**Hadoop 2 cluster build**

**Cluster:**

**Nodes and roles assignments**

192.168.0.51 nn.cluster.com

192.168.0.52 rm.cluster.com

192.168.0.53 snn.cluster.com

192.168.0.54 dn1.cluster.com

192.168.0.55 dn2.cluster.com

**Build namenode:-**

**@nn**

[root@hadoop ~]# ll soft/

total 329596

-rw-r--r-- 1 root root 195257604 May 12 21:04 hadoop-2.6.0.tar.gz

-rw-r--r-- 1 root root 142245547 May 12 21:04 jdk-7u75-linux-x64.tar.gz

[root@hadoop ~]# groupadd hadoop

[root@hadoop ~]# useradd -g hadoop hadoop

[root@hadoop ~]#

[root@hadoop ~]#

[root@hadoop ~]# passwd hadoop

passwd: all authentication tokens updated successfully.

[root@hadoop ~]#

[root@hadoop ~]#

[root@hadoop ~]# mv soft/\* /home/hadoop/

[root@hadoop ~]#

[root@hadoop ~]#

[root@hadoop ~]# ll /home/hadoop/

total 329596

-rw-r--r-- 1 root root 195257604 May 12 21:04 hadoop-2.6.0.tar.gz

-rw-r--r-- 1 root root 142245547 May 12 21:04 jdk-7u75-linux-x64.tar.gz

[root@hadoop ~]#

[root@hadoop ~]#

[root@hadoop ~]# chown -R hadoop:hadoop /home/hadoop/\*

[root@hadoop ~]#

[root@hadoop ~]#

[root@hadoop ~]# ll /home/hadoop/

total 329596

-rw-r--r-- 1 hadoop hadoop 195257604 May 12 21:04 hadoop-2.6.0.tar.gz

-rw-r--r-- 1 hadoop hadoop 142245547 May 12 21:04 jdk-7u75-linux-x64.tar.gz

[root@hadoop ~]#

[root@hadoop ~]#

[root@hadoop ~]# su - hadoop

[hadoop@hadoop ~]$

[hadoop@hadoop ~]$

[hadoop@hadoop ~]$  
[hadoop@hadoop ~]$ tar zxvf hadoop-2.6.0.tar.gz

[hadoop@hadoop ~]$ tar zxvf jdk-7u75-linux-x64.tar.gz

[hadoop@hadoop ~]$ ln -s hadoop-2.6.0 hadoop

[hadoop@hadoop ~]$

[hadoop@hadoop ~]$

[hadoop@hadoop ~]$ ln -s jdk1.7.0\_75 jdk

[hadoop@hadoop ~]$

[hadoop@hadoop ~]$

[hadoop@hadoop ~]$  
[hadoop@hadoop ~]$ ll

total 329604

lrwxrwxrwx 1 hadoop hadoop 12 Aug 15 16:44 hadoop -> hadoop-2.6.0

drwxr-xr-x 9 hadoop hadoop 4096 Nov 14 2014 hadoop-2.6.0

-rw-r--r-- 1 hadoop hadoop 195257604 May 12 21:04 hadoop-2.6.0.tar.gz

lrwxrwxrwx 1 hadoop hadoop 11 Aug 15 16:44 jdk -> jdk1.7.0\_75

drwxr-xr-x 8 hadoop hadoop 4096 Dec 19 2014 jdk1.7.0\_75

-rw-r--r-- 1 hadoop hadoop 142245547 May 12 21:04 jdk-7u75-linux-x64.tar.gz

[hadoop@hadoop ~]$

[hadoop@hadoop ~]$

[hadoop@hadoop ~]$ vim **.bash\_profile**

JAVA\_HOME=/home/hadoop/jdk

HADOOP\_HOME=/home/hadoop/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\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

export YARN\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

PATH=$PATH:$HOME/bin:$HADOOP\_HOME:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin:$JAVA\_HOME:$JAVA\_HOME/bin

export PATH



[hadoop@hadoop ~]$ vim hadoop/etc/hadoop/**hadoop-env.sh**

JAVA\_HOME=/home/hadoop/jdk

[hadoop@hadoop ~]$ vim hadoop/etc/hadoop/**core-site.xml**

<property>

<name>fs.defaultFS</name>

<value>hdfs://192.168.0.51:8020</value>

</property>

[hadoop@hadoop ~]$ vim hadoop/etc/hadoop/**hdfs-site.xml**

<configuration>

<property>

<name>dfs.namenode.name.dir</name>

<value>/home/hadoop/data/nn</value>  
</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>/home/hadoop/data/dn</value>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.block.size</name>

<value>268435456</value>

</property>

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>192.168.0.53:50090</value>

</property>

<property>

<name>dfs.namenode.checkpoint.dir</name>

<value>/home/hadoop/data/snn</value>

</property>

[hadoop@hadoop ~]$ cp hadoop/etc/hadoop/mapred-site.xml.template hadoop/etc/hadoop/mapred-site.xml

[hadoop@hadoop ~]$ vim hadoop/etc/hadoop**/mapred-site.xml**

<property>

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

<value>yarn</value>

</property>

[hadoop@hadoop ~]$ vim hadoop/etc/hadoop/**yarn-site.xml**

<property>

<name>yarn.resourcemanager.address</name>

<value>192.168.0.52:8032</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>192.168.0.52:8030</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>192.168.0.52:8025</value>

</property>

<property>

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

<value>mapreduce\_shuffle</value>

</property>   
<property>

<name>yarn.nodemanager.aux-services.mapreduce\_shuffle</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

[hadoop@hadoop ~]$ vim **hadoop/etc/hadoop/slaves**

192.168.0.54

192.168.0.55

[root@hadoop ~]# vim **/etc/hosts**

192.168.0.51 nn.cluster.com nn

192.168.0.52 rm.cluster.com rm

192.168.0.53 snn.cluster.com snn

192.168.0.54 dn1.cluster.com dn1

192.168.0.55 dn2.cluster.com dn2

**Add the ssh keys:**

[hadoop@nn ~]$ ssh-copy-id -i .ssh/id\_rsa.pub 192.168.0.52

hadoop@192.168.0.52's password:

Now try logging into the machine, with "ssh '192.168.0.52'", and check in:

.ssh/authorized\_keys

to make sure we haven't added extra keys that you weren't expecting.

Create clones of nn as rm, snn, dn1, dn2 (vmware clone functionality)

Copy ssh public keys to other hosts

**Format hdfs namenode**

[hadoop@nn ~]$ hdfs namenode –format

**Start dfs**

[hadoop@nn ~]$ start-dfs.sh

**Start yarn**

hadoop@rm ~]$ start-yarn.sh

**Start jobhistory server**

[hadoop@rm ~]$ mr-jobhistory-daemon.sh start historyserver

[hadoop@nn ~]$ for i in {1..5}; do ssh 192.168.0.5$i "hostname;jdk/bin/jps;echo -e '\n'";done

**nn.cluster.com**

1624 NameNode

**rm.cluster.com**

1852 JobHistoryServer

1559 ResourceManager   
 **snn.cluster.com**

1410 SecondaryNameNode

**dn1.cluster.com**

1408 DataNode

1544 NodeManager

**dn2.cluster.com**

1567 NodeManager

1431 DataNode

**Testing**

hdfs dfs -ls /

[hadoop@rm ~]$ yarn jar hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar pi 20 20

**Add Datanode to cluster**

new datanode newdn.cluster.com 192.168.0.56

1. Add a new node with Linux OS.

2. Create the same group and user hadoop.

3. Copy the same hadoop and jdk binaries from one of the existing datanode. this will also copy the properties.

4. Copy the same .bash\_profile file in the new datanode.

5. Edit /etc/hosts file on namenode and new datanode and add the new datanode address.

6. Setup passwordless ssh from Namenode to new datanode and resource manager to new datanode.

7. in the Namenode:

Add the property:

in hdfs-site.xml --> name = dfs.hosts, value = location of a file that lists all the existing and new datanodes which are allowed to connect to the namenode.

8. In the Resource Manager:

Add the property:

in yarn-site.xml --> name = yarn.resourcemanager.nodes.include-path, value = location of a file that lists all the existing and new nodemanagers which are allowed to connect to the resourcemanager. Generally it is the same file as namenode has.

9. Run the commands:

on Namenode:

hdfs dfsadmin -refreshNodes

on Resource manager:

yarn rmadmin -refreshNodes

10. on the new datanode:

hadoop-daemon.sh start datanode

yarn-daemon.sh start nodemanager

11. Check if the new datanode and nodemanager appears in the web UI

12. Check dfsadmin –report

**Decommission datanode:**

1. Add the properties:

in Namenode: in hdfs-site.xml ->

name: dfs.hosts.exclude

value: location of a file that has list of all the datanodes which are not allowed to connect to the namenode

in Resource Manager: in yarn-site.xml ->

name: yarn.resourcemanager.nodes.exclude-path

value: location of a file that has list of all the node managers which are not allowed to connect to the resource manager

\* Make sure that the node to be decommissioned is not listed in the "includes" file

2. Run the commands:

on namenode: hdfs dfsadmin -refreshNodes

on Resource Manager: yarn rmadmin -refreshNodes

3. Check dfsadmin -report

4. Check if the new datanode and nodemanager disappears from the web UI

5. The namenode UI will show the admin state change to 'DECOMMISSION IN PROGRESS' for affected datanodes.

6. When all the datnodes report their state as 'DECOMMISSIONED' all the blocks will have been replicated elsewhere.

7. Shut the decommissioned datanode.

8. Remove the nodes from the 'includes' and 'excludes' files and update the namenode and resource manager.

**hdfs balancer -threshold x**

default - 1MB/sec

dfs.datanode.balance.bandwidthPerSec - hdfs-site.xml

Recommendation is: 0.1 x network speed

for 1 Gbps network .. use 10MB/sec