Word Count in Scala using Apache Spark Structured Streaming

This project demonstrates a simple Word Count application in Scala using Apache Spark Structured Streaming. It reads text data from a socket, splits the lines into words, counts them, and displays the word counts continuously on the console.

Prerequisites

- Scala installed
- Java (JDK 8 or later) installed
- Internet connection to download Apache Spark
- Terminal or command prompt access

Setup Steps

1. Download Spark 3.5.5

Click to Download: https://dlcdn.apache.org/spark/spark-3.5.5/spark-3.5.5-bin-hadoop3.tgz

Or run the command:

```
wget https://dlcdn.apache.org/spark/spark-3.5.5/spark-3.5.5-bin-hadoop3.tgz
```

2. Extract the downloaded archive

```
tar -xvzf spark-3.5.5-bin-hadoop3.tgz
```

3. Start a socket server (input terminal)

In one terminal window, run:

```
nc -lk 9999
```

This opens a TCP socket on port 9999 for text input.

4. Create the Scala file

Create a new file named wordCount.scala and paste the following code:

```
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions._

val spark = SparkSession
  .builder
  .appName("StructuredNetworkWordCount")
  .master("local[*]")
  .getOrCreate()

import spark.implicits._

val lines = spark.readStream
  .format("socket")
  .option("host", "localhost")
```

```
  .option("port", 9999)
  .load()

val words = lines.as[String].flatMap(_.split(" "))

val wordCounts = words.groupBy("value").count()

val query = wordCounts.writeStream
  .outputMode("complete")
  .format("console")
  .start()

query.awaitTermination()
```

5. Run the word count script using spark-shell

Open another terminal, navigate to Spark's bin directory:

```
cd spark-3.5.5-bin-hadoop3/bin
```

Then execute:

```
./spark-shell -i /path/to/your/wordCount.scala
```

Replace /path/to/your/wordCount.scala with the actual path to your file.

Try It Out

In the nc terminal (step 3), type:

```
hello world
spark spark word count
```

The Spark terminal will show a live word count output like this:

```
+-----+-----+
|value|count|
+-----+-----+
|hello|    1|
|world|    1|
|spark|    2|
|word |    1|
|count|    1|
+-----+-----+
```

Notes

- Use Ctrl+C to stop the streaming query.
- You can modify the logic (e.g., filter stopwords) inside the .flatMap() or .groupBy() stage.