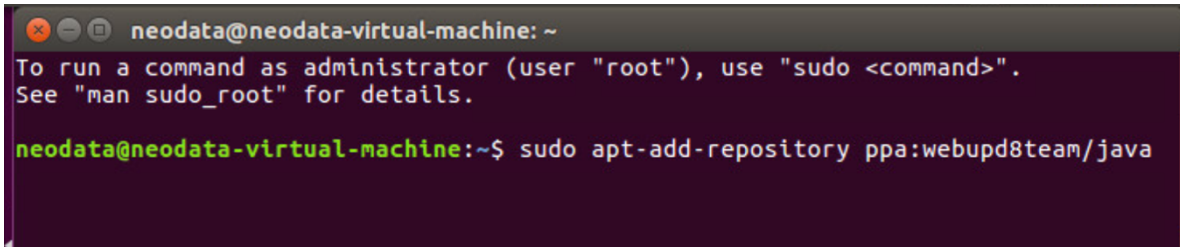
Installing and Building Spark

Author: Rahul Sharma (sharma1@student.unimelb.edu.au)

Go to terminal:



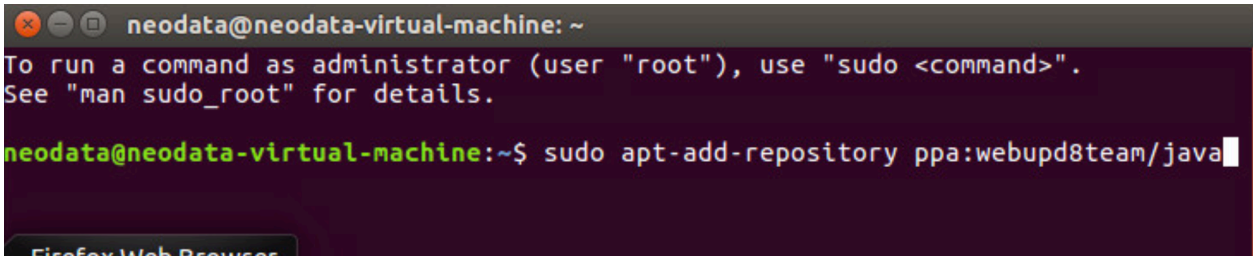
Spark uses Scala for its processing and analysis capabilities which in turn uses the Java Virtual Machine (JVM) for compiling its code. Hence, we will install Java first to progress through the building procedure.

Note: You need the JDK and not just the JRE. If you install the JRE only then you cannot build spark.

1. **To Install Java:**

Type the commands:

```
sudo apt-add-repository ppa:webupd8team/java
```



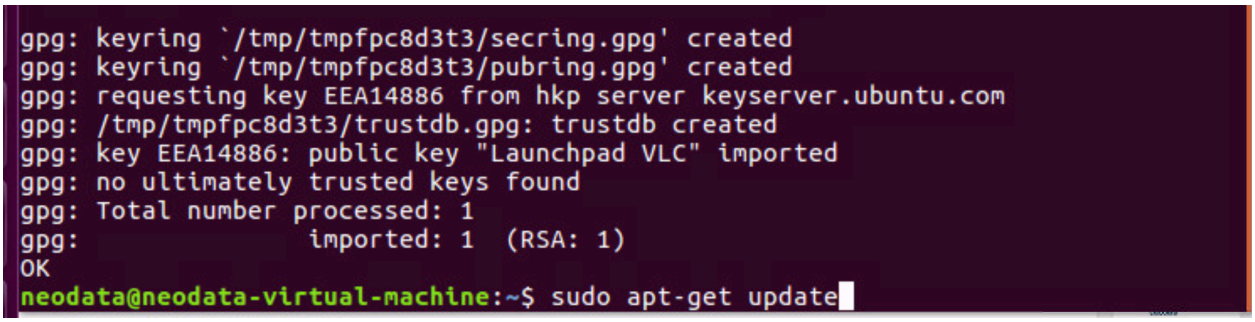
It will prompt you to provide a password. Enter the password: Exalytics12$ and press enter.

Press Enter.

Now type:

sudo apt-get update

Now press Enter



Finally, type:

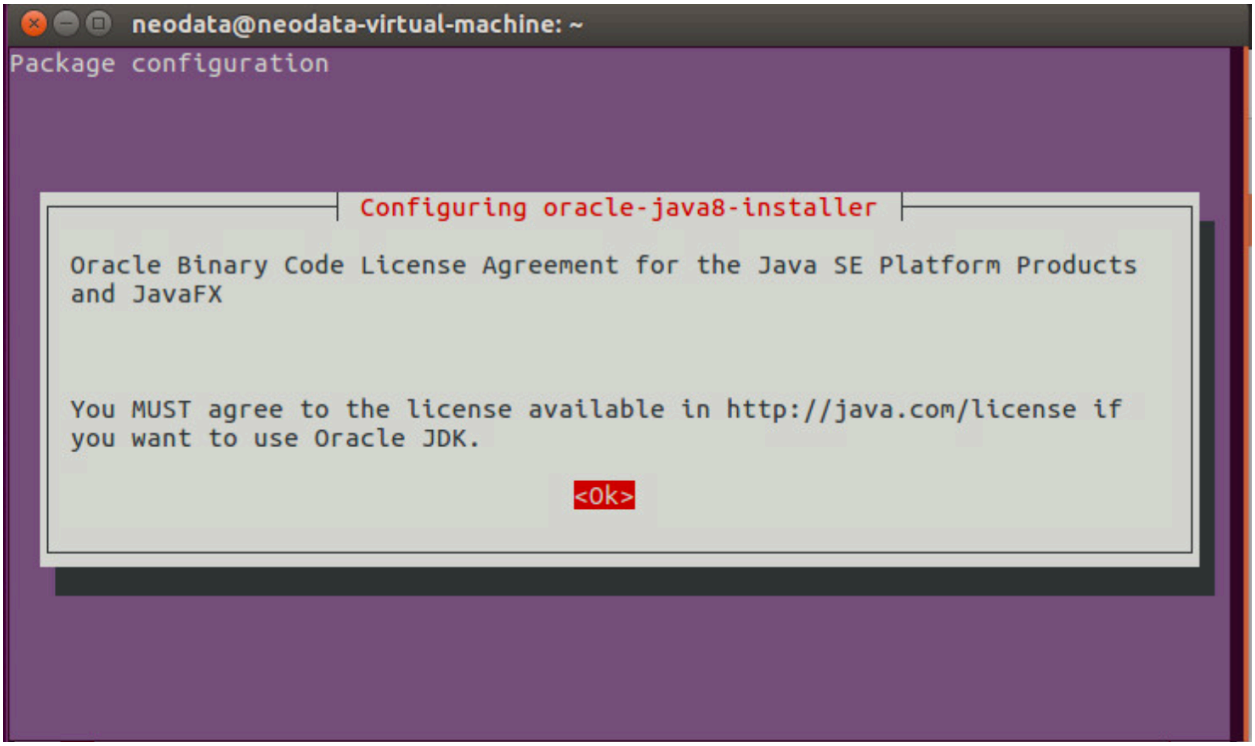
sudo apt-get install oracle-java8-installer

```
Get:11 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial/main i386 Package
s [2,840 B]
Get:12 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial/main Translation-
en [1,260 B]
Fetched 1,608 kB in 1s (856 kB/s)
Reading package lists... Done
neodata@neodata-virtual-machine:~$ sudo apt-get install oracle-java8-installer
```

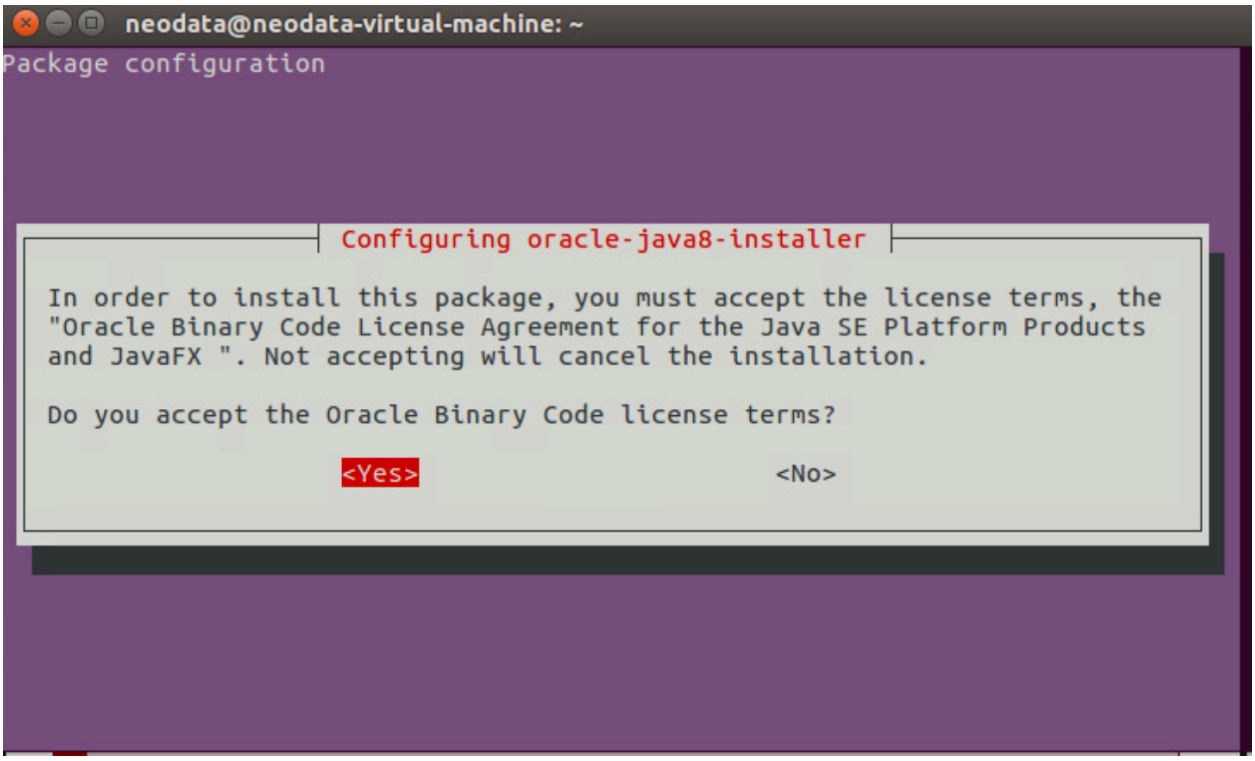
Press Enter.

```
neodata@neodata-virtual-machine:~$ sudo apt-get install oracle-java8-installer
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  gsfonts-x11 java-common
Suggested packages:
  binfmt-support visualvm ttf-baekmuk | ttf-unfonts | ttf-unfonts-core
  ttf-kochi-gothic | ttf-sazanami-gothic ttf-kochi-mincho
  | ttf-sazanami-mincho ttf-arphic-uming
The following NEW packages will be installed:
  gsfonts-x11 java-common oracle-java8-installer
0 to upgrade, 3 to newly install, 0 to remove and 63 not to upgrade.
Need to get 38.6 kB of archives.
After this operation, 227 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

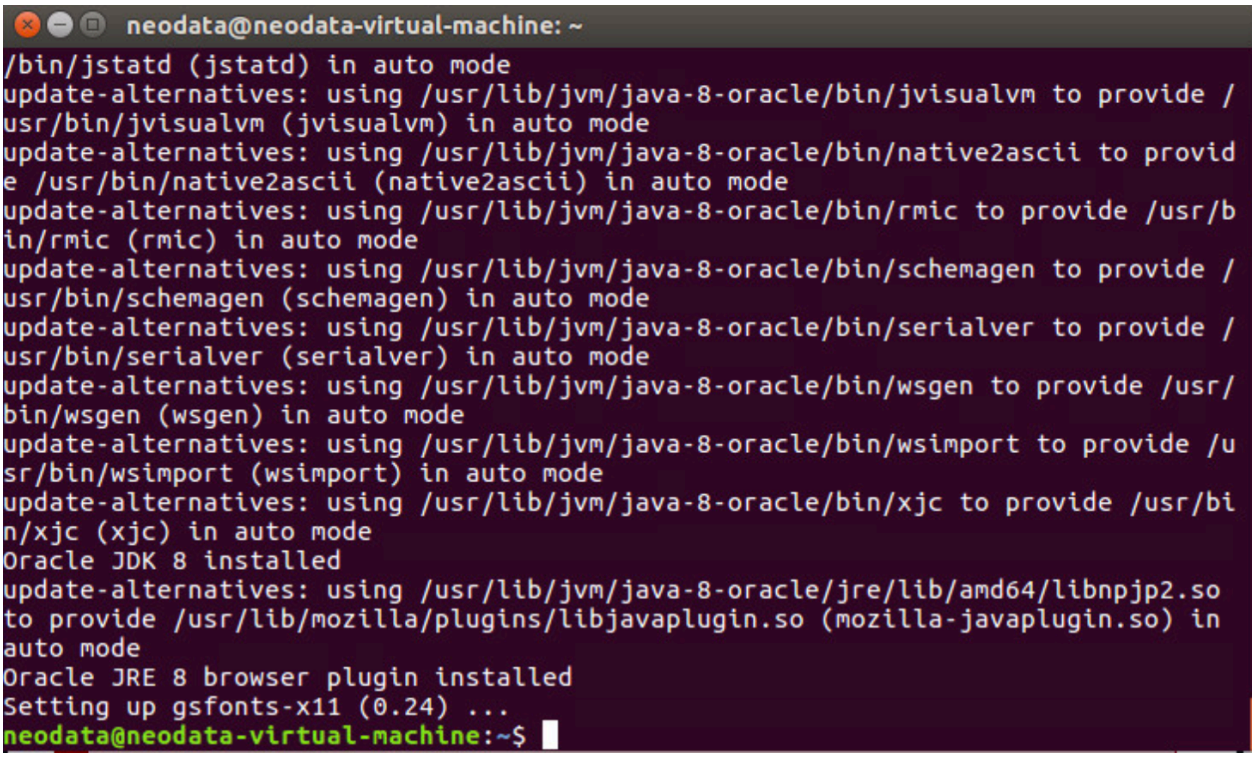
Type y and press enter. You will be led to the License agreement page from here.



Press Enter.



Go to Yes and press Enter.

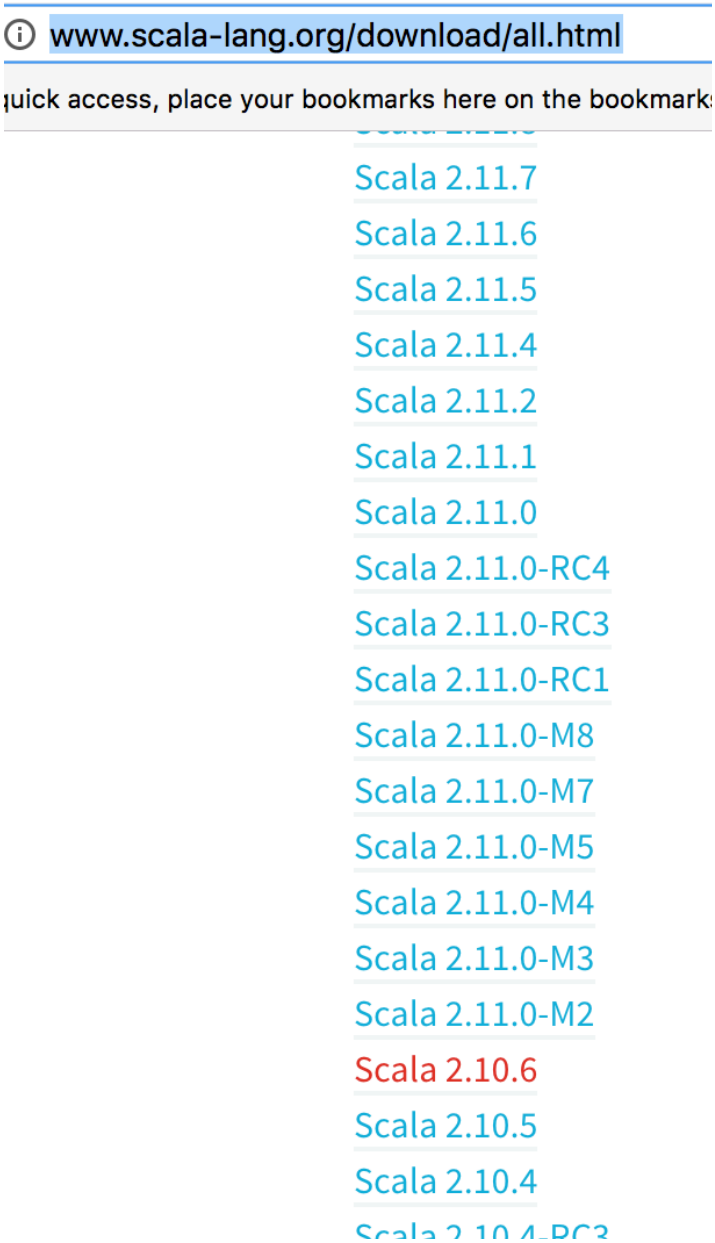


Once you install java, we have to download and install Scala.

2. Download and Install Scala:

To install Scala, we first need to go to their Downloads tab on their site: http://www.scala-lang.org/download/all.html

Pick a version of Scala that you wish to use for installation. I am using Scala 2.10.6



Click on Scala 2.10.6 and you will be led to a downloads page. Click on Option 1: Scala 2.10.6 and it will transition to a Download option. This will download a tar file for that Scala version to your Downloads folder. (Used Scala 2.10.6!!!!!!!!)

Now type:

sudo mkdir /usr/local/src/scala

This will create a folder for scala to be stored in.

```
update-alternatives: using /usr/lib/jvm/java-8-oracle/jre/lib/amd64/libnpjp2.so
to provide /usr/lib/mozilla/plugins/libjavaplugin.so (mozilla-javaplugin.so) in
auto mode
Oracle JRE 8 browser plugin installed
Setting up gsfonts-x11 (0.24) ...
neodata@neodata-virtual-machine:~$ sudo mkdir /usr/local/src/scala
neodata@neodata-virtual-machine:~$
```

Now direct to the Downloads folder.

```
Oracle JRE 8 browser plugin installed
Setting up gsfonts-x11 (0.24) ...
neodata@neodata-virtual-machine:~$ sudo mkdir /usr/local/src/scala
neodata@neodata-virtual-machine:~$ cd Downloads/
neodata@neodata-virtual-machine:~/Downloads$ ls
scala-2.10.6.tgz
neodata@neodata-virtual-machine:~/Downloads$
```

We need to extract the Scala files from the tgz file. To extract the files, type:

sudo tar -xvf scala-2.10.6.tgz -C /usr/local/src/scala/

```
neodata@neodata-virtual-machine:~$ cd Downloads/
neodata@neodata-virtual-machine:~/Downloads$ ls
scala-2.10.6.tgz
neodata@neodata-virtual-machine:~/Downloads$ sudo tar -xvf scala-2.10.6.tgz -C /
usr/local/src/scala/
```

Press Enter. The files will be extracted in the scala folder we created in /usr/local/src/scala. Next, we need to add the path for scala into the bashrc file for the bash shell to automatically set the path for scala whenever the terminal is opened. To get to the bashrc file, type cd, following which, type:

nano .bashrc

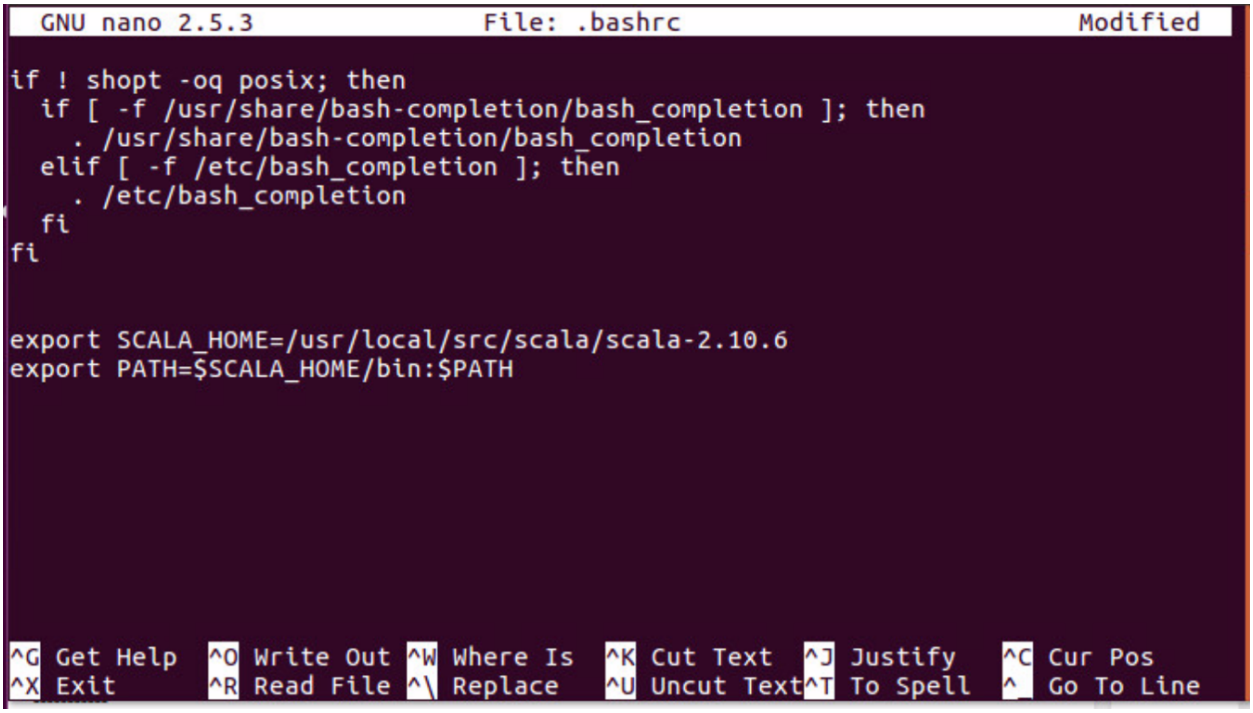
Nano is a file editor in Ubuntu and is pre-installed like vi. You can also use vi or vim for this.

```
scala-2.10.6/lib/scala-swing.jar
scala-2.10.6/lib/scala-actors-migration.jar
scala-2.10.6/lib/typesafe-config.jar
scala-2.10.6/lib/scala-actors.jar
scala-2.10.6/lib/jline.jar
scala-2.10.6/lib/scala-library.jar
scala-2.10.6/lib/scala-compiler.jar
scala-2.10.6/lib/akka-actors.jar
neodata@neodata-virtual-machine:~/Downloads$ cd
neodata@neodata-virtual-machine:~$ nano .bashrc
```

Once you open the bashrc file, get to the end of the file and type the following lines:

```
export SCALA_HOME=/usr/local/src/scala/scala-2.10.6
export PATH=$SCALA_HOME/bin:$PATH
```

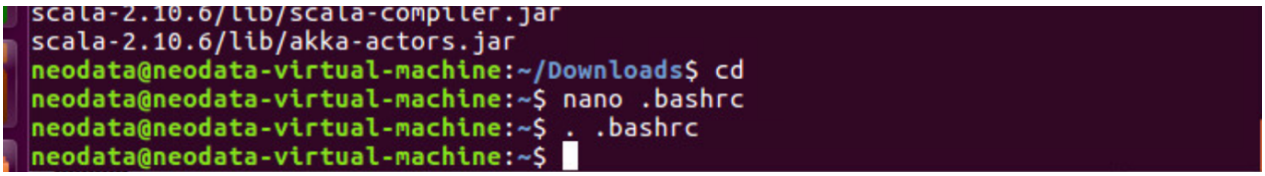
This would add the path for scala into the bash shell.



Once you have typed these lines, choose the command for exiting the editor as shown on the screen. This will prompt the editor to ask you as to whether you wish to save the file or not. Type y. It will now ask you as to whether you wish to save it in the current file (it will specify the name of the file being modified). Type Enter.

To refresh the shell and enable the changes, type:

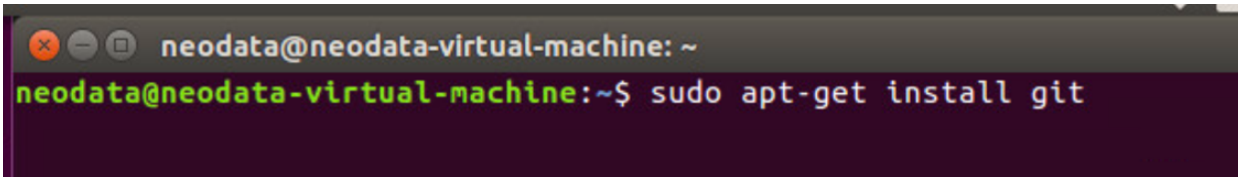
. .bashrc



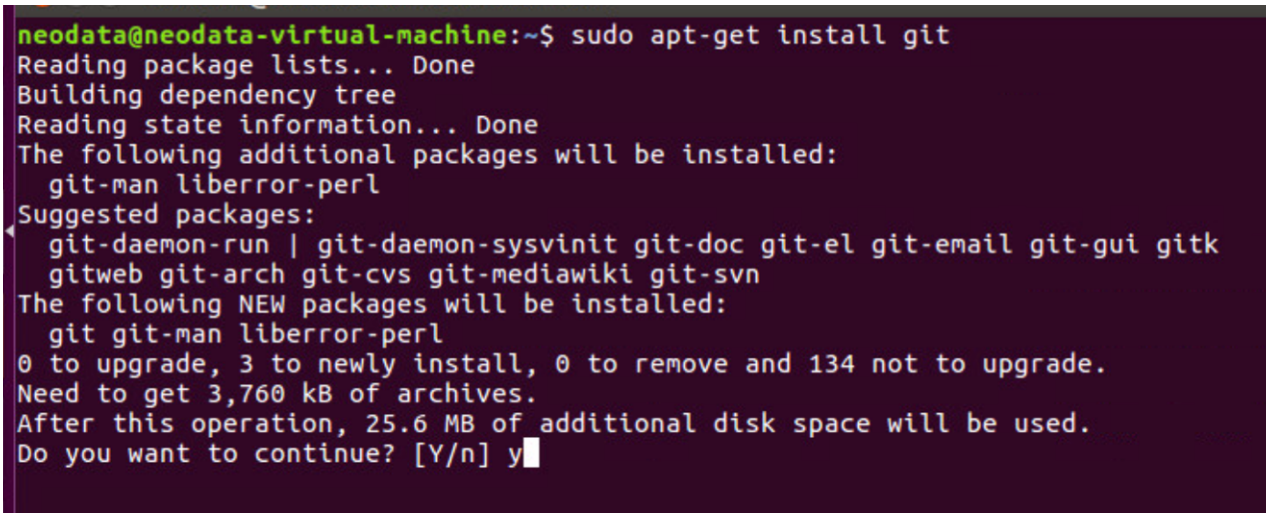
3. **Install Git:**

Type:

sudo apt-get install git

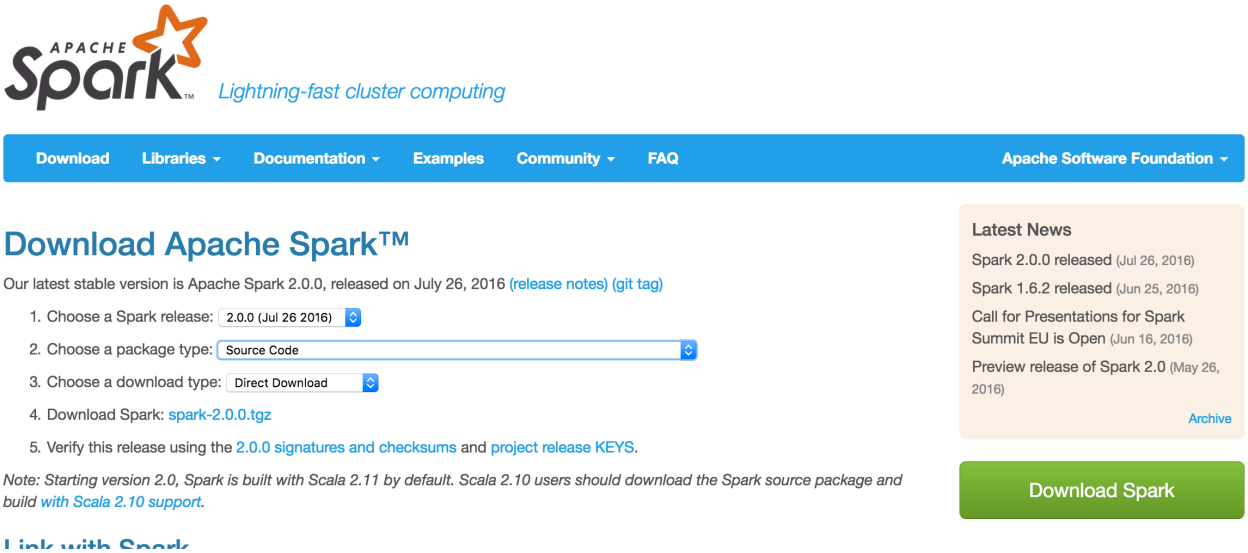


Press Enter. It will prompt you to verify if you wish to continue with the installation. Type y and press Enter.



4. **Download and build Spark:**

Go to: http://spark.apache.org/downloads.html



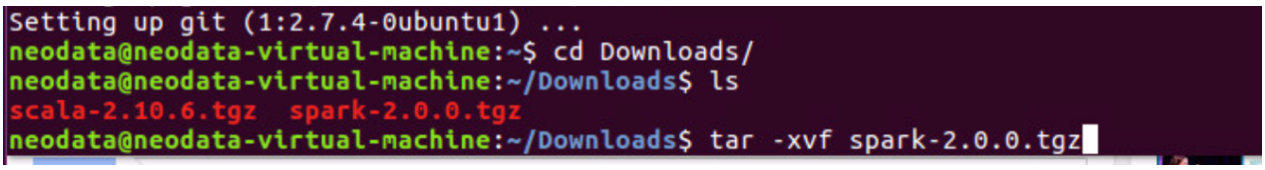
Pick the latest release: 2.0.0

For Package type: I do not have Hadoop installed for now so I will run Spark on Standalone mode. Click on the dropdown arrow and pick Source code. Click on the Download spark link in Step 4.

Now redirect to the Downloads folder where you downloaded Spark.

The downloaded file is a tgz file. We need to unzip this and get the source code content to build. Type:

tar -xvf spark-2.0.0.tgz

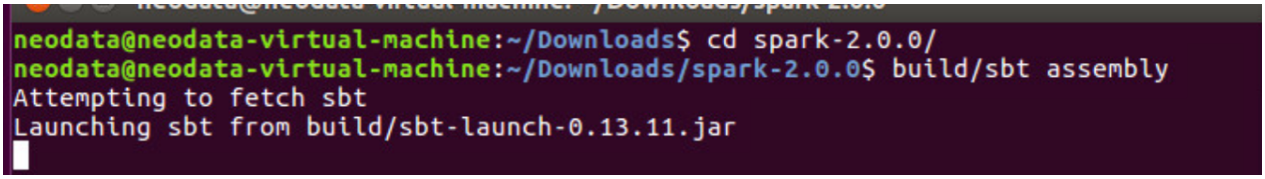


Now press Enter.

Once the file has been unzipped – direct yourself into Sparks folder. We have access to Spark's files now, the next step we need to take is to build it. We can build spark by typing:

build/sbt assembly

Press Enter.



```
h strategy 'first'
[warn] Merging 'org/apache/hadoop/yarn/util/package-info.class' with strategy 'f
irst'
[warn] Merging 'rootdoc.txt' with strategy 'first'
[warn] Strategy 'discard' was applied to a file
[warn] Strategy 'filterDistinctLines' was applied to 7 files
[warn] Strategy 'first' was applied to 125 files
[info] SHA-1: cdb633dcba09c350856778b185573a3046fe991a
[info] Packaging /home/neodata/Downloads/spark-2.0.0/external/kafka-0-10-assembl
y/target/scala-2.11/spark-streaming-kafka-0-10-assembly-2.0.0.jar ...
[info] Done packaging.
[success] Total time: 2188 s, completed 08/09/2016 2:00:32 PM
neodata@neodata-virtual-machine:~/Downloads/spark-2.0.0$
```

It takes about 45 minutes to build spark, following which type:

build/sbt package

```
[warn] Merging 'rootdoc.txt' with strategy 'first'
[warn] Strategy 'discard' was applied to a file
[warn] Strategy 'filterDistinctLines' was applied to 7 files
[warn] Strategy 'first' was applied to 125 files
[info] SHA-1: cdb633dcba09c350856778b185573a3046fe991a
[info] Packaging /home/neodata/Downloads/spark-2.0.0/external/kafka-0-10-assembl
y/target/scala-2.11/spark-streaming-kafka-0-10-assembly-2.0.0.jar ...
[info] Done packaging.
[success] Total time: 2188 s, completed 08/09/2016 2:00:32 PM
neodata@neodata-virtual-machine:~/Downloads/spark-2.0.0$ build/sbt package
```

This will build the scala version present in the spark distribution (scala 2.11) along with the relevant jar files.

```
[warn]
[warn] Multiple main classes detected.  Run 'show discoveredMainClasses' to see
the list
[info] Packaging /home/neodata/Downloads/spark-2.0.0/repl/target/scala-2.11/spar
k-repl_2.11-2.0.0.jar ...
[info] Done packaging.
[info] Packaging /home/neodata/Downloads/spark-2.0.0/examples/target/scala-2.11/
jars/spark-examples_2.11-2.0.0.jar ...
[info] Done packaging.
[success] created output: /home/neodata/Downloads/spark-2.0.0/assembly/target
[info] Packaging /home/neodata/Downloads/spark-2.0.0/assembly/target/scala-2.11/
jars/spark-assembly_2.11-2.0.0.jar ...
[info] Done packaging.
[success] Total time: 3646 s, completed 08/09/2016 3:04:07 PM
neodata@neodata-virtual-machine:~/Downloads/spark-2.0.0$
```

Now, spark has been built completely and is ready for use. You can directly access Spark's terminal by typing:

bin/spark-shell

Press Enter. This will launch the spark shell.

```
k-repl_2.11-2.0.0.jar ...
[info] Done packaging.
[info] Packaging /home/neodata/Downloads/spark-2.0.0/examples/target/scala-2.11/
jars/spark-examples_2.11-2.0.0.jar ...
[info] Done packaging.
[success] created output: /home/neodata/Downloads/spark-2.0.0/assembly/target
[info] Packaging /home/neodata/Downloads/spark-2.0.0/assembly/target/scala-2.11/
jars/spark-assembly_2.11-2.0.0.jar ...
[info] Done packaging.
[success] Total time: 3646 s, completed 08/09/2016 3:04:07 PM
neodata@neodata-virtual-machine:~/Downloads/spark-2.0.0$ bin/spark-shell
```

```
[info] Done packaging.
[success] Total time: 3646 s, completed 08/09/2016 3:04:07 PM
neodata@neodata-virtual-machine:~/Downloads/spark-2.0.0$ bin/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).
```

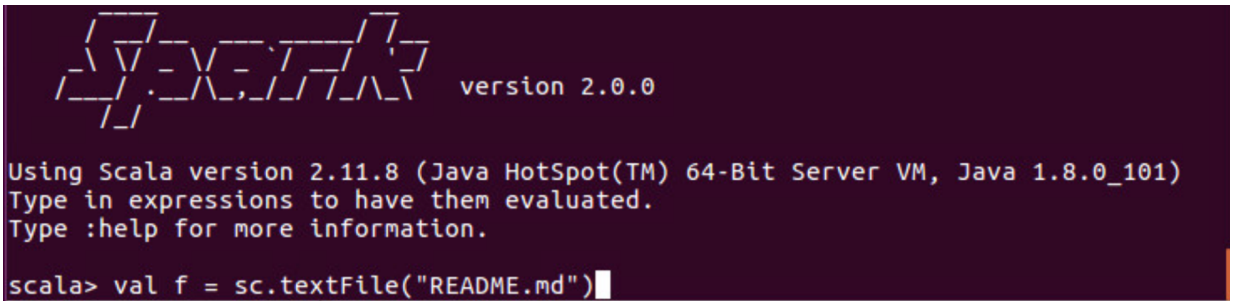
```
r your platform... using builtin-java classes where applicable
16/09/08 15:08:53 WARN Utils: Your hostname, neodata-virtual-machine resolves to
 a loopback address: 127.0.1.1; using 192.168.1.176 instead (on interface ens192
)
16/09/08 15:08:53 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
16/09/08 15:08:57 WARN SparkContext: Use an existing SparkContext, some configur
ation may not take effect.
Spark context Web UI available at http://192.168.1.176:4040
Spark context available as 'sc' (master = local[*], app id = local-1473311336243
).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.0.0
      /_/

Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_101)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

The commands written in this shell are written using Scala. For example:

There is a README.md file in the spark folder. We can read that folder by typing:

val f = sc.textFile(“README.md”)

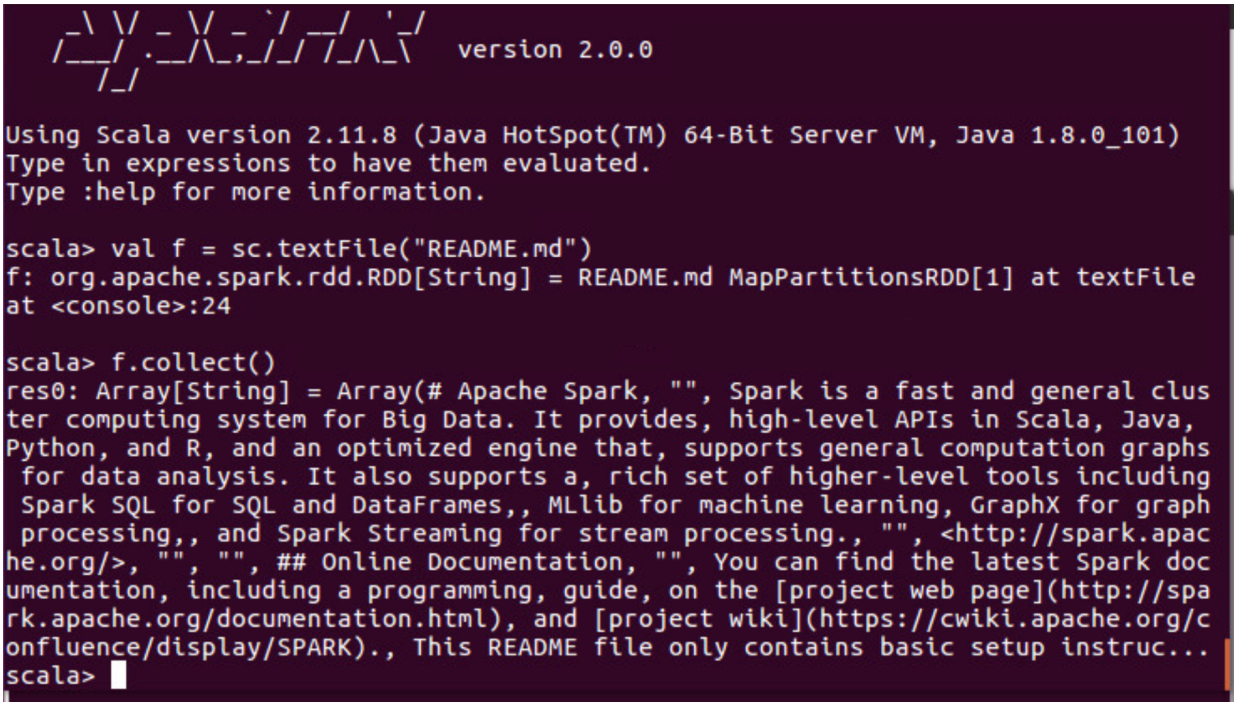
Type enter.



Following which, type:

f.collect()

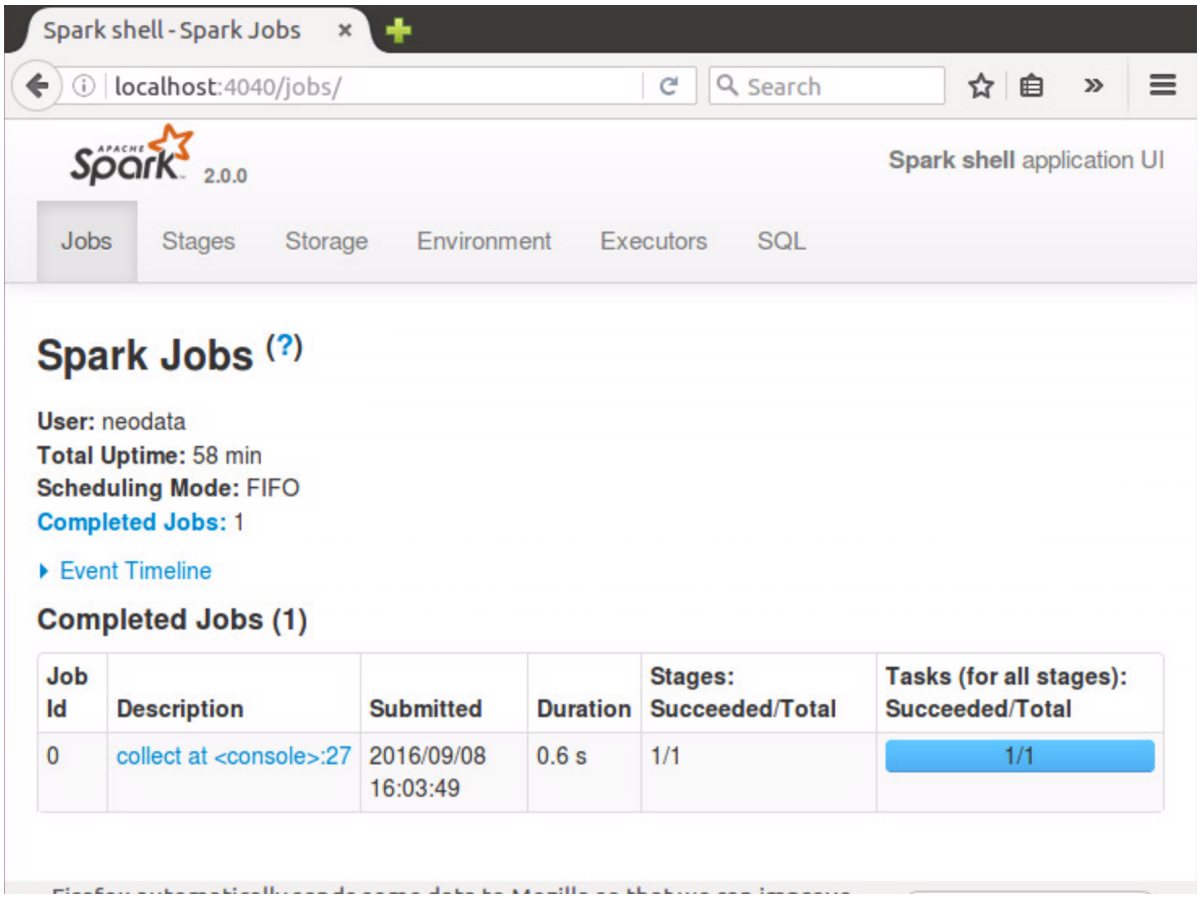
This will display the information in the .md file.



Apart from this, Spark also has a web portal to check various stats about jobs currently running. This web portal can be accessed by opening a web browser and typing:

localhost:4040

This will open up the web portal specifying information on jobs.


Spark shell - Spark Jobs
localhost:4040/jobs/
Search
Spark 2.0.0
Spark shell application UI
Jobs
Stages
Storage
Environment
Executors
SQL
Spark Jobs (?)
User: neodata
Total Uptime: 58 min
Scheduling Mode: FIFO
Completed Jobs: 1
Event Timeline
Completed Jobs (1)
Job Id
Description
Submitted
Duration
Stages: Succeeded/Total
Tasks (for all stages): Succeeded/Total
0
collect at <console>:27
2016/09/08 16:03:49
0.6 s
1/1
1/1