LAB REPORT

Exercise-4

Question-1

By

Ambuj Mishra

202116003

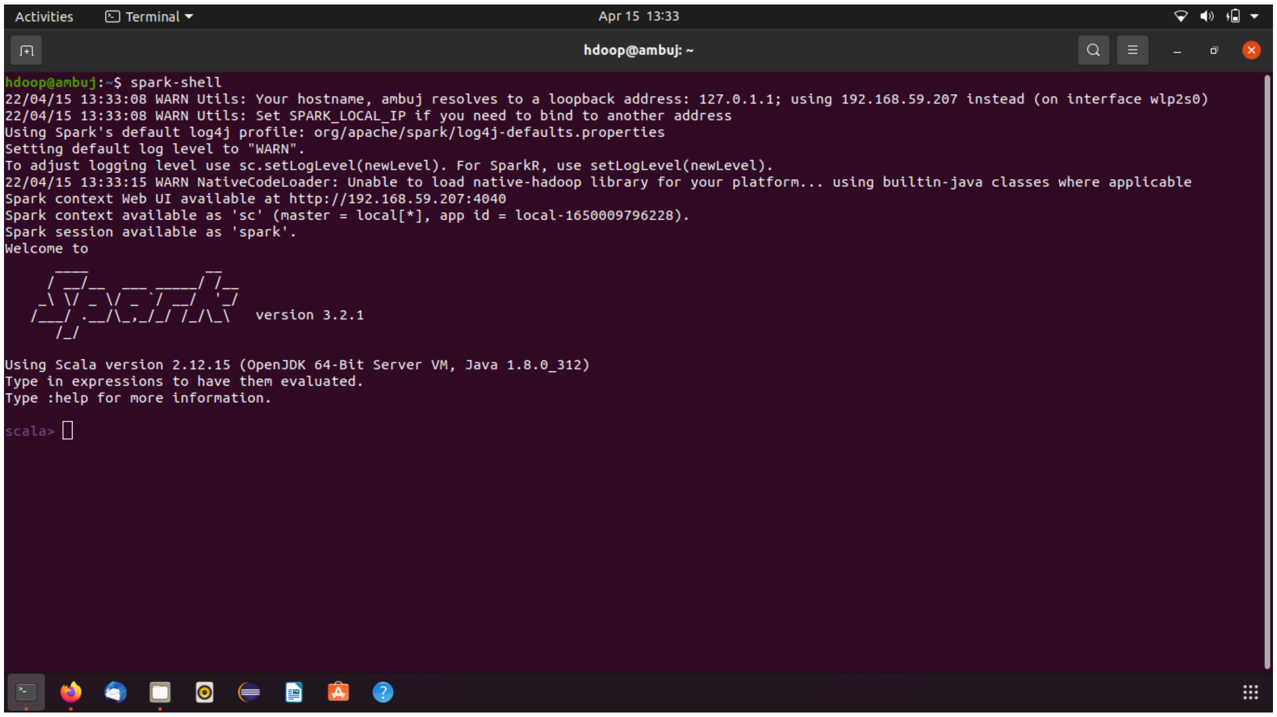
Big Data & Large-Scale Computing

DA-IICT, Gandhinagar

15-April-2022

Spark Installation:

We have followed the attached link for Scala and spark installation (https://www.tutorialspoint.com/apache_spark/apache_spark_installation.htm). Step by step procedure helps in installing the spark.



```
hdoop@ambuj:~$ spark-shell
22/04/15 13:33:08 WARN Utils: Your hostname, ambuj resolves to a loopback address: 127.0.1.1; using 192.168.59.207 instead (on interface wlp2s0)
22/04/15 13:33:08 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/04/15 13:33:15 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Spark context Web UI available at http://192.168.59.207:4040
Spark context available as 'sc' (master = local[*], app id = local-1650009796228).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.2.1
      /_/

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 1.8.0_312)
Type in expressions to have them evaluated.
Type :help for more information.

scala> 
```

Fig : Installing Spark and running spark-shell

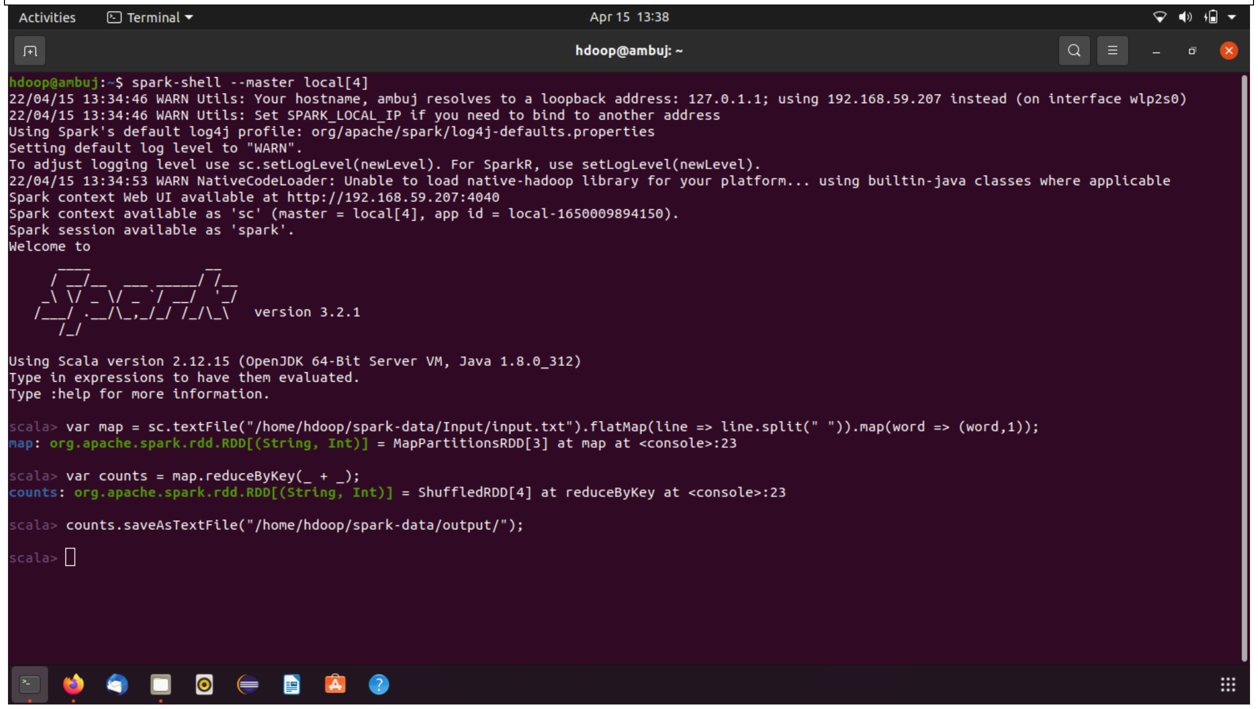
1st Question:

Part-(a)

We were initially asked to perform the word count program on spark-shell on local system. We have used the below code for that:

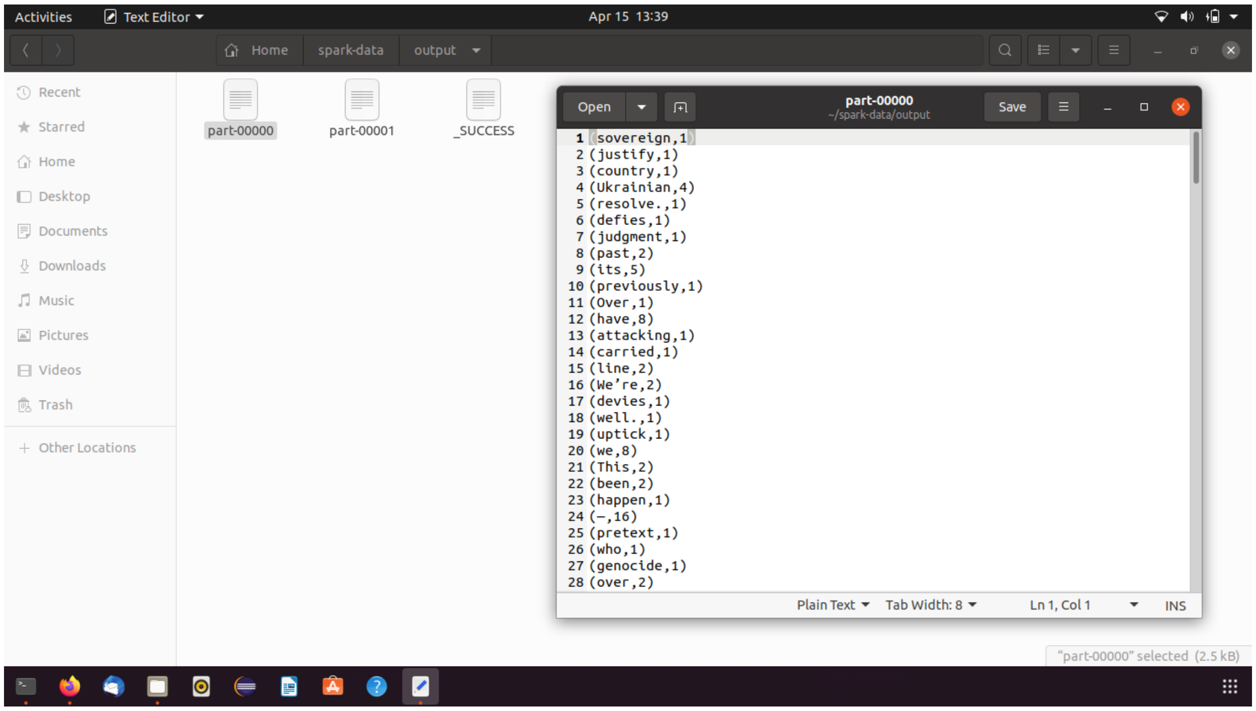
```
var map = sc.textFile("/home/hdoop/spark-data/Input/input.txt").flatMap(line => line.split(" ")).map(word => (word,1));
var counts = map.reduceByKey(_ + _);
counts.saveAsTextFile("/home/hdoop/spark-data/output");
```

Execution of code is as below:



```
hdoop@ambuj:~$ spark-shell --master local[4]
22/04/15 13:34:46 WARN Utils: Your hostname, ambuj resolves to a loopback address: 127.0.1.1; using 192.168.59.207 instead (on interface wlp2s0)
22/04/15 13:34:46 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/04/15 13:34:53 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Spark context Web UI available at http://192.168.59.207:4040
Spark context available as 'sc' (master = local[4], app id = local-1650009894150).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.2.1
      /_/

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 1.8.0_312)
Type in expressions to have them evaluated.
Type :help for more information.

scala> var map = sc.textFile("/home/hdoop/spark-data/Input/input.txt").flatMap(line => line.split(" ")).map(word => (word,1));
map: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[3] at map at <console>:23

scala> var counts = map.reduceByKey(_ + _);
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:23

scala> counts.saveAsTextFile("/home/hdoop/spark-data/output/");

scala> 
```

Fig : Running Word Count on Scala on local system

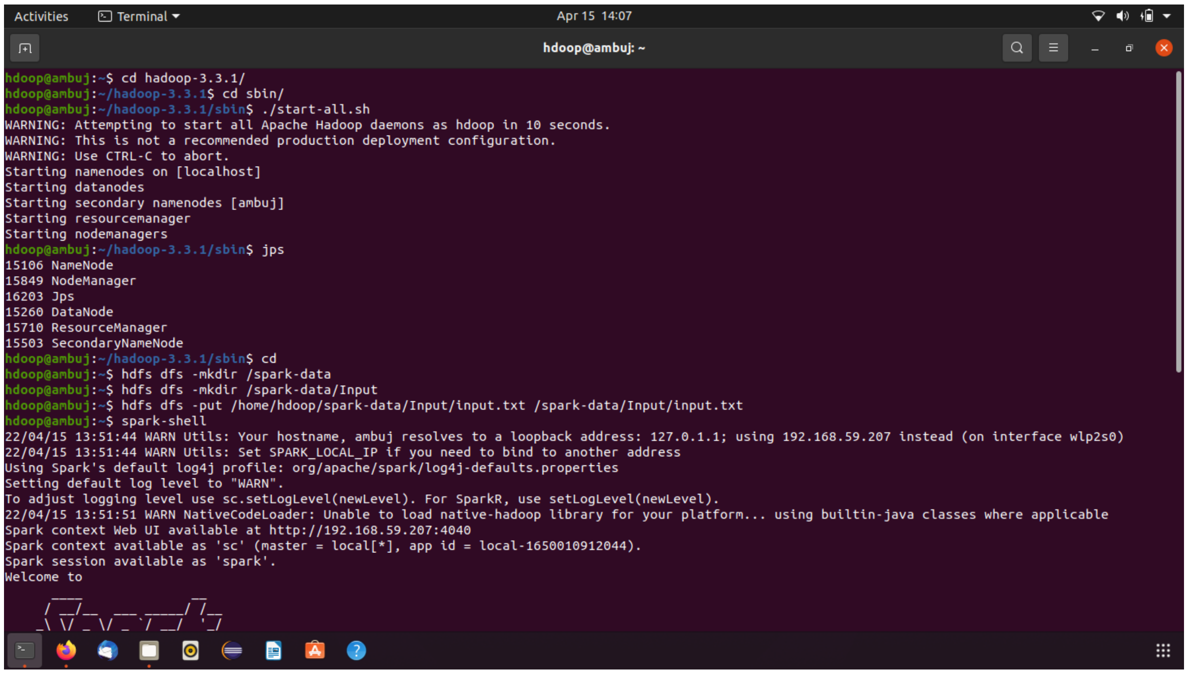


```
(sovereign,1)
(justify,1)
(country,1)
(Ukrainian,4)
(resolve.,1)
(defies,1)
(judgment,1)
(past,2)
(its,5)
(previously,1)
(Over,1)
(have,8)
(attacking,1)
(carried,1)
(line,2)
(We're,2)
(devies,1)
(well.,1)
(uptick,1)
(we,8)
(This,2)
(been,2)
(happen,1)
(–,16)
(pretext,1)
(who,1)
(genocide,1)
(over,2)
```

Fig : Output of Word Count on Scala on local system

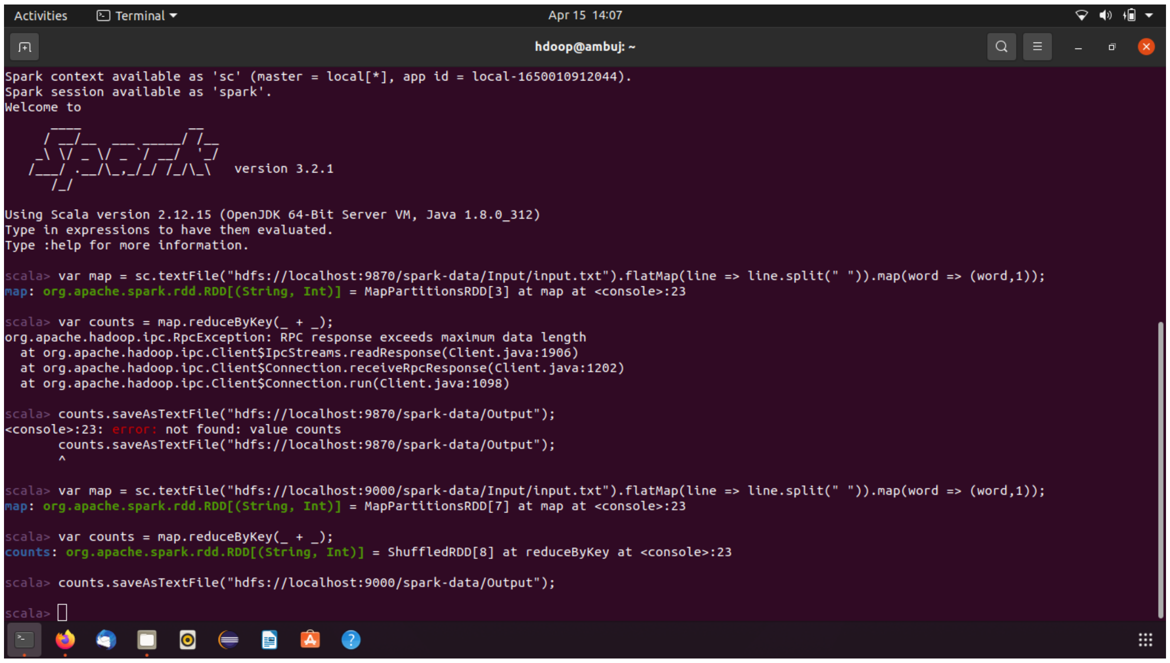
We were then asked to perform the word count program on spark-shell with an input file stored on HDFS. We started Hadoop first. Then we put the file in HDFS and then using that same HDFS path, we generated the word count. We have used the below code for that:

1. hdfs dfs –mkdir /spark-data
2. hdfs dfs –mkdir /spark-data/Input
3. hdfs dfs –put /home/hdoop/spark-data/Input/input.txt /spark-data/Input/input.txt
4. spark-shell
5. var map = sc.textFile("hdfs://localhost:9000/spark-data/Input/input.txt").flatMap(line => line.split(" ")).map(word => (word,1));
6. var counts = map.reduceByKey(_ + _);
7. counts.saveAsTextFile(" hdfs://localhost:9000/spark-data/Output");



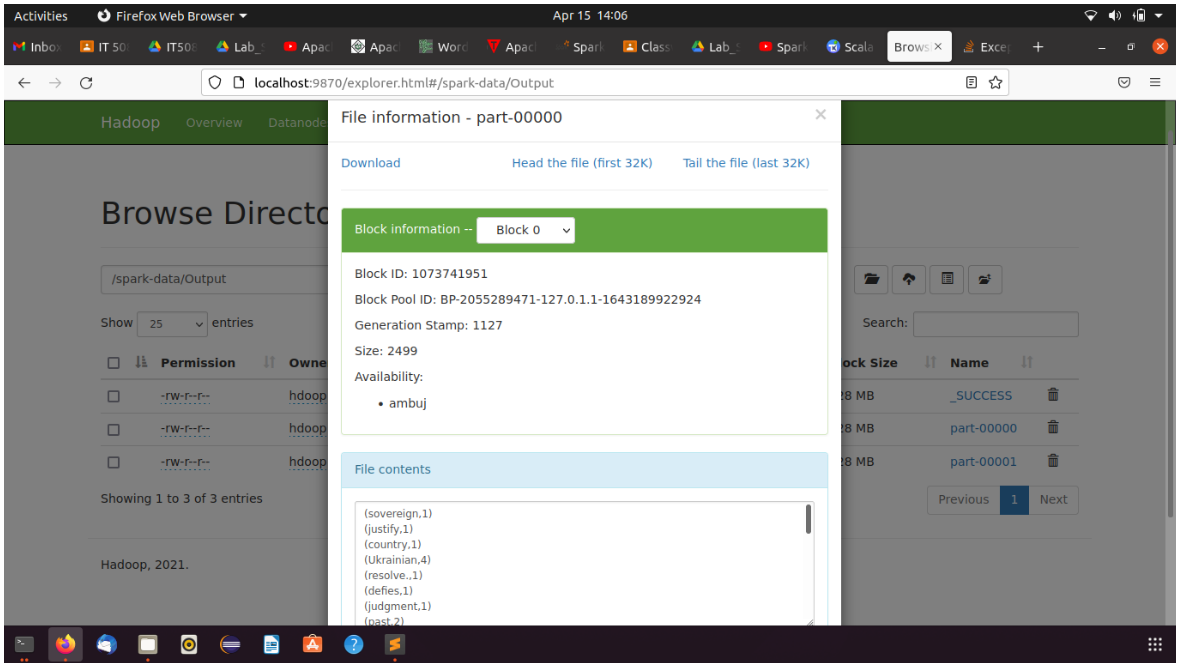
```
hdoop@ambuj:~$ cd hadoop-3.3.1/
hdoop@ambuj:~/hadoop-3.3.1$ cd sbin/
hdoop@ambuj:~/hadoop-3.3.1/sbin$ ./start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hdoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ambuj]
Starting resourcemanager
Starting nodemanagers
hdoop@ambuj:~/hadoop-3.3.1/sbin$ jps
15106 NameNode
15849 NodeManager
16203 Jps
15260 DataNode
15710 ResourceManager
15503 SecondaryNameNode
hdoop@ambuj:~/hadoop-3.3.1/sbin$ cd
hdoop@ambuj:~$ hdfs dfs -mkdir /spark-data
hdoop@ambuj:~$ hdfs dfs -mkdir /spark-data/Input
hdoop@ambuj:~$ hdfs dfs -put /home/hdoop/spark-data/Input/input.txt /spark-data/Input/input.txt
hdoop@ambuj:~$ spark-shell
22/04/15 13:51:44 WARN Utils: Your hostname, ambuj resolves to a loopback address: 127.0.1.1; using 192.168.59.207 instead (on interface wlp2s0)
22/04/15 13:51:44 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/04/15 13:51:51 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Spark context Web UI available at http://192.168.59.207:4040
Spark context available as 'sc' (master = local[*], app id = local-1650010912044).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
```

Fig : Starting Hadoop, Putting file in HDFS and starting Scala



```
Spark context available as 'sc' (master = local[*], app id = local-1650010912044).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.2.1
      /_/

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 1.8.0_312)
Type in expressions to have them evaluated.
Type :help for more information.

scala> var map = sc.textFile("hdfs://localhost:9870/spark-data/Input/input.txt").flatMap(line => line.split(" ")).map(word => (word,1));
map: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[3] at map at <console>:23

scala> var counts = map.reduceByKey(_ + _);
org.apache.hadoop.ipc.RpcException: RPC response exceeds maximum data length
  at org.apache.hadoop.ipc.Client$IpcStreams.readResponse(Client.java:1906)
  at org.apache.hadoop.ipc.Client$Connection.receiveRpcResponse(Client.java:1202)
  at org.apache.hadoop.ipc.Client$Connection.run(Client.java:1098)

scala> counts.saveAsTextFile("hdfs://localhost:9870/spark-data/Output");
<console>:23: error: not found: value counts
       counts.saveAsTextFile("hdfs://localhost:9870/spark-data/Output");
       ^

scala> var map = sc.textFile("hdfs://localhost:9000/spark-data/Input/input.txt").flatMap(line => line.split(" ")).map(word => (word,1));
map: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[7] at map at <console>:23

scala> var counts = map.reduceByKey(_ + _);
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[8] at reduceByKey at <console>:23

scala> counts.saveAsTextFile("hdfs://localhost:9000/spark-data/Output");

scala> 
```

Fig : Running Word count with file in HDFS



localhost:9870/explorer.html#/spark-data/Output

Browse Directory

/spark-data/Output

File information - part-00000

Download | Head the file (first 32K) | Tail the file (last 32K)

Block information -- Block 0

Block ID: 1073741951
Block Pool ID: BP-2055289471-127.0.1.1-1643189922924
Generation Stamp: 1127
Size: 2499
Availability:
- ambuj

File contents

```
(sovereign,1)
(justify,1)
(country,1)
(Ukrainian,4)
(resolve.,1)
(defies,1)
(judgment,1)
(past,2)
```

Permission	Owner	Block Size	Name
-rw-r--r--	hdoop	128 MB	_SUCCESS
-rw-r--r--	hdoop	128 MB	part-00000
-rw-r--r--	hdoop	128 MB	part-00001

Showing 1 to 3 of 3 entries

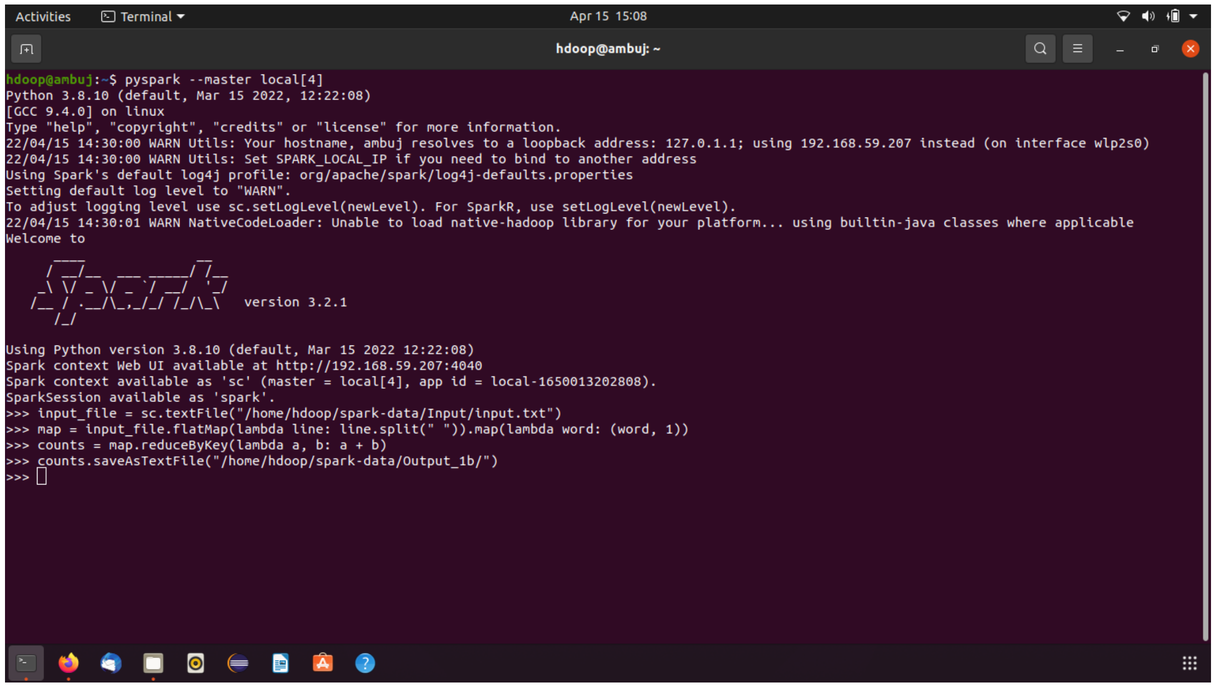
Hadoop, 2021.

Fig : Result file on HDFS

<u>Part-(b)</u>

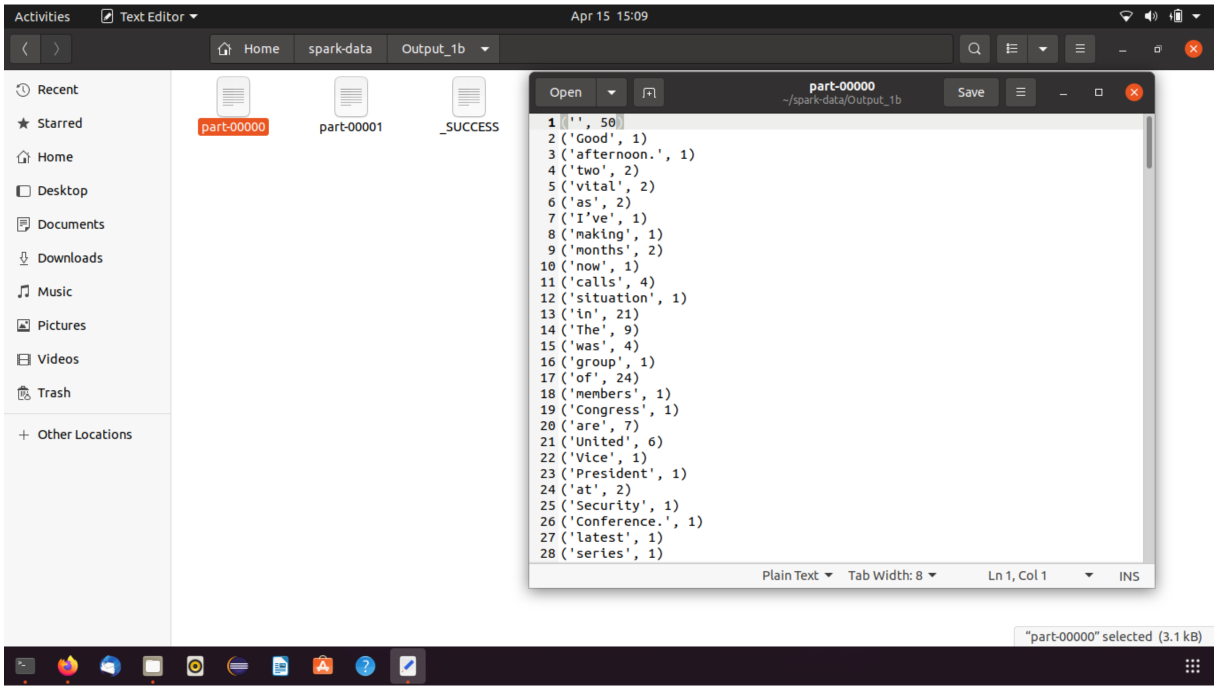
We were asked to install pyspark which automatically got installed with Apache spark installation. We ran the following code to calculate the word count on the data -

1. pyspark --master local[4]
2. input_file = sc.textFile("/home/hdoop/spark-data/Input/input.txt")
3. map = input_file.flatMap(lambda line: line.split(" ")).map(lambda word: (word, 1))
4. counts = map.reduceByKey(lambda a, b: a + b)
5. counts.saveAsTextFile("/home/hdoop/spark-data/Output_1b/")



```
hdoop@ambuj:~$ pyspark --master local[4]
Python 3.8.10 (default, Mar 15 2022, 12:22:08)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
22/04/15 14:30:00 WARN Utils: Your hostname, ambuj resolves to a loopback address: 127.0.1.1; using 192.168.59.207 instead (on interface wlp2s0)
22/04/15 14:30:00 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/04/15 14:30:01 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.2.1
      /_/

Using Python version 3.8.10 (default, Mar 15 2022 12:22:08)
Spark context Web UI available at http://192.168.59.207:4040
Spark context available as 'sc' (master = local[4], app id = local-1650013202808).
SparkSession available as 'spark'.
>>> input_file = sc.textFile("/home/hdoop/spark-data/Input/input.txt")
>>> map = input_file.flatMap(lambda line: line.split(" ")).map(lambda word: (word, 1))
>>> counts = map.reduceByKey(lambda a, b: a + b)
>>> counts.saveAsTextFile("/home/hdoop/spark-data/Output_1b/")
>>> 
```

Fig : Running Word Count on pyspark



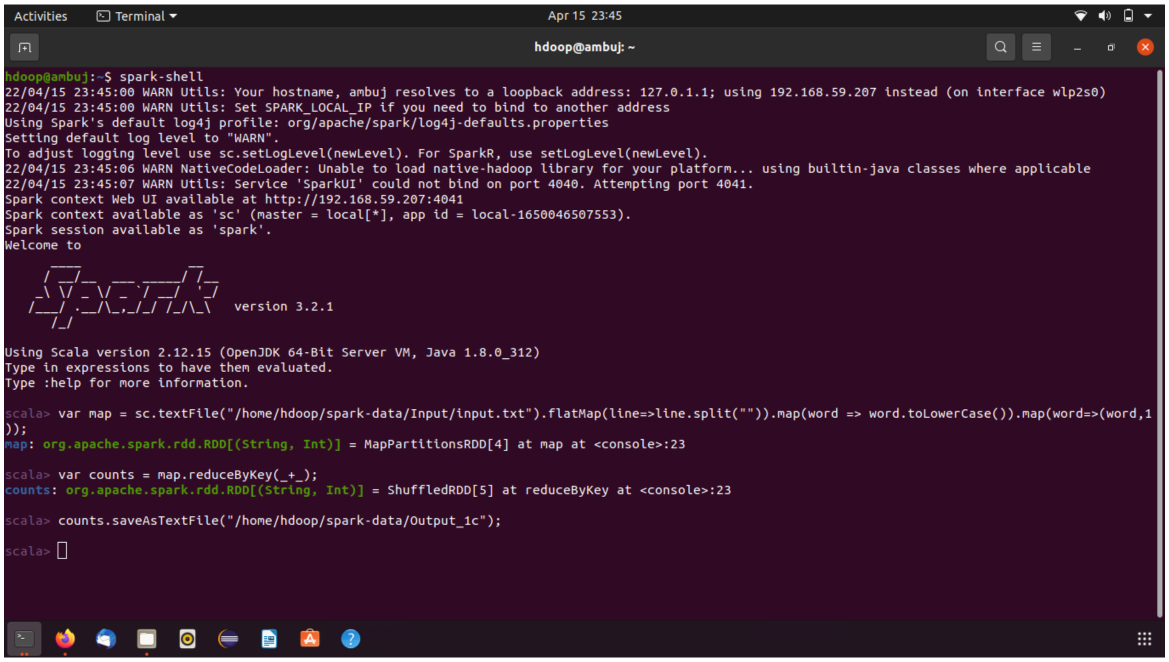
```
('', 50)
('Good', 1)
('afternoon.', 1)
('two', 2)
('vital', 2)
('as', 2)
('I've', 1)
('making', 1)
('months', 2)
('now', 1)
('calls', 4)
('situation', 1)
('in', 21)
('The', 9)
('was', 4)
('group', 1)
('of', 24)
('members', 1)
('Congress', 1)
('are', 7)
('United', 6)
('Vice', 1)
('President', 1)
('at', 2)
('Security', 1)
('Conference.', 1)
('latest', 1)
('series', 1)
```

Fig : Output of Word Count on pyspark

Part-(c)

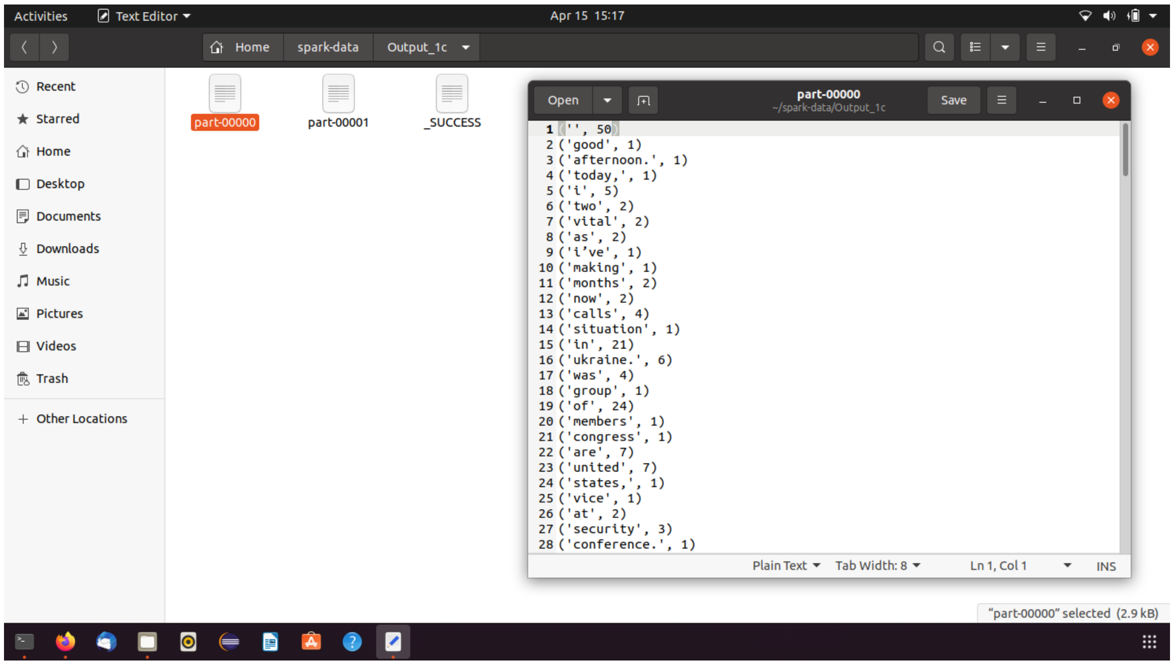
We want to perform lower() so that we get case-insensitive output. For that we have done following code on spark-shell -

1. var map = sc.textFile("/home/hdoop/spark-data/Input/input.txt").flatMap(line => line.split(" ")).map(word => word.toLowerCase()).map(word => (word,1));
2. var counts = map.reduceByKey(_ + _);
3. counts.saveAsTextFile("/home/hdoop/spark-data/output_1c");



```
hdoop@ambuj:~$ spark-shell
22/04/15 23:45:00 WARN Utils: Your hostname, ambuj resolves to a loopback address: 127.0.1.1; using 192.168.59.207 instead (on interface wlp2s0)
22/04/15 23:45:00 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/04/15 23:45:06 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
22/04/15 23:45:07 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
Spark context Web UI available at http://192.168.59.207:4041
Spark context available as 'sc' (master = local[*], app id = local-1650046507553).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.2.1
      /_/

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 1.8.0_312)
Type in expressions to have them evaluated.
Type :help for more information.

scala> var map = sc.textFile("/home/hdoop/spark-data/Input/input.txt").flatMap(line=>line.split("")).map(word => word.toLowerCase()).map(word=>(word,1
));
map: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[4] at map at <console>:23

scala> var counts = map.reduceByKey(_+_);
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[5] at reduceByKey at <console>:23

scala> counts.saveAsTextFile("/home/hdoop/spark-data/Output_1c");

scala> 
```

Fig : Getting Case-insensitive Word count output on scala

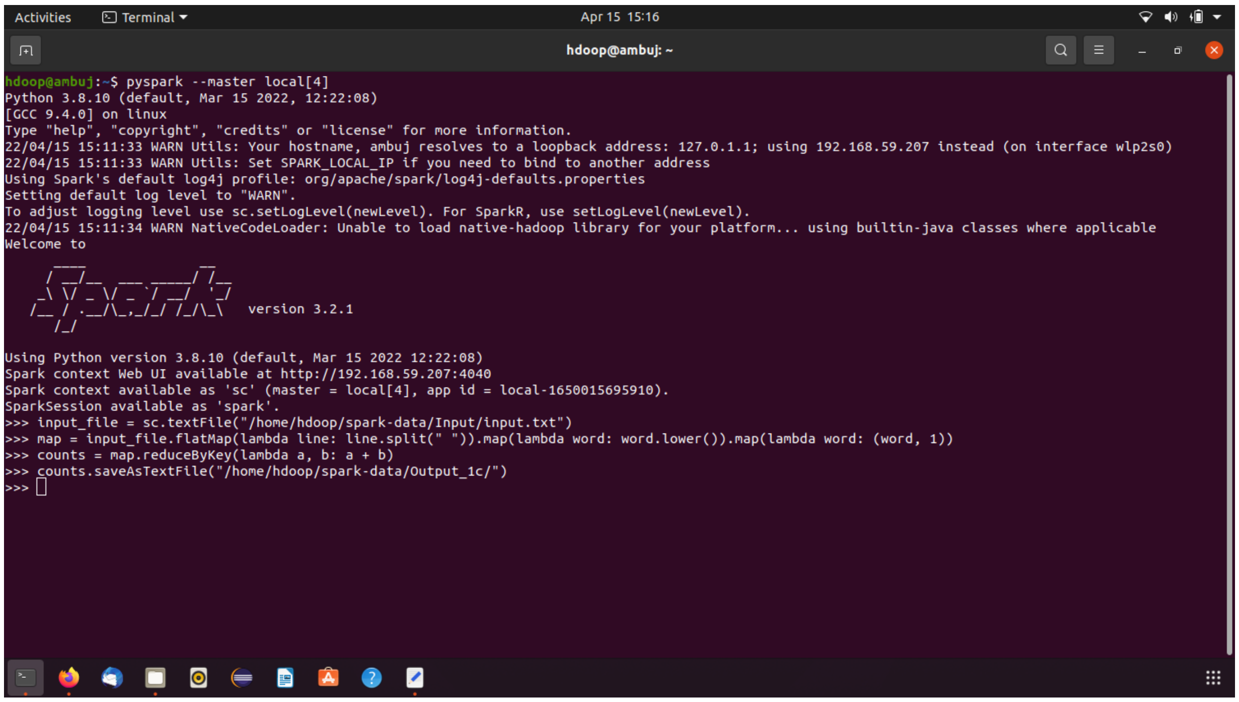


```
('', 50)
('good', 1)
('afternoon.', 1)
('today,', 1)
('i', 5)
('two', 2)
('vital', 2)
('as', 2)
('i’ve', 1)
('making', 1)
('months', 2)
('now', 2)
('calls', 4)
('situation', 1)
('in', 21)
('ukraine.', 6)
('was', 4)
('group', 1)
('of', 24)
('members', 1)
('congress', 1)
('are', 7)
('united', 7)
('states,', 1)
('vice', 1)
('at', 2)
('security', 3)
('conference.', 1)
```

FIg : Output of Case-insensitive Word count output on scala

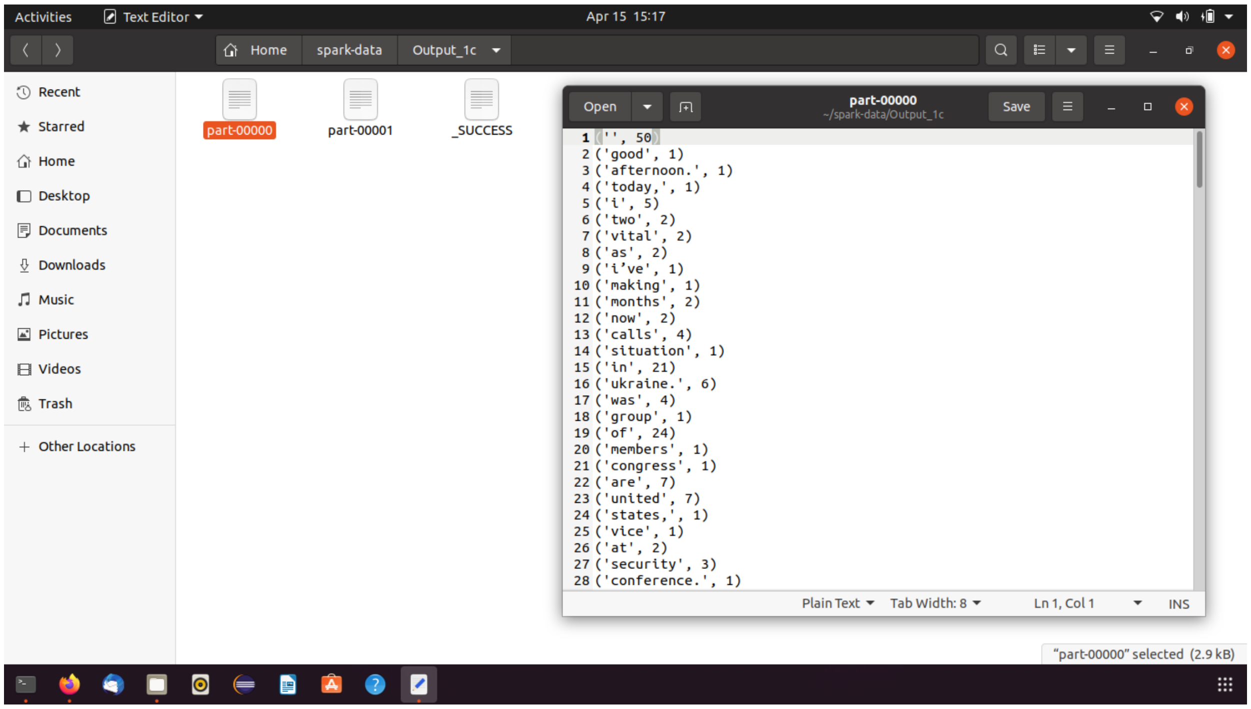
We want to perform lower() so that we get case-insensitive output. For that we have done following code on pyspark -

1. pyspark --master local[4]
2. input_file = sc.textFile("/home/hdoop/spark-data/Input/input.txt")
3. map = input_file.flatMap(lambda line: line.split(" ")).map(lambda word: word.lower()).map(lambda word: (word, 1))
4. counts = map.reduceByKey(lambda a, b: a + b)
5. counts.saveAsTextFile("/home/hdoop/spark-data/Output_1b/")



```
hdoop@ambuj:~$ pyspark --master local[4]
Python 3.8.10 (default, Mar 15 2022, 12:22:08)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
22/04/15 15:11:33 WARN Utils: Your hostname, ambuj resolves to a loopback address: 127.0.1.1; using 192.168.59.207 instead (on interface wlp2s0)
22/04/15 15:11:33 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/04/15 15:11:34 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.2.1
      /_/

Using Python version 3.8.10 (default, Mar 15 2022 12:22:08)
Spark context Web UI available at http://192.168.59.207:4040
Spark context available as 'sc' (master = local[4], app id = local-1650015695910).
SparkSession available as 'spark'.
>>> input_file = sc.textFile("/home/hdoop/spark-data/Input/input.txt")
>>> map = input_file.flatMap(lambda line: line.split(" ")).map(lambda word: word.lower()).map(lambda word: (word, 1))
>>> counts = map.reduceByKey(lambda a, b: a + b)
>>> counts.saveAsTextFile("/home/hdoop/spark-data/Output_1c/")
>>> 
```

Fig : Getting Case-insensitive Word count output on pyspark



```
('', 50)
('good', 1)
('afternoon.', 1)
('today,', 1)
('i', 5)
('two', 2)
('vital', 2)
('as', 2)
('i’ve', 1)
('making', 1)
('months', 2)
('now', 2)
('calls', 4)
('situation', 1)
('in', 21)
('ukraine.', 6)
('was', 4)
('group', 1)
('of', 24)
('members', 1)
('congress', 1)
('are', 7)
('united', 7)
('states,', 1)
('vice', 1)
('at', 2)
('security', 3)
('conference.', 1)
```

FIg : Output of Case-insensitive Word count output on pyspark