Lab 7: Spark Streaming WordCount

About This Lab

Objective:	Create a Streaming application that outputs all words said in a `Dstream`, utilize the `nc` command to simulate a data source
File locations:	No files
Successful outcome:	Output words from simulated source to screen You
Before you begin	should be logged in to your lab environment Spark
Related lesson:	Streaming

Lab Steps

Perform the following steps:

1 . Close the REPL

2 . Start a new REPL specifying the following information:

```
#pyspark --master local[2]
```

3 . Create a Spark Streaming application that performs a wordcount on a socket text stream

a. Import the Streaming library:

```
>>>from pyspark.streaming import StreamingContext
```

b. Create the streaming context, with a 5 second batch duration:

```
>>>ssc = StreamingContext(sc, 5)
```

c. Create the `Dstream` using `sandbox` and port 9999:

```
>>>inputDS = ssc.socketTextStream("VM-IPADDRESS",9999)
```

d. Transform the RDD to create a wordcount application, split on spaces:

```
>>>wc = inputDS.flatMap(lambda line: line.split(" ")).map(lambda word:
(word,1)).reduceByKey(lambda a,b: a+b)
```

e. Print out the output to the client:

```
>>>wc.pprint()
```

f. Set the log level to `ERROR` to avoid

clutter: `>>>sc.setLogLevel("ERROR")`

g. Start the streaming application:

```
>>>ssc.start()
```

Note
You will see an error when it starts, it's waiting for an input connection.

4 . In a new terminal, run the following command to start outputting data:

```
#nc -lkv 9999
```

a. Start typing words separated by space, press return occasionally to submit them

b. Look at the other terminal where the streaming application is running

c. While the application is running, navigate to the web UI in Firefox and explore the web
UI tabs:

`sandbox:4040`

d. To quit the streaming application, press `control-d`, `control-c` for the terminal
running NC.

Result

You have now successfully created and run a stateless application.