Hands-On Exercise: Creating A Hadoop Cluster

In this exercise, you will create and test a fully-distributed Hadoop cluster. You will work in teams for this exercise; your instructor will divide the class into groups, and each group will create its own cluster.

Disabling The Pseudo-Distributed Cluster and Setting Up A New Configuration Directory

The first stage is to stop the existing, pseudo-distributed cluster.

1. Stop your existing pseudo-distributed cluster.

```
$ for service in /etc/init.d/hadoop-*
do
sudo $service stop
done
```

The CDH installation uses the Linux *alternatives* framework, which allows you to specify multiple different configurations. However, for simplicity we will just use the /etc/hadoop/conf directory and edit files directly there. First, we will copy some blank default files to replace the existing pseudo-distributed configuration files.

```
$ sudo cp /etc/hadoop/conf.empty/* /etc/hadoop/conf
```

2. Change directories into the configuration directory.

```
$ cd /etc/hadoop/conf
```

Setting Up The Configuration Files

Your group should choose one machine to be the NameNode, one to be the Secondary NameNode, and one to be the JobTracker. All machines will act as

Copyright © 2010-2013 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

TaskTrackers and DataNodes. We will put the same configuration files on all machines. **Where we refer to *namenode_name*, *jobtracker_name* etc. in the instructions below, you should replace that with the name of the relevant machine.**

Note: when editing the configuration files, you will need to use sudo. For example:

```
$ sudo vi core-site.xml
```

Also note: when you add configuration properties to the configuration files, the format is:

```
<property>
     <name>property_name</name>
     <value>property_value</value>
</property>
<property>
     <name>another_property_name</name>
     <value>another_value</value>
</property>
...
```

1. Determine your machine's IP address. Run the following command:

   ```
   $ ifconfig
   ```

   You should see data about your network connections. The eth0 interface should have an IP address, noted as 'inet addr'. Make a note of the IP addresses of the Master node, and all the Slave nodes. As a group, decide on a name for each machine, and note that down, too. (Your machine can be named after you, or given an arbitrary name.)

2. In order for the nodes to report their names correctly, we will need to modify our networking information. First, edit the `/etc/hosts` file (you will need to

Copyright © 2010-2013 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

use `sudo vi /etc/hosts` for this). Remove the line which refers to `cloudera-training`. Add new lines at the bottom of the file, of the form <ip_address><tab><machine_name> – one line per machine in your group. For example:

```
10.123.234.99        bart
10.123.234.101       maggie
10.123.234.104       homer
10.123.234.202       lisa
```

3. Edit `/etc/sysconfig/network` (you will need to use `sudo vi /etc/sysconfig/network` for this), and change the `HOSTNAME=` line to reflect your machine's name.

4. Change your hostname for the current session

```
$ sudo hostname your-host-name
```

5. Check that you can ping the other machines in your group. For example:

```
$ ping bart
```

6. Change directories back to the configuration directory:

```
$ cd /etc/hadoop/conf
```


Copyright © 2010-2013 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

7. Edit `core-site.xml`, adding the following property and value between the `<configuration>...</configuration>` tags. Don't forget your `<property>`, `<name>` and `<value>` tags, and don't forget to use `sudo` when you edit Hadoop configuration files.

Name	**Value**
fs.default.name	hdfs://*namenode_name*:8020

8. Edit `hdfs-site.xml`, adding the following properties and values. Note that although you are adding properties for several different daemons, you will only be starting certain daemons on your machine depending on its role (NameNode, JobTracker and so on).

   Note that for the `dfs.name.dir` property:

   - The `hdfs-site.xml` file is already pre-populated with this property. You do not need to add the property again, simply replace the existing value with the value in the following table.
   - We are specifying two directories on the same disk. In production, of course, these should be on different disks, and you should also include an NFS mount.

Name	**Value**
dfs.name.dir	/disk1/dfs/nn,/disk2/dfs/nn
dfs.data.dir	/disk1/dfs/dn,/disk2/dfs/dn
dfs.http.address	*namenode_name*:50070

Copyright © 2010-2013 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

9. Edit `mapred-site.xml`, adding the following properties and values. Note that for the `mapred.job.tracker` property, the `mapred-site.xml` file is already pre-populated with this property. You do not need to add the property again, simple replace the existing value with the value in the following table.

Name	**Value**
mapred.local.dir	/disk1/mapred/local,/disk2/mapred/local
mapred.job.tracker	*jobtracker_name*:8021
mapred.system.dir	/mapred/system
mapreduce.jobtracker.staging.root.dir	/user

10. Create the directories specified in the properties above.

```
$ 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
```


Copyright © 2010-2013 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

11. Change the ownership of the directories. The `hdfs` user should own the HDFS directories:

```
$ 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
```

The `mapred` user should own the MapReduce directories.

```
$ sudo chown -R mapred:hadoop /disk1/mapred/local
$ sudo chown -R mapred:hadoop /disk2/mapred/local
```

12. Because we are running on a Virtual Machine with very limited RAM, we will reduce the heap size of the Hadoop daemons to 200MB. Create a file called `hadoop-env.sh` and add the following line

```
export HADOOP_HEAPSIZE=200
```

13. Make the `hadoop-env.sh` file executable:

```
chmod +x hadoop-env.sh
```

14. When everyone has completed these steps, format the NameNode. Note: **Do this only on the node which will be the NameNode.**

```
$ sudo -u hdfs hdfs namenode -format
```

(Note: if you are asked whether you want to reformat the filesystem, answer 'Y'.)

Copyright © 2010-2013 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

15. Start the HDFS daemons.

On the NameNode machine only:

```
$ sudo /etc/init.d/hadoop-hdfs-namenode start
$ sudo /etc/init.d/hadoop-hdfs-secondarynamenode start
```

On all machines:

```
$ sudo /etc/init.d/hadoop-hdfs-datanode start
```

16. Create a home directory for the user `training`. **Only one person should do this. Choose one member of the team to enter these commands.**

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

17. Create the MapReduce system directory within HDFS. **Only one person should do this**. **Choose one member of the team to enter these commands.**

```
$ sudo -u hdfs hadoop fs -mkdir /mapred/system
$ sudo -u hdfs hadoop fs -chown mapred:hadoop \
/mapred/system
```

18. Start the MapReduce daemons.

On the JobTracker machine only:

```
$ sudo /etc/init.d/hadoop-0.20-mapreduce-jobtracker \
start
```

On all machines:

```
$ sudo /etc/init.d/hadoop-0.20-mapreduce-tasktracker \
start
```


Copyright © 2010-2013 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

19. Use the `sudo jps` command to ensure that the relevant daemons are running successfully on your node. If they are not, find and fix the problem(s) and restart the daemons.

20. Test the cluster! Upload some data to the cluster, and run a MapReduce job.

```
$ cd /home/training/assets
$ hadoop fs -mkdir <yourname>/shakespeare
$ hadoop fs -put shakespeare.txt <yourname>/shakespeare
$ hadoop jar /usr/lib/hadoop-0.20*/hadoop-examples.jar \
wordcount <yourname>/shakespeare <yourname>/output
```

Once the job has run, examine the output to ensure it was successful.

This is the end of the Exercise.

Copyright © 2010-2013 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.