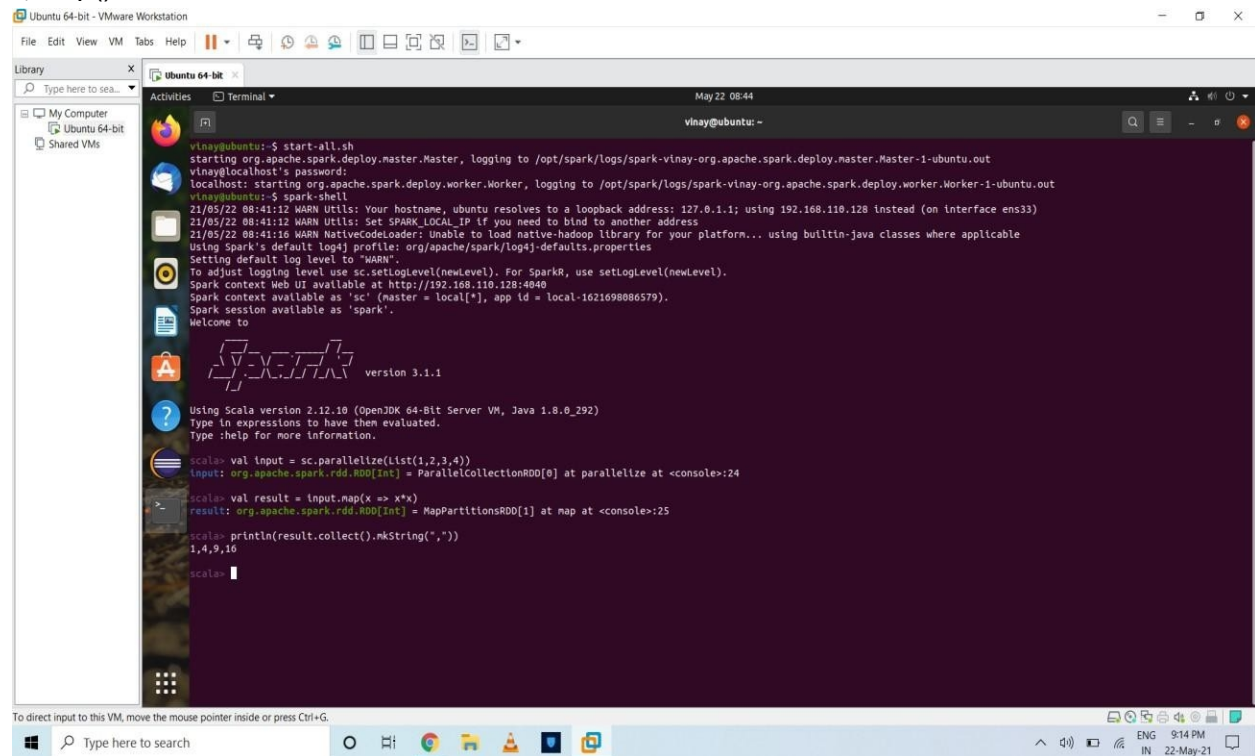


# SASWAT SINGH 1BM18CS094

## 1)map()



The screenshot shows a terminal window in a VM titled 'Ubuntu 64-bit - VMware Workstation'. The user 'vinay' is logged in. The terminal output shows the Spark 3.1.1 installation process, including the Spark logo and version information. The user then runs a Scala script to demonstrate the map() function. The script defines an input RDD, applies a map operation, and prints the result.

```
vinay@ubuntu:~$ start-all.sh
starting org.apache.spark.deploy.master.Master, logging to /opt/spark/logs/spark-vinay-org.apache.spark.deploy.master.Master-1-ubuntu.out
vinay@localhost:~$ start-all.sh
starting org.apache.spark.deploy.worker.Worker, logging to /opt/spark/logs/spark-vinay-org.apache.spark.deploy.worker.Worker-1-ubuntu.out
vinay@ubuntu:~$ spark-shell
21/05/22 08:41:12 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.1.1; using 192.168.110.128 instead (on interface ens33)
21/05/22 08:41:12 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
21/05/22 08:41:16 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
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).
Spark context Web UI available at http://192.168.110.128:4040
Spark context available as 'sc' (master = local[*], app id = local-1621698086579).
Spark session available as 'spark'.
Welcome to

Spark version 3.1.1

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_292)
Type in expressions to have them evaluated.
Type :help for more information.

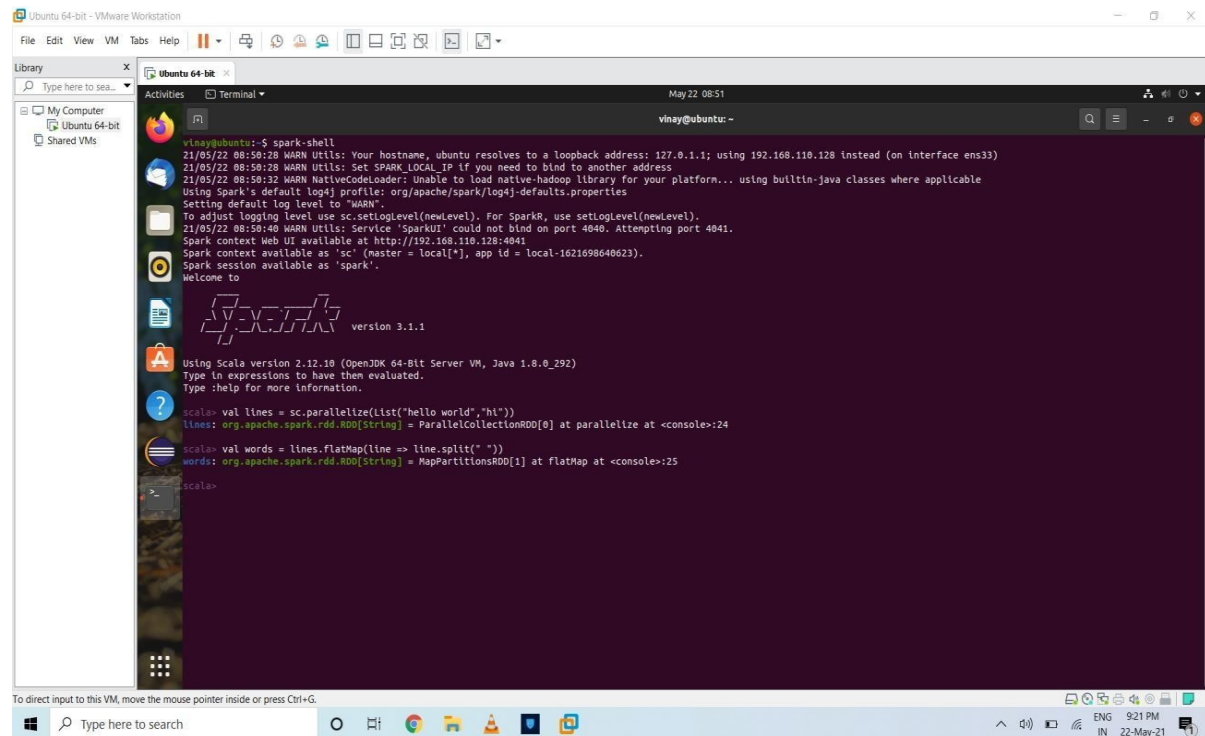
scala> val input = sc.parallelize(List(1,2,3,4))
input: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val result = input.map(x => x*x)
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[1] at map at <console>:25

scala> println(result.collect().mkString(","))
1,4,9,16

scala>
```

## 2)flatMap()



The screenshot shows a terminal window in a VM titled 'Ubuntu 64-bit - VMware Workstation'. The user 'vinay' is logged in. The terminal output shows the Spark 3.1.1 installation process, including the Spark logo and version information. The user then runs a Scala script to demonstrate the flatMap() function. The script defines a list of lines, applies a flatMap operation to split the lines into words, and prints the result.

```
vinay@ubuntu:~$ spark-shell
21/05/22 08:50:28 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.1.1; using 192.168.110.128 instead (on interface ens33)
21/05/22 08:50:28 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
21/05/22 08:50:32 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
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).
21/05/22 08:50:40 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
Spark context Web UI available at http://192.168.110.128:4041
Spark context available as 'sc' (master = local[*], app id = local-1621698640623).
Spark session available as 'spark'.
Welcome to

Spark version 3.1.1

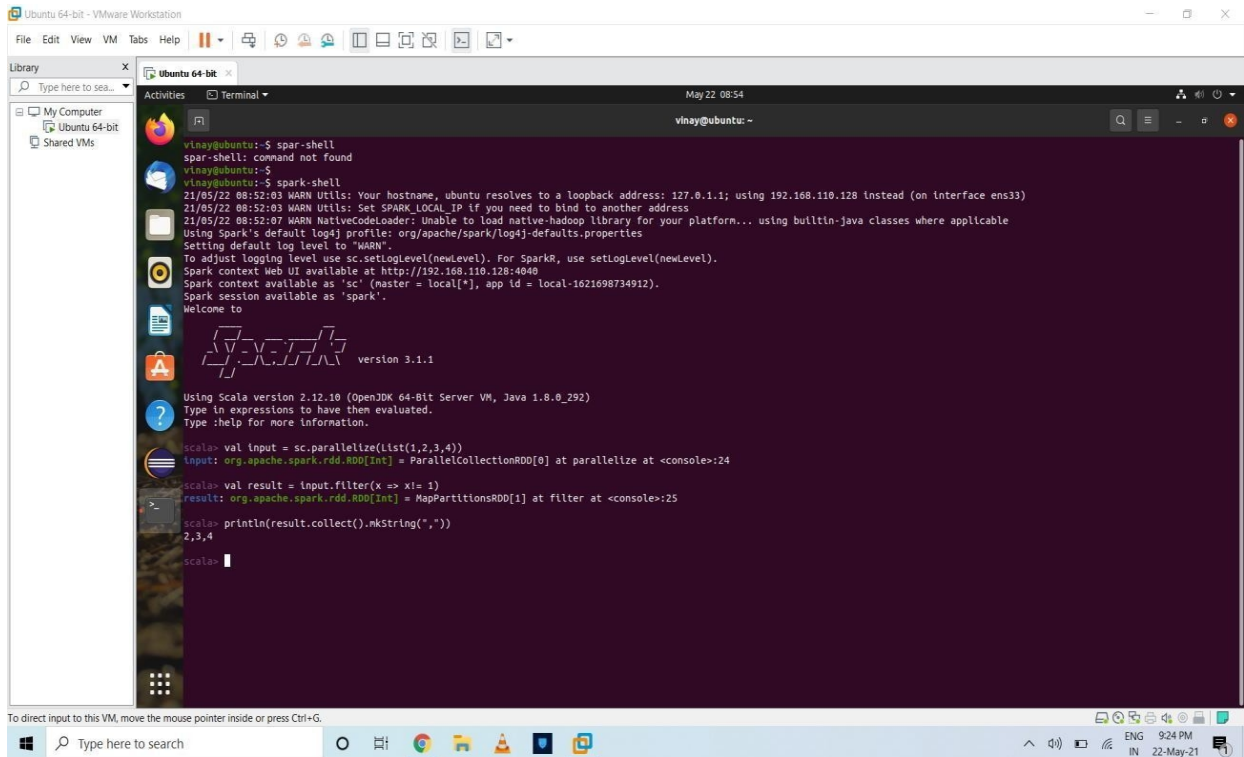
Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_292)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val lines = sc.parallelize(List("hello world", "hi"))
lines: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val words = lines.flatMap(line => line.split(" "))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[1] at flatMap at <console>:25

scala>
```

### 3)filter()



```
vinay@ubuntu:~$ spar-shell
spar-shell: command not found
vinay@ubuntu:~$ spark-shell
21/05/22 08:52:03 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.1.1; using 192.168.110.128 instead (on interface ens33)
21/05/22 08:52:03 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
21/05/22 08:52:07 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
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).
Spark context Web UI available at http://192.168.110.128:4040
Spark context available as 'sc' (master = local[*], app id = local-1621698734912).
Spark session available as 'spark'.
Welcome to

      ____              __
     / __ )____ _  ___/ /_  __
    / __ \_ / __ / __ \ __/ / /_
   / ___/___/ ___/ ___/___/_/_/

version 3.1.1

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_292)
Type in expressions to have them evaluated.
Type :help for more information.

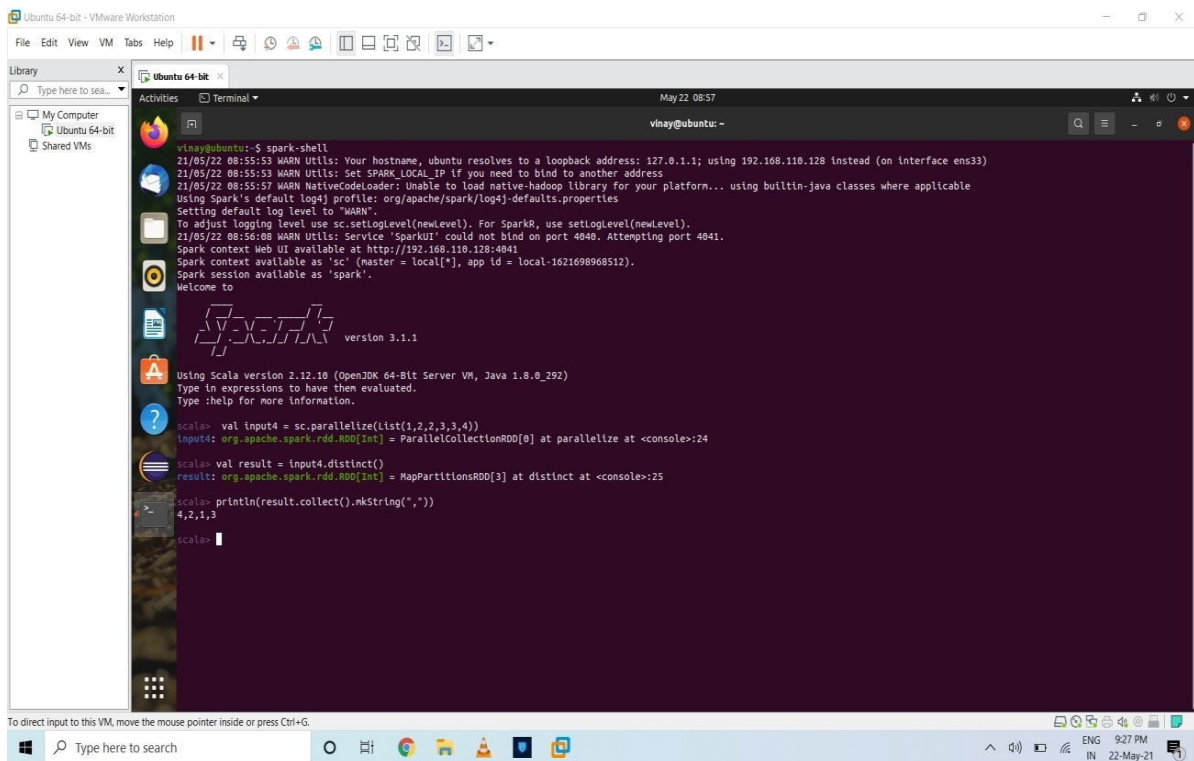
scala> val input = sc.parallelize(List(1,2,3,4))
input: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val result = input.filter(x => x != 1)
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[1] at filter at <console>:25

scala> println(result.collect().mkString(","))
2,3,4

scala>
```

### 4 distinct()



```
vinay@ubuntu:~$ spark-shell
21/05/22 08:55:53 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.1.1; using 192.168.110.128 instead (on interface ens33)
21/05/22 08:55:53 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
21/05/22 08:55:57 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
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).
21/05/22 08:56:08 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
Spark context Web UI available at http://192.168.110.128:4041
Spark context available as 'sc' (master = local[*], app id = local-1621698968512).
Spark session available as 'spark'.
Welcome to

      ____              __
     / __ )____ _  ___/ /_  __
    / __ \_ / __ / __ \ __/ / /_
   / ___/___/ ___/ ___/___/_/_/

version 3.1.1

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_292)
Type in expressions to have them evaluated.
Type :help for more information.

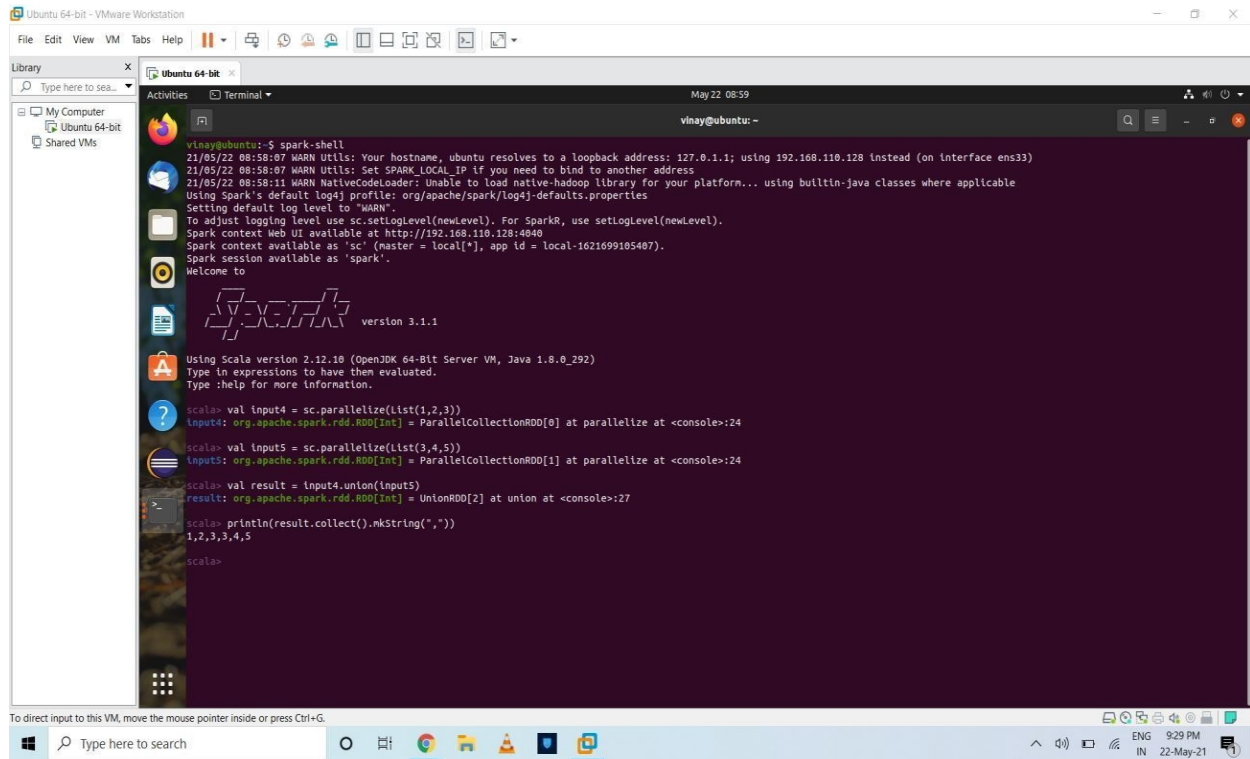
scala> val input4 = sc.parallelize(List(1,2,2,3,4))
input4: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val result = input4.distinct()
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[3] at distinct at <console>:25

scala> println(result.collect().mkString(","))
4,2,1,3

scala>
```

5)



```
vinay@ubuntu:~$ spark-shell
21/05/22 08:58:07 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.0.1; using 192.168.110.128 instead (on interface ens33)
21/05/22 08:58:07 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
21/05/22 08:58:11 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
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).
Spark context Web UI available at http://192.168.110.128:4040
Spark context available as 'sc' (master = local[*], app id = local-16216991054077).
Spark session available as 'spark'.
Welcome to

      ____              __
     /  _/             /  |
    /  /_  _____  /  |  Spark version 3.1.1
   /  __/              /___|
  /____/              ____|

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_292)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val input4 = sc.parallelize(List(1,2,3))
input4: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

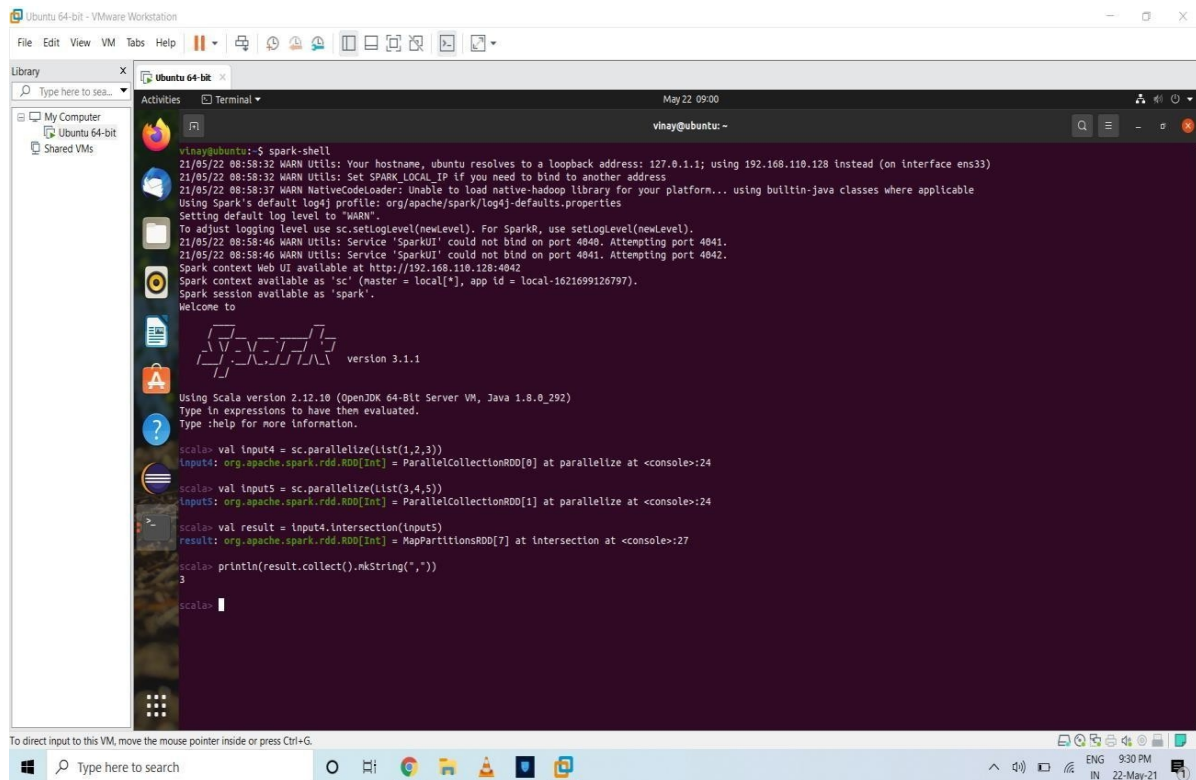
scala> val input5 = sc.parallelize(List(3,4,5))
input5: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[1] at parallelize at <console>:24

scala> val result = input4.union(input5)
result: org.apache.spark.rdd.RDD[Int] = UnionRDD[2] at union at <console>:27

scala> println(result.collect().mkString(","))
1,2,3,3,4,5

scala>
```

6 intersection()



```
vinay@ubuntu:~$ spark-shell
21/05/22 08:58:32 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.0.1; using 192.168.110.128 instead (on interface ens33)
21/05/22 08:58:32 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
21/05/22 08:58:37 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
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).
21/05/22 08:58:46 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
21/05/22 08:58:46 WARN Utils: Service 'SparkUI' could not bind on port 4041. Attempting port 4042.
Spark context Web UI available at http://192.168.110.128:4042
Spark context available as 'sc' (master = local[*], app id = local-16216991267977).
Spark session available as 'spark'.
Welcome to

      ____              __
     /  _/             /  |
    /  /_  _____  /  |  Spark version 3.1.1
   /  __/              /___|
  /____/              ____|

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_292)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val input4 = sc.parallelize(List(1,2,3))
input4: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

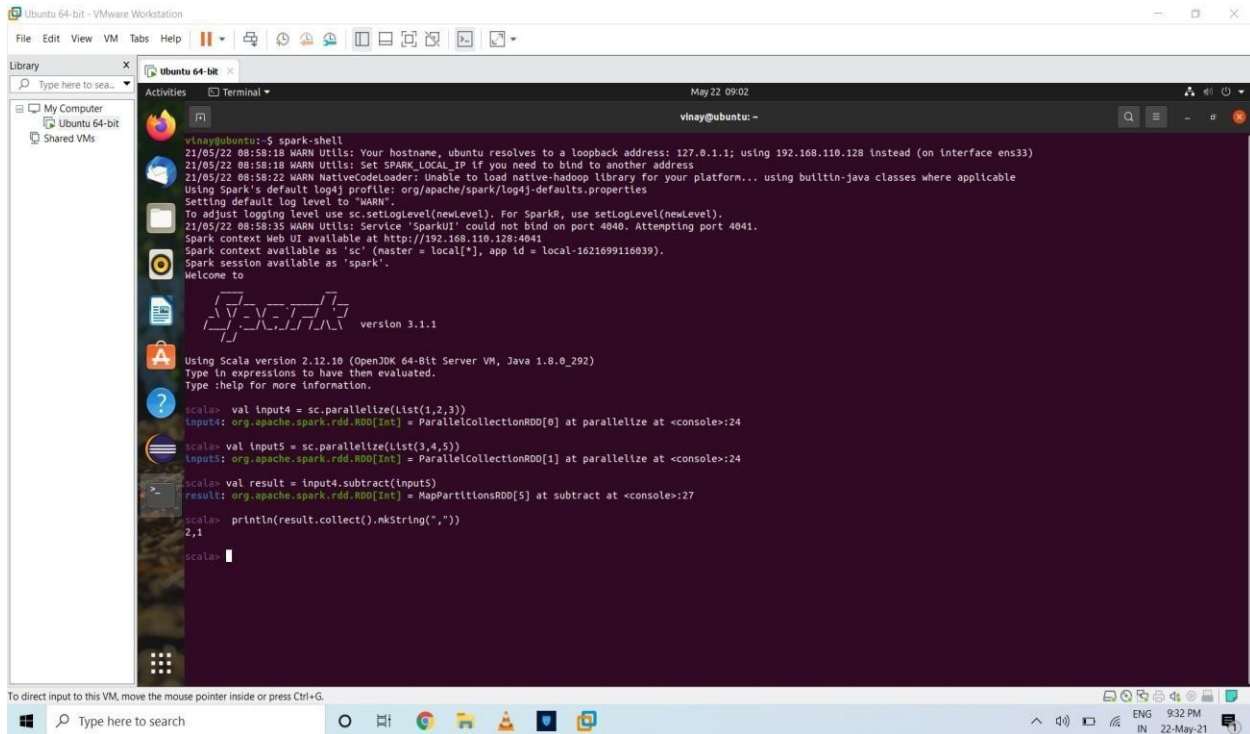
scala> val input5 = sc.parallelize(List(3,4,5))
input5: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[1] at parallelize at <console>:24

scala> val result = input4.intersection(input5)
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[7] at intersection at <console>:27

scala> println(result.collect().mkString(","))
3

scala>
```

## 7)subtract()



```
vinay@ubuntu:~$ spark-shell
21/05/22 08:58:18 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.1.1; using 192.168.110.128 instead (on interface ens33)
21/05/22 08:58:18 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
21/05/22 08:58:22 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
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).
21/05/22 08:58:35 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
Spark context Web UI available at http://192.168.110.128:4041
Spark context available as 'sc' (master = local[*], app id = local-1621699116039).
Spark session available as 'spark'.
Welcome to
      ____              __
     /  _/             /  |
    /  /_  _____/  __|
   /  __/          /  /_ |
  /  /_           /  / __|
 /  __/         /  /_/ |
/_  /_         /____/_|

version 3.1.1

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_292)
Type in expressions to have them evaluated.
Type :help for more information.

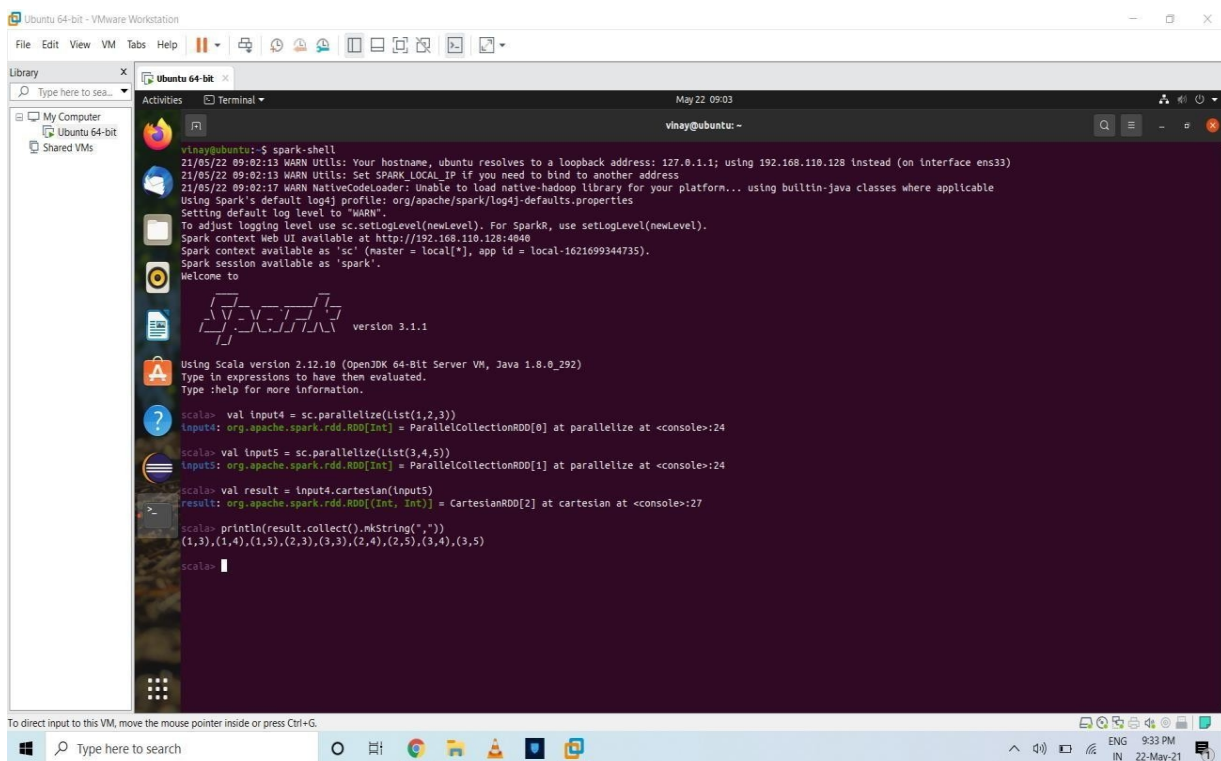
scala> val input4 = sc.parallelize(List(1,2,3))
input4: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val input5 = sc.parallelize(List(3,4,5))
input5: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[1] at parallelize at <console>:24

scala> val result = input4.subtract(input5)
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[5] at subtract at <console>:27

scala> println(result.collect().mkString(","))
2,1
scala>
```

## 8)cartesian()



```
vinay@ubuntu:~$ spark-shell
21/05/22 09:02:13 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.1.1; using 192.168.110.128 instead (on interface ens33)
21/05/22 09:02:17 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
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).
Spark context Web UI available at http://192.168.110.128:4040
Spark context available as 'sc' (master = local[*], app id = local-1621699344735).
Spark session available as 'spark'.
Welcome to
      ____              __
     /  _/             /  |
    /  /_  _____/  __|
   /  __/          /  /_ |
  /  /_           /  / __|
 /  __/         /  /_/ |
/_  /_         /____/_|

version 3.1.1

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_292)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val input4 = sc.parallelize(List(1,2,3))
input4: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val input5 = sc.parallelize(List(3,4,5))
input5: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[1] at parallelize at <console>:24

scala> val result = input4.cartesian(input5)
result: org.apache.spark.rdd.RDD[(Int, Int)] = CartesianRDD[2] at cartesian at <console>:27

scala> println(result.collect().mkString(","))
(1,3),(1,4),(1,5),(2,3),(3,3),(2,4),(2,5),(3,4),(3,5)
scala>
```