

# Michael G. Noll

**Applied Research. Big Data. Distributed Systems. Open Source.**

- [RSS](#)

Search
Navigate... ▼

- [Blog](#)
- [Archive](#)
- [Tutorials](#)
- [Projects](#)
- [Publications](#)

## Running Hadoop on Ubuntu Linux (Single-Node Cluster)

Table of Contents

- [Prerequisites](#)
  - [Sun Java 6](#)
  - [Adding a dedicated Hadoop system user](#)
  - [Configuring SSH](#)
  - [Disabling IPv6](#)
    - [Alternative](#)
- [Hadoop](#)
  - [Installation](#)
  - [Update \\$HOME/.bashrc](#)
  - [Excursus: Hadoop Distributed File System \(HDFS\)](#)
  - [Configuration](#)
    - [hadoop-env.sh](#)
    - [conf/\\*-site.xml](#)
  - [Formatting the HDFS filesystem via the NameNode](#)
  - [Starting your single-node cluster](#)
  - [Stopping your single-node cluster](#)
  - [Running a MapReduce job](#)
    - [Download example input data](#)
    - [Restart the Hadoop cluster](#)
    - [Copy local example data to HDFS](#)
    - [Run the MapReduce job](#)
    - [Retrieve the job result from HDFS](#)
  - [Hadoop Web Interfaces](#)
    - [NameNode Web Interface \(HDFS layer\)](#)
    - [JobTracker Web Interface \(MapReduce layer\)](#)
    - [TaskTracker Web Interface \(MapReduce layer\)](#)
- [What's next?](#)
- [Related Links](#)
- [Change Log](#)

In this tutorial I will describe the required steps for setting up a *pseudo-distributed, single-node* Hadoop cluster backed by the Hadoop Distributed File System, running on Ubuntu Linux.

Are you looking for the [multi-node cluster tutorial](#)? Just [head over there](#).

Hadoop is a framework written in Java for running applications on large clusters of commodity hardware and incorporates features similar to those of the [Google File System \(GFS\)](#) and of the [MapReduce](#) computing paradigm. Hadoop's [HDFS](#) is a highly fault-tolerant distributed file system and, like Hadoop in general, designed to be deployed on low-cost hardware. It provides high throughput access to application data and is suitable for applications that have large data sets.

The main goal of this tutorial is to get a simple Hadoop installation up and running so that you can play around with the software and learn more about it.

This tutorial has been tested with the following software versions:

- [Ubuntu Linux](#) 10.04 LTS (deprecated: 8.10 LTS, 8.04, 7.10, 7.04)
- [Hadoop](#) 1.0.3, released May 2012



Figure 1: Cluster of machines running Hadoop at Yahoo! (Source: Yahoo!)

## Prerequisites

### Sun Java 6

Hadoop requires a working Java 1.5+ (aka Java 5) installation. However, using [Java 1.6 \(aka Java 6\) is recommended](#) for running Hadoop. For the sake of this tutorial, I will therefore describe the installation of Java 1.6.

Important Note: The apt instructions below are taken from [this SuperUser.com thread](#). I got notified that the previous instructions that I provided no longer work. Please be aware that adding a third-party repository to your Ubuntu configuration is considered a security risk. If you do not want to proceed with the apt instructions below, feel free to install Sun JDK 6 via alternative means (e.g. by [downloading the binary package from Oracle](#)) and then continue with the next section in the tutorial.

```
1 # Add the Ferramosca Roberto's repository to your apt repositories
2 # See https://launchpad.net/~ferramroberto/
3 #
4 $ sudo apt-get install python-software-properties
5 $ sudo add-apt-repository ppa:ferramroberto/java
6
7 # Update the source list
8 $ sudo apt-get update
9
10 # Install Sun Java 6 JDK
11 $ sudo apt-get install sun-java6-jdk
12
13 # Select Sun's Java as the default on your machine.
14 # See 'sudo update-alternatives --config java' for more information.
15 #
16 $ sudo update-java-alternatives -s java-6-sun
```

The full JDK which will be placed in `/usr/lib/jvm/java-6-sun` (well, this directory is actually a symlink on Ubuntu).

After installation, make a quick check whether Sun's JDK is correctly set up:

```
1 user@ubuntu:~# java -version
2 java version "1.6.0_20"
3 Java(TM) SE Runtime Environment (build 1.6.0_20-b02)
4 Java HotSpot(TM) Client VM (build 16.3-b01, mixed mode, sharing)
```

## Adding a dedicated Hadoop system user

We will use a dedicated Hadoop user account for running Hadoop. While that's not required it is recommended because it helps to separate the Hadoop installation from other software applications and user accounts running on the same machine (think: security, permissions, backups, etc).

```
1 $ sudo addgroup hadoop
2 $ sudo adduser --ingroup hadoop hduser
```

This will add the user `hduser` and the group `hadoop` to your local machine.

## Configuring SSH

Hadoop requires SSH access to manage its nodes, i.e. remote machines plus your local machine if you want to use Hadoop on it (which is what we want to do in this short tutorial). For our single-node setup of Hadoop, we therefore need to configure SSH access to `localhost` for the `hduser` user we created in the previous section.

I assume that you have SSH up and running on your machine and configured it to allow SSH public key authentication. If not, there are [several online guides](#) available.

First, we have to generate an SSH key for the `hduser` user.

```

1 user@ubuntu:~$ su - hduser
2 hduser@ubuntu:~$ ssh-keygen -t rsa -P ""
3 Generating public/private rsa key pair.
4 Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
5 Created directory '/home/hduser/.ssh'.
6 Your identification has been saved in /home/hduser/.ssh/id_rsa.
7 Your public key has been saved in /home/hduser/.ssh/id_rsa.pub.
8 The key fingerprint is:
9 9b:82:ea:58:b4:e0:35:d7:ff:19:66:a6:ef:ae:0e:d2 hduser@ubuntu
10 The key's randomart image is:
11 [...snipp...]
12 hduser@ubuntu:~$

```

The second line will create an RSA key pair with an empty password. Generally, using an empty password is not recommended, but in this case it is needed to unlock the key without your interaction (you don't want to enter the passphrase every time Hadoop interacts with its nodes).

Second, you have to enable SSH access to your local machine with this newly created key.

```

1 hduser@ubuntu:~$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

```

The final step is to test the SSH setup by connecting to your local machine with the `hduser` user. The step is also needed to save your local machine's host key fingerprint to the `hduser` user's `known_hosts` file. If you have any special SSH configuration for your local machine like a non-standard SSH port, you can define host-specific SSH options in `$HOME/.ssh/config` (see `man ssh_config` for more information).

```

1 hduser@ubuntu:~$ ssh localhost
2 The authenticity of host 'localhost (:::1)' can't be established.
3 RSA key fingerprint is d7:87:25:47:ae:02:00:eb:1d:75:4f:bb:44:f9:36:26.
4 Are you sure you want to continue connecting (yes/no)? yes
5 Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
6 Linux ubuntu 2.6.32-22-generic #33-Ubuntu SMP Wed Apr 28 13:27:30 UTC 2010 i686 GNU/Linux
7 Ubuntu 10.04 LTS
8 [...snipp...]
9 hduser@ubuntu:~$

```

If the SSH connect should fail, these general tips might help:

- Enable debugging with `ssh -vvv localhost` and investigate the error in detail.
- Check the SSH server configuration in `/etc/ssh/sshd_config`, in particular the options `PubkeyAuthentication` (which should be set to `yes`) and `AllowUsers` (if this option is active, add the `hduser` user to it). If you made any changes to the SSH server configuration file, you can force a configuration reload with `sudo /etc/init.d/ssh reload`.

## Disabling IPv6

One problem with IPv6 on Ubuntu is that using `0.0.0.0` for the various networking-related Hadoop configuration options will result in Hadoop binding to the IPv6 addresses of my Ubuntu box. In my case, I realized that there's no practical point in enabling IPv6 on a box when you are not connected to any IPv6 network. Hence, I simply disabled IPv6 on my Ubuntu machine. Your mileage may vary.

To disable IPv6 on Ubuntu 10.04 LTS, open `/etc/sysctl.conf` in the editor of your choice and add the following lines to the end of the file:

```

/etc/sysctl.conf

1 # disable ipv6
2 net.ipv6.conf.all.disable_ipv6 = 1
3 net.ipv6.conf.default.disable_ipv6 = 1
4 net.ipv6.conf.lo.disable_ipv6 = 1

```

You have to reboot your machine in order to make the changes take effect.

You can check whether IPv6 is enabled on your machine with the following command:

```

1 $ cat /proc/sys/net/ipv6/conf/all/disable_ipv6

```

A return value of 0 means IPv6 is enabled, a value of 1 means disabled (that's what we want).

## Alternative

You can also disable IPv6 only for Hadoop as documented in [HADOOP-3437](#). You can do so by adding the following line to `conf/hadoop-env.sh`:

```

conf/hadoop-env.sh

1 export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true

```

## Hadoop

## Installation

[Download Hadoop](#) from the [Apache Download Mirrors](#) and extract the contents of the Hadoop package to a location of your choice. I picked `/usr/local/hadoop`. Make sure to change the owner of all the files to the `hduser` user and `hadoop` group, for example:

```
1 $ cd /usr/local
2 $ sudo tar xzf hadoop-1.0.3.tar.gz
3 $ sudo mv hadoop-1.0.3 hadoop
4 $ sudo chown -R hduser:hadoop hadoop
```

(Just to give you the idea, YMMV – personally, I create a symlink from `hadoop-1.0.3` to `hadoop`.)

## Update \$HOME/.bashrc

Add the following lines to the end of the `$HOME/.bashrc` file of user `hduser`. If you use a shell other than `bash`, you should of course update its appropriate configuration files instead of `.bashrc`.

`$HOME/.bashrc`

```
1 # Set Hadoop-related environment variables
2 export HADOOP_HOME=/usr/local/hadoop
3
4 # Set JAVA_HOME (we will also configure JAVA_HOME directly for Hadoop later on)
5 export JAVA_HOME=/usr/lib/jvm/java-6-sun
6
7 # Some convenient aliases and functions for running Hadoop-related commands
8 unalias fs &> /dev/null
9 alias fs="hadoop fs"
10 unalias hls &> /dev/null
11 alias hls="fs -ls"
12
13 # If you have LZO compression enabled in your Hadoop cluster and
14 # compress job outputs with LZOP (not covered in this tutorial):
15 # Conveniently inspect an LZOP compressed file from the command
16 # line; run via:
17 #
18 # $ lzohead /hdfs/path/to/lzop/compressed/file.lzo
19 #
20 # Requires installed 'lzop' command.
21 #
22 lzohead () {
23     hadoop fs -cat $1 | lzop -dc | head -1000 | less
24 }
25
26 # Add Hadoop bin/ directory to PATH
27 export PATH=$PATH:$HADOOP_HOME/bin
```

You can repeat this exercise also for other users who want to use Hadoop.

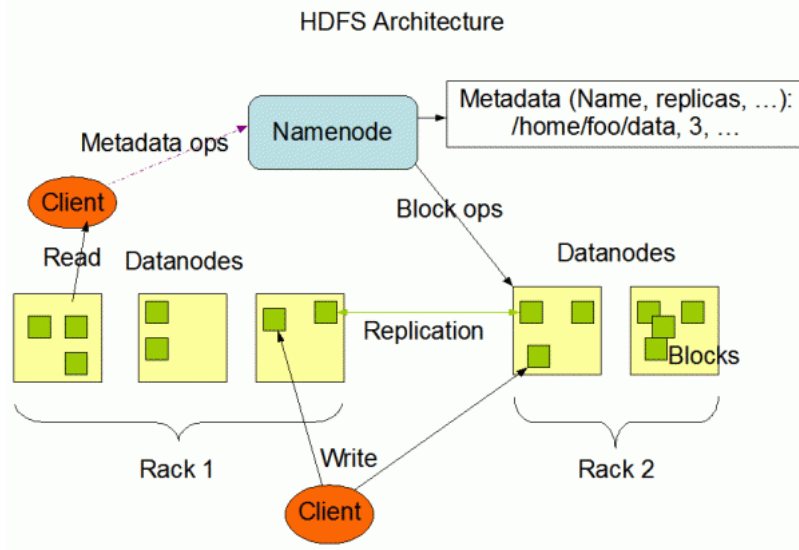
## Excursus: Hadoop Distributed File System (HDFS)

Before we continue let us briefly learn a bit more about Hadoop's distributed file system.

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. HDFS relaxes a few POSIX requirements to enable streaming access to file system data. HDFS was originally built as infrastructure for the Apache Nutch web search engine project. HDFS is part of the Apache Hadoop project, which is part of the Apache Lucene project.

**The Hadoop Distributed File System: Architecture and Design** [hadoop.apache.org/hdfs/docs/...](http://hadoop.apache.org/hdfs/docs/)

The following picture gives an overview of the most important HDFS components.



## Configuration

Our goal in this tutorial is a single-node setup of Hadoop. More information of what we do in this section is available on the [Hadoop Wiki](#).

### hadoop-env.sh

The only required environment variable we have to configure for Hadoop in this tutorial is `JAVA_HOME`. Open `conf/hadoop-env.sh` in the editor of your choice (if you used the installation path in this tutorial, the full path is `/usr/local/hadoop/conf/hadoop-env.sh`) and set the `JAVA_HOME` environment variable to the Sun JDK/JRE 6 directory.

Change

`conf/hadoop-env.sh`

```
1 # The java implementation to use. Required.
2 # export JAVA_HOME=/usr/lib/j2sdk1.5-sun
```

to

`conf/hadoop-env.sh`

```
1 # The java implementation to use. Required.
2 export JAVA_HOME=/usr/lib/jvm/java-6-sun
```

Note: If you are on a Mac with OS X 10.7 you can use the following line to set up `JAVA_HOME` in `conf/hadoop-env.sh`.

`conf/hadoop-env.sh` (on Mac systems)

```
1 # for our Mac users
2 export JAVA_HOME="/usr/libexec/java_home"
```

### conf/\*-site.xml

In this section, we will configure the directory where Hadoop will store its data files, the network ports it listens to, etc. Our setup will use Hadoop's Distributed File System, [HDFS](#), even though our little "cluster" only contains our single local machine.

You can leave the settings below "as is" with the exception of the `hadoop.tmp.dir` parameter – this parameter you must change to a directory of your choice. We will use the directory `/app/hadoop/tmp` in this tutorial. Hadoop's default configurations use `hadoop.tmp.dir` as the base temporary directory both for the local file system and HDFS, so don't be surprised if you see Hadoop creating the specified directory automatically on HDFS at some later point.

Now we create the directory and set the required ownerships and permissions:

```
1 $ sudo mkdir -p /app/hadoop/tmp
2 $ sudo chown hduser:hadoop /app/hadoop/tmp
3 # ...and if you want to tighten up security, chmod from 755 to 750...
4 $ sudo chmod 750 /app/hadoop/tmp
```

If you forget to set the required ownerships and permissions, you will see a `java.io.IOException` when you try to format the name node in the next section).

Add the following snippets between the `<configuration>` ... `</configuration>` tags in the respective configuration XML file.

In file `conf/core-site.xml`:

conf/core-site.xml

```
1 <property>
2   <name>hadoop.tmp.dir</name>
3   <value>/app/hadoop/tmp</value>
4   <description>A base for other temporary directories.</description>
5 </property>
6
7 <property>
8   <name>fs.default.name</name>
9   <value>hdfs://localhost:54310</value>
10  <description>The name of the default file system. A URI whose
11  scheme and authority determine the FileSystem implementation. The
12  uri's scheme determines the config property (fs.SCHEME.impl) naming
13  the FileSystem implementation class. The uri's authority is used to
14  determine the host, port, etc. for a filesystem.</description>
15</property>
```

In file conf/mapred-site.xml:

conf/mapred-site.xml

```
1 <property>
2   <name>mapred.job.tracker</name>
3   <value>localhost:54311</value>
4   <description>The host and port that the MapReduce job tracker runs
5   at. If "local", then jobs are run in-process as a single map
6   and reduce task.
7   </description>
8</property>
```

In file conf/hdfs-site.xml:

conf/hdfs-site.xml

```
1 <property>
2   <name>dfs.replication</name>
3   <value>1</value>
4   <description>Default block replication.
5   The actual number of replications can be specified when the file is created.
6   The default is used if replication is not specified in create time.
7   </description>
8</property>
```

See [Getting Started with Hadoop](#) and the documentation in [Hadoop's API Overview](#) if you have any questions about Hadoop's configuration options.

## Formatting the HDFS filesystem via the NameNode

The first step to starting up your Hadoop installation is formatting the Hadoop filesystem which is implemented on top of the local filesystem of your “cluster” (which includes only your local machine if you followed this tutorial). You need to do this the first time you set up a Hadoop cluster.

Do not format a running Hadoop filesystem as you will lose all the data currently in the cluster (in HDFS)!

To format the filesystem (which simply initializes the directory specified by the `dfs.name.dir` variable), run the command

```
1 hduser@ubuntu:~$ /usr/local/hadoop/bin/hadoop namenode -format
```

The output will look like this:

```
1 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop namenode -format
2 10/05/08 16:59:56 INFO namenode.NameNode: STARTUP_MSG:
3 /*****
4 STARTUP_MSG: Starting NameNode
5 STARTUP_MSG:   host = ubuntu/127.0.1.1
6 STARTUP_MSG:   args = [-format]
7 STARTUP_MSG:   version = 0.20.2
8 STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-0.20 -r 911707; compiled by 'chrisdo' on Fri
9 *****/
10 10/05/08 16:59:56 INFO namenode.FSNamesystem: fsOwner=hduser,hadoop
11 10/05/08 16:59:56 INFO namenode.FSNamesystem: supergroup=supergroup
12 10/05/08 16:59:56 INFO namenode.FSNamesystem: isPermissionEnabled=true
13 10/05/08 16:59:56 INFO common.Storage: Image file of size 96 saved in 0 seconds.
14 10/05/08 16:59:57 INFO common.Storage: Storage directory ../hadoop-hduser/dfs/name has been successfully formatted.
15 10/05/08 16:59:57 INFO namenode.NameNode: SHUTDOWN_MSG:
16 /*****
17 SHUTDOWN_MSG: Shutting down NameNode at ubuntu/127.0.1.1
18 *****/
19 hduser@ubuntu:/usr/local/hadoop$
```

## Starting your single-node cluster

Run the command:

```
1 hduser@ubuntu:~$ /usr/local/hadoop/bin/start-all.sh
```

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

The output will look like this:

```
1 hduser@ubuntu:/usr/local/hadoop$ bin/start-all.sh
2 starting namenode, logging to /usr/local/hadoop/bin/../logs/hadoop-hduser-namenode-ubuntu.out
3 localhost: starting datanode, logging to /usr/local/hadoop/bin/../logs/hadoop-hduser-datanode-ubuntu.out
4 localhost: starting secondarynamenode, logging to /usr/local/hadoop/bin/../logs/hadoop-hduser-secondarynamenode-ubuntu.out
5 starting jobtracker, logging to /usr/local/hadoop/bin/../logs/hadoop-hduser-jobtracker-ubuntu.out
6 localhost: starting tasktracker, logging to /usr/local/hadoop/bin/../logs/hadoop-hduser-tasktracker-ubuntu.out
7 hduser@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](#).

```
1 hduser@ubuntu:/usr/local/hadoop$ jps
2 2287 TaskTracker
3 2149 JobTracker
4 1938 DataNode
5 2085 SecondaryNameNode
6 2349 Jps
7 1788 NameNode
```

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

```
1 hduser@ubuntu:~$ sudo netstat -plten | grep java
2 tcp 0 0 0.0.0.0:50070 0.0.0.0:* LISTEN 1001 9236 2471/java
3 tcp 0 0 0.0.0.0:50010 0.0.0.0:* LISTEN 1001 9998 2628/java
4 tcp 0 0 0.0.0.0:48159 0.0.0.0:* LISTEN 1001 8496 2628/java
5 tcp 0 0 0.0.0.0:53121 0.0.0.0:* LISTEN 1001 9228 2857/java
6 tcp 0 0 127.0.0.1:54310 0.0.0.0:* LISTEN 1001 8143 2471/java
7 tcp 0 0 127.0.0.1:54311 0.0.0.0:* LISTEN 1001 9230 2857/java
8 tcp 0 0 0.0.0.0:59305 0.0.0.0:* LISTEN 1001 8141 2471/java
9 tcp 0 0 0.0.0.0:50060 0.0.0.0:* LISTEN 1001 9857 3005/java
10 tcp 0 0 0.0.0.0:49900 0.0.0.0:* LISTEN 1001 9037 2785/java
11 tcp 0 0 0.0.0.0:50030 0.0.0.0:* LISTEN 1001 9773 2857/java
12 hduser@ubuntu:~$
```

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

## Stopping your single-node cluster

Run the command

```
1 hduser@ubuntu:~$ /usr/local/hadoop/bin/stop-all.sh
```

to stop all the daemons running on your machine.

Example output:

```
1 hduser@ubuntu:/usr/local/hadoop$ bin/stop-all.sh
2 stopping jobtracker
3 localhost: stopping tasktracker
4 stopping namenode
5 localhost: stopping datanode
6 localhost: stopping secondarynamenode
7 hduser@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 text files in Plain Text UTF-8 encoding and store the files in a local temporary directory of choice, for example /tmp/gutenberg.

```
1 hduser@ubuntu:~$ ls -l /tmp/gutenberg/
2 total 3604
3 -rw-r--r-- 1 hduser hadoop 674566 Feb 3 10:17 pg20417.txt
4 -rw-r--r-- 1 hduser hadoop 1573112 Feb 3 10:18 pg4300.txt
5 -rw-r--r-- 1 hduser hadoop 1423801 Feb 3 10:18 pg5000.txt
6 hduser@ubuntu:~$
```

## Restart the Hadoop cluster

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

```
1 hduser@ubuntu:~$ /usr/local/hadoop/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](#).

```
1 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -copyFromLocal /tmp/gutenberg /user/hduser/gutenberg
2 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -ls /user/hduser
3 Found 1 items
4 drwxr-xr-x - hduser supergroup 0 2010-05-08 17:40 /user/hduser/gutenberg
5 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -ls /user/hduser/gutenberg
6 Found 3 items
7 -rw-r--r-- 3 hduser supergroup 674566 2011-03-10 11:38 /user/hduser/gutenberg/pg20417.txt
8 -rw-r--r-- 3 hduser supergroup 1573112 2011-03-10 11:38 /user/hduser/gutenberg/pg4300.txt
9 -rw-r--r-- 3 hduser supergroup 1423801 2011-03-10 11:38 /user/hduser/gutenberg/pg5000.txt
10 hduser@ubuntu:/usr/local/hadoop$
```

## Run the MapReduce job

Now, we actually run the WordCount example job.

```
1 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop jar hadoop*examples*.jar wordcount /user/hduser/gutenberg /user/hduser/gutenberg-output
```

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

Note: Some people run the command above and get the following error message:

```
Exception in thread "main" java.io.IOException: Error opening job jar: hadoop*examples*.jar
at org.apache.hadoop.util.RunJar.main (RunJar.java: 90)
Caused by: java.util.zip.ZipException: error in opening zip file
```

In this case, re-run the command with the full name of the Hadoop Examples JAR file, for example:

```
hduser@ubuntu:/usr/local/hadoop$ bin/hadoop jar hadoop-examples-1.0.3.jar wordcount /user/hduser/gutenberg /user/hduser/gutenberg-output
```

Example output of the previous command in the console:

```
1 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop jar hadoop*examples*.jar wordcount /user/hduser/gutenberg /user/hduser/gutenberg-outpu
2 10/05/08 17:43:00 INFO input.FileInputFormat: Total input paths to process : 3
3 10/05/08 17:43:01 INFO mapred.JobClient: Running job: job_201005081732_0001
4 10/05/08 17:43:02 INFO mapred.JobClient: map 0% reduce 0%
5 10/05/08 17:43:14 INFO mapred.JobClient: map 66% reduce 0%
6 10/05/08 17:43:17 INFO mapred.JobClient: map 100% reduce 0%
7 10/05/08 17:43:26 INFO mapred.JobClient: map 100% reduce 100%
8 10/05/08 17:43:28 INFO mapred.JobClient: Job complete: job_201005081732_0001
9 10/05/08 17:43:28 INFO mapred.JobClient: Counters: 17
10 10/05/08 17:43:28 INFO mapred.JobClient: Job Counters
11 10/05/08 17:43:28 INFO mapred.JobClient: Launched reduce tasks=1
12 10/05/08 17:43:28 INFO mapred.JobClient: Launched map tasks=3
13 10/05/08 17:43:28 INFO mapred.JobClient: Data-local map tasks=3
14 10/05/08 17:43:28 INFO mapred.JobClient: FileSystemCounters
15 10/05/08 17:43:28 INFO mapred.JobClient: FILE_BYTES_READ=2214026
16 10/05/08 17:43:28 INFO mapred.JobClient: HDFS_BYTES_READ=3639512
17 10/05/08 17:43:28 INFO mapred.JobClient: FILE_BYTES_WRITTEN=3687918
18 10/05/08 17:43:28 INFO mapred.JobClient: HDFS_BYTES_WRITTEN=880330
19 10/05/08 17:43:28 INFO mapred.JobClient: Map-Reduce Framework
20 10/05/08 17:43:28 INFO mapred.JobClient: Reduce input groups=82290
21 10/05/08 17:43:28 INFO mapred.JobClient: Combine output records=102286
22 10/05/08 17:43:28 INFO mapred.JobClient: Map input records=77934
```



```

23 10/05/08 17:43:28 INFO mapred.JobClient:      Reduce shuffle bytes=1473796
24 10/05/08 17:43:28 INFO mapred.JobClient:      Reduce output records=82290
25 10/05/08 17:43:28 INFO mapred.JobClient:      Spilled Records=255874
26 10/05/08 17:43:28 INFO mapred.JobClient:      Map output bytes=6076267
27 10/05/08 17:43:28 INFO mapred.JobClient:      Combine input records=629187
28 10/05/08 17:43:28 INFO mapred.JobClient:      Map output records=629187
29 10/05/08 17:43:28 INFO mapred.JobClient:      Reduce input records=102286

```

Check if the result is successfully stored in HDFS directory `/user/hduser/gutenberg-output`:

```

1 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -ls /user/hduser
2 Found 2 items
3 drwxr-xr-x  - hduser supergroup          0 2010-05-08 17:40 /user/hduser/gutenberg
4 drwxr-xr-x  - hduser supergroup          0 2010-05-08 17:43 /user/hduser/gutenberg-output
5 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -ls /user/hduser/gutenberg-output
6 Found 2 items
7 drwxr-xr-x  - hduser supergroup          0 2010-05-08 17:43 /user/hduser/gutenberg-output/_logs
8 -rw-r--r--  1 hduser supergroup      880802 2010-05-08 17:43 /user/hduser/gutenberg-output/part-r-00000
9 hduser@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:

```
1 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop jar hadoop*examples*.jar wordcount -D mapred.reduce.tasks=16 /user/hduser/gutenberg /us
```

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 you 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

```
1 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -cat /user/hduser/gutenberg-output/part-r-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.

```

1 hduser@ubuntu:/usr/local/hadoop$ mkdir /tmp/gutenberg-output
2 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -getmerge /user/hduser/gutenberg-output /tmp/gutenberg-output
3 hduser@ubuntu:/usr/local/hadoop$ head /tmp/gutenberg-output/gutenberg-output
4 "(Lo) cra"          1
5 "1490"              1
6 "1498,"             1
7 "35"                1
8 "40,"              1
9 "A"                 2
10 "AS-IS".           1
11 "A_"               1
12 "Absoluti"         1
13 "Alack! 1
14 hduser@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.

The command `fs -getmerge` will simply concatenate any files it finds in the directory you specify. This means that the merged file might (and most likely will) **not be sorted**.

## Hadoop Web Interfaces

Hadoop comes with several web interfaces which are by default (see `conf/hadoop-default.xml`) available at these locations:

- <http://localhost:50070/> – web UI of the NameNode daemon
- <http://localhost:50030/> – web UI of the JobTracker daemon
- <http://localhost:50060/> – web UI of the TaskTracker daemon

These web interfaces provide concise information about what's happening in your Hadoop cluster. You might want to give them a try.

### NameNode Web Interface (HDFS layer)

The name node web UI shows you a cluster summary including information about total/remaining capacity, live and dead nodes. Additionally, it allows you to browse the HDFS namespace and view the contents of its files in the web browser. It also gives access to the local machine's Hadoop log files.

By default, it's available at <http://localhost:50070/>.

A screenshot of Hadoop's NameNode web interface



### JobTracker Web Interface (MapReduce layer)

The JobTracker web UI provides information about general job statistics of the Hadoop cluster, running/completed/failed jobs and a job history log file. It also gives access to the “local machine’s” Hadoop log files (the machine on which the web UI is running on).

By default, it’s available at <http://localhost:50030/>.

A screenshot of Hadoop's Job Tracker web interface



### TaskTracker Web Interface (MapReduce layer)

The task tracker web UI shows you running and non-running tasks. It also gives access to the “local machine’s” Hadoop log files.

By default, it’s available at <http://localhost:50060/>.



## What’s next?

If you’re feeling comfortable, you can continue your Hadoop experience with my follow-up tutorial [Running Hadoop On Ubuntu Linux \(Multi-Node Cluster\)](#) where I describe how to build a Hadoop “multi-node” cluster with two Ubuntu boxes (this will increase your current cluster size by 100%, heh).

In addition, I wrote [a tutorial](#) on [how to code a simple MapReduce job](#) in the Python programming language which can serve as the basis for writing your own MapReduce programs.

## Related Links

From yours truly:

- [Running Hadoop On Ubuntu Linux \(Multi-Node Cluster\)](#)
- [Writing An Hadoop MapReduce Program In Python](#)

From other people:

- [How to debug MapReduce programs](#)
- [Hadoop API Overview](#) (for Hadoop 2.x)

## Change Log

Only important changes to this article are listed here:

- 2011-07-17: Renamed the Hadoop user from `hadoop` to `hduser` based on readers' feedback. This should make the distinction between the local Hadoop user (now `hduser`), the local Hadoop group (`hadoop`), and the Hadoop CLI tool (`hadoop`) more clear.

Tweet 10

## Comments

772 comments

★ 28



Leave a message...

Best Community

Share



Frederic Stahl • 3 months ago

Hi,

I'm sorry to be a pain. Some time ago I tried this tutorial and I couldn't get it to work. I have now made a second attempt, and this time I belief I have followed it word by word with a fresh XUbuntu installation. I have indeed succeeded!!!

Well, then I tried it again a couple of hours later. I formatted the namenode again using `namenode -format`. Then I tried starting the cluster again. What always happens is that either the namenode or the datanode are not starting. Please find below the commandline output, I can't see anything odd happening.

Thanks in advance for the help!

Frederic

```
hduser@Else:/usr/local/hadoop/hadoop/bin$ ./start-all.sh
Warning: $HADOOP_HOME is deprecated.
```

```
starting namenode, logging to /usr/local/hadoop/hadoop/libexec/../logs/hadoop-hduser-namenode-Else.out
localhost: starting datanode, logging to /usr/local/hadoop/hadoop/libexec/../logs/hadoop-hduser-datanode-Else.out
localhost: starting secondarynamenode, logging to /usr/local/hadoop/hadoop/libexec/../logs/hadoop-hduser-secondarynamenode-Else.out
starting jobtracker, logging to /usr/local/hadoop/hadoop/libexec/../logs/hadoop-hduser-jobtracker-Else.out
```

[see more](#)

12 ^ | v Reply Share >



Aparajith Chandran • 8 months ago

Also when i try to start

I get the following errors

```
hduser@Aparajith-GNU:~$ /usr/local/hadoop/bin/start-all.sh
/usr/local/hadoop/libexec/../conf/hadoop-env.sh: line 1: '$\357\273\277#': command not found
/usr/local/hadoop/libexec/../conf/hadoop-env.sh: line 1: '$\357\273\277#': command not found
/usr/local/hadoop/libexec/../conf/hadoop-env.sh: line 1: '$\357\273\277#': command not found
/usr/local/hadoop/libexec/../conf/hadoop-env.sh: line 1: '$\357\273\277#': command not found
namenode running as process 3360. Stop it first.
/usr/local/hadoop/libexec/../conf/hadoop-env.sh: line 1: '$\357\273\277#': command not found
/usr/local/hadoop/libexec/../conf/hadoop-env.sh: line 1: '$\357\273\277#': command not found
/usr/local/hadoop/libexec/../conf/hadoop-env.sh: line 1: '$\357\273\277#': command not found
localhost: /home/hduser/.bashrc: line 1: syntax error near unexpected token `('
localhost: /home/hduser/.bashrc: line 1: `# ~/.bashrc: executed by bash(1) for non-login shells.'
localhost: /usr/local/hadoop/libexec/../conf/hadoop-env.sh: line 1: '$\357\273\277#': command not found
localhost: /usr/local/hadoop/libexec/../conf/hadoop-env.sh: line 1: '$\357\273\277#': command not found
localhost: datanode running as process 3585. Stop it first.
```

[see more](#)

9 ^ | v Reply Share >



**Nagarjuna Vissa** → Aparajith Chandran • 7 months ago

/usr/local/hadoop/hadoop-1.0.3/bin/start-all.sh try it

2 ^ | 1 v Reply Share ›



**Michael G. Noll** Owner → Aparajith Chandran • 7 months ago

@Aparajith Chandran: This looks like a copy-paste error to me -- apparently your [hadoop-env.sh](#) is messed up with weird characters.

^ | v Reply Share ›



Prathibha.P.G. • 9 months ago

Sir,

I have a problem. When I run [./start-all.sh](#), I have secondarynode only. When I use jps, there is no jobtracker, tasktracker, namenode etc. There is nothing written inside the log file.

Kindly help me. I am waiting for your reply

8 ^ | v Reply Share ›



**Melvin Lobo** → Prathibha.P.G. • 7 months ago

run [stop-all.sh](#)

Then run only namenode first and check.

bin/[hadoop-daemon.sh](#) start namenode

4 ^ | v Reply Share ›



**Ozan Kiraz** → Prathibha.P.G. • a month ago

This happened to me, I did clean install of Virtual machine, then did fresh install of hadoop using this tutorial second time and worked. Taking snapshots every time you got successful step is helpful.

Regards,

^ | v Reply Share ›



**@dasfaha** → Prathibha.P.G. • 8 months ago

Reformat the node and check that you are not getting any error. When I had that problem it was because I didn't notice the errors when I tried to format the node, and those errors were due to me specifying the hdfs directory incorrectly in this section:

```
<property>
  <name>hadoop.tmp.dir</name>
  <value>/app/hadoop/tmp</value>
  <description>A base for other temporary directories.</description>
</property>
```

^ | v Reply Share ›



Dietrich Speer • 10 months ago

The best tutorial in a while. Got it running on the first attempt on Ubuntu 12.04 using openjdk6. Beautiful!

8 ^ | v Reply Share ›



Saraj • 9 months ago

Hi Michael

I'm having a problem during the procedure. The terminal doesn't show any output at all when I format the file system, or start a new single node cluster. And on running the jps command, it shows only one jps process.

I earlier thought that this could be due to conflicting Java versions, so I purged them and installed only Sun Java 6, with Hadoop 1.0.4. Can you please suggest something?

Thanks

Saraj

6 ^ | v Reply Share ›



**Ekta Grover** → Saraj • 4 months ago

When you run "sudo jps" instead of (just) "jps", you will see all the nodes running. I saw that distinction in my session too :) (assuming you are logged in as hduser)

1 ^ | v Reply Share ›



Prathibha.P.G. • 9 months ago

I have started my map reduce job pi by giving command as given below. I am also giving the error. Please help me

```
mtech11@cse-desktop:~/hadoop1/bin$ hadoop jar hadoop1-examples-0.22.0.jar pi 3 10
Exception in thread "main" java.io.IOException: Error opening job jar: hadoop1-examples-0.22.0.jar
at org.apache.hadoop.util.RunJar.main(RunJar.java:130)
Caused by: java.util.zip.ZipException: error in opening zip file
```

```
at java.util.zip.ZipFile.open(Native Method)
at java.util.zip.ZipFile.(ZipFile.java:127)
at java.util.jar.JarFile.(JarFile.java:135)
at java.util.jar.JarFile.(JarFile.java:72)
at org.apache.hadoop.util.RunJar.main(RunJar.java:128)
4 ^ | v Reply Share >
```



**Melvin Lobo** → Prathibha.P.G • 7 months ago  
change your hadoop directory.

./bin/hadoop jar hadoop.examples-x.x.x.jar .....  
This should work  
^ | v Reply Share >



**Michael G. Noll** Owner → Prathibha.P.G • 8 months ago

@Prathibha P G This doesn't look like a correct Hadoop examples jar filename: "hadoop1-examples-0.22.0.jar". Are you sure you didn't misspell the filename?  
^ | v Reply Share >



**Prathibha.P.G** → Michael G. Noll • 8 months ago

Sir ,I have solved the problem and ran the job.After that the next day when I gave ./start-all.sh, datanode is not running.Can you give me a solution  
^ | v Reply Share >



**Karthik** → Prathibha.P.G • 7 months ago

If you have rebooted your machine, sshd might be stopped. Check if sshd is running..  
^ | v Reply Share >



**Prathibha P G** • 9 months ago

Sir,  
When I run ./start-all.sh jobtracker and tasktracker is not started.Also there is nothing written inside the log files.

Kindly help me  
4 ^ | v Reply Share >



**Bilal Saeed** • 5 months ago

I got it working at first attempt, ALHAMDULILAH :)  
Could you please let me know how can I get the web interface ?

Thanks :)  
3 ^ | v Reply Share >



**Abhi** • 6 months ago

Wonderful tutorial.. very well written. I started working on Linux today for the 1st time and still was able to execute this tutorial successfully. Thank you.  
3 ^ | v Reply Share >



**Greg Clarke** • 6 months ago

Really great tutorial! Thanks!  
2 ^ | v Reply Share >



**Mohamed Nadjib MAMI** • 6 months ago

Worked perfectly. Thanks you man for your very helpful tutorial. It's the best on the web till now. I'll continue with multi-node then the example with Python.  
2 ^ | v Reply Share >



**Md Shahnawaz Sakib** • 9 months ago

Great tutorial,if we follow the tutorial step by step it will give the expected output ,also the explanation is very clear,please sir provide us with some example to import data from MYSQL to HADOOP using SQOOPE.  
Great job :)  
2 ^ | v Reply Share >



**Minh Lê** • a year ago

I got the error: "mkdir: cannot create directory `/var/run/hadoop': Permission denied". I can run "sudo start-all.sh" but it defeats the purpose of using a normal dedicated user account.  
2 ^ | v Reply Share >



**Arun Kumar Venkatarama** • 4 months ago

This is the best tutorial on the hadoop setup. I got it run on a Cent OS linux machine with no difficulties.  
I would like to see more technical tutorials on hadoop and map reduce.  
1 ^ | v Reply Share >

**Abhisek Das** • 4 months ago

Awesome tutorial.. worked like charm in the first try..

Thanks michael !

1 ^ | v Reply Share &gt;

**peter mcgonigal** • 7 months ago

Hi Michael -

I'm hoping you can help me. I'm getting the following error when I run the WordCount example job.  
Have you any ideas why I'm getting this.

BTW, this is a great Hadoop tutorial.

Thanks in advance,  
Peter.

```
hduser@hadoop:/usr/local/hadoop$ bin/hadoop dfs -ls /home/hduser
Found 1 items
drwxr-xr-x  - hduser supergroup          0 2013-02-27 17:27 /home/hduser/gutenberg

hduser@hadoop:/usr/local/hadoop$ bin/hadoop dfs -ls /home/hduser/gutenberg
Found 3 items
-rw-r--r--  1 hduser supergroup    661811 2013-02-27 16:30 /home/hduser/gutenberg/pg20417.txt
```

[see more](#)

1 ^ | v Reply Share &gt;

**keval dalasaniya** → peter mcgonigal • 6 months ago

give the whole path of hdfs file llike hdfs://localhost:54310/home/hduser hdfs://localhost:54310/home/\*\*(your out put folder)

^ | v Reply Share &gt;

**Aparajith Chandran** • 8 months ago

Whenever i try to fromat i get this error

```
hduser@Aparajith-GNU:~$ /usr/local/hadoop/bin/hadoop namenode -format
/usr/local/hadoop/libexec/./conf/hadoop-env.sh: line 1: $'\357\273\277#': command not found
13/01/17 11:38:23 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:  host = Aparajith-GNU/127.0.1.1
STARTUP_MSG:  args = [-format]
STARTUP_MSG:  version = 1.1.1
STARTUP_MSG:  build = https://svn.apache.org/repos/a... -r 1411108; compiled by 'hortonfo' on Mon Nov 19 10:48:11 UTC 2012
*****/
Re-format filesystem in /app/hadoop/tmp/dfs/name ? (Y or N) y
Format aborted in /app/hadoop/tmp/dfs/name
13/01/17 13:26:22 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at Aparajith-GNU/127.0.1.1
*****/
```

1 ^ | v Reply Share &gt;

**Michael G. Noll** Owner → Aparajith Chandran • 7 months ago@Aparajith Chandran: This looks like a copy-paste error to me -- apparently your `hadoop-env.sh` is messed up with weird characters.

1 ^ | v Reply Share &gt;

**Anju Singh** → Aparajith Chandran • 8 months ago

delete dfs directory inside tmp and create an empty dfs folder and then format the namenode.  
It will solve your problem.

1 ^ | v Reply Share &gt;

**Kory Godfrey** → Aparajith Chandran • 7 months ago

Check the upper/lower case. When answering Y or N, use Y and not y. This seemed to work for me.

^ | v Reply Share &gt;

**Michael Griffiths** • 8 days ago

Thanks. this is a great tutorial. Just ran through it again. and vev everything works as it should. If there are anv readers out there who can answer the following

Thank you for a great tutorial! I've been struggling to get Hadoop running on Ubuntu Linux. I've been able to get it running on a single node, but I'm having trouble getting it to run on a multi-node cluster. I'm getting the following error when I try to run the wordcount example: java.net.UnknownHostException: ubuntu: ubuntu. I'm not sure what this means. Can you help me?

Many thanks

^ | v Reply Share >



Connor James Leech · 9 days ago

Hi there. I can't add your repository to my repositories, I get:

Fetches 1,941 kB in 25s (75.4 kB/s)

W: Failed to fetch <http://ppa.launchpad.net/ferra...> 404 Not Found

W: Failed to fetch <http://ppa.launchpad.net/ferra...> 404 Not Found

W: Failed to fetch <http://ppa.launchpad.net/ferra...> 404 Not Found

E: Some index files failed to download. They have been ignored, or old ones used instead.

Is there a different location I can download from?

^ | v Reply Share >



Mahesh · 9 days ago

i am getting a following error when i run the wordcount problem can you tell me the solution

13/09/17 07:13:59 WARN snappy.LoadSnappy: Snappy native library not loaded

^ | v Reply Share >



Michael G. Noll Owner → Mahesh · 9 days ago

@Mahesh: You can safely ignore this warning. You do not have the Snappy codec installed, but it's not needed for the WordCount example.

^ | v Reply Share >



Mahesh · 13 days ago

when i type following command it shows the error later it display the content of the directory. whats the problem can tell me

```
user@ubuntu:~/hadoop-1.2.0$ hadoop dfs -ls /
Warning: $HADOOP_HOME is deprecated.

13/09/13 07:07:54 INFO metrics.MetricsUtil: Unable to obtain hostName
java.net.UnknownHostException: ubuntu: ubuntu
    at java.net.InetAddress.getLocalHost(InetAddress.java:1454)
[...snip...]
Found 6 items
-rw-r--r-- 1 user supergroup 0 2013-09-11 15:14 /hadoop-1.2.0
drwxr-xr-x - user supergroup 0 2013-09-11 15:19 /home
drwxr-xr-x - user supergroup 0 2013-09-11 14:35 /tmp
drwxr-xr-x - user supergroup 0 2013-09-13 03:21 /user
-rw-r--r-- 1 user supergroup 0 2013-09-12 14:33 /wordcount.jar
drwxr-xr-x - user supergroup 0 2013-09-13 04:13 /wordcountproblem
```

^ | v Reply Share >



Michael G. Noll Owner → Mahesh · 13 days ago

@Mahesh: Seems like your networking settings are incorrect -- "unknown host exception: ubuntu".

^ | v Reply Share >



Harshad Tambekar · 15 days ago

Thanks Michael,

It is very nice tutorial. I have one question. I am trying to read from HDFS file generated by MapReduce program. I am using Java interface to connect and to read from HDFS file.

I can run my small java program and read successfully from HDFS file using url hdfs://localhost:54310/user/hduser/MyOutput/part-r-00000 on the machine where Hadoop is installed.

But when I am trying to read from another machine I am getting error message as 'org.apache.hadoop.ipc.Client\$Connection handleConnectionFailure'

I just put ip-address instead of 'localhost' in above url.

Is this because another machine is not configured for SSH?

Or is there some network problem. (I am able to ping from both machines to each other).

^ | v Reply Share >



Anindya Bhattacharya • 17 days ago

I am facing some problems while trying to compile the WordCount.java, and saw that the following files as mentioned in the import statements are failing to compile:

```
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
```

On having a closer look it looked they are part of Hadoop 2.0.1 (beta) and not part of the hadoop 1.2.1 (stable) library which I was using.

Please let me know if I need to use the Beta version of the library in order to compile this piece and make this work.

^ | v Reply Share ›



Michael G. Noll Owner → Anindya Bhattacharya • 13 days ago

@Anindya Bhattacharya: This tutorial is for Hadoop 1.x, not Hadoop 2.x.

^ | v Reply Share ›



Andre • 21 days ago

Is this tutorial for a 32-bit or 64-bit Version of Ubuntu ?

Or is that not relevant ?

^ | v Reply Share ›



Michael G. Noll Owner → Andre • 13 days ago

@Andre: 32/64 bit should not matter. I haven't tested it on 32bit though as all our stuff runs on 64bit.

^ | v Reply Share ›



pradeekyahvi • 25 days ago

Man Great Tutorial!

^ | v Reply Share ›



Debapriyo Sarkar • 25 days ago

I really appreciate you taking the time and posting this excellent tutorial. I have setup a single node Hadoop cluster in Solaris 11 by following your tutorial. Worked like a charm! Thank You!

^ | v Reply Share ›



ankit • a month ago

For all those new to ubuntu like me. Refer this link "<http://mysolvedproblem.blogspot...>" its written keeping the above post in mind but explains stuff this post assumes known.

Btw thanks a ton.

^ | v Reply Share ›



Tim Dunphy • a month ago

"Generally, using an empty password is not recommended, but in this case it is needed to unlock the key without your interaction (you don't want to enter the passphrase every time Hadoop interacts with its nodes)."

Actually, you should NEVER use pass-phraseless SSH keys under ANY circumstances!! But of course you still don't want to type it in every time. That's what ssh-agent is for, to cache your passphrase so you won't have to, yet you maintain the security of having one.

<http://kb.iu.edu/data/aeww.htm...>

^ | v Reply Share ›



Srinivasan R.K. • a month ago

I configured the Single node hadoop cluster as per instructions in Ubuntu...Each and every steps are excellent.

Thank,Srinivasan R K

^ | v Reply Share ›



Santhilata K V • a month ago

Thanks a lot. it helped me so much

^ | v Reply Share ›



Darknessss • 2 months ago

Great tutorial, it helped me successfully install hadoop on a debian installation (along with hbase and opentsdb).



Since some of my linux skills are a little rusty, I was hoping you could give some pointers to automatically start hadoop (or the [start-all.sh](#) script) at boot time so it automatically starts if the server reboots?

I tried to figure out how to start it as a service but no luck so far...

^ | v Reply Share ›



Hrushikesh Deshmukh • 2 months ago

Just awesome...worked like a charm in one go. Never seen so precise tutorial anywhere. Thanks a lot Michael!!!

^ | v Reply Share ›

Load more comments

#### ALSO ON MICHAEL G. NOLL

WHAT'S THIS?

### Understanding the Internal Message Buffers of Storm

7 comments • 3 months ago



Chitra Raveendran — Hi Michael. Great post, Helped understand easily some concepts I was struggling with. May I ask, how do we fix the number of ...

### Running a Multi-Node Storm Cluster

19 comments • 4 months ago



Michael G. Noll — @Stavros Kyriakopoulos: Having a dedicated machine for ZooKeeper or dual/triple/...-purposing machines for ZooKeeper is an ...

### Reading and Writing Avro Files from the Command Line

2 comments • 5 months ago



Joe Travaglini — Nice writeup - lol at the SC reference

### How To Extract Audio From FLV Files Using VLC

33 comments • 4 years ago



Tim — Reference: Windows v2.0.1 of VLC Media Player File Type: .FLV off YouTubeSettings for Audio Encoding: Did not change from Profile "Audio-MP3" ...

✉ Subscribe

➦ Add Disqus to your site

## About Me



I am a researcher and software engineer based in Switzerland, Europe. I work for the .COM and .NET DNS registry operator [Verisign](#) as the technical lead of its large-scale computing infrastructure based on the Apache Hadoop stack and as a research affiliate at [Verisign Labs](#). [Read more »](#)

## Contact

[michael@michael-noll.com](mailto:michael@michael-noll.com)

## Follow Me

[@miguno](#)

[Blog RSS](#)

[GitHub](#)

## Recent Posts

- [Replephant: Analyzing Hadoop Cluster Usage with Clojure](#)
- [Using Avro in MapReduce jobs with Hadoop. Pig. Hive](#)
- [Understanding the Internal Message Buffers of Storm](#)
- [Installing and Running Graphite via RPM and Supervisor](#)
- [Multi-Node Storm Cluster Tutorial Published](#)

## Latest Tweets

- Status updating...

Follow @miguno

Copyright © 2004-2013 [Michael G. Noll](#). All rights reserved. Views expressed here are my own. [Privacy Policy](#). Powered by [Octopress](#).