# SQOOP - INSTALLATION

http://www.tutorialspoint.com/sqoop/sqoop_installation.htm

As Sqoop is a sub-project of Hadoop, it can only work on Linux operating system. Follow the steps given below to install Sqoop on your system.

## Step 1: Verifying JAVA Installation

You need to have Java installed on your system before installing Sqoop. Let us verify Java installation using the following command:

```
$ 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)
```

If Java is not installed on your system, then follow the steps given below.

## Installing Java

Follow the simple steps given below to install Java on your system.

## Step 1

Download Java $JDK < latestversion > -X64.tar.gz$ by visiting the following link.

Then jdk-7u71-linux-x64.tar.gz will be downloaded onto your system.

## Step 2

Generally, you can find the downloaded Java file in the Downloads folder. Verify it and extract the jdk-7u71-linux-x64.gz file using the following commands.

```
$ 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/”. Open root, and type the following commands.

```
$ su
password:

# mv jdk1.7.0_71 /usr/local/java
# exitStep IV:
```

## 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
```

Now apply all the changes into 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 as explained above.

## Step 2: Verifying Hadoop Installation

Hadoop must be installed on your system before installing Sqoop. 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 and extract Hadoop 2.4.1 from Apache Software Foundation 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

Follow the steps given below 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
```

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”. 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
```

Given below is the list of files that you need to edit to configure Hadoop.

### core-site.xml

The core-site.xml file contains information such as the port number used for Hadoop instance, memory allocated for the file system, memory limit for storing the data, and the 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

The hdfs-site.xml file contains information such as the value of replication data, namenode path, and datanode path of your local file systems. It means the place where you want to store the Hadoop infrastructure.

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 in this file.

```
<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 in this file.

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

### mapred-site.xml

This file is used to specify which 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 in this file.

```
<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

The following command is used to start dfs. Executing this command will 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:

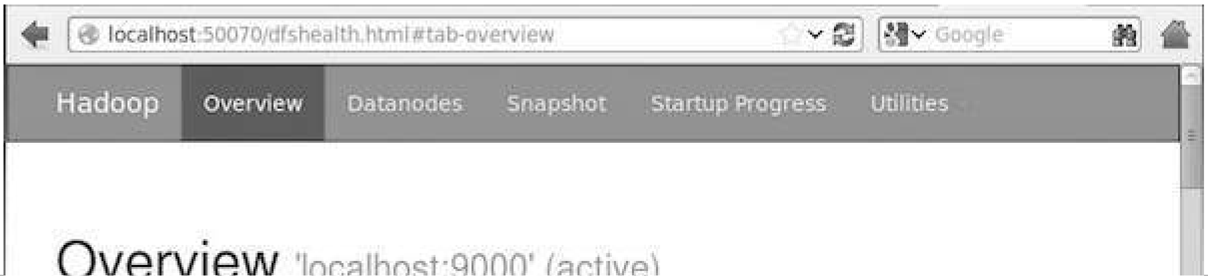
```
starting yarn daemons
starting resourcemanager, logging to /home/hadoop/hadoop-
2.4.1/logs/yarn-hadoop-resourcemanager-localhost.out
localhost: starting node manager, logging to /home/hadoop/hadoop-
2.4.1/logs/yarn-hadoop-nodemanager-localhost.out
```

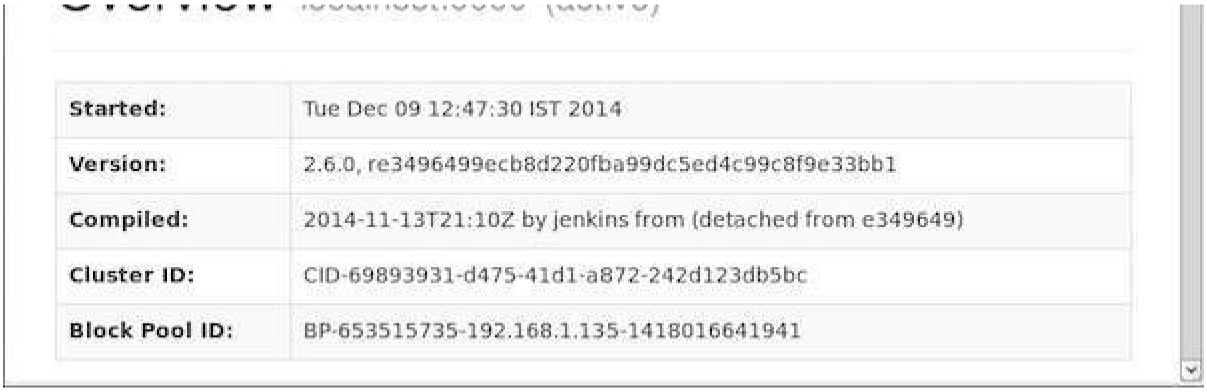
## Step 4: Accessing Hadoop on Browser

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 image depicts a Hadoop browser.


localhost:50070/dfshealth.html#tab-overview
Google
Hadoop
Overview
Datanodes
Snapshot
Startup Progress
Utilities
Overview 'localhost:9000' (active)



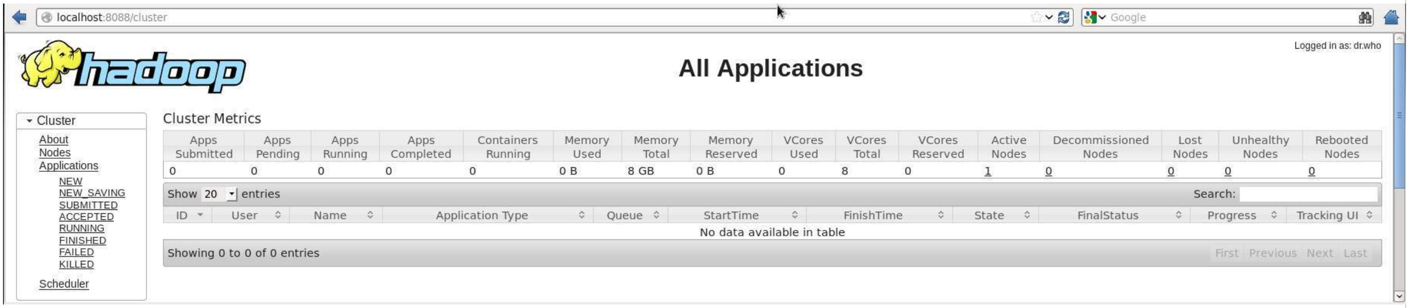
| | |
|---|---|
| **Started:** | Tue Dec 09 12:47:30 IST 2014 |
| **Version:** | 2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1 |
| **Compiled:** | 2014-11-13T21:10Z by jenkins from (detached from e349649) |
| **Cluster ID:** | CID-69893931-d475-41d1-a872-242d123db5bc |
| **Block Pool ID:** | BP-653515735-192.168.1.135-1418016641941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

## 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/
```

The following image depicts the Hadoop cluster browser.



All Applications

Cluster Metrics

| Apps Submitted | Apps Pending | Apps Running | Apps Completed | Containers Running | Memory Used | Memory Total | Memory Reserved | VCores Used | VCores Total | VCores Reserved | Active Nodes | Decommissioned Nodes | Lost Nodes | Unhealthy Nodes | Rebooted Nodes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 B | 8 GB | 0 B | 0 | 8 | 0 | 1 | 0 | 0 | 0 | 0 |

Show 20 entries Search:

| ID | User | Name | Application Type | Queue | StartTime | FinishTime | State | FinalStatus | Progress | Tracking UI |
|---|---|---|---|---|---|---|---|---|---|---|
| No data available in table | | | | | | | | | | |

Showing 0 to 0 of 0 entries

## Step 3: Downloading Sqoop

We can download the latest version of Sqoop from the following link For this tutorial, we are using version 1.4.5, that is, **sqoop-1.4.5.bin__hadoop-2.0.4-alpha.tar.gz**.

## Step 4: Installing Sqoop

The following commands are used to extract the Sqoop tar ball and move it to “/usr/lib/sqoop” directory.

```
$tar -xvf sqoop-1.4.4.bin__hadoop-2.0.4-alpha.tar.gz
$ su
password:

# mv sqoop-1.4.4.bin__hadoop-2.0.4-alpha /usr/lib/sqoop
#exit
```

## Step 5: Configuring bashrc

You have to set up the Sqoop environment by appending the following lines to ~/**.bashrc** file:

```
#Sqoop
export SQOOP_HOME=/usr/lib/sqoop export PATH=$PATH:$SQOOP_HOME/bin
```

The following command is used to execute ~/**.bashrc** file.

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

## Step 6: Configuring Sqoop

To configure Sqoop with Hadoop, you need to edit the **sqoop-env.sh** file, which is placed in the **$SQOOP_HOME/conf** directory. First of all, Redirect to Sqoop config directory and copy the template file using the following command:

```
$ cd $SQOOP_HOME/conf
$ mv sqoop-env-template.sh sqoop-env.sh
```

Open **sqoop-env.sh** and edit the following lines:

```
export HADOOP_COMMON_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=/usr/local/hadoop
```

## Step 7: Download and Configure mysql-connector-java

We can download **mysql-connector-java-5.1.30.tar.gz** file from the following link.

The following commands are used to extract mysql-connector-java tarball and move **mysql-connector-java-5.1.30-bin.jar** to /usr/lib/sqoop/lib directory.

```
$ tar -zxf mysql-connector-java-5.1.30.tar.gz
$ su
password:

# cd mysql-connector-java-5.1.30
# mv mysql-connector-java-5.1.30-bin.jar /usr/lib/sqoop/lib
```

## Step 8: Verifying Sqoop

The following command is used to verify the Sqoop version.

```
$ cd $SQOOP_HOME/bin
$ sqoop-version
```

Expected output:

```
14/12/17 14:52:32 INFO sqoop.Sqoop: Running Sqoop version: 1.4.5
Sqoop 1.4.5 git commit id 5b34accaca7de251fc91161733f906af2eddbe83
Compiled by abe on Fri Aug 1 11:19:26 PDT 2014
```

Sqoop installation is complete.