

Load data from Kafka to Hadoop

<Steps to run the python file to load data from Kafka>

1. We start with first importing the modules

```
*C:\Users\vancy\Downloads\CabRides_Kafka_to_Hadoop.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
new 2 CabRides_Kafka_to_Hadoop.py
1
2
3 # Script is to get all the cab rides related click stream data residing in given Kafka server to Hadoop cluster
4
5
6 # we start with first importing the modules
7 import os,sys
8 from pyspark.sql import SparkSession
9 from pyspark.sql.functions import *
10
11
```

2. Set required environment variables needed to get the data

```
# Next step is to set required environment variables needed to get the data
os.environ["PYSPARK_PYTHON"] = "/opt/cloudera/parcels/Anaconda/bin/python"
os.environ["JAVA_HOME"] = "/usr/java/jdk1.8.0_161/jre"
os.environ["SPARK_HOME"] = "/opt/cloudera/parcels/SPARK2-2.3.0.cloudera2-1.cdh5.13.3.p0.316101/lib/spark2/"
os.environ["PYLIB"] = os.environ["SPARK_HOME"] + "/python/lib"
sys.path.insert(0, os.environ["PYLIB"] + "/py4j-0.10.6-src.zip")
sys.path.insert(0, os.environ["PYLIB"] + "/pyspark.zip")
```

3. Next initialize Spark session

```
# We will now initialize Spark session
spark = SparkSession \
    .builder \
    .appName("Kafka-to-local") \
    .getOrCreate()
```

4. Read Data from kafka server from given Kafka server details

```
# From the given server connection details connect to kafka and get the stream in a dataframe
# Read Input from kafka
streamdf = spark.readStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers", "18.211.252.152:9092") \
    .option("startingOffsets", "earliest") \
    .option("subscribe", "de-capstone3") \
    .load()
```

5. Keep relevant field 'value' rename it to 'value_str' and drop other irrelevant fields

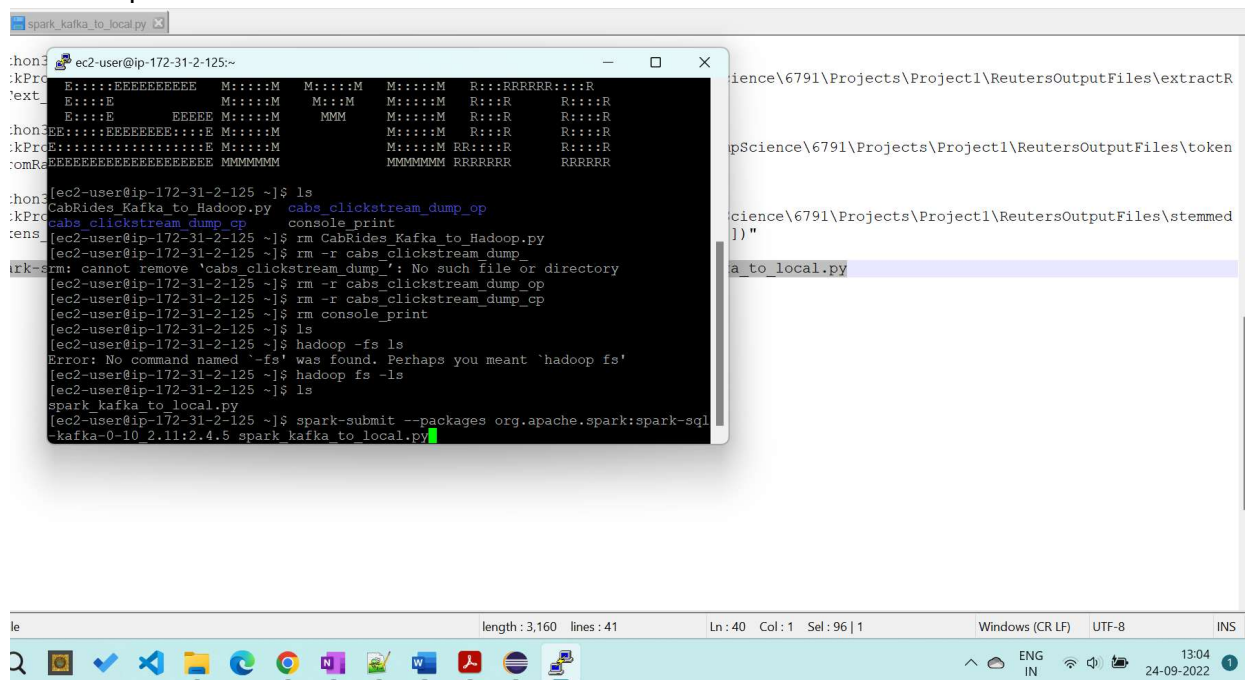
```
# get only relevant fields and drop others
streamdf= streamdf \
    .withColumn('value_str',streamdf['value'].cast('string').alias('key_str')).drop('value') \
    .drop('key','topic','partition','offset','timestamp','timestampType')
```

<Steps to load the data into Hadoop>

1. Write the click stream to folder 'cabs_clickstream_dump_op' in Hadoop

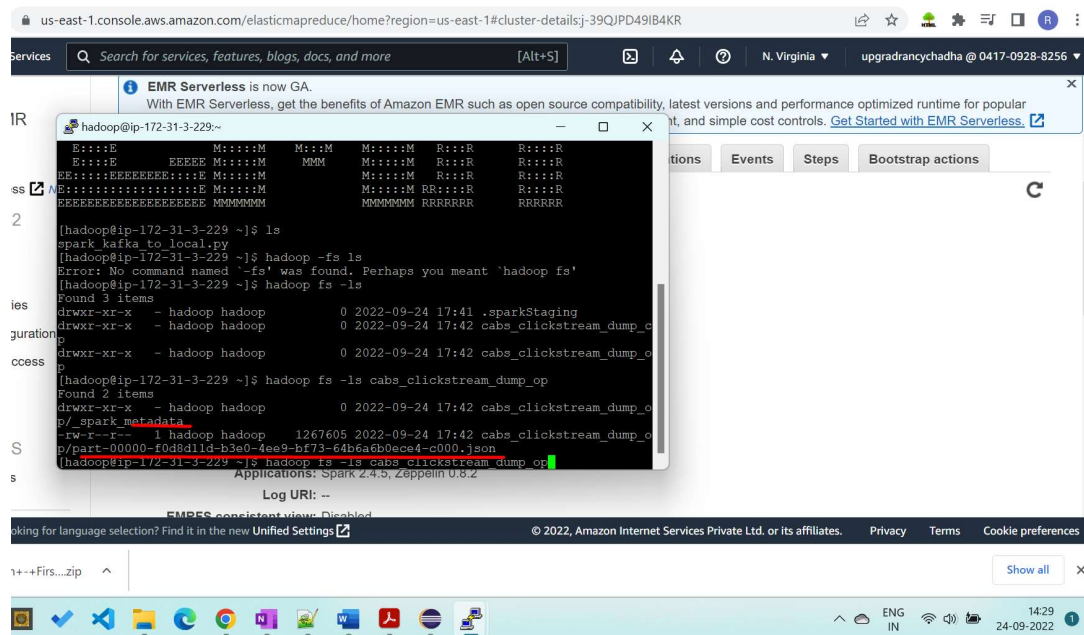
```
#Writing the click stream to a folder in Hadoop
streamdf.writeStream \
    .format("json") \
    .outputMode("append") \
    .option("path", "cabs_clickstream_dump_op") \
    .option("checkpointLocation", "cabs_clickstream_dump_cp") \
    .start() \
    .awaitTermination()
```

2. Submit Spark Job



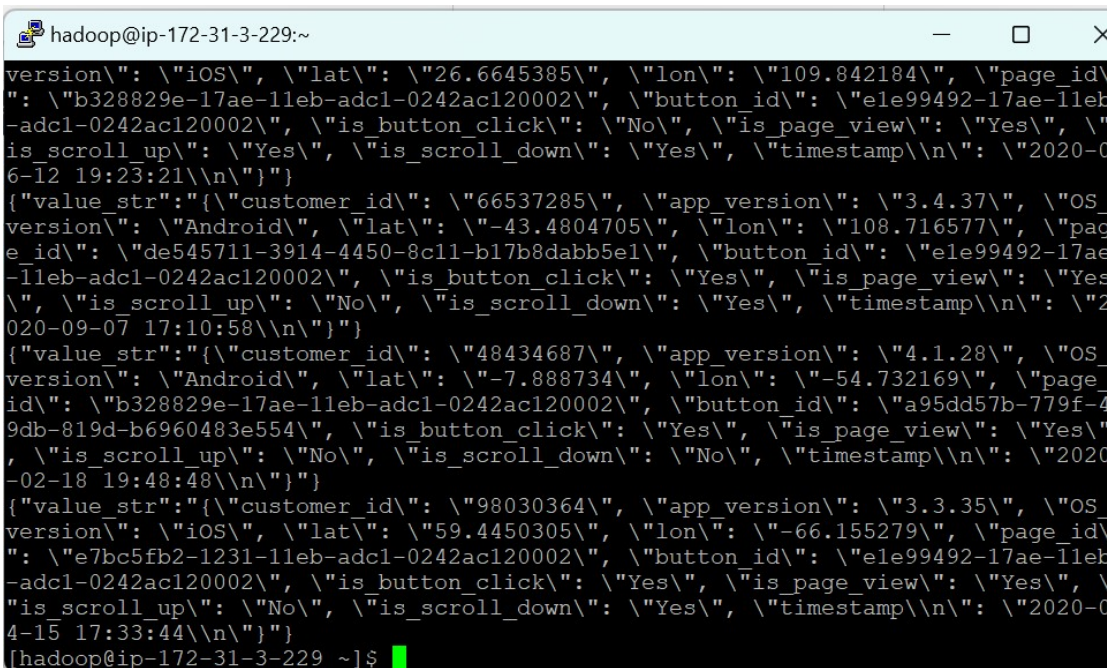
```
spark_kafka_to_local.py
ec2-user@ip-172-31-2-125:~$ ls
CabRides Kafka to Hadoop.py  cabs_clickstream_dump_op
cabs_clickstream_dump_cp    console_print
ec2-user@ip-172-31-2-125 ~$ rm CabRides Kafka to Hadoop.py
ec2-user@ip-172-31-2-125 ~$ rm -r cabs_clickstream_dump
rm: cannot remove 'cabs_clickstream_dump': No such file or directory
ec2-user@ip-172-31-2-125 ~$ rm -r cabs_clickstream_dump_op
ec2-user@ip-172-31-2-125 ~$ rm -r cabs_clickstream_dump_cp
ec2-user@ip-172-31-2-125 ~$ rm console_print
ec2-user@ip-172-31-2-125 ~$ ls
ec2-user@ip-172-31-2-125 ~$ hadoop -fs ls
Error: No command named '-fs' was found. Perhaps you meant 'hadoop fs'
ec2-user@ip-172-31-2-125 ~$ hadoop fs -ls
ec2-user@ip-172-31-2-125 ~$ ls
spark_kafka_to_local.py
ec2-user@ip-172-31-2-125 ~$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 spark_kafka_to_local.py
```

<Screenshot of the data>



The screenshot shows the AWS EMR console interface. A terminal window is open, displaying the following commands and output:

```
hadoop@ip-172-31-3-229:~$ ls
spark_kafka_to_local.py
hadoop@ip-172-31-3-229:~$ hadoop -fs ls
Error: No command named '-fs' was found. Perhaps you meant 'hadoop fs'
hadoop@ip-172-31-3-229:~$ hadoop fs -ls
Found 3 items
drwxr-xr-x - hadoop hadoop 0 2022-09-24 17:41 .sparkStaging
drwxr-xr-x - hadoop hadoop 0 2022-09-24 17:42 cabs_clickstream_dump_op
drwxr-xr-x - hadoop hadoop 0 2022-09-24 17:42 cabs_clickstream_dump_op
hadoop@ip-172-31-3-229:~$ hadoop fs -ls cabs_clickstream_dump_op
Found 2 items
drwxr-xr-x - hadoop hadoop 0 2022-09-24 17:42 cabs_clickstream_dump_op
-rw-r--r-- 1 hadoop hadoop 1267605 2022-09-24 17:42 cabs_clickstream_dump_op
part-00000-f0d8d11d-b3e0-4ee9-bf73-64b6a6b0ece4-c000.json
hadoop@ip-172-31-3-229:~$ hadoop fs -ls cabs_clickstream_dump_op
Application: Spark2Job, Zeppelin User
```



The screenshot shows a terminal window with the following JSON output:

```
version": "iOS", "lat": "26.6645385", "lon": "109.842184", "page_id": "b328829e-17ae-11eb-adc1-0242ac120002", "button_id": "e1e99492-17ae-11eb-adc1-0242ac120002", "is_button_click": "No", "is_page_view": "Yes", "is_scroll_up": "Yes", "is_scroll_down": "Yes", "timestamp": "2020-06-12 19:23:21"}
{"value_str": {"customer_id": "66537285", "app_version": "3.4.37", "OS_version": "Android", "lat": "-43.4804705", "lon": "108.716577", "page_id": "de545711-3914-4450-8c11-b17b8dabb5e1", "button_id": "e1e99492-17ae-11eb-adc1-0242ac120002", "is_button_click": "Yes", "is_page_view": "Yes", "is_scroll_up": "No", "is_scroll_down": "Yes", "timestamp": "2020-09-07 17:10:58"}}
{"value_str": {"customer_id": "48434687", "app_version": "4.1.28", "OS_version": "Android", "lat": "-7.888734", "lon": "-54.732169", "page_id": "b328829e-17ae-11eb-adc1-0242ac120002", "button_id": "a95dd57b-779f-49db-819d-b6960483e554", "is_button_click": "Yes", "is_page_view": "Yes", "is_scroll_up": "No", "is_scroll_down": "No", "timestamp": "2020-02-18 19:48:48"}}
{"value_str": {"customer_id": "98030364", "app_version": "3.3.35", "OS_version": "iOS", "lat": "59.4450305", "lon": "-66.155279", "page_id": "e7bc5fb2-1231-11eb-adc1-0242ac120002", "button_id": "e1e99492-17ae-11eb-adc1-0242ac120002", "is_button_click": "Yes", "is_page_view": "Yes", "is_scroll_up": "No", "is_scroll_down": "Yes", "timestamp": "2020-04-15 17:33:44"}}
hadoop@ip-172-31-3-229:~$
```