**Hadoop program**

Name of the Cluster: Hadoop\_processing

Master public DNS : ec2-13-127-87-85.ap-south-1.compute.amazonaws.com

**Step 1**:]# time hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar -input /user/root/consumer\_file.txt -output /user/root/Streamingoutput -file wordcount\_mapper.py -file wordcount\_reducer.py -mapper wordcount\_mapper.py -reducer wordcount\_reducer.py

**Step 2**:Optional) : How to change no of **reducers** in program

time hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar **-Dmapred.reduce.tasks=32** -input /user/root/consumer\_file.txt -output /user/root/ Streamingoutput1 -file mapper.py -file reducer.py -mapper mapper.py -reducer reducer.py

Output on web UI

<http://ec2-13-127-87-85.ap-south-1.compute.amazonaws.com:8088/cluster/apps>

**Spark program**

Name of the cluster: Spark\_processing

Master public DNS: ec2-13-127-141-84.ap-south-1.compute.amazonaws.com

**Step 1**: Open spark-shell

**Step 2**: Write below commands and run them line by line

val textfile = sc.textFile("s3://hivetableau/consumer\_file.txt")

val counts = textfile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(\_ + \_)

counts.saveAsTextFile("/user/root/out")

**Step 3**: Quit spark shell (:q)

**To run program through script**

Step 1: vi wordcount.scala

val textFile = sc.textFile("s3://hivetableau/consumer\_file.txt")

val counts = textFile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(\_ + \_)

counts.saveAsTextFile("/user/root/out")

Step 2: spark-shell -i wordcount.scala

Step 3: Quit spark shell (:q)

Output on web UI

<http://ec2-13-127-141-84.ap-south-1.compute.amazonaws.com:4043/>