10. Deployment and Maintenance

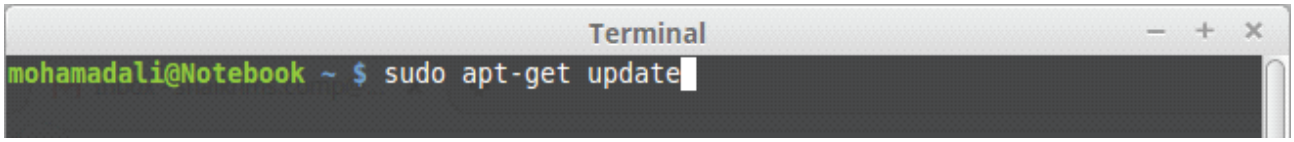
10.1 Installation:

10.1.1 Installing Hadoop 2.6.0 or Hadoop 2.x.x

This method of install Hadoop is to install any version of Hadoop 2.x.x
As we know that Hadoop requires JVM to run. So we need to install Java before installing Hadoop.So before installing java let us update our package list doing this will automatically give latest version of Java form the Linux vender.

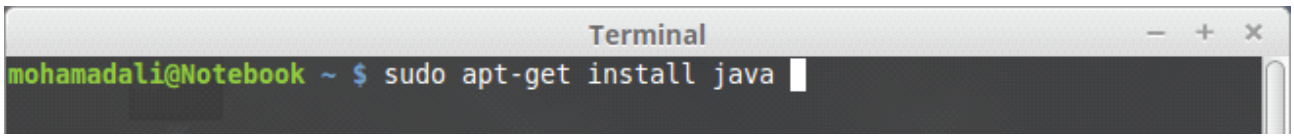
To update Package list type this command in your terminal.

$ sudo apt-get update



doing this will require internet connection. Once you complete the above step you can install java by typing below command in terminal.(note- you can use any other command to install java)

$ sudo apt-get install default-jdk



when you are done to check which java version to do that type

$ java -version

```
Terminal
mohamadali@Notebook ~ $ java -version
java version "1.7.0_65"
OpenJDK Runtime Environment (IcedTea 2.5.3) (7u71-2.5.3-0ubuntu0.14.04.1)
OpenJDK 64-Bit Server VM (build 24.65-b04, mixed mode)
mohamadali@Notebook ~ $
```

above result describes that our installed version in 1.7.65 make sure that you have java 1.6 or above.

Now we require to install ssh

ssh is Secure shell. This application allows us to get remote access of any machine(or Local host) by different password other then root and also allows us to bypass the password by setting it to empty. To install ssh use following command

$ sudo apt-get install ssh

```
Terminal
mohamadali@Notebook ~ $ sudo apt-get install ssh
```

if we try to connect local host or local machine though ssh it will ask user password. To check this you can type this command in terminal.

$ ssh localhost

you will get output as

```
Terminal
mohamadali@Notebook ~ $ ssh localhost
mohamadali@localhost's password:
Welcome to Linux Mint 17.1 Rebecca (GNU/Linux 3.13.0-37-generic x86_64)

Welcome to Linux Mint
 * Documentation:  http://www.linuxmint.com
Last login: Wed Jan 28 18:48:29 2015 from localhost
mohamadali@Notebook ~ $
```

Note-Before going further we need to exit ssh just type “exit” in same terminal.

so we need to set our ssh for password less communication. To do that execute following command in terminal.

$ ssh-keygen -t rsa -P ''

```
Terminal
mohamadali@Notebook ~ $ ssh-keygen -t rsa -P ''
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mohamadali/.ssh/id_rsa):
Your identification has been saved in /home/mohamadali/.ssh/id_rsa.
Your public key has been saved in /home/mohamadali/.ssh/id_rsa.pub.
The key fingerprint is:
e7:0c:b3:2b:2b:27:bb:e3:ef:5c:12:bf:c4:dc:6e:3f mohamadali@Notebook
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|                 |
|      . S .      |
|       = 0       |
|      . B +      |
|    +.o+ +. E    |
|   .+X=oo.....   |
+-----------------+
mohamadali@Notebook ~ $
```

Please note that there is two single quotes after 'P' in command without space. After entering this command it will ask “Enter file in which to save the key (/home/mohamadali/.ssh/id_rsa):” press Enter without typing any single word. You will get Image after entering this doing this, this image is called as randomart image. This image will vary machine to machine and this key will be used to communicate between any two machine for authentication. This command will create an RSA key pair with an empty password. Generally, using an empty password is not recommended, but in this case it is needed to unlock the key without your interaction (you don’t want to enter the passphrase every time Hadoop interacts with its nodes).

Now we need to save this generated key to local machine’s host key fingerprint to the user’s known hosts file. To do this use this command.

$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

```
Terminal
mohamadali@Notebook ~ $  cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
mohamadali@Notebook ~ $
```

To check we have bypass the password we need to again execute

$ssh localhost

```
Terminal
mohamadali@Notebook ~ $ ssh localhost
Welcome to Linux Mint 17.1 Rebecca (GNU/Linux 3.13.0-37-generic x86_64)

Welcome to Linux Mint
 * Documentation:  http://www.linuxmint.com
Last login: Thu Jan 29 22:32:45 2015 from localhost
mohamadali@Notebook ~ $
```

if this step asks you for a password that means you have done something wrong. So to repair this you need to repeat the above steps from next to installing ssh again.

Note-Before going further we need to exit ssh just type “exit” in same terminal.

Once we have completed this we will need to download Hadoop 2.6 or any version from its official site http://hadoop.apache.org/

then extract this Hadoop tar.gz manually or through terminal. Now we need to move Hadoop folder to root this step is optional but its recommend that you may move file to root. To move Hadoop folder to its appropriate location use following command (note this command is only use to move folder to root if you are placing to other location you can do it manually).

$sudo mv Desktop/hadoop-2.6.0 /usr/local/hadoop

```
Terminal
mohamadali@Notebook ~ $ sudo mv Desktop/hadoop-2.6.0 /usr/local/hadoop
[sudo] password for mohamadali:
mohamadali@Notebook ~ $
```

Explanation of this command

sudo : This is keyword which allows user to grant super user permission temporary. This command is Linux native command. It means “super user do”.

mv : This is Linux native command to move any file or directory to any location. it has two parameters as

parameter 1: source Address
parameter 2:destination Address
Note both above address should be complete qualified address

In above command my source address is “Desktop/hadoop-2.6.0” you can change this according to your source location and my destination address is “/usr/local/hadoop”

note- I haven't given '/' after my destination this means that I am renaming my source Folder name from “hadoop-2.6.0” to “hadoop”.

Now we need to set system environment variable so that our system identifies Hadoop. To do this open bashrc file as a root in any text editor.(in my case I am using gedit).

```
$sudo gedit ~/.bashrc
```

Note – some time you get blank file please make sure that this file is ~/.bashrc
Append below content to this file.

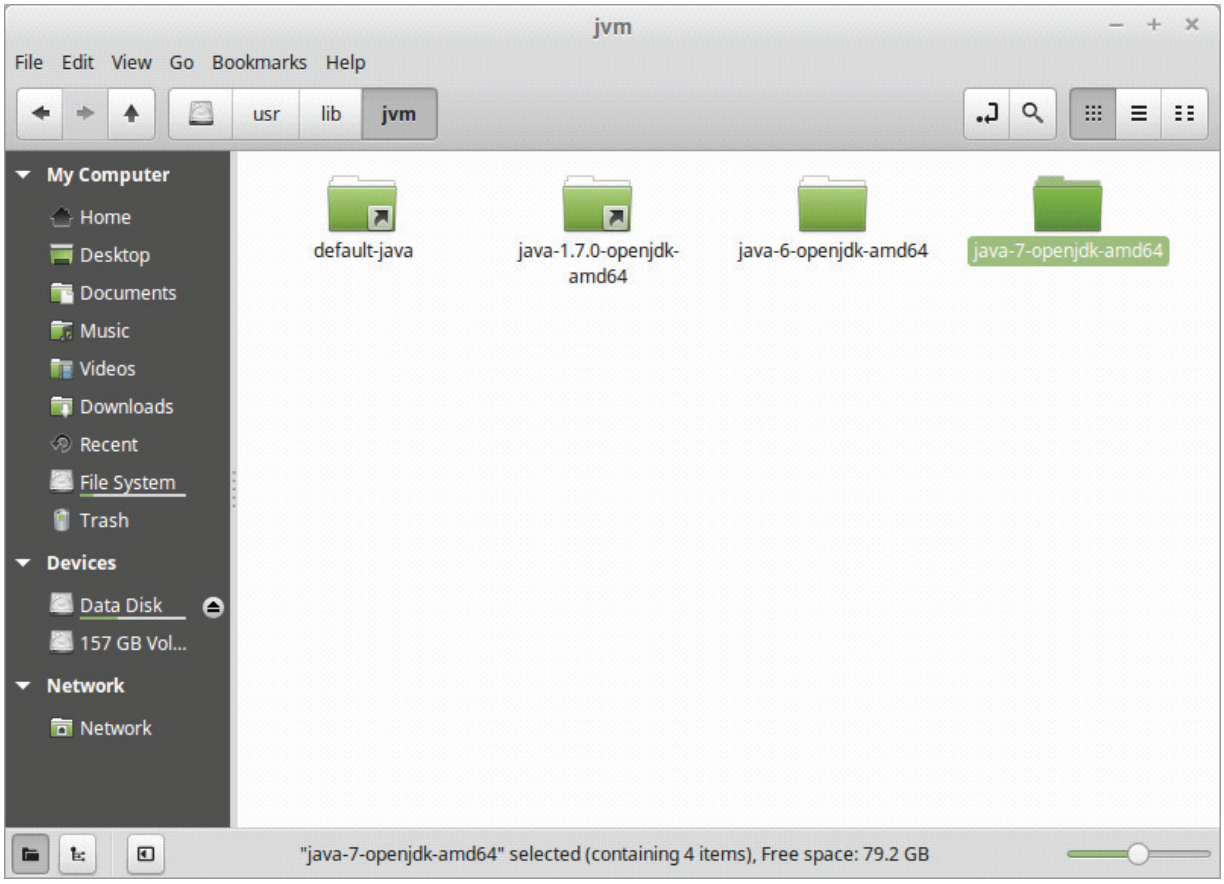
```
#Hadoop variables
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
#end of Hadoop variable declaration
```

Explanation of Above code

Line 1: export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64

We are setting Java installation path so that Hadoop can use this path where ever required.

To get your installation path go to /usr/lib search for jvm open folder thee you will get many folders open one without arrow on it (Arrow marked folders are Symbolic links in Linux, similar to shortcuts in windows). That's your installed java.



Line 2:export HADOOP_INSTALL=/usr/local/hadoop

this line is to identify installed location of Java in the system. Note if you have kept this folder in some other location you need to change path accordingly.

Line3 to 8:

these are Hadoop components locations, We are defining these to reduce or work later, I will explain the use of these lines later in depth.

Save and close this ~/.bashrc.

As we have add successfully added the environment variable we need to reflect these to our system for this you can do two things

1.Close all terminals and reopen them as needed.

2.use following command

$source ~/.bashrc

once you have done this type this command to check we have installed our Hadoopproperly or not you can use this command.

$hadoop version

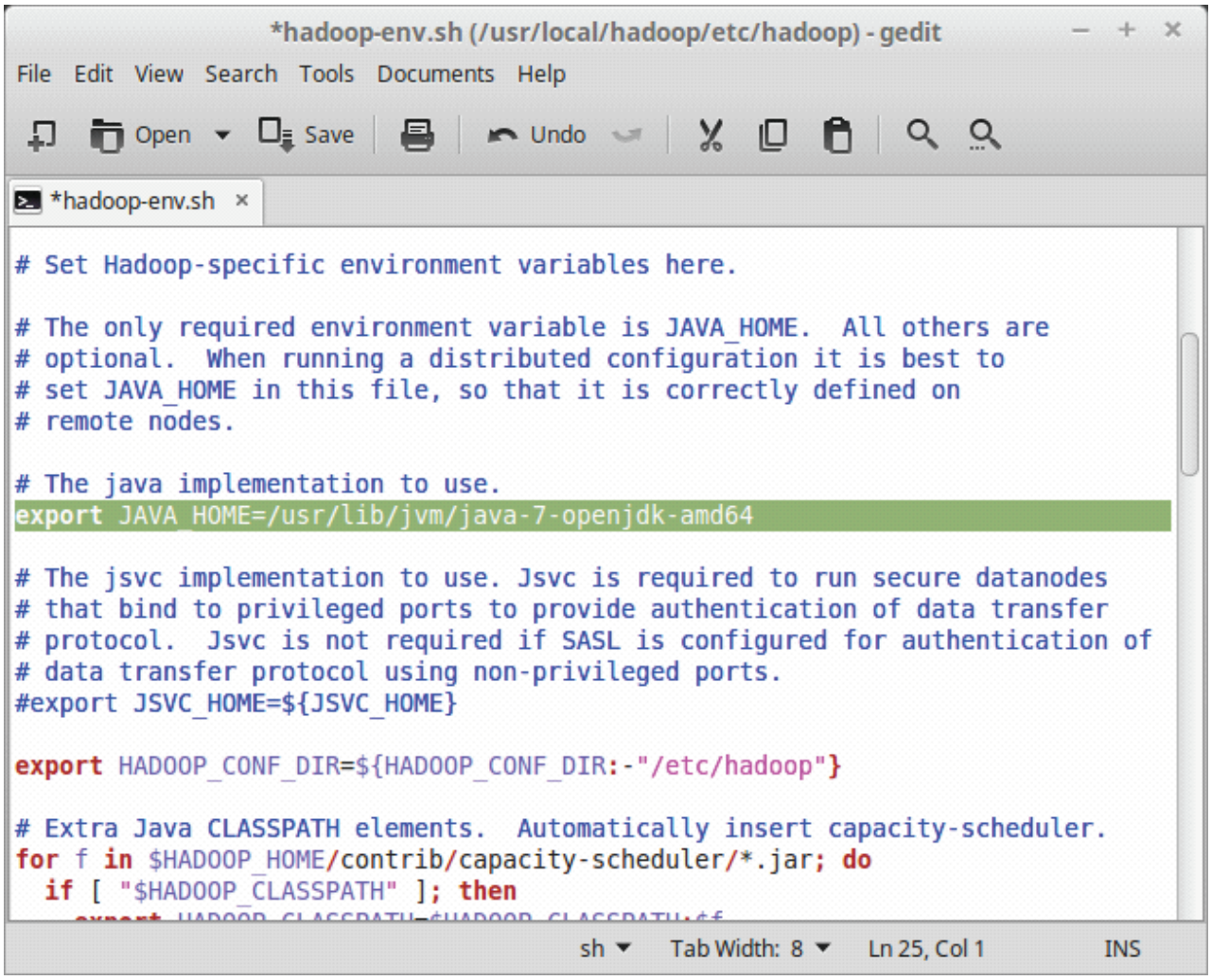
```
Terminal
mohamadali@Notebook ~ $ hadoop version
Hadoop 2.6.0
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r e3496499ecb8d220fba99dc5ed4c99c8f9e33bb1
Compiled by jenkins on 2014-11-13T21:10Z
Compiled with protoc 2.5.0
From source with checksum 18e43357c8f927c0695f1e9522859d6a
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-2.6.0.jar
mohamadali@Notebook ~ $
```

if you get something like this it means you have successful set up Hadoop in your system.

Now the last thing we need to update JAVA_HOME in Hadoop so open "/hadoop/etc/hadoop/hadoop-env.sh" from you installed Hadoop path and find these line in it

```
hadoop-env.sh (/usr/local/hadoop/etc/hadoop) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
hadoop-env.sh
# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME.  All others are
# optional.  When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use.
export JAVA_HOME=${JAVA_HOME}

# The jsvc implementation to use. Jsvc is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol.  Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
#export JSVC_HOME=${JSVC_HOME}

export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}

# Extra Java CLASSPATH elements.  Automatically insert capacity-scheduler.
for f in $HADOOP_HOME/contrib/capacity-scheduler/*.jar; do
  if [ "$HADOOP_CLASSPATH" ]; then
sh  Tab Width: 8  Ln 25, Col 1  INS
```

replace this line with your installed java path



save it and exit.

In this way we have installed Hadoop 2.6.0 in our Linux.

Note- above all steps are exactly same for installing all 2.x.x versions of Hadoop.

Now Hadoop can be use in three different ways

1) Stand Alone Mode

This mode generally does not requires any configuration to be done.

This mode is usually used for Debugging purpose.

All default configuration of Hadoop are done in this mode.

2) Pseudo Distributed Mode (will be Explained in depth Later)

This mode is also called single node mode.

This mode needs little configuration.

This mode is used for Development purpose

3)Distributed Mode (will be Explained in depth Later)

This mode is also called as Multinode node.

This mode needs some changes to be done in Psedudistrbuted mode along with ssh

This mode is generally use for commercial purpose.

10.1.2 Configuring Hadoop 2.6.0 Single Mode/Pseudo Distributed Mode in Linux

Hadoop is by default is configured in Standalone mode. This stand alone mode is used only for debugging purpose but to develops any application we need to configure hadoop in Pseudo Distributed mode.

To configure hadoop in Pseudo Distributed mode we need to edit following files

1)core-site.xml
2)hdfs-site.xml
3)mapred-site.xml
4)yarn-site.xml

Please note that we need to carry out the steps as explained in Previous Document of Setting up hadoop 2.6.0 on Linux.

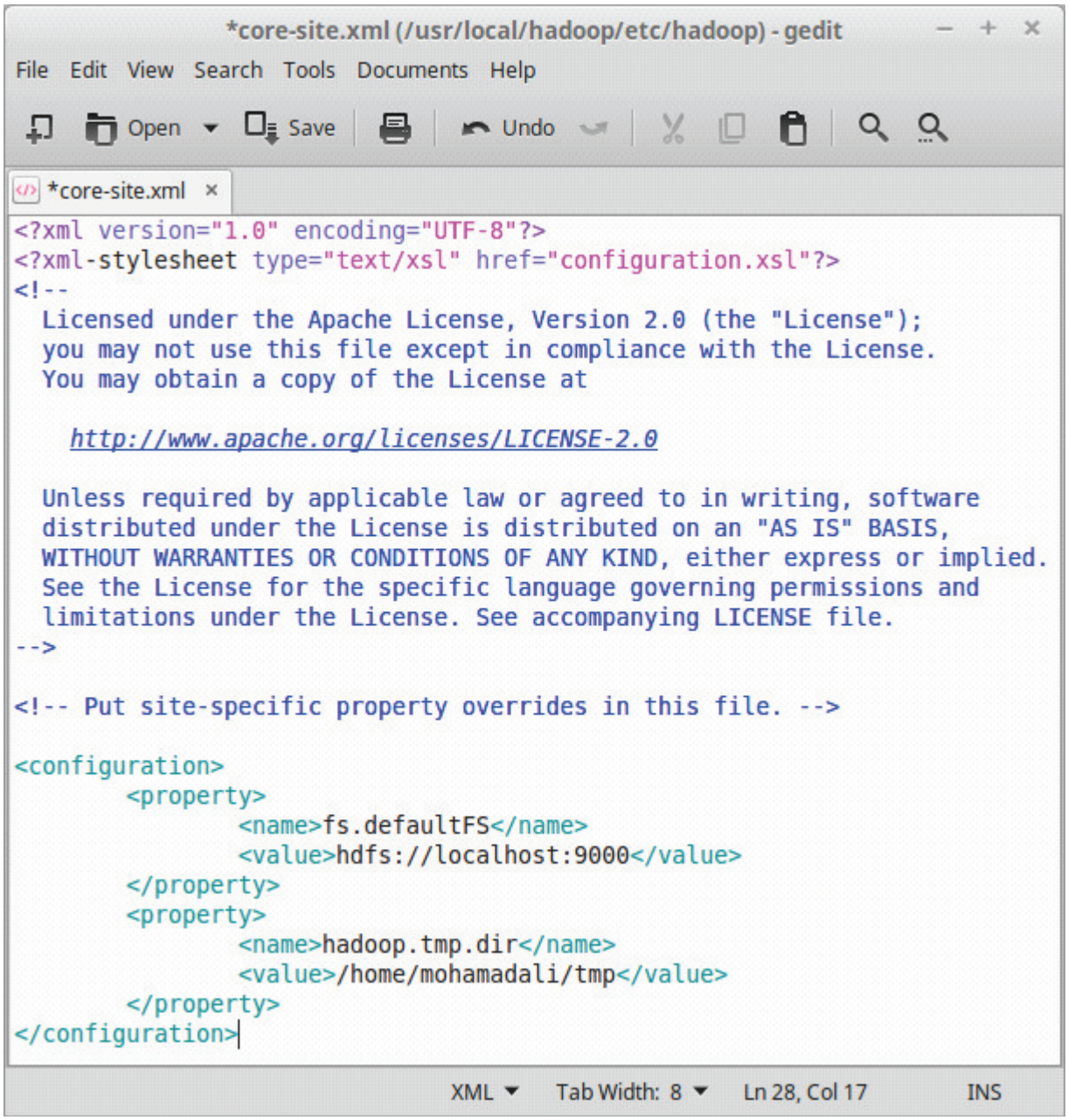
All mentioned files are present in hadoop installation directory under "/etc/hadoop" in my case as per previous document its address is "/usr/local/hadoop/etc/hadoop"

1) configuring core-site.xml
core site xml is a file containing all core property of hadoop. For example. Namenode url, Temporary storage directory path, etc. Hadoop has predefined configuration which we need to override them if we mention any of the configuration in core-site.xml then during startup of hadoop, hadoop will read these configuration an run hadoop using this. To get more details of default configuration in hadoop you can visit **https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/core-default.xml**

so let us configure some of our requirement.
Open this file in any of the text editor and add these contents in it between <configurations></configurations>

<property>

```
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
</property>
<property>
        <name>hadoop.tmp.dir</name>
        <value>/home/mohamadali/tmp</value>
</property>
```



Explanation of the above code

property 1: fs.defaultFS

This property overrides the default namenode url its syntax is hdfs://<ip-address of namenode>:<port number> .This property was named as fs.default.name in hadoop 1.x.x version. Note: Port number can be any number above 255 to 65536

property 2: hadoop.tmp.dir

This property is used to change the temporary storage directory during execution of any algorithm in hadoop by default its location is "/tmp/hadoop-${user.name}" in my case I have created this directory in my home folder name tmp so its "/home/mohamadali/tmp".

2) Configuring hdfs-site.xml

This file contains all configuration about hadoop distributed file system also called as HDFS such as storage location for namenode, storage location for datanode, replication factor of HDFS, etc.

Similar to core-site.xml we need to place below content between configuration fields to get more information on this you can visit above mentioned link.

```
<property>
        <name>dfs.replication</name>
        <value>1</value>
</property>
<property>
        <name>dfs.namenode.name.dir</name>
        <value>/home/mohamadali/tmp/namenode</value>
</property>
<property>
        <name>dfs.datanode.data.dir</name>
        <value>/home/mohamadali/tmp/datanode</value>
</property>
```

hdfs-site.xml (/usr/local/hadoop/etc/hadoop) - gedit

File Edit View Search Tools Documents Help

Open Save Undo

hdfs-site.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
        <property>
                <name>dfs.replication</name>
                <value>1</value>
         </property>
         <property>
                <name>dfs.namenode.name.dir</name>
                <value>/home/mohamadali/tmp/namenode</value>
         </property>
         <property>
                <name>dfs.datanode.data.dir</name>
                <value>/home/mohamadali/tmp/datanode</value>
         </property>
</configuration>
```

XML Tab Width: 8 Ln 31, Col 21 INS

Explanation of above properties in detail.

Property 1: dfs.replication

This property overrides the replication factor in hadoop. By default its value is 3 but in single node cluster it is recommended to be 1.

Property 2: dfs.namenode.name.dir

Central repository using dynamic data node allocation in multi node HDFS

This property overrides storage location of namenode data by default its storage location is inside "/tmp/hadoop-${user.name}". To change this you have set value of your folder location in my case it is inside tmp directory created during core-site.xml

Property 3: dfs.datanode.data.dir

This property overrides storage location of datanode data by default its storage location is inside "/tmp/hadoop-${user.name}". To change this you have set value of your folder location in my case it is also inside tmp directory created during core-site.xml

Note: for property 1 and property 2

Please make sure if your location of both datanode and namenode is in your root directory then you should change its ownership and read write access using chown and chmode command in Linux. Also you can create these directory manually before this setting them to your path else hadoop will create them for you.

3) Configuring mapred-site.xml

This file contain all configuration about Map Reduce component in hadoop. Please note that this file doesn't exist but you can copy or rename it from mapred-site.xml.template. Configuration for this file is should be as followed.

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

Explanation of above property

As we know that from hadoop 2.x.x hadoop has introduced new layer of technology developed by hadoop to improve performance of map reduce algorithm this layer is called as "yarn" that is Yet Another Resource Negotiator. So here we are configuring that our hadoop framework is yarn if

we don't specify this property then our hadoop will use Map reduce 1 also called as MR1.

4) Configuring yarn-site.xml

This file contains all information about YARN as we will be using MR2 we need to specify the auxiliary services that need to be used with MR2 so add these lines to yarn-site.xml

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

Now we have successfully configured hadoop 2.6.0 or say hadoop 2.x.x in Pseudo distributed mode.
Before starting hadoop we need to format our namenode. Execute this command to format namenode.

```
$hdfs  namenode -format
```

Now to start hadoop you can use two command

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

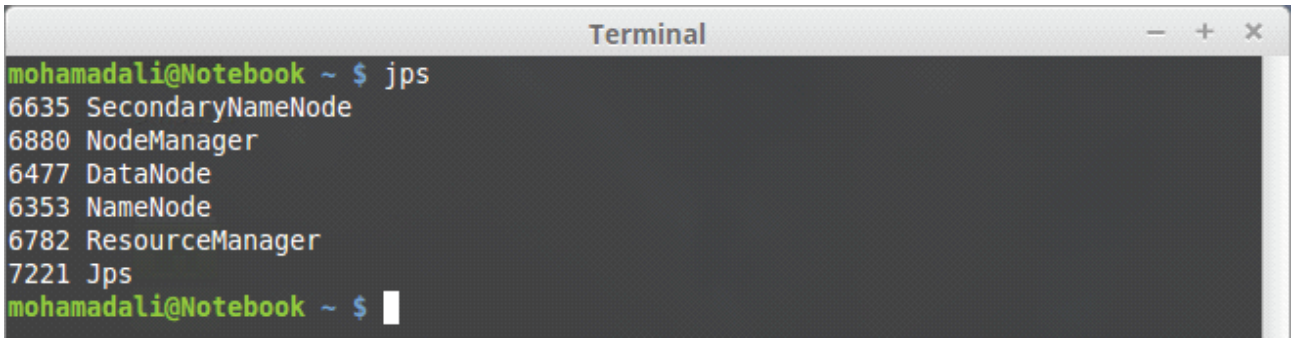
or you can also use deprecated command as

```
$ start-all.sh
```

To check the which components are working you can use bellow command

```
$ jps
```

you will get output as



Please make note that number preceding are port number they many vary with machine to machine every time you start. If any one component is missing that means you have incorrectly configured above xml files.

10.1.3 Configuring Hadoop 2.6.0 Distributed Mode / Multinode Mode in Linux

Please note that this document is continuation of my pervious document, in this document we will be configuring hadoop to run on Multi Node also called as Distributed mode.

Hadoop is by default is configured in Standalone mode. This standalone mode is used only for debugging purpose but to develop any distributed application we need to configure hadoop in Distributed mode.

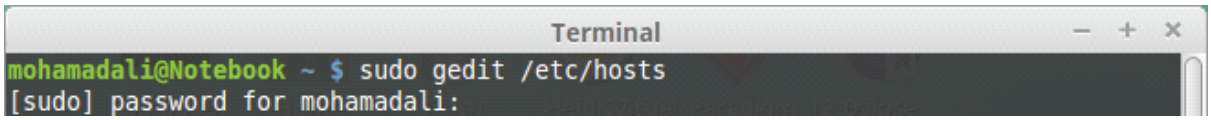
Please refer to first two document of the installing of single node and configuration of hadoop.
To configure hadoop in Pseudo Distributed mode we need to edit following files

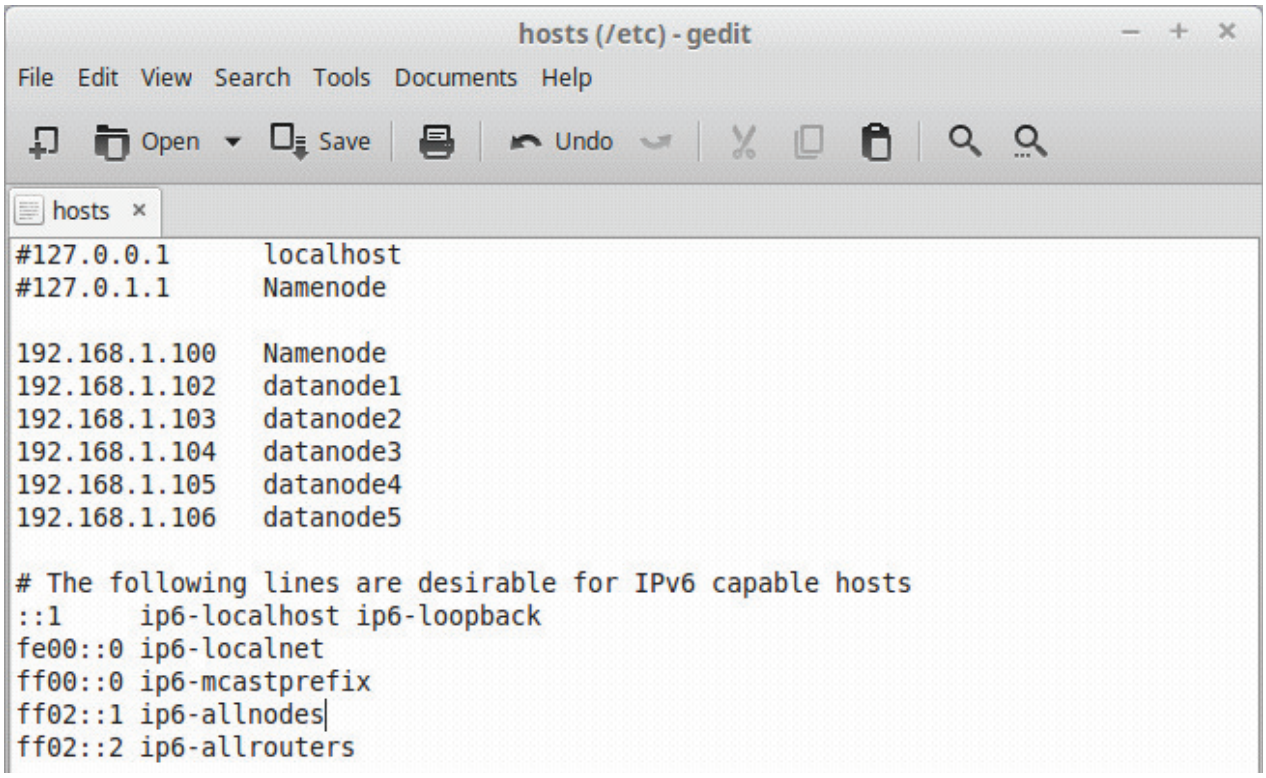
1) Add all Slaves/ Other Machines (Datanode, Job tracker, node Manager) to /etc/hosts.
2) Copy SSH key from Master (Namenode) to Slave file.
3) Edit core-site.xml
4) Edit hdfs-site.xml
5) Edit yarn-site.xml
6) Format Cluster
7) Start Cluster

1) Add all Slaves to hosts file

Adding Other Machines (Slaves to /etc/hosts) to do that open host file on Namenode by below command and list down all slave machine to it.
Please note that one machine in single line and also don't forget to comment out loop back address.

$sudo gedit /etc/hosts





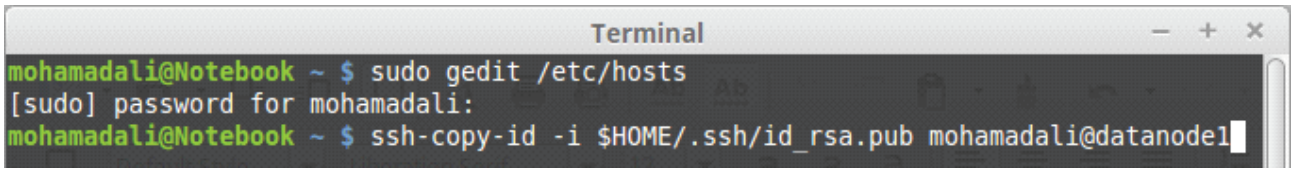
2) **Copy ssh keys to all slave files (Namenode only)**

Note: we have already generated ssh-key during installing of pseudo distributed mode. If not we can use same command to generate keys.

This will allow our namenode to communicate our Datanode using password less communication. Command for it is

```
$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub user-name@pc-name
```

Please note that user name is slave user account on which you want to run hadoop and pc-name is name of slave you have entered in step 1 in /etc/hosts file. Do same process for each Datanode.



Once you are complete with this step you can check for password less communication using

```
$ ssh user-name@pc-name
```

```
Terminal
mohamadali@Notebook ~ $ ssh localhost
Welcome to Linux Mint 17.1 Rebecca (GNU/Linux 3.13.0-37-generic x86_64)

Welcome to Linux Mint
 * Documentation:  http://www.linuxmint.com
Last login: Thu Jan 29 22:32:45 2015 from localhost
mohamadali@Notebook ~ $
```

Please note that we need to carry out the steps as explained in Previous Document of Setting up hadoop 2.6.0 on Linux.

All mentioned files are present in hadoop installation directory under "/etc/hadoop" in my case as per previous document its address is "/usr/local/hadoop/etc/hadoop"

3) Edit core-site.xml

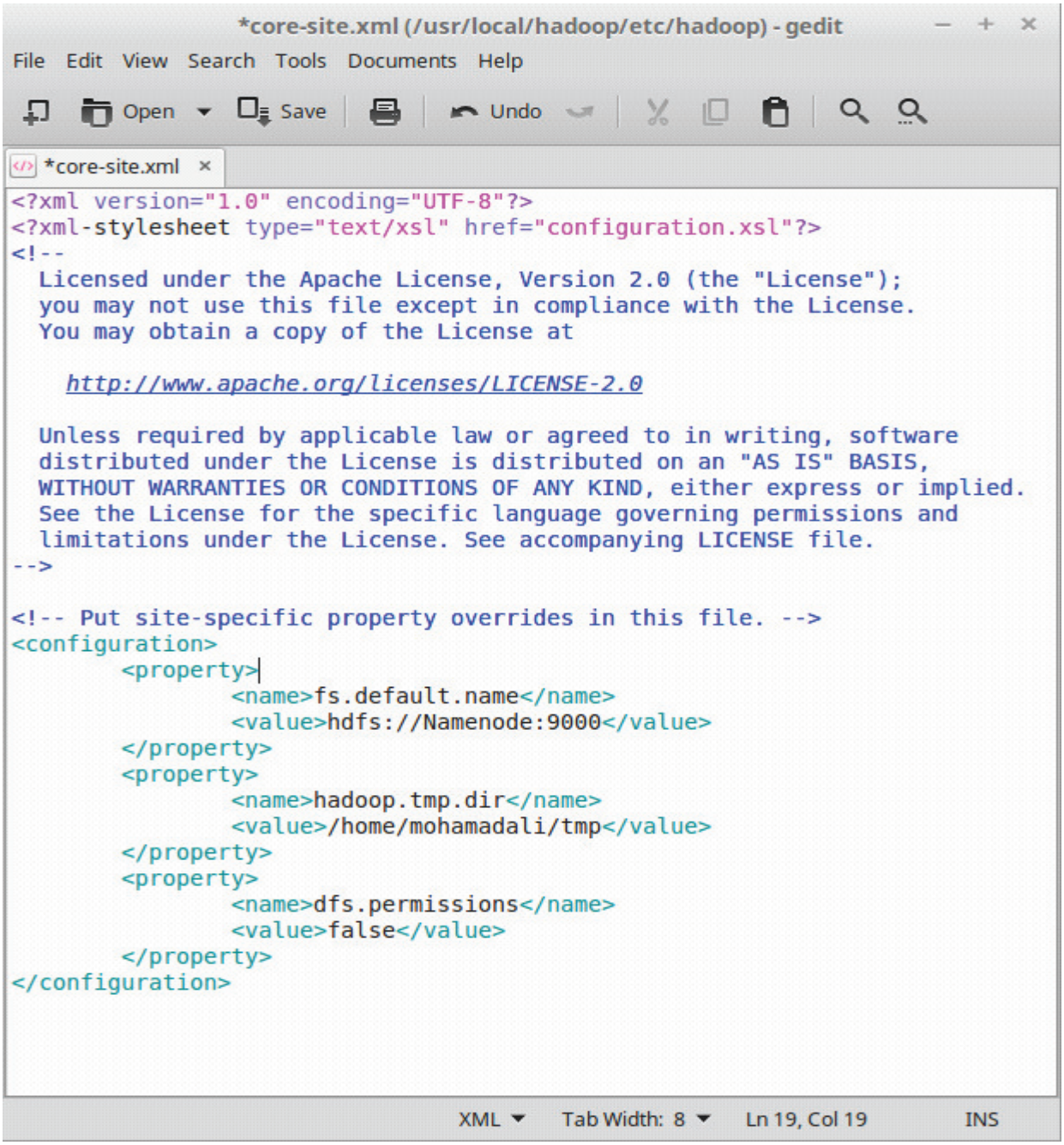
Core site xml is a file containing all core property of hadoop. For example. Namenode url, Temporary storage directory path, etc. Hadoop has predefined configuration which we need to override them if we mention any of the configuration in core-site.xml then during start-up of hadoop, hadoop will read these configuration a run hadoop using this. To get more detailsof default configuration in hadoop you can visit

https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/core-default.xml

So let us configure some of our requirement.
Open this file in any of the text editor and add these contents in it between
(For Both Namenode & Datanode)
<configurations></configurations>

```
<property>
        <name>fs.default.name</name>
        <value>hdfs://Namenode:9000</value>
</property>
<property>
        <name>hadoop.tmp.dir</name>
        <value>/home/mohamadali/tmp</value>
```

```
</property>
<property>
        <name>dfs.permissions</name>
        <value>false</value>
</property>
```



Explanation of the above code

Property 1: fs.default.name

This property overrides the default Namenode url its syntax is hdfs://<ip-address of Namenode>:<port number> .This property is named as fs.defaultFS in hadoop new version. Note: Port number can be any number above 255 to 65536

Property 2: hadoop.tmp.dir

This property is used to change the temporary storage directory during execution of any algorithm in hadoop by default its location is "/tmp/hadoop-${user.name}" in my case I have created this directory in my home folder name tmp so its "/home/mohamadali/tmp".

Property 3: dfs.permissions

This property is used to change the privileges of HDFS so that other machines can use it. It can be done by setting the permissions off. To achieve it we will make the value of this property False.

<value>false</value>.

4) Edit hdfs-site.xml

This file contains all configuration about hadoop distributed file system also called as HDFS such as storage location for Namenode, storage location for Datanode, replication factor of HDFS, etc.

Similar to core-site.xml we need to place below content between configuration fields to get more information on this you can visit above mentioned link.

(For Both Namenode & Datanode)

```
<property>
        <name>dfs.replication</name>
        <value>2</value>
</property>
```

hdfs-site.xml (/usr/local/hadoop/etc/hadoop) - gedit

File Edit View Search Tools Documents Help

Open Save Undo

hdfs-site.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
        <property>
                <name>dfs.replication</name>
                <value>2</value>
        </property>
</configuration>
```

XML Tab Width: 8 Ln 20, Col 19 INS

Explanation of above properties in detail.

Property 1: dfs.replication

This property overrides the replication factor in hadoop. By default its value is 3 but in single node cluster it is recommended to be 1.

5) Edit yarn-site.xml

This file contains all information about YARN as we will be using MR2 we need to specify the auxiliary services that need to be used with MR2 so add these lines to yarn-site.xml

Please note you can change below file as per your system need. I have kept Namenode as master so I am using it as tasktracker.

(For Namenode/Master Node)

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

(For Slave/ Datanode)

```
<property>
        <name>
                yarn.nodemanager.aux-services.mapreduce.shuffle.class
        </name>
        <value>org.apache.hadoop.mapred.shuffleHandler</value>
</property>
<property>
        <name>yarn.resourcemanager.resource-tracker.address</name>
        <value>Namenode:9025</value>
</property>
<property>
        <name>yarn.resourcemanager.scheduler.address</name>
        <value>Namenode:9030</value>
</property>
<property>
        <name>yarn.resourcemanager.address</name>
        <value>Namenode:9040</value>
</property>
```

*yarn-site.xml (/usr/local/hadoop/etc/hadoop) - gedit

File Edit View Search Tools Documents Help

Open Save Undo

*yarn-site.xml

```xml
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>
        <property>
                <name>yarn.nodemanager.aux-services</name>
                <value>mapreduce_shuffle</value>
        </property>
        <property>
                <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
                <value>org.apache.hadoop.mapred.ShuffleHandler</value>
        </property>
        <property>
                <name>yarn.resourcemanager.resource-tracker.address</name>
                <value>Namenode:9025</value>
        </property>
        <property>
                <name>yarn.resourcemanager.scheduler.address</name>
                <value>Namenode:9030</value>
        </property>
        <property>
                <name>yarn.resourcemanager.address</name>
                <value>Namenode:9040</value>
        </property>
</configuration>
```

XML Tab Width: 8 Ln 34, Col 32 INS

Now we have successfully configured hadoop 2.6.0 or say hadoop 2.x.x in distributed mode.

Before starting hadoop we need to format our Namenode. Execute this command to format Namenode.

```
$hdfs namenode -format
```

Now to start hadoop you can use two command

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

$ start-yarn.sh

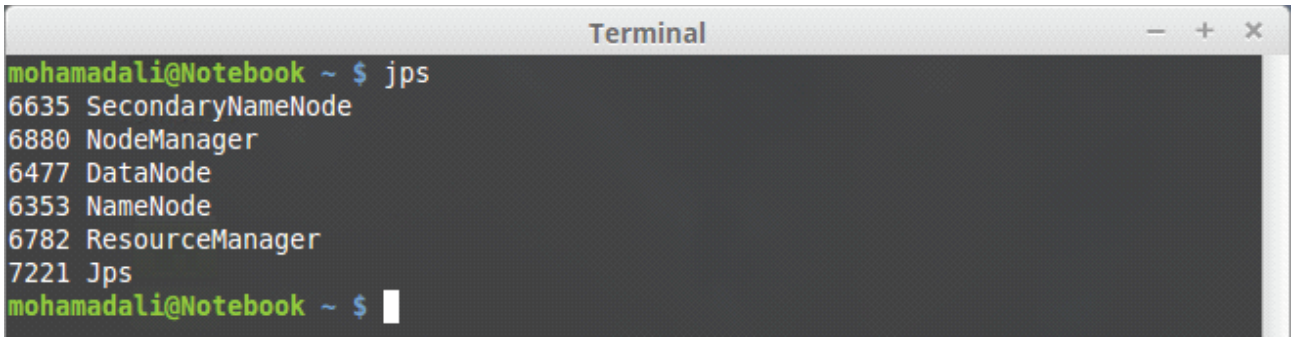
Or you can also use deprecated command as

$ start-all.sh

To check the which components are working you can use bellow command

$ jps

You will get output as



Please make note that number preceding are port number they many vary with machine to machine every time you start. If any one component is missing that means you have incorrectly configured above xml files.

Central repository using dynamic data node allocation in multi node HDFS

Now to check status of Hadoop through web UI you can visit https://namenode:50070/

Hadoop | Overview | Datanodes | Snapshot | Startup Progress | Utilities

Overview 'SIGComputer:9000' (active)

Started:	Sat Apr 25 13:19:40 IST 2015
Version:	2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1
Compiled:	2014-11-13T21:10Z by jenkins from (detached from e349649)
Cluster ID:	CID-84b85310-695f-426f-a2c6-18c741ba6a39
Block Pool ID:	BP-1775115477-172.16.2.86-1426314429609

Summary

Security is off.

Safemode is off.

194 files and directories, 118 blocks = 312 total filesystem object(s).

Heap Memory used 64.09 MB of 148.5 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 27.62 MB of 28.94 MB Commited Non Heap Memory. Max Non Heap Memory is 214 MB.

Configured Capacity:	783.21 GB
DFS Used:	4.2 GB
Non DFS Used:	59.01 GB
DFS Remaining:	720 GB
DFS Used%:	0.54%
DFS Remaining%:	91.93%
Block Pool Used:	4.2 GB
Block Pool Used%:	0.54%
DataNodes usages% (Min/Median/Max/stdDev):	0.32% / 0.64% / 0.64% / 0.15%
Live Nodes	3 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0
Block Deletion Start Time	25/4/2015 1:19:40 pm

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
mmcoe-desktop (172.16.2.101:50010)	2	In Service	262.32 GB	1.68 GB	18.25 GB	242.39 GB	118	1.68 GB (0.64%)	0	2.6.0
SIGComputer (172.16.2.86:50010)	1	In Service	258.57 GB	860.33 MB	22.56 GB	235.17 GB	94	860.33 MB (0.32%)	0	2.6.0
localhost (172.16.2.102:50010)	1	In Service	262.32 GB	1.68 GB	18.19 GB	242.44 GB	118	1.68 GB (0.64%)	0	2.6.0

Decomissioning

Node	Last contact	Under replicated blocks	Blocks with no live replicas	Under Replicated Blocks In files under construction

Hadoop, 2014.

Legacy UI