Atul Yadav

Practical 4 - Write a Spark code to Handle the Streaming c using RDD and Data frame.

START SERVICES – SPARK IN CLOUDERA MANAGER

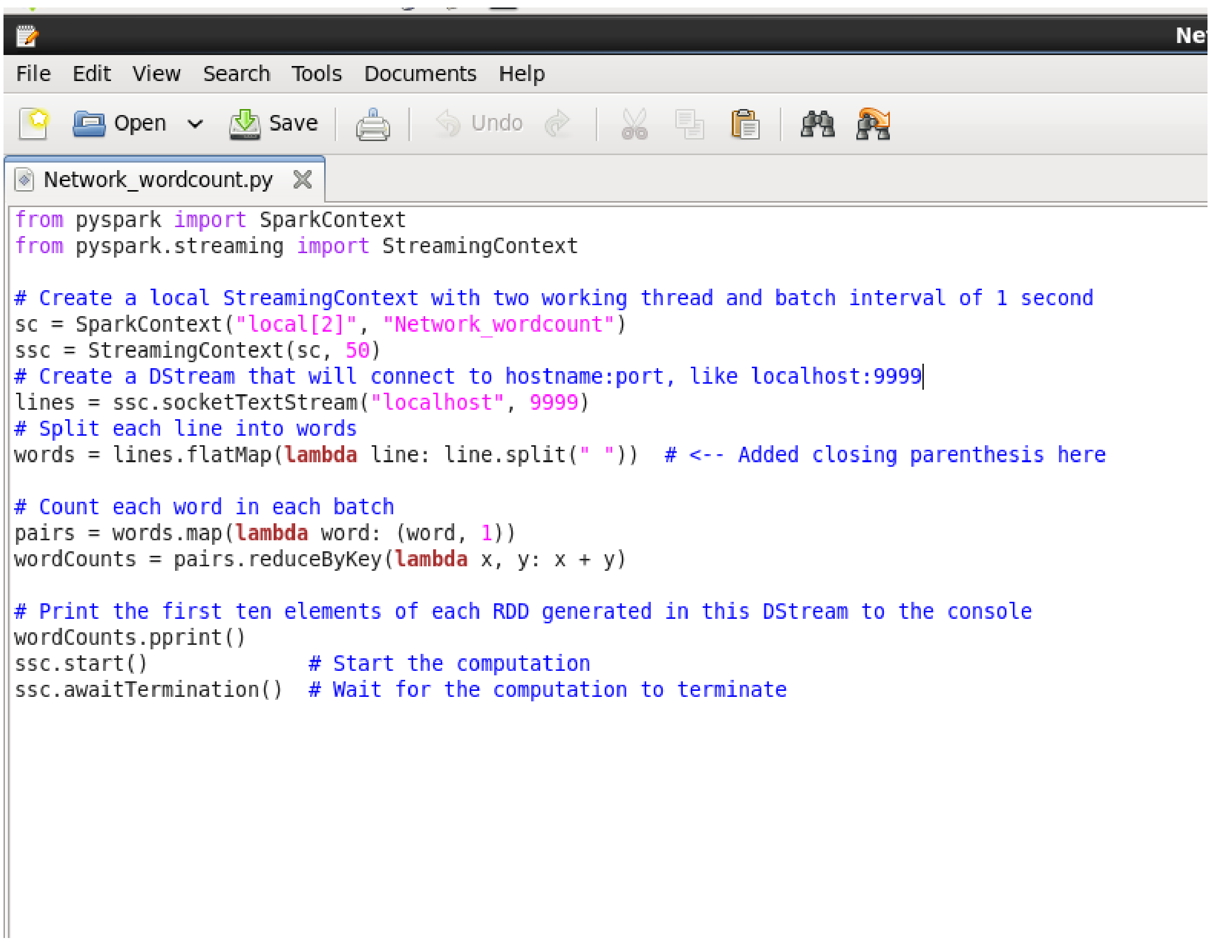
https://spark.apache.org/docs/latest/streaming-programming-guide.html

STEP 1 – Write Your Python Code:

Inside the text editor, write your Python code (Network_wordcount.py).

PS – save the file in Home/Cloudera/Documents

Below is the code



```
from pyspark import SparkContext
from pyspark.streaming import StreamingContext

# Create a local StreamingContext with two working thread and batch interval of 1 second
sc = SparkContext("local[2]", "Network_wordcount")
ssc = StreamingContext(sc, 50)
# Create a DStream that will connect to hostname:port, like localhost:9999
lines = ssc.socketTextStream("localhost", 9999)
# Split each line into words
words = lines.flatMap(lambda line: line.split(" "))  # <-- Added closing parenthesis here

# Count each word in each batch
pairs = words.map(lambda word: (word, 1))
wordCounts = pairs.reduceByKey(lambda x, y: x + y)

# Print the first ten elements of each RDD generated in this DStream to the console
wordCounts.pprint()
ssc.start()             # Start the computation
ssc.awaitTermination()  # Wait for the computation to terminate
```

Step 2: Save and Exit

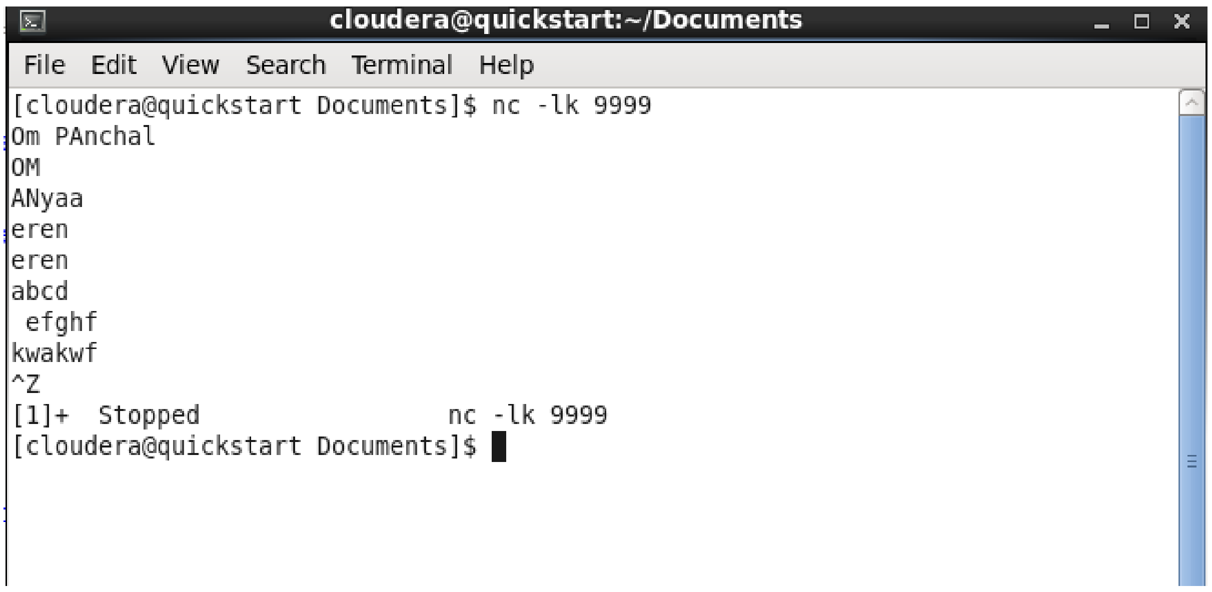
Save the file (in python extension) and exit the text editor.

Step 3: Open a New Terminal Window

Open a new terminal window or tab to perform the following steps simultaneously.

Step 4: Start Netcat

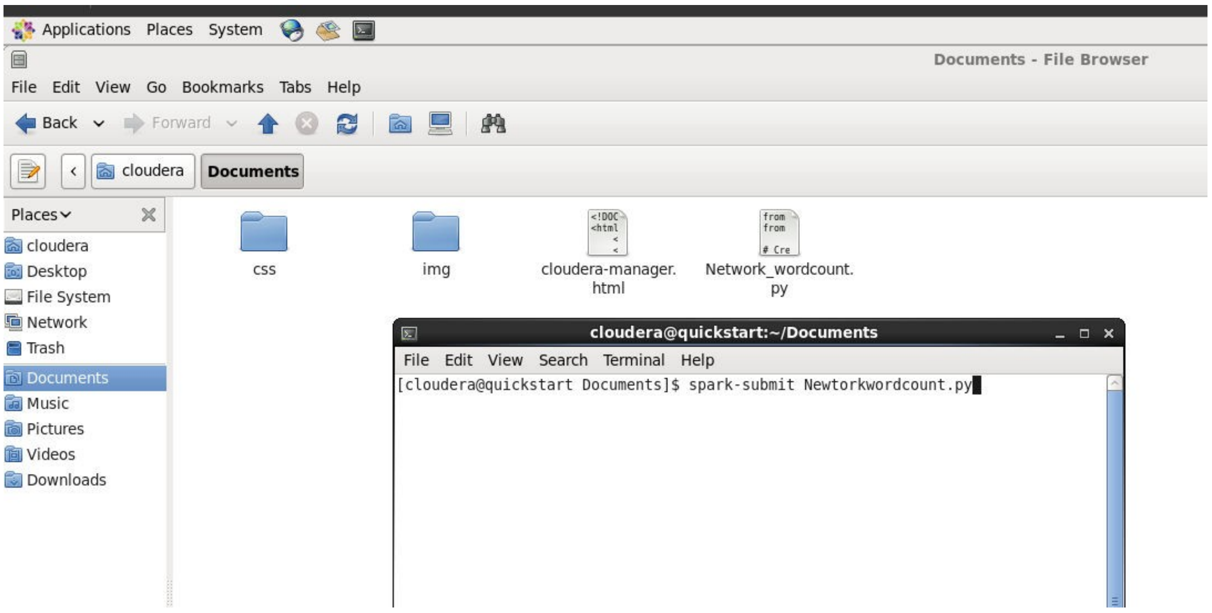
In the new terminal, start Netcat in listening mode with the specified port (9999).


cloudera@quickstart:~/Documents
File Edit View Search Terminal Help

```
[cloudera@quickstart Documents]$ nc -lk 9999
Om PAnchal
OM
ANyaa
eren
eren
abcd
 efghf
kwakwf
^Z
[1]+  Stopped                 nc -lk 9999
[cloudera@quickstart Documents]$ 
```

Step 5: Submit the Spark Job

Switch to terminal and submit the Spark job using spark-submit.



```
[cloudera@quickstart Documents]$ spark-submit Newtorkwordcount.py
```

RUN BOTH TERMINALS

Step 6: Verify Results

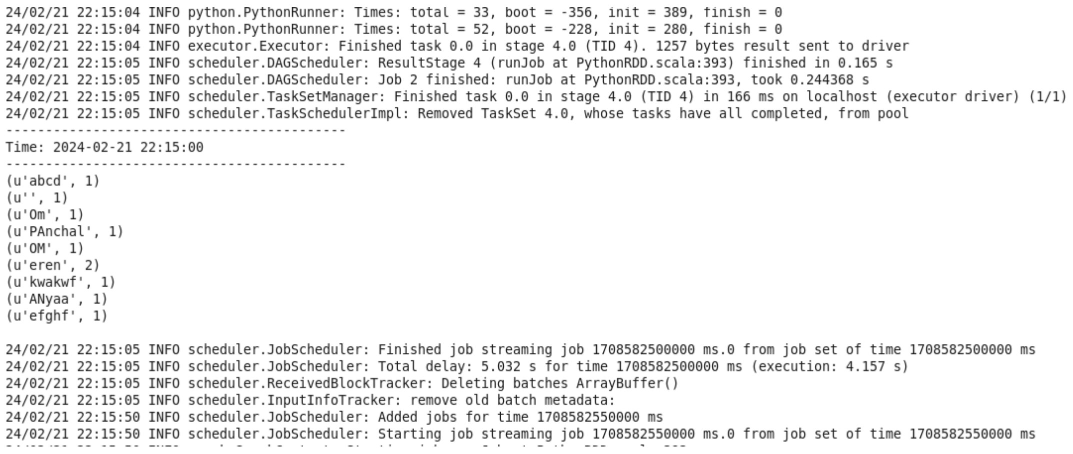
Check the terminal where Spark Streaming is running. You should see word counts printed as l
are processed.

Step 7: Stop Spark Streaming

Terminate the Spark Streaming application when you are done.(CTRL +Z)

Atul Yadav

OUTPUT –



```
24/02/21 22:15:04 INFO python.PythonRunner: Times: total = 33, boot = -356, init = 389, finish = 0
24/02/21 22:15:04 INFO python.PythonRunner: Times: total = 52, boot = -228, init = 280, finish = 0
24/02/21 22:15:04 INFO executor.Executor: Finished task 0.0 in stage 4.0 (TID 4). 1257 bytes result sent to driver
24/02/21 22:15:05 INFO scheduler.DAGScheduler: ResultStage 4 (runJob at PythonRDD.scala:393) finished in 0.165 s
24/02/21 22:15:05 INFO scheduler.DAGScheduler: Job 2 finished: runJob at PythonRDD.scala:393, took 0.244368 s
24/02/21 22:15:05 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 4.0 (TID 4) in 166 ms on localhost (executor driver) (1/1)
24/02/21 22:15:05 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 4.0, whose tasks have all completed, from pool
-------------------------------------------
Time: 2024-02-21 22:15:00
-------------------------------------------
(u'abcd', 1)
(u'', 1)
(u'Om', 1)
(u'PAnchal', 1)
(u'OM', 1)
(u'eren', 2)
(u'kwakwf', 1)
(u'ANyaa', 1)
(u'efghf', 1)

24/02/21 22:15:05 INFO scheduler.JobScheduler: Finished job streaming job 1708582500000 ms.0 from job set of time 1708582500000 ms
24/02/21 22:15:05 INFO scheduler.JobScheduler: Total delay: 5.032 s for time 1708582500000 ms (execution: 4.157 s)
24/02/21 22:15:05 INFO scheduler.ReceivedBlockTracker: Deleting batches ArrayBuffer()
24/02/21 22:15:05 INFO scheduler.InputInfoTracker: remove old batch metadata:
24/02/21 22:15:50 INFO scheduler.JobScheduler: Added jobs for time 1708582550000 ms
24/02/21 22:15:50 INFO scheduler.JobScheduler: Starting job streaming job 1708582550000 ms.0 from job set of time 1708582550000 ms
```

Atul Yadav