

# Hadoop Quickstart

## Table of contents

1 Purpose.....	2
2 Pre-requisites.....	2
2.1 Supported Platforms.....	2
2.2 Required Software.....	2
2.3 Installing Software.....	2
3 Download.....	2
4 Standalone Operation.....	3
5 Pseudo-Distributed Operation.....	3
5.1 Configuration.....	3
5.2 Setup passphraseless ssh.....	4
5.3 Execution.....	4
6 Fully-Distributed Operation.....	5

## 1. Purpose

The purpose of this document is to help users get a single-node Hadoop installation up and running very quickly so that users can get a flavour of the [Hadoop Distributed File System \(HDFS\)](#) and the Map-Reduce framework i.e. perform simple operations on HDFS, run example/simple jobs etc.

## 2. Pre-requisites

### 2.1. Supported Platforms

- Hadoop has been demonstrated on GNU/Linux clusters with 2000 nodes.
- Win32 is supported as a *development platform*. Distributed operation has not been well tested on Win32, so this is not a *production platform*.

### 2.2. Required Software

1. Java™ 1.5.x, preferably from Sun, must be installed. Set JAVA\_HOME to the root of your Java installation.
2. **ssh** must be installed and **sshd** must be running to use the Hadoop scripts that manage remote Hadoop daemons.

#### 2.2.1. Additional requirements for Windows

1. [Cygwin](#) - Required for shell support in addition to the required software above.

### 2.3. Installing Software

If your cluster doesn't have the requisite software you will need to install it.

For example on Ubuntu Linux:

```
$ sudo apt-get install ssh
$ sudo apt-get install rsync
```

On Windows, if you did not install the required software when you installed cygwin, start the cygwin installer and select the packages:

- openssh - the *Net* category

## 3. Download

First, you need to get a Hadoop distribution: download a recent [stable release](#) and unpack it.

Once done, in the distribution edit the file `conf/hadoop-env.sh` to define at least `JAVA_HOME`.

Try the following command:

```
$ bin/hadoop
```

This will display the usage documentation for the **hadoop** script.

## 4. Standalone Operation

By default, Hadoop is configured to run things in a non-distributed mode, as a single Java process. This is useful for debugging.

The following example copies the unpacked `conf` directory to use as input and then finds and displays every match of the given regular expression. Output is written to the given output directory.

```
$ mkdir input
$ cp conf/*.xml input
$ bin/hadoop jar hadoop-*-examples.jar grep input output
'dfs[a-z.]+ '
$ cat output/*
```

## 5. Pseudo-Distributed Operation

Hadoop can also be run on a single-node in a pseudo-distributed mode where each Hadoop daemon runs in a separate Java process.

### 5.1. Configuration

Use the following `conf/hadoop-site.xml`:

<configuration>
<property>
<name>fs.default.name</name>
<value>localhost:9000</value>
</property>
<property>
<name>mapred.job.tracker</name>
<value>localhost:9001</value>

</property>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>

## 5.2. Setup passphraseless ssh

Now check that you can ssh to the localhost without a passphrase:

```
$ ssh localhost
```

If you cannot ssh to localhost without a passphrase, execute the following commands:

```
$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

## 5.3. Execution

Format a new distributed-filesystem:

```
$ bin/hadoop namenode -format
```

Start The hadoop daemons:

```
$ bin/start-all.sh
```

The hadoop daemon log output is written to the `$(HADOOP_LOG_DIR)` directory (defaults to `$(HADOOP_HOME)/logs`).

Browse the web-interface for the NameNode and the JobTracker, by default they are available at:

- NameNode - <http://localhost:50070/>
- JobTracker - <http://localhost:50030/>

Copy the input files into the distributed filesystem:

```
$ bin/hadoop dfs -put conf input
```

Run some of the examples provided:

```
$ bin/hadoop jar hadoop-*-examples.jar grep input output
'dfs[a-z. ]+'
```

Examine the output files:

Copy the output files from the distributed filesystem to the local filesystem and examine them:

```
$ bin/hadoop dfs -get output output  
$ cat output/*
```

or

View the output files on the distributed filesystem:

```
$ bin/hadoop dfs -cat output/*
```

When you're done, stop the daemons with:

```
$ bin/stop-all.sh
```

## **6. Fully-Distributed Operation**

Information on setting up fully-distributed non-trivial clusters can be found [here](#).

*Java and JNI are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.*