

Hadoop 2.5.2 Installation Steps:

1. Install recent Java JDK. Installed Oracle JDK 8.
2. Download the hadoop 2.5.2 tar.gz file from one of the mirrors found [here](#)
3. Unpacking the tar.gz file:
 - a. You can untar the hadoop package and store it in any directory needed. The hadoop has lots of service which runs as daemon process and invoke some process as required. Hence, we can store the untar the hadoop package in /srv folder.

```
~/Downloads$ tar -xzf hadoop-2.5.2.tar.gz //untar the downloaded file
~/Downloads$ sudo mv hadoop-2.5.2 /srv/ // move the hadoop folder to /srv/
~/Downloads$ sudo chown -R hadoop:hadoop /srv/hadoop-2.5. //assign ownership to hadoop user-
group:user
~/Downloads$ sudo chmod g+w -R /srv/hadoop-2.5.2 // allow other group users to
write
```

4. Before configuring the hadoop, create a new user and user-group for hadoop. Use following commands:

```
~$ sudo addgroup hadoop
~$ sudo useradd -m -g hadoop hadoop
~$ sudo usermod -a -G hadoop root
```

- a. The first command creates a usergroup hadoop.
- b. The next command creates a user named hadoop in group hadoop. The -m flag creates a home directory for hadoop user.
- c. The third command adds root to the hadoop user group.

It is a good practice to keep the hadoop user separate so as to limit the hadoop user only with limited access to the file system of the operating system. This will give us a better control over the hadoop's spread out.

5. Next step is configure SSH for hadoop user.
 - a. Check if SSH server is installed/running.

```
~$ sudo service ssh status
```

- b. If running we can proceed further. Now switch to the hadoop user.

```
~$ sudo su hadoop
~$ ssh-keygen
```

ssh-keygen will lead to generate ssh keys and will prompt to enter passphrase. Just press Enter whenever prompted. Since, all the communication between different nodes takes place through ssh it is feasible to keep ssh passphraseless.

To allow ssh into the host, we need to copy the new generated keys into the authorized_key file. Use the following command:

```
~$ cat /home/hadoop/.ssh/id_rsa.pub >> /home/hadoop/.ssh/authorized_keys
~$ chmod 600 /home/hadoop/.ssh/authorized_keys
```

To test the ssh setup, check with following command:

```
~$ ssh -l hadoop localhost
```

This will cause entry into the ssh shell. Type exit to return to hadoop user shell; exit again to return to root shell.

6. Disabling IPv6

The hadoop installations so far have some conflicts with IPv6, hence it is recommended to disable it. Also, the single-node/pseudo distributed configuration doesn't require the IPv6, so we can disable it.

Use the following commands:

```
~$ sudo sysctl net.ipv6.conf.all.disable_ipv6=1
~$ sudo sysctl net.ipv6.conf.default.disable_ipv6=1
~$ sudo sysctl net.ipv6.conf.lo.disable_ipv6=1
```

Else, you can make changes in the file /etc/sysctl.conf

7. Configuring Hadoop

Finally, we are all set now to setup the hadoop configuration.

Let's set Environment variables which will be required quite often:

a. Switch to hadoop user,

```
~$ sudo su hadoop
```

Open /home/hadoop/.bashrc in your favorite editor and append following lines to the end:

```
#Set Hadoop environment variables
export HADOOP_HOME=/srv/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
#Set JAVA Home
export JAVA_HOME=/usr/lib/jvm/java-{version-info}
```

b. Switch to root user

Make changes in ~/.profile file

Append following contents in the .profile file.

```
#Set the Hadoop Related Environment variables
export HADOOP_HOME=/srv/hadoop
export HADOOP_STREAMING=$HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-2.5.2.jar
export PATH=$PATH:$HADOOP_HOME/bin
```

```
#Set the JAVA_HOME
export JAVA_HOME=/usr/lib/jvm/java-{version-info}
```

c. Test your environment:

Execute following command:

```
~$ hadoop version
Hadoop 2.5.2
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r
cc72e9b000545b86b75a61f4835eb86d57bfafc0
Compiled by jenkins on 2014-11-14T23:45Z
Compiled with protoc 2.5.0
From source with checksum df7537a4faa4658983d397abf4514320
This command was run using /srv/hadoop-2.5.2/share/hadoop/common/hadoop-common-
2.5.2.jar
```

The output will be similar to above. If this is not the output, probably some thing is wrong with the configuration.

8. Change in configuration files:

a. Edit in \$HADOOP_HOME/etc/hadoop/hadoop-env.sh

```
# The java implementation to use.
Export JAVA_HOME=/usr/lib/jvm/java-{version-info}
```

b. Edit in \$HADOOP_HOME/etc/hadoop/core-site.xml

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
<!-- Default path location of temp files -->
<property>
```

```
<name>hadoop.tmp.dir</name>
<value>/var/app/hadoop/data</value>
</property>
</configuration>
```

c. Edit in \$HADOOP_HOME/etc/hadoop/mapred-site.xml

First you will have to copy the default template mapred-site.xml.template to mapred-site.xml and perform following edit:

```
<configuration>
<!-- map reduce framework that overrides default map reduce framework -->
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

d. Edit in \$HADOOP_HOME/etc/hadoop/hdfs-site.xml

This file is used to point to local file system where the hdfs will store the data and metadata. However, here the main property we will be describing is for replication factor, since we are using a single-node configuration we are setting this value to 1.

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