop@ip-172-31-73-3 ~]$ hadoop fs -ls /user/ec2-user
Found 4 items
drwxr-xr-x   - hadoop hadoop          0 2023-02-10 11:24 /user/ec2-user/country_kpi
drwxr-xr-x   - hadoop hadoop          0 2023-02-10 11:22 /user/ec2-user/country_kpi_checkpoints
drwxr-xr-x   - hadoop hadoop          0 2023-02-10 11:24 /user/ec2-user/time_kpi
drwxr-xr-x   - hadoop hadoop          0 2023-02-10 11:22 /user/ec2-user/time_kpi_checkpoints
[hadoop@ip-172-31-73-3 ~]$
```

We also checked inside created folders to see created JSON files through below command for both time KPI and country KPI.

hadoop fs -ls /user/ec2-user/time_kpi/

```
[hadoop@ip-172-31-73-3 ~]$ hadoop fs -ls /user/ec2-user/time_kpi/
Found 207 items
drwxr-xr-x   - hadoop hadoop          0 2023-02-10 11:24 /user/ec2-user/time_kpi/_spark_metadata
-rw-r--r--   1 hadoop hadoop          0 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00000-061b59e9-219d-474b-9322-9d618e7d1dbc-c000.json
-rw-r--r--   1 hadoop hadoop          0 2023-02-10 11:24 /user/ec2-user/time_kpi/part-00000-136eb62f-ecc2-43d2-8e53-2c3ba637cd66-c000.json
-rw-r--r--   1 hadoop hadoop          0 2023-02-10 11:25 /user/ec2-user/time_kpi/part-00000-8bcb2f5f-309c-40ca-95da-61dc558922a7-c000.json
-rw-r--r--   1 hadoop hadoop          0 2023-02-10 11:22 /user/ec2-user/time_kpi/part-00000-99d04c3e-28d0-4d2e-8087-96b19d2bfb7d-c000.json
-rw-r--r--   1 hadoop hadoop      37465 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00000-c7d4f0a4-4ebd-4c96-9643-83c76491fec3-c000.json
-rw-r--r--   1 hadoop hadoop      36580 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00001-5a32018f-0ada-4583-bf57-07dfa9cd11cd-c000.json
-rw-r--r--   1 hadoop hadoop      34919 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00002-852690ec-4796-4e6a-84f6-9078ff15612d-c000.json
-rw-r--r--   1 hadoop hadoop      33745 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00003-184e83f0-0506-4654-aba8-1f38a1306245-c000.json
-rw-r--r--   1 hadoop hadoop      30155 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00004-d3f63c49-a668-48a1-a47e-915f73bda723-c000.json
-rw-r--r--   1 hadoop hadoop      36722 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00005-b15fa700-e26f-40d0-8e44-3b1667b6df5b-c000.json
-rw-r--r--   1 hadoop hadoop      30659 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00006-6b29cc05-5887-4e07-9e6e-9147849849de-c000.json
-rw-r--r--   1 hadoop hadoop      30375 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00007-39b322ef-ae28-4745-a451-980540138e9d-c000.json
-rw-r--r--   1 hadoop hadoop      38744 2023-02-10 11:22 /user/ec2-user/time_kpi/part-00008-23d790ce-aedd-44ae-b843-14c35b1adb37-c000.json
-rw-r--r--   1 hadoop hadoop      31724 2023-02-10 11:22 /user/ec2-user/time_kpi/part-00009-4b37ec84-e02f-4e16-ac74-dd9bdc7673fa-c000.json
-rw-r--r--   1 hadoop hadoop      29987 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00010-fe102876-2f77-4ed6-92f9-5ebe6068e98a-c000.json
-rw-r--r--   1 hadoop hadoop      32630 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00011-41fb67b8-72d1-4759-b9db-e157ec35f5c0-c000.json
-rw-r--r--   1 hadoop hadoop      37073 2023-02-10 11:22 /user/ec2-user/time_kpi/part-00012-9388788a-f3a5-4390-8100-36062cf0ae52-c000.json
-rw-r--r--   1 hadoop hadoop      38782 2023-02-10 11:22 /user/ec2-user/time_kpi/part-00013-5a98d76f-f466-4eac-8a10-66935bd8d6a7-c000.json
-rw-r--r--   1 hadoop hadoop      34740 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00014-894ecad3-e225-456b-a7a7-d5d0a5245a24-c000.json
-rw-r--r--   1 hadoop hadoop      32043 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00015-576d578a-b6a2-4f74-818b-4a7245825e59-c000.json
-rw-r--r--   1 hadoop hadoop      30969 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00016-90c33df4-2a92-4069-bc83-97dd00238dc3-c000.json
-rw-r--r--   1 hadoop hadoop      33042 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00017-9a345f28-6455-46d9-9a43-6636774cda20-c000.json
-rw-r--r--   1 hadoop hadoop      36549 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00018-98953280-4c48-42c0-b00a-bc91f3cc01b1-c000.json
-rw-r--r--   1 hadoop hadoop      32775 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00019-21540ca8-0617-4590-a2c6-20b1f386eef7-c000.json
-rw-r--r--   1 hadoop hadoop      29285 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00020-00ca3a32-4bf5-478f-a4a5-6f72f7d27cc3-c000.json
-rw-r--r--   1 hadoop hadoop      34752 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00021-e9fad46a-32e2-4564-b43f-b4c6047ba90c-c000.json
-rw-r--r--   1 hadoop hadoop      33179 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00022-beaecd96-064f-4a07-951b-cdc3500abc02-c000.json
-rw-r--r--   1 hadoop hadoop      34776 2023-02-10 11:22 /user/ec2-user/time_kpi/part-00023-ede099aa-6d5b-435c-b2bb-a1820ae05f65-c000.json
-rw-r--r--   1 hadoop hadoop      37853 2023-02-10 11:22 /user/ec2-user/time_kpi/part-00024-775a0adb-d1ca-455c-bea7-96c6815c3fb4-c000.json
-rw-r--r--   1 hadoop hadoop      29780 2023-02-10 11:22 /user/ec2-user/time_kpi/part-00025-5ff28bd8-01d8-4e04-8494-5bd574bc14c7-c000.json
-rw-r--r--   1 hadoop hadoop      31304 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00026-29085a47-7433-4aaf-905a-59492e20329f-c000.json
-rw-r--r--   1 hadoop hadoop      38756 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00027-b7223e4a-a0ae-4a3d-8fcd-39ded6366925-c000.json
-rw-r--r--   1 hadoop hadoop      35823 2023-02-10 11:22 /user/ec2-user/time_kpi/part-00028-79e8a7f7-af3b-460c-936d-e77f94c0daab-c000.json
-rw-r--r--   1 hadoop hadoop      33776 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00029-17e3f271-e44a-40c4-8676-9fb16ce2cfa5-c000.json
-rw-r--r--   1 hadoop hadoop      31180 2023-02-10 11:23 /user/ec2-user/time_kpi/part-00030-d2f4a445-e743-4891-9a84-24a2bab8daa4-c000.json
```

hadoop fs -ls /user/ec2-user/country_kpi/

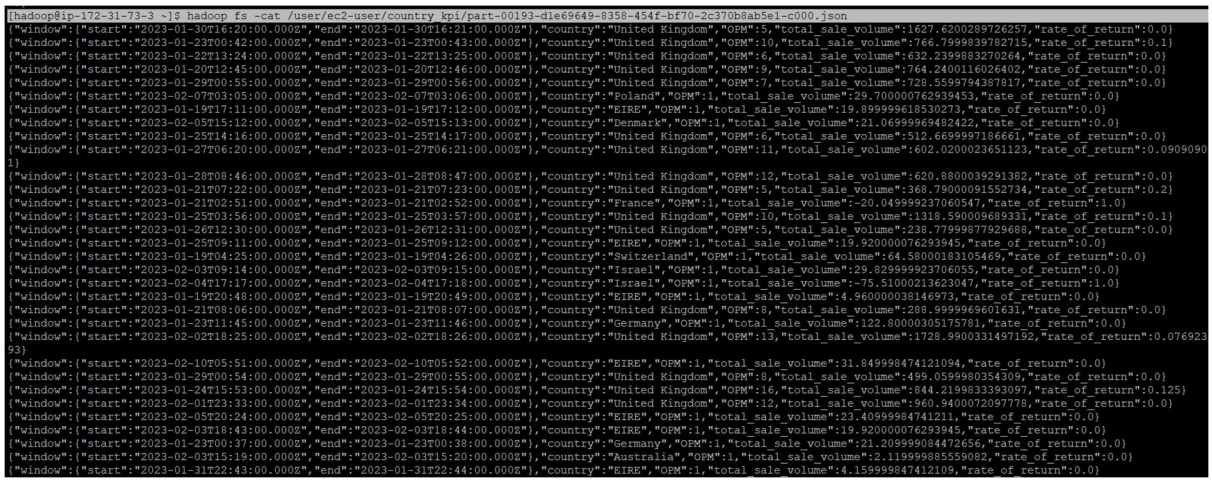
```
[hadoop@ip-172-31-73-3 ~]$ hadoop fs -ls /user/ec2-user/country_kpi/
Found 209 items
drwxr-xr-x   - hadoop hadoop          0 2023-02-10 11:25 /user/ec2-user/country_kpi/_spark_metadata
-rw-r--r--   1 hadoop hadoop          0 2023-02-10 11:24 /user/ec2-user/country_kpi/part-00000-1a0dc098-1268-46ac-9ade-9642e42ca125-c000.json
-rw-r--r--   1 hadoop hadoop          0 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00000-1b88fca9-d215-45a1-b544-5126de8ba804-c000.json
-rw-r--r--   1 hadoop hadoop      54444 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00000-5dad3d36-c0de-46e8-840b-0a8a1e22628f-c000.json
-rw-r--r--   1 hadoop hadoop          0 2023-02-10 11:23 /user/ec2-user/country_kpi/part-00000-7a872539-fcbc-4b52-9e6f-528c4318a4b3-c000.json
-rw-r--r--   1 hadoop hadoop          0 2023-02-10 11:25 /user/ec2-user/country_kpi/part-00000-d661ae4e-a8f6-4c14-9bb3-1270660ba7da-c000.json
-rw-r--r--   1 hadoop hadoop      53445 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00001-e6e1c09e-45e4-4629-a0d6-a0db1b1300d1-c000.json
-rw-r--r--   1 hadoop hadoop      58206 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00002-37047192-0b13-4423-ad44-2225b87af198-c000.json
-rw-r--r--   1 hadoop hadoop      50201 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00003-940c82f2-f3ff-4771-b200-a988c347d20f-c000.json
-rw-r--r--   1 hadoop hadoop      51543 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00004-2afee078-8ffc-4d82-b36c-9d05d5eb9afd-c000.json
-rw-r--r--   1 hadoop hadoop      57158 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00005-826fbc94-c5a1-47f6-a9de-32ebaffa0939-c000.json
-rw-r--r--   1 hadoop hadoop      51820 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00006-14dd3fc9-aa85-4ba8-b092-c92124fdfd15-c000.json
-rw-r--r--   1 hadoop hadoop      51568 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00007-3b15a076-b7d2-4f81-9894-387e64eec1db-c000.json
-rw-r--r--   1 hadoop hadoop      54042 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00008-97df4059-b853-463c-b72d-49f2e5ff3d8c-c000.json
-rw-r--r--   1 hadoop hadoop      58344 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00009-f098ce72-0e83-4997-9bbe-3a3ee9561208-c000.json
-rw-r--r--   1 hadoop hadoop      54644 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00010-6d13d2af-b1e5-40a4-ad42-ee87e12202fe-c000.json
-rw-r--r--   1 hadoop hadoop      51132 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00011-d0062558-d77c-4d3b-94ca-767bd18374e2-c000.json
-rw-r--r--   1 hadoop hadoop      51672 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00012-476a0a4e-93a6-4985-9e39-be7a6e5f4cf8-c000.json
-rw-r--r--   1 hadoop hadoop      53840 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00013-41056b19-8e1a-45a3-8484-0100cd2ea42d-c000.json
-rw-r--r--   1 hadoop hadoop      56542 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00014-79a128cb-f772-44da-a214-9b683fbbba5e-c000.json
-rw-r--r--   1 hadoop hadoop      54011 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00015-91a0dc98-db77-4a47-8cf4-83f49cd23daf-c000.json
-rw-r--r--   1 hadoop hadoop      57456 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00016-c56c06b6-a31d-466b-afb9-1d05f32dd5b6-c000.json
-rw-r--r--   1 hadoop hadoop      48475 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00017-bdda90cd-ca3f-4e1e-ab7f-d0ce31455ffa-c000.json
-rw-r--r--   1 hadoop hadoop      53356 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00018-b1128517-0ad5-405f-bd27-fe225a3687ea-c000.json
-rw-r--r--   1 hadoop hadoop      56931 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00019-22cda3ba-58fc-4ed8-9415-47dff9df9c9b-c000.json
-rw-r--r--   1 hadoop hadoop      54988 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00020-44f614d3-5135-4a2e-90e7-6a1f6c6610d5-c000.json
-rw-r--r--   1 hadoop hadoop      47612 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00021-3a6a795d-12d6-4bb2-b818-f429176ea64e-c000.json
-rw-r--r--   1 hadoop hadoop      57270 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00022-5d9a10be-e9ec-4325-825d-2cf33210b53a-c000.json
-rw-r--r--   1 hadoop hadoop      55309 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00023-3f6300c8-0efd-48eb-813e-840f3ba60128-c000.json
-rw-r--r--   1 hadoop hadoop      59112 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00024-fc4be501-3a0e-444c-917e-05db59976b42-c000.json
-rw-r--r--   1 hadoop hadoop      54916 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00025-87bec448-2f31-4d7f-b272-0ebd47ee6155-c000.json
-rw-r--r--   1 hadoop hadoop      59491 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00026-44e8094c-af4c-4f2e-b645-01535d9caeea-c000.json
-rw-r--r--   1 hadoop hadoop      58192 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00027-840b0726-2144-4003-965e-72e100808b12-c000.json
-rw-r--r--   1 hadoop hadoop      53578 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00028-bb4f626b-fb31-4b66-a6bc-74bbe736bd3f-c000.json
-rw-r--r--   1 hadoop hadoop      52366 2023-02-10 11:22 /user/ec2-user/country_kpi/part-00029-51db1dcd-448c-466d-b73a-aea239a43c0c-c000.json
```

We also checked inside individual JSON files for data to verify for both time KPI and country KPI.

hadoop fs -cat /user/ec2-user/time_kpi/part-00186-82f71735-faf0-4cdb-aed5-a6a2a116774c-c000.json

```
[hadoop@ip-172-31-73-3 ~]$ hadoop fs -cat /user/ec2-user/time_kpi/part-00186-82f71735-faf0-4cdb-aed5-a6a2a116774c-c000.json
{"window":{"start":"2023-01-29T10:55:00.000Z","end":"2023-01-29T10:56:00.000Z"},"OPM":9,"total_sale_volume":398.81999683380127,"average_transaction_size":44.31333298153348,"rate_of_return":
0.1111111111111111}
{"window":{"start":"2023-02-08T05:14:00.000Z","end":"2023-02-08T05:15:00.000Z"},"OPM":10,"total_sale_volume":96.59999752044678,"average_transaction_size":9.659999752044678,"rate_of_return":
0.1}
{"window":{"start":"2023-01-26T10:43:00.000Z","end":"2023-01-26T10:44:00.000Z"},"OPM":12,"total_sale_volume":-140.01998686790466,"average_transaction_size":-11.668332238992056,"rate_of_retu
rn":0.08333333333333333}
{"window":{"start":"2023-01-31T11:52:00.000Z","end":"2023-01-31T11:53:00.000Z"},"OPM":14,"total_sale_volume":733.9699745178223,"average_transaction_size":52.42642675127302,"rate_of_return":
0.0}
{"window":{"start":"2023-01-30T16:27:00.000Z","end":"2023-01-30T16:28:00.000Z"},"OPM":12,"total_sale_volume":377.0599994659424,"average_transaction_size":31.421666622161865,"rate_of_return"
:0.0}
{"window":{"start":"2023-01-25T22:40:00.000Z","end":"2023-01-25T22:41:00.000Z"},"OPM":11,"total_sale_volume":124.1799955368042,"average_transaction_size":11.289090503345836,"rate_of_return"
:0.09090909090909091}
{"window":{"start":"2023-01-20T21:01:00.000Z","end":"2023-01-20T21:02:00.000Z"},"OPM":14,"total_sale_volume":771.0200109481812,"average_transaction_size":55.072857924870085,"rate_of_return"
:0.0}
{"window":{"start":"2023-02-02T20:37:00.000Z","end":"2023-02-02T20:38:00.000Z"},"OPM":7,"total_sale_volume":505.90998792648315,"average_transaction_size":72.27285541806903,"rate_of_return":
0.14285714285714285}
{"window":{"start":"2023-02-07T05:33:00.000Z","end":"2023-02-07T05:34:00.000Z"},"OPM":14,"total_sale_volume":1552.0399932861328,"average_transaction_size":110.85999952043805,"rate_of_return
":0.07142857142857142}
{"window":{"start":"2023-01-22T01:28:00.000Z","end":"2023-01-22T01:29:00.000Z"},"OPM":10,"total_sale_volume":678.4999761581421,"average_transaction_size":67.8499976158142,"rate_of_return":0
.1}
{"window":{"start":"2023-01-19T23:31:00.000Z","end":"2023-01-19T23:32:00.000Z"},"OPM":12,"total_sale_volume":727.9899928569794,"average_transaction_size":60.66583273808161,"rate_of_return":
0.08333333333333333}
{"window":{"start":"2023-01-22T20:39:00.000Z","end":"2023-01-22T20:40:00.000Z"},"OPM":9,"total_sale_volume":540.9199953079224,"average_transaction_size":60.102221700880264,"rate_of_return":
0.1111111111111111}
{"window":{"start":"2023-01-18T02:33:00.000Z","end":"2023-01-18T02:34:00.000Z"},"OPM":5,"total_sale_volume":616.1899719238281,"average_transaction_size":123.23799438476563,"rate_of_return":
0.0}
{"window":{"start":"2023-01-19T12:55:00.000Z","end":"2023-01-19T12:56:00.000Z"},"OPM":9,"total_sale_volume":630.7000045776367,"average_transaction_size":70.07777828640408,"rate_of_return":0
.0}
{"window":{"start":"2023-02-08T05:50:00.000Z","end":"2023-02-08T05:51:00.000Z"},"OPM":11,"total_sale_volume":1204.6099739074707,"average_transaction_size":109.50999762795188,"rate_of_return
":0.18181818181818182}
{"window":{"start":"2023-01-26T12:09:00.000Z","end":"2023-01-26T12:10:00.000Z"},"OPM":4,"total_sale_volume":54.80000114440918,"average_transaction_size":13.700000286102295,"rate_of_return":
0.25}
```

hadoop fs -cat /user/ec2-user/country_kpi/part-00193-d1e69649-8358-454f-bf70-2c370b8ab5e1-c000.json



Transfer of all files to HDFS and finally to Local System:

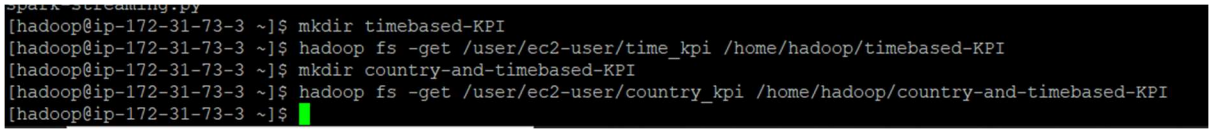
Finally, we transferred the generated files from EC2 location to hadoop HDFS location. For that we created 2 different directories in HDFS and copied all data inside it. we used below commands for that. So that we can download to our local machine from it through WinSCP.

mkdir timebased-KPI

hadoop fs -get /user/ec2-user/time_kpi /home/hadoop/timebased-KPI

mkdir country-and-timebased-KPI

hadoop fs -get /user/ec2-user/country_kpi /home/hadoop/country-and-timebased-KPI



Finally, we transferred that data into local system through WinSCP.