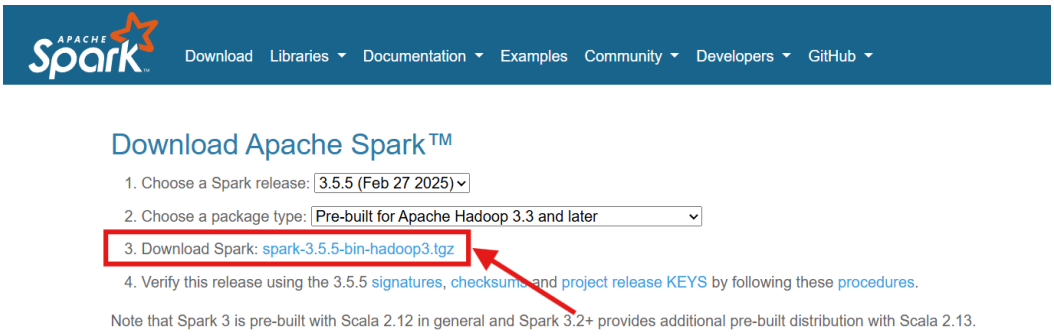
Install Spark in Ubuntu

---

1. Download Apache Spark.

    a. Go to the downloads page.
    b. Select the latest release. For the package type, choose 'Pre-built for Apache Hadoop'.
    c. Page will look like this:



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

    e. Download spark using below command:

```
~$ su - hdoop

~$ wget https://dlcdn.apache.org/spark/spark-3.5.5/spark-3.5.5-bin-hadoop3.tgz
```

```
hdoop@Siddhartha-Shakya:~$ wget https://dlcdn.apache.org/spark/spark-3.5.5/s
park-3.5.5-bin-hadoop3.tgz
--2025-04-13 05:26:52--  https://dlcdn.apache.org/spark/spark-3.5.5/spark-3.
5.5-bin-hadoop3.tgz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::6
44
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... conn
ected.
HTTP request sent, awaiting response... 200 OK
Length: 400724056 (382M) [application/x-gzip]
Saving to: ‘spark-3.5.5-bin-hadoop3.tgz’

tgz                 46%[======>          ] 176.70M  19.1MB/s    eta 8s
```

2. Extract the Spark tarball using below command:

```
~$ tar xvf spark-3.5.5-bin-hadoop3.tgz
```

3. Rename the extracted folder to ‘**spark**’.

```
~$ mv spark-3.5.5-bin-hadoop3 spark
```

4. Set Spark environment

   a. Open your **bashrc** configuration file with below command:

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

   b. Add below lines:

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

```
#Hadoop Related Options
export HADOOP_HOME=/home/hdoop/hadoop-3.2.4
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
export YARN_EXAMPLES=/home/hdoop/hadoop-3.2.4/share/hadoop/mapreduce

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

c. Activate the changes.

```
~$ source ~/.bashrc
```

5. Test Installation:

a. Test Spark Shell

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

```
hdoop@Siddhartha-Shakya:~$ spark-shell
25/04/13 05:45:22 WARN Utils: Your hostname, Siddhartha-Shakya resolves to a loopback a
25/04/13 05:45:22 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another addres
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newL
25/04/13 05:45:27 WARN NativeCodeLoader: Unable to load native-hadoop library for your
Spark context Web UI available at http://10.255.255.254:4040
Spark context available as 'sc' (master = local[*], app id = local-1744502427954).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.5.5
      /_/

Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 1.8.0_442)
Type in expressions to have them evaluated.
Type :help for more information.

scala> |
```

Type :q or :quit to exit scala.

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:

```
hdoop@Siddhartha-Shakya:~$ pyspark
Python 3.10.12 (main, Feb  4 2025, 14:57:36) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
25/04/13 05:50:46 WARN Utils: Your hostname, Siddhartha-Shakya resolves to a loopback address:
25/04/13 05:50:46 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).
25/04/13 05:50:46 WARN NativeCodeLoader: Unable to load native-hadoop library for your platforr
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.5.5
      /_/

Using Python version 3.10.12 (main, Feb  4 2025 14:57:36)
Spark context Web UI available at http://10.255.255.254:4040
Spark context available as 'sc' (master = local[*], app id = local-1744502747512).
SparkSession available as 'spark'.
>>> |
```

Type exit() to exit PySpark Shell.

6. Test with **RDD** and Dataframe

RDD stands for **Resilient Distributed Dataset**. It's the fundamental data structure in Apache Spark.

   a. 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.

```
~$ spark-shell

scala > val nums = Array(1,2,3,5,6)

scala> val rdd = sc.parallelize(nums)
```

Above will create RDD.

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

```
scala> import spark.implicits._

scala> val df = rdd.toDF("num")
```

Above code will create a Dataframe with num as a column.

To display the data in Dataframe use below command

```
scala> df.show()
```

Below is the screenshot of the above code

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