UNIVERSIDADE DE BRASÍLIA
Faculdade do Gama

Programação em Sistemas Paralelos e Distribuídos
Conectando um contador de palavra com o Kibana

Ítalo Alves Guimarães - 18/0113666
Gabriela da Gama Pivetta - 18/0052845
Murilo Gomes de Souza - 18/0025601
Pedro Rodrigues Pereira - 17/0062686

Brasília, DF
2023

1. Introdução

    Objetivos do Laboratório

    Este laboratório tem como objetivo fazer uma integração de um contador de palavras implementado usando spark e kafka com o kibana e o elastic search.

2. Passo a Passo
    - Primeiro foram criados dois tópicos no kafka afim de ter a entrada das palavras em um e a saída junto com a contagem das palavras em outro.

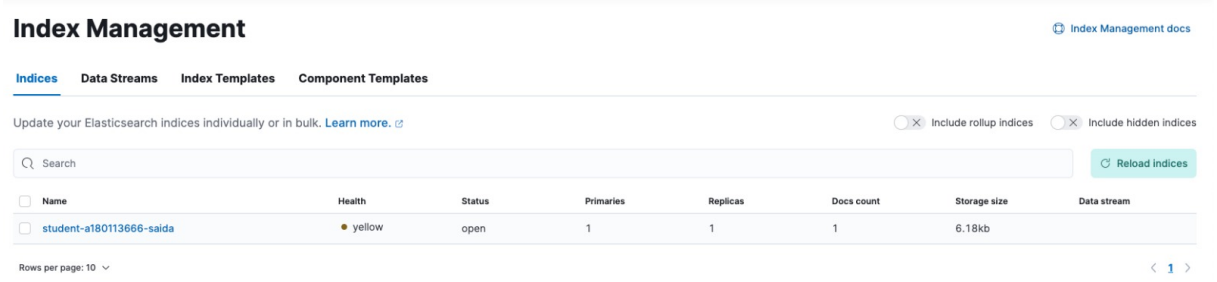
```
a180113666@chococino:~$ kafka-console-consumer.sh -topic student-a180113666-saida --bootstrap-server cm3:9092
olá
```

```
a180113666@chococino:~$ kafka-console-producer.sh -topic student-a180113666-saida --bootstrap-server cm3:9092
>olá
>
```

    - Depois foi configurada a conexão entre o kafka e o ElasticSearch/Kibana
    - Logo em seguida foi desenvolvido o código no notebook:

```
In [13]: from pyspark.sql import SparkSession

In [14]: SPARK_MASTER="spark://cm1:7077"
         APP_NAME="Exemplo Spark, Kafka e ElasticSearch"

         KAFKA_HOST="cm3:9092"

         KAFKA_TOPIC_IN="student-a180113666-entrada"
         KAFKA_TOPIC_OUT="student-a180113666-saida"

         INTERVAL="10 seconds"

In [15]: spark = SparkSession.builder.master(SPARK_MASTER).appName(APP_NAME).getOrCreate()

In [16]: messages = spark.readStream.format("kafka").option("kafka.bootstrap.servers", KAFKA_HOST) \
             .option("subscribe", KAFKA_TOPIC_IN) \
             .option('includeTimestamp', 'true').load()

In [17]: from pyspark.sql.functions import explode, split, col, upper, window, to_json, struct, lit

In [18]: words = messages \
             .select(
                 explode(split(col("value"), "\s+")).alias("word"),
                 messages.timestamp
             ).select (
                 upper(col("word")).alias('word'),
                 col("timestamp")
             )

In [19]: wordCounts = words.withWatermark("timestamp", INTERVAL) \
             .groupBy(
                 window(words.timestamp, INTERVAL, INTERVAL),
                 "word"
             ).count()

In [20]: wordCountsJson = wordCounts.select(
             lit('1').alias('key'),
             to_json(struct("word", "count")).alias('value')
         )

In [21]: wk = wordCountsJson.writeStream.outputMode("update").format("kafka").option("kafka.bootstrap.servers", KAFKA_HOST) \
             .option('topic', KAFKA_TOPIC_OUT).option('checkpointLocation', '/tmp/spark-a180113666') \
             .trigger(processingTime=INTERVAL) \
             .start()
         23/05/09 22:21:47 WARN ResolveWriteToStream: spark.sql.adaptive.enabled is not supported in streaming DataFrames/Data
         sets and will be disabled.
         23/05/09 22:21:47 WARN StreamingQueryManager: Stopping existing streaming query [id=dd37c952-0ad4-4510-b15b-0c45bfc3c
         e99, runId=3c22afla-16b4-40c9-9fc7-04933aa64034], as a new run is being started.

In [ ]:
```

3. Depois foi enviado um json pelo diálogo produtor e o index apareceu no elastic



4. Depois foi criado o gráfico: