

KAFKA E SPARK STREAMING

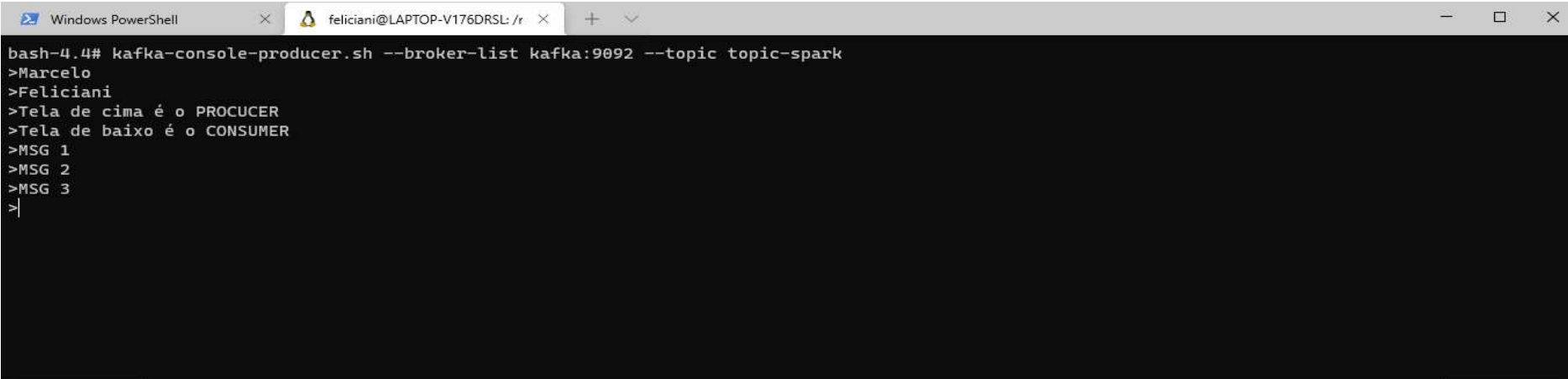
1. Preparação do ambiente no Kafka

a) Criar o tópico “topic-spark” com 1 partição e o fator de replicação = 1

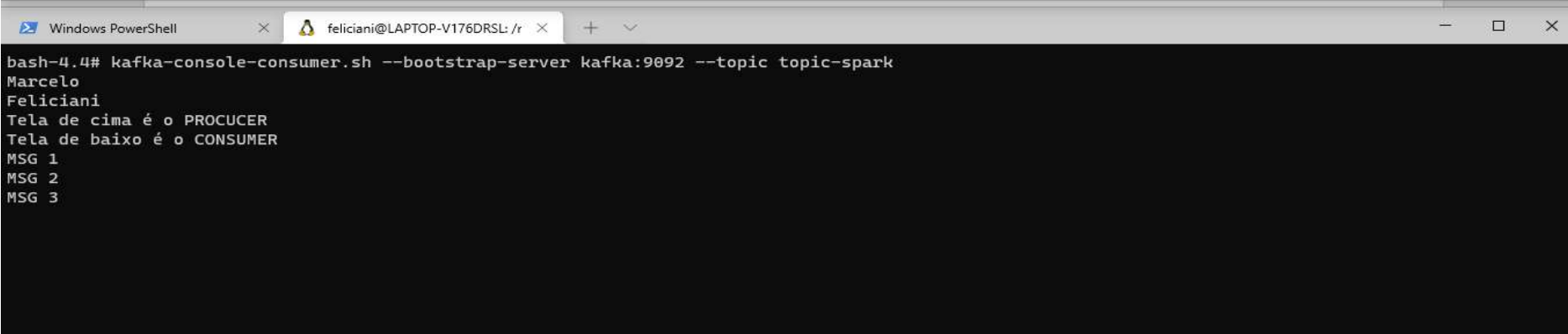
```
PS E:\projetos\docker-spark\spark> docker exec -it kafka bash
bash-4.4# kafka-topics.sh --bootstrap-server kafka:9092 --create --topic topic-spark --partitions 1 --replication-factor 1
bash-4.4# kafka-topics.sh --bootstrap-server kafka:9092 --list
topic-spark
bash-4.4# kafka-topics.sh --bootstrap-server kafka:9092 --describe
Topic:topic-spark      PartitionCount:1      ReplicationFactor:1   Configs:segment.bytes=1073741824
      Topic: topic-spark      Partition: 0      Leader: 1001      Replicas: 1001      Isr: 1001
bash-4.4#
```

b) Inserir as seguintes mensagens no tópico: - Msg1, Msg2, Msg3

c) Criar um consumidor no Kafka para ler o “topic-spark”



```
bash-4.4# kafka-console-producer.sh --broker-list kafka:9092 --topic topic-spark
>Marcelo
>Feliciani
>Tela de cima é o PROCUCER
>Tela de baixo é o CONSUMER
>MSG 1
>MSG 2
>MSG 3
>
```



```
bash-4.4# kafka-console-consumer.sh --bootstrap-server kafka:9092 --topic topic-spark
Marcelo
Feliciani
Tela de cima é o PROCUCER
Tela de baixo é o CONSUMER
MSG 1
MSG 2
MSG 3
```

SPARK no SCALA

Fonte: <https://spark.apache.org/docs/2.4.1/streaming-kafka-0-10-integration.html>

```
feliciani@LAPTOP-V176DRSL:/mnt/e/projetos/docker-spark/spark$ docker exec -it jupyter-spark bash
root@jupyter-spark:/# spark-shell --packages org.apache.spark:spark-streaming-kafka-0-10_2.11:2.4.1
Ivy Default Cache set to: /root/.ivy2/cache
The jars for the packages stored in: /root/.ivy2/jars
:: loading settings :: url = jar:file:/opt/spark-2.4.1-bin-without-hadoop/jars/ivy-2.4.0.jar!/org/apache/ivy/core/settings/ivysettings.xml
org.apache.spark#spark-streaming-kafka-0-10_2.11 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-fad4dade-8e94-4503-9e4f-491a2702a4de;1.0
  confs: [default]
  found org.apache.spark#spark-streaming-kafka-0-10_2.11;2.4.1 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;1.1.7.1 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-streaming-kafka-0-10_2.11/2.4.1/spark-streaming-kafka-0-10_2.11-2.4.1.jar ...
[SUCCESSFUL ] org.apache.spark#spark-streaming-kafka-0-10_2.11;2.4.1!spark-streaming-kafka-0-10_2.11.jar (1262ms)
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 (1124ms)
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 (192ms)
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 (238ms)
downloading https://repo1.maven.org/maven2/org/xerial/snappy/snappy-java/1.1.7.1/snappy-java-1.1.7.1.jar ...
[SUCCESSFUL ] org.xerial.snappy#snappy-java;1.1.7.1!snappy-java.jar (711ms)
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 (225ms)
:: resolution report :: resolve 29055ms :: artifacts dl 3779ms
  :: modules in use:
    org.apache.kafka#kafka-clients;2.0.0 from central in [default]
    org.apache.spark#spark-streaming-kafka-0-10_2.11;2.4.1 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;1.1.7.1 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-fad4dade-8e94-4503-9e4f-491a2702a4de
confs: [default]
6 artifacts copied, 0 already retrieved (4436kB/281ms)
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
21/07/04 00:27:00 WARN spark.SparkConf: Note that spark.local.dir will be overridden by the value set by the cluster manager (via SPARK_LOCAL_DIRS i
n mesos/standalone/kubernetes and LOCAL_DIRS in YARN).
Spark context Web UI available at http://jupyter-spark:4040
Spark context available as 'sc' (master = local[*], app id = local-1625358424217).
Spark session available as 'spark'.
Welcome to

```
  /---  
 _\ \ / - - - - - / \--  
/---/ .--/\_/_/_/_/_/_/_/_ version 2.4.1  
/_/
```

Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_201)
Type in expressions to have them evaluated.
Type :help for more information.

scala> |

```
scala> import org.apache.kafka.common.serialization.StringDeserializer
import org.apache.kafka.common.serialization.StringDeserializer

scala> import org.apache.spark.streaming.kafka010._
import org.apache.spark.streaming.kafka010._

scala> import org.apache.spark.streaming.kafka010.LocationStrategies.PreferConsistent
import org.apache.spark.streaming.kafka010.LocationStrategies.PreferConsistent

scala> import org.apache.spark.streaming.kafka010.ConsumerStrategies.Subscribe
import org.apache.spark.streaming.kafka010.ConsumerStrategies.Subscribe

scala> |
```

```
scala> val kafkaParams = Map[String, Object](
  |   "bootstrap.servers" -> "kafka:9092",
  |   "key.deserializer" -> classOf[StringDeserializer],
  |   "value.deserializer" -> classOf[StringDeserializer],
  |   "group.id" -> "aplicacao1",
  |   "auto.offset.reset" -> "earliest",
  |   "enable.auto.commit" -> (false: java.lang.Boolean)
  | )
kafkaParams: scala.collection.immutable.Map[String,Object] = Map(key.deserializer -> class org.apache.kafka.common.serialization.StringDeserializer,
  auto.offset.reset -> earliest, group.id -> aplicacao1, bootstrap.servers -> kafka:9092, enable.auto.commit -> false, value.deserializer -> class or
g.apache.kafka.common.serialization.StringDeserializer)

scala> |
```

Configurando o Spark Streaming Context para 5 segundos

```
scala> import org.apache.spark.streaming.{StreamingContext, Seconds}
import org.apache.spark.streaming.{StreamingContext, Seconds}

scala> val ssc = new StreamingContext(sc, Seconds(5))
ssc: org.apache.spark.streaming.StreamingContext = org.apache.spark.streaming.StreamingContext@5503e208

scala> sc
res0: org.apache.spark.SparkContext = org.apache.spark.SparkContext@648e25f2

scala> |
```

```
scala> val topic = Array("topic-spark")
topic: Array[String] = Array(topic-spark)

scala> val stream = KafkaUtils.createDirectStream[String, String](
  |   ssc,
  |   PreferConsistent,
  |   Subscribe[String, String](topic, kafkaParams)
  | )
21/07/04 00:47:08 WARN kafka010.KafkaUtils: overriding enable.auto.commit to false for executor
21/07/04 00:47:08 WARN kafka010.KafkaUtils: overriding auto.offset.reset to none for executor
21/07/04 00:47:08 WARN kafka010.KafkaUtils: overriding executor group.id to spark-executor-aplicacao1
21/07/04 00:47:08 WARN kafka010.KafkaUtils: overriding receive.buffer.bytes to 65536 see KAFKA-3135
stream: org.apache.spark.streaming.dstream.InputDStream[org.apache.kafka.clients.consumer.ConsumerRecord[String,String]] = org.apache.spark.streamin
g.kafka010.DirectKafkaInputDStream@52b7d2d3

scala> |
```

2. Visualizar o tópico com as seguintes informações

- Nome do tópico
- Partição
- Valor

```
scala> val info_stream = stream.map(record => (  
  | record.to  
toString  topic  
  | record.topic,  
  | record.partition,  
  | record.value  
  | ))  
info_stream: org.apache.spark.streaming.dstream.DStream[(String, Int, String)] = org.apache.spark.streaming.dstream.MappedDStream@3bf66f94  
scala> |
```

3. Salvar o tópico no diretório `hdfs://namenode:8020/user/<nome>/kafka/dstream`

```
scala> info_stream.print()  
  
scala> info_stream.saveAsTextFiles("hdfs://namenode:8020/user/feliciani/kafka/dsstream")  
  
scala> |
```

LEU OS DADOS DO KAFKA

```
scala> ssc.start()

-----
Time: 1625360950000 ms
-----
(topic-spark,0,dfd)
(topic-spark,0,)
(topic-spark,0,sdfsd)
(topic-spark,0,sdf)
(topic-spark,0,sdf)
(topic-spark,0,sdf)
(topic-spark,0,sdf)
(topic-spark,0,sdf)
(topic-spark,0,Marcelo)
(topic-spark,0,Feliciani)
...
```

```
-----
Time: 1625360955000 ms
-----
```

```
-----
Time: 1625360960000 ms
-----
```

```
-----
Time: 1625360965000 ms
-----
```

TELA DE CIMA É O KAFKA PRODUCER PRODUZINDO MENSAGEM E A TELA DE BAIXO É SPARK PROCESSANDO NO SCALA

```
Windows PowerShell x feliciani@LAPTOP-V176DRSL: /r x + v
bash-4.4# kafka-console-producer.sh --broker-list kafka:9092 --topic topic-spark
>Marcelo
>Feliciani
>Tela de cima é o PRODUCER
>Tela de baixo é o CONSUMER
>MSG 1
>MSG 2
>MSG 3
>CHEGANDO INFORMACAO NO SPARK
TESTE DO SPARK[

Cursos | [topic-spark-0-producer]
Windows PowerShell x feliciani@LAPTOP-V176DRSL: /r x feliciani@LAPTOP-V176DRSL: /r x + v
-----
Time: 1625361120000 ms
-----
(topic-spark,0,CHEGANDO INFORMACAO NO SPARK)
-----
Time: 1625361125000 ms
-----
Time: 1625361130000 ms
-----
```



```
Windows PowerShell | feliciani@LAPTOP-V176DRSL: /r | + | -
bash-4.4# kafka-console-producer.sh --broker-list kafka:9092 --topic topic-spark
>Marcelo
>Feliciani
>Tela de cima é o PRODUCER
>Tela de baixo é o CONSUMER
>MSG 1
>MSG 2
>MSG 3
>CHEGANDO INFORMACAO NO SPARK
TESTE DO SPARK[
>MARCELO FELICIANI, SÁBADO DE ESTUDOS
>|
```

```
Windows PowerShell | feliciani@LAPTOP-V176DRSL: /r | feliciani@LAPTOP-V176DRSL: /r | + | -
-----
Time: 1625361245000 ms
-----

-----
Time: 1625361250000 ms
-----
(topic-spark,0,MARCELO FELICIANI, SÁBADO DE ESTUDOS)
-----
Time: 1625361255000 ms
-----
```


ARQUIVOS GRAVADOS NO HDFS

[illegible]

ENTREI NUM ARQUIVO PARA VER A PARTIÇÃO CRIADA

```
root@jupyter-spark:/# hdfs dfs -ls /user/feliciani/kafka/dsstream-1625360950000
Found 2 items
-rw-r--r--    2 root supergroup          0 2021-07-04 01:09 /user/feliciani/kafka/dsstream-1625360950000/_SUCCESS
-rw-r--r--    2 root supergroup      641 2021-07-04 01:09 /user/feliciani/kafka/dsstream-1625360950000/part-000000
root@jupyter-spark:/# |
```

INFORMAÇÕES SALVAS NO ARQUIVO

```
root@jupyter-spark:/# hdfs dfs -cat /user/feliciani/kafka/dsstream-1625360950000/part-000000
(topic-spark,0,dfd)
(topic-spark,0,)
(topic-spark,0,sdfsd)
(topic-spark,0,sdf)
(topic-spark,0,sdf)
(topic-spark,0,sdf)
(topic-spark,0,sdf)
(topic-spark,0,sdf)
(topic-spark,0,sdf)
(topic-spark,0,Marcelo)
(topic-spark,0,Feliciani)
(topic-spark,0,tela)
(topic-spark,0,tela)
(topic-spark,0,tela do producer)
(topic-spark,0,Marcelo)
(topic-spark,0,Feliciani)
(topic-spark,0,Tela de cima é o PRODUCER)
(topic-spark,0,Tela de baixo é o CONSUMER)
(topic-spark,0,)
(topic-spark,0,Marcelo)
(topic-spark,0,Feliciani)
(topic-spark,0,Tela de cima é o PROCUCER)
(topic-spark,0,Tela de baixo é o CONSUMER)
(topic-spark,0,MSG 1)
(topic-spark,0,MSG 2)
(topic-spark,0,MSG 3)
root@jupyter-spark:/# |
```