## Load data from Kafka to Hadoop

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

1. Import all libraries required for running the code.

import os

import sys

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")

from pyspark.sql import SparkSession

from pyspark.sql.functions import \*

1. Initialize the spark session

spark = SparkSession \

.builder \

.appName("Kafka-to-local") \

.getOrCreate()

1. Read data from Kafka by subscribing to **de-capstone3 topic**

df = spark.readStream \

.format("kafka") \

.option("kafka.bootstrap.servers", "18.211.252.152:9092") \

.option("startingOffsets", "earliest") \

.option("subscribe", "de-capstone3") \

.load()

1. Drop the columns which are not required and change the column name of value to value\_str

df= df \

.withColumn('value\_str',df['value'].cast('string').alias('key\_str')).drop('value') \

.drop('key','topic','partition','offset','timestamp','timestampType')

1. Write the data to hadoop using below code

df.writeStream \

.format("json") \

.outputMode("append") \

.option("path", "/user/root/clickstream\_dump\_op") \

.option("checkpointLocation", "/user/root/clickstream\_dump\_cp") \

.start() \

.awaitTermination()

1. Run below command for running the python file.

Spark2-submit <spark-sql jarfile> spark\_kafka\_to\_local.py

**<Steps to load the data into Hadoop>**

df.writeStream \

.format("json") \

.outputMode("append") \

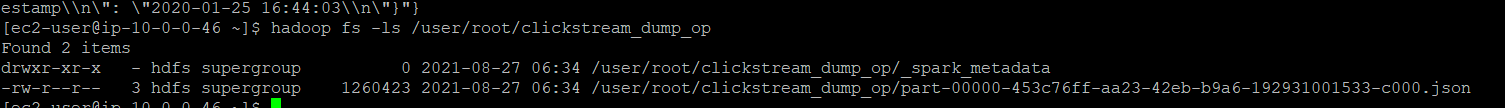
.option("path", "/user/root/clickstream\_dump\_op") \

.option("checkpointLocation", "/user/root/clickstream\_dump\_cp") \

.start() \

.awaitTermination()

**<Screenshot of the data>**

****

