

Hands-on Exercises – AMPCAMP 4

Note: Prerequisite – Cluster is running fine with data loaded.

These exercises are divided into sections designed to give a hands-on experience with various software components of the Berkeley Data Analytics Stack (BDAS). For Spark, we will walk you through using the Spark shell for interactive exploration of data. You have the choice of doing the exercises using Scala or using Python. For Shark, you will be using SQL in the Shark console to interactively explore the same data. The advanced modules may use other data sets, such as Twitter data for Spark Streaming.

Link to exercise : <http://ampcamp.berkeley.edu/4/exercises/logging-into-the-cluster.html>

Get the public ip of the masternode from any one way described in the eariler document.

The screenshot shows the AWS Management Console with the EC2 Dashboard. The 'Instances' tab is active, showing a list of EC2 instances. The instance 'i-2495ecc9' is selected, and its details are displayed. The instance is in a 'running' state, with a public IP of 54.165.177.11 and a public DNS of ec2-54-165-177-11.compute-1.amazonaws.com. The instance type is m1.xlarge, and it is using the 'mysparkkey' key pair. The monitoring is disabled, and it was launched on October 10, 2014, at 11:01:40 AM. The security groups associated with the instance are 'ampcamp3-master'.

Login to master node (use your public IP).

```
ssh -i mysparkkey.pem root@ec2-54-165-177-11.compute-1.amazonaws.com
```

The screenshot shows a terminal window with the following text:

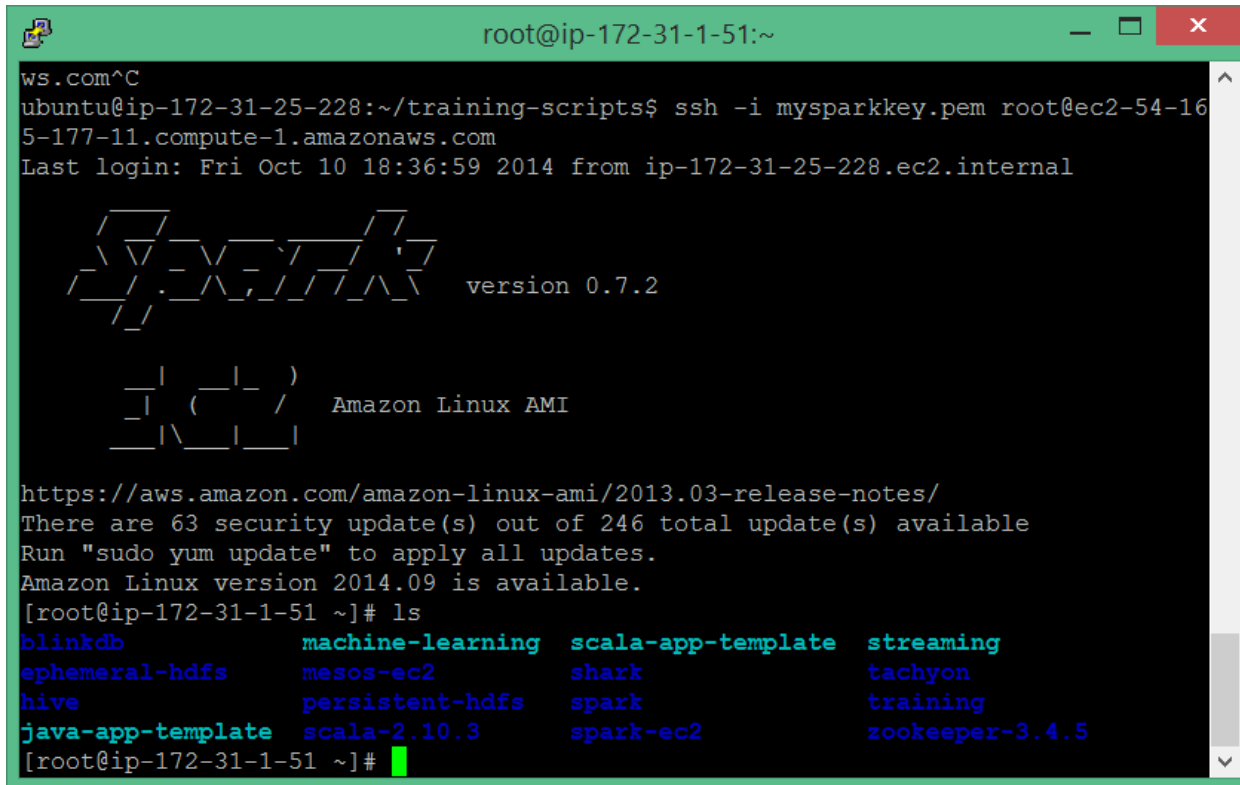
```
root@ip-172-31-1-51:~  
Connection to ec2-54-165-177-11.compute-1.amazonaws.com closed.  
SUCCESS: Data copied successfully! You can login to the master at ec2-54-165-177-11.compute-1.amazonaws.com  
ubuntu@ip-172-31-25-228:~/training-scripts$ ec2-54-165-177-11.compute-1.amazonaws.com^C  
ubuntu@ip-172-31-25-228:~/training-scripts$ ssh -i mysparkkey.pem root@ec2-54-165-177-11.compute-1.amazonaws.com  
Last login: Fri Oct 10 18:36:59 2014 from ip-172-31-25-228.ec2.internal  
  
  _ _ _ _ _  
 / _ _ _ _ \  version 0.7.2  
/_ _ _ _ _ \  
/ _ _ _ _ \  Amazon Linux AMI  
/_ _ _ _ _ \  
  
https://aws.amazon.com/amazon-linux-ami/2013.03-release-notes/  
There are 63 security update(s) out of 246 total update(s) available  
Run "sudo yum update" to apply all updates.  
Amazon Linux version 2014.09 is available.  
[root@ip-172-31-1-51 ~]#
```

1. Cluster Details

Your cluster contains 6 m1.xlarge Amazon EC2 nodes. One of these 6 nodes is the master node, responsible for scheduling tasks as well as maintaining the HDFS metadata (a.k.a. HDFS name node). The other 5 are the slave nodes on which tasks are actually executed. You will mainly interact with the master node. If you haven't already, let's ssh onto the master node (see instructions above).

Once you've used SSH to log into the master, run the `ls` command and you will see a number of directories. Some of the more important ones are listed below:

`ls`



```
root@ip-172-31-1-51:~
ws.com^C
ubuntu@ip-172-31-25-228:~/training-scripts$ ssh -i mysparkkey.pem root@ec2-54-16
5-177-11.compute-1.amazonaws.com
Last login: Fri Oct 10 18:36:59 2014 from ip-172-31-25-228.ec2.internal

  _ _ _ _ _
 /_/_/_/_/_\  version 0.7.2
/_/_/_/_/_\

 _ | _ | _ )
 _ | ( _ | /  Amazon Linux AMI
 _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-ami/2013.03-release-notes/
There are 63 security update(s) out of 246 total update(s) available
Run "sudo yum update" to apply all updates.
Amazon Linux version 2014.09 is available.
[root@ip-172-31-1-51 ~]# ls
blinkdb          machine-learning  scala-app-template  streaming
ephemeral-hdfs   mesos-ec2        shark               tachyon
hive             persistent-hdfs   spark               training
java-app-template scala-2.10.3      spark-ec2           zookeeper-3.4.5
[root@ip-172-31-1-51 ~]#
```

```
cat spark-ec2/slaves
```

For stand-alone Spark programs, you will have to know the Spark cluster URL. You can find that in `spark-ec2/cluster-url`:

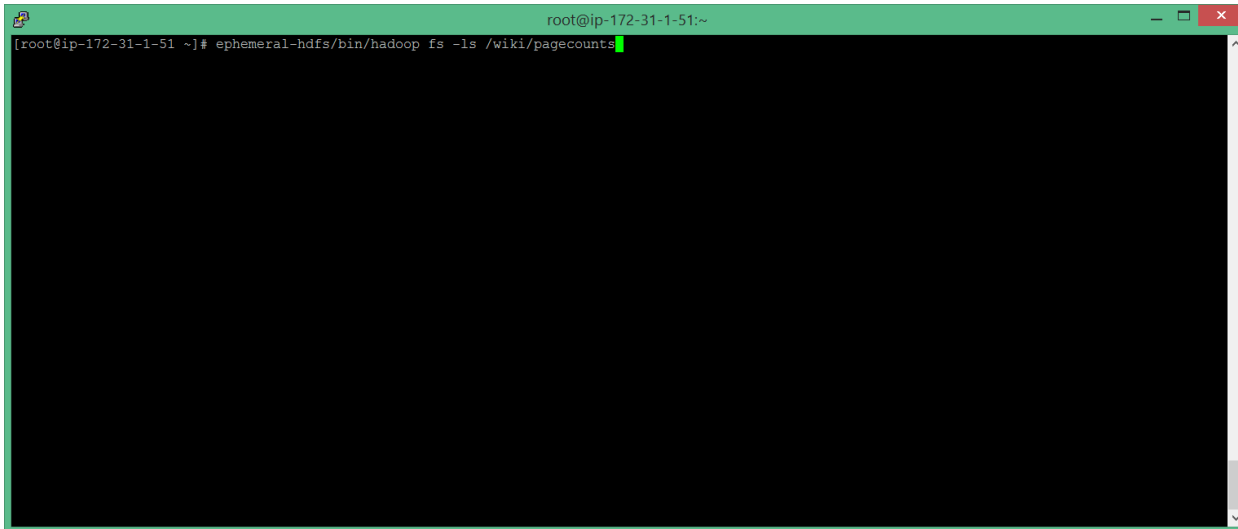
```
root@ip-172-31-1-51:~  
_/_/  
_ | ( _ | _ )  
_ | ( _ / Amazon Linux AMI  
_ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-ami/2013.03-release-notes/  
There are 63 security update(s) out of 246 total update(s) available  
Run "sudo yum update" to apply all updates.  
Amazon Linux version 2014.09 is available.  
[root@ip-172-31-1-51 ~]# ls  
blinkdb machine-learning scala-app-template streaming  
ephemeral-hdfs mesos-ec2 shark tachyon  
hive persistent-hdfs spark training  
java-app-template scala-2.10.3 spark-ec2 zookeeper-3.4.5  
[root@ip-172-31-1-51 ~]# cat spark-ec2/slaves  
ec2-54-172-163-14.compute-1.amazonaws.com  
ec2-54-172-166-224.compute-1.amazonaws.com  
ec2-54-172-166-222.compute-1.amazonaws.com  
ec2-54-172-166-220.compute-1.amazonaws.com  
ec2-54-172-166-223.compute-1.amazonaws.com  
[root@ip-172-31-1-51 ~]# cat spark-ec2/cluster-url  
spark://ec2-54-165-177-11.compute-1.amazonaws.com:7077  
[root@ip-172-31-1-51 ~]#
```

```
spark://ec2-54-165-177-11.compute-1.amazonaws.com:7077
```

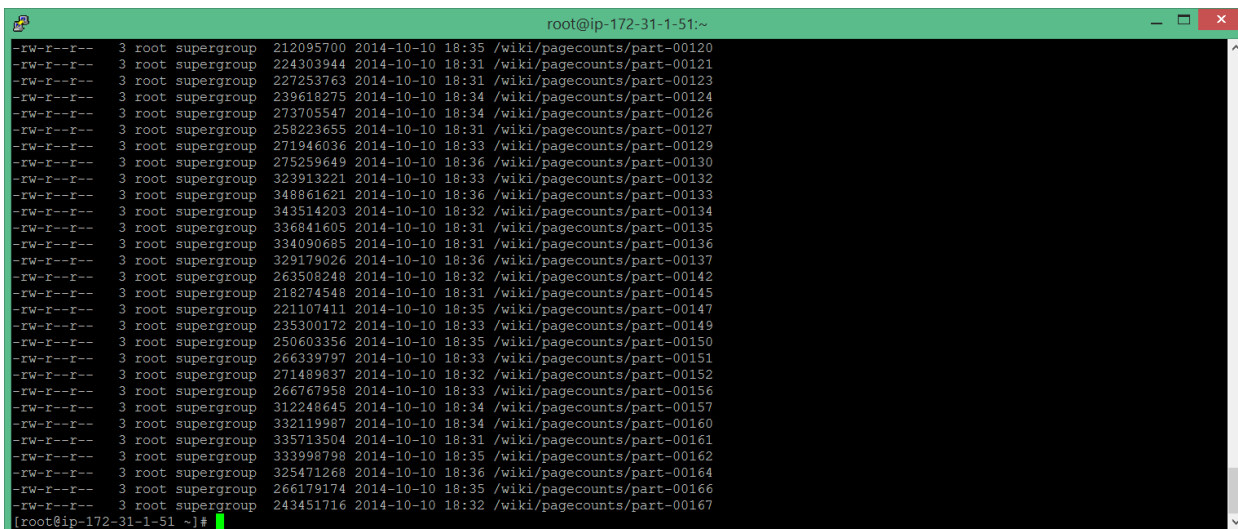
2. Dataset For Exploration

Among other datasets, your HDFS cluster should come preloaded with 20GB of Wikipedia traffic statistics data obtained from <http://aws.amazon.com/datasets/4182> . To make the analysis feasible (within the short timeframe of the exercise), we took three days worth of data (May 5 to May 7, 2009; roughly 20G and 329 million entries). You can list the files:

```
ephemeral-hdfs/bin/hadoop fs -ls /wiki/pagecounts
```



```
root@ip-172-31-1-51:~  
[root@ip-172-31-1-51 ~]# ephemeral-hdfs/bin/hadoop fs -ls /wiki/pagecounts
```

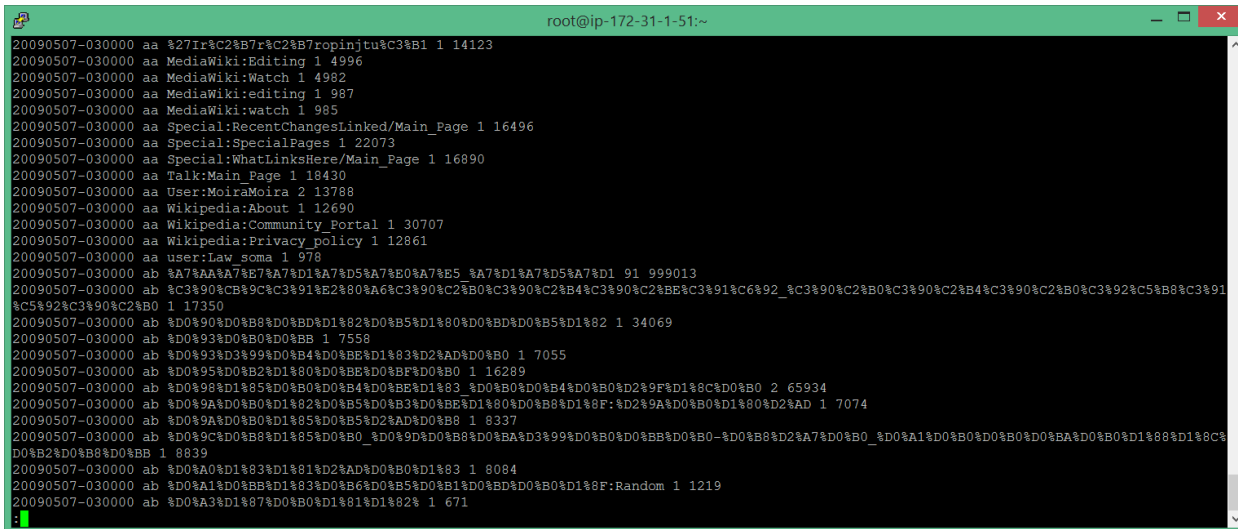


```
root@ip-172-31-1-51:~  
-rw-r--r-- 3 root supergroup 212095700 2014-10-10 18:35 /wiki/pagecounts/part-00120  
-rw-r--r-- 3 root supergroup 224303944 2014-10-10 18:31 /wiki/pagecounts/part-00121  
-rw-r--r-- 3 root supergroup 227253763 2014-10-10 18:31 /wiki/pagecounts/part-00123  
-rw-r--r-- 3 root supergroup 239618275 2014-10-10 18:34 /wiki/pagecounts/part-00124  
-rw-r--r-- 3 root supergroup 273705547 2014-10-10 18:34 /wiki/pagecounts/part-00126  
-rw-r--r-- 3 root supergroup 258223655 2014-10-10 18:31 /wiki/pagecounts/part-00127  
-rw-r--r-- 3 root supergroup 271946036 2014-10-10 18:33 /wiki/pagecounts/part-00129  
-rw-r--r-- 3 root supergroup 275259649 2014-10-10 18:36 /wiki/pagecounts/part-00130  
-rw-r--r-- 3 root supergroup 323913221 2014-10-10 18:33 /wiki/pagecounts/part-00132  
-rw-r--r-- 3 root supergroup 348861621 2014-10-10 18:36 /wiki/pagecounts/part-00133  
-rw-r--r-- 3 root supergroup 343514203 2014-10-10 18:32 /wiki/pagecounts/part-00134  
-rw-r--r-- 3 root supergroup 336841605 2014-10-10 18:31 /wiki/pagecounts/part-00135  
-rw-r--r-- 3 root supergroup 334090685 2014-10-10 18:31 /wiki/pagecounts/part-00136  
-rw-r--r-- 3 root supergroup 329179026 2014-10-10 18:36 /wiki/pagecounts/part-00137  
-rw-r--r-- 3 root supergroup 263508248 2014-10-10 18:32 /wiki/pagecounts/part-00142  
-rw-r--r-- 3 root supergroup 218274548 2014-10-10 18:31 /wiki/pagecounts/part-00145  
-rw-r--r-- 3 root supergroup 221107411 2014-10-10 18:35 /wiki/pagecounts/part-00147  
-rw-r--r-- 3 root supergroup 235300172 2014-10-10 18:33 /wiki/pagecounts/part-00149  
-rw-r--r-- 3 root supergroup 250603356 2014-10-10 18:35 /wiki/pagecounts/part-00150  
-rw-r--r-- 3 root supergroup 266339797 2014-10-10 18:33 /wiki/pagecounts/part-00151  
-rw-r--r-- 3 root supergroup 271489837 2014-10-10 18:32 /wiki/pagecounts/part-00152  
-rw-r--r-- 3 root supergroup 266767958 2014-10-10 18:33 /wiki/pagecounts/part-00156  
-rw-r--r-- 3 root supergroup 312248645 2014-10-10 18:34 /wiki/pagecounts/part-00157  
-rw-r--r-- 3 root supergroup 332119987 2014-10-10 18:34 /wiki/pagecounts/part-00160  
-rw-r--r-- 3 root supergroup 335713504 2014-10-10 18:31 /wiki/pagecounts/part-00161  
-rw-r--r-- 3 root supergroup 333998798 2014-10-10 18:35 /wiki/pagecounts/part-00162  
-rw-r--r-- 3 root supergroup 325471268 2014-10-10 18:36 /wiki/pagecounts/part-00164  
-rw-r--r-- 3 root supergroup 266179174 2014-10-10 18:35 /wiki/pagecounts/part-00166  
-rw-r--r-- 3 root supergroup 243451716 2014-10-10 18:32 /wiki/pagecounts/part-00167  
[root@ip-172-31-1-51 ~]#
```

There are 74 files (2 of which are intentionally left empty). - Found 49 items

The data are partitioned by date and time. Each file contains traffic statistics for all pages in a specific hour. Let's take a look at the file:

ephemeral-hdfs/bin/hadoop fs -cat /wiki/pagecounts/part-00147 | less



```
root@ip-172-31-1-51:~
20090507-030000 aa %27lr%C2%B7r%C2%B7ropinjt%C3%B1 1 14123
20090507-030000 aa MediaWiki:Editing 1 4996
20090507-030000 aa MediaWiki:Watch 1 4982
20090507-030000 aa MediaWiki:editing 1 987
20090507-030000 aa MediaWiki:watch 1 985
20090507-030000 aa Special:RecentChangesLinked/Main_Page 1 16496
20090507-030000 aa Special:SpecialPages 1 22073
20090507-030000 aa Special:WhatLinksHere/Main_Page 1 16890
20090507-030000 aa Talk:Main_Page 1 18430
20090507-030000 aa User:MoiriMoiri 2 13788
20090507-030000 aa Wikipedia:About 1 12690
20090507-030000 aa Wikipedia:Community Portal 1 30707
20090507-030000 aa Wikipedia:Privacy policy 1 12861
20090507-030000 aa user:Law_soma 1 978
20090507-030000 ab %A7%AA%A7%E7%A7%D1%A7%D5%A7%E0%A7%E5 %A7%D1%A7%D5%A7%D1 91 999013
20090507-030000 ab %C3%90%CB%9C%C3%91%E2%80%A6%C3%90%C2%B0%C3%90%C2%B4%C3%90%C2%B0%C3%92%C5%B8%C3%91
%C5%92%C3%90%C2%B0 1 17350
20090507-030000 ab %D0%90%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82 1 34069
20090507-030000 ab %D0%93%D0%B0%D0%BB 1 7558
20090507-030000 ab %D0%93%D3%99%D0%B4%D0%BE%D1%83%D2%AD%D0%B0 1 7055
20090507-030000 ab %D0%95%D0%B2%D1%80%D0%BE%D0%BF%D0%B0 1 16289
20090507-030000 ab %D0%98%D1%85%D0%B0%D0%B4%D0%BE%D1%83 %D0%B0%D0%B4%D0%B0%D2%9F%D1%8C%D0%B0 2 65934
20090507-030000 ab %D0%9A%D0%B0%D1%82%D0%B5%D0%B3%D0%BE%D1%80%D0%B8%D1%8F:%D2%9A%D0%B0%D1%80%D2%AD 1 7074
20090507-030000 ab %D0%9A%D0%B0%D1%85%D0%B5%D2%AD%D0%B8 1 8337
20090507-030000 ab %D0%9C%D0%B8%D1%85%D0%B0 %D0%9D%D0%B8%D0%BA%D3%99%D0%B0%D0%BB%D0%B0-%D0%B8%D2%A7%D0%B0_%D0%A1%D0%B0%D0%B0%D0%BA%D0%B0%D1%88%D1%8C%
%D0%B2%D0%B8%D0%BB 1 8839
20090507-030000 ab %D0%A0%D1%83%D1%81%D2%AD%D0%B0%D1%83 1 8084
20090507-030000 ab %D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:Random 1 1219
20090507-030000 ab %D0%A3%D1%87%D0%B0%D1%81%D1%82% 1 671
:
```

The first few lines of the file are copied here:

```
20090507-030000 aa %27lr%C2%B7r%C2%B7ropinjt%C3%B1 1 14123

20090507-030000 aa MediaWiki:Editing 1 4996

20090507-030000 aa MediaWiki:Watch 1 4982

20090507-030000 aa MediaWiki:editing 1 987

20090507-030000 aa MediaWiki:watch 1 985

20090507-030000 aa Special:RecentChangesLinked/Main_Page 1 16496
```

Each line, delimited by a space, contains stats for one page. The schema is:

<date_time> <project_code> <page_title> <num_hits> <page_size>

The <date_time> field specifies a date in the YYYYMMDD format (year, month, day) followed by a hyphen and then the hour in the HHmmSS format (hour, minute, second). There is no information in mmSS. The <project_code> field contains information about the language of the pages. For example, project code "en" indicates an English page. The <page_title> field gives the title of the Wikipedia

page. The `<num_hits>` field gives the number of page views in the hour-long time slot starting at `<data_time>`. The `<page_size>` field gives the size in bytes of the Wikipedia page.

To quit less, stop viewing the file, and return to the command line, press `q`.

Introduction to SCALA

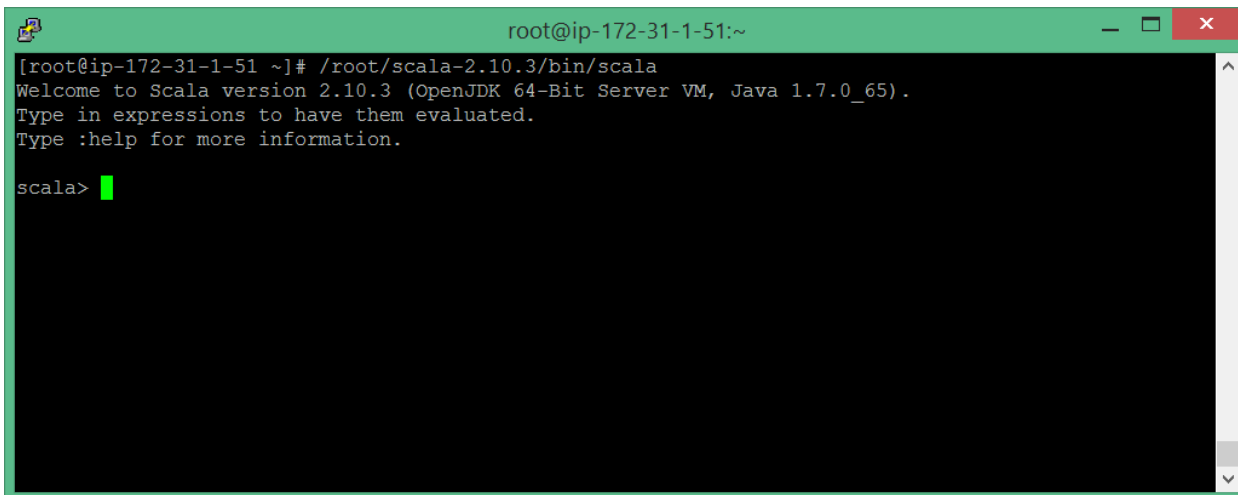
<http://ampcamp.berkeley.edu/4/exercises/introduction-to-the-scala-shell.html>

This chapter will teach you the basics of using the Scala shell and introduce you to functional programming with collections.

<http://www.artima.com/scalazine/articles/steps.html>

1. Launch the Scala console by typing:

```
/root/scala-2.10.3/bin/scala
```

A screenshot of a terminal window with a green title bar. The title bar text is 'root@ip-172-31-1-51:~'. The terminal content shows the command '/root/scala-2.10.3/bin/scala' being executed. The output is: 'Welcome to Scala version 2.10.3 (OpenJDK 64-Bit Server VM, Java 1.7.0_65). Type in expressions to have them evaluated. Type :help for more information.' The prompt 'scala>' is followed by a green cursor.

2. Declare a list of integers as a variable called "myNumbers".

```
val myNumbers = List(1, 2, 5, 4, 7, 3)
```

```
root@ip-172-31-1-51:~  
[root@ip-172-31-1-51 ~]# /root/scala-2.10.3/bin/scala  
Welcome to Scala version 2.10.3 (OpenJDK 64-Bit Server VM, Java 1.7.0_65).  
Type in expressions to have them evaluated.  
Type :help for more information.  
  
scala> val myNumbers = List(1,2,5,4,7,3)  
myNumbers: List[Int] = List(1, 2, 5, 4, 7, 3)  
  
scala> █
```

Declare a function, `cube`, that computes the cube (third power) of an `Int`. See steps 2-4 of First Steps to Scala.

```
def cube(a: Int): Int = a * a * a
```

```
cube: (a: Int)Int
```

```
root@ip-172-31-1-51:~  
[root@ip-172-31-1-51 ~]# /root/scala-2.10.3/bin/scala  
Welcome to Scala version 2.10.3 (OpenJDK 64-Bit Server VM, Java 1.7.0_65).  
Type in expressions to have them evaluated.  
Type :help for more information.  
  
scala> val myNumbers = List(1,2,5,4,7,3)  
myNumbers: List[Int] = List(1, 2, 5, 4, 7, 3)  
  
scala> def cube(a: Int): Int = a*a*a  
cube: (a: Int)Int  
  
scala> █
```

Apply the function to `myNumbers` using the `map` function. Hint: read about the `map` function in [the Scala List API](#) and also in Table 1 about halfway through the [First Steps to Scala](#) tutorial.

```
myNumbers.map(x => cube(x))
```



```
root@ip-172-31-1-51:~  
[root@ip-172-31-1-51 ~]# /root/scala-2.10.3/bin/scala  
Welcome to Scala version 2.10.3 (OpenJDK 64-Bit Server VM, Java 1.7.0_65).  
Type in expressions to have them evaluated.  
Type :help for more information.  
  
scala> val myNumbers = List(1,2,5,4,7,3)  
myNumbers: List[Int] = List(1, 2, 5, 4, 7, 3)  
  
scala> def cube(a:Int): Int = a*a*a  
cube: (a: Int)Int  
  
scala> myNumbers.map(x=>cube(x))  
res0: List[Int] = List(1, 8, 125, 64, 343, 27)  
  
scala> █
```

5. Then also try writing the function inline in a map call, using closure notation.

```
myNumbers.map{x => x * x * x}
```

```
root@ip-172-31-1-51:~  
[root@ip-172-31-1-51 ~]# /root/scala-2.10.3/bin/scala  
Welcome to Scala version 2.10.3 (OpenJDK 64-Bit Server VM, Java 1.7.0_65).  
Type in expressions to have them evaluated.  
Type :help for more information.  
  
scala> val myNumbers = List(1,2,5,4,7,3)  
myNumbers: List[Int] = List(1, 2, 5, 4, 7, 3)  
  
scala> def cube(a:Int): Int = a*a*a  
cube: (a: Int)Int  
  
scala> myNumbers.map(x=>cube(x))  
res0: List[Int] = List(1, 8, 125, 64, 343, 27)  
  
scala> myNumbers.map{x => x * x * x}  
res1: List[Int] = List(1, 8, 125, 64, 343, 27)  
  
scala> █
```

Define a factorial function that computes $n! = 1 * 2 * \dots * n$ given input n . You can use either a loop or recursion, in our solution we use recursion (see steps 5-7 of [First Steps to Scala](#)). Then compute the sum of factorials in myNumbers. Hint: check out the [sum](#) function in [the Scala List API](#).

```
def factorial(n:Int):Int = if (n==0) 1 else n * factorial(n-1)
```

```
myNumbers.map(factorial).sum
```

Note : To clear screen use 'CTRL + L'

```
root@ip-172-31-1-51:~  
scala> def factorial(n:Int):Int = if (n==0) 1 else n * factorial(n-1)  
factorial: (n: Int)Int  
  
scala> myNumbers.map(factorial).sum  
res4: Int = 5193  
  
scala> █
```

7. BONUS QUESTION. This is a more challenging task and may require 10 minutes or more to complete

1. `import scala.io.Source`
2. `val lines = Source.fromFile("/root/spark/README.md").getLines.toArray`
3. `val lines = Source.fromFile("/root/spark/README.md").getLines.toArray`

```
root@ip-172-31-1-51:~  
scala> import scala.io.Source  
import scala.io.Source  
  
scala> val lines = Source.fromFile("/root/spark/README.md").getLines.toArray  
lines: Array[String] = Array(# Apache Spark, "", Lightning-Fast Cluster Computing - <http://spark.i  
ncubator.apache.org/>, "", "", ## Online Documentation, "", You can find the latest Spark documenta  
tion, including a programming, guide, on the project webpage at <http://spark.incubator.apache.org/  
documentation.html>., This README file only contains basic setup instructions., "", "", ## Building  
, "", Spark requires Scala 2.10. The project is built using Simple Build Tool (SBT),, which can be  
obtained [here](http://www.scala-sbt.org). If SBT is installed we, will use the system version of s  
bt otherwise we will attempt to download it, automatically. To build Spark and its example programs  
, run:, "", "    ./sbt/sbt assembly", "", Once you've built Spark, the easiest way to start using i  
t is ...  
scala> █
```

```
val counts = new collection.mutable.HashMap[String, Int].withDefaultValue(0)
```

```
root@ip-172-31-1-51:~  
scala> import scala.io.Source  
import scala.io.Source  
  
scala> val lines = Source.fromFile("/root/spark/README.md").getLines.toArray  
lines: Array[String] = Array(# Apache Spark, "", Lightning-Fast Cluster Computing - <http://spark.i  
ncubator.apache.org/>, "", "", ## Online Documentation, "", You can find the latest Spark documenta  
tion, including a programming, guide, on the project webpage at <http://spark.incubator.apache.org/  
documentation.html>., This README file only contains basic setup instructions., "", "", ## Building  
, "", Spark requires Scala 2.10. The project is built using Simple Build Tool (SBT),, which can be  
obtained [here](http://www.scala-sbt.org). If SBT is installed we, will use the system version of s  
bt otherwise we will attempt to download it, automatically. To build Spark and its example programs  
, run:, "", "    ./sbt/sbt assembly", "", Once you've built Spark, the easiest way to start using i  
t is ...  
scala> val counts = new collection.mutable.HashMap[String, Int].withDefaultValue(0)  
counts: scala.collection.mutable.Map[String,Int] = Map()  
  
scala>
```

```
lines.flatMap(line => line.split(" ")).foreach(word => counts(word) += 1)
```

```
root@ip-172-31-1-51:~  
import scala.io.Source  
  
scala> val lines = Source.fromFile("/root/spark/README.md").getLines.toArray  
lines: Array[String] = Array(# Apache Spark, "", Lightning-Fast Cluster Computing - <http://spark.i  
ncubator.apache.org/>, "", "", ## Online Documentation, "", You can find the latest Spark documenta  
tion, including a programming, guide, on the project webpage at <http://spark.incubator.apache.org/  
documentation.html>., This README file only contains basic setup instructions., "", "", ## Building  
, "", Spark requires Scala 2.10. The project is built using Simple Build Tool (SBT),, which can be  
obtained [here](http://www.scala-sbt.org). If SBT is installed we, will use the system version of s  
bt otherwise we will attempt to download it, automatically. To build Spark and its example programs  
, run:, "", "    ./sbt/sbt assembly", "", Once you've built Spark, the easiest way to start using i  
t is ...  
scala> val counts = new collection.mutable.HashMap[String, Int].withDefaultValue(0)  
counts: scala.collection.mutable.Map[String,Int] = Map()  
  
scala> lines.flatMap(line => line.split(" ")).foreach(word => counts(word) += 1)  
  
scala>
```

counts

```
root@ip-172-31-1-51:~  
, run:, "", "    ./sbt/sbt assembly", "", Once you've built Spark, the easiest way to start using i  
t is ...  
scala> val counts = new collection.mutable.HashMap[String, Int].withDefaultValue(0)  
counts: scala.collection.mutable.Map[String,Int] = Map()  
  
scala> lines.flatMap(line => line.split(" ")).foreach(word => counts(word) += 1)  
  
scala> counts  
res1: scala.collection.mutable.Map[String,Int] = Map(request, -> 1, Documentation -> 1, requires ->  
2, Each -> 1, their -> 1, code, -> 1, ./sbt/sbt -> 1, instructions. -> 1, MRv1, -> 1, basic -> 1,  
SPARK HADOOP_VERSION=2.0.0-cdh4.2.0 -> 1, must -> 1, Incubator -> 1, Regression -> 1, Hadoop, -> 1  
Online -> 1, thread, -> 1, projects -> 1, v2 -> 1, org.apache.spark.examples.SparkLR -> 1, Clouder  
a -> 4, POM -> 1, To -> 2, is -> 10, contribution -> 1, Building -> 1, yet -> 1, 2.10. -> 1, adding  
-> 1, required -> 1, usage -> 1, Versions -> 1, does -> 1, application, -> 1, If -> 2, sponsored -  
> 1, 2 -> 1, About -> 1, uses -> 1, can -> 5, email, -> 1, This -> 2, MapReduce -> 2, gladly -> 1,  
Please -> 1, one -> 2, # -> 6, including -> 1, against -> 1, While -> 1, ASF -> 1, using -> 4, unti  
l -> ...  
scala>
```