

Single Node Apache Spark 1.6.1 Setup and Configuration on Ubuntu 14.04



Data Science Lab, The Department of Computer Science,
KSKV Kachchh University.

Web: <http://cs.kutchuni.edu.in>

The MIT License (MIT)

Copyright (c) 2016. Data Science Lab, University of Kachchh.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1) Follow Document '*Apache Hadoop Single Node Setup- CSKSKV-v1.3.docx*' and setup everything accordingly.

2) `~/bashrc`

Edit above file and add below environment variables.

```
#SPARK VARIABLES START
```

```
export SPARK_DIST_CLASSPATH=$(hadoop --config /usr/local/hadoop/etc/hadoop classpath)
```

```
export SPARK_HOME=/usr/local/spark-1.6.1
```

```
#SPARK VARIABLES END
```

3) Download Apache Spark unzip and move to `/usr/local/spark`

```
wget http://d3kbcqa49mib13.cloudfront.net/spark-1.6.1-bin-hadoop2.6.tgz
```

```
$ tar xzf spark-1.6.1-bin-hadoop2.6.tgz
```

```
$ sudo mv spark-1.6.1 /usr/local/hadoop
```

4) Launching Spark-Shell

```
/usr/local/spark$ bin/spark-shell
```

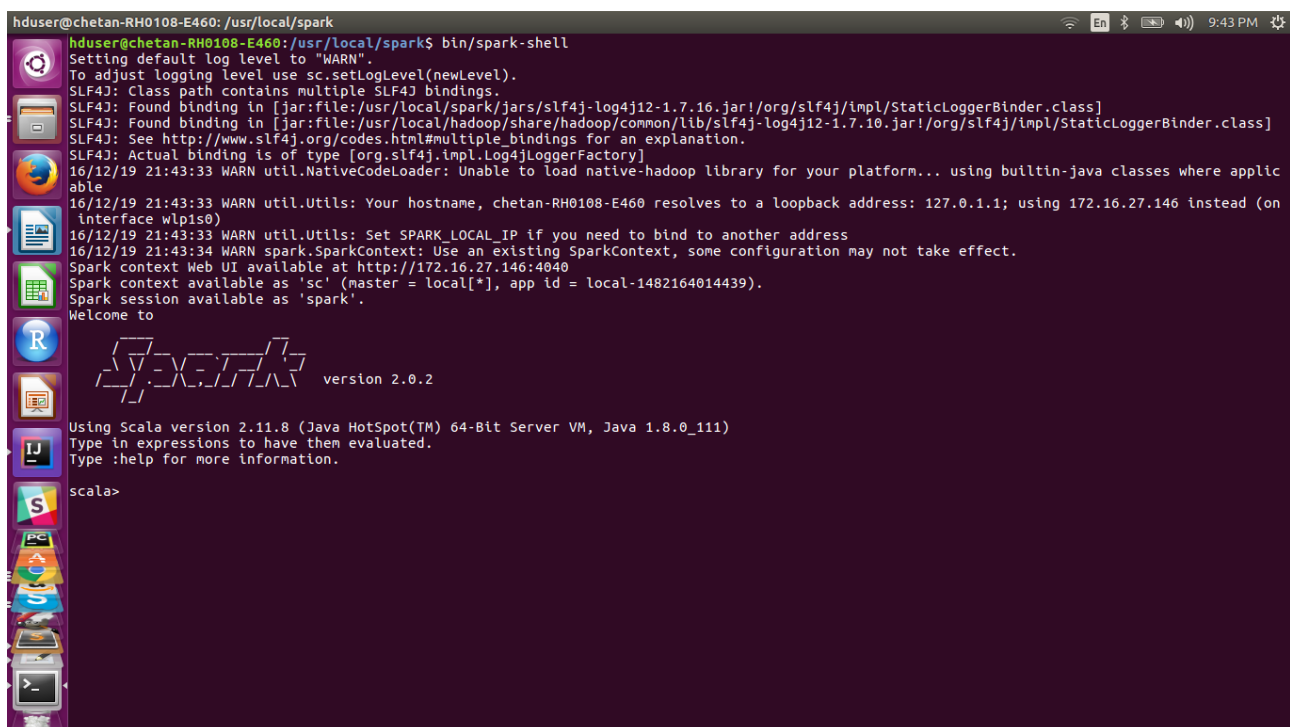
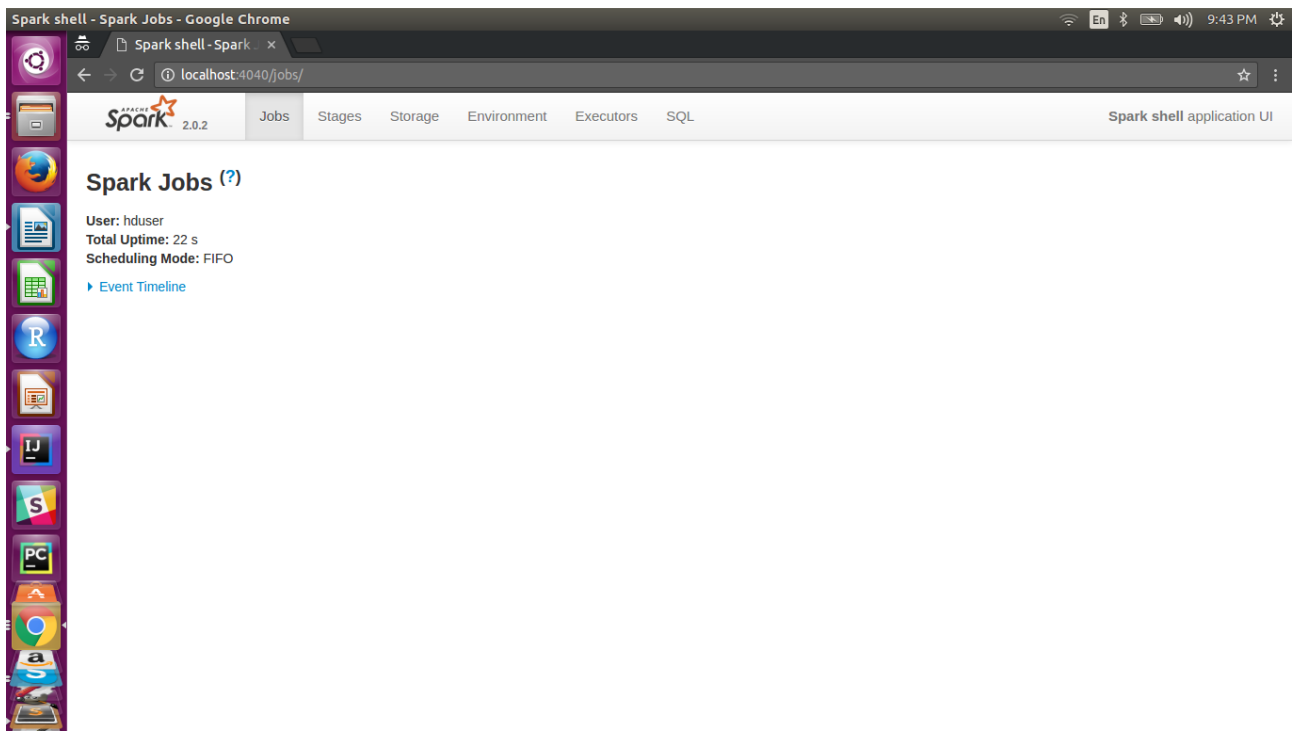
The screenshot shows a terminal window titled 'hduser@chetan-RH0108-E460: /usr/local/spark'. The user has executed 'bin/spark-shell'. The terminal output includes: 'Setting default log level to "WARN". To adjust logging level use sc.setLogLevel(newLevel). SLF4J: Class path contains multiple SLF4J bindings. SLF4J: Found binding in [jar:file:/usr/local/spark/jars/slf4j-log4j12-1.7.16.jar!/org/slf4j/impl/StaticLoggerBinder.class] SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class] SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation. SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]'. There are several WARN messages about hostname resolution and Spark context configuration. The terminal then displays the 'Welcome to' message, the 'DREXEL' logo, 'version 2.0.2', and 'Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_111)'. It prompts the user to 'Type in expressions to have them evaluated.' and 'Type :help for more information.' The prompt 'scala>' is visible at the bottom.

Figure 1: Spark Shell Launcher Screen with Scala Interpreter



Spark UI Screen for Jobs Monitoring

<http://localhost:4040/jobs/>

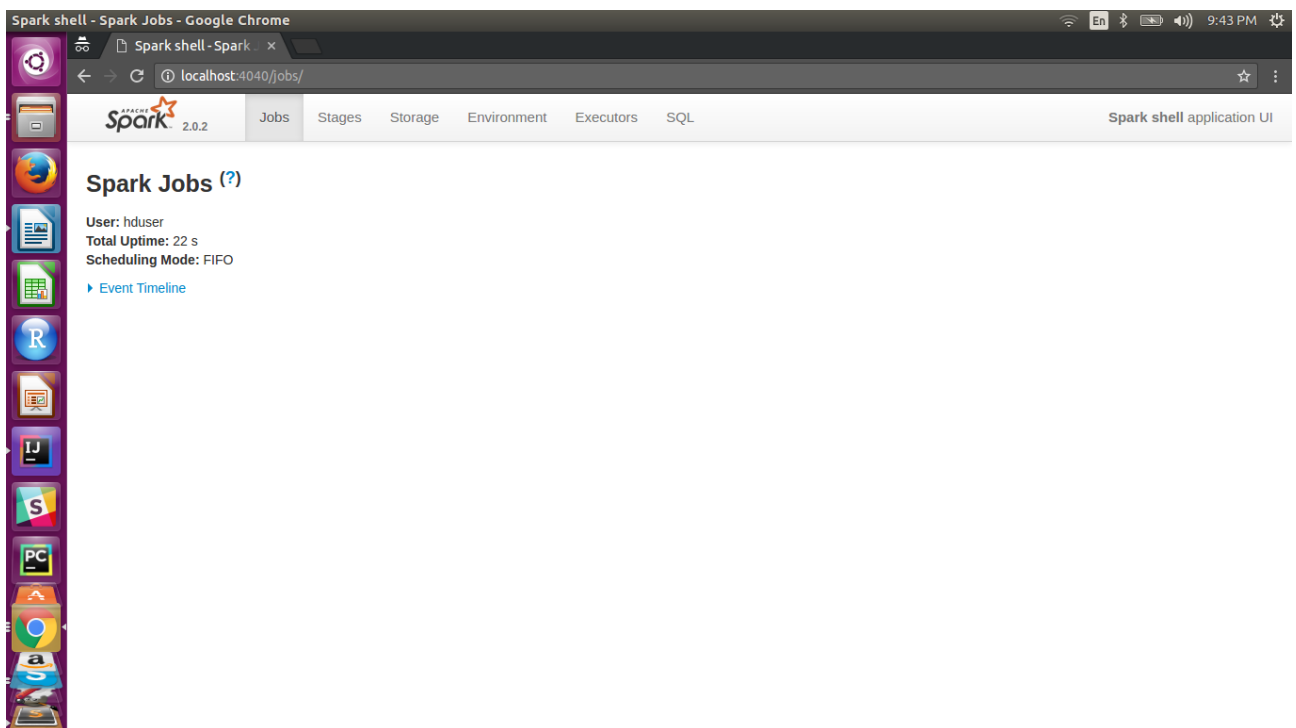


Figure 2: Spark UI Job Monitoring Screen

5) Submit Spark Jobs

There are 3 ways to submit Spark Jobs

- 1) Spark-Shell
- 2) Spark Submit

3) Spark Launcher

Spark Submit Command

```
./bin/spark-submit \  
--class <main-class> \  
--master <master-url> \  
--deploy-mode <deploy-mode> \  
--conf <key>=<value> \  
... # other options  
<application-jar> \  
[application-arguments]
```

Example,

```
bin/spark-submit --class hbase.spark.chetan.com.SparkHbaseJob /home/chetan/hbase-  
spark/Sparkassembly-1.0.jar
```

Where Jar must be Uber(Super) Jar.