Apache Hadoop 2.6.0 Multi-Node Cluster on CentOS

A guide to install and setup Multi-Node Apache Hadoop 2.6.0 Cluster

Version 1.0

Software Requirements
✓ VMware Player or Oracle Virtual Box ✓ CentOS Virtual Machine
Hardware Requirements
✓ Intel Core i3 processor or higher ✓ **8** GB RAM Recommended ✓ **3**00 GB for VM Recommended (By default 40 GB is taken)

Introduction

This setup and configuration document is a guide to setup a Multi-Node Apache Hadoop 2.6 cluster on a CentOS virtual machine on your PC.

The guide describes the whole process in four parts:

Section 1: Setting up the Cent OS for Hadoop 2.6.0

This section describes step by step guide to download, configure a CentOS Virtual Machine image in Oracle virtual box, and provides steps to install pre-requisites for Hadoop Installation on CentOS.

Section 2: Setting up ssh key

This section explains How to set up the Ssh key for login to the node without authentication.

Section 3: Installing Java and setting the Path for Java and Hadoop

This section describe how to set up the Path for Hadoop and Java environment variables.

Section 4: Setting up Hadoop-2.6.0 Multimode cluster

This section explains how to edit the hadoop configuration files, and start the daemons in all the nodes.

Note: The configuration described here is intended for learning purposes only.

Section-1: Setting up the CentOS Virtual Machine.

1.1: Download the CentOS from the below link.

https://edureka.wistia.com/medias/n8s4sh3tek/download?media_file_id=44348215

Extract the CentOS using WinRAR. You will get the CentOS virtual machine Image.

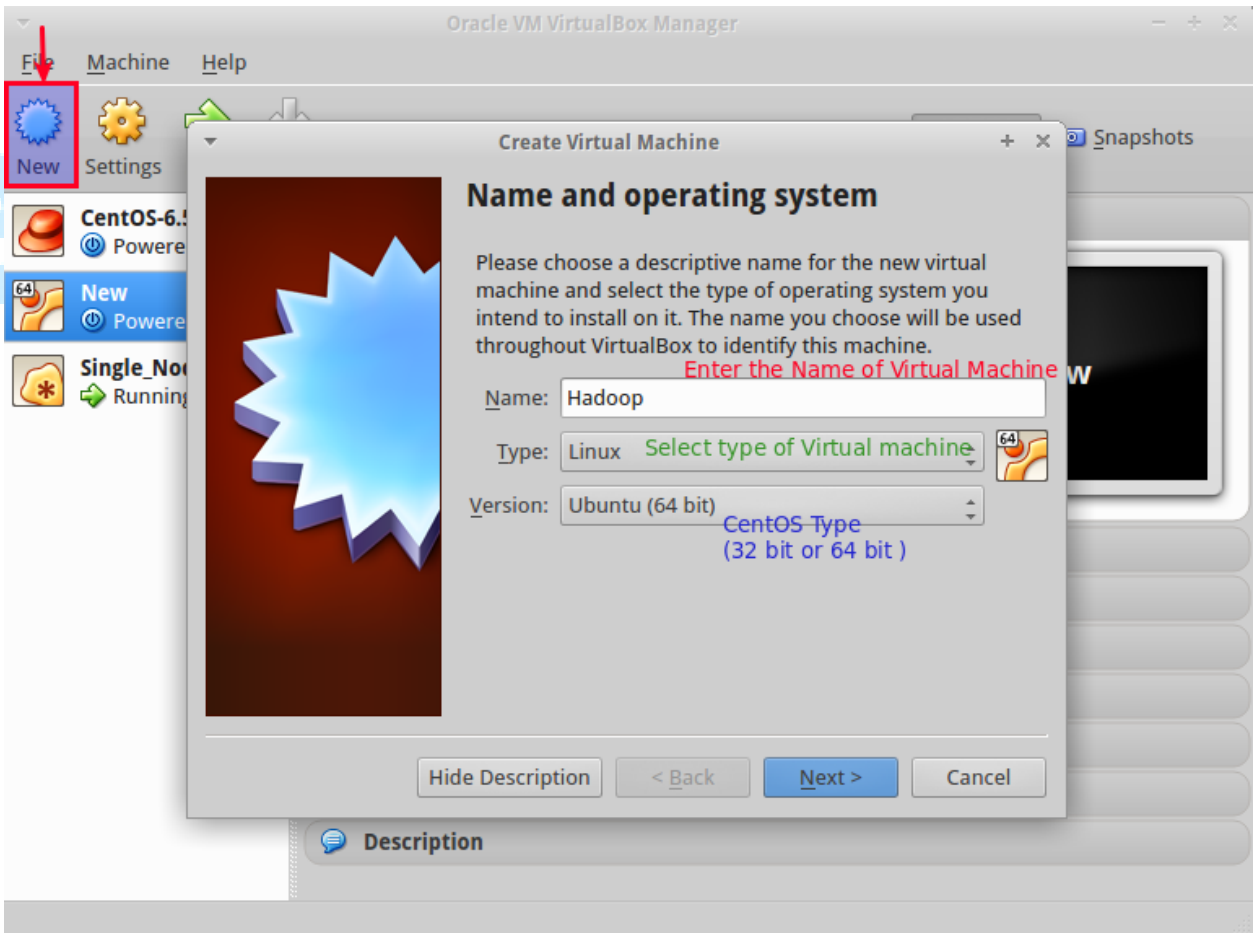
1.2: Download the install the Oracle virtual box or VMware player to open the CentOS Virtual machine.

Oracle Virtual box: **http://www.oracle.com/technetwork/server-storage/virtualbox/downloads/index.html**

Or

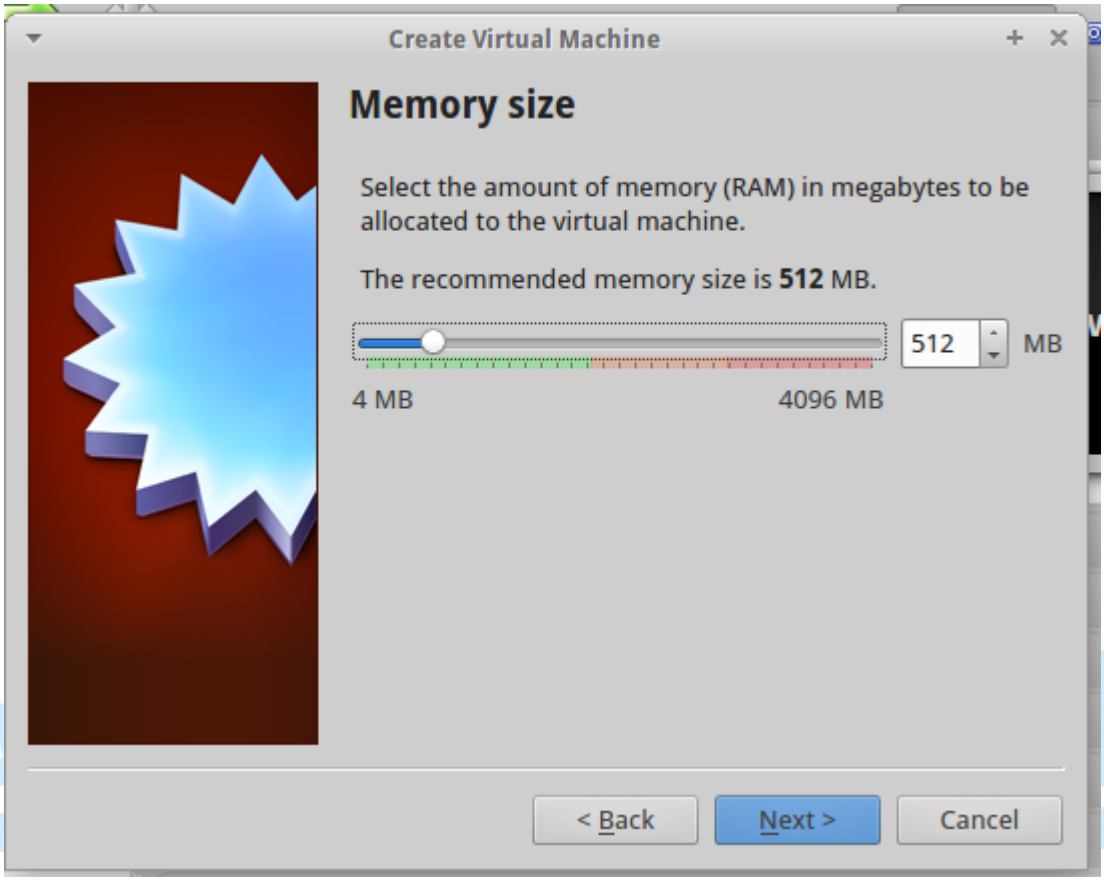
VMware Player: https://www.vmware.com/tryvmware/?p=player

1.3: In an oracle virtual box Click on New and Add the CentOS properties.



Click on Next button.

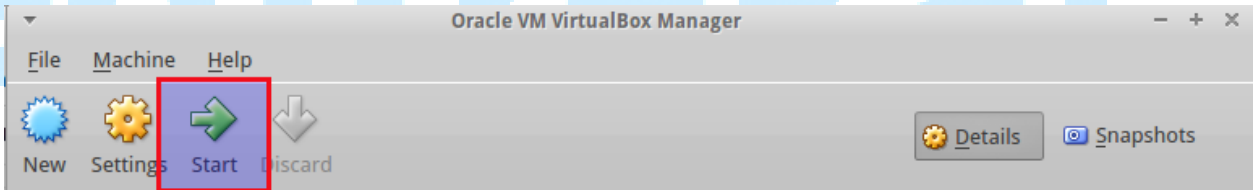
1.4: Add the RAM to your Virtual Machine. You can increase the Virtual machine RAM by dragging forward and backward.



Click on next.

1.5 : Select 3rd Option (Use an existing virtual hard drive file)and click on the folder icon, and go to the path where you have extracted the CentOS virtual machine in 1.1 step, Select centos-6.2-x64-virtual-machine-org.vmdk file. Click on Create button.

Click on Start Button.



1.6: It open the CentOS virtual Machine with the user tom.

User name: tom

Password: tomtom

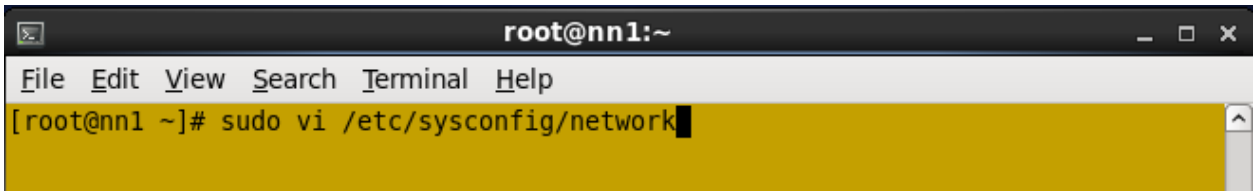
Open the terminal and login to root user.

Command: su - root

Password: tomtom

1.7: Change the Hostname to the virtual machine.

Command: sudo vi /etc/sysconfig/network



Modify the HOSTNAME value to your host name. In my case the host name for the name node is **nn1.cluster.com**.

Add the nn1.cluster.com at the HOSTNAME. We have open the network file using vi editor, To edit this file press button i.

```
root@nn1:~
File  Edit  View  Search  Terminal  Help
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=nn1.cluster.com
```

Save the file by pressing Esc button, colon (:) wq buttons, and press enter.

1.8: Stop the iptables

Run the below commands to stop the Iptables.

Service iptables stop

Chkconfig iptables off

```
root@nn1:~
File  Edit  View  Search  Terminal  Help
[root@nn1 ~]# service iptables stop
[root@nn1 ~]# chkconfig iptables off
[root@nn1 ~]#
```

1.9: Create the user.

Command: adduser edureka

```
root@nn1:~
File  Edit  View  Search  Terminal  Help
[root@nn1 ~]# adduser edureka
```

Create the password for the user edureka.

Command: passwd edureka

```
root@nn1:~
File  Edit  View  Search  Terminal  Help
[root@nn1 ~]# passwd edureka
Changing password for user edureka.
New password:
BAD PASSWORD: it is based on a dictionary word
BAD PASSWORD: is too simple
Retype new password:
passwd: all authentication tokens updated successfully.
[root@nn1 ~]#
```

1.10: Add the user to sudoers file, to give the sudoers permissions to user edureka.

Command: visudo

```
root@nn1:~
File  Edit  View  Search  Terminal  Help
[root@nn1 ~]# visudo
```

Add the user edureka below link ##Allow root to run any commands anywhere.

Edureka ALL= (ALL) ALL

```
root@nn1:~
File  Edit  View  Search  Terminal  Help
# commands via sudo.
#
# Defaults   env_keep += "HOME"

Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
edureka ALL=(ALL) ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOC
ATE, DRIVERS
```

Close the editor by Press the Esc button, Colon (:) wq buttons.

Do above all the process to all the Nodes (Data node, Resource manager Node

Virtual machines).

Reboot the Virtual Machine and log in to the user edureka.

1.11: Add the host names to every host file in the cluster.

Enter **ifconfig** command to get the IP address of your Virtual Machine.

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ ifconfig
eth1      Link encap:Ethernet  HWaddr 08:00:27:1A:80:08
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe1a:8008/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:217000 errors:0 dropped:0 overruns:0 frame:0
          TX packets:84699 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:207045749 (197.4 MiB)  TX bytes:4621710 (4.4 MiB)

eth2      Link encap:Ethernet  HWaddr 08:00:27:70:4F:30
          inet addr:192.168.1.96  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe70:4f30/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:337617 errors:0 dropped:0 overruns:0 frame:0
          TX packets:759507 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:183179504 (174.6 MiB)  TX bytes:1026770002 (979.2 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2077 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2077 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:199510 (194.8 KiB)  TX bytes:199510 (194.8 KiB)

[edureka@nn1 ~]$
```

Find the IP address of each node and add every node's IP address and hostname to all the nodes in hosts file.

1.12: Open the Hosts file.

Command: sudo gedit /etc/hosts

```
[edureka@nn1 ~]$ sudo gedit /etc/hosts
[sudo] password for edureka:
```

*hosts (/etc) - gedit

File Edit View Search Tools Documents Help

Open Save Undo

*hosts

```
#127.0.0.1              localhost.localdomain localhost
#::1            localhost6.localdomain6 localhost6
192.168.1.96    nn1.cluster.com nn1
192.168.1.71    dd1.cluster.com dd1
192.168.1.72    dd2.cluster.com dd2
192.168.1.73    rm.cluster.com rm
```

In my case I have two data nodes (Data node1 IP address is 192.168.1.71 and it host name is dd1.cluster.com, Data node2 IP address is 192.168.1.72 and it host name is dd2.cluster.com, Resource manager IP address is 192.168.1.73 and it's hostname is rm.cluster.com).

Add the all nodes IP addresses and host names to every VM hosts file.

Close the file and restart all the virtual machine.

Log in to the root user and restart the sshd service.

Command Service sshd restart

root@nn1:~

File Edit View Search Terminal Help

```
[root@nn1 ~]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
[root@nn1 ~]#
```

Section 2: Setting up Ssh key

2.1: Create the ssh key in all the nodes.

Command: ssh-keygen –t rsa

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/edureka/.ssh/id_rsa):
/home/edureka/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/edureka/.ssh/id_rsa.
Your public key has been saved in /home/edureka/.ssh/id_rsa.pub.
The key fingerprint is:
45:a2:af:97:03:3a:72:e3:37:c5:e3:79:83:9d:72:c0 edureka@nn1.cluster.com
The key's randomart image is:
+--[ RSA 2048]----+
|        . .      |
|       . o       |
|      .   .      |
|       . .       |
|      .oS        |
|     . oE.       |
|  . = .o+* .     |
|   + oo.=.*      |
|    .. . + .     |
+-----------------+
[edureka@nn1 ~]$
```

Don't give any path at the Enter file in which to save the key and don't give any passphrase, Press enter button.

Do the ssh key generation process in all the nodes.

Once ssh key is generated you will get the public key and private key.

2.2: Copy the public key to all the nodes.

You have to copy the Name nodes ssh public key to all the nodes.

2.3: In Namenode copy the id_rsa.pub using cat command.

Command: cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
[edureka@nn1 ~]$
```

2.4: Copy the Namenode public key to all the nodes using **SCP** command.

Command: scp ~/.ssh/id_rsa.pub edureka@<IP address of node>:/home/edureka/.ssh/authorized_keys

Name Node to Datanode1 (Data node 1 IP address is 192.168.1.71)

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ scp ~/.ssh/id_rsa.pub edureka@192.168.1.71:/home/edureka/.ssh/authorized_keys
```

Name Node to Datanode2 (Data node 2 IP address is 192.168.1.72)

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ scp ~/.ssh/id_rsa.pub edureka@192.168.1.72:/home/edureka/.ssh/authorized_keys
```

Name Node to Resource Manager (Resource manager IP address is 192.168.1.73)

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ scp ~/.ssh/id_rsa.pub edureka@192.168.1.73:/home/edureka/.ssh/authorized_keys
```

Change the Permission to authorized_keys. (Do the step in all the nodes)

Command: chmod 755 ~/.ssh/authorized_keys

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ chmod 755 ~/.ssh/authorized_keys
```

Now you can log in to every node from name node without authentication.

Section 3: Installing Java and setting the Path for Java and Hadoop

3.1: Download the JDK1.7 tar ball.

Command: wget http://download.oracle.com/otn-pub/java/jdk/7u79-b15/jdk-7u79-linux-x64.tar.gz

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ wget http://download.oracle.com/otn-pub/java/jdk/7u79-b15/jdk-7u79-linux-x64.tar.gz
```

3.2: Extract the Java tar ball.

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ tar -xvf jdk-7u79-linux-x64.tar.gz
```

3.3: Download the stable Hadoop tar ball to from apache Hadoop site.

Command: wget http://mirrors.advancedhosters.com/apache/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ wget http://mirror.reverse.net/pub/apache/hadoop/common/hadoop-
2.6.0/hadoop-2.6.0.tar.gz
```

3.4: Extract the Hadoop tar ball.

Command: tar –xvf hadoop-2.6.0.tar.gz

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ tar -xvf hadoop-2.6.0.tar.gz
```

3.5: Add the Hadoop and Java paths .bashrc file.

Open the .bashrc file.

Command: sudo gedit ~/.bashrc

Add the below paths:

export HADOOP_HOME=< Path to your Hadoop-2.6.0 directory>

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_CONF_DIR=$HADOOP_HOME/etc/hadoop

export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop

export JAVA_HOME=<Path to your Java Directory>

export PATH=$PATH: $JAVA_HOME/bin: $HADOOP_HOME/bin: $HADOOP_HOME/sbin

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ sudo gedit ~/.bashrc
[sudo] password for edureka:
```

```
*.bashrc (/home/edureka) - gedit
File  Edit  View  Search  Tools  Documents  Help
Open  Save  Undo
*.bashrc
        . /etc/bashrc
fi

# User specific aliases and functions

#----------HADOOP PATH--------------

export HADOOP_HOME=/home/edureka/HA/hadoop-2.6.0
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_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
#-----------------------------------

#----------JAVA PATH----------------
export JAVA_HOME=/home/edureka/HA/jdk1.7.0_79
export PATH=$PATH:$JAVA_HOME/bin
#-----------------------------------
```

Save the file and close it.

To apply all these changes to current running Terminal run the source command.

Command: source ~/.bashrc

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ source ~/.bashrc
[edureka@nn1 ~]$
```

Check Java and Hadoop is installed or not by finding Java and Hadoop.

Find the Java Version

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ java -version
java version "1.7.0_79"
Java(TM) SE Runtime Environment (build 1.7.0_79-b15)
Java HotSpot(TM) 64-Bit Server VM (build 24.79-b02, mixed mode)
[edureka@nn1 ~]$
```

Find the Hadoop version.

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ hadoop version
Hadoop 2.6.0
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r e3496499ecb8d22
0fba99dc5ed4c99c8f9e33bb1
Compiled by jenkins on 2014-11-13T21:10Z
Compiled with protoc 2.5.0
From source with checksum 18e43357c8f927c0695f1e9522859d6a
This command was run using /home/edureka/HA/hadoop-2.6.0/share/hadoop/common/had
oop-common-2.6.0.jar
[edureka@nn1 ~]$
```

Section 4: Setting up Hadoop-2.6.0 Multimode cluster

4.1: Edit the Hadoop Configuration files.

All the Hadoop configuration files are located in Hadoop-2.6.0/etc/hadoop directory.

Change the directory to hadoop directory.

```
edureka@nn1:~/HA/hadoop-2.6.0/etc/hadoop
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ cd /home/edureka/HA/hadoop-2.6.0/etc/hadoop/
[edureka@nn1 hadoop]$ ls
capacity-scheduler.xml      httpfs-env.sh            mapred-env.sh
configuration.xsl           httpfs-log4j.properties  mapred-queues.xml.template
container-executor.cfg      httpfs-signature.secret  mapred-site.xml
core-site.xml               httpfs-site.xml          mapred-site.xml.template
hadoop-env.cmd              kms-acls.xml             slaves
hadoop-env.sh               kms-env.sh               ssl-client.xml.example
hadoop-metrics2.properties  kms-log4j.properties     ssl-server.xml.example
hadoop-metrics.properties   kms-site.xml             yarn-env.cmd
hadoop-policy.xml           log4j.properties         yarn-env.sh
hdfs-site.xml               mapred-env.cmd           yarn-site.xml
[edureka@nn1 hadoop]$
```

The configuration files that need to change is:

Configuration Filenames	Description
hadoop-env.sh	Environment variables that are used in the scripts to run Hadoop.
core-site.xml	Configuration settings for Hadoop Core such as I/O settings that Are common to HDFS and MapReduce.
hdfs-site.xml	Configuration settings for HDFS daemons, the namenode, The secondary namenode and the data nodes.
mapred-site.xml	Configuration settings for MapReduce Applications.
yarn-site.xml	Configuration settings for ResourceManager and NodeManager.
Slaves	Contain the Each Datanode IP address to identify the slave nodes.

4.2: Open the core-site.xml. In a core-site.xml file you have to add the Namenode ip address or hostname. And core-site.xml file properties ae same in all the nodes.

Add the property tab between the configuration tag.

The Property tag contain the name tag and value tags.

Property	Description
fs.defaultFS	The name of the default file system. A URI whose scheme and authority Determine the File System implementation. The Uri's scheme determines the config property (fs.SCHEME.impl) naming The File System implementation class. The uri's authority is used to determine the host, port, etc. for a filesystem.

In the core-site.xml we have to mention the namenode hostname to identify the namenode daemons.

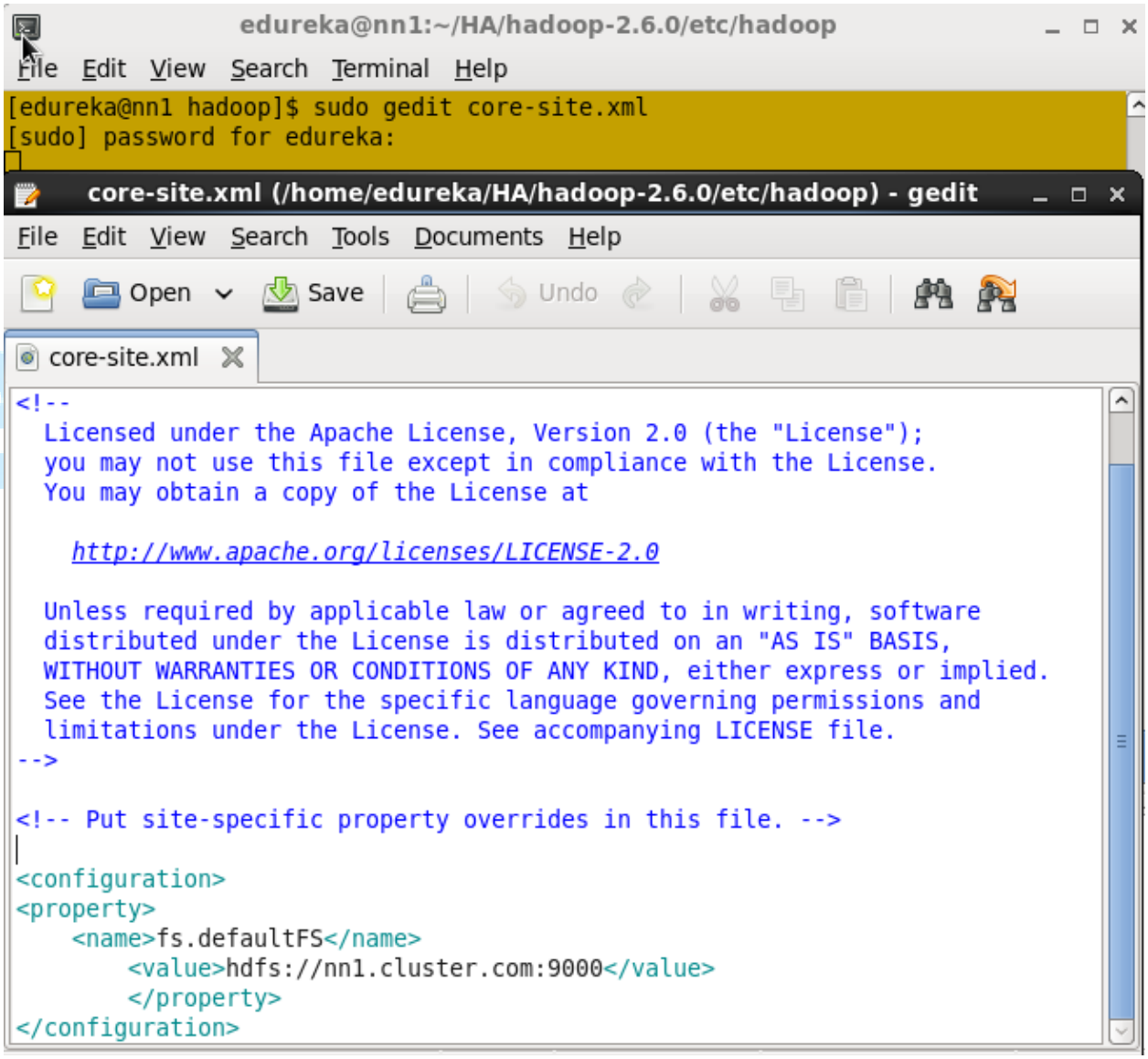
<property>

<name>**fs.defaultFS**</name>

<value>**hdfs://<NameNode Hostname>:<port number>**</value>

</property>

In my case namenode Hostname is nn1.cluster.com



4.3: Hdfs-site.xml

Property	Description
dfs.namenode.name.dir	Determines where on the local filesystem the DFS name node Should store the name table (fsimage). If this is a comma-delimited list of directories then the name Table is replicated in all of the directories, for redundancy.
dfs.namenode.name.dir	Determines where on the local filesystem a DFS data node Should store its blocks. If this is a comma-delimited list of Directories, then data will be stored in all named directories, different devices. Directories that do not exist are ignored.
dfs.replication	Default block replication. The actual number of replications can when the file is created. The default is used if replication is not Specified in create time.
dfs.permissions.enabled	If "true", enable permission checking in HDFS. If "false", permission Checking is turned off, but all other behaviour is unchanged. Switching from one parameter value to the other does not Change the mode, owner or group of files or directories.

Before editing hdfs-site.xml you have to create the one directory and to store the namenode Meta data.

Command: mkdir namenode

Change the permission to namenode directory.

Command: Chmod 755 namenode

Use this namenode directory path in name node's hdfs-site.xml file.

In a namenode hdfs-site.xml use this namenode directory path for the dfs.namenode.name.dir property.

Namenode hdfs-site.xml :

```
edureka@nn1:~/HA/hadoop-2.6.0/etc/hadoop
File  Edit  View  Search  Terminal  Help
[edureka@nn1 hadoop]$ sudo gedit hdfs-site.xml
*hdfs-site.xml (/home/edureka/HA/hadoop-2.6.0/etc/hadoop) - gedit
File  Edit  View  Search  Tools  Documents  Help
Open  Save  Undo
*hdfs-site.xml
  limitations under the License. See accompanying LICENSE file.
-->

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

<configuration>
        <property>
        <name>dfs.namenode.name.dir</name>
        <value>/home/edureka/HA/data/namenode</value>
        </property>

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

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

</configuration>
```

4.4: Edit the mapred-site.xml file.

Property	Description
mapreduce.framework.name	Execution framework set to Hadoop YARN.

In some cases mapred-site.xml file will not available, you have to create the mapred-site.xml using mapred-site.xml template.

```
edureka@nn1:~/HA/hadoop-2.6.0/etc/hadoop
File  Edit  View  Search  Terminal  Help
[edureka@nn1 hadoop]$ cp mapred-site.xml.template mapred-site.xml
```

Open the mapred-site.xml file.

File Edit View Search Terminal Help

```
[edureka@nn1 hadoop]$ gedit mapred-site.xml
```

*mapred-site.xml (~/HA/hadoop-2.6.0/etc/hadoop) - gedit

File Edit View Search Tools Documents Help

Open Save Undo

*mapred-site.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.
-->

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

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

XML Tab Width: 8 Ln 12, Col 32 INS

4.5: Open the yarn-site.xml file to add the resource manager properties.

Property	Description
yarn.nodemanager.aux-services	Selects a shuffle service that needs to be set for MapReduce This property, in conjunction with other properties, sets "Direct shuffle" as the default shuffle for MapReduce. Default value: mapreduce_shuffle, mapr_direct_shuffle
yarn.nodemanager.aux-services.mapreduce_shuffle.class	This property, in conjunction with other properties, sets "Direct shuffle" as the default shuffle for MapReduce. Default value: org.apache.hadoop.mapred.ShuffleHandler
yarn.resourcemanager.resource-tracker.address	Provide the resource tracker details to Yarn services.
yarn.resourcemanager.scheduler.address	Applications in the cluster talk to the ResourceManager.
yarn.resourcemanager.address	The hostname of the ResourceManager and the port on whic can talk to the Resource Manager. Example value: ${yarn.resourcemanager.hostname}:{Port number}

```
<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><Resource manager Hostname>:9001</value>
</property>

<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value><Resource manager Hostname>:9002</value>
</property>

<property>
<name>yarn.resourcemanager.address</name>
<value><Resource manager Hostname>:9003</value>
</property>
```

edureka@nn1:~/HA/hadoop-2.6.0/etc/hadoop

File Edit View Search Terminal Help

```
[edureka@nn1 hadoop]$ gedit yarn-site.xml
```

yarn-site.xml (~/HA/hadoop-2.6.0/etc/hadoop) - gedit

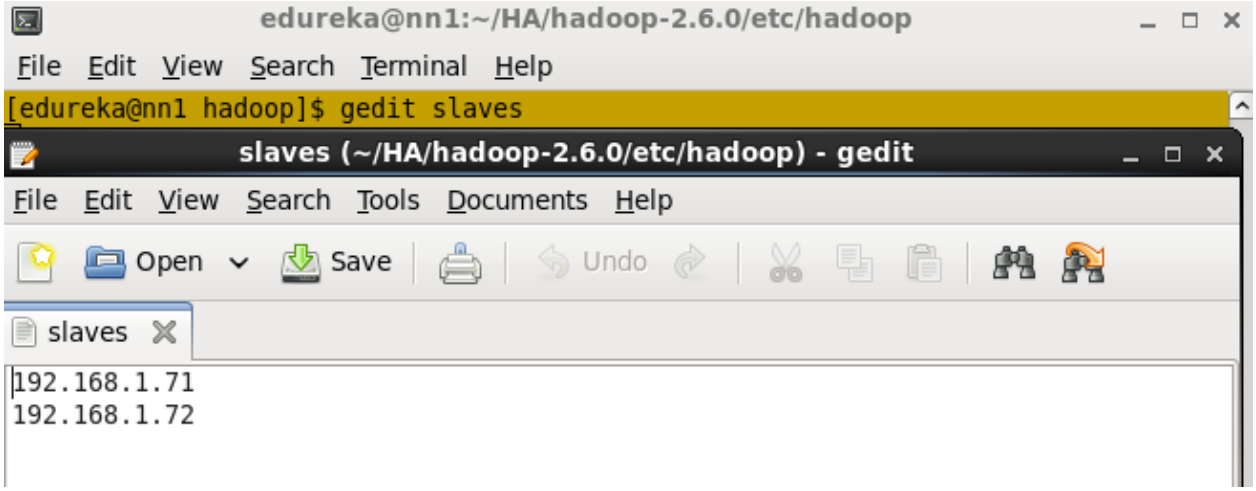
File Edit View Search Tools Documents Help

Open Save Undo

yarn-site.xml

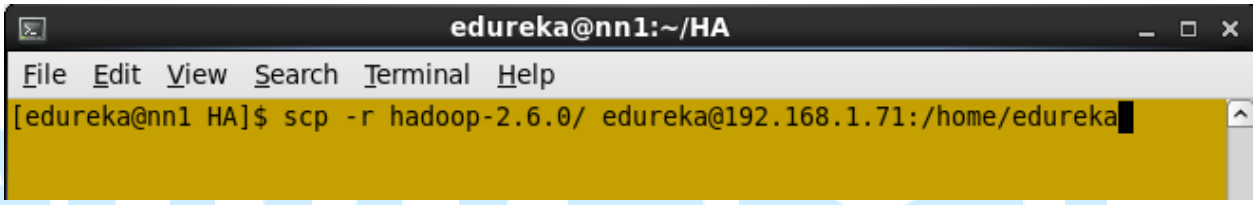
```
<property>
<name>yarn.nodemanager.aux-services.mapreduce_shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>

<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>rm.cluster.com:9001</value>
</property>

<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>rm.cluster.com:9002</value>
</property>

<property>
<name>yarn.resourcemanager.address</name>
<value>rm.cluster.com:9003</value>
</property>
</configuration>
```

4.6: Open slaves file and add the Data nodes IP address.

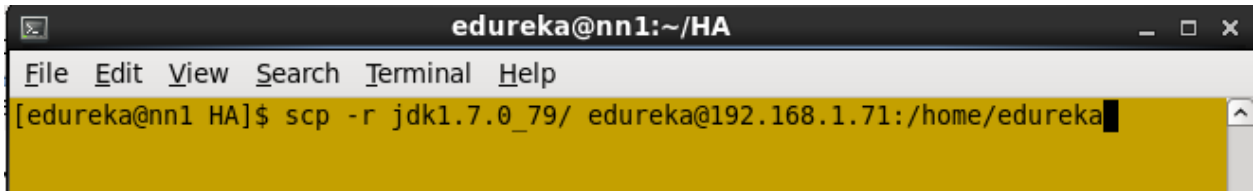


4.7: Copy the Hadoop, Java and .bashrc file to all the nodes from the namenode using SCP command.

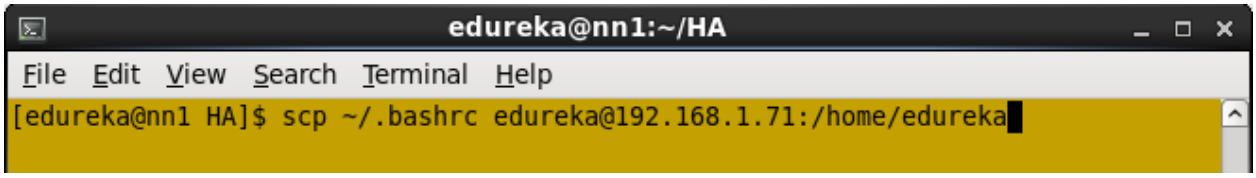
Copy the hadoop files:



Copy the Java files:



Copy the .bashrc file:



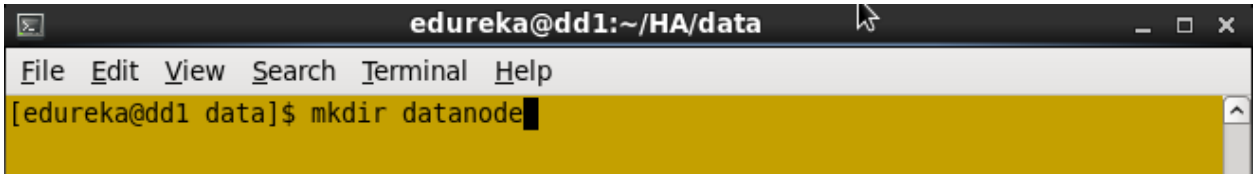
Copy the hadoop, Java and .bashrc file to all the nodes form the namenode.

Change the Hadoop and java paths in .bashrc file in each node according to the respective node.

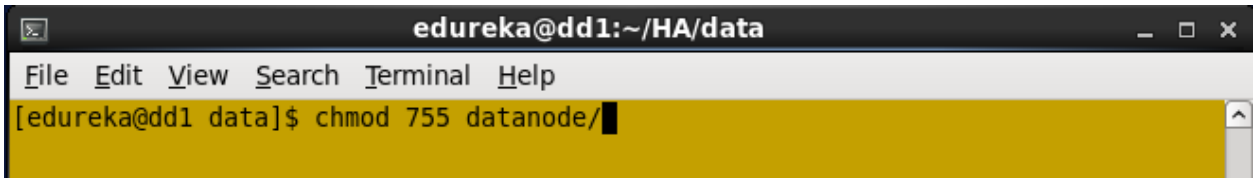
4.8: In a data node you have to add the dfs.datanode.data.dir properties.

Create the one directory in each Datanodes to store the blocks.

In my case created datanode directory to store the blocks.



Change the Permission to data node directory.



4.9: Open the hdfs-site.xml file, add this Datanode directory path in dfs.datanode.data.dir property.

```
edureka@dd1:~/HA/hadoop-2.6.0/etc/hadoop
File  Edit  View  Search  Terminal  Help
[edureka@dd1 hadoop]$ sudo gedit hdfs-site.xml
[sudo] password for edureka:
*hdfs-site.xml (/home/edureka/HA/hadoop-2.6.0/etc/hadoop) - gedit
File  Edit  View  Search  Tools  Documents  Help
Open  Save  Undo
*hdfs-site.xml

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

<configuration>

        <property>
        <name>dfs.datanode.data.dir</name>
        <value>/home/edureka/HA/data/datanode</value>
        </property>

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

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

</configuration>
```

4.10: Open the Slave file in each data node and add the It's IP address only.

```
edureka@dd1:~/HA/hadoop-2.6.0/etc/hadoop
File  Edit  View  Search  Terminal  Help
[edureka@dd1 hadoop]$ gedit slaves
slaves (~/HA/hadoop-2.6.0/etc/hadoop) - gedit
File  Edit  View  Search  Tools  Documents  Help
Open  Save  Undo
slaves
192.168.1.71
```

Do the same in each data node.

4.11: Format the Name Node.

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ hadoop namenode -format
```

4.12: Once you formatted the namenode, start all the daemons.

You can start the daemons from namenode to all the nodes, you can use start-all.sh or use start-dfs.sh and start-yarn.sh script.

Start the DFS daemons.

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ start-dfs.sh
15/04/28 15:14:02 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Starting namenodes on [nn1.cluster.com]
nn1.cluster.com: starting namenode, logging to /home/edureka/HA/hadoop-2.6.0/log
s/hadoop-edureka-namenode-nn1.cluster.com.out
192.168.1.71: starting datanode, logging to /home/edureka/HA/hadoop-2.6.0/logs/h
adoop-edureka-datanode-dd1.cluster.com.out
192.168.1.72: starting datanode, logging to /home/edureka/HA/hadoop-2.6.0/logs/h
adoop-edureka-datanode-dd2.cluster.com.out
```

Once you run the start-dfs.sh file you can see the Name node daemon in master and Data node daemon in slave machines.

4.13: Start yarn daemons.

```
edureka@nn1:~
File  Edit  View  Search  Terminal  Help
[edureka@nn1 ~]$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /home/edureka/HA/hadoop-2.6.0/logs/yarn-edu
reka-resourcemanager-nn1.cluster.com.out
192.168.1.71: starting nodemanager, logging to /home/edureka/HA/hadoop-2.6.0/log
s/yarn-edureka-nodemanager-dd1.cluster.com.out
192.168.1.72: starting nodemanager, logging to /home/edureka/HA/hadoop-2.6.0/log
s/yarn-edureka-nodemanager-dd2.cluster.com.out
[edureka@nn1 ~]$
```

Once you run the start-yarn.sh file you can see the Resource manager daemon in master and Node manager daemon in slave machines.