

# Hadoop Installation Steps

*Since I am using Ubuntu as my operating system, all the setup steps are for the linux environment only.*

## 1. SSH Setup and Key Generation

SSH setup is required to do different operations on a cluster such as starting, stopping, distributed daemon shell operations. To authenticate different users of Hadoop, it is required to provide public/private key pair for a Hadoop user and share it with different users.

```
$ ssh-keygen -t rsa
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ chmod 0600 ~/.ssh/authorized_keys
```

## 2. Install Java

- Check if Java is already installed

```
$ java -version
```

### Step 1:

If you did not see anything, you have to install Java JDK (download the latest version).

### Step 2:

After download Java JDK (jdk1.8.0\_91 is the current latest version), generally you will find the downloaded java file in Downloads folder, verify it and extract the jdk-8u91-linux-x64.gz file using the following commands.

```
$ cd Downloads/
$ ls
jdk-8u91-linux-x64.gz
$ tar xzf jdk-8u91-linux-x64.gz
$ ls
jdk1.8.0_91  jdk-8u91-linux-x64.gz
```

### Step 3:

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

```
$ su
password:
# mv jdk1.8.0_91 /usr/local/
# exit
```

#### Step 4:

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

```
export JAVA_HOME=/usr/local/jdk1.8.0_91
export PATH=$PATH:$JAVA_HOME/bin
```

Now apply all the changes into the current running system.

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

### 3. Download and extract Hadoop to /usr/local folder

Download and extract Hadoop 2.4.6 from Apache software foundation using the following commands.

```
$ su
password:
# cd /usr/local
# wget http://apache.claz.org/hadoop/common/hadoop-2.6.4/hadoop-2.6.4.tar.gz/
hadoop-2.4.1.tar.gz
# tar xzf hadoop-2.4.6.tar.gz
# mv hadoop-2.4.6/* to hadoop/
# exit
```

### 4. Install Hadoop in Standalone Mode

#### Step 1: Setting Up Hadoop

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

```
export HADOOP_HOME=/home/your_username/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
export HADOOP_INSTALL=$HADOOP_HOME
```

Now apply all the changes into 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”. It is required to make changes in those configuration files according to your Hadoop infrastructure.

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

In order to develop Hadoop programs in 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/jdk1.8.0_91
```

The following are the list of files that you have to edit to configure Hadoop.

### core-site.xml

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

### hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.name.dir</name>
    <value>file:///home/your_username/hadoop/hadoopinfra/hdfs/namenode </value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>file:///home/your_username/hadoop/hadoopinfra/hdfs/datanode </value>
  </property>
</configuration>
```

```
</property>
</configuration>
```

### 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 in this file.

yarn.log-aggregation-enable is to enable log history

```
<configuration>

  <property>

    <name>yarn.nodemanager.aux-services</name>

    <value>mapreduce_shuffle</value>

  </property>

  <property>

    <name>yarn.log-aggregation-enable</name>

    <value>true</value>

  </property>
</configuration>
```

### mapred-site.xml

By default, Hadoop contains a template of yarn-site.xml. First of all, it is required 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
```

copy the configuration to mapred-site.xml

```
<configuration>

  <property>

    <name>mapreduce.framework.name</name>

    <value>yarn</value>

  </property>
</configuration>
```

## 5. 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
```

### ***Step 2: Verifying Hadoop dfs***

The following command is used to start dfs. Executing this command will start your Hadoop file system.

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

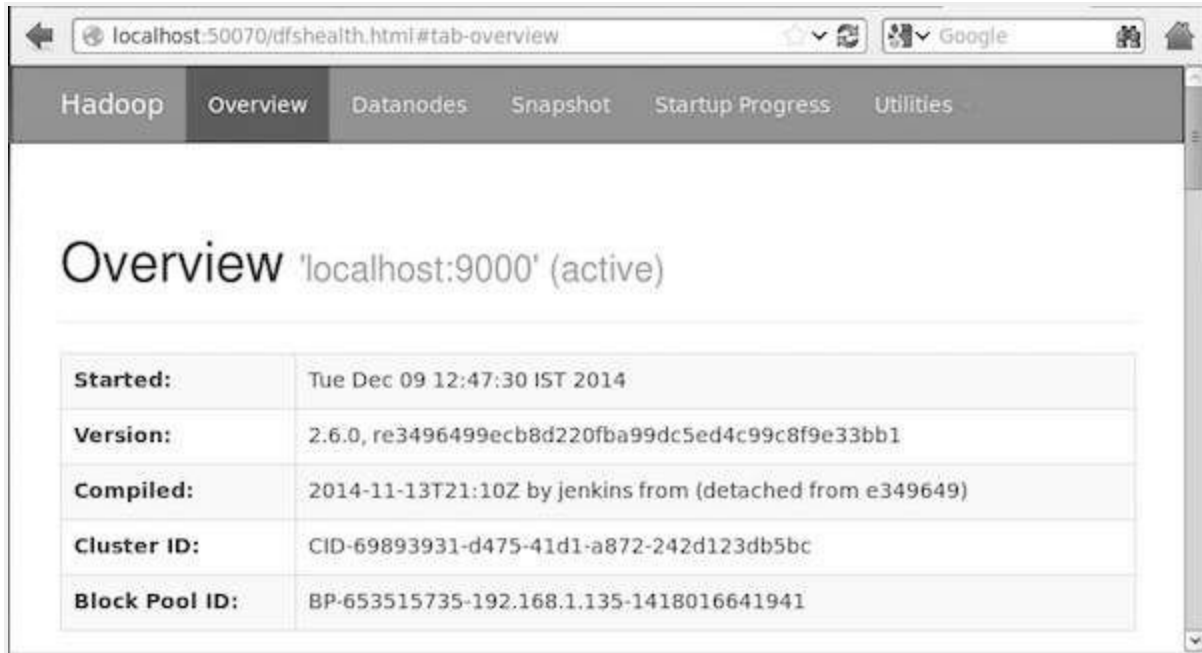
If there is a problem starting the ssh services, then maybe you did not install sshd in your system. Go to this link :

<http://cloudfront.blogspot.in/2012/07/how-to-setup-and-configure-ssh-on-ubuntu.html#.UcvbF0AW38t>

### ***Step 3: Verifying Hadoop on browser***

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

<http://localhost:50070/>

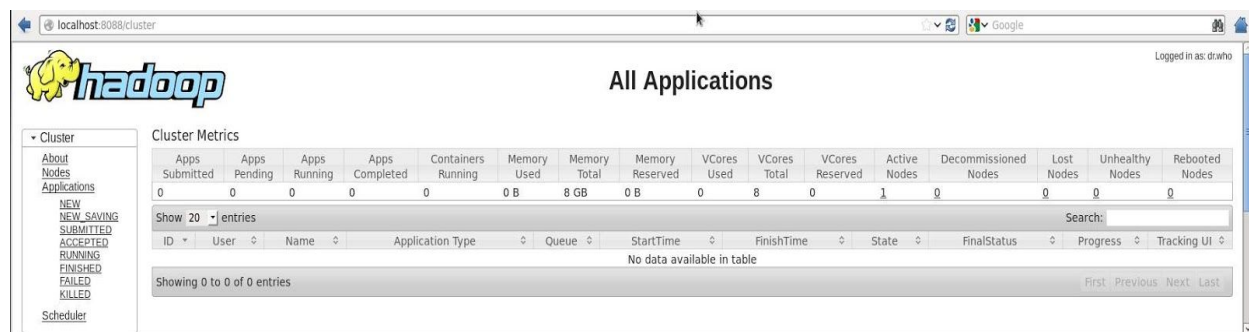


| Overview 'localhost:9000' (active) |   |
|------------------------------------|---|
| <b>Started:</b>                    | Tue Dec 09 12:47:30 IST 2014                              |
| <b>Version:</b>                    | 2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1          |
| <b>Compiled:</b>                   | 2014-11-13T21:10Z by jenkins from (detached from e349649) |
| <b>Cluster ID:</b>                 | CID-69893931-d475-41d1-a872-242d123db5bc                  |
| <b>Block Pool ID:</b>              | BP-653515735-192.168.1.135-1418016641941                  |

### Step 5: Verify All Applications for Cluster

The default port number to access all applications of cluster is 8088. Use the following url to visit this service.

<http://localhost:8088/>



## 6. Testing the WordCount sample

- a. Download the WordCount sample

Go to this link:

[http://www.cloudera.com/documentation/other/tutorial/CDH5/Hadoop-Tutorial/ht\\_wordcount1\\_source.html](http://www.cloudera.com/documentation/other/tutorial/CDH5/Hadoop-Tutorial/ht_wordcount1_source.html)

Then click download link for all source code.

- b. Install Eclipse JavaEE

Go to <https://www.eclipse.org/downloads/>

Then download and install eclipse for your pc.

- c. Create a new Maven project with the Hadoop dependencies in pom.xml file  
Modify your pom.xml file which look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>WordCount</groupId>
    <artifactId>WordCount</artifactId>
    <version>1.0</version>
    <dependencies>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-client</artifactId>
            <version>2.7.1</version>
        </dependency>
    </dependencies>

</project>

```

If you have error with this pom.xml file, update your project by:

Right click on your project -> Maven -> Update Project -> Check the "Force Update of Snapshots/Release" box.

- d. Copy the "WordCount.java" to this new project, i.e, src -> main -> java
- e. Clean and install your Maven project; Right click on your project -> Run As -> Maven -> Maven Clean  
Then Right click on your project -> Run As -> Maven -> Maven Install  
This step will create your .jar file in the "target" folder.
- f. Create a folder "input" to store all the input files (the same level to "src")
- g. Run your hadoop command in the terminal:  
Create a batch file, "wordcountbatch", to run multiple commands at the same time.

```

#!/bin/bash
clear
echo "Running WordCount Approach"
hadoop fs -rm -r wordcount/input
hadoop fs -rm -r wordcount/output
hadoop fs -mkdir -p wordcount
hadoop fs -put input wordcount

```

```
cd target
hadoop jar WordCount-1.0.jar edu.mum.cs522.WordCount
wordcount/input wordcount/output
echo "wordcount Approach - Output"
hadoop fs -cat wordcount/output/*
```

Then just type “./**wordcountbatch**” in the terminal, it will run the wordcount project. This command will produce the output of the word count into the terminal, and we can check the output file with this link:

<http://localhost:50070/explorer.html#/>

## References

[http://www.tutorialspoint.com/hadoop/hadoop\\_environment\\_setup.htm](http://www.tutorialspoint.com/hadoop/hadoop_environment_setup.htm)

[http://www.cloudera.com/documentation/other/tutorial/CDH5/Hadoop-Tutorial/ht\\_wordcount1.html](http://www.cloudera.com/documentation/other/tutorial/CDH5/Hadoop-Tutorial/ht_wordcount1.html)