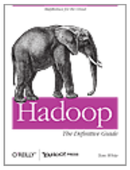
# Installing Apache Hadoop - Hadoop: The Definitive Guide

by Tom White

Print
Subscribe to Newsletters
ShareThis

At the time of this writing, Git is (seemingly) not installed by default on any GNU/Linux distribution or any other operating system. So, before you can use Git, you must install it. The steps to install Git depend greatly on the vendor and version of your operating system. This chapter describes how to install Git on Linux and Microsoft Windows and within Cygwin.



This excerpt is from **Hadoop: The Definitive Guide**.Hadoop: The Definitive Guide is a comprehensive resource for using Hadoop to build reliable, scalable, distributed systems. Programmers will find details for analyzing large datasets with Hadoop, and administrators will learn how to set up and run Hadoop clusters. The book includes case studies that illustrate how Hadoop is used to solve specific problems.

It's easy to install Hadoop on a single machine to try it out. (For installation on a cluster, please refer to Chapter 9, *Setting Up a Hadoop Cluster*.) The quickest way is to download and run a binary release from an Apache Software Foundation Mirror.

In this appendix, we cover how to install Hadoop Core, HDFS, and MapReduce. Instructions for installing Pig, HBase, and ZooKeeper are included in the relevant chapter (Chapters 11, 12, and 13).

## Prerequisites

Hadoop is written in Java, so you will need to have Java installed on your machine, version 6 or later. Sun's JDK is the one most widely used with Hadoop, although others have been reported to work.

Hadoop runs on Unix and on Windows. Linux is the only supported production platform, but other flavors of Unix (including Mac OS X) can be used to run Hadoop for development. Windows is only supported as a development platform, and additionally requires Cygwin to run. During the Cygwin installation process, you should include the openssh package if you plan to run Hadoop in pseudo-distributed mode (see following explanation).

## Installation

Start by deciding which user you'd like to run Hadoop as. For trying out Hadoop or developing Hadoop programs, it is simplest to run Hadoop on a single machine using your own user account.

Download a stable release, which is packaged as a gzipped tar file, from the Apache Hadoop releases page (http://hadoop.apache.org/core/releases.html) and unpack it somewhere on your filesystem:

```
% tar xzf hadoop-x.y.z.tar.gz
```

Before you can run Hadoop, you need to tell it where Java is located on your system. If you have the `JAVA_HOME` environment variable set to point to a suitable Java installation, that will be used, and you don't have to configure anything further. Otherwise, you can set the Java installation that Hadoop uses by editing `conf/hadoop-env.sh`, and specifying the `JAVA_HOME` variable. For example, on my Mac I changed the line to read:

```
export JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/Versions/1.6.0/Ho
```

to point to version 1.6.0 of Java. On Ubuntu, the equivalent line is:

```
export JAVA_HOME=/usr/lib/jvm/java-6-sun
```

It's very convenient to create an environment variable that points to the Hadoop installation directory (`HADOOP_INSTALL`, say) and to put the Hadoop binary directory on your command-line path. For example:

```
% export HADOOP_INSTALL=/home/tom/hadoop-x.y.z

% export PATH=$PATH:$HADOOP_INSTALL/bin
```

Check that Hadoop runs by typing:

```
% hadoop version
Hadoop 0.20.0
Subversion https://svn.apache.org/repos/asf/hadoop/core/branches/branch-0.20 -
Compiled by ndaley on Thu Apr  9 05:18:40 UTC 2009
```

## Configuration

Each component in Hadoop is configured using an XML file. Core properties go in `core-site.xml`, HDFS properties go in `hdfs-site.xml`, and MapReduce properties go in `mapred-site.xml`. These files are all located in the `conf` subdirectory.

### Note

In earlier versions of Hadoop, there was a single site configuration file for the Core, HDFS, and MapReduce components, called `hadoop-site.xml`. From release 0.20.0 onward this file has been split into three: one for each component. The property names have not changed, just the configuration file they have to go in. You can see the default settings for all the properties that are governed by these configuration files by looking in the `docs` directory of your Hadoop installation for HTML files called `core-default.html`, `hdfs-default.html`, and `mapred-default.html`.

Hadoop can be run in one of three modes:

**Standalone (or local) mode**

There are no daemons running and everything runs in a single JVM. Standalone mode is suitable for running MapReduce programs during development, since it is easy to test and debug them.

**Pseudo-distributed mode**

The Hadoop daemons run on the local machine, thus simulating a cluster on a small scale.

**Fully distributed mode**

The Hadoop daemons run on a cluster of machines. This setup is described in Chapter 9, *Setting Up a Hadoop Cluster*.

To run Hadoop in a particular mode, you need to do two things: set the appropriate properties, and start the Hadoop daemons. Table A.1, "Key configuration properties for different modes" shows the minimal set of properties to configure each mode. In standalone mode, the local filesystem and the local MapReduce job runner are used, while in the distributed modes the HDFS and MapReduce daemons are started.

**Table A.1. Key configuration properties for different modes**

| Component | Property | Standalone | Pseudo-distributed | Fully distributed |
|---|---|---|---|---|
| Core | `fs.default.name` | `file:///` (default) | `hdfs://localhost/` | `hdfs://`*`namenode`*`/` |
| HDFS | `dfs.replication` | N/A | `1` | `3` (default) |
| MapReduce | `mapred.job.tracker` | `local` (default) | `localhost:8021` | *`jobtracker`*`:8021` |

You can read more about configuration in the section called "Hadoop Configuration".

### Standalone Mode

In standalone mode, there is no further action to take, since the default properties are set for standalone mode, and there are no daemons to run.

### Pseudo-Distributed Mode

The configuration files should be created with the following contents, and placed in the `conf` directory (although you can place configuration files in any directory as long as you start the daemons with the `--config` option).

```
<?xml version="1.0"?>
<!-- core-site.xml -->
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost/</value>

  </property>
</configuration>
```

```
<?xml version="1.0"?>
<!-- hdfs-site.xml -->
<configuration>
  <property>
    <name>dfs.replication</name>

    <value>1</value>
  </property>
</configuration>
```

```
<?xml version="1.0"?>
<!-- mapred-site.xml -->
<configuration>
  <property>

    <name>mapred.job.tracker</name>
    <value>localhost:8021</value>
  </property>
</configuration>
```

#### Configuring SSH

In pseudo-distributed mode, we have to start daemons, and to do that, we need to have SSH installed. Hadoop doesn't actually distinguish between pseudo-distributed and fully distributed modes: it merely starts daemons on the set of hosts in the cluster (defined by the `slaves` file) by SSH-ing to each host and starting a daemon process. Pseudo-distributed mode is just a special case of fully distributed mode in which the (single) host is localhost, so we need to make sure that we can SSH to localhost and log in without having to enter a password.

First, make sure that SSH is installed and a server is running. On Ubuntu, for example, this is achieved with:

```
% sudo apt-get install ssh
```

### Note

On Windows with Cygwin, you can set up an SSH server (after having installed the `openssh` package) by running `ssh-host-config -y`.

Then to enable password-less login, generate a new SSH key with an empty passphrase:

```
% ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
```

```
% cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Test this with:

```
% ssh localhost
```

You should be logged in without having to type a password.

#### Formatting the HDFS filesystem

Before it can be used, a brand-new HDFS installation needs to be formatted. The formatting process creates an empty filesystem by creating the storage directories and the initial versions of the namenode's persistent data structures. Datanodes are not involved in the initial formatting process, since the namenode manages all of the filesystem's metadata, and datanodes can join or leave the cluster dynamically. For the same reason, you don't need to say how large a filesystem to create, since this is determined by the number of datanodes in the cluster, which can be increased as needed, long after the filesystem was formatted.

Formatting HDFS is quick to do. Just type the following:

```
% hadoop namenode -format
```

### Starting and stopping the daemons

To start the HDFS and MapReduce daemons, type:

```
% start-dfs.sh

% start-mapred.sh
```

## Note

If you have placed configuration files outside the default `conf` directory, start the daemons with the `--config` option, which takes an absolute path to the configuration directory:

```
% start-dfs.sh --config path-to-config-directory
% start-mapred.sh --config path-to-config-directory
```

Three daemons will be started on your local machine: a namenode, a secondary namenode, and a datanode. You can check whether the daemons started successfully by looking at the logfiles in the `logs` directory (in the Hadoop installation directory), or by looking at the web UIs, at `http://localhost:50030/` for the jobtracker, and at `http://localhost:50070/` for the namenode. You can also use Java's `jps` command to see whether they are running.

Stopping the daemons is done in the obvious way:

```
% stop-dfs.sh

% stop-mapred.sh
```

## Fully Distributed Mode

Setting up a cluster of machines brings many additional considerations, so this mode is covered in Chapter 9, *Setting Up a Hadoop Cluster*.

If you enjoyed this excerpt, buy a copy of **Hadoop: The Definitive Guide**.

**About O'Reilly**

: Sign In
Academic Solutions
Jobs
Contacts
Corporate Information
Press Room
Privacy Policy
Terms of Service
Writing for O'Reilly

**Community**

Authors
Community & Featured Users
Forums
Membership
Newsletters
O'Reilly Answers
RSS Feeds
User Groups

**More O'Reilly Sites**

igniteshow.com
makerfaire.com
makezine.com
craftzine.com
labs.oreilly.com

**Partner Sites**

PayPal Developer Zone
O'Reilly Insights on Forbes.com

**Shop O'Reilly**

Customer Service
Contact Us
Shipping Information
Ordering & Payment
The O'Reilly Guarantee