

Apache Hadoop Cluster Installation and Configuration (Not Vendor Specific, from Apache Software Foundation)

- [Apache Hadoop Cluster Installation and Configuration](#)
 - [Reference Links](#) :
 - [Software Vendor](#)
 - [Additional Hadoop resource links](#)
 - [Hadoop Lab Layout and Architecture](#) :
 - [1. Pre-configuration and Setup](#)
 - [1.1 Ensure that the /etc/hosts file appears configured below as so](#) :
 - [1.2 Set the password on ALL machines in the hadoop configuration, store these somewhere locally for later reference.](#)
 - [2. Installation of Hadoop and Single Node Cluster configuration](#)
 - [3. SSH Access between master and slave configuration](#)
 - [3.1 Configure ssh access locally for the 'hdfs' user:](#)
 - [3.2 Test for 'hdfs' user on the 'localhost' machine \(master\):](#)
 - [3.3 Test for the 'hdfs' user on the master by using the hostname :](#)
 - [3.4 Copy the public ssh key of the master to the authorized keys file of the slave under the 'hdfs' user:](#)
 - [3.5 Repeat from the master to the remaining 'slave' machines\(hadoop3, and hadoop4\) in the cluster:](#)
 - [3.6 Test from master\(hadoop1\) to slave\(hadoop3\) machine:](#)
 - [3.7 From the hadoop master machine to hadoop4 :](#)
 - [4. Configure masters and the slaves within the cluster.](#)
 - [4.1 On the hadoop 'master', configure the 'masters' file. If you have more than one hadoop master, all of the machines hostnames will go here\(configure all 'master' machines with the same "/etc/hadoop/masters" file\):](#)
 - [4.2 On the hadoop master, configure the 'slaves' file. Place the hostname of all of the hadoop 'slaves' in the cluster into here.](#)
 - [4.3 Configure *-site.xml files on ALL\(all masters, and all slaves\) machines within the cluster :](#)
 - [4.4 Edit the "/etc/hadoop/core-site.xml" file :](#)
 - [4.5 Distribute this configuration file to all other hadoop machines in the cluster:](#)
 - [4.6 Edit the "/etc/hadoop/core-site.xml" file :](#)
 - [4.7 Distribute this configuration files to all other hadoop machines in the cluster:](#)
 - [4.8 Edit the "/etc/hadoop/hdfs-site.xml" file :](#)
 - [5. Formatting the HDFS filesystem via the NameNode](#)
 - [6. Starting the multi-node cluster](#)
 - [6.1 HDFS daemons are started: the NameNode daemon is started on master\(hadoop1\), and DataNode daemons are started on all slaves \(hadoop2, hadoop3, hadoop4\).](#)
 - [6.2 MapReduce daemons are started: the JobTracker is started on master\(hadoop1\), and TaskTracker daemons are started on all slaves \(hadoop2, hadoop3, hadoop4\).](#)
 - [6.1.a Starting the NameNode daemon on the master\(hadoop1\):](#)
 - [6.1.b Starting the DataNode daemons on all slaves\(hadoop2, hadoop3, hadoop4\)](#)
 - [6.2.a Starting the Mapreduce daemon\(JobTracker\) daemon on the master\(hadoop1\):](#)
 - [6.2.b Starting the Mapreduce daemons\(TaskTracker\) daemon on all slaves\(hadoop2,hadoop3,hadoop4\):](#)
 - [7. Use the Web GUI tool on the hadoop master:](#)
 - [7.1 The NameNode web UI shows you a cluster summary including information about total/remaining capacity, live and dead nodes. Additionally, it allows you to browse the HDFS namespace and view the contents of its files in the web browser. It also gives access to the "local machine's" Hadoop log files.](#)

- [7.2 Click the link entitled "Live Nodes" to see the Hadoop DataNodes \(slaves\) status :](#)
- [8. Stopping the multi-node cluster:](#)
 - [8.1 First, stop the MapReduce daemons: the JobTracker is stopped on master \(hadoop1\), and TaskTracker daemons are stopped on all slaves \(hadoop2,hadoop3,hadoop4\)](#)
 - [8.2 Second, HDFS daemons are stopped: the NameNode daemon is stopped on master \(hadoop1\), and DataNode daemons are stopped on all slaves \(hadoop2,hadoop3,hadoop4\).](#)
- [9. Running a MapReduce job:](#)
 - [9.1 Start the Hadoop Cluster from the master \(hadoop1\) :](#)
 - [9.2 Copy the data input files to the master\(hadoop1\) to a local directory:](#)
 - [9.3 Ensure that the hadoop user \(hdfs\) has ownership of the data:](#)
 - [9.4 Copy local example data to HDFS :](#)
 - [9.5 Verify the data in the HDFS :](#)
 - [9.6 Now, we actually run the WordCount example job.\(A MapReduce job\):](#)
 - [9.7 Monitor the logs on the hadoop slaves :](#)
 - [9.8 The MapReduce job should complete shortly with output like below :](#)
 - [9.9 Now check the 'datanode' logs on the slave machines for completion:](#)
 - [9.10 Check if the result is successfully stored in HDFS directory](#)
 - [9.11 Retrieve the job result from HDFS to the local filesystem:](#)
- [10. Caveats and Known Problems:](#)
 - [10.1 Hadoop Troubleshooting Table :](#)
 - [10.2 Other resources for hadoop troubleshooting :](#)

Apache Hadoop Cluster Installation and Configuration

Reference Links :

Software Vendor

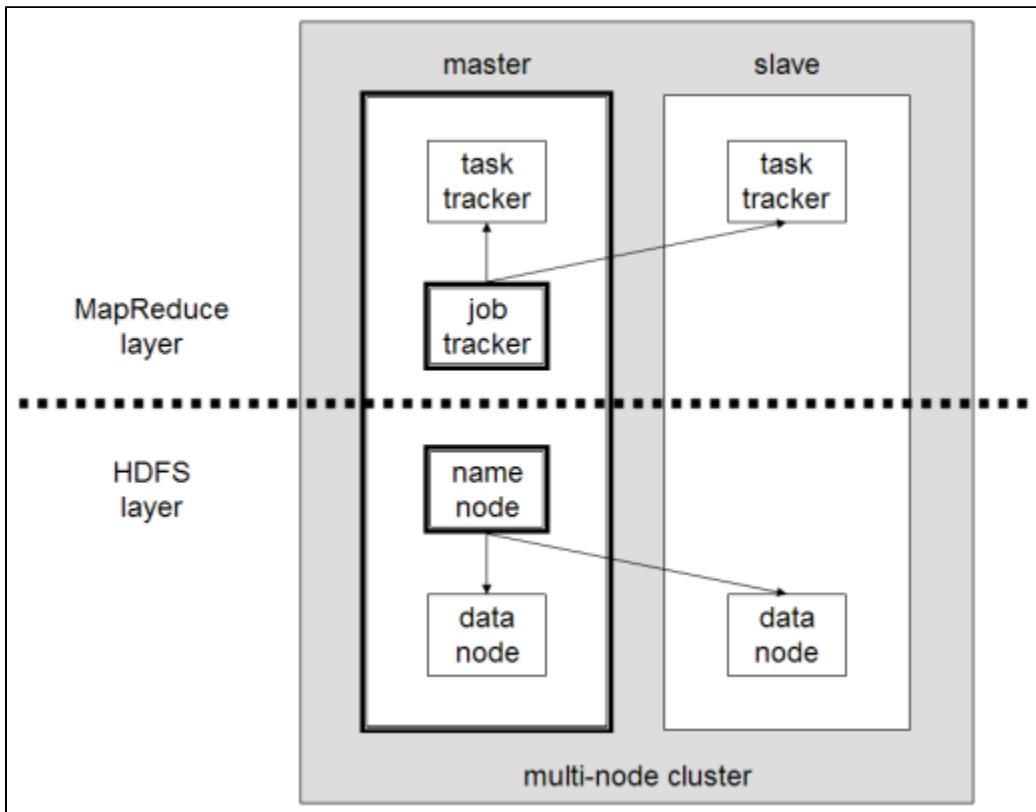
[Hortonworks](#)

Additional Hadoop resource links

[Apache Software Foundation Hadoop Home](#)
[Apache Hadoop stable release 1.0.3 Documentation](#)
[Apache Hadoop Wiki](#)
[Yahoo YDN Hadoop Tutorial](#)
[Running Hadoop On Linux \(Multi-Node Cluster\) - Michael G. Noll](#)

Hadoop Lab Layout and Architecture :

Server	Function
hadoop1.savvis.lab	master (NameNode, SecondaryNameNode and JobTracker)
hadoop2.savvis.lab	slave (DataNode and TaskTracker)
hadoop3.savvis.lab	slave (DataNode and TaskTracker)
hadoop4.savvis.lab	slave (DataNode and TaskTracker)



Hadoop multi-node cluster example

1. Pre-configuration and Setup

1.1 Ensure that the `/etc/hosts` file appears configured below as so :

```
# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
10.12.48.50   hadoop1 master
10.12.48.51   hadoop2 slave
10.12.48.52   hadoop3 master2
10.12.48.53   hadoop4 slave2
```

1.2 Set the password on ALL machines in the hadoop configuration, store these somewhere locally for later reference.


The following step has been done on the lab servers hadoop1, hadoop2, hadoop3, hadoop4, but here is only shown executed on machine 'hadoop1'.

```
[root@hadoop1 ~]# passwd hdfs
Changing password for user hdfs.
New password:
BAD PASSWORD: it is too short
BAD PASSWORD: is too simple
Retype new password:
passwd: all authentication tokens updated successfully.
```

2. Installation of Hadoop and Single Node Cluster configuration

[Hadoop Install and Configure Single Node Cluster](#)

3. SSH Access between master and slave configuration

 NOTE: The user for hadoop 'hdfs' will need to be able to access it's own user account (hdfs) on the master(hadoop1), as well as 'localhost' on the master, and all slaves under the user 'hdfs' as well.

3.1 Configure ssh access locally for the 'hdfs' user:

```

[root@hadoop1 hadoop]# su - hdfs
-bash-4.1$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
Generating public/private dsa key pair.
Your identification has been saved in /tmp/.ssh/id_dsa.
Your public key has been saved in /tmp/.ssh/id_dsa.pub.
The key fingerprint is:
84:eb:f7:10:9d:85:a2:1d:e7:0d:90:0f:c5:10:2b:ac hdfs@hadoop1
The key's randomart image is:
+--[ DSA 1024]-----+
|          +*          |
|         . .oo..     |
|          + =o+  .    |
|         . * *. =    |
|        E o S +  .    |
|          .          |
|         . o         |
|          . o         |
|           .         |
+-----+


-bash-4.1$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys

-bash-4.1$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
RSA key fingerprint is 81:7d:d6:30:1f:ab:b8:2f:e0:0b:20:75:e4:3f:47:16.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.

#####
This computer system and associated networks are for the sole business
use of SAVVIS, Inc.  authorized users.  The company's computers and
proprietary data and information stored on them remain at all times the
property of SAVVIS, Inc.  Users have no right to privacy as to any
information transmitted or stored in, by or through any portion of this
system.
#####

```

3.2 Test for 'hdfs' user on the 'localhost' machine (master):


 NOTE: You may need to first 'cache' the ssh key, if prompted for this, simply enter 'yes', and retry this test.

```
[hdfs@hadoop1 ~]$ ssh localhost

#####
This computer system and associated networks are for the sole business
use of SAVVIS, Inc.  authorized users.  The company's computers and
proprietary data and information stored on them remain at all times the
property of SAVVIS, Inc.  Users have no right to privacy as to any
information transmitted or stored in, by or through any portion of this
system.
#####

Last login: Mon Aug 20 20:13:27 2012 from localhost
[hdfs@hadoop1 ~]$
```

3.3 Test for the 'hdfs' user on the master by using the hostname :

 NOTE: You may want to test and cache the 'fully qualified domain name' as well (i.e. hadoop1.savvis.lab)

```
[hdfs@hadoop1 ~]$ ssh hadoop1
The authenticity of host 'hadoop1 (10.12.48.50)' can't be established.
RSA key fingerprint is 81:7d:d6:30:1f:ab:b8:2f:e0:0b:20:75:e4:3f:47:16.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop1,10.12.48.50' (RSA) to the list of known
hosts.

#####
This computer system and associated networks are for the sole business
use of SAVVIS, Inc.  authorized users.  The company's computers and
proprietary data and information stored on them remain at all times the
property of SAVVIS, Inc.  Users have no right to privacy as to any
information transmitted or stored in, by or through any portion of this
system.
#####

Last login: Thu Aug 23 16:31:31 2012 from localhost
[hdfs@hadoop1 ~]$
```

3.4 Copy the public ssh key of the master to the authorized_keys file of the slave under the 'hdfs' user:

```
[hdfs@hadoop1 ~]$ ssh-copy-id -i $HOME/.ssh/id_dsa.pub hdfs@hadoop2

#####
This computer system and associated networks are for the sole business
use of SAVVIS, Inc.  authorized users.  The company's computers and
proprietary data and information stored on them remain at all times the
property of SAVVIS, Inc.  Users have no right to privacy as to any
information transmitted or stored in, by or through any portion of this
system.
#####

hdfs@hadoop2's password:  <--- Need to enter the 'local password' for the
'hdfs' user on 'hadoop2', this is NOT the SSH Passphrase.
Now try logging into the machine, with "ssh 'hdfs@hadoop2'", and check in:

    .ssh/authorized_keys

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

Test the ssh access from master to slave under the 'hdfs' user:

```
[hdfs@hadoop1 ~]$ ssh hdfs@hadoop2

#####
This computer system and associated networks are for the sole business
use of SAVVIS, Inc.  authorized users.  The company's computers and
proprietary data and information stored on them remain at all times the
property of SAVVIS, Inc.  Users have no right to privacy as to any
information transmitted or stored in, by or through any portion of this
system.
#####

Last login: Tue Aug 21 18:34:24 2012 from localhost
[hdfs@hadoop2 ~]$
```

3.5 Repeat from the master to the remaining 'slave' machines(hadoop3, and hadoop4) in the cluster:

From the hadoop master machine to hadoop3 :

```
[hdfs@hadoop1 ~]$ ssh-copy-id -i $HOME/.ssh/id_dsa.pub hduser@hadoop3
The authenticity of host 'hadoop3 (10.12.48.52)' can't be established.
RSA key fingerprint is 81:7d:d6:30:1f:ab:b8:2f:e0:0b:20:75:e4:3f:47:16.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoop3,10.12.48.52' (RSA) to the list of known
hosts.
```

```
#####
This computer system and associated networks are for the sole business
use of SAVVIS, Inc. authorized users. The company's computers and
proprietary data and information stored on them remain at all times the
property of SAVVIS, Inc. Users have no right to privacy as to any
information transmitted or stored in, by or through any portion of this
system.
#####
```

```
hduser@hadoop3's password: <--- user 'hdfs' local password on machine
hadoop3
Now try logging into the machine, with "ssh 'hdfs@hadoop3'", and check in:

    .ssh/authorized_keys

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

3.6 Test from master(hadoop1) to slave(hadoop3) machine:

```
[hdfs@hadoop1 ~]$ ssh hduser@hadoop3

#####
This computer system and associated networks are for the sole business
use of SAVVIS, Inc. authorized users. The company's computers and
proprietary data and information stored on them remain at all times the
property of SAVVIS, Inc. Users have no right to privacy as to any
information transmitted or stored in, by or through any portion of this
system.
#####

Last login: Tue Aug 21 18:34:54 2012 from localhost
[hdfs@hadoop3 ~]$
```

3.7 From the hadoop master machine to hadoop4 :


```

[hdfs@hadoop1 ~]$ ssh-copy-id -i $HOME/.ssh/id_dsa.pub hdfs@hadoop4

#####
This computer system and associated networks are for the sole business
use of SAVVIS, Inc.  authorized users.  The company's computers and
proprietary data and information stored on them remain at all times the
property of SAVVIS, Inc.  Users have no right to privacy as to any
information transmitted or stored in, by or through any portion of this
system.
#####

hdfs@hadoop4's password:
Now try logging into the machine, with "ssh 'hdfs@hadoop4'", and check in:

    .ssh/authorized_keys

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

[hdfs@hadoop1 ~]$

h4. 3.7 Test from master(hadoop1) to slave(hadoop4) machine:

[hdfs@hadoop1 ~]$ ssh hdfs@hadoop4

#####
This computer system and associated networks are for the sole business
use of SAVVIS, Inc.  authorized users.  The company's computers and
proprietary data and information stored on them remain at all times the
property of SAVVIS, Inc.  Users have no right to privacy as to any
information transmitted or stored in, by or through any portion of this
system.
#####

Last login: Thu Aug 23 16:49:22 2012 from hadoop1

```

4. Configure masters and the slaves within the cluster.

The master node(hadoop1) will run the "master" daemons for each layer: NameNode for the HDFS storage layer, and JobTracker for the MapReduce processing layer. Only the "slave" machines(hadoop2,hadoop3, hadoop4) will run the "slave" daemons: DataNode for the HDFS layer, and TaskTracker for MapReduce processing layer. Basically, the "master" daemons are responsible for coordination and management of the "slave" daemons while the latter will do the actual data storage and data processing work.

4.1 On the hadoop 'master', configure the 'masters' file. If you have more than one hadoop master, all of the machines hostnames will go here(configure all 'master' machines with the same "/etc/hadoop/masters" file):


```

[root@hadoop1 tmp]# cat "/etc/hadoop/masters"
hadoop1


```


4.2 On the hadoop master, configure the 'slaves' file. Place the hostname of all of the hadoop 'slaves' in the cluster into here.

```
[root@hadoop1 tmp]# cat "/etc/hadoop/slaves"  
hadoop2  
hadoop3  
hadoop4
```

 Note: The conf/slaves file on master is used only by the scripts like bin/start-dfs.sh or bin/stop-dfs.sh. For example, if you want to add DataNodes on the fly , you can "manually" start the DataNode daemon on a new slave machine via bin/hadoop-daemon.sh start datanode. Using the conf/slaves file on the master simply helps you to make "full" cluster restarts easier.

4.3 Configure *-site.xml files on ALL(all masters, and all slaves) machines within the cluster :

 Note: As of Hadoop 0.20.x and 1.x, the configuration settings previously found in hadoop-site.xml were moved to "/etc/hadoop/core-site.xml" (fs.default.name), "/etc/hadoop/mapred-site.xml" (mapred.job.tracker) and "/etc/hadoop/hdfs-site.xml" (dfs.replication).

 **WARNING** : Important: You have to change the following three configuration files on ALL machines:

```
[root@hadoop1 tmp]# find /etc -name "*-site.xml"  
/etc/hadoop/hdfs-site.xml  
/etc/hadoop/mapred-site.xml  
/etc/hadoop/core-site.xml
```

4.4 Edit the "/etc/hadoop/core-site.xml" file :

```
[root@hadoop1 tmp]# cat /etc/hadoop/core-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

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

<configuration>
<property>
  <name>fs.default.name</name>
  <value>hdfs://hadoop1:54310</value>
  <description>The name of the default file system. A URI whose
scheme and authority determine the FileSystem implementation. The
uri's scheme determines the config property (fs.SCHEME.impl) naming
the FileSystem implementation class. The uri's authority is used to
determine the host, port, etc. for a filesystem.</description>
</property>
</configuration>
```

4.5 Distribute this configuration file to all other hadoop machines in the cluster:

```
[root@hadoop1 tmp]# scp -p /etc/hadoop/core-site.xml
root@hadoop2:/etc/hadoop/
root@hadoop2's password:
core-site.xml
100% 609    0.6KB/s   00:00

[root@hadoop1 tmp]# scp -p /etc/hadoop/core-site.xml
root@hadoop3:/etc/hadoop/
root@hadoop3's password:
core-site.xml
100% 609    0.6KB/s   00:00

[root@hadoop1 tmp]# scp -p /etc/hadoop/core-site.xml
root@hadoop4:/etc/hadoop/
root@hadoop4's password:
core-site.xml
100% 609    0.6KB/s   00:00
```

4.6 Edit the "/etc/hadoop/core-site.xml" file :

```
[root@hadoop1 tmp]# cat /etc/hadoop/mapred-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

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

<configuration>
<property>
  <name>mapred.job.tracker</name>
  <value>hadoop1:54311</value>
  <description>The host and port that the MapReduce job tracker runs
  at.  If "local", then jobs are run in-process as a single map
  and reduce task.
  </description>
</property>
</configuration>
```

4.7 Distribute this configuration files to all other hadoop machines in the cluster:

```
[root@hadoop1 tmp]# scp -p /etc/hadoop/mapred-site.xml
root@hadoop2:/etc/hadoop/
root@hadoop2's password:
mapred-site.xml
100% 434    0.4KB/s   00:00

[root@hadoop1 tmp]# scp -p /etc/hadoop/mapred-site.xml
root@hadoop3:/etc/hadoop/
root@hadoop3's password:
mapred-site.xml
100% 434    0.4KB/s   00:00

[root@hadoop1 tmp]# scp -p /etc/hadoop/mapred-site.xml
root@hadoop4:/etc/hadoop/
root@hadoop4's password:
mapred-site.xml
100% 434    0.4KB/s   00:00
```

4.8 Edit the "/etc/hadoop/hdfs-site.xml" file :

```
[root@hadoop1 tmp]# cat /etc/hadoop/hdfs-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

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


<configuration>
<property>
  <name>dfs.replication</name>
  <value>3</value>
  <description>Default block replication.
  The actual number of replications can be specified when the file is
  created.
  The default is used if replication is not specified in create time.
  </description>
</property>
</configuration>

h4. 4.9 Distribute this configuration file to all other hadoop machines in
the cluster:

[root@hadoop1 tmp]# scp -p /etc/hadoop/hdfs-site.xml
root@hadoop2:/etc/hadoop/
root@hadoop2's password:
mapred-site.xml
100% 434    0.4KB/s   00:00

[root@hadoop1 tmp]# scp -p /etc/hadoop/hdfs-site.xml
root@hadoop3:/etc/hadoop/
root@hadoop3's password:
mapred-site.xml
100% 434    0.4KB/s   00:00

[root@hadoop1 tmp]# scp -p /etc/hadoop/hdfs-site.xml
root@hadoop4:/etc/hadoop/
root@hadoop4's password:
mapred-site.xml
100% 434    0.4KB/s   00:00
```


 **NOTE:** The "dfs.replication" value should never be higher than the number of slave nodes (more precisely, the number of DataNodes) that are have available.

5. Formatting the HDFS filesystem via the NameNode

```

[hdfs@hadoop1 ~]$ /usr/bin/hadoop namenode -format
12/08/23 19:44:23 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:  host = hadoop1/10.12.48.50
STARTUP_MSG:  args = [-format]
STARTUP_MSG:  version = 1.0.3
STARTUP_MSG:  build =
https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.0 -r 1335192;
compiled by 'hortonfo' on Tue May  8 20:35:39 UTC 2012
*****/
Re-format filesystem in /tmp/hadoop-hdfs/dfs/name ? (Y or N) Y
12/08/23 19:44:28 INFO util.GSet: VM type      = 64-bit
12/08/23 19:44:29 INFO util.GSet: 2% max memory = 19.7975 MB
12/08/23 19:44:29 INFO util.GSet: capacity    = 2^21 = 2097152 entries
12/08/23 19:44:29 INFO util.GSet: recommended=2097152, actual=2097152
12/08/23 19:44:31 INFO namenode.FSNamesystem: fsOwner=hdfs
12/08/23 19:44:31 INFO namenode.FSNamesystem: supergroup=supergroup
12/08/23 19:44:31 INFO namenode.FSNamesystem: isPermissionEnabled=true
12/08/23 19:44:31 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
12/08/23 19:44:31 INFO namenode.FSNamesystem: isAccessTokenEnabled=false
accessKeyUpdateInterval=0 min(s), accessTokenLifetime=0 min(s)
12/08/23 19:44:31 INFO namenode.NameNode: Caching file names occuring more than 10
times
12/08/23 19:44:33 INFO common.Storage: Image file of size 110 saved in 0 seconds.
12/08/23 19:44:33 INFO common.Storage: Storage directory /tmp/hadoop-hdfs/dfs/name
has been successfully formatted.
12/08/23 19:44:33 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at hadoop1/10.12.48.50
*****/

```

 Note: Background: The HDFS name table is stored on the NameNode's (here: master) local filesystem in the directory specified by dfs.name.dir. The name table is used by the NameNode to store tracking and coordination information for the DataNodes.

6. Starting the multi-node cluster

Starting the cluster is done in two steps.

6.1 HDFS daemons are started: the NameNode daemon is started on master(hadoop1), and DataNode daemons are started on all slaves (hadoop2, hadoop3, hadoop4).

6.2 MapReduce daemons are started: the JobTracker is started on master(hadoop1), and TaskTracker daemons are started on all slaves (hadoop2, hadoop3, hadoop4).

6.1.a Starting the NameNode daemon on the master(hadoop1):

```

[hdfs@hadoop1 ~]$ /usr/sbin/start-dfs.sh
starting namenode, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-namenode-hadoop1.out

```

```
hadoop3:
hadoop3:
#####
hadoop3: This computer system and associated networks are for the sole
business
hadoop3: use of SAVVIS, Inc. authorized users. The company's computers
and
hadoop3: proprietary data and information stored on them remain at all
times the
hadoop3: property of SAVVIS, Inc. Users have no right to privacy as to
any
hadoop3: information transmitted or stored in, by or through any portion
of this
hadoop3: system.
hadoop3:
#####
hadoop3:
hadoop2:
hadoop2:
#####
hadoop2: This computer system and associated networks are for the sole
business
hadoop2: use of SAVVIS, Inc. authorized users. The company's computers
and
hadoop2: proprietary data and information stored on them remain at all
times the
hadoop2: property of SAVVIS, Inc. Users have no right to privacy as to
any
hadoop2: information transmitted or stored in, by or through any portion
of this
hadoop2: system.
hadoop2:
#####
hadoop2:
hadoop4:
hadoop4:
#####
hadoop4: This computer system and associated networks are for the sole
business
hadoop4: use of SAVVIS, Inc. authorized users. The company's computers
and
hadoop4: proprietary data and information stored on them remain at all
times the
hadoop4: property of SAVVIS, Inc. Users have no right to privacy as to
any
hadoop4: information transmitted or stored in, by or through any portion
of this
hadoop4: system.
hadoop4:
hadoop4:
#####
hadoop4:
hadoop4: starting datanode, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-datanode-hadoop4.out
hadoop3: starting datanode, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-datanode-hadoop3.out
hadoop2: starting datanode, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-datanode-hadoop2.out
hadoop1:
hadoop1:
```

```
#####
hadoop1: This computer system and associated networks are for the sole
business
hadoop1: use of SAVVIS, Inc. authorized users. The company's computers
and
hadoop1: proprietary data and information stored on them remain at all
times the
hadoop1: property of SAVVIS, Inc. Users have no right to privacy as to
any
hadoop1: information transmitted or stored in, by or through any portion
of this
hadoop1: system.
hadoop1:
#####
hadoop1:
hadoop1: starting secondarynamenode, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-secondarynamenode-hadoop1.out
```



```
[hdfs@hadoop1 ~]$ ps -ef | grep -v grep | grep hadoop | wc -l
2
```

On the slave hadoop machines, inspect the following logs output for success or failure:

```
hadoop2:
[hdfs@hadoop2 ~]$ tail
/var/log/hadoop/hdfs/hadoop-hdfs-datanode-hadoop2.log
2012-08-23 20:17:11,462 INFO org.apache.hadoop.ipc.Server: IPC Server
handler 0 on 50020: starting
2012-08-23 20:17:11,463 INFO org.apache.hadoop.ipc.Server: IPC Server
handler 1 on 50020: starting
2012-08-23 20:17:11,463 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: using BLOCKREPORT_INTERVAL
of 3600000msec Initial delay: 0msec
2012-08-23 20:17:11,464 INFO org.apache.hadoop.ipc.Server: IPC Server
Responder: starting
2012-08-23 20:17:11,466 INFO org.apache.hadoop.ipc.Server: IPC Server
handler 2 on 50020: starting
2012-08-23 20:17:11,578 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: Reconciled asynchronous
block report against current state in 0 ms
2012-08-23 20:17:11,586 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: BlockReport of 0 blocks
took 0 msec to generate and 7 msecs for RPC and NN processing
2012-08-23 20:17:11,586 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: Starting Periodic block
scanner.
2012-08-23 20:17:11,587 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: Generated rough (lockless)
block report in 0 ms
2012-08-23 20:17:11,587 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: Reconciled asynchronous
block report against current state in 0 ms

hadoop3:
[hdfs@hadoop3 ~]$ tail
/var/log/hadoop/hdfs/hadoop-hdfs-datanode-hadoop3.log
2012-08-23 20:17:11,463 INFO org.apache.hadoop.ipc.Server: IPC Server
listener on 50020: starting
2012-08-23 20:17:11,463 INFO org.apache.hadoop.ipc.Server: IPC Server
handler 0 on 50020: starting
2012-08-23 20:17:11,464 INFO org.apache.hadoop.ipc.Server: IPC Server
handler 1 on 50020: starting
2012-08-23 20:17:11,464 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: using BLOCKREPORT_INTERVAL
of 3600000msec Initial delay: 0msec
2012-08-23 20:17:11,468 INFO org.apache.hadoop.ipc.Server: IPC Server
handler 2 on 50020: starting
2012-08-23 20:17:11,579 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: Reconciled asynchronous
block report against current state in 0 ms
2012-08-23 20:17:11,585 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: BlockReport of 0 blocks
took 1 msec to generate and 6 msecs for RPC and NN processing
2012-08-23 20:17:11,586 INFO
```

```
org.apache.hadoop.hdfs.server.datanode.DataNode: Starting Periodic block scanner.
2012-08-23 20:17:11,586 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: Generated rough (lockless) block report in 0 ms
2012-08-23 20:17:11,587 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: Reconciled asynchronous block report against current state in 0 ms


hadoop4:
[hdfs@hadoop4 ~]$ tail
/var/log/hadoop/hdfs/hadoop-hdfs-datanode-hadoop4.log
2012-08-23 20:17:11,465 INFO org.apache.hadoop.ipc.Server: IPC Server handler 2 on 50020: starting
2012-08-23 20:17:11,579 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: Reconciled asynchronous block report against current state in 0 ms
2012-08-23 20:17:11,585 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: BlockReport of 0 blocks took 0 msec to generate and 6 msec for RPC and NN processing
2012-08-23 20:17:11,586 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: Starting Periodic block scanner.
2012-08-23 20:17:11,587 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: Generated rough (lockless) block report in 0 ms
2012-08-23 20:17:11,587 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: Reconciled asynchronous block report against current state in 0 ms
2012-08-23 20:18:08,481 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: Starting asynchronous block report scan
2012-08-23 20:18:08,481 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: Finished asynchronous block report scan in 0ms
2012-08-23 20:18:08,482 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: Reconciled asynchronous block report against current state in 0 ms
2012-08-23 20:18:08,483 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode: BlockReport of 0 blocks
```

```
took 0 msec to generate and 1 msec for RPC and NN processing
```

Check the process on the master hadoop server (hadoop1) at this point:

```
[hdfs@hadoop1 ~]$ jps
23562 SecondaryNameNode
23683 Jps
23402 NameNode
```

6.1.b Starting the DataNode daemons on all slaves(hadoop2, hadoop3, hadoop4)

 Note: The command "/usr/sbin/start-dfs.sh" from the master server(hadoop1) above run previously should have already started the DataNode daemons on the slaves.

Check the process on the slave hadoop servers (hadoop2, hadoop3, hadoop4) at this point :

```
[hdfs@hadoop2 ~]$ jps
1596 Jps
1505 DataNode
```

```
[hdfs@hadoop3 ~]$ jps
14341 DataNode
14429 Jps
```

```
[hdfs@hadoop4 ~]$ jps
24766 DataNode
24855 Jps
```

6.2.a Starting the Mapreduce daemon(JobTracker) daemon on the master(hadoop1):

```
[hdfs@hadoop1 ~]$ start-mapred.sh
starting jobtracker, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-jobtracker-hadoop1.out
hadoop4:
hadoop4:
#####
hadoop4: This computer system and associated networks are for the sole
business
hadoop4: use of SAVVIS, Inc. authorized users. The company's computers
and
hadoop4: proprietary data and information stored on them remain at all
times the
hadoop4: property of SAVVIS, Inc. Users have no right to privacy as to
any
hadoop4: information transmitted or stored in, by or through any portion
of this
```

```
hadoop4: system.
hadoop4:
#####
hadoop4:
hadoop2:
hadoop2:
#####
hadoop2: This computer system and associated networks are for the sole
business
hadoop2: use of SAVVIS, Inc. authorized users. The company's computers
and
hadoop2: proprietary data and information stored on them remain at all
times the
hadoop2: property of SAVVIS, Inc. Users have no right to privacy as to
any
hadoop2: information transmitted or stored in, by or through any portion
of this
hadoop2: system.
hadoop2:
#####
hadoop2:
hadoop3:
hadoop3:
#####
hadoop3: This computer system and associated networks are for the sole
business
hadoop3: use of SAVVIS, Inc. authorized users. The company's computers
and
hadoop3: proprietary data and information stored on them remain at all
times the
hadoop3: property of SAVVIS, Inc. Users have no right to privacy as to
any
hadoop3: information transmitted or stored in, by or through any portion
of this
hadoop3: system.
hadoop3:
#####
hadoop3:
hadoop4: starting tasktracker, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-tasktracker-hadoop4.out
hadoop2: starting tasktracker, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-tasktracker-hadoop2.out
hadoop3: starting tasktracker, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-tasktracker-hadoop3.out
```

```
[hdfs@hadoop1 ~]$
```

On the slave, examine the following logs to determine success or failure:

```
hadoop2:
```

```
[hdfs@hadoop2 ~]$ tail
/var/log/hadoop/hdfs/hadoop-hdfs-tasktracker-hadoop2.log
2012-08-23 20:30:28,045 INFO org.apache.hadoop.mapred.TaskTracker: Using
ResourceCalculatorPlugin :
org.apache.hadoop.util.LinuxResourceCalculatorPlugin@1664023c
2012-08-23 20:30:28,048 WARN org.apache.hadoop.mapred.TaskTracker:
TaskTracker's totalMemoryAllottedForTasks is -1. TaskMemoryManager is
disabled.
2012-08-23 20:30:28,057 INFO org.apache.hadoop.mapred.IndexCache:
IndexCache created with max memory = 10485760
2012-08-23 20:30:28,268 INFO
org.apache.hadoop.metrics2.impl.MetricsSourceAdapter: MBean for source
ShuffleServerMetrics registered.
2012-08-23 20:30:28,280 INFO org.apache.hadoop.http.HttpServer: Port
returned by webServer.getConnectors()[0].getLocalPort() before open() is
-1. Opening the listener on 50060
2012-08-23 20:30:28,281 INFO org.apache.hadoop.http.HttpServer:
listener.getLocalPort() returned 50060
webServer.getConnectors()[0].getLocalPort() returned 50060
2012-08-23 20:30:28,281 INFO org.apache.hadoop.http.HttpServer: Jetty bound
to port 50060
2012-08-23 20:30:28,281 INFO org.mortbay.log: jetty-6.1.26
2012-08-23 20:30:30,410 INFO org.mortbay.log: Started
SelectChannelConnector@0.0.0.0:50060
2012-08-23 20:30:30,410 INFO org.apache.hadoop.mapred.TaskTracker:
FILE_CACHE_SIZE for mapOutputServlet set to : 2000
```

```
hadoop3:
```

```
[hdfs@hadoop3 ~]$ tail
/var/log/hadoop/hdfs/hadoop-hdfs-tasktracker-hadoop3.log
2012-08-23 20:30:27,096 INFO org.apache.hadoop.mapred.TaskTracker: Using
ResourceCalculatorPlugin :
org.apache.hadoop.util.LinuxResourceCalculatorPlugin@1664023c
2012-08-23 20:30:27,099 WARN org.apache.hadoop.mapred.TaskTracker:
TaskTracker's totalMemoryAllottedForTasks is -1. TaskMemoryManager is
disabled.
2012-08-23 20:30:27,103 INFO org.apache.hadoop.mapred.IndexCache:
IndexCache created with max memory = 10485760
2012-08-23 20:30:27,110 INFO
org.apache.hadoop.metrics2.impl.MetricsSourceAdapter: MBean for source
ShuffleServerMetrics registered.
2012-08-23 20:30:27,113 INFO org.apache.hadoop.http.HttpServer: Port
returned by webServer.getConnectors()[0].getLocalPort() before open() is
-1. Opening the listener on 50060
2012-08-23 20:30:27,113 INFO org.apache.hadoop.http.HttpServer:
listener.getLocalPort() returned 50060
webServer.getConnectors()[0].getLocalPort() returned 50060
2012-08-23 20:30:27,113 INFO org.apache.hadoop.http.HttpServer: Jetty bound
to port 50060
2012-08-23 20:30:27,113 INFO org.mortbay.log: jetty-6.1.26
```

```
2012-08-23 20:30:29,352 INFO org.mortbay.log: Started
SelectChannelConnector@0.0.0.0:50060
2012-08-23 20:30:29,352 INFO org.apache.hadoop.mapred.TaskTracker:
FILE_CACHE_SIZE for mapOutputServlet set to : 2000


hadoop4:
[hdfs@hadoop4 ~]$ tail
/var/log/hadoop/hdfs/hadoop-hdfs-tasktracker-hadoop4.log
2012-08-23 20:30:26,463 INFO org.apache.hadoop.mapred.TaskTracker: Using
ResourceCalculatorPlugin :
org.apache.hadoop.util.LinuxResourceCalculatorPlugin@51d8d39f
2012-08-23 20:30:26,465 WARN org.apache.hadoop.mapred.TaskTracker:
TaskTracker's totalMemoryAllottedForTasks is -1. TaskMemoryManager is
disabled.
2012-08-23 20:30:26,468 INFO org.apache.hadoop.mapred.IndexCache:
IndexCache created with max memory = 10485760
2012-08-23 20:30:26,484 INFO
org.apache.hadoop.metrics2.impl.MetricsSourceAdapter: MBean for source
ShuffleServerMetrics registered.
2012-08-23 20:30:26,487 INFO org.apache.hadoop.http.HttpServer: Port
returned by webServer.getConnectors()[0].getLocalPort() before open() is
-1. Opening the listener on 50060
2012-08-23 20:30:26,487 INFO org.apache.hadoop.http.HttpServer:
listener.getLocalPort() returned 50060
webServer.getConnectors()[0].getLocalPort() returned 50060
2012-08-23 20:30:26,487 INFO org.apache.hadoop.http.HttpServer: Jetty bound
to port 50060
2012-08-23 20:30:26,487 INFO org.mortbay.log: jetty-6.1.26
2012-08-23 20:30:27,204 INFO org.mortbay.log: Started
SelectChannelConnector@0.0.0.0:50060
2012-08-23 20:30:27,204 INFO org.apache.hadoop.mapred.TaskTracker:
```

```
FILE_CACHE_SIZE for mapOutputServlet set to : 2000
```

Check the process on the master hadoop server (hadoop1) at this point(JobTracker should be running):

```
[hdfs@hadoop1 ~]$ jps
23780 JobTracker
23934 Jps
23562 SecondaryNameNode
23402 NameNode
```

6.2.b Starting the Mapreduce daemons(TaskTracker) daemon on all slaves(hadoop2,hadoop3,hadoop4):

 Note: The command "/usr/sbin/start-mapred.sh" from the master server(hadoop1) above run previously should have already started the TaskTracker daemons on the slaves.

Check the process on the slave hadoop servers (hadoop2, hadoop3, hadoop4) at this point(TaskTracker should be running) :

```
hadoop2:
[hdfs@hadoop2 ~]$ jps
1802 Jps
1505 DataNode
1689 TaskTracker

hadoop3:
[hdfs@hadoop3 ~]$ jps
14520 TaskTracker
14341 DataNode
14637 Jps

hadoop4:
[hdfs@hadoop4 ~]$ jps
24766 DataNode
24945 TaskTracker
25064 Jps
```

7. Use the Web GUI tool on the hadoop master:

7.1 The NameNode web UI shows you a cluster summary including information about total/remaining capacity, live and dead nodes. Additionally, it allows you to browse the HDFS namespace and view the contents of its files in the web browser. It also gives access to the "local machine's" Hadoop log files.

NameNode 'hadoop1:54310'

Started: Thu Aug 23 20:17:04 UTC 2012
 Version: 1.0.3, r1335192
 Compiled: Tue May 8 20:35:39 UTC 2012 by hortonfo
 Upgrades: There are no upgrades in progress.

[Browse the filesystem](#)
[Namenode Logs](#)

Cluster Summary

6 files and directories, 1 blocks = 7 total. Heap Size is 28.66 MB / 989.88 MB (2%)

Configured Capacity	: 5.91 GB
DFS Used	: 120 KB
Non DFS Used	: 508.66 MB
DFS Remaining	: 5.41 GB
DFS Used%	: 0 %
DFS Remaining%	: 91.59 %
Live Nodes	: 3
Dead Nodes	: 0
Decommissioning Nodes	: 0
Number of Under-Replicated Blocks	: 0

NameNode Storage:

Storage Directory	Type	State

7.2 Click the link entitled "Live Nodes" to see the Hadoop DataNodes (slaves) status :

NameNode 'hadoop1:54310'

Started: Thu Aug 23 20:17:04 UTC 2012
 Version: 1.0.3, r1335192
 Compiled: Tue May 8 20:35:39 UTC 2012 by hortonfo
 Upgrades: There are no upgrades in progress.

[Browse the filesystem](#)
[Namenode Logs](#)
[Go back to DFS home](#)

Live Datanodes : 3

Node	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks
hadoop2	0	In Service	1.97	0	0.17	1.8	0		91.59	1
hadoop3	0	In Service	1.97	0	0.17	1.8	0		91.59	1
hadoop4	1	In Service	1.97	0	0.17	1.8	0		91.59	1

This is [Apache Hadoop](#) release 1.0.3

8. Stopping the multi-node cluster:

Stop the HDFS daemons on the master (hadoop1) :

There is an order to stopping the Hadoop HDFS cluster, it is the reverse order of the start with aptly named scripts:

8.1 First, stop the MapReduce daemons: the JobTracker is stopped on master (hadoop1), and TaskTracker daemons are stopped on all slaves (hadoop2,hadoop3,hadoop4)

Run the following stop script for the MapReduce daemon as the 'hdfs' user on the hadoop master(hadoop1) :

```
[hdfs@hadoop1 ~]$ stop-mapred.sh
stopping jobtracker
hadoop4:
hadoop4: #####
hadoop4: This computer system and associated networks are for the sole business
hadoop4: use of SAVVIS, Inc. authorized users. The company's computers and
hadoop4: proprietary data and information stored on them remain at all times the
hadoop4: property of SAVVIS, Inc. Users have no right to privacy as to any
hadoop4: information transmitted or stored in, by or through any portion of this
hadoop4: system.
hadoop4: #####
hadoop4:
hadoop2:
hadoop2: #####
hadoop2: This computer system and associated networks are for the sole business
hadoop2: use of SAVVIS, Inc. authorized users. The company's computers and
hadoop2: proprietary data and information stored on them remain at all times the
hadoop2: property of SAVVIS, Inc. Users have no right to privacy as to any
hadoop2: information transmitted or stored in, by or through any portion of this
hadoop2: system.
hadoop2: #####
hadoop2:
hadoop3:
hadoop3: #####
hadoop3: This computer system and associated networks are for the sole business
hadoop3: use of SAVVIS, Inc. authorized users. The company's computers and
hadoop3: proprietary data and information stored on them remain at all times the
hadoop3: property of SAVVIS, Inc. Users have no right to privacy as to any
hadoop3: information transmitted or stored in, by or through any portion of this
hadoop3: system.
hadoop3: #####
hadoop3:
hadoop4: stopping tasktracker
hadoop2: stopping tasktracker
hadoop3: stopping tasktracker
```

Checking the three DataNodes,or Slave tasktracker logs we can see a graceful shutdown :

```
hadoop2:
[hdfs@hadoop2 ~]$ tail /var/log/hadoop/hdfs/hadoop-hdfs-tasktracker-hadoop2.log
2012-08-24 18:33:13,404 INFO org.apache.hadoop.mapred.TaskTracker: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down TaskTracker at hadoop2/10.12.48.51
*****/
```

```
hadoop2:
[hdfs@hadoop3 ~]$ tail /var/log/hadoop/hdfs/hadoop-hdfs-tasktracker-hadoop3.log
2012-08-24 18:33:13,398 INFO org.apache.hadoop.mapred.TaskTracker: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down TaskTracker at hadoop3/10.12.48.52
*****/
```

```
hadoop4:
[hdfs@hadoop4 ~]$ tail /var/log/hadoop/hdfs/hadoop-hdfs-tasktracker-hadoop4.log
2012-08-24 18:33:13,334 INFO org.apache.hadoop.mapred.TaskTracker: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down TaskTracker at hadoop4/10.12.48.54
*****/
```

8.2 Second, HDFS daemons are stopped: the NameNode daemon is stopped on master (hadoop1), and DataNode daemons are stopped on all slaves (hadoop2,hadoop3,hadoop4).

Run the following stop script for the HDFS daemon as the 'hdfs' user on the hadoop master(hadoop1) :

```

[hdfs@hadoop1 ~]$ stop-dfs.sh
stopping namenode
hadoop3:
hadoop3: #####
hadoop3: This computer system and associated networks are for the sole business
hadoop3: use of SAVVIS, Inc. authorized users. The company's computers and
hadoop3: proprietary data and information stored on them remain at all times the
hadoop3: property of SAVVIS, Inc. Users have no right to privacy as to any
hadoop3: information transmitted or stored in, by or through any portion of this
hadoop3: system.
hadoop3: #####
hadoop3:
hadoop4:
hadoop4: #####
hadoop4: This computer system and associated networks are for the sole business
hadoop4: use of SAVVIS, Inc. authorized users. The company's computers and
hadoop4: proprietary data and information stored on them remain at all times the
hadoop4: property of SAVVIS, Inc. Users have no right to privacy as to any
hadoop4: information transmitted or stored in, by or through any portion of this
hadoop4: system.
hadoop4: #####
hadoop4:
hadoop2:
hadoop2: #####
hadoop2: This computer system and associated networks are for the sole business
hadoop2: use of SAVVIS, Inc. authorized users. The company's computers and
hadoop2: proprietary data and information stored on them remain at all times the
hadoop2: property of SAVVIS, Inc. Users have no right to privacy as to any
hadoop2: information transmitted or stored in, by or through any portion of this
hadoop2: system.
hadoop2: #####
hadoop2:
hadoop4: stopping datanode
hadoop3: stopping datanode
hadoop2: stopping datanode
hadoop1:
hadoop1: #####
hadoop1: This computer system and associated networks are for the sole business
hadoop1: use of SAVVIS, Inc. authorized users. The company's computers and
hadoop1: proprietary data and information stored on them remain at all times the
hadoop1: property of SAVVIS, Inc. Users have no right to privacy as to any
hadoop1: information transmitted or stored in, by or through any portion of this
hadoop1: system.
hadoop1: #####
hadoop1:
hadoop1: stopping secondarynamenode

```

Checking the three DataNodes, or Slave datanode logs we can see a graceful shutdown :

```

hadoop2:
[hdfs@hadoop2 ~]$ tail /var/log/hadoop/hdfs/hadoop-hdfs-datanode-hadoop2.log
2012-08-24 17:16:30,883 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Reconciled asynchronous block report against current state in 0 ms
2012-08-24 17:16:30,884 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:

```

```
BlockReport of 1 blocks took 0 msec to generate and 1 msec for RPC and NN
processing
2012-08-24 18:16:29,082 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Starting asynchronous block report scan
2012-08-24 18:16:29,082 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Finished asynchronous block report scan in 0ms
2012-08-24 18:16:29,083 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Reconciled asynchronous block report against current state in 0 ms
2012-08-24 18:16:29,084 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
BlockReport of 1 blocks took 0 msec to generate and 1 msec for RPC and NN
processing
2012-08-24 18:36:11,280 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down DataNode at hadoop2/10.12.48.51
*****/
```

hadoop3:

```
[hdfs@hadoop3 ~]$ tail /var/log/hadoop/hdfs/hadoop-hdfs-datanode-hadoop3.log
2012-08-24 16:54:48,460 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Reconciled asynchronous block report against current state in 1 ms
2012-08-24 16:54:48,461 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
BlockReport of 1 blocks took 1 msec to generate and 1 msec for RPC and NN
processing
2012-08-24 17:54:46,663 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Starting asynchronous block report scan
2012-08-24 17:54:46,664 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Finished asynchronous block report scan in 1ms
2012-08-24 17:54:46,664 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Reconciled asynchronous block report against current state in 0 ms
2012-08-24 17:54:46,665 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
BlockReport of 1 blocks took 0 msec to generate and 1 msec for RPC and NN
processing
2012-08-24 18:36:11,244 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down DataNode at hadoop3/10.12.48.52
*****/
```

hadoop4:

```
[hdfs@hadoop4 ~]$ tail /var/log/hadoop/hdfs/hadoop-hdfs-datanode-hadoop4.log
2012-08-24 17:18:08,630 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Reconciled asynchronous block report against current state in 0 ms
2012-08-24 17:18:08,631 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
BlockReport of 1 blocks took 0 msec to generate and 1 msec for RPC and NN
processing
2012-08-24 18:18:09,829 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Starting asynchronous block report scan
2012-08-24 18:18:09,829 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Finished asynchronous block report scan in 0ms
2012-08-24 18:18:09,830 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Reconciled asynchronous block report against current state in 0 ms
2012-08-24 18:18:09,831 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
BlockReport of 1 blocks took 0 msec to generate and 1 msec for RPC and NN
processing
2012-08-24 18:36:11,198 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
SHUTDOWN_MSG:
/*****
```

```
SHUTDOWN_MSG: Shutting down DataNode at hadoop4/10.12.48.54
*****/
```

Lastly, if we run the jps command on all nodes, there are no more hadoop processes shown running:

```
[hdfs@hadoop1 ~]$ jps
32330 Jps

[hdfs@hadoop2 ~]$ jps
10148 Jps

[hdfs@hadoop3 ~]$ jps
22963 Jps

[hdfs@hadoop4 ~]$ jps
950 Jps
```

9. Running a MapReduce job:

9.1 Start the Hadoop Cluster from the master (hadoop1) :

```
[hdfs@hadoop1 ~]$ start-dfs.sh
hadoop4: starting datanode, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-datanode-hadoop4.out
hadoop2: starting datanode, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-datanode-hadoop2.out
hadoop3: starting datanode, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-datanode-hadoop3.out

[hdfs@hadoop1 ~]$ start-mapred.sh
hadoop3: starting tasktracker, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-tasktracker-hadoop3.out
hadoop2: starting tasktracker, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-tasktracker-hadoop2.out
hadoop4: starting tasktracker, logging to
/var/log/hadoop/hdfs/hadoop-hdfs-tasktracker-hadoop4.out
```

9.2 Copy the data input files to the master(hadoop1) to a local directory:

```
[4][rhill@sl7jump1 hadoop]$ scp -p *.txt root@hadoop1:/data01/gutenberg
```

```
root@hadoop1's password:
```

```
pg132.txt  
100% 336KB 0.0KB/s 00:00  
pg1661.txt  
100% 581KB 0.0KB/s 00:00  
pg19699.txt  
100% 1900KB 1.9MB/s 00:01  
pg20417.txt  
100% 659KB 0.0KB/s 00:00  
pg4300.txt  
100% 1536KB 0.0KB/s 00:00  
pg5000.txt  
100% 1390KB 1.4MB/s 00:01  
pg972.txt  
100% 385KB 0.0KB/s 00:00
```

9.3 Ensure that the hadoop user (hdfs) has ownership of the data:

```
[root@hadoop1 ~]# chown -R hdfs:hadoop /data01/gutenberg
```

9.4 Copy local example data to HDFS :

```
[hdfs@hadoop1 ~]$ /usr/bin/hadoop dfs -copyFromLocal /data01/gutenberg/  
/usr/share/hadoop/gutenberg
```

9.5 Verify the data in the HDFS :

```
[hdfs@hadoop1 ~]$ /usr/bin/hadoop dfs -ls /usr/share/hadoop/gutenberg
Found 7 items
-rw-r--r--  3 hdfs supergroup    343692 2012-08-28 16:15
/usr/share/hadoop/gutenberg/pg132.txt
-rw-r--r--  3 hdfs supergroup    594933 2012-08-28 16:15
/usr/share/hadoop/gutenberg/pg1661.txt
-rw-r--r--  3 hdfs supergroup    1945886 2012-08-28 16:15
/usr/share/hadoop/gutenberg/pg19699.txt
-rw-r--r--  3 hdfs supergroup     674566 2012-08-28 16:15
/usr/share/hadoop/gutenberg/pg20417.txt
-rw-r--r--  3 hdfs supergroup    1573150 2012-08-28 16:14
/usr/share/hadoop/gutenberg/pg4300.txt
-rw-r--r--  3 hdfs supergroup    1423801 2012-08-28 16:15
/usr/share/hadoop/gutenberg/pg5000.txt
-rw-r--r--  3 hdfs supergroup     393968 2012-08-28 16:15
/usr/share/hadoop/gutenberg/pg972.txt
```

9.6 Now, we actually run the WordCount example job.(A MapReduce job):

```
[hdfs@hadoop1 ~]$ cd /usr/share/hadoop/

[hdfs@hadoop1 hadoop]$ ls -lrt hadoop-examples-1.0.3.jar
-rw-r--r--. 1 hdfs hadoop 142452 May  8 20:37 hadoop-examples-1.0.3.jar

[hdfs@hadoop1 hadoop]$ /usr/bin/hadoop jar hadoop-examples-1.0.3.jar wordcount
/usr/share/hadoop/gutenberg /usr/share/hadoop/gutenberg-output
12/08/28 16:24:01 INFO input.FileInputFormat: Total input paths to process : 7
12/08/28 16:24:01 INFO util.NativeCodeLoader: Loaded the native-hadoop library
12/08/28 16:24:01 WARN snappy.LoadSnappy: Snappy native library not loaded
12/08/28 16:24:02 INFO mapred.JobClient: Running job: job_201208281536_0001
12/08/28 16:24:03 INFO mapred.JobClient:  map 0% reduce 0%
12/08/28 16:24:26 INFO mapred.JobClient:  map 28% reduce 0%
```

9.7 Monitor the logs on the hadoop slaves :

```

[hdfs@hadoop2 ~]$ tail /var/log/hadoop/hdfs/hadoop-hdfs-datanode-hadoop2.log
2012-08-28 16:24:02,966 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
PacketResponder 0 for block blk_-5617381391876833085_1016 terminating
2012-08-28 16:24:03,003 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src:
/10.12.48.51:50010, dest: /10.12.48.50:51970, bytes: 234, op: HDFS_READ, cliID:
DFSCClient_-655313188, offset: 0, srvID:
DS-127717122-10.12.48.51-50010-1345753031355, blockid: blk_3529927905714908934_1012,
duration: 16124009
2012-08-28 16:24:14,036 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src:
/10.12.48.51:50010, dest: /10.12.48.51:59638, bytes: 110, op: HDFS_READ, cliID:
DFSCClient_201205399, offset: 0, srvID: DS-127717122-10.12.48.51-50010-1345753031355,
blockid: blk_-5617381391876833085_1016, duration: 233823
2012-08-28 16:24:14,076 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src:
/10.12.48.51:50010, dest: /10.12.48.51:59640, bytes: 20481, op: HDFS_READ, cliID:
DFSCClient_-860271374, offset: 0, srvID:
DS-127717122-10.12.48.51-50010-1345753031355, blockid: blk_4011335527832964217_1013,
duration: 450326
2012-08-28 16:24:14,654 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src:
/10.12.48.51:50010, dest: /10.12.48.51:59641, bytes: 143568, op: HDFS_READ, cliID:
DFSCClient_-860271374, offset: 0, srvID:
DS-127717122-10.12.48.51-50010-1345753031355, blockid:
blk_-2732319372433467617_1010, duration: 2896946
2012-08-28 16:24:25,674 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src:
/10.12.48.51:50010, dest: /10.12.48.51:59646, bytes: 876, op: HDFS_READ, cliID:
DFSCClient_attempt_201208281536_0001_m_000004_0, offset: 0, srvID:
DS-127717122-10.12.48.51-50010-1345753031355, blockid: blk_-878023163961426070_1011,
duration: 1967520
2012-08-28 16:24:26,305 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src:
/10.12.48.51:50010, dest: /10.12.48.51:59647, bytes: 360, op: HDFS_READ, cliID:
DFSCClient_attempt_201208281536_0001_m_000005_0, offset: 512, srvID:
DS-127717122-10.12.48.51-50010-1345753031355, blockid: blk_-878023163961426070_1011,
duration: 531060
2012-08-28 16:24:30,973 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src:
/10.12.48.51:50010, dest: /10.12.48.51:59648, bytes: 599581, op: HDFS_READ, cliID:
DFSCClient_attempt_201208281536_0001_m_000004_0, offset: 0, srvID:
DS-127717122-10.12.48.51-50010-1345753031355, blockid: blk_-451990966209824734_1004,
duration: 2304882879
2012-08-28 16:24:32,654 INFO
org.apache.hadoop.hdfs.server.datanode.DataNode.clienttrace: src:
/10.12.48.51:50010, dest: /10.12.48.51:59649, bytes: 397048, op: HDFS_READ, cliID:
DFSCClient_attempt_201208281536_0001_m_000005_0, offset: 0, srvID:
DS-127717122-10.12.48.51-50010-1345753031355, blockid:
blk_-8119426451923107821_1008, duration: 2973806826
2012-08-28 16:24:56,798 INFO
org.apache.hadoop.hdfs.server.datanode.DataBlockScanner: Verification succeeded for
blk_-2632613381915044023_1007

```

9.8 The MapReduce job should complete shortly with output like below :


```

[hdfs@hadoop1 ~]$ cd /usr/share/hadoop/
[hdfs@hadoop1 hadoop]$ /usr/bin/hadoop jar hadoop-examples-1.0.3.jar wordcount
/usr/share/hadoop/gutenberg /usr/share/hadoop/gutenberg-out
12/08/28 21:41:50 INFO input.FileInputFormat: Total input paths to process : 7
12/08/28 21:41:50 INFO util.NativeCodeLoader: Loaded the native-hadoop library
12/08/28 21:41:50 WARN snappy.LoadSnappy: Snappy native library not loaded
12/08/28 21:41:51 INFO mapred.JobClient: Running job: job_201208282021_0004
12/08/28 21:41:52 INFO mapred.JobClient: map 0% reduce 0%
12/08/28 21:42:14 INFO mapred.JobClient: map 25% reduce 0%
12/08/28 21:42:17 INFO mapred.JobClient: map 42% reduce 0%
12/08/28 21:42:20 INFO mapred.JobClient: map 85% reduce 0%
12/08/28 21:42:32 INFO mapred.JobClient: map 100% reduce 0%
12/08/28 21:42:44 INFO mapred.JobClient: map 100% reduce 100%
12/08/28 21:42:49 INFO mapred.JobClient: Job complete: job_201208282021_0004
12/08/28 21:42:49 INFO mapred.JobClient: Counters: 29
12/08/28 21:42:49 INFO mapred.JobClient: Job Counters
12/08/28 21:42:49 INFO mapred.JobClient: Launched reduce tasks=1
12/08/28 21:42:49 INFO mapred.JobClient: SLOTS_MILLIS_MAPS=138302
12/08/28 21:42:49 INFO mapred.JobClient: Total time spent by all reduces waiting
after reserving slots (ms)=0
12/08/28 21:42:49 INFO mapred.JobClient: Total time spent by all maps waiting
after reserving slots (ms)=0
12/08/28 21:42:49 INFO mapred.JobClient: Launched map tasks=7
12/08/28 21:42:49 INFO mapred.JobClient: Data-local map tasks=7
12/08/28 21:42:49 INFO mapred.JobClient: SLOTS_MILLIS_REDUCE=21741
12/08/28 21:42:49 INFO mapred.JobClient: File Output Format Counters
12/08/28 21:42:49 INFO mapred.JobClient: Bytes Written=1412493
12/08/28 21:42:49 INFO mapred.JobClient: FileSystemCounters
12/08/28 21:42:49 INFO mapred.JobClient: FILE_BYTES_READ=4462538
12/08/28 21:42:49 INFO mapred.JobClient: HDFS_BYTES_READ=6950857
12/08/28 21:42:49 INFO mapred.JobClient: FILE_BYTES_WRITTEN=7550301
12/08/28 21:42:49 INFO mapred.JobClient: HDFS_BYTES_WRITTEN=1412493
12/08/28 21:42:49 INFO mapred.JobClient: File Input Format Counters
12/08/28 21:42:49 INFO mapred.JobClient: Bytes Read=6949996
12/08/28 21:42:49 INFO mapred.JobClient: Map-Reduce Framework
12/08/28 21:42:49 INFO mapred.JobClient: Map output materialized bytes=2915042
12/08/28 21:42:49 INFO mapred.JobClient: Map input records=137147
12/08/28 21:42:49 INFO mapred.JobClient: Reduce shuffle bytes=2915042
12/08/28 21:42:49 INFO mapred.JobClient: Spilled Records=507854
12/08/28 21:42:49 INFO mapred.JobClient: Map output bytes=11435844
12/08/28 21:42:49 INFO mapred.JobClient: CPU time spent (ms)=31060
12/08/28 21:42:49 INFO mapred.JobClient: Total committed heap usage
(bytes)=1053700096
12/08/28 21:42:49 INFO mapred.JobClient: Combine input records=1174991
12/08/28 21:42:49 INFO mapred.JobClient: SPLIT_RAW_BYTES=861
12/08/28 21:42:49 INFO mapred.JobClient: Reduce input records=201008
12/08/28 21:42:49 INFO mapred.JobClient: Reduce input groups=128512
12/08/28 21:42:49 INFO mapred.JobClient: Combine output records=201008
12/08/28 21:42:49 INFO mapred.JobClient: Physical memory (bytes)
snapshot=1366478848
12/08/28 21:42:49 INFO mapred.JobClient: Reduce output records=128512
12/08/28 21:42:49 INFO mapred.JobClient: Virtual memory (bytes)
snapshot=8445665280
12/08/28 21:42:49 INFO mapred.JobClient: Map output records=1174991

```

9.9 Now check the 'datanode' logs on the slave machines for completion:

```
[hdfs@hadoop4 hdfs]$ tail /var/log/hadoop/hdfs/hadoop-hdfs-datanode-hadoop4.log
2012-08-28 21:42:49,819 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Scheduling block blk_-6501103866201620254_1018 file
/tmp/hadoop-hdfs/dfs/data/current/blk_-6501103866201620254 for deletion
2012-08-28 21:42:49,820 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Scheduling block blk_-4181631826785300398_1012 file
/tmp/hadoop-hdfs/dfs/data/current/blk_-4181631826785300398 for deletion
2012-08-28 21:42:49,820 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Scheduling block blk_-3339764523210320144_1014 file
/tmp/hadoop-hdfs/dfs/data/current/blk_-3339764523210320144 for deletion
2012-08-28 21:42:49,820 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Scheduling block blk_3712644856305112385_1015 file
/tmp/hadoop-hdfs/dfs/data/current/blk_3712644856305112385 for deletion
2012-08-28 21:42:49,821 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Scheduling block blk_7707301092082061852_1013 file
/tmp/hadoop-hdfs/dfs/data/current/blk_7707301092082061852 for deletion
2012-08-28 21:42:49,821 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Deleted block blk_-6501103866201620254_1018 at file
/tmp/hadoop-hdfs/dfs/data/current/blk_-6501103866201620254
2012-08-28 21:42:49,821 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Deleted block blk_-4181631826785300398_1012 at file
/tmp/hadoop-hdfs/dfs/data/current/blk_-4181631826785300398
2012-08-28 21:42:49,821 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Deleted block blk_-3339764523210320144_1014 at file
/tmp/hadoop-hdfs/dfs/data/current/blk_-3339764523210320144
2012-08-28 21:42:49,821 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Deleted block blk_3712644856305112385_1015 at file
/tmp/hadoop-hdfs/dfs/data/current/blk_3712644856305112385
2012-08-28 21:42:49,821 INFO org.apache.hadoop.hdfs.server.datanode.DataNode:
Deleted block blk_7707301092082061852_1013 at file
/tmp/hadoop-hdfs/dfs/data/current/blk_7707301092082061852
```

9.10 Check if the result is successfully stored in HDFS directory

```
[hdfs@hadoop1 hadoop]$ /usr/bin/hadoop dfs -ls /usr/share/hadoop
Found 2 items
drwxr-xr-x - hdfs supergroup          0 2012-08-28 21:39
/usr/share/hadoop/gutenberg
drwxr-xr-x - hdfs supergroup          0 2012-08-28 21:42
/usr/share/hadoop/gutenberg-out

[hdfs@hadoop1 hadoop]$ /usr/bin/hadoop dfs -ls /usr/share/hadoop/gutenberg-out
Found 3 items
-rw-r--r--  3 hdfs supergroup          0 2012-08-28 21:42
/usr/share/hadoop/gutenberg-out/_SUCCESS
drwxr-xr-x - hdfs supergroup          0 2012-08-28 21:41
/usr/share/hadoop/gutenberg-out/_logs
-rw-r--r--  3 hdfs supergroup 1412493 2012-08-28 21:42
/usr/share/hadoop/gutenberg-out/part-r-00000
```

9.11 Retrieve the job result from HDFS to the local filesystem:

```
[hdfs@hadoop1 ~]$ /usr/bin/hadoop dfs -getmerge /usr/share/hadoop/gutenberg-out
/tmp/gutenberg-out
12/08/28 22:00:59 INFO util.NativeCodeLoader: Loaded the native-hadoop library

[hdfs@hadoop1 ~]$ head /tmp/gutenberg-out/gutenberg-out
"          34
"'A          1
"'About      1
"'Absolute    1
"'Ah!'        2
"'Ah,         2
"'Ample.'     1
"'And         10
"'Arthur!'    1
"'As          1
```

 NOTE: Alternatively, you can use the command to inspect the job:

```
[hdfs@hadoop1 ~]$ /usr/bin/hadoop dfs -cat
/usr/share/hadoop/gutenberg-out/part-r-00000|head
"          34
"'A          1
"'About      1
"'Absolute    1
"'Ah!'        2
"'Ah,         2
"'Ample.'     1
"'And         10
"'Arthur!'    1
"'As          1
cat: Unable to write to output stream.
```

10. Caveats and Known Problems:

10.1 Hadoop Troubleshooting Table :

Problem Description or Error	Resolution
java.io.IOException: Incompatible namespaceIDs	----- Work around 1. ----- Stop the hadoop cluster :

```
[hdfs@hadoop1 ~]$ stop-mapred.sh
[hdfs@hadoop1 ~]$ stop-dfs.sh
[hdfs@hadoop1 ~]$ ps -ef | grep
hadoop
```

Delete the previous output :

```
[hdfs@hadoop1 ~]$ cd
/usr/share/hadoop/
[hdfs@hadoop1 ~]$ /usr/bin/hadoop
dfs -rmr
/usr/share/hadoop/gutenberg-output
```

Delete the hadoop temporary files on ALL hadoop nodes:

```
[hdfs@hadoop1 ~]$ cd /tmp
[hdfs@hadoop1 ~]$ ls -rlt
[hdfs@hadoop1 ~]$ rm -rf Jetty*
hadoop* hsuperfdata*
```

Start the Hadoop cluster :

```
[hdfs@hadoop1 ~]$ start-dfs.sh
[hdfs@hadoop1 ~]$ start-mapred.sh
```

Work around 2.

Stop the hadoop cluster :

```
[hdfs@hadoop1 ~]$ stop-mapred.sh
[hdfs@hadoop1 ~]$ stop-dfs.sh
[hdfs@hadoop1 ~]$ ps -ef | grep
hadoop
```

Delete the previous output :

```
[hdfs@hadoop1 ~]$ cd
/usr/share/hadoop/
[hdfs@hadoop1 ~]$ /usr/bin/hadoop
dfs -rmr
/usr/share/hadoop/gutenberg-output
```

Delete the hadoop temporary files on ALL hadoop nodes:

```
[hdfs@hadoop1 ~]$ cd /tmp
[hdfs@hadoop1 ~]$ ls -rlt
[hdfs@hadoop1 ~]$ rm -rf Jetty*
hadoop* hspcrfdata*
```

Reformat the HDFS : (WARNING !!! All previous HDFS data will be lost) :

```
[hdfs@hadoop1 ~]$ /usr/bin/hadoop
namenode -format
```

Start the Hadoop cluster :

```
[hdfs@hadoop1 ~]$ start-dfs.sh
[hdfs@hadoop1 ~]$ start-mapred.sh
```

Work around 3.

1. Stop the DataNode

```
[hdfs@hadoop1 ~]$
/usr/sbin/stop-dfs.sh
```

2. Edit the value of namespaceID in /current/VERSION to match the value of the current NameNode
On all hadoop nodes :

```
[hdfs@hadoop1 ~]$ find
/usr/share/hadoop/ -name
"VERSION"
/usr/share/hadoop/contrib/hod/bin/
VERSION
```

```
[hdfs@hadoop1 ~]$ cat
/usr/share/hadoop/contrib/hod/bin/
VERSION
1.0.3
```

3. Restart the DataNode

```
[hdfs@hadoop1 ~]$
/usr/sbin/start-dfs.sh
```

10.2 Other resources for hadoop troubleshooting :

[Hadoop Troubleshooting](#)