Twitter Streaming Analyze usando Java 8, Spark Streaming, Kafka e Cassandra

Um projeto de demonstração usando Spark Streaming para analisar hashtags populares do Twitter. Os dados vêm da fonte da Twitter Streaming API e são fornecidos ao Kafka. O consumidor com.twitter.producer.service recebe dados do Kafka e, em seguida, os processa em um fluxo usando o Spark Streaming.

1. Estrutura

1.1. É uma estrutura de um projeto multi-módulos

```
1)  shangrila-producer - Realiza a consulta a api do Twitter e funciona
 como produtor de topico no Kafka
```

```
2)  shangrila-producer - Tem a responsabilidade de implementar as regras
 de negocio atraves do spark streaming e salvar no Banco de Dados
 (Cassandra). Também tem a responsabilidade de ser o consumidor dos
 topicos do Kafka.
```

2. Desenho da Arquitetura da Solução

O desenho e implementação de uma arquitetura distribuída, que realize a integração com o Twitter que seja tolerante a falhas e escalável horizontalmente, que exponha através de uma aplicação web as informações sumarizadas e descritas no Case de integração.

2.1. Requisitos

- Apache Maven 3.x
- JVM 8

- Docker machine
- Registratar um aplicativo no Twtter.
- Em seguida, processa em um fluxo usando Sparj Streaming: [Como criar uma aplicação no Twitter.](http://docs.inboundnow.com/guide/create-twitter-application/).

Na próxima etapa irei demonstrar como configurar o ambiente para que executar nossa aplicação.

2.2. Guia de início Rapído

Neste quie rápido mostrarei como configurar sua máquina para executar nosso aplicativo.

Apache Spark no Windows

- Faça o download do Spark em https://spark.apache.org/downloads.html ![Alt text](images/downloads-apache-spark.png)

    1. Descompacte o arquivo spark-2.4.1-bin-hadoop2.7.tgz em um diretório. ![Alt text](images/sparkinstallation.png)
    2. Agora defina variáveis de ambiente SPARK_HOME = C:\Installations \spark-2.4.1-bin-hadoop2.7

![Alt text](images/spark_env.png)

```
~>  SPARK_HOME = C:\Installations \spark-2.4.1-bin-hadoop2.7
```

- Instalando o binário winutils

    1. Faça o download do [winutils.exe](https://github.com/steveloughran/winutils/raw/master/hadoop-2.7.1/bin/winutils.exe) do Hadoop 2.7 e coloque-o em um diretório C: \ Installations \ Hadoop \ bin
    2. Agora defina variáveis de ambiente HADOOP_HOME = C: \ Installations \ Hadoop.

![Alt text](images/hadoop_env.png)

```
~>  HADOOP_HOME = C:\Installations\Hadoop
```

Agora inicie o shell do Windows; você pode receber alguns avisos, que você pode ignorar por enquanto.

![Alt text](images/spark_install_sucess.png)

1. Mude a configuração do Twitter no arquivo `\producer\src\main\resources \application.yml`colocando suas credencias do Twtter, client Id e Secret Id.
2. Execute a imagem kafka usando o docker-compose (lembre-se de que a imagem kafka também precisa extrair o zookeper):

```
~>  docker-compose -f shangrila-producer/src/main/docker/kafka-docker-
compose.yml up -d
```

Execute a imagem do Cassandra usando o docker-compose.

```
~>  docker-compose -f shangrila-consumer\src\main\docker\cassandra.yml
 up -d
```

Verifique se o Cassandra, ZooKeeper e o Kafka estão em execução (no prompt de comando)

```
~> docker ps
```

1. Execute o poducer e o aplicativo do consumidor com:

```
~> mvn spring-boot:run
```

2.3. Esta documentação esta em desenvolvimento

2.4. Referências

- [Instalação do Apache Spark Windows](https://dzone.com/articles/working-on-apache-spark-on-windows)
- [Spring for Apache Kafka](https://projects.spring.io/spring-kafka/)

- [Spring Social Twitter](http://projects.spring.io/spring-social-twitter/)
- [Spark Overview](http://spark.apache.org/docs/latest/)
- [Apache Kafka Documentation](http://kafka.apache.org/documentation.html)
- [Big Data Processing with Apache Spark - Part 3: Spark Streaming](https://www.infoq.com/articles/apache-spark-streaming)
- [Spring Kafka - Embedded Unit Test Example](https://www.codenotfound.com/spring-kafka-embedded-unit-test-example.html)
- https://github.com/FoxtrotSystems