

## KAFKA E SPARK STREAMING

### KAFKA

#### 1. Preparação do ambiente no Kafka

##### a) Criar o tópico “topic-kvspark” com 2 partições e o fator de replicação = 1

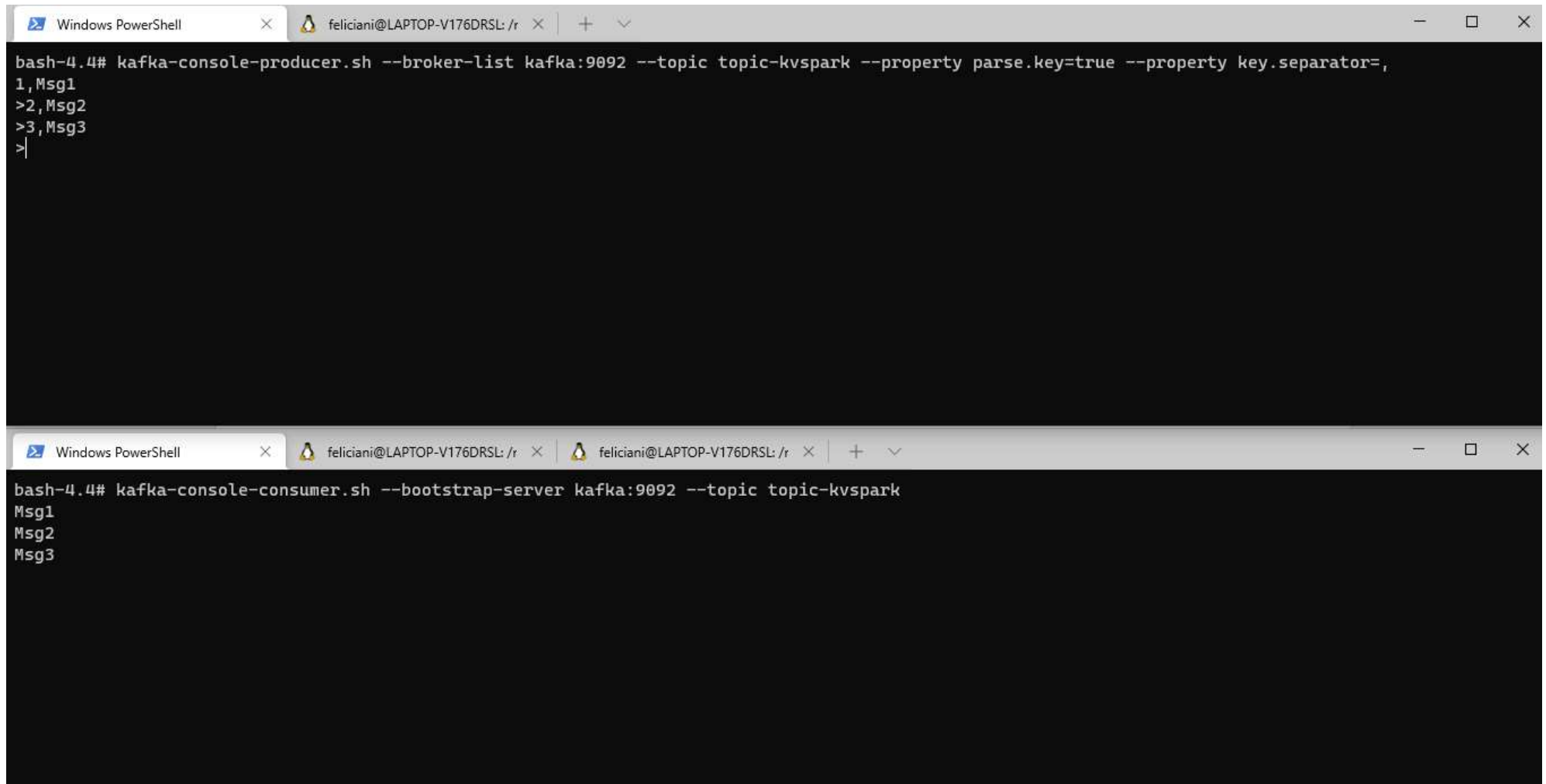
```
PS E:\projetos\docker-spark\spark> wsl -l -v
NAME                STATE      VERSION
* docker-desktop-data Running    2
  Ubuntu-20.04       Running    2
  docker-desktop     Running    2
PS E:\projetos\docker-spark\spark> docker exec -it kafka bash
bash-4.4# kafka-topics.sh --bootstrap-server kafka:9092 --create --topic topic-kvspark --partitions 2 --replication-factor 1
bash-4.4# kafka-topics.sh --bootstrap-server kafka:9092 --list
--topic-spark
__consumer_offsets
topic-kvspark
topic-spark
bash-4.4# kafka-topics.sh --bootstrap-server kafka:9092 --describe topic-kvspark
Topic:topic-kvspark    PartitionCount:2      ReplicationFactor:1    Configs:segment.bytes=1073741824
      Topic: topic-kvspark Partition: 0    Leader: 1001    Replicas: 1001    Isr: 1001
      Topic: topic-kvspark Partition: 1    Leader: 1001    Replicas: 1001    Isr: 1001
```

##### b) Criar um consumidor no Kafka para ler o “topic-kvspark”

```
PS E:\projetos\docker-spark\spark>
PS E:\projetos\docker-spark\spark> wsl -l -v
NAME                STATE      VERSION
* docker-desktop-data Running    2
  Ubuntu-20.04       Running    2
  docker-desktop     Running    2
PS E:\projetos\docker-spark\spark> docker exec -it kafka bash
bash-4.4# kafka-console-consumer.sh --bootstrap-server kafka:9092 --topic topic-kvspark
|
```

c) Inserir as seguintes mensagens no tópico (Chave, Valor):

Msg1, Msg2, Msg3



The image displays two terminal windows from a Windows PowerShell environment. The top window shows the execution of the `kafka-console-producer.sh` command with various options to send messages to a Kafka topic. The bottom window shows the execution of the `kafka-console-consumer.sh` command to receive those messages.

```
Windows PowerShell
feliciani@LAPTOP-V176DRSL: /r

bash-4.4# kafka-console-producer.sh --broker-list kafka:9092 --topic topic-kvspark --property parse.key=true --property key.separator=,
1,Msg1
>2,Msg2
>3,Msg3
>|

Windows PowerShell
feliciani@LAPTOP-V176DRSL: /r feliciani@LAPTOP-V176DRSL: /r

bash-4.4# kafka-console-consumer.sh --bootstrap-server kafka:9092 --topic topic-kvspark
Msg1
Msg2
Msg3
```

## SPARK Streaming utilizando a linguagem Scala

### 1. Criar um consumidor em Scala usando Spark Streaming para ler o “topic-kvspark” no cluster Kafka ”kafka:9092”

#### Integração do Spark Streaming com o Kafka

```
Windows PowerShell x feliciani@LAPTOP-V176DRSL: /r x feliciani@LAPTOP-V176DRSL: /r x + v
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-f02ca96d-b58f-4d9b-8a36-a8439ec2de77;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
:: resolution report :: resolve 3593ms :: artifacts dl 83ms
  :: 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      |
|               | number| search|dwnlded|evicted|| number|dwnlded|
|-----|-----|-----|-----|-----|
| default      | 6    | 0    | 0    | 0    || 6    | 0    |
|-----|-----|-----|-----|-----|
:: retrieving :: org.apache.spark#spark-submit-parent-f02ca96d-b58f-4d9b-8a36-a8439ec2de77
  confs: [default]
  0 artifacts copied, 6 already retrieved (0kB/50ms)
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
21/07/04 15:48:37 WARN spark.SparkConf: Note that spark.local.dir will be overridden by the value set by the cluster manager (via SPARK_LOCAL_DIRS in
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-1625413729994).
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> |
```

## Importando as bibliotecas para criar o Direct Stream

```
  /---/  /---/  /---/  /---/  /---/  /---/  /---/  /---/  /---/  /---/
 _\  \  _\  \  _\  \  _\  \  _\  \  _\  \  _\  \  _\  \  _\  \
/---/  /---/  /---/  /---/  /---/  /---/  /---/  /---/  /---/  /---/
  /_/_/  /_/_/  /_/_/  /_/_/  /_/_/  /_/_/  /_/_/  /_/_/  /_/_/  /_/_/
                                     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> import org.apache.kafka.clients.consumer.ConsumerRecord
import org.apache.kafka.clients.consumer.ConsumerRecord
```

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

```
scala> |
```

## Parâmetros

```
scala> val kafkaParams = Map[String, Object](
  |   "bootstrap.servers" -> "kafka:9092",
  |   "key.deserializer" -> classOf[StringDeserializer],
  |   "value.deserializer" -> classOf[StringDeserializer],
  |   "group.id" -> "aplicacao2",
  |   "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 -> aplicacao2, bootstrap.servers -> kafka:9092, enable.auto.commit -> false, value.deserializer -> class or
g.apache.kafka.common.serialization.StringDeserializer)
```

## Captura dos Streams do topic do Kafka será a cada 5 segundos

```
scala> val ssc = new StreamingContext(sc, Seconds(5))
ssc: org.apache.spark.streaming.StreamingContext = org.apache.spark.streaming.StreamingContext@13f6395d

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

## Variável para o Direct Stream

```
scala> val stream = KafkaUtils.createDirectStream[String, String](
  |   ssc,
  |   PreferConsistent,
  |   Subscribe[String, String](topic, kafkaParams)
  | )
21/07/04 16:19:09 WARN kafka010.KafkaUtils: overriding enable.auto.commit to false for executor
21/07/04 16:19:09 WARN kafka010.KafkaUtils: overriding auto.offset.reset to none for executor
21/07/04 16:19:09 WARN kafka010.KafkaUtils: overriding executor group.id to spark-executor-aplicacao2
21/07/04 16:19:09 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@ad7d724

scala> |
```

## 2. Visualizar o tópico com as seguintes informações

- Nome do tópico
- Partição
- Chave
- Valor

### Criada variável para receber os dados do Kafka

```
scala> val info_stream = stream.map(record => (  
  | record.topic,  
  | record.partition,  
  | record.key,  
  | record.value  
  | ))  
info_stream: org.apache.spark.streaming.dstream.DStream[(String, Int, String, String)] = org.apache.spark.streaming.dstream.MappedDStream@1e7544bc  
  
scala> info_stream.print()  
  
scala> |
```

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

```
scala> info_stream.saveAsTextFiles("/user/feliciani/kafka/dstreamdv")
```

### Recebendo os dados de Stream vindos do Kafka

```
scala> ssc.start()  
  
-----  
Time: 1625418850000 ms  
-----  
(topic-kvspark,1,1,Msg1)  
(topic-kvspark,1,1,Msg1)  
(topic-kvspark,1,3,Msg3)  
(topic-kvspark,1,,Msg1)  
(topic-kvspark,1,3,Msg3)  
(topic-kvspark,0,2,Msg2)  
(topic-kvspark,0,2,Msg2)  
  
|
```



Por exemplo:

A Msg 1 veio da partição 1, chave 1

A Msg 3 veio da partição 1, chave 3

A Msg 2 veio da partição 0, chave 2

Novas mensagens enviada pelo Kafka

```
-----  
Time: 1625419300000 ms  
-----
```

```
(topic-kvspark,1,1,Marcelo)  
(topic-kvspark,1,3,Estudo de domingo 04/07/2021)  
(topic-kvspark,1,1,Marcelo)  
(topic-kvspark,1,3,Estudo de domingo 04/07/2021)  
(topic-kvspark,0,2,Feliciani)  
(topic-kvspark,0,2,Feliciani)
```

```
-----  
Time: 1625419305000 ms  
-----
```

```
(topic-kvspark,1,1,Marcelo)  
(topic-kvspark,1,3,Estudo de domingo 04/07/2021)  
(topic-kvspark,0,2,Feliciani)
```

Marcelo na Partição 1, chave 1

Estudo de domingo 04/07/2021 na Partição 1, chave 3

Feliciani na Partição 0, chave 2

ESTÁ EXISTINDO UM BALANCEAMENTO DAS MENSAGENS ENVIADAS PARA AS PARTIÇÕES.

AS INFORMAÇÕES DAS CHAVES ESTARÃO SEMPRE NAS MESMAS PARTIÇÕES

## Verificando os arquivos criados no HDFS

### Print dos arquivos do primeiro exercício do dsstream

```
root@jupyter-spark:/# hdfs dfs -ls /user/feliciani/kafka
Found 256 items
drwxr-xr-x - root supergroup 0 2021-07-04 01:09 /user/feliciani/kafka/dsstream-1625360950000
drwxr-xr-x - root supergroup 0 2021-07-04 01:09 /user/feliciani/kafka/dsstream-1625360955000
drwxr-xr-x - root supergroup 0 2021-07-04 01:09 /user/feliciani/kafka/dsstream-1625360960000
drwxr-xr-x - root supergroup 0 2021-07-04 01:09 /user/feliciani/kafka/dsstream-1625360965000
drwxr-xr-x - root supergroup 0 2021-07-04 01:09 /user/feliciani/kafka/dsstream-1625360970000
drwxr-xr-x - root supergroup 0 2021-07-04 01:09 /user/feliciani/kafka/dsstream-1625360975000
drwxr-xr-x - root supergroup 0 2021-07-04 01:09 /user/feliciani/kafka/dsstream-1625360980000
drwxr-xr-x - root supergroup 0 2021-07-04 01:09 /user/feliciani/kafka/dsstream-1625360985000
drwxr-xr-x - root supergroup 0 2021-07-04 01:09 /user/feliciani/kafka/dsstream-1625360990000
drwxr-xr-x - root supergroup 0 2021-07-04 01:09 /user/feliciani/kafka/dsstream-1625360995000
drwxr-xr-x - root supergroup 0 2021-07-04 01:10 /user/feliciani/kafka/dsstream-1625361000000
drwxr-xr-x - root supergroup 0 2021-07-04 01:10 /user/feliciani/kafka/dsstream-1625361005000
drwxr-xr-x - root supergroup 0 2021-07-04 01:10 /user/feliciani/kafka/dsstream-1625361010000
drwxr-xr-x - root supergroup 0 2021-07-04 01:10 /user/feliciani/kafka/dsstream-1625361015000
drwxr-xr-x - root supergroup 0 2021-07-04 01:10 /user/feliciani/kafka/dsstream-1625361020000
drwxr-xr-x - root supergroup 0 2021-07-04 01:10 /user/feliciani/kafka/dsstream-1625361025000
drwxr-xr-x - root supergroup 0 2021-07-04 01:10 /user/feliciani/kafka/dsstream-1625361030000
drwxr-xr-x - root supergroup 0 2021-07-04 01:10 /user/feliciani/kafka/dsstream-1625361035000
drwxr-xr-x - root supergroup 0 2021-07-04 01:10 /user/feliciani/kafka/dsstream-1625361040000
drwxr-xr-x - root supergroup 0 2021-07-04 01:10 /user/feliciani/kafka/dsstream-1625361045000
drwxr-xr-x - root supergroup 0 2021-07-04 01:10 /user/feliciani/kafka/dsstream-1625361050000
drwxr-xr-x - root supergroup 0 2021-07-04 01:10 /user/feliciani/kafka/dsstream-1625361055000
drwxr-xr-x - root supergroup 0 2021-07-04 01:11 /user/feliciani/kafka/dsstream-1625361060000
drwxr-xr-x - root supergroup 0 2021-07-04 01:11 /user/feliciani/kafka/dsstream-1625361065000
drwxr-xr-x - root supergroup 0 2021-07-04 01:11 /user/feliciani/kafka/dsstream-1625361070000
drwxr-xr-x - root supergroup 0 2021-07-04 01:11 /user/feliciani/kafka/dsstream-1625361075000
drwxr-xr-x - root supergroup 0 2021-07-04 01:11 /user/feliciani/kafka/dsstream-1625361080000
drwxr-xr-x - root supergroup 0 2021-07-04 01:11 /user/feliciani/kafka/dsstream-1625361085000
drwxr-xr-x - root supergroup 0 2021-07-04 01:11 /user/feliciani/kafka/dsstream-1625361090000
drwxr-xr-x - root supergroup 0 2021-07-04 01:11 /user/feliciani/kafka/dsstream-1625361095000
drwxr-xr-x - root supergroup 0 2021-07-04 01:11 /user/feliciani/kafka/dsstream-1625361100000
drwxr-xr-x - root supergroup 0 2021-07-04 01:11 /user/feliciani/kafka/dsstream-1625361105000
```



## Print do 2º exercício dsstreamdv

[illegible]

## Partições criadas e o conteúdo do Streaming originado no Kafka

```
root@jupyter-spark:/# hdfs dfs -ls /user/feliciani/kafka/dstreamdv-1625418850000
Found 3 items
-rw-r--r--  2 root supergroup          0 2021-07-04 17:16 /user/feliciani/kafka/dstreamdv-1625418850000/_SUCCESS
-rw-r--r--  2 root supergroup    124 2021-07-04 17:16 /user/feliciani/kafka/dstreamdv-1625418850000/part-00000
-rw-r--r--  2 root supergroup     50 2021-07-04 17:16 /user/feliciani/kafka/dstreamdv-1625418850000/part-00001
root@jupyter-spark:/# hdfs dfs -cat /user/feliciani/kafka/dstreamdv-1625418850000/part-00000
(topic-kvspark,1,1,Msg1)
(topic-kvspark,1,1,Msg1)
(topic-kvspark,1,3,Msg3)
(topic-kvspark,1,,Msg1)
(topic-kvspark,1,3,Msg3)
root@jupyter-spark:/# hdfs dfs -cat /user/feliciani/kafka/dstreamdv-1625418850000/part-00001
(topic-kvspark,0,2,Msg2)
(topic-kvspark,0,2,Msg2)
root@jupyter-spark:/# |
```