

Hadoop: Getting Start *

Starting your single-node cluster

Run the command:

```
hadoop@ubuntu:~$ <HADOOP_INSTALL>/bin/start-all.sh
```

This will startup a Namenode, Datanode, Jobtracker and a Tasktracker on your machine.

The output will look like this:

```
hadoop@ubuntu:/usr/local/hadoop$ bin/start-all.sh
starting namenode, logging to /usr/local/hadoop/bin/../../logs/hadoop-hadoop-
namenode-ubuntu.out
localhost: starting datanode, logging to /usr/local/hadoop/bin/../../logs/hadoop-
hadoop-datanode-ubuntu.out
localhost: starting secondarynamenode, logging to
/usr/local/hadoop/bin/../../logs/hadoop-hadoop-secondarynamenode-ubuntu.out
starting jobtracker, logging to /usr/local/hadoop/bin/../../logs/hadoop-hadoop-
jobtracker-ubuntu.out
localhost: starting tasktracker, logging to
/usr/local/hadoop/bin/../../logs/hadoop-hadoop-tasktracker-ubuntu.out
hadoop@ubuntu:/usr/local/hadoop$
```

A nifty tool for checking whether the expected Hadoop processes are running is `jps` (part of Sun's Java since v1.5.0). See also [How to debug MapReduce programs](#).

```
hadoop@sea:/usr/local/hadoop/$ jps
19811 TaskTracker
19674 SecondaryNameNode
19735 JobTracker
19497 NameNode
20879 TaskTracker$Child
21810 Jps
```

You can also check with `netstat` if Hadoop is listening on the configured ports.

```
hadoop@ubuntu:~$ sudo netstat -plten | grep java
tcp 0 0 0.0.0.0:50050 0.0.0.0:* LISTEN 1001 86234 23634/java
tcp 0 0 127.0.0.1:54310 0.0.0.0:* LISTEN 1001 85800 23317/java
tcp 0 0 127.0.0.1:54311 0.0.0.0:* LISTEN 1001 86383 23543/java
tcp 0 0 0.0.0.0:50090 0.0.0.0:* LISTEN 1001 86119 23478/java
tcp 0 0 0.0.0.0:50060 0.0.0.0:* LISTEN 1001 86233 23634/java
tcp 0 0 0.0.0.0:50030 0.0.0.0:* LISTEN 1001 86393 23543/java
tcp 0 0 0.0.0.0:50070 0.0.0.0:* LISTEN 1001 85964 23317/java
tcp 0 0 0.0.0.0:50010 0.0.0.0:* LISTEN 1001 86045 23389/java
tcp 0 0 0.0.0.0:50075 0.0.0.0:* LISTEN 1001 86102 23389/java
hadoop@ubuntu:~$
```

If there are any errors, examine the log files in the `<HADOOP_INSTALL>/logs/` directory.

Stopping your single-node cluster

Run the command

```
hadoop@ubuntu:~$ <HADOOP_INSTALL>/bin/stop-all.sh
```

to stop all the daemons running on your machine.

Exemplary output:

```
hadoop@ubuntu:/usr/local/hadoop$ bin/stop-all.sh
stopping jobtracker
localhost: Ubuntu 8.04
localhost: stopping tasktracker
stopping namenode
localhost: Ubuntu 8.04
localhost: stopping datanode
localhost: Ubuntu 8.04
localhost: stopping secondarynamenode
hadoop@ubuntu:/usr/local/hadoop$
```

Running a MapReduce job

We will now run your first Hadoop [MapReduce](#) job. We will use the [WordCount example job](#) which reads text files and counts how often words occur. The input is text files and the output is text files, each line of which contains a word and the count of how often it occurred, separated by a tab. More information of [what happens behind the scenes](#) is available at the [Hadoop Wiki](#).

Download example input data

We will use three ebooks from Project Gutenberg for this example:

- [The Outline of Science, Vol. 1 \(of 4\) by J. Arthur Thomson](#)
- [The Notebooks of Leonardo Da Vinci](#)
- [Ulysses by James Joyce](#)

Download each ebook as plain text files in `us-ascii` encoding and store the uncompressed files in a temporary directory of choice, for example `/tmp/gutenberg`.

```
hadoop@ubuntu:~$ ls -l /tmp/gutenberg/
total 3592
-rw-r--r-- 1 hadoop hadoop 674425 2007-01-22 12:56 20417-8.txt
-rw-r--r-- 1 hadoop hadoop 1423808 2006-08-03 16:36 71dvc10.txt
-rw-r--r-- 1 hadoop hadoop 1561677 2004-11-26 09:48 ulyss12.txt
hadoop@ubuntu:~$
```

Restart the Hadoop cluster

Restart your Hadoop cluster if it's not running already.

```
hadoop@ubuntu:~$ <HADOOP_INSTALL>/bin/start-all.sh
```

Copy local example data to HDFS

Before we run the actual MapReduce job, we first [have to copy](#) the files from our local file system to Hadoop's [HDFS](#).

```
hadoop@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -copyFromLocal /tmp/gutenberg
gutenberg
hadoop@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -ls
Found 1 items
/user/hadoop/gutenberg <dir>
hadoop@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -ls gutenberg
Found 3 items
/user/hadoop/gutenberg/20417-8.txt      <r 1>    674425
/user/hadoop/gutenberg/71dvc10.txt     <r 1>   1423808
```

```
/user/hadoop/gutenberg/ulyss12.txt      <r 1> 1561677
```

Run the MapReduce job

Now, we actually run the WordCount example job.

```
hadoop@ubuntu:/usr/local/hadoop$ bin/hadoop jar hadoop-0.19.1-examples.jar
wordcount gutenberg gutenberg-output
```

This command will read all the files in the HDFS directory gutenberg, process it, and store the result in the HDFS directory gutenberg-output.

Exemplary output of the previous command in the console:

```
hadoop@ubuntu:/usr/local/hadoop$ bin/hadoop jar hadoop-0.19.1-examples.jar
wordcount gutenberg gutenberg-output
07/09/21 13:00:30 INFO mapred.FileInputFormat: Total input paths to process : 3
07/09/21 13:00:31 INFO mapred.JobClient: Running job: job_200709211255_0001
07/09/21 13:00:32 INFO mapred.JobClient: map 0% reduce 0%
07/09/21 13:00:42 INFO mapred.JobClient: map 66% reduce 0%
07/09/21 13:00:47 INFO mapred.JobClient: map 100% reduce 22%
07/09/21 13:00:54 INFO mapred.JobClient: map 100% reduce 100%
07/09/21 13:00:55 INFO mapred.JobClient: Job complete: job_200709211255_0001
07/09/21 13:00:55 INFO mapred.JobClient: Counters: 12
07/09/21 13:00:55 INFO mapred.JobClient: Job Counters
07/09/21 13:00:55 INFO mapred.JobClient: Launched map tasks=3
07/09/21 13:00:55 INFO mapred.JobClient: Launched reduce tasks=1
07/09/21 13:00:55 INFO mapred.JobClient: Data-local map tasks=3
07/09/21 13:00:55 INFO mapred.JobClient: Map-Reduce Framework
07/09/21 13:00:55 INFO mapred.JobClient: Map input records=77637
07/09/21 13:00:55 INFO mapred.JobClient: Map output records=628439
07/09/21 13:00:55 INFO mapred.JobClient: Map input bytes=3659910
07/09/21 13:00:55 INFO mapred.JobClient: Map output bytes=6061344
07/09/21 13:00:55 INFO mapred.JobClient: Combine input records=628439
07/09/21 13:00:55 INFO mapred.JobClient: Combine output records=103910
07/09/21 13:00:55 INFO mapred.JobClient: Reduce input groups=85096
07/09/21 13:00:55 INFO mapred.JobClient: Reduce input records=103910
07/09/21 13:00:55 INFO mapred.JobClient: Reduce output records=85096
hadoop@ubuntu:/usr/local/hadoop$
```

Check if the result is successfully stored in HDFS directory gutenberg-output:

```
hadoop@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -ls
Found 2 items
/user/hadoop/gutenberg <dir>
/user/hadoop/gutenberg-output <dir>
hadoop@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -ls gutenberg-output
Found 1 items
/user/hadoop/gutenberg-output/part-00000      <r 1> 903193
hadoop@ubuntu:/usr/local/hadoop$
```

If you want to modify some Hadoop settings on the fly like increasing the number of Reduce tasks, you can use the "-D" option:

```
hadoop@ubuntu:/usr/local/hadoop$ bin/hadoop jar hadoop-0.19.1-examples.jar
wordcount -D mapred.reduce.tasks=16 gutenberg gutenberg-output
```

An important note about `mapred.map.tasks`: [Hadoop does not honor `mapred.map.tasks`](#) beyond considering it a hint. But it accepts the user specified `mapred.reduce.tasks` and doesn't manipulate that. You cannot force

mapred.map.tasks but can specify mapred.reduce.tasks.

Retrieve the job result from HDFS

To inspect the file, you can copy it from HDFS to the local file system. Alternatively, you can use the command

```
hadoop@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -cat gutenber-output/part-00000
```

to read the file directly from HDFS without copying it to the local file system. In this tutorial, we will copy the results to the local file system though.

```
hadoop@ubuntu:/usr/local/hadoop$ mkdir /tmp/gutenberg-output
hadoop@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -copyToLocal gutenber-output/part-00000 /tmp/gutenberg-output
hadoop@ubuntu:/usr/local/hadoop$ head /tmp/gutenberg-output/part-00000
"(Lo)cra"          1
"1490"            1
"1498,"           1
"35"              1
"40,"             1
"A"               2
"AS-IS".          2
"A_"              1
"Absoluti"        1
"Alack!"          1
hadoop@ubuntu:/usr/local/hadoop$
```

Note that in this specific output the quote signs (") enclosing the words in the head output above have not been inserted by Hadoop. They are the result of the word tokenizer used in the WordCount example, and in this case they matched the beginning of a quote in the ebook texts. Just inspect the part-00000 file further to see it for yourself.

Other Commands

There are several other commands associated with the FsShell subsystem; these can perform most common filesystem manipulations (cat, head, less, rm, mv, cp, mkdir, etc.). Try playing around with a few of these if you'd like.

* Tutorial adapted from [http://www.michael-noll.com/wiki/Running_Hadoop_On_Ubuntu_Linux_\(Single-Node_Cluster\)#Starting_your_single-node_cluster](http://www.michael-noll.com/wiki/Running_Hadoop_On_Ubuntu_Linux_(Single-Node_Cluster)#Starting_your_single-node_cluster)