Configure HDFS

	NameNode	SecundaryNamenode	DataNode	JobTracker	TaskTracker
elephant	x		x		x
tiger		x	x		x
horse			x	x	x
monkey			x		x

1. Uninstall packages from **elephant**

- $ sudo yum remove -y hadoop-hdfs-secondarynamenode hadoop-0.20-mapreduce-jobtracker

2. Uninstall packages from **tiger**
   $ sudo yum remove -y hadoop-hdfs-namenode hadoop-0.20-mapreduce-jobtracker

3. Uninstall packages from **horse**
   $ sudo yum remove -y hadoop-hdfs-namenode hadoop-hdfs-secondarynamenode

4. Uninstall packages from **monkey**
   $ sudo yum remove -y hadoop-hdfs-datanode hadoop-hdfs-secondarynamenode hadoop-0.20-mapreduce-jobtracker

5. Stop process on **all nodes**

   $ for s in `cd /etc/init.d/; ls hadoop-hdfs*`; do sudo service $s stop; done

6. Create OS path for Hadoop HDFS and MapReduce

```
$ sudo mkdir -p /disk1/dfs/nn
$ sudo mkdir -p /disk2/dfs/nn
$ sudo mkdir -p /disk1/dfs/dn
$ sudo mkdir -p /disk2/dfs/dn
$ sudo mkdir -p /disk1/mapred/local
$ sudo mkdir -p /disk2/mapred/local
```

7. Change properties

```
$ sudo chown -R hdfs:hadoop /disk1/dfs/nn
$ sudo chown -R hdfs:hadoop /disk2/dfs/nn
$ sudo chown -R hdfs:hadoop /disk1/dfs/dn
$ sudo chown -R hdfs:hadoop /disk2/dfs/dn
$ sudo chown -R mapred:hadoop /disk1/mapred/local
$ sudo chown -R mapred:hadoop /disk2/mapred/local
```

8. Configure HDFS on **elephant**

- `$ sudo nano /etc/hadoop/conf/`**`core-site.xml`**

```
<configuration>
  <property>
```

```
   <name>fs.default.name</name>
   <value>hdfs://elephant:8020</value>
  </property>
</configuration>
```

- $ sudo nano /etc/hadoop/conf/**hdfs-site.xml**

```
<configuration>
  <property>
    <name>dfs.name.dir</name>
    <value>/disk1/dfs/nn,/disk2/dfs/nn</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>/disk1/dfs/dn,/disk2/dfs/dn</value>
  </property>
  <property>
    <name>dfs.http.address</name>
    <value>elephant:50070</value>
  </property>
</configuration>
```

9. Configure enviroment vars for Java memory usage

- $ sudo nano /etc/hadoop/conf/hadoop-env.sh

```
export HADOOP_NAMENODE_OPTS="-Xmx64m"
export HADOOP_SECONDARYNAMENODE_OPTS="-Xmx64m"
export HADOOP_DATANODE_OPTS="-Xmx64m"
export HADOOP_JOBTRACKER_OPTS="-Xmx64m"
export HADOOP_TASKTRACKER_OPTS="-Xmx64m"
```

- $ sudo chmod +x /etc/hadoop/conf/hadoop-env.sh

10. Copy configuration to all nodes

- $ sudo yum install -y sshpass
- $ nano ~/copy_config.sh

```
#!/bin/bash
sshpass -p 'cloudera' scp /etc/hadoop/conf/core-site.xml
root@tiger:/etc/hadoop/conf/
sshpass -p 'cloudera' scp /etc/hadoop/conf/core-site.xml
root@horse:/etc/hadoop/conf/
sshpass -p 'cloudera' scp /etc/hadoop/conf/core-site.xml
root@monkey:/etc/hadoop/conf/
sshpass -p 'cloudera' scp /etc/hadoop/conf/hdfs-site.xml
root@tiger:/etc/hadoop/conf/
```

```
sshpass -p 'cloudera' scp /etc/hadoop/conf/hdfs-site.xml
root@horse:/etc/hadoop/conf/

sshpass -p 'cloudera' scp /etc/hadoop/conf/hdfs-site.xml
root@monkey:/etc/hadoop/conf/

sshpass -p 'cloudera' scp /etc/hadoop/conf/mapred-site.xml
root@tiger:/etc/hadoop/conf/

sshpass -p 'cloudera' scp /etc/hadoop/conf/mapred-site.xml
root@horse:/etc/hadoop/conf/

sshpass -p 'cloudera' scp /etc/hadoop/conf/mapred-site.xml
root@monkey:/etc/hadoop/conf/

sshpass -p 'cloudera' scp /etc/hadoop/conf/hadoop-env.sh
root@tiger:/etc/hadoop/conf/

sshpass -p 'cloudera' scp /etc/hadoop/conf/hadoop-env.sh
root@horse:/etc/hadoop/conf/

sshpass -p 'cloudera' scp /etc/hadoop/conf/hadoop-env.sh
root@monkey:/etc/hadoop/conf/

echo done
```

- `$ chmod +x copy_config.sh`
- `Exec first time ssh againts consoles (to storage RSA key fingerprint)`
  - `$ ssh tiger (write yes and cancel later with Control+C)`
  - `$ ssh horse (write yes and cancel later with Control+C)`
  - `$ ssh monkey (write yes and cancel later with Control+C)`
- `./copy_config.sh`

11. Format HDFS and start NameNode

- `$ sudo service hadoop-hdfs-namenode stop`
  `Wait until NaneNode kill Java Process`
- `$ sudo jps |grep NameNode`
- `$ sudo -u hdfs hdfs namenode -format`

```
14/10/06 23:25:04 INFO blockmanagement.HeartbeatManager: Setting heartbeat recheck interval to
30000 since dfs.namenode.stale.datanode.interval is less than dfs.namenode.heartbeat.recheck-in
terval
14/10/06 23:25:04 INFO blockmanagement.DatanodeManager: dfs.block.invalidate.limit=1000
14/10/06 23:25:04 INFO util.GSet: Computing capacity for map BlocksMap
14/10/06 23:25:04 INFO util.GSet: VM type       = 64-bit
14/10/06 23:25:04 INFO util.GSet: 2.0% max memory 61.9 MB = 1.2 MB
14/10/06 23:25:04 INFO util.GSet: capacity      = 2^17 = 131072 entries
14/10/06 23:25:04 INFO blockmanagement.BlockManager: dfs.block.access.token.enable=false
14/10/06 23:25:04 INFO blockmanagement.BlockManager: defaultReplication         = 3
14/10/06 23:25:04 INFO blockmanagement.BlockManager: maxReplication             = 512
14/10/06 23:25:04 INFO blockmanagement.BlockManager: minReplication             = 1
14/10/06 23:25:04 INFO blockmanagement.BlockManager: maxReplicationStreams      = 2
14/10/06 23:25:04 INFO blockmanagement.BlockManager: shouldCheckForEnoughRacks  = false
14/10/06 23:25:04 INFO blockmanagement.BlockManager: replicationRecheckInterval = 3000
14/10/06 23:25:04 INFO blockmanagement.BlockManager: encryptDataTransfer        = false
14/10/06 23:25:04 INFO blockmanagement.BlockManager: maxNumBlocksToLog          = 1000
14/10/06 23:25:04 INFO namenode.FSNamesystem: fsOwner             = hdfs (auth:SIMPLE)
14/10/06 23:25:04 INFO namenode.FSNamesystem: supergroup          = supergroup
14/10/06 23:25:04 INFO namenode.FSNamesystem: isPermissionEnabled = true
14/10/06 23:25:04 INFO namenode.FSNamesystem: HA Enabled: false
14/10/06 23:25:04 INFO namenode.FSNamesystem: Append Enabled: true
14/10/06 23:25:05 INFO namenode.NameNode: Caching file names occuring more than 10 times
14/10/06 23:25:05 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.999000012
8746033
14/10/06 23:25:05 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
14/10/06 23:25:05 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension     = 30000
Re-format filesystem in Storage Directory /disk1/dfs/nn ? (Y or N) y
Re-format filesystem in Storage Directory /disk2/dfs/nn ? (Y or N) Y
14/10/06 23:25:11 INFO namenode.NNStorage: Storage directory /disk1/dfs/nn has been successfull
y formatted.
14/10/06 23:25:11 INFO namenode.NNStorage: Storage directory /disk2/dfs/nn has been successfull
y formatted.
14/10/06 23:25:11 INFO namenode.FSImage: Saving image file /disk2/dfs/nn/current/fsimage.ckpt_0
000000000000000000 using no compression
14/10/06 23:25:11 INFO namenode.FSImage: Saving image file /disk1/dfs/nn/current/fsimage.ckpt_0
000000000000000000 using no compression
14/10/06 23:25:11 INFO namenode.FSImage: Image file of size 119 saved in 0 seconds.
14/10/06 23:25:11 INFO namenode.FSImage: Image file of size 119 saved in 0 seconds.
14/10/06 23:25:11 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >
= 0
14/10/06 23:25:11 INFO util.ExitUtil: Exiting with status 0
14/10/06 23:25:11 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at elephant.gpul.org/192.168.1.1
************************************************************/
[cloudera@elephant ~]$
```

- `$ sudo service hadoop-hdfs-namenode start`

12. Check Java processes with parameters

- sudo jps

13. Start Secundary NameNode on **tiger**

- `$ sudo service hadoop-hdfs-secondarynamenode start`

14. Start DataNode on **all nodes**

- `$ sudo service hadoop-hdfs-datanode start`

15. Verify daemons on Java process

    $ sudo jps

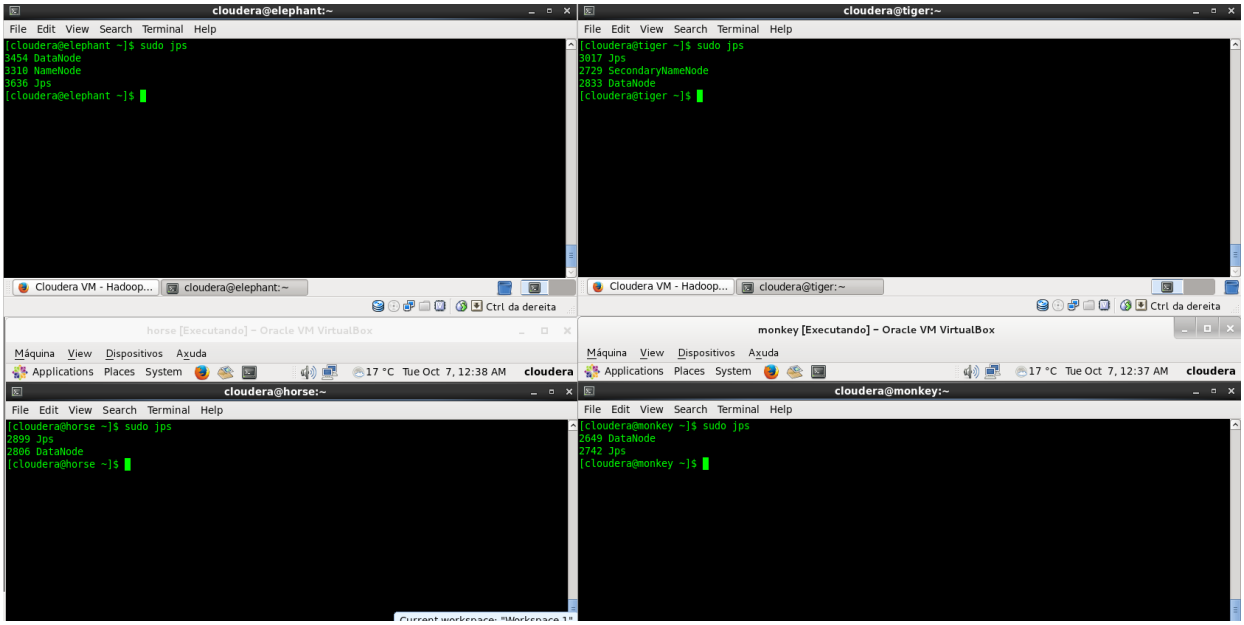
16. If there are some problems check *log files*

    $ tail -n 99 /var/log/hadoop-hdfs/hadoop-hdfs-datanode-elephant.log

    $ tail -n 99 /var/log/hadoop-hdfs/hadoop-hdfs-datanode-tiger.log

    $ tail -n 99 /var/log/hadoop-hdfs/hadoop-hdfs-datanode-horse.log

    $ tail -n 99 /var/log/hadoop-hdfs/hadoop-hdfs-datanode-monkey.log



17. Create path on HDFS for user

- `$ sudo -u hdfs hadoop fs -mkdir /user/cloudera`
- `$ sudo -u hdfs hadoop fs -chown cloudera /user/cloudera`