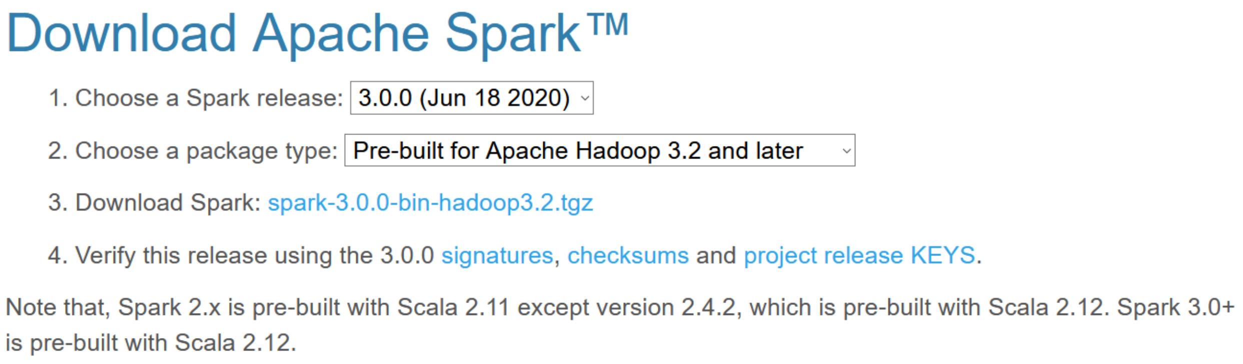
Install Spark in Ubuntu

Prepared By: Gyaneshwar Bohara

1. Install Java
2. Download Apache Spark
   - Go to the downloads page.
   - Select latest release. For the package type, choose 'Pre-built for Apache Hadoop'.
   - Page will look like this:

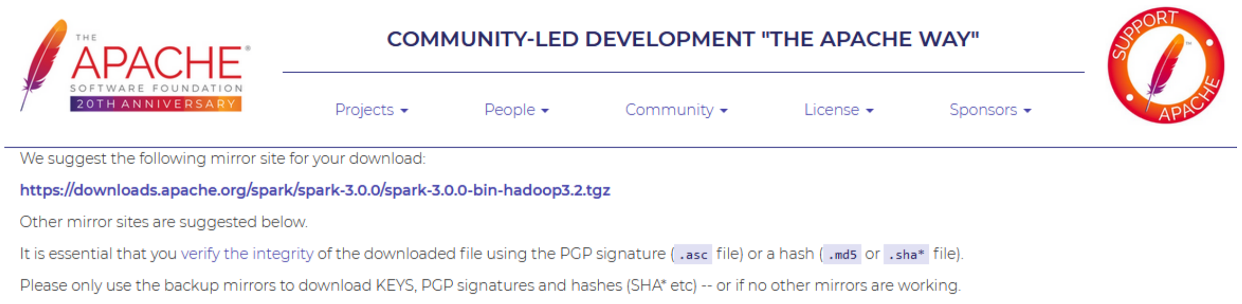


Download Apache Spark™

1. Choose a Spark release: 3.0.0 (Jun 18 2020)
2. Choose a package type: Pre-built for Apache Hadoop 3.2 and later
3. Download Spark: spark-3.0.0-bin-hadoop3.2.tgz
4. Verify this release using the 3.0.0 signatures, checksums and project release KEYS.

Note that, Spark 2.x is pre-built with Scala 2.11 except version 2.4.2, which is pre-built with Scala 2.12. Spark 3.0+ is pre-built with Scala 2.12.

   - Select the Download link of step 3 in above image. It will give the link as shown below.


THE APACHE SOFTWARE FOUNDATION 20TH ANNIVERSARY
COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"
Projects
People
Community
License
Sponsors
SUPPORT APACHE

We suggest the following mirror site for your download:

https://downloads.apache.org/spark/spark-3.0.0/spark-3.0.0-bin-hadoop3.2.tgz

Other mirror sites are suggested below.

It is essential that you verify the integrity of the downloaded file using the PGP signature (`.asc` file) or a hash (`.md5` or `.sha*` file).

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or if no other mirrors are working.

   - Download spark using below command:

```
wget https://downloads.apache.org/spark/spark-3.0.0/spark-3.0.0-bin-hadoop3.2.tgz
```

3. **Extract the Spark tarball using below command:**

```
tar xvf spark-3.0.0-bin-hadoop3.2.tgz
```

4. Rename the extracted folder to 'spark'

```
mv spark-3.0.0-bin-hadoop3.2 spark
```

5. Set Spark environment
    - Open your bashrc configuration file with below command:

      ```
      sudo nano ~/.bashrc
      ```

    - Add below lines:

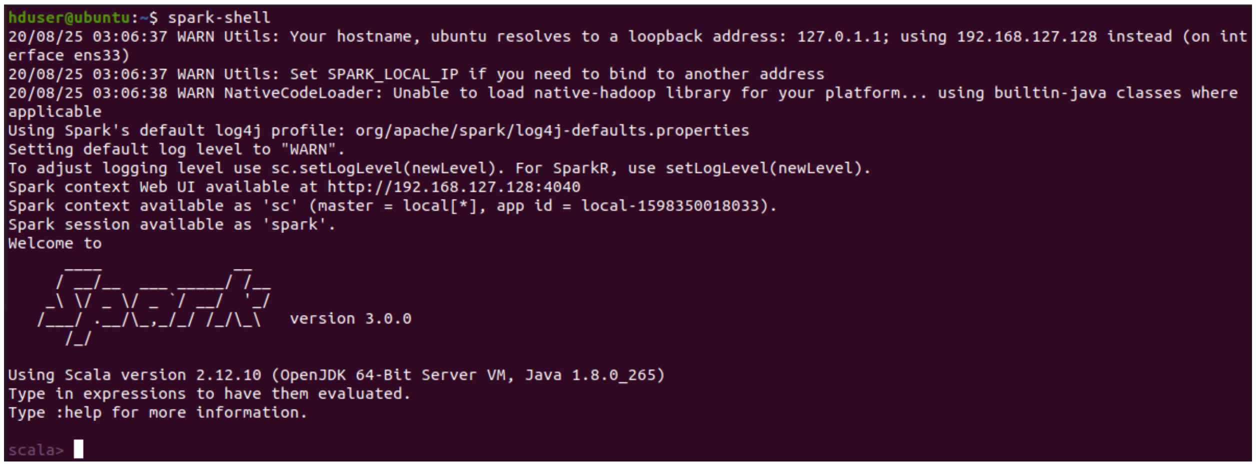
      ```
      export SPARK_HOME=/home/hduser/spark
      export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin
      export PYSPARK_PYTHON=/usr/bin/python3
      ```

    - Activate the changes.

      ```
      source ~/.bashrc
      ```

6. Test Installation:
    - Test Spark Shell

Use the *spark-shell* command to access Spark Shell. It should show screen similar to below screen:



```
hduser@ubuntu:~$ spark-shell
20/08/25 03:06:37 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.1.1; using 192.168.127.128 instead (on int
erface ens33)
20/08/25 03:06:37 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
20/08/25 03:06:38 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://192.168.127.128:4040
Spark context available as 'sc' (master = local[*], app id = local-1598350018033).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.0.0
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_265)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

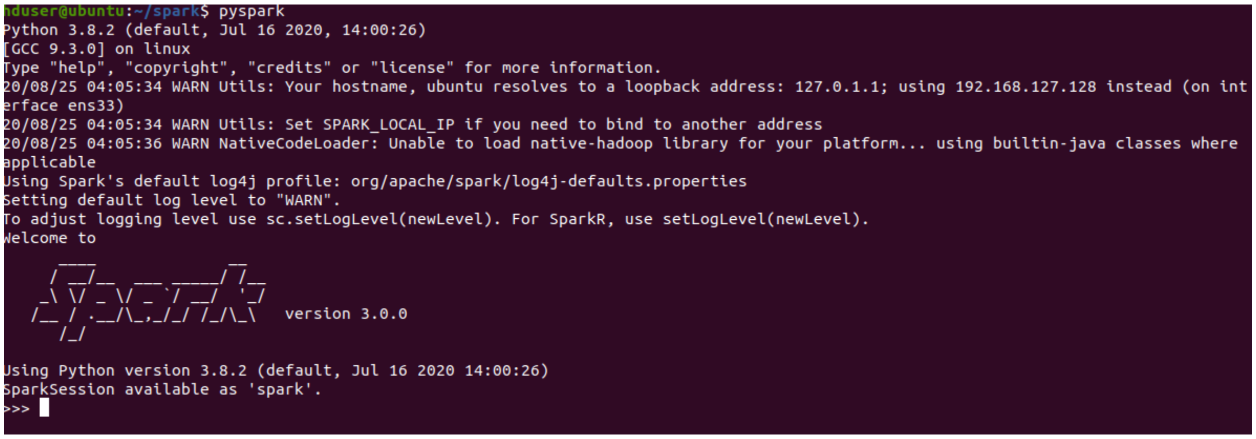
Type: **q** and press **Enter** to exit Scala.

- Test Python in Spark

If you do not want to use the default Scala interface, you can switch to Python. Make sure you quit Scala and then run this command:

```
pyspark
```

It should show screen similar to shown below:



```
hduser@ubuntu:~/spark$ pyspark
Python 3.8.2 (default, Jul 16 2020, 14:00:26)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
20/08/25 04:05:34 WARN Utils: Your hostname, ubuntu resolves to a loopback address: 127.0.1.1; using 192.168.127.128 instead (on int
erface ens33)
20/08/25 04:05:34 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
20/08/25 04:05:36 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).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.0.0
      /_/

Using Python version 3.8.2 (default, Jul 16 2020 14:00:26)
SparkSession available as 'spark'.
>>> 
```

Type exit() to exit PySpark Shell.

7. Test with RDD and Dataframe
   Let's create RDD and Dataframe in Scala based Spark-Shell.

- We can create RDD in 3 ways, we will use one way to create RDD.

Define any list then parallelize it. It will create RDD. Below are the codes. Copy paste it one by one on the command line.

```
val nums = Array(1,2,3,5,6)
val rdd = sc.parallelize(nums)
```

Above will create RDD.

- Now we will create a Data frame from RDD. Follow the below steps to create Dataframe.

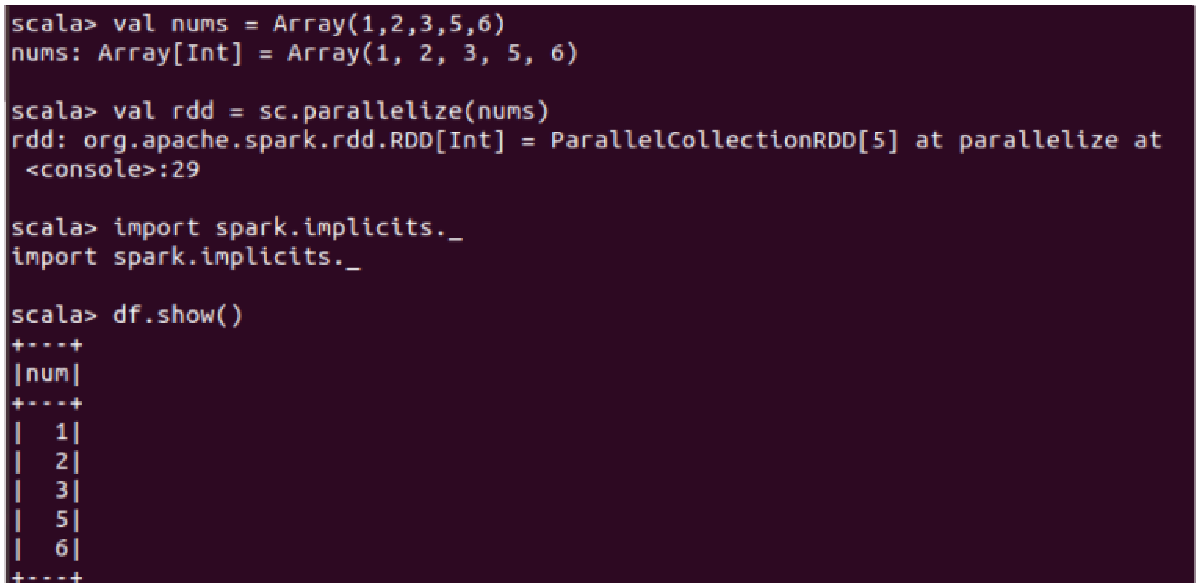
```
import spark.implicits._
val df = rdd.toDF("num")
```

Above code will create Dataframe with num as a column.

To display the data in Dataframe use below command

```
df.show()
```

Below is the screenshot of the above code.



```
scala> val nums = Array(1,2,3,5,6)
nums: Array[Int] = Array(1, 2, 3, 5, 6)

scala> val rdd = sc.parallelize(nums)
rdd: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[5] at parallelize at
 <console>:29

scala> import spark.implicits._
import spark.implicits._

scala> df.show()
+---+
|num|
+---+
|  1|
|  2|
|  3|
|  5|
|  6|
+---+
```

Now we have successfully installed spark on Ubuntu System and verified it with RDD and Dataframe.