In [1]:

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

spark = SparkSession\
    .builder\
    .appName("chapter-21-stream-kafka")\
    .getOrCreate()

import os
SPARK_BOOK_DATA_PATH = os.environ['SPARK_BOOK_DATA_PATH']
```

Kafka

https://spark.apache.org/docs/2.4.0/structured-streaming-kafka-integration.html#deploying (https://spark.apache.org/docs/2.4.0/structured-streaming-kafka-integration.html#deploying)

`./bin/spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.0`

setup ¶

steps to run kafka and create topic (https://github.com/wgong/py4kids/blob/master/lesson-71-kafka/Calories-Alert-Kafka/kafka.README.md)

In [2]:

```
# Subscribe to 1 topic
streaming = spark.readStream.format("kafka")\
  .option("kafka.bootstrap.servers", "localhost:9092")\
  .option("subscribe", "Hello-Kafka")\
  .load()
```

In [3]:

```
streaming.printSchema()
```

```
root
 |-- key: binary (nullable = true)
 |-- value: binary (nullable = true)
 |-- topic: string (nullable = true)
 |-- partition: integer (nullable = true)
 |-- offset: long (nullable = true)
 |-- timestamp: timestamp (nullable = true)
 |-- timestampType: integer (nullable = true)
```

write to memory for test

In [4]:

```
streaming.selectExpr("CAST(key AS STRING)", "CAST(value AS STRING)")
  .writeStream\
  .queryName("test_transform")\
  .format("memory")\
  .outputMode("append")\
  .start()
```

Out[4]:

```
<pyspark.sql.streaming.StreamingQuery at 0x7fe08da2cac8>
```

In [8]:

```
spark.sql("select * from test_transform").show(truncate=False)
```

```
+----+------------------------------------------+
|key |value                                     |
+----+------------------------------------------+
|null|kafka is a distributed pub/sub message broker|
|null|spark is a distributed big-data platform  |
+----+------------------------------------------+
```

In [9]:

```
spark.sql("select * from test_transform").show(truncate=False)
```

```
+----+------------------------------------------+
|key |value                                     |
+----+------------------------------------------+
|null|kafka is a distributed pub/sub message broker|
|null|spark is a distributed big-data platform  |
|null|spark also has mllib for machine learning |
|null|databricks is the company behind spark    |
+----+------------------------------------------+
```

In [10]:

```
spark.sql("select * from test_transform").show(truncate=False)
```

```
+----+------------------------------------------+
|key |value                                     |
+----+------------------------------------------+
|null|kafka is a distributed pub/sub message broker|
|null|spark is a distributed big-data platform  |
|null|spark also has mllib for machine learning |
|null|databricks is the company behind spark    |
|null|tensorflow 2.0 was released last week     |
|null|tensorflow.js is very interesting         |
+----+------------------------------------------+
```

write to another topic

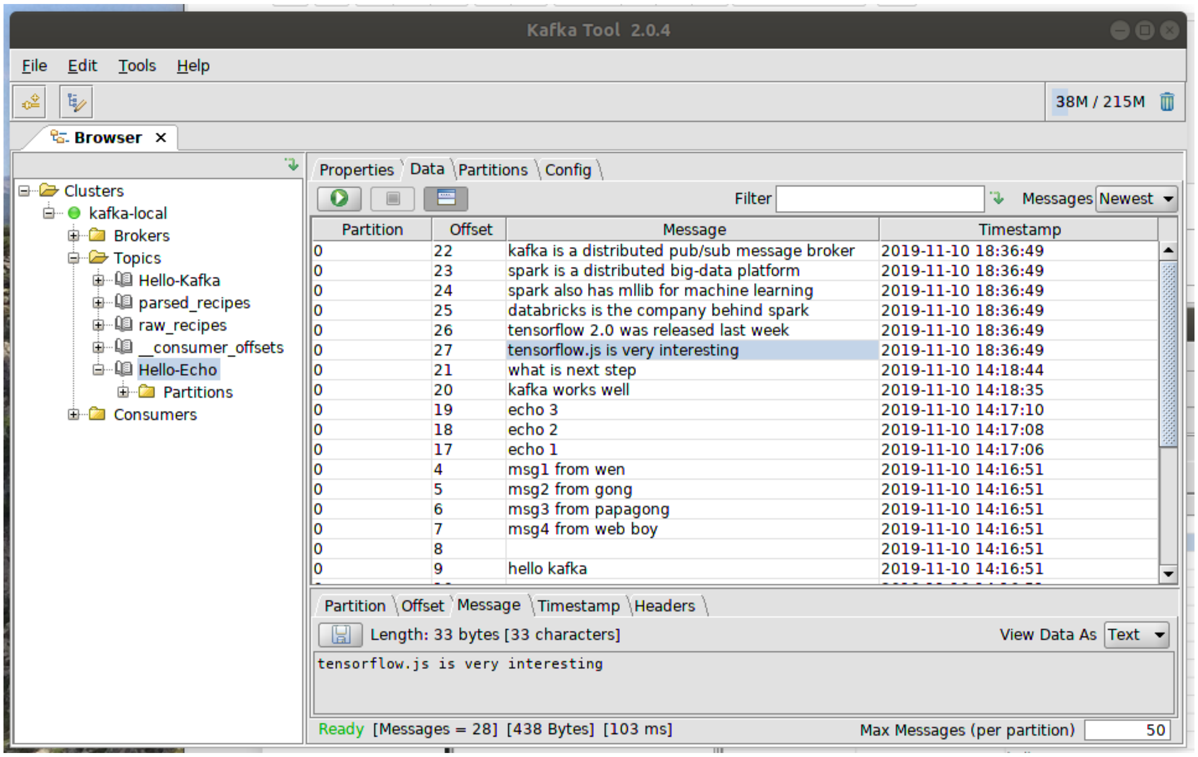
In [11]:

```
streaming.selectExpr("CAST(key AS STRING)", "CAST(value AS STRING)")
  .writeStream\
  .format("kafka")\
  .option("kafka.bootstrap.servers", "localhost:9092")\
  .option("checkpointLocation", "/tmp/kafka-checkpoint")\
  .option("topic", "Hello-Echo")\
  .start()
```

Out[11]:

```
<pyspark.sql.streaming.StreamingQuery at 0x7fe08da2cf60>
```

Check in Kafkatool to see messages are echoed to the new topic = "Hello-Echo"



Partition	Offset	Message	Timestamp
0	22	kafka is a distributed pub/sub message broker	2019-11-10 18:36:49
0	23	spark is a distributed big-data platform	2019-11-10 18:36:49
0	24	spark also has mllib for machine learning	2019-11-10 18:36:49
0	25	databricks is the company behind spark	2019-11-10 18:36:49
0	26	tensorflow 2.0 was released last week	2019-11-10 18:36:49
0	27	tensorflow.js is very interesting	2019-11-10 18:36:49
0	21	what is next step	2019-11-10 14:18:44
0	20	kafka works well	2019-11-10 14:18:35
0	19	echo 3	2019-11-10 14:17:10
0	18	echo 2	2019-11-10 14:17:08
0	17	echo 1	2019-11-10 14:17:06
0	4	msg1 from wen	2019-11-10 14:16:51
0	5	msg2 from gong	2019-11-10 14:16:51
0	6	msg3 from papagong	2019-11-10 14:16:51
0	7	msg4 from web boy	2019-11-10 14:16:51
0	8		2019-11-10 14:16:51
0	9	hello kafka	2019-11-10 14:16:51

Length: 33 bytes [33 characters]

```
tensorflow.js is very interesting
```

Ready [Messages = 28] [438 Bytes] [103 ms]

Max Messages (per partition) 50

In [ ]:

```

```

In [ ]:

```

```

below codes are not tested

In [ ]:

```
# Subscribe to 1 topic
df1 = spark.readStream.format("kafka")\
  .option("kafka.bootstrap.servers", "host1:port1,host2:port2")\
  .option("subscribe", "topic1")\
  .load()
```

In [ ]:

```
# Subscribe to multiple topics
df2 = spark.readStream.format("kafka")\
  .option("kafka.bootstrap.servers", "host1:port1,host2:port2")\
  .option("subscribe", "topic1,topic2")\
  .load()
```

In [ ]:

```
# Subscribe to a pattern
df3 = spark.readStream.format("kafka")\
  .option("kafka.bootstrap.servers", "host1:port1,host2:port2")\
  .option("subscribePattern", "topic.*")\
  .load()
```

In [ ]:

```
# COMMAND ----------

df1.selectExpr("topic", "CAST(key AS STRING)", "CAST(value AS STRIN
  .writeStream\
  .format("kafka")\
  .option("kafka.bootstrap.servers", "host1:port1,host2:port2")\
  .option("checkpointLocation", "/to/HDFS-compatible/dir")\
  .start()

df1.selectExpr("CAST(key AS STRING)", "CAST(value AS STRING)")\
  .writeStream\
  .format("kafka")\
  .option("kafka.bootstrap.servers", "host1:port1,host2:port2")\
  .option("checkpointLocation", "/to/HDFS-compatible/dir")\
  .option("topic", "topic1")\
  .start()


# COMMAND ----------

socketDF = spark.readStream.format("socket")\
  .option("host", "localhost").option("port", 9999).load()
```

In [ ]:

```
# COMMAND ----------

activityCounts.writeStream.trigger(processingTime='5 seconds')\
  .format("console").outputMode("complete").start()
```

In [ ]:

```
# COMMAND ----------

activityCounts.writeStream.trigger(once=True)\
  .format("console").outputMode("complete").start()
```

In [ ]:

```
# COMMAND ----------
```