Problem statement :

Count the number of words in a text data received from a server listening on a host and a port

-------------------------------------------------------------------------------------------------------------------------------------------------------------------

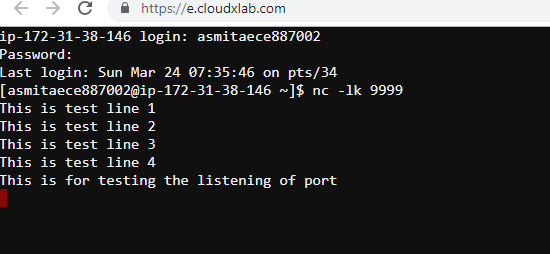
We will use spark streaming for the same .

Spark streaming code listens to a host and port . For creating a server , we use the nc utility , which is already installed for me

**nc** is the command which runs netcat, a simple **Unix** utility that reads and writes data across network connections, using the TCP or UDP protocol. It is designed to be a reliable "back-end" tool that can be used directly or driven by other programs and scripts.

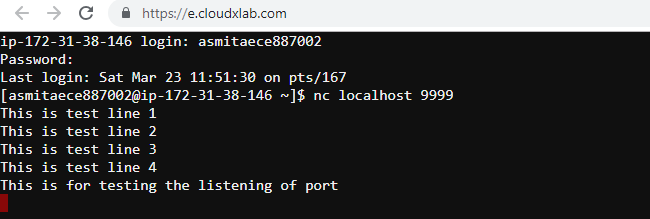
Server generates data on a host and a port . The server will work like a producer.

Create a producer . Connect to web console to port 9999. The following command listens at port 9999



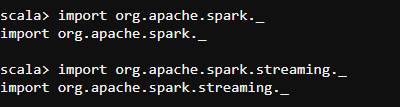
Nc-k ----Forces **nc** to stay listening for another connection after its current connection is completed. It is an error to use this option without the **-l**option.

We can test whether the server is listening to port 9999 . To do so , we can do the same by connecting another terminal and typing the following command



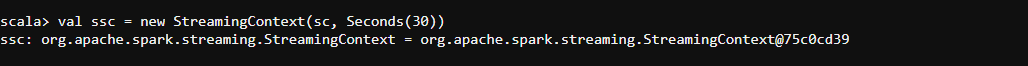
Now launch spark shell in the same shell as the client terminal

Import necessary libraries for spark – streaming

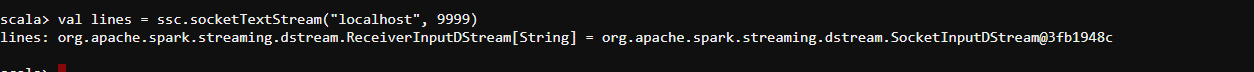


Next initialize the streaming context , by initializing ssc variable in the code

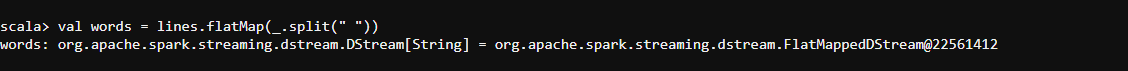
We will create a local Streaming contxt with batch interval of 30 seconds



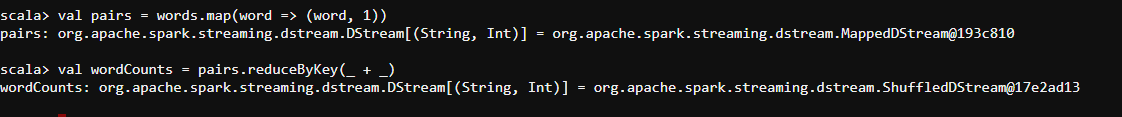
Create a DStream that will connect to hostname:port, like localhost:9999



Next , Split each line in each batch into words



Next , Count each word in each batch

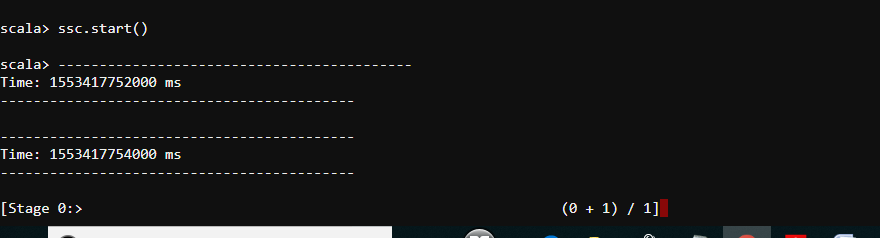




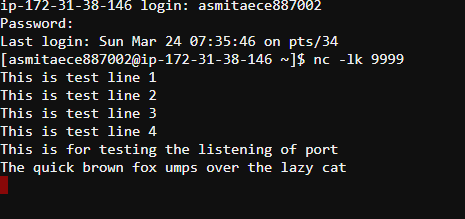
Start receiving data and processing it using Streamcontext.start()



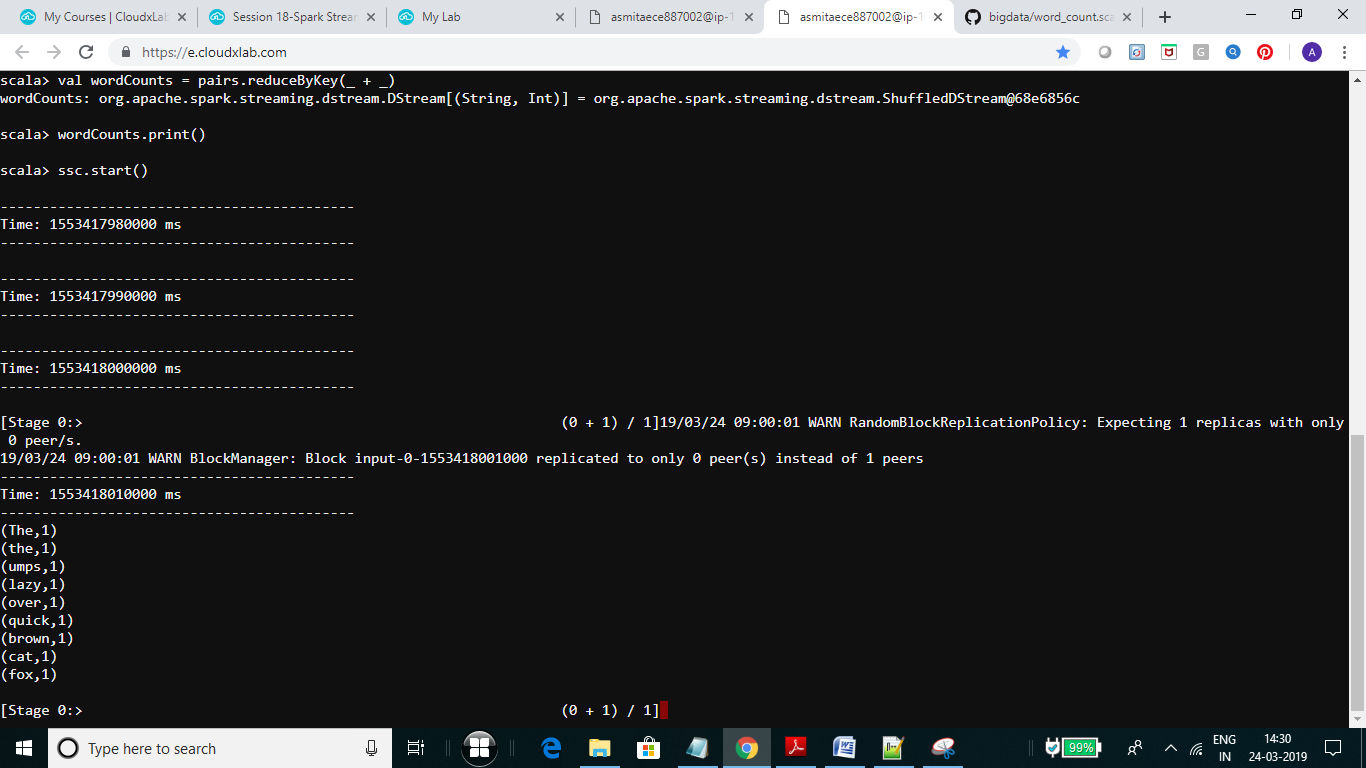
The streaming context started



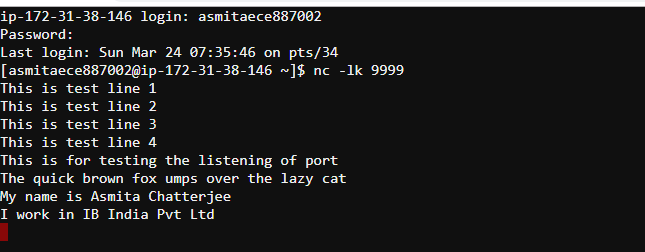
Now in the server terminal , enterthe



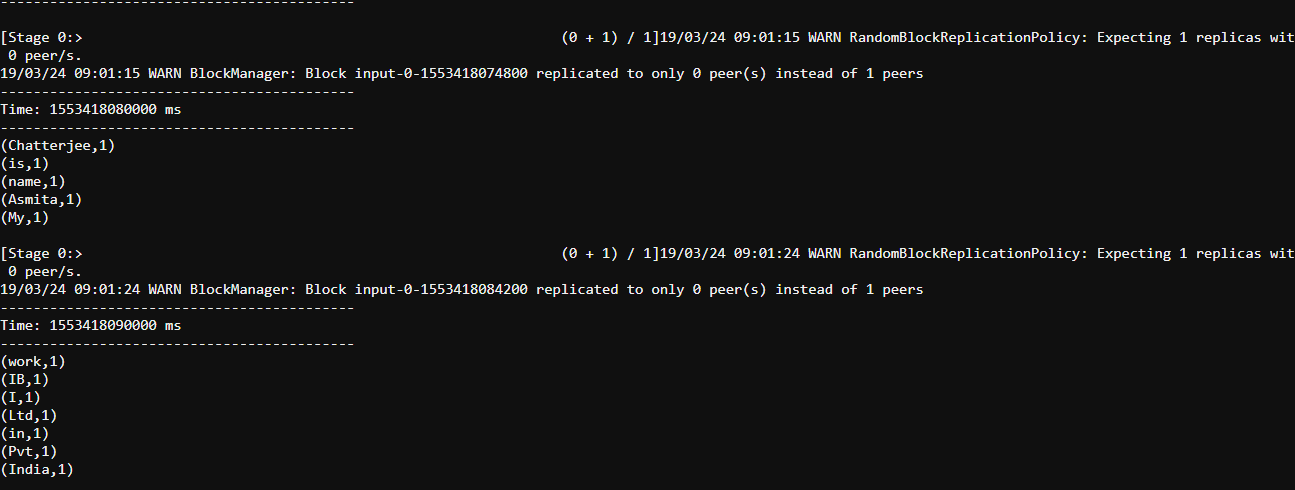
In the other terminal , we have the spark-shell running .



Entering another 2 lnes ,



we get the following in the terminal of the spark shell



To terminate the streaming read manually , type ssc.stop()