

# MAPREDUCE - INSTALLATION

[http://www.tutorialspoint.com/map\\_reduce/map\\_reduce\\_installation.htm](http://www.tutorialspoint.com/map_reduce/map_reduce_installation.htm)

Copyright © tutorialspoint.com

MapReduce works only on Linux flavored operating systems and it comes inbuilt with a Hadoop Framework. We need to perform the following steps in order to install Hadoop framework.

## Verifying JAVA Installation

Java must be installed on your system before installing Hadoop. Use the following command to check whether you have Java installed on your system.

```
$ java -version
```

If Java is already installed on your system, you get to see the following response –

```
java version "1.7.0_71"  
Java(TM) SE Runtime Environment (build 1.7.0_71-b13)  
Java HotSpot(TM) Client VM (build 25.0-b02, mixed mode)
```

In case you don't have Java installed on your system, then follow the steps given below.

## Installing Java

### Step 1

Download the latest version of Java from the following link – [this link](#).

After downloading, you can locate the file **jdk-7u71-linux-x64.tar.gz** in your Downloads folder.

### Step 2

Use the following commands to extract the contents of jdk-7u71-linux-x64.gz.

```
$ cd Downloads/  
$ ls  
jdk-7u71-linux-x64.gz  
$ tar zxf jdk-7u71-linux-x64.gz  
$ ls  
jdk1.7.0_71 jdk-7u71-linux-x64.gz
```

### Step 3

To make Java available to all the users, you have to move it to the location “/usr/local/”. Go to root and type the following commands –

```
$ su  
password:  
# mv jdk1.7.0_71 /usr/local/java  
# exit
```

### Step 4

For setting up PATH and JAVA\_HOME variables, add the following commands to ~/.bashrc file.

```
export JAVA_HOME=/usr/local/java  
export PATH=$PATH:$JAVA_HOME/bin
```

Apply all the changes to the current running system.

```
$ source ~/.bashrc
```

## Step 5

Use the following commands to configure Java alternatives –

```
# alternatives --install /usr/bin/java java usr/local/java/bin/java 2
# alternatives --install /usr/bin/javac javac usr/local/java/bin/javac 2
# alternatives --install /usr/bin/jar jar usr/local/java/bin/jar 2
# alternatives --set java usr/local/java/bin/java
# alternatives --set javac usr/local/java/bin/javac
# alternatives --set jar usr/local/java/bin/jar
```

Now verify the installation using the command **java -version** from the terminal.

## Verifying Hadoop Installation

Hadoop must be installed on your system before installing MapReduce. Let us verify the Hadoop installation using the following command –

```
$ hadoop version
```

If Hadoop is already installed on your system, then you will get the following response –

```
Hadoop 2.4.1
--
Subversion https://svn.apache.org/repos/asf/hadoop/common -r 1529768
Compiled by hortonmu on 2013-10-07T06:28Z
Compiled with protoc 2.5.0
From source with checksum 79e53ce7994d1628b240f09af91e1af4
```

If Hadoop is not installed on your system, then proceed with the following steps.

## Downloading Hadoop

Download Hadoop 2.4.1 from Apache Software Foundation and extract its contents using the following commands.

```
$ su
password:
# cd /usr/local
# wget http://apache.claz.org/hadoop/common/hadoop-2.4.1/
hadoop-2.4.1.tar.gz
# tar xzf hadoop-2.4.1.tar.gz
# mv hadoop-2.4.1/* to hadoop/
# exit
```

## Installing Hadoop in Pseudo Distributed mode

The following steps are used to install Hadoop 2.4.1 in pseudo distributed mode.

### Step 1 – Setting up Hadoop

You can set Hadoop environment variables by appending the following commands to `~/bashrc` file.

```
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
```

```
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
```

Apply all the changes to the current running system.

```
$ source ~/.bashrc
```

## Step 2 – Hadoop Configuration

You can find all the Hadoop configuration files in the location “\$HADOOP\_HOME/etc/hadoop”. You need to make suitable changes in those configuration files according to your Hadoop infrastructure.

```
$ cd $HADOOP_HOME/etc/hadoop
```

In order to develop Hadoop programs using Java, you have to reset the Java environment variables in **hadoop-env.sh** file by replacing JAVA\_HOME value with the location of Java in your system.

```
export JAVA_HOME=/usr/local/java
```

You have to edit the following files to configure Hadoop –

- core-site.xml
- hdfs-site.xml
- yarn-site.xml
- mapred-site.xml

### core-site.xml

core-site.xml contains the following information–

- Port number used for Hadoop instance
- Memory allocated for the file system
- Memory limit for storing the data
- Size of Read/Write buffers

Open the core-site.xml and add the following properties in between the <configuration> and </configuration> tags.

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:9000 </value>
  </property>
</configuration>
```

### hdfs-site.xml

hdfs-site.xml contains the following information –

- Value of replication data
- The namenode path
- The datanode path of your local file systems *the place where you want to store the Hadoop infra*

Let us assume the following data.

```
dfs.replication (data replication value) = 1
```

(In the following path /hadoop/ is the user name.  
hadoopinfra/hdfs/namenode is the directory created by hdfs file system.)  
namenode path = //home/hadoop/hadoopinfra/hdfs/namenode

(hadoopinfra/hdfs/datanode is the directory created by hdfs file system.)  
datanode path = //home/hadoop/hadoopinfra/hdfs/datanode

Open this file and add the following properties in between the <configuration>, </configuration> tags.

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

  <property>
    <name>dfs.name.dir</name>
    <value>file:///home/hadoop/hadoopinfra/hdfs/namenode</value>
  </property>

  <property>
    <name>dfs.data.dir</name>
    <value>file:///home/hadoop/hadoopinfra/hdfs/datanode </value>
  </property>
</configuration>
```

**Note** – In the above file, all the property values are user-defined and you can make changes according to your Hadoop infrastructure.

## yarn-site.xml

This file is used to configure yarn into Hadoop. Open the yarn-site.xml file and add the following properties in between the <configuration>, </configuration> tags.

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

## mapred-site.xml

This file is used to specify the MapReduce framework we are using. By default, Hadoop contains a template of yarn-site.xml. First of all, you need to copy the file from mapred-site.xml.template to mapred-site.xml file using the following command.

```
$ cp mapred-site.xml.template mapred-site.xml
```

Open mapred-site.xml file and add the following properties in between the <configuration>, </configuration> tags.

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

## Verifying Hadoop Installation

The following steps are used to verify the Hadoop installation.

## Step 1 – Name Node Setup

Set up the namenode using the command “hdfs namenode -format” as follows –

```
$ cd ~
$ hdfs namenode -format
```

The expected result is as follows –

```
10/24/14 21:30:55 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = localhost/192.168.1.11
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 2.4.1
...
...
10/24/14 21:30:56 INFO common.Storage: Storage directory
/home/hadoop/hadoopinfra/hdfs/namenode has been successfully formatted.
10/24/14 21:30:56 INFO namenode.NNStorageRetentionManager: Going to
retain 1 images with txid >= 0
10/24/14 21:30:56 INFO util.ExitUtil: Exiting with status 0
10/24/14 21:30:56 INFO namenode.NameNode: SHUTDOWN_MSG:

/*****
SHUTDOWN_MSG: Shutting down NameNode at localhost/192.168.1.11
*****/
```

## Step 2 – Verifying Hadoop dfs

Execute the following command to start your Hadoop file system.

```
$ start-dfs.sh
```

The expected output is as follows –

```
10/24/14 21:37:56
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/hadoop/hadoop-
2.4.1/logs/hadoop-hadoop-namenode-localhost.out
localhost: starting datanode, logging to /home/hadoop/hadoop-
2.4.1/logs/hadoop-hadoop-datanode-localhost.out
Starting secondary namenodes [0.0.0.0]
```

## Step 3 – Verifying Yarn Script

The following command is used to start the yarn script. Executing this command will start your yarn daemons.

```
$ start-yarn.sh
```

The expected output is as follows –

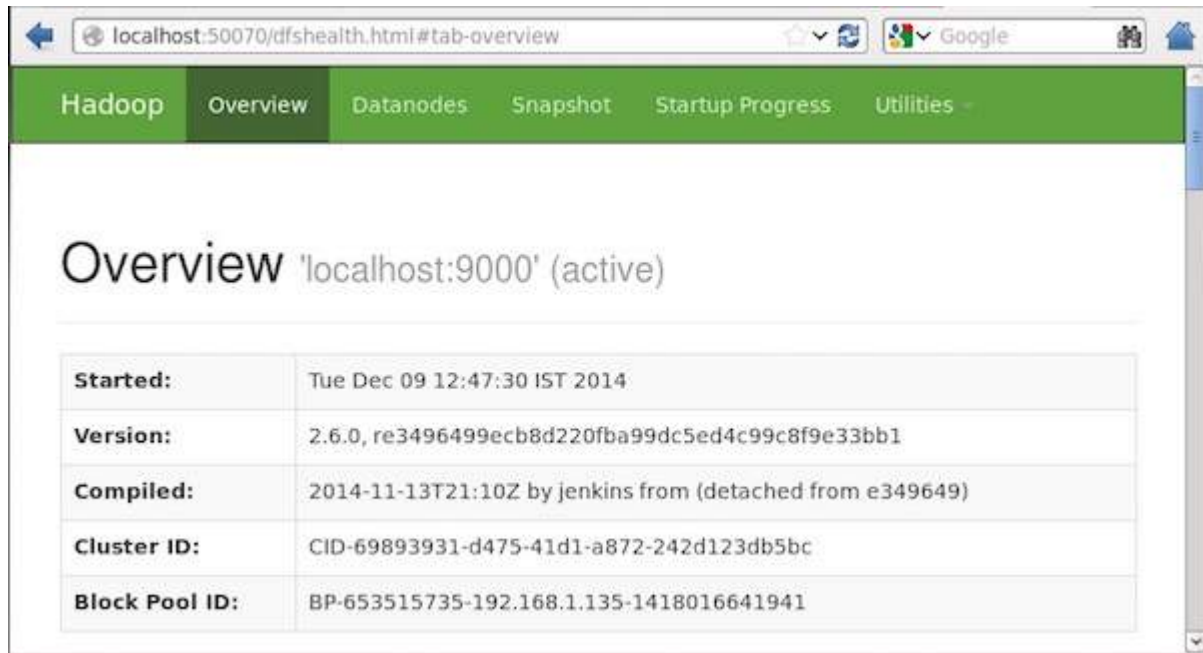
```
starting yarn daemons
starting resourcemanager, logging to /home/hadoop/hadoop-
2.4.1/logs/yarn-hadoop-resourcemanager-localhost.out
localhost: starting node manager, logging to /home/hadoop/hadoop-
2.4.1/logs/yarn-hadoop-nodemanager-localhost.out
```

## Step 4 – Accessing Hadoop on Browser

The default port number to access Hadoop is 50070. Use the following URL to get Hadoop services on your browser.

```
http://localhost:50070/
```

The following screenshot shows the Hadoop browser.



## Step 5 – Verify all Applications of a Cluster

The default port number to access all the applications of a cluster is 8088. Use the following URL to use this service.

```
http://localhost:8088/
```

The following screenshot shows a Hadoop cluster browser.

