edureka!

Hadoop Lab - Setting a 3 node Cluster

Packages

Hadoop Packages can be downloaded from:

http://hadoop.apache.org/releases.html

Java - http://wiki.apache.org/hadoop/HadoopJavaVersions

Note: *I have tested the Lab environment with OpenJdk and it works fine. But, if you need to run Pig scripts or spawn many jobs, then use an Oracle_sun-jdk.*

My Packages

jdk-7u25-linux-x64.rpm (*Can be downloaded from Oracle site*)

hadoop-20.205.tar.gz (*Use this for Hadoop 1.0 setup and for Hadoop 2.0, version greater then 0.23*)

For simplicity of Labs and get started with things, use Hadoop 1.0 - i.e hadoop-20.205.tar.gz

edureka!

Base Machine

Model Name: MacBook Pro

Processor Name: Intel Core i7

Processor Speed: 2.7 GHz

Number of Processors: 1

Total Number of Cores: 4

L2 Cache (per Core): 256 KB

L3 Cache: 6 MB

Memory: 16 GB

As you can see that the base machine has really good configurations, you need at-least a 4 GB machine for 2 nodes Virtualization of Hadoop.

LAB Setup

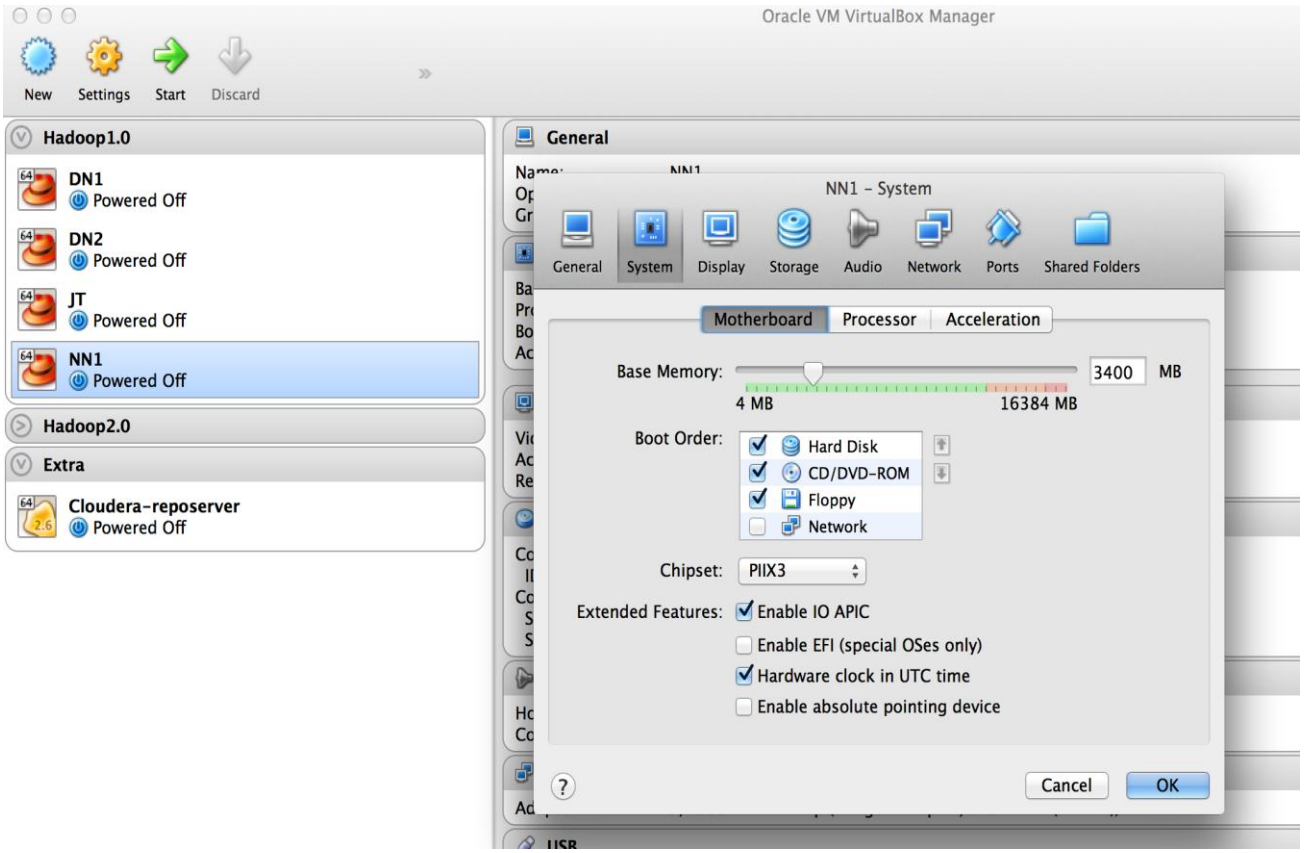
I am using **CentOS 5.4**, but you can use any Linux Flavor.

Virtualization: Can use VMware or VirtualBox.

Oracle VirtualBox - Can be downloaded free from the Oracle site. Download and install on Base Machine.

- Create a Linux Virtual Machine in the Vbox, allocate minimum 2 GB RAM, 4 GB HDD.
- Do a Minimal install of OS.
- Once a Virtual machine is created, run and test it. Then clone it - Do a full clone, using the clone option in the Vbox.

- Can do the number of clone, as per the number of data nodes required and the Base system configuration you have.
- Assign a unique hostname and IP to each machine. Make sure both the hosts are in the same subnet and can ping each other.
- Can create extra virtual hard drives and mount them under **/data**. We will use them for hadoop storage in LAB.

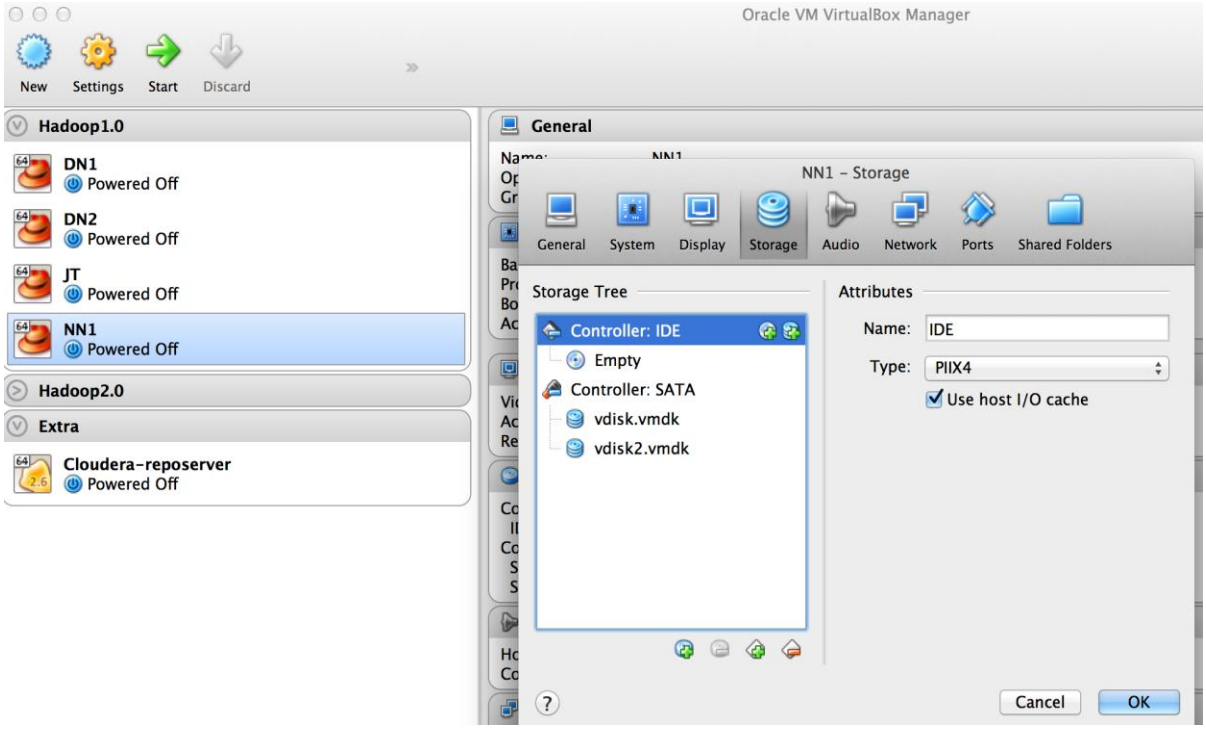


Namenode:

Hostname: nn1.cluster1.com

Domain: cluster1.com

IP: 192.168.1.70

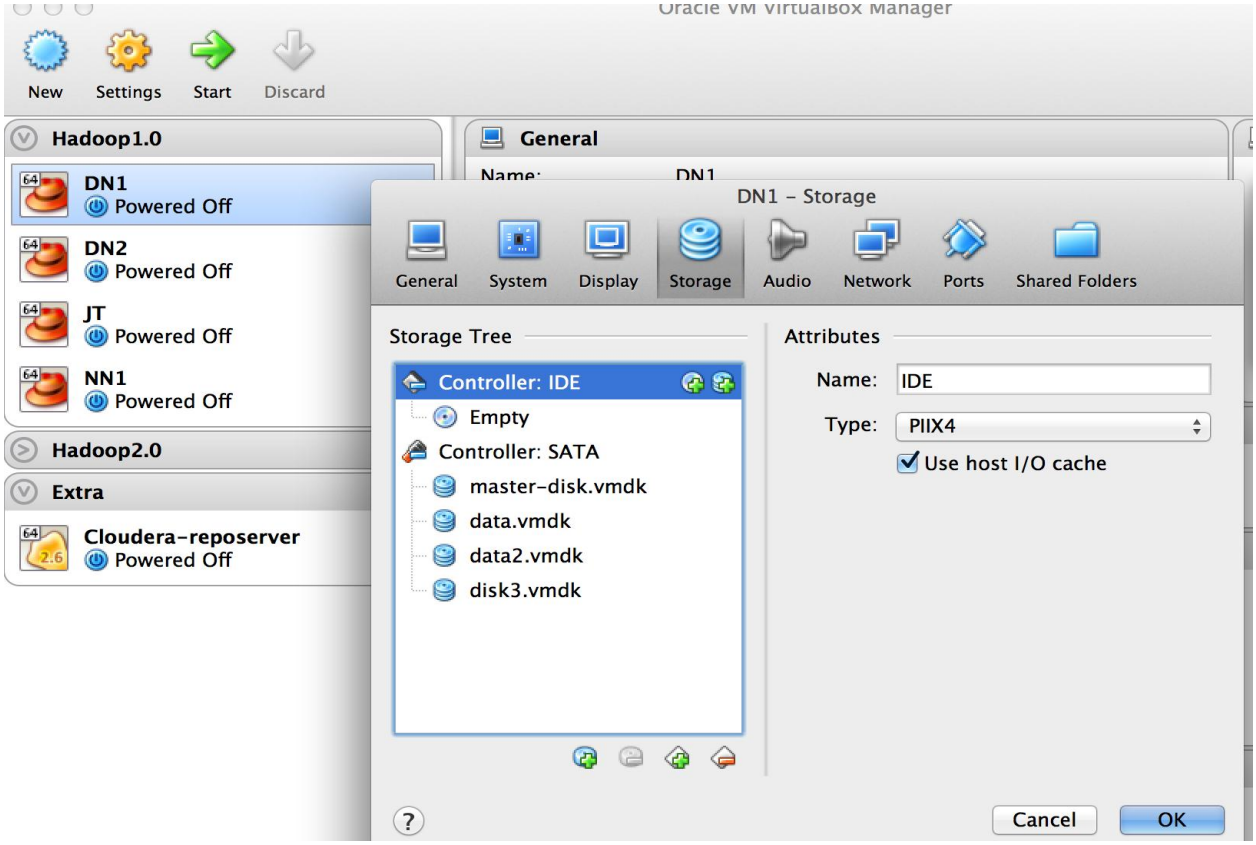


Datanode1:

Hostname: dn1.cluster1.com

Domain: cluster1.com

IP: 192.168.1.71



Datanode2:

Hostname: dn2.cluster1.com

Domain: cluster1.com

IP: 192.168.1.72

For the hosts to communicate using host names - Please configure DNS on any one of the nodes. My Namenode is acting as a DNS server.

On All host:

/etc/resolv.conf

nameserver 192.168.1.70

Test, that you can resolve DNS from all hosts.

$ nslookup nn1.cluster1.com

If it returns the IP, we are all set.

Hadoop Installation

1. Log-in to namenode (nn1) as root - Download the Hadoop and Java Packages. Best is to download it on the base Machine by using wget or scp to all the nodes.

2. Create a user hadoop - "**useradd hadoop**" and set a password for the user using "**echo hadoop | passwd --stdin hadoop**" (Remember we are doing this as "root" or use "sudo")

3. Install Java using "**rpm -ivh --aid jdk-7u25-linux-x64.rpm**" or use yum/apt-get as per your distro.

4. Extract Hadoop using "**tar -zxvf hadoop-20.205.tar.gz /home/hadoop/**".

5. **chown hadoop:hadoop -R /home/hadoop/hadoop**

6. **sudo - hadoop** (Change as hadoop user)

7. Execute command "**pwd**" - You must be in /home/hadoop

8. **cd hadoop/conf**

9. Do "**ls -l**" and take a look at the config files there. This is the Hadoop configuration directory.

10. We need to edit 3 important files - **hadoop-env.sh, hdfs-site.xml, core-site.xml, mappred-site-xml**

The files will contain:

1. **hadoop-env.sh**

This is the file, which sets the Hadoop environment, like JAVA PATH, memory requirement etc.

Add a line in the file depending upon the JAVA installation PATH

- *export JAVA_HOME=/usr/bin/java/*

or

- *export JAVA_HOME=/usr/java/jdk1.7.0_25/*

Hadoop Path

- *export HADOOP_HOME=/home/hadoop/hadoop*

2. **core-site.sh**

```
<configuration>

<property>
<name>fs.default.name</name>
<value>hdfs://nn1.cluster1.com:9000</value>
</property>

</configuration>
```

3. **hdfs-site.xml**

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

<property>
```

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

<value>/var/datastore</value> <------ this section only on Namenode

<final>true</final>

</property>

<property>

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

<value>/home/data</value> <------ this section only on Datanode

<final>true</final>

</property>

</configuration>

Note: /home/data - Can be a separate HDD or a partition for storing data on the Data node.

4. **mapred-site.xml**

<configuration>

<property>

<name>mapred.job.tracker</name>

<value>nn1.cluster1.com:9001</value>

</property>

</configuration>

5. Another important file to setup is the **".bash_profile"**

In you home directory i.e /home/hadoop

1. **vi .bash_profile**

Add the below lines to the file, do not remove the existing lines

User specific environment and startup programs

PATH=$PATH:$HOME/bin

PATH=$PATH:/usr/java/jdk1.7.0_25/bin

export JAVA_HOME=/usr/java/jdk1.7.0_25/

export HADOOP_HOME=/home/hadoop/hadoop

export PATH=$PATH:$HADOOP_HOME/bin

export HADOOP_HOME_WARN_SUPPRESS="TRUE"

2. Either logout and login back or just run " *. ./bash_profile"*

Starting Hadoop

As user hadoop, on Namenode (nn1)

1. Execute "**hadoop namenode -format**" - If this executes, then all your PATH and environment is fine. This will format the namenode. Do not worry, format here does not mean the same as file system formatting. All you data will be safe.
2. $ **hadoop-daemon.sh start namenode**
3. $ **jps**

Now, you should see Namenode running in the process listed.

You can check namenode also from the browser: **http://nn1.cluster1.com:50070**

4. $ **hadoop-daemon.sh start jobtracker**

Check Job tracker URI : **http://nn1.cluster1.com:50030**

5. $ **jps**

Now, you will see jobtracker running as well.

Setting up Datanode.

As user Hadoop on datanodes

Do all the steps as above before the Hadoop startup.

- $ No need to format a "datanode". We will see the details on why?
- $ **hadoop-daemon.sh start datanode**
- $ **hadoop-daemon.sh start tasktracker**
- $ **jps**

What daemons you see?

Check **http://nn1.cluster1.com:50070**

to see the difference in the storage presentation then before.

Play around with the daemons, to see how they are related.

5. **Some hadoop commands**

$ **hadoop fs -ls /**

$ **hadoop fsck /**