

## 1 - Install Netcat

```
brew install netcat
```

## 2 - Run netcat command

```
netcat -l localhost -p 9999
```

## Spark Code For Word Count

```
import org.apache.spark.sql.{Encoders, SparkSession}

object SparkStreaming {
  def main(args: Array[String]): Unit = {

    val spark = SparkSession.builder.master("local").
      appName("SparkByExample")
      .getOrCreate()
    import spark.implicits._

    val streamingDf = spark.readStream.format("socket").option("host",
"localhost").option("port", 9999).load()

    val flatMapDF = streamingDf.flatMap(row => {
      row.getString(0).split(" ")
    })

    val countApp = flatMapDF.groupBy("value").count

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

    query.awaitTermination()

  }
}
```

Bu bölümde Structured Streaming ile IOT cihazından gelen örnek veriler üzerinde anlık analiz uygulamasını inceleyeceğiz

Özet olarak hava sıcaklığı ve rüzgar yönü bilgilerini gönderen cihazlardan gelen verileri anlık olarak analiz edeceğiz

Veri tipi

- Cihazın Seri numarası
- Gelen veri tipi (1 : hava sıcaklığı , 2 rüzgar ölçümü)
- Değer 1
  - Hava sıcaklığı için '+' veya '-'
  - Rüzgar ölçümü için rüzgar yönü : KD(Kuzey Doğu)
- Değer 2
  - Hava sıcaklığı değeri
  - Rüzgar ölçümü değeri 1-2

Örnek veri

SN123,1,+,15  
SN124,1,+,16  
SN125,1,+,16  
SN126,1,-,16  
SN123,2,GD,1  
SN125,2,GD,12

```

import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.types.{IntegerType, StringType, StructField,
StructType}

object WeatherIOTSparkStreaming {
  def main(args: Array[String]): Unit = {

    val spark = SparkSession.builder.master("local").
      appName("SparkByExample")
      .getOrCreate()
    spark.sparkContext.setLogLevel("ERROR")
    import spark.implicits._

    val schema = StructType(
      Array(
        StructField("sn",StringType,true),
        StructField("type",IntegerType,true),
        StructField("value1",StringType,true),
        StructField("value2", StringType, true)
      )
    )

    val df =
    spark.readStream.option("sep",",").schema(schema).csv("/Users/serkan/Desktop/Course
/Spark/data/stream")

    val result = df.select("sn", "value2").where("type == 1 and value1=='+' and
CAST(value2 AS INT) > 15")

    val query = result.writeStream.format("console").start()

    query.awaitTermination()

  }
}

```