**HU Extension School E-185 Big Data Analytics Assignment 05**

**Handed out: 03/09/2013 Due by 5:30PM on Friday, 03/15/2013**

**Problem 1) Please download and install VMware Workstation 9 or VMWare Fusion 5 if on MAC. Create a virtual machine based on CentOS 6.3 operating system. If you have a 32 bit machine, please pay attention and install 32 bit version of CentOS. If you have a 64 bit machine be free to download 64 bit CentOS and create a 64 bit VM. If you work with another flavor of Linux, please be free to create a virtual machine based on your favorite flavor. Name the main user of your VM cloudera. Do not use name hadoop, as I did in my notes. “hadoop” is a bad name for a user, since Hadoop framework has an executable called hadoop and it creates many directories with that same name and those are not necessarily owned by that VM user called hadoop. On that VM create yet another user called joe. Make both users sudo users.**

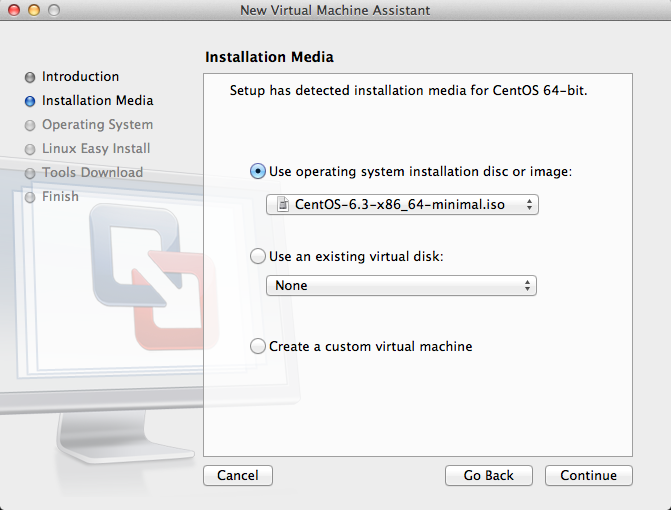
**Follow closely steps in the attached CDH4 Quick Start Quide which will guide you through an “automated” process of installing Hadoop. Read that guide very carefully. If you are working with 32 bit CentOS VM, use only commands for 32 bit CentOS. Do not execute commands for other flavors of Linux. You will know that you have successfully installed Hadoop is all of tests described in the guide work properly. Guide describes installation of two versions of Map Reduce package. Go through both installations, if for no other reason then to learn how you remove an installed package.**

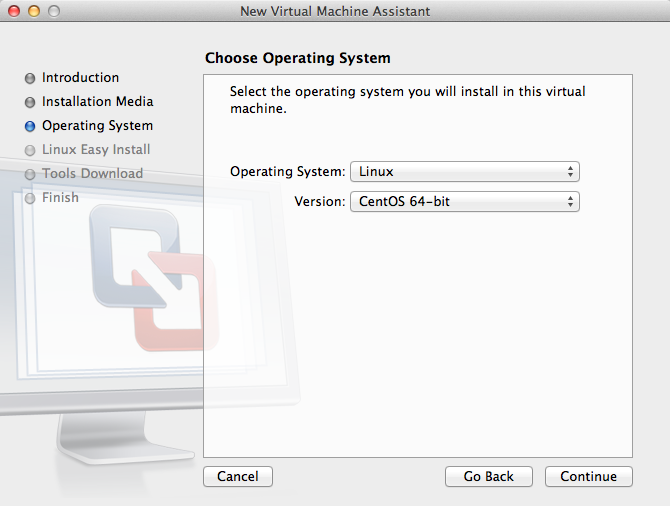
**Capture all steps of your implementation with comments indicating what is it you are accomplishing with every step. Place those in an MS Word document.**

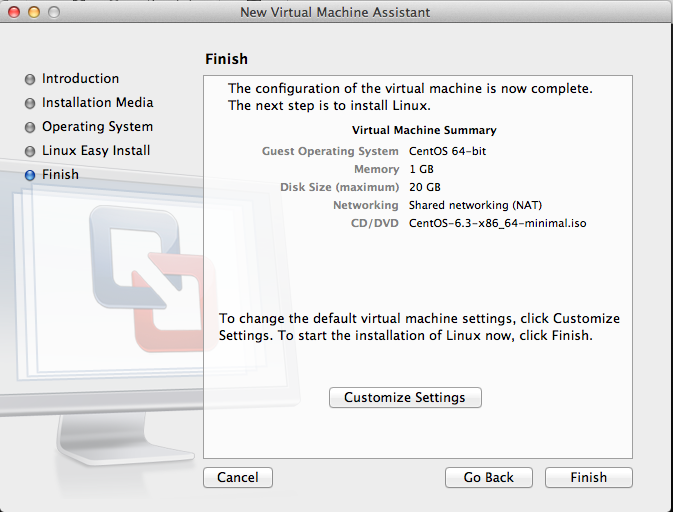
**Please place all files you want to submit in a folder named: HW05. Compress that folder into an archive named E185\_LastNameFirstNameHW05. ZIP. Upload the archive to the course drop box, i.e. the web site. Please send comments and questions to the Discussion Forum on the class site. If some questions are not clarified on the discussion forum please submit them to cscie185@fas.harvard.edu**

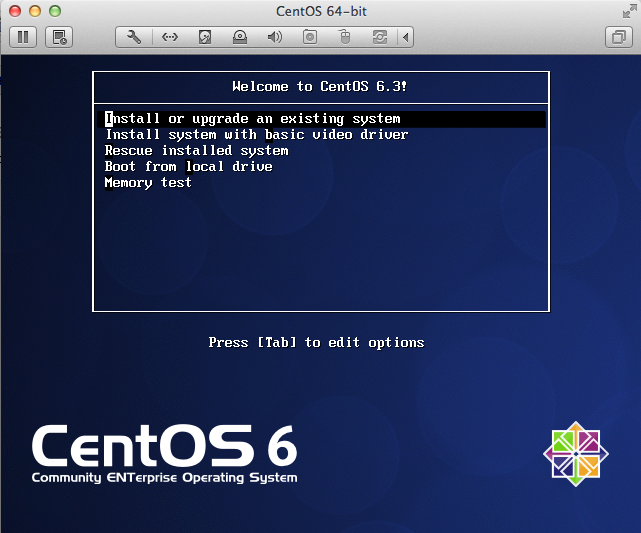
Initial Install on the Virtual Machine

We begin by downloading the CentOS-6.3-x86\_64-minimal.iso file, as we do not want the graphical interface or a lot of clutter on our VM. Once it is downloaded, we create the virtual machine by installing the ISO on Vmware Fusion.

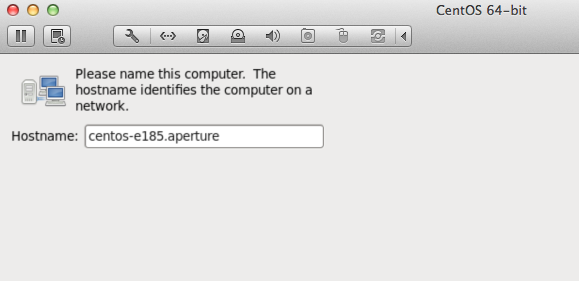


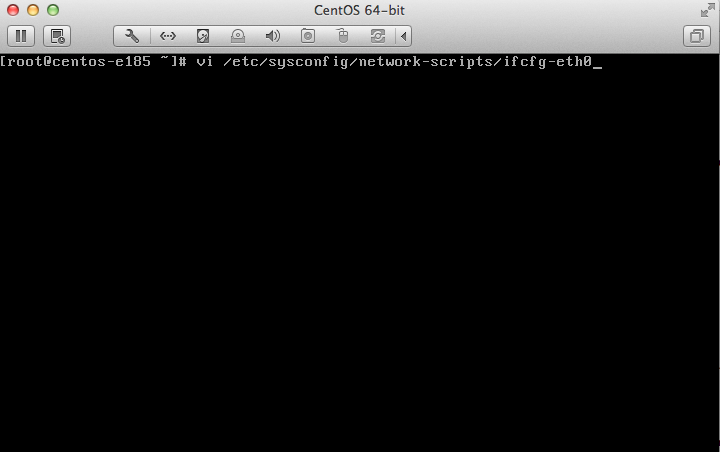


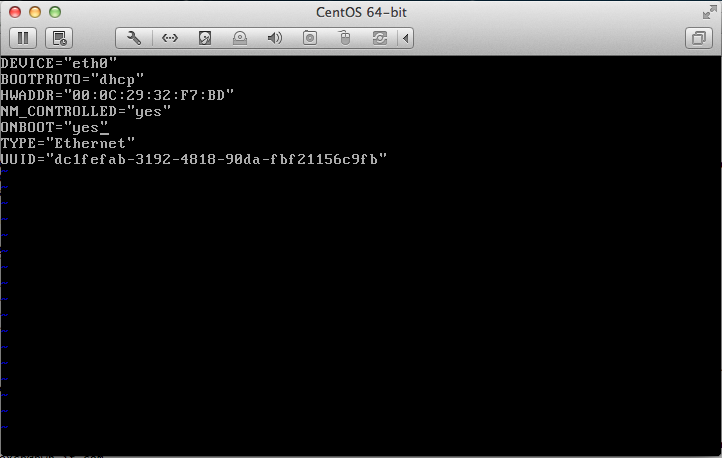


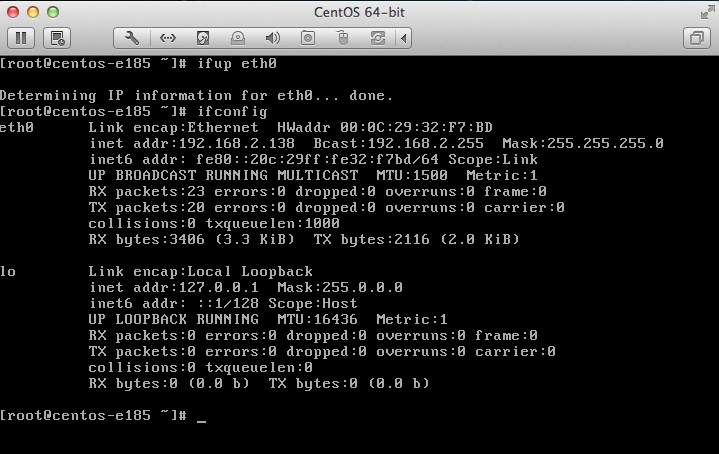


After accepting almost all of the default options and setting the hostname to centos-e185.aperture, we move forward to connecting to it through ssh on our local machine.









aperture:~ alexcp$ ssh root@192.168.2.138

The authenticity of host '192.168.2.138 (192.168.2.138)' can't be established.

RSA key fingerprint is 64:97:bb:77:dd:f7:72:f8:45:26:83:7d:0d:88:0b:18.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '192.168.2.138' (RSA) to the list of known hosts.

root@192.168.2.138's password:

Last login: Wed Mar 13 04:38:13 2013

[root@centos-e185 ~]#

Now, we create the 'cloudera' and 'joe' users, and set some meaningful passwords for them:

[root@centos-e185 ~]# useradd cloudera

[root@centos-e185 ~]# passwd cloudera

Changing password for user cloudera.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

[root@centos-e185 ~]# useradd joe

[root@centos-e185 ~]# passwd joe

Changing password for user joe.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

[root@centos-e185 ~]#

Since this is really a minimal install, we need to add the package sudo to it:

[root@centos-e185 ~]# yum install sudo

Loaded plugins: fastestmirror, presto

Loading mirror speeds from cached hostfile

(...)

Installed:

sudo.x86\_64 0:1.8.6p3-7.el6

Complete!

[root@centos-e185 ~]#

And we add the users to the sudoers file:

[root@centos-e185 ~]# visudo

(...)

## Allow root to run any commands anywhere

root ALL=(ALL) ALL

cloudera ALL=(ALL) ALL

joe ALL=(ALL) ALL

(...)

[root@centos-e185 ~]#

And we run a quick test to make sure that adding them to the sudoers file worked:

[root@centos-e185 ~]# logout

Connection to 192.168.2.138 closed.

aperture:~ alexcp$ ssh cloudera@192.168.2.138

cloudera@192.168.2.138's password:

[cloudera@centos-e185 ~]$ reboot

reboot: Need to be root

[cloudera@centos-e185 ~]$ sudo reboot

We trust you have received the usual lecture from the local System

Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.

#2) Think before you type.

#3) With great power comes great responsibility.

[sudo] password for cloudera:

Broadcast message from cloudera@centos-e185.aperture

(/dev/pts/0) at 5:06 ...

The system is going down for reboot NOW!

[cloudera@centos-e185 ~]$ Connection to 192.168.2.138 closed by remote host.

Connection to 192.168.2.138 closed.

Now we can begin following the Cloudera document. :)

Installing Java

After downloading the JDK 6u31 from the Oracle website, we copy it over to our VM to start the installation.

aperture:~ alexcp$ scp jdk-6u31-linux-x64-rpm.bin root@192.168.2.138:

root@192.168.2.138's password:

jdk-6u31-linux-x64-rpm.bin 100% 77MB 77.3MB/s 00:01

aperture:~ alexcp$

We create the directory and extract the file to install the JDK:

aperture:~ alexcp$ ssh root@192.168.2.138

root@192.168.2.138's password:

Last login: Wed Mar 13 04:39:21 2013 from 192.168.2.1

[root@centos-e185 ~]# ls

anaconda-ks.cfg install.log install.log.syslog jdk-6u31-linux-x64-rpm.bin

[root@centos-e185 ~]# mkdir jdk

[root@centos-e185 ~]# mv jdk-6u31-linux-x64-rpm.bin jdk

[root@centos-e185 ~]# cd jdk

[root@centos-e185 jdk]# ls

jdk-6u31-linux-x64-rpm.bin

[root@centos-e185 jdk]# chmod +x jdk-6u31-linux-x64-rpm.bin

[root@centos-e185 jdk]# ./jdk-6u31-linux-x64-rpm.bin

Unpacking...

Checksumming...

Extracting...

UnZipSFX 5.50 of 17 February 2002, by Info-ZIP (Zip-Bugs@lists.wku.edu).

inflating: jdk-6u31-linux-amd64.rpm

inflating: sun-javadb-common-10.6.2-1.1.i386.rpm

inflating: sun-javadb-core-10.6.2-1.1.i386.rpm

inflating: sun-javadb-client-10.6.2-1.1.i386.rpm

inflating: sun-javadb-demo-10.6.2-1.1.i386.rpm

inflating: sun-javadb-docs-10.6.2-1.1.i386.rpm

inflating: sun-javadb-javadoc-10.6.2-1.1.i386.rpm

Preparing... ########################################### [100%]

1:jdk ########################################### [100%]

Unpacking JAR files...

rt.jar...

jsse.jar...

charsets.jar...

tools.jar...

localedata.jar...

plugin.jar...

javaws.jar...

deploy.jar...

Installing JavaDB

Preparing... ########################################### [100%]

1:sun-javadb-common ########################################### [ 17%]

2:sun-javadb-core ########################################### [ 33%]

3:sun-javadb-client ########################################### [ 50%]

4:sun-javadb-demo ########################################### [ 67%]

5:sun-javadb-docs ########################################### [ 83%]

6:sun-javadb-javadoc ########################################### [100%]

(...)

Done.

[root@centos-e185 jdk]# which java

/usr/bin/java

[root@centos-e185 jdk]# java -version

java version "1.6.0\_31"

Java(TM) SE Runtime Environment (build 1.6.0\_31-b04)

Java HotSpot(TM) 64-Bit Server VM (build 20.6-b01, mixed mode)

[root@centos-e185 jdk]#

Finally, we look for the location of the JAVA\_HOME directory and add it to the variables that are loaded to the environment when a user logs in:

[root@centos-e185 jdk]# find / -name java -print

/usr/bin/java

/usr/java

/usr/java/jdk1.6.0\_31/bin/java

/usr/java/jdk1.6.0\_31/jre/bin/java

/etc/pki/java

/opt/sun/javadb/demo/programs/scores/java

/opt/sun/javadb/demo/programs/vtis/java

Scripts on the /etc/profile.d directory are automatically run when any user logs in, so we create a separate file for this purpose, as to keep things modular

[root@centos-e185 jdk]# vi /etc/profile.d/java.sh

[root@centos-e185 jdk]# cat /etc/profile.d/java.sh

## Exporting JAVA\_HOME and an updated PATH to all user on the machine

export JAVA\_HOME=/usr/java/jdk1.6.0\_31

export PATH=$JAVA\_HOME/bin:$PATH

[root@centos-e185 jdk]#

And then we test that the export works properly.

aperture:~ alexcp$ ssh cloudera@192.168.2.138

cloudera@192.168.2.138's password:

Last login: Wed Mar 13 05:06:31 2013 from 192.168.2.1

[cloudera@centos-e185 ~]$ echo $JAVA\_HOME

/usr/java/jdk1.6.0\_31

[cloudera@centos-e185 ~]$ echo $PATH

/usr/java/jdk1.6.0\_31/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/cloudera/bin

[cloudera@centos-e185 ~]$

Installing Hadoop MRv1

Initially, we download the correct cloudera-cdh rpm and upload it to our VM.

aperture:~ alexcp$ scp cloudera-cdh-4-0.x86\_64.rpm cloudera@192.168.2.138:

cloudera@192.168.2.138's password:

cloudera-cdh-4-0.x86\_64.rpm 100% 9108 8.9KB/s 00:00

aperture:~ alexcp$ ssh cloudera@192.168.2.138

cloudera@192.168.2.138's password:

Last login: Wed Mar 13 05:38:33 2013 from 192.168.2.1

[cloudera@centos-e185 ~]$

And then we follow the first steps described on the Quick Start Document:

[cloudera@centos-e185 ~]$ sudo yum --nogpgcheck localinstall cloudera-cdh-4-0.x86\_64.rpm

[sudo] password for cloudera:

Loaded plugins: fastestmirror, presto

Setting up Local Package Process

Examining cloudera-cdh-4-0.x86\_64.rpm: cloudera-cdh-4-0.x86\_64

(...)

Installed:

cloudera-cdh.x86\_64 0:4-0

Complete!

[cloudera@centos-e185 ~]$ sudo rpm --import http://archive.cloudera.com/cdh4/redhat/6/x86\_64/cdh/RPM-GPG-KEY-cloudera

[cloudera@centos-e185 ~]$

And install the hadoop packages:

[cloudera@centos-e185 ~]$ sudo yum install hadoop-0.20-conf-pseudo

Loaded plugins: fastestmirror, presto

Loading mirror speeds from cached hostfile

\* base: linux.mirrors.es.net

\* extras: mirror.hmc.edu

\* updates: centos.sonn.com

(...)

Setting up Install Process

(...)

Complete!

[cloudera@centos-e185 ~]$ rpm -ql hadoop-0.20-conf-pseudo

/etc/hadoop/conf.pseudo.mr1

/etc/hadoop/conf.pseudo.mr1/README

/etc/hadoop/conf.pseudo.mr1/core-site.xml

/etc/hadoop/conf.pseudo.mr1/hadoop-metrics.properties

/etc/hadoop/conf.pseudo.mr1/hdfs-site.xml

/etc/hadoop/conf.pseudo.mr1/log4j.properties

/etc/hadoop/conf.pseudo.mr1/mapred-site.xml

/var/lib/hadoop

/var/lib/hadoop/cache

/var/lib/hdfs

/var/lib/hdfs/cache

Starting up Hadoop

First we format the NameNode, as directed by the document:

[cloudera@centos-e185 ~]$ sudo -u hdfs hdfs namenode -format

[sudo] password for cloudera:

13/03/13 06:49:13 INFO namenode.NameNode: STARTUP\_MSG:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STARTUP\_MSG: Starting NameNode

STARTUP\_MSG: host = centos-e185.aperture/67.215.65.132

STARTUP\_MSG: args = [-format]

STARTUP\_MSG: version = 2.0.0-cdh4.2.0

(...)

13/03/13 06:49:14 INFO namenode.NNStorage: Storage directory /var/lib/hadoop-hdfs/cache/hdfs/dfs/name has been successfully formatted.

13/03/13 06:49:14 INFO namenode.FSImage: Saving image file /var/lib/hadoop-hdfs/cache/hdfs/dfs/name/current/fsimage.ckpt\_0000000000000000000 using no compression

13/03/13 06:49:15 INFO namenode.FSImage: Image file of size 119 saved in 0 seconds.

13/03/13 06:49:15 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0

13/03/13 06:49:15 INFO util.ExitUtil: Exiting with status 0

13/03/13 06:49:15 INFO namenode.NameNode: SHUTDOWN\_MSG:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SHUTDOWN\_MSG: Shutting down NameNode at centos-e185.aperture/67.215.65.132

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

[cloudera@centos-e185 ~]$

And we start HDFS:

[cloudera@centos-e185 ~]$ for x in `cd /etc/init.d ; ls hadoop-hdfs-\*` ; do sudo service $x start ; done

Starting Hadoop datanode: [ OK ]

starting datanode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-datanode-centos-e185.aperture.out

Starting Hadoop namenode: [ OK ]

starting namenode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-namenode-centos-e185.aperture.out

Starting Hadoop secondarynamenode: [ OK ]

starting secondarynamenode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-secondarynamenode-centos-e185.aperture.out

[cloudera@centos-e185 ~]$

We can verify that the services are operational by checking on the NameNode web console

[cloudera@centos-e185 ~]$ lynx http://localhost:50070



Now we create the appropriate directory on the HDFS environment and set the right access permissions.

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -mkdir /tmp

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -chmod -R 1777 /tmp

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -mkdir -p /var/lib/hadoop-hdfs/cache/mapred/mapred/staging

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -chmod 1777 /var/lib/hadoop-hdfs/cache/mapred/mapred/staging

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -chown -R mapred /var/lib/hadoop-hdfs/cache/mapred

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -ls -R /

drwxrwxrwt - hdfs supergroup 0 2013-03-13 07:14 /tmp

drwxr-xr-x - hdfs supergroup 0 2013-03-13 07:14 /var

drwxr-xr-x - hdfs supergroup 0 2013-03-13 07:14 /var/lib

drwxr-xr-x - hdfs supergroup 0 2013-03-13 07:14 /var/lib/hadoop-hdfs

drwxr-xr-x - hdfs supergroup 0 2013-03-13 07:14 /var/lib/hadoop-hdfs/cache

drwxr-xr-x - mapred supergroup 0 2013-03-13 07:14 /var/lib/hadoop-hdfs/cache/mapred

drwxr-xr-x - mapred supergroup 0 2013-03-13 07:14 /var/lib/hadoop-hdfs/cache/mapred/mapred

drwxrwxrwt - mapred supergroup 0 2013-03-13 07:14 /var/lib/hadoop-hdfs/cache/mapred/mapred/staging

[cloudera@centos-e185 ~]$

And we start the MapReduce services:

[cloudera@centos-e185 ~]$ for x in `cd /etc/init.d ; ls hadoop-0.20-mapreduce-\*` ; do sudo service $x start ; done

Starting Hadoop jobtracker daemon (hadoop-jobtracker): starting jobtracker, logging to /var/log/hadoop-0.20-mapreduce/hadoop-hadoop-jobtracker-centos-e185.aperture.out

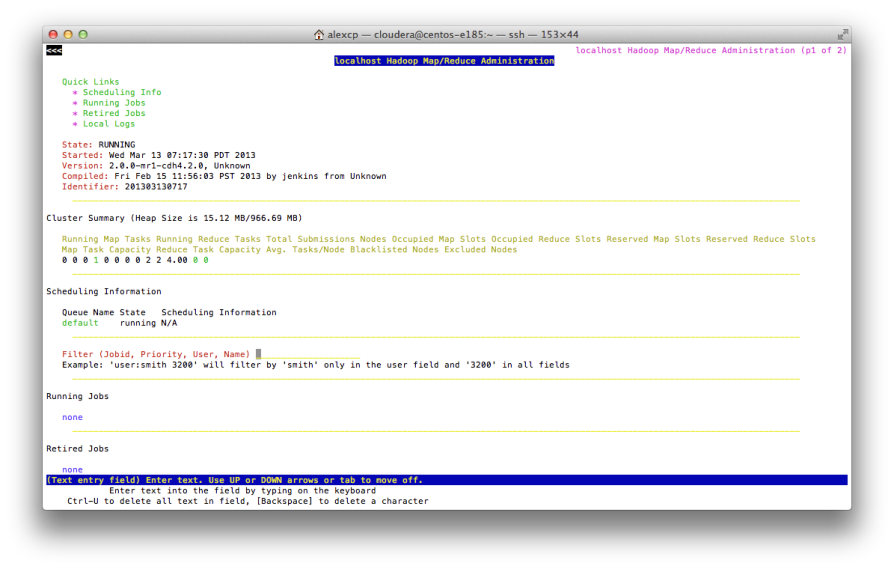
Starting Hadoop tasktracker daemon (hadoop-tasktracker): starting tasktracker, logging to /var/log/hadoop-0.20-mapreduce/hadoop-hadoop-tasktracker-centos-e185.aperture.out

[cloudera@centos-e185 ~]$ [ OK ]

[cloudera@centos-e185 ~]$

We can verify the services are up and running on the JobTracker console:

[cloudera@centos-e185 ~]$ lynx http://localhost:50030/



Finally, we create home directories for our 'cloudera' and 'joe' users:

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -mkdir /user/cloudera

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -chown cloudera /user/cloudera

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -mkdir /user/joe

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -chown joe /user/joe

Running an example application with user 'joe'

To run the example application, we log into the machine with the user 'joe' and copy the input files to a local directory called input

aperture:~ alexcp$ ssh joe@192.168.2.138

joe@192.168.2.138's password:

[joe@centos-e185 ~]$ hadoop fs -mkdir input

[joe@centos-e185 ~]$ hadoop fs -put /etc/hadoop/conf/\*.xml input

[joe@centos-e185 ~]$ hadoop fs -ls input

Found 3 items

-rw-r--r-- 1 joe supergroup 1461 2013-03-13 07:30 input/core-site.xml

-rw-r--r-- 1 joe supergroup 1854 2013-03-13 07:30 input/hdfs-site.xml

-rw-r--r-- 1 joe supergroup 1001 2013-03-13 07:30 input/mapred-site.xml

[joe@centos-e185 ~]$

We run a simple MapReduce job on the input data:

[joe@centos-e185 ~]$ /usr/bin/hadoop jar /usr/lib/hadoop-0.20-mapreduce/hadoop-examples.jar grep input output 'dfs[a-z.]+'

13/03/13 09:50:49 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.

13/03/13 09:50:49 INFO mapred.FileInputFormat: Total input paths to process : 3

13/03/13 09:50:50 INFO mapred.JobClient: Running job: job\_201303130939\_0001

(...)

13/03/13 09:51:18 INFO mapred.JobClient: BYTES\_READ=234

[joe@centos-e185 ~]$

And we check the results it returned successfully:

[joe@centos-e185 ~]$ hadoop fs -ls

Found 2 items

drwxr-xr-x - joe supergroup 0 2013-03-13 07:30 input

drwxr-xr-x - joe supergroup 0 2013-03-13 09:51 output

[joe@centos-e185 ~]$ hadoop fs -ls output

Found 3 items

-rw-r--r-- 1 joe supergroup 0 2013-03-13 09:51 output/\_SUCCESS

drwxr-xr-x - joe supergroup 0 2013-03-13 09:51 output/\_logs

-rw-r--r-- 1 joe supergroup 150 2013-03-13 09:51 output/part-00000

[joe@centos-e185 ~]$ hadoop fs -cat output/part-00000 | head

1 dfs.datanode.data.dir

1 dfs.namenode.checkpoint.dir

1 dfs.namenode.name.dir

1 dfs.replication

1 dfs.safemode.extension

1 dfs.safemode.min.datanodes

[joe@centos-e185 ~]$

Installing MapReduce YARN (MRv2)

First of all we need to uninstall the packages that we installed for MRv1. We first stop the services:

[joe@centos-e185 ~]$ for x in `cd /etc/init.d ; ls hadoop-hdfs-\*` ; do sudo service $x stop ; done

Stopping Hadoop datanode: [ OK ]

stopping datanode

Stopping Hadoop namenode: [ OK ]

stopping namenode

Stopping Hadoop secondarynamenode: [ OK ]

stopping secondarynamenode

[joe@centos-e185 ~]$ for x in 'cd /etc/init.d ; ls hadoop-0.20-mapreduce-\* ; do sudo service $x stop ; done

> ^C

[joe@centos-e185 ~]$ for x in `cd /etc/init.d ; ls hadoop-0.20-mapreduce-\*` ; do sudo service $x stop ; done

Stopping Hadoop jobtracker daemon (hadoop-jobtracker): stopping jobtracker

Stopping Hadoop tasktracker daemon (hadoop-tasktracker): stopping tasktracker

[joe@centos-e185 ~]$ [ OK ]

And we uninstall the services using yum:

[joe@centos-e185 ~]$ sudo yum remove hadoop-0.20-conf-pseudo hadoop-0.20-mapreduce-\*

Loaded plugins: fastestmirror, presto

Setting up Remove Process

Resolving Dependencies

(...)

Removed:

hadoop-0.20-conf-pseudo.x86\_64 0:0.20.2+1341-1.cdh4.2.0.p0.21.el6 hadoop-0.20-mapreduce.x86\_64 0:0.20.2+1341-1.cdh4.2.0.p0.21.el6

hadoop-0.20-mapreduce-jobtracker.noarch 0:0.20.2+1341-1.cdh4.2.0.p0.21.el6 hadoop-0.20-mapreduce-tasktracker.noarch 0:0.20.2+1341-1.cdh4.2.0.p0.21.el6

Complete!

[joe@centos-e185 ~]$

We also clean up the cache directory for good measure, deleting all in the /var/lib/hadoop-hdfs/cache directory before we continue.

[cloudera@centos-e185 ~]$ cd /var/lib/hadoop-hdfs/cache/

[cloudera@centos-e185 cache]$ ls -l

total 8

drwxr-xr-x. 3 hdfs hdfs 4096 Mar 13 06:49 hdfs

drwxr-xr-x. 3 mapred mapred 4096 Mar 13 07:17 mapred

[cloudera@centos-e185 cache]$ sudo rm -rf \*

[cloudera@centos-e185 cache]$ ls -l

total 0

Now that everything is uninstalled, we install the YARN packages:

[cloudera@centos-e185 ~]$ sudo yum install hadoop-conf-pseudo

[sudo] password for cloudera:

Loaded plugins: fastestmirror, presto

Loading mirror speeds from cached hostfile

\* base: linux.mirrors.es.net

\* extras: mirror.hmc.edu

\* updates: centos.sonn.com

Setting up Install Process

(...)

Installed:

hadoop-conf-pseudo.x86\_64 0:2.0.0+922-1.cdh4.2.0.p0.12.el6

Dependency Installed:

hadoop-mapreduce.x86\_64 0:2.0.0+922-1.cdh4.2.0.p0.12.el6 hadoop-mapreduce-historyserver.x86\_64 0:2.0.0+922-1.cdh4.2.0.p0.12.el6

hadoop-yarn.x86\_64 0:2.0.0+922-1.cdh4.2.0.p0.12.el6 hadoop-yarn-nodemanager.x86\_64 0:2.0.0+922-1.cdh4.2.0.p0.12.el6

hadoop-yarn-resourcemanager.x86\_64 0:2.0.0+922-1.cdh4.2.0.p0.12.el6

Complete!

[cloudera@centos-e185 ~]$ rpm -ql hadoop-conf-pseudo

/etc/hadoop/conf.pseudo

/etc/hadoop/conf.pseudo/README

/etc/hadoop/conf.pseudo/core-site.xml

/etc/hadoop/conf.pseudo/hadoop-env.sh

/etc/hadoop/conf.pseudo/hadoop-metrics.properties

/etc/hadoop/conf.pseudo/hdfs-site.xml

/etc/hadoop/conf.pseudo/log4j.properties

/etc/hadoop/conf.pseudo/mapred-site.xml

/etc/hadoop/conf.pseudo/yarn-site.xml

Then, we format the NameNode:

[cloudera@centos-e185 ~]$ sudo -u hdfs hdfs namenode -format

13/03/13 11:24:40 INFO namenode.NameNode: STARTUP\_MSG:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

STARTUP\_MSG: Starting NameNode

STARTUP\_MSG: host = centos-e185.aperture/127.0.0.1

STARTUP\_MSG: args = [-format]

STARTUP\_MSG: version = 2.0.0-cdh4.2.0

(...)

13/03/13 11:24:46 INFO namenode.NNStorage: Storage directory /var/lib/hadoop-hdfs/cache/hdfs/dfs/name has been successfully formatted.

13/03/13 11:24:46 INFO namenode.FSImage: Saving image file /var/lib/hadoop-hdfs/cache/hdfs/dfs/name/current/fsimage.ckpt\_0000000000000000000 using no compression

13/03/13 11:24:46 INFO namenode.FSImage: Image file of size 119 saved in 0 seconds.

13/03/13 11:24:46 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0

13/03/13 11:24:46 INFO util.ExitUtil: Exiting with status 0

13/03/13 11:24:46 INFO namenode.NameNode: SHUTDOWN\_MSG:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SHUTDOWN\_MSG: Shutting down NameNode at centos-e185.aperture/127.0.0.1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

[cloudera@centos-e185 ~]$

And we start the HDFS services up and we connect to the NameNode console page to verify everything is working properly.

[cloudera@centos-e185 ~]$ for x in `cd /etc/init.d ; ls hadoop-hdfs-\*` ; do sudo service $x start ; done

Starting Hadoop datanode: [ OK ]

starting datanode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-datanode-centos-e185.aperture.out

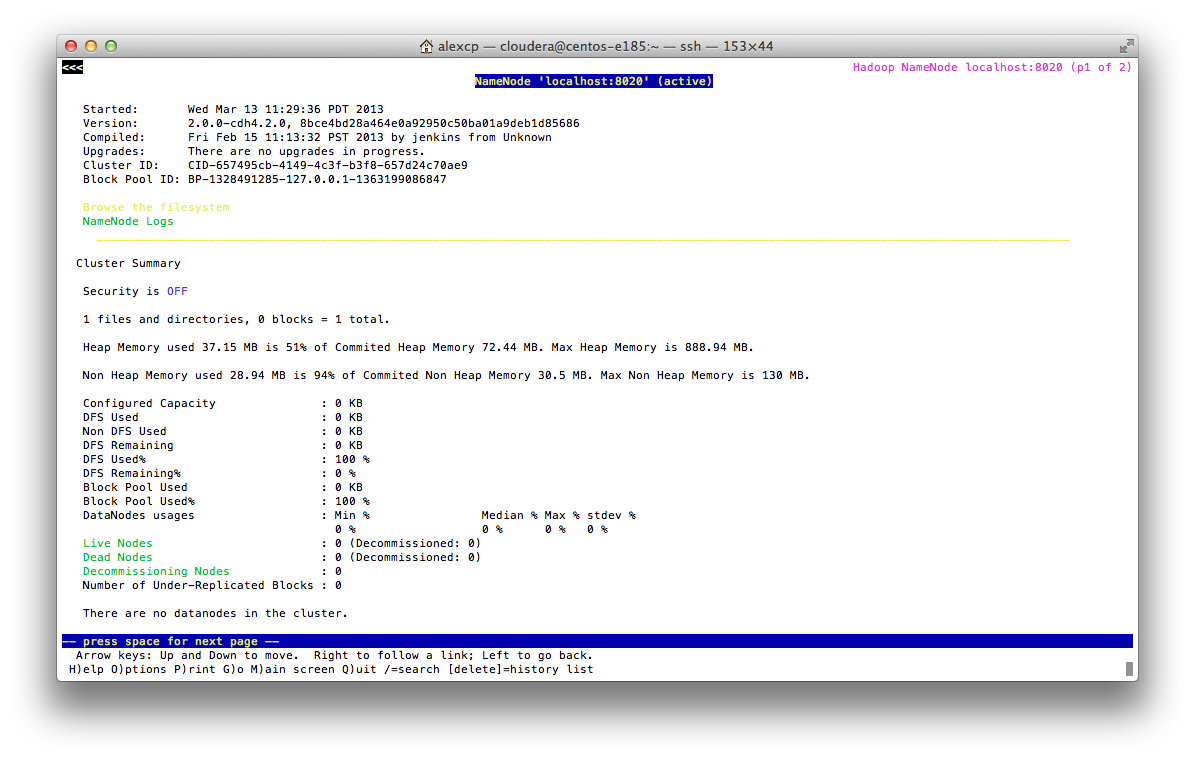
Starting Hadoop namenode: [ OK ]

starting namenode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-namenode-centos-e185.aperture.out

Starting Hadoop secondarynamenode: [ OK ]

starting secondarynamenode, logging to /var/log/hadoop-hdfs/hadoop-hdfs-secondarynamenode-centos-e185.aperture.out

[cloudera@centos-e185 ~]$ lynx http://localhost:50070/



Then we create the /tmp and staging directories in accordance with the document:

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -rm -r /tmp

rm: `/tmp': No such file or directory

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -mkdir /tmp

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -chmod -R 1777 /tmp

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -ls /

Found 1 items

drwxrwxrwt - hdfs supergroup 0 2013-03-13 11:34 /tmp

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -mkdir /tmp/hadoop-yarn/staging

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -chmod -R 1777 /tmp/hadoop-yarn/staging

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -mkdir /tmp/hadoop-yarn/staging/history/done\_intermediate

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -chmod -R 1777 /tmp/hadoop-yarn/staging/history/done\_intermediate

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -chown -R mapred:mapred /tmp/hadoop-yarn/staging

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -mkdir /var/log/hadoop-yarn

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -chown yarn:mapred /var/log/hadoop-yarn

[cloudera@centos-e185 ~]$

We just check that everything is up and running:

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -ls -R /

drwxrwxrwt - hdfs supergroup 0 2013-03-13 11:42 /tmp

drwxr-xr-x - hdfs supergroup 0 2013-03-13 11:41 /tmp/hadoop-yarn

drwxrwxrwt - mapred mapred 0 2013-03-13 11:43 /tmp/hadoop-yarn/staging

drwxr-xr-x - mapred mapred 0 2013-03-13 11:43 /tmp/hadoop-yarn/staging/history

drwxrwxrwt - mapred mapred 0 2013-03-13 11:43 /tmp/hadoop-yarn/staging/history/done\_intermediate

drwxr-xr-x - hdfs supergroup 0 2013-03-13 11:44 /var

drwxr-xr-x - hdfs supergroup 0 2013-03-13 11:44 /var/log

drwxr-xr-x - yarn mapred 0 2013-03-13 11:44 /var/log/hadoop-yarn

[cloudera@centos-e185 ~]$

And we start-up the YARN services:

[cloudera@centos-e185 ~]$ sudo service hadoop-yarn-resourcemanager start

Starting Hadoop resourcemanager: [ OK ]

starting resourcemanager, logging to /var/log/hadoop-yarn/yarn-yarn-resourcemanager-centos-e185.aperture.out

[cloudera@centos-e185 ~]$ sudo service hadoop-yarn-nodemanager start

Starting Hadoop nodemanager: [ OK ]

starting nodemanager, logging to /var/log/hadoop-yarn/yarn-yarn-nodemanager-centos-e185.aperture.out

[cloudera@centos-e185 ~]$ sudo service hadoop-mapreduce-historyserver start

Starting Hadoop historyserver: [ OK ]

starting historyserver, logging to /var/log/hadoop-mapreduce/yarn-mapred-historyserver-centos-e185.aperture.out

[cloudera@centos-e185 ~]$

Finally, we create the user directories on the HDFS structure:

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -mkdir /user/cloudera

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -chown cloudera /user/cloudera

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -mkdir /user/joe

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -chown joe /user/joe

[cloudera@centos-e185 ~]$ sudo -u hdfs hadoop fs -ls /user

Found 2 items

drwxr-xr-x - cloudera supergroup 0 2013-03-13 11:54 /user/cloudera

drwxr-xr-x - joe supergroup 0 2013-03-13 11:55 /user/joe

[cloudera@centos-e185 ~]$

Running an YARN example application with user 'joe'

Once again, we ask our user 'joe' to help out with an example. We first create the input directory with the XML files:

[joe@centos-e185 ~]$ hadoop fs -mkdir input

[joe@centos-e185 ~]$ hadoop fs -put /etc/hadoop/conf/\*.xml input

[joe@centos-e185 ~]$ hadoop fs -ls input

Found 4 items

-rw-r--r-- 1 joe supergroup 1461 2013-03-13 12:50 input/core-site.xml

-rw-r--r-- 1 joe supergroup 1854 2013-03-13 12:50 input/hdfs-site.xml

-rw-r--r-- 1 joe supergroup 1325 2013-03-13 12:50 input/mapred-site.xml

-rw-r--r-- 1 joe supergroup 2262 2013-03-13 12:50 input/yarn-site.xml

And we run the MapReduce job:

[joe@centos-e185 ~]$ export HADOOP\_MAPRED\_HOME=/usr/lib/hadoop-mapreduce

[joe@centos-e185 ~]$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar grep input output23 'dfs[a-z.]+'

13/03/13 12:54:05 INFO service.AbstractService: Service:org.apache.hadoop.yarn.client.YarnClientImpl is inited.

13/03/13 12:54:05 INFO service.AbstractService: Service:org.apache.hadoop.yarn.client.YarnClientImpl is started.

13/03/13 12:54:05 WARN mapreduce.JobSubmitter: No job jar file set. User classes may not be found. See Job or Job#setJar(String).

(...)

Shuffle Errors

BAD\_ID=0

CONNECTION=0

IO\_ERROR=0

WRONG\_LENGTH=0

WRONG\_MAP=0

WRONG\_REDUCE=0

File Input Format Counters

Bytes Read=320

File Output Format Counters

Bytes Written=150

[joe@centos-e185 ~]$

And we finally check our output as correct:

[joe@centos-e185 ~]$ hadoop fs -ls

Found 2 items

drwxr-xr-x - joe supergroup 0 2013-03-13 12:50 input

drwxr-xr-x - joe supergroup 0 2013-03-13 12:54 output23

[joe@centos-e185 ~]$ hadoop fs -ls output23

Found 2 items

-rw-r--r-- 1 joe supergroup 0 2013-03-13 12:54 output23/\_SUCCESS

-rw-r--r-- 1 joe supergroup 150 2013-03-13 12:54 output23/part-r-00000

[joe@centos-e185 ~]$ hadoop fs -cat output23/part-r-00000 | head

1 dfs.safemode.min.datanodes

1 dfs.safemode.extension

1 dfs.replication

1 dfs.namenode.name.dir

1 dfs.namenode.checkpoint.dir

1 dfs.datanode.data.dir