

Aim

Algorithm/Procedure

- Open **Oracle VirtualBox**.
- Double-click the **Hadoop** virtual machine.
- Open the terminal and type:
- `spark-shell`
- Confirm Spark version and ensure Spark shell is running.

```

ponny@ubuntu:~$ spark-shell
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).
25/03/24 11:32:27 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
25/03/24 11:32:28 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.0.1; using 10.0.2.15 instead (on interface eth0)
25/03/24 11:32:28 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
25/03/24 11:32:33 WARN ObjectStore: Failed to get database global_temp, returning NoSuchObjectException
Spark context Web UI available at http://10.0.2.15:4040
Spark context available as 'sc' (master = local[*], app id = local-1742796148803).
Spark session available as 'spark'.
Welcome to

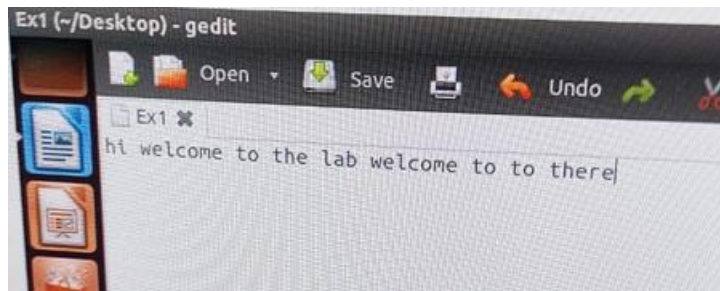
  ____              __
 / ___/____  ____  /  /
/ /  / __ \/ __ \/  /
/___/___/ /___/ /___/

 version 2.2.0

Using Scala version 2.11.8 (Java HotSpot(TM) Client VM, Java 1.8.0_45)
Type in expressions to have them evaluated.
Type :help for more information.

```

- On the Ubuntu desktop, create a sample text file named Ex1.txt.
- Add sample content
- Save this file on the desktop.



3. Running Word Count Program in Spark

- In the Spark shell (Scala prompt), run the following commands:
- Load text file from desktop (use appropriate path)
- Split lines into words
- Map words to (word, 1) pairs
- Reduce by key to count occurrences
- Collect and print the result

```
scala> var a = sc.textFile("/home/ponny/Desktop/Ex1").flatMap(line => line.split(" ")).map(word => (word,1))
25/03/24 11:36:08 WARN SizeEstimator: Failed to check whether UseCompressedOops is set; assuming yes
a: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[3] at map at <console>:24

scala> var b = a.reduceByKey(_+_);
b: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:26

scala> b.collect
res0: Array[(String, Int)] = Array((lab,1), (hi,1), (to,2), (welcome,2), (the,2), (world,1))

scala>
```

4. View the activity

- Open your browser and go to Spark UI:
 - <http://localhost:4040>
- You can monitor job progress, stages, tasks, and storage here.

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	collect at <console>:29	2025/03/24 11:37:02	0.4 s	2/2	2/2

Details for Job 0

Status: SUCCEEDED

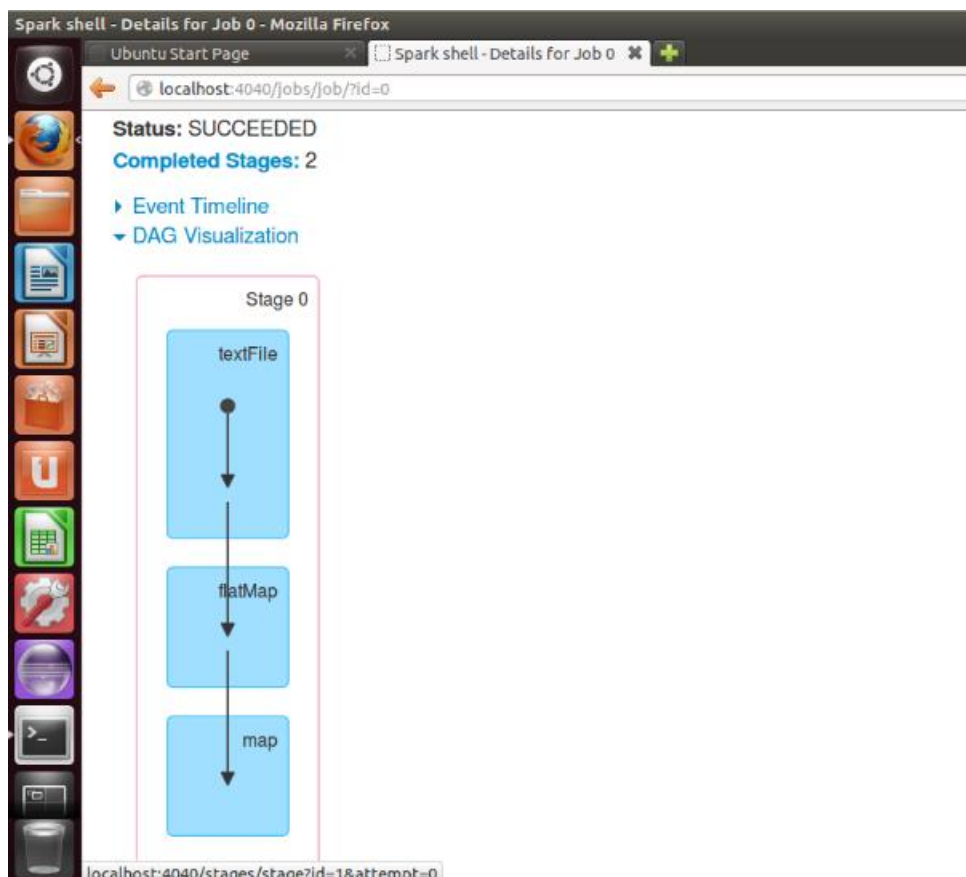
Completed Stages: 2

▶ Event Timeline

▶ DAG Visualization

Completed Stages (2)

Stage Id ▼	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
1	collect at <console>:29 +details	2025/03/24 11:37:03	57 ms	1/1			86.0 B	
0	map at <console>:24 +details	2025/03/24 11:37:02	0.2 s	1/1	43.0 B			86.0 B



Program (Word Count in Spark using Scala)

```
var a= sc.textFile("/home/ponny/Desktop/Ex1").flatMap(line => line.split("")).map(word => (word, 1))
```

```
var ba.reduceByKey(_+_);
```

```
b.collect
```

Output

For the sample text:

hi welcome to the lab welcome to the world

The output in Spark shell will be:

Array[(String, Int)] = Array((lab,1), (hi,1), (to,2), (welcome, 2), (the, 2), (world, 1))

```
scala> b.collect
res0: Array[(String, Int)] = Array((lab,1), (hi,1), (to,2), (welcome,2), (the,2), (world,1))
```

Result

The Word Count program was successfully executed using the Spark framework. The frequency of each word in the input file was counted and displayed as output.