

Step 1: Install Apache Spark

1. **Download Apache Spark:** Go to the Apache Spark website (<https://spark.apache.org/downloads.html>) and download the latest version of Apache Spark.
2. **Extract Spark:** After downloading, extract the contents of the downloaded file to a directory of your choice. (Create DSBDAL folder in home directory and extract the contents)
3. **Set up Environment Variables:** Add Spark's `bin` directory to your `PATH` environment variable. You can do this by modifying your shell profile file (e.g., `.bashrc`, `.bash_profile`, `.zshrc`, etc.) (Our is bash)
 - a. **Open your Terminal:** Launch your terminal application. This process may differ slightly depending on your operating system (e.g., Terminal on macOS, Command Prompt or PowerShell on Windows, Terminal on Linux).
 - b. **Determine your Shell:** Before proceeding, determine which shell you're using. Common shells include Bash, Zsh, and Fish. You can typically find out your current shell by running the following command:

```
echo $SHELL
```

- c. **Edit the Shell Profile File:** Based on your shell, you'll edit the corresponding profile file:
 - i. For Bash (`~/.bashrc`):
- d. **Add Spark's bin Directory to PATH:** Inside the opened file, add the following line at the end:

```
export
PATH="/home/student/DSBDAL/spark-3.5.1-bin-hadoop3/bin:$P
ATH"
```

```
student@student: ~
GNU nano 6.2 /home/student/.bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *(*) ;;
  *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend
export PATH="/home/student/DSBDAL/spark-3.5.1-bin-hadoop3/bin:$PATH"
# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

Read 133 lines
^G Help    ^O Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit    ^R Read File  ^_ Replace   ^U Paste     ^J Justify   ^_/ Go To Line
```

- e. **Save and Exit:** After adding the line, save the file and exit the editor. In Nano, you can do this by pressing **Ctrl + O** to write the file and **Ctrl + X** to exit.
- f. **Apply Changes:** To apply the changes to your current terminal session, either close and reopen the terminal or run:

```
source ~/.bashrc
```

- g. **Verify:** You can verify that Spark's **bin** directory has been added to your **PATH** by running:

```
echo $PATH
```

You should see the path to Spark's **bin** directory listed in the output.

https://www.tutorialspoint.com/apache_spark/apache_spark_introduction.htm

Create wordcount_input.txt file in DSBDAL folder and pest any paragraph into it

Option 1: one by one statement execution on Scala Terminal

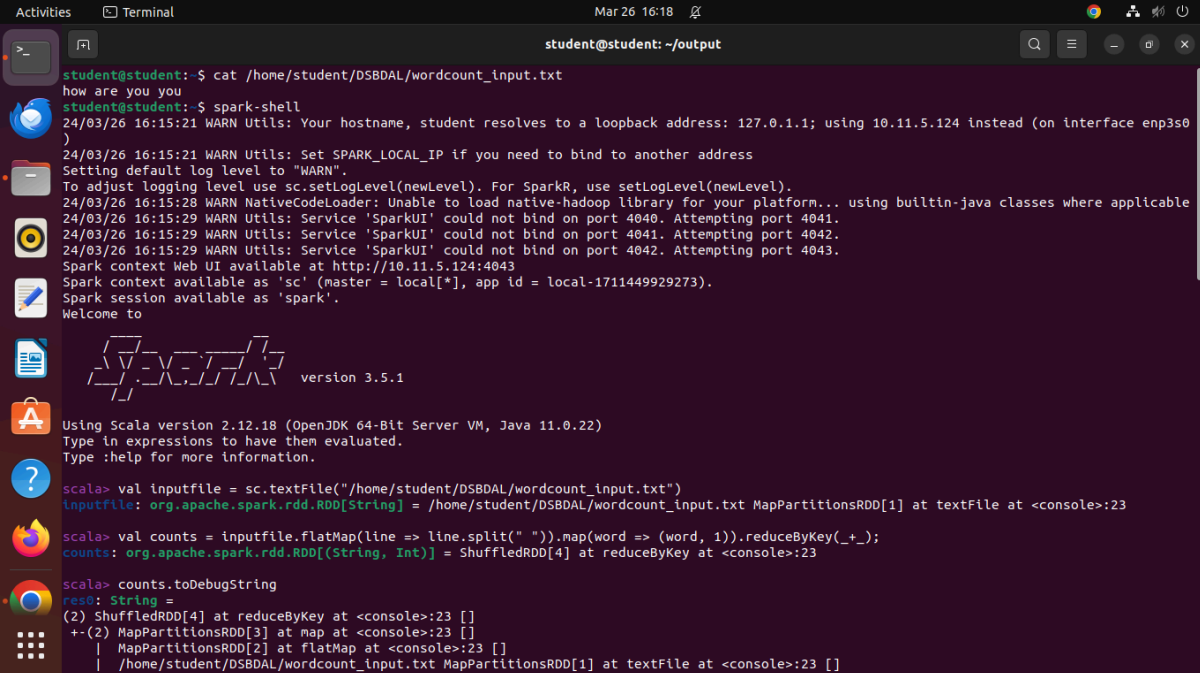
```
val inputfile = sc.textFile("/home/student/DSBDAL/input.txt")
```

```
val counts = inputfile.flatMap(line => line.split(" ")).map(word => (word,  
1)).reduceByKey(_+_);
```

```
counts.toDebugString
```

```
counts.cache()
```

```
counts.saveAsTextFile("output")
```



```
student@student:~$ cat /home/student/DSBDAL/wordcount_input.txt
how are you you
student@student:~$ spark-shell
24/03/26 16:15:21 WARN Utils: Your hostname, student resolves to a loopback address: 127.0.1.1; using 10.11.5.124 instead (on interface enp3s0)
24/03/26 16:15:21 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/03/26 16:15:28 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
24/03/26 16:15:29 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
24/03/26 16:15:29 WARN Utils: Service 'SparkUI' could not bind on port 4041. Attempting port 4042.
24/03/26 16:15:29 WARN Utils: Service 'SparkUI' could not bind on port 4042. Attempting port 4043.
Spark context Web UI available at http://10.11.5.124:4043
Spark context available as 'sc' (master = local[*], app id = local-1711449929273).
Spark session available as 'spark'.
Welcome to

      ____
     / ___/
    / __/   version 3.5.1
   /___/

Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 11.0.22)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val inputfile = sc.textFile("/home/student/DSBDAL/wordcount_input.txt")
inputfile: org.apache.spark.rdd.RDD[String] = /home/student/DSBDAL/wordcount_input.txt MapPartitionsRDD[1] at textFile at <console>:23

scala> val counts = inputfile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_+_);
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:23

scala> counts.toDebugString
res0: String =
(2) ShuffledRDD[4] at reduceByKey at <console>:23 []
+- (2) MapPartitionsRDD[3] at map at <console>:23 []
|   MapPartitionsRDD[2] at flatMap at <console>:23 []
|   /home/student/DSBDAL/wordcount_input.txt MapPartitionsRDD[1] at textFile at <console>:23 []
|   /home/student/DSBDAL/wordcount_input.txt MapPartitionsRDD[1] at textFile at <console>:23 []
```

```
Activities Terminal Mar 26 16:19 student@student: ~/output

scala> counts.cache()
res2: counts.type = ShuffledRDD[4] at reduceByKey at <console>:23

scala> counts.saveAsTextFile("output")
org.apache.hadoop.mapred.FileAlreadyExistsException: Output directory file:/home/student/output already exists
    at org.apache.hadoop.mapred.FileOutputFormat.checkOutputSpecs(FileOutputFormat.java:131)
    at org.apache.spark.internal.io.HadoopMapRedWriteConfigUtil.assertConf(SparkHadoopWriter.scala:299)
    at org.apache.spark.internal.io.SparkHadoopWriter$.write(SparkHadoopWriter.scala:71)
    at org.apache.spark.rdd.PairRDDFunctions.$anonfun$saveAsHadoopDataset$1(PairRDDFunctions.scala:1091)
    at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)
    at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
    at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
    at org.apache.spark.rdd.RDD.withScope(RDD.scala:410)
    at org.apache.spark.rdd.PairRDDFunctions.saveAsHadoopDataset(PairRDDFunctions.scala:1089)
    at org.apache.spark.rdd.PairRDDFunctions.$anonfun$saveAsHadoopFile$4(PairRDDFunctions.scala:1062)
    at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)
    at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
    at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
    at org.apache.spark.rdd.RDD.withScope(RDD.scala:410)
    at org.apache.spark.rdd.PairRDDFunctions.saveAsHadoopFile(PairRDDFunctions.scala:1027)
    at org.apache.spark.rdd.PairRDDFunctions.$anonfun$saveAsHadoopFile$3(PairRDDFunctions.scala:1009)
    at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)
    at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
    at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
    at org.apache.spark.rdd.RDD.withScope(RDD.scala:410)
    at org.apache.spark.rdd.PairRDDFunctions.saveAsHadoopFile(PairRDDFunctions.scala:1008)
    at org.apache.spark.rdd.PairRDDFunctions.$anonfun$saveAsHadoopFile$2(PairRDDFunctions.scala:965)
    at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)
    at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
    at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
    at org.apache.spark.rdd.RDD.withScope(RDD.scala:410)
    at org.apache.spark.rdd.PairRDDFunctions.saveAsHadoopFile(PairRDDFunctions.scala:963)
    at org.apache.spark.rdd.RDD.$anonfun$saveAsTextFile$2(RDD.scala:1623)
    at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)
    at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
    at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
    at org.apache.spark.rdd.RDD.withScope(RDD.scala:410)
    at org.apache.spark.rdd.RDD.saveAsTextFile(RDD.scala:1623)
    at org.apache.spark.rdd.RDD.$anonfun$saveAsTextFile$1(RDD.scala:1609)
    at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)
    at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
    at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
    at org.apache.spark.rdd.RDD.withScope(RDD.scala:410)
    at org.apache.spark.rdd.RDD.saveAsTextFile(RDD.scala:1609)
    ... 47 elided
```

```
Activities Terminal Mar 26 16:19 student@student: ~/output

at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)
at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
at org.apache.spark.rdd.RDD.withScope(RDD.scala:410)
at org.apache.spark.rdd.PairRDDFunctions.saveAsHadoopFile(PairRDDFunctions.scala:1008)
at org.apache.spark.rdd.PairRDDFunctions.$anonfun$saveAsHadoopFile$2(PairRDDFunctions.scala:965)
at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)
at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
at org.apache.spark.rdd.RDD.withScope(RDD.scala:410)
at org.apache.spark.rdd.PairRDDFunctions.saveAsHadoopFile(PairRDDFunctions.scala:963)
at org.apache.spark.rdd.RDD.$anonfun$saveAsTextFile$2(RDD.scala:1623)
at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)
at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
at org.apache.spark.rdd.RDD.withScope(RDD.scala:410)
at org.apache.spark.rdd.RDD.saveAsTextFile(RDD.scala:1623)
at org.apache.spark.rdd.RDD.$anonfun$saveAsTextFile$1(RDD.scala:1609)
at scala.runtime.java8.JFunction0$mcV$sp.apply(JFunction0$mcV$sp.java:23)
at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
at org.apache.spark.rdd.RDD.withScope(RDD.scala:410)
at org.apache.spark.rdd.RDD.saveAsTextFile(RDD.scala:1609)
... 47 elided

scala>
[4]+ Stopped spark-shell
student@student:~$ cd output/
student@student:~/output$ ls -l
part-00000
part-00001
_SUCCESS
student@student:~/output$ cat part-00000
(are,1)
(how,1)
student@student:~/output$ cat part-00001
(you,2)
student@student:~/output$
```

Option 2: Using scala program file

Create file named wordcount.scala with following code

```
val inputfile = sc.textFile("/home/student/DSBDAL/wordcount_input.txt")
```

```
val counts = inputfile.flatMap(line => line.split(" ")).map(word => (word,
1)).reduceByKey(_+_);
```

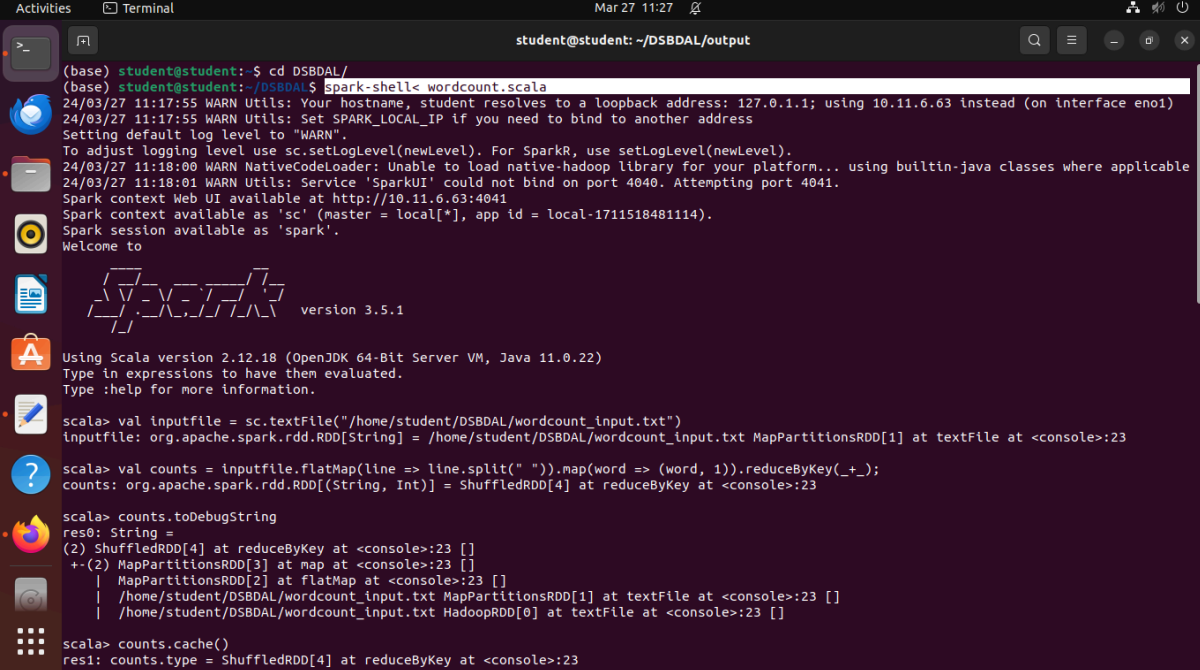
```
counts.toDebugString
```

```
counts.cache()
```

```
counts.saveAsTextFile("output")
```

Use following command to execute the code

```
spark-shell< wordcount.scala
```



```
(base) student@student: $ cd DSBDAL/
(base) student@student:~/DSBDAL$ spark-shell< wordcount.scala
24/03/27 11:17:55 WARN Utils: Your hostname, student resolves to a loopback address: 127.0.1.1; using 10.11.6.63 instead (on interface eno1)
24/03/27 11:17:55 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/03/27 11:18:00 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
24/03/27 11:18:01 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
Spark context Web UI available at http://10.11.6.63:4041
Spark context available as 'sc' (master = local[*], app id = local-1711518481114).
Spark session available as 'spark'.
Welcome to

      ____              __
     / __ )__  ___  ___/  /
    / __  /  / _ > / _ > /
   / ____/  / ___/ / ___/
  /_/    /_/____/_/_/___/

version 3.5.1

Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 11.0.22)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val inputfile = sc.textFile("/home/student/DSBDAL/wordcount_input.txt")
inputfile: org.apache.spark.rdd.RDD[String] = /home/student/DSBDAL/wordcount_input.txt MapPartitionsRDD[1] at textFile at <console>:23

scala> val counts = inputfile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_+_);
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:23

scala> counts.toDebugString
res0: String =
(2) ShuffledRDD[4] at reduceByKey at <console>:23 []
+- (2) MapPartitionsRDD[3] at map at <console>:23 []
|   MapPartitionsRDD[2] at flatMap at <console>:23 []
|   /home/student/DSBDAL/wordcount_input.txt MapPartitionsRDD[1] at textFile at <console>:23 []
|   /home/student/DSBDAL/wordcount_input.txt HadoopRDD[0] at textFile at <console>:23 []

scala> counts.cache()
res1: counts.type = ShuffledRDD[4] at reduceByKey at <console>:23
```

```
scala> :quit
(base) student@student:~/DSBDAL$ cd output/
(base) student@student:~/DSBDAL/output$ cat part-00000
(are,1)
(how,1)
(base) student@student:~/DSBDAL/output$ cat part-00001
(you,2)
```