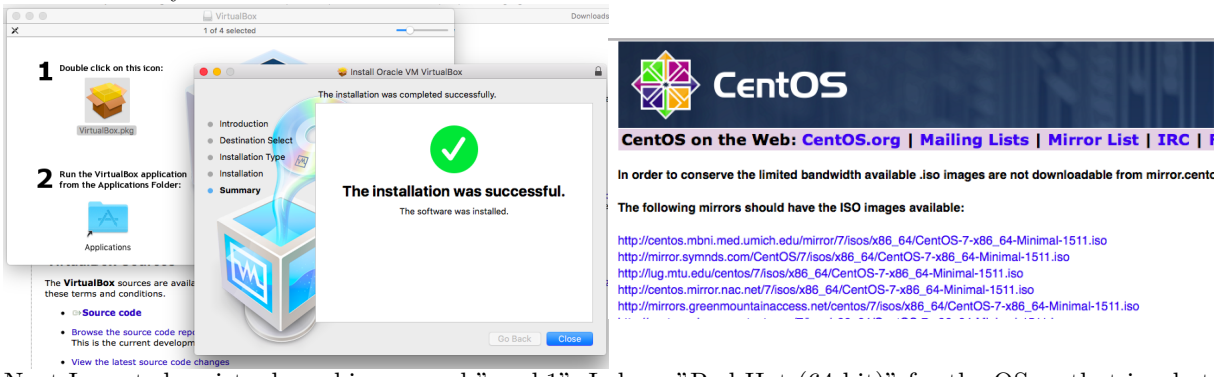
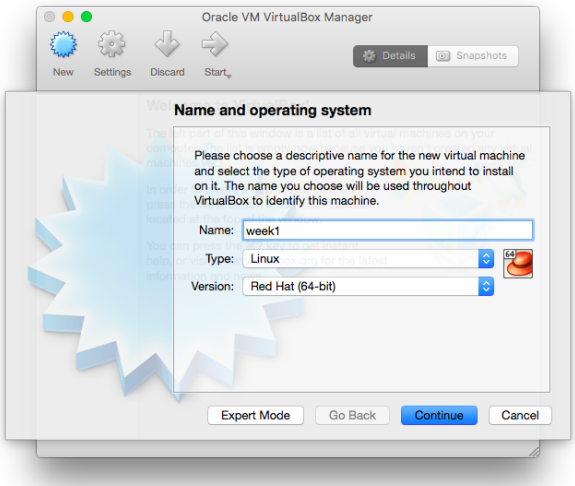
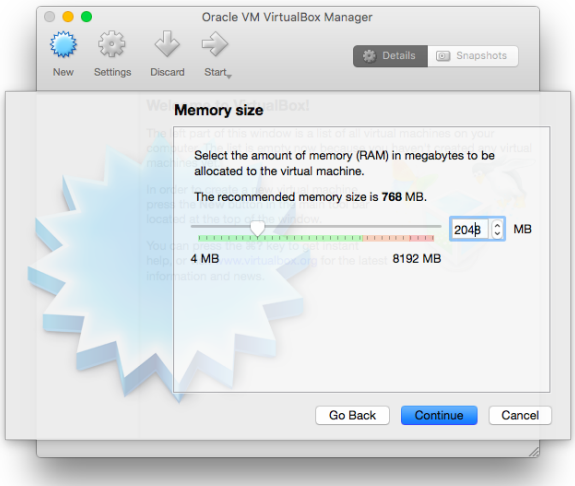
MSDS610 Week 1 Install Hadoop Assignment - Nathan Worsham

Following the instructions for this week, I started with downloading VirtualBox (currently using a Mac OSX) and CentOS 7 minimal install iso. I chose to stick with minimal install because I am comfortable with using command line only on a machine.

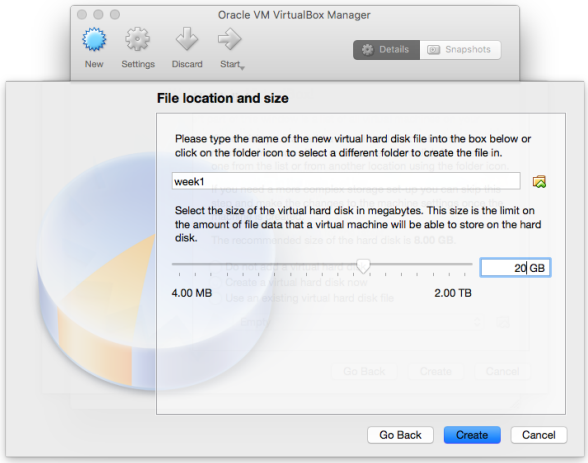


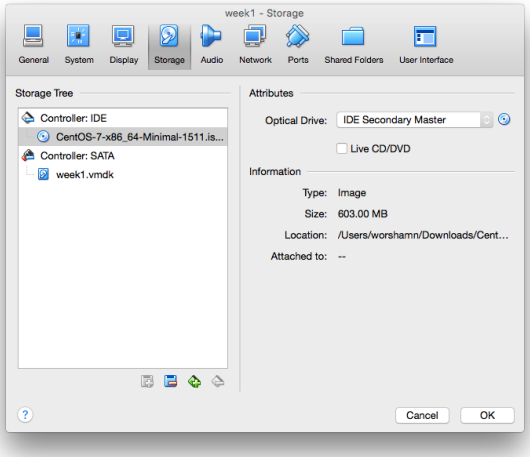
Next I created a virtual machine named "week1", I chose "Red Hat (64-bit)" for the OS as that is what CentOS is based on and gave it 2 GB of RAM:



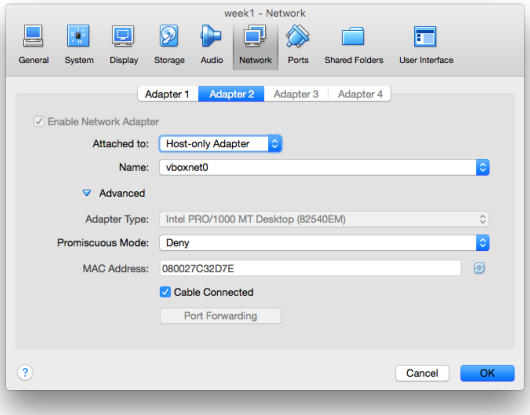


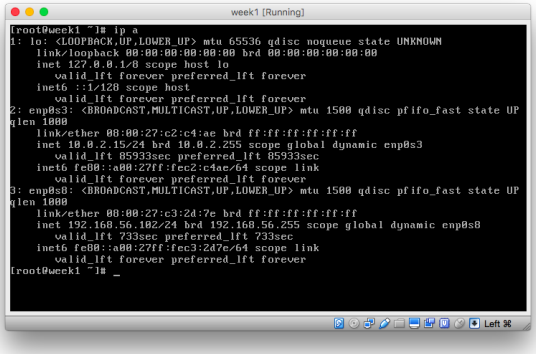
I chose to use the VMDK format for the disk in case I need to move this to VMWare Player (on my Windows machine) gave the hard drive the 20GB of recommended space. I then enabled the CentOS minimal install ISO to be virtually placed in the cd drive.



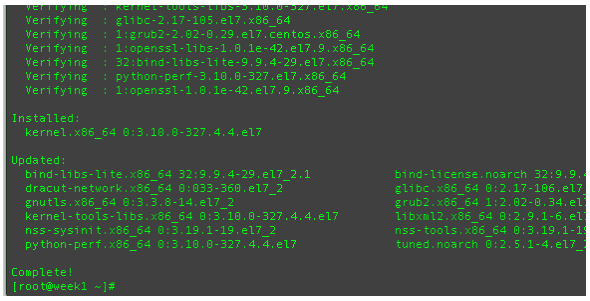


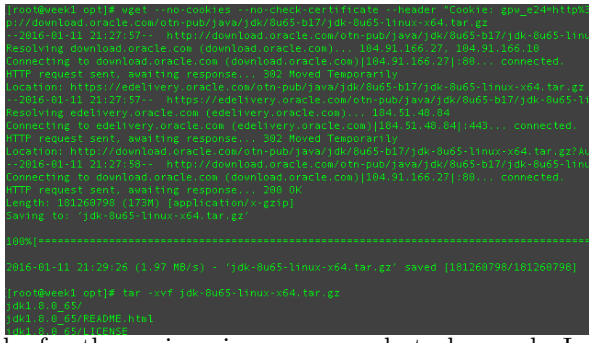
Logging into the computer was not an issue, but I did run into an issue with what "networking mode" I should use. When I am at home doing the exercises running in Bridged mode, it is not an issue and I can easily both SSH to the virtual machine from the "host" (my Mac laptop) and connect to the internet from the virtual machine. But when I am at my work trying to do the exercises I run into issues. This is because of security in place at my company–802.1x on the wired network, and EAP/TLS on the wireless network. So to allow myself to work on these exercises while at work (on my break time, mind you), I needed to change the configuration. Researching what the different mode options meant from Virtualbox.org (n.d.), I realized I would need to use 2 adapters to accomplish being able to both SSH to the VM and allow the VM out to the Internet given the security constraints. First to get the VM out to the internet I used NAT mode. This hides the VM's IP address behind the host's IP address, but the problem here is that this makes the VM invisible to the host. To solve this I had to add a second adapter running in "Host-Only" mode. To enable this I first had to create a virtual adapter that is a new loopback interface on the host and then assign that adapter the mode. Now I was able to ssh to the VM and get the VM out onto the internet.





As instructed, I got the latest updates for the OS and then installed wget. Next I was able to use wget to get the java install. I then un-tar'd the file and used `chown` to change the ownership–though I believe root may have already owned it.





I used the `alternatives` command to create symlinks for the various java commands to be used. In my experience, I had only ever just manually created links (with `ln -s`) for programs such as java, I have never even known about the `alternatives` command. It looks like the command keeps basically a note or metadata about the symlink created. When I ran the command

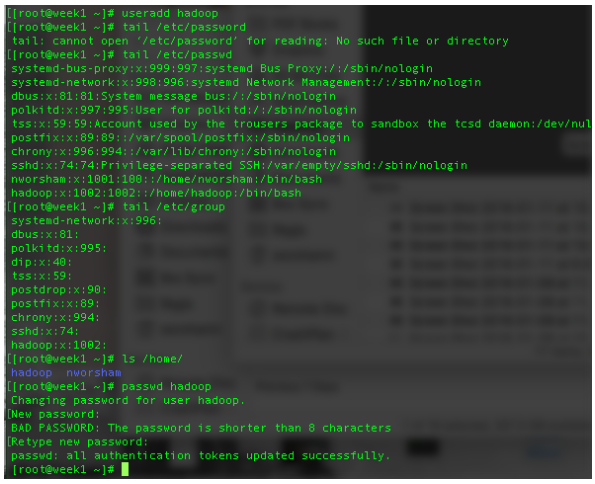
```
alternatives --install /usr/bin/javac javac /opt/jdk1.8.0_65/bin/javac 2
```

I ran into this error:

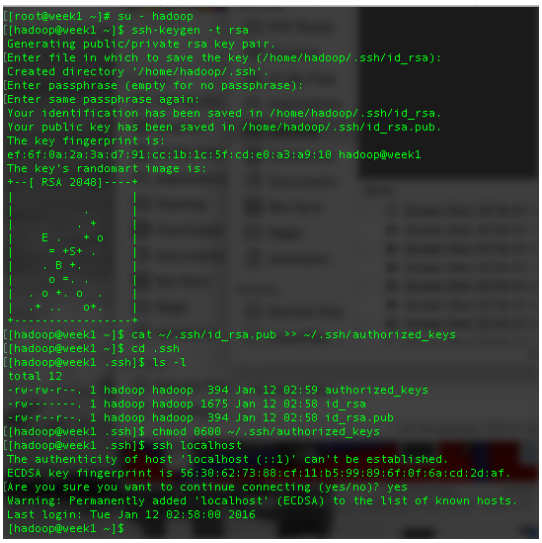
```
/opt/jdk1.8.0_65/bin/javac has not been configured as an alternative for javac
```

So I carefully typed out both commands again, this time getting no output, meaning success. Looking through my history using the up arrow, I see an extra character got into my command. In the next section the exercise asks us to create the hadoop user and then later create ssh keys. I am used to creating a user with a home directory to begin with so I did not realize that running just `useradd hadoop` is enough to

do so. After running the command I looked at both /etc/passwd and /etc/group files to also find out that the command by default creates a group named hadoop and assigns it to the user. I then gave the user a password.



Again I am used to setting up ssh keys and copying the public key around for connecting without a password (useful for scripts) from other servers/workstations, but I admit I am a bit confused by copying the public key to the id's own authorized_keys file. I suppose this is just because this is setting up for another exercise so that perhaps the entire file can be copied is my best guess.



I am very comfortable with VI as my editor as I use it day in and day out, mostly because you can always find it installed anywhere. However I went ahead and installed nano and tried it, I can see where someone who is not comfortable or confused by how VI works would like this as a basic editor, but that being said I went back to VI and edited the `.bashrc` profile and completed the remainder of the part 1 assignment.

```
[[hadoop@week1 ~]$ nano .bashrc
[[hadoop@week1 ~]$ vi .bashrc
[[hadoop@week1 ~]$ tail .bashrc
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
export JAVA_HOME=/opt/jdk1.8.0_65
export JRE_HOME=/opt/jdk1.8.0_65/jre
export PATH=$PATH:$JAVA_HOME/bin
[[hadoop@week1 ~]$ . .bashrc
[[hadoop@week1 ~]$ printenv PATH
/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/hadoop/.local/bin:/home/hadoop/bin:/opt/jdk1.8.0_65/bin
[hadoop@week1 ~]$
```

Starting on part 2, I downloaded hadoop and unpacked it. I changed the folder name to `hadoop` but wonder a bit why we wouldn't just set a link or use the alternatives command we previously used. I then set the environment variables:

```
[[hadoop@week1 ~]$ vi .bashrc
[[hadoop@week1 ~]$ source .bashrc
[[hadoop@week1 ~]$ printenv PATH
/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/opt/jdk1.8.0_65/bin:/home/hadoop/.local/bin:/home/hadoop/bin:/opt/jdk1.8.0_65/bin:/home/hadoop/hadoop/sbin
:/home/hadoop/hadoop/bin
[hadoop@week1 ~]$
```

Next I used the newly created environment variable `$HADOOP_HOME`, listed directory contents, and updated the `hadoop-env.sh` script to edit its `JAVA_HOME`.

```
[[hadoop@week1 hadoop]$ cd $HADOOP_HOME/etc/hadoop
[[hadoop@week1 hadoop]$ ls
capacity-scheduler.xml  hadoop-env.sh               httpfs-env.sh            kms-env.sh            mapred-env.sh              ssl-server.xml.example
configuration.xsl       hadoop-metrics2.properties  httpfs-log4j.properties  kms-log4j.properties  mapred-queues.xml.template  yarn-env.cmd
container-executor.cfg  hadoop-metrics.properties   httpfs-signature.secret  kms-site.xml          mapred-site.xml.template    yarn-env.sh
core-site.xml           hadoop-policy.xml           httpfs-site.xml          log4j.properties      slaves                      yarn-site.xml
hadoop-env.cmd          hdfs-site.xml               kms-acls.xml             mapred-env.cmd        ssl-client.xml.example
[[hadoop@week1 hadoop]$ vi hadoop-env.sh
[[hadoop@week1 hadoop]$ cat hadoop-env.sh |grep "export JAVA_HOME"
export JAVA_HOME=/opt/jdk1.8.0_65/
```

After editing the four Hadoop config files, when I tried to run the `hdfs namenode -format` command, I received several errors about the host:

```
STARTUP_MSG:   host = java.net.UnknownHostException: week1: week1: unknown error
...
16/01/12 18:06:18 WARN net.DNS: Unable to determine address of the host-falling back to
 "localhost" address
java.net.UnknownHostException: week1: week1: unknown error
...
SHUTDOWN_MSG: Shutting down NameNode at java.net.UnknownHostException: week1:
week1: unknown error
```

Off to Google I went with my error message and while Stackoverflow.com (n.d.) did not have the exact answer, it had a similar enough article to make me realize what I needed to do. I'm thinking (but only after the fact) that because I had returned to the exercise after a period of time and did not ssh to localhost from the hadoop user or because I gave my VM a hostname. The error was using my VM's hostname, but since the Stackoverflow.com article pointed at the /etc/hosts file needing a listing for 127.0.0.1 I went there. The file already had the required `localhost` listing, so I just added my VM's hostname (`week1`) to 127.0.0.1 and then the command completed correctly.

```
[hadoop@week1 ~]$ hdfs namenode -format
16/01/12 18:15:30 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = localhost/127.0.0.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.7.1
```

```
16/01/12 18:15:33 INFO common.Storage: Storage directory /home/hadoop/hadoop
16/01/12 18:15:33 INFO namenode.NNStorageRetentionManager: Going to retain 1
16/01/12 18:15:33 INFO util.ExitUtil: Exiting with status 0
16/01/12 18:15:33 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at localhost/127.0.0.1
************************************************************/
```

Started hdfs and yarn.

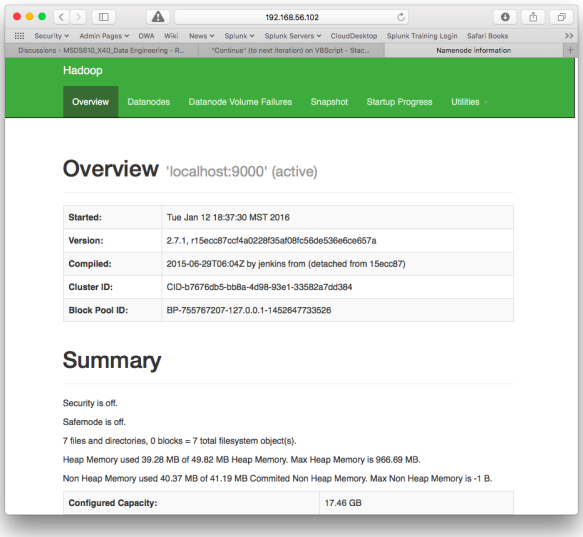
```
[hadoop@week1 sbin]$ start-dfs.sh
16/01/12 18:37:26 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/hadoop/hadoop/logs/hadoop-hadoop-namenode-week1.out
localhost: starting datanode, logging to /home/hadoop/hadoop/logs/hadoop-hadoop-datanode-week1.out
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is 56:30:62:73:88:cf:11:b5:99:89:6f:0f:6a:cd:2d:af.
Are you sure you want to continue connecting (yes/no)? yes
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: starting secondarynamenode, logging to /home/hadoop/hadoop/logs/hadoop-hadoop-secondarynamenode-
16/01/12 18:37:52 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
[hadoop@week1 sbin]$ ./start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /home/hadoop/hadoop/logs/yarn-hadoop-resourcemanager-week1.out
localhost: starting nodemanager, logging to /home/hadoop/hadoop/logs/yarn-hadoop-nodemanager-week1.out
```

Started job history server, and confirmed processes with `jps`.

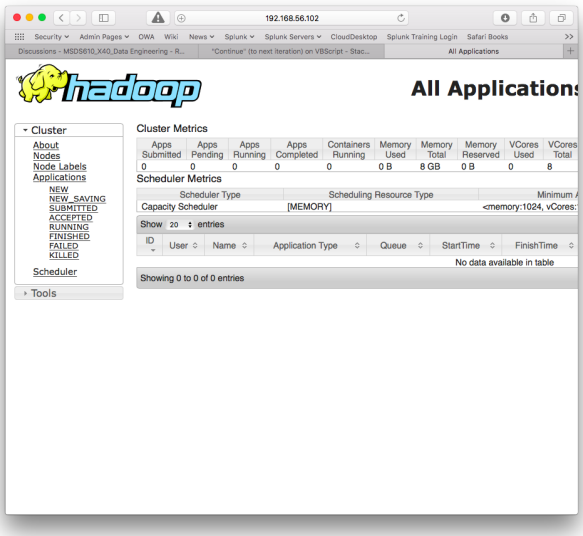
```
[hadoop@week1 sbin]$ ./mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/hadoop/hadoop/logs/mapred-hadoop-historyserver-week1.out
[hadoop@week1 sbin]$ jps
7920 SecondaryNameNode
8065 ResourceManager
7731 DataNode
7637 NameNode
8166 NodeManager
8522 JobHistoryServer
8588 Jps
```

Trying to run the `firewall-cmd` command, realized firewalld was not installed by both using the command `service firewall status` and opening up the HDFS web interface without configuring anything first.

```
[root@week1 ~]# service firewalld status
Redirecting to /bin/systemctl status  firewalld.service
● firewalld.service
   Loaded: not-found (Reason: No such file or directory)
   Active: inactive (dead)
```



I went ahead and installed firewalld anyway and confirmed I could no longer get to the web interface for HDFS. So now I opened the ports, however I just used the `--permanent` option and then used `firewall-cmd --reload` to make those changes active. I could then once again get to the HDFS web interface and also tried the YARN interface.

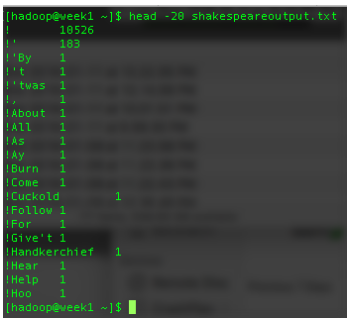


Last I tested the Hadoop installation but first I went through a Stackoverflow.com (2015) article's solutions to try to figure out how to get rid of the "native-hadoop library" warnings. I finally found that adding the following line to my `.bashrc` file fixed the issue for me.

```
export JAVA_LIBRARY_PATH=$HADOOP_HOME/lib/native:$JAVA_LIBRARY_PATH
```

Showing `head` of the final output file to show I went through the commands. I found on the line about `hadoop-mapreduce-examples-2.7.1.jar`, I had to use the find command to see where this file was located because it gave an error message of `Not a valid JAR: /home/hadoop/hadoop-mapreduce-examples-2.7.1.jar`. Once I gave it the full path of the file it then worked–

```
/home/hadoop/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar
```



References

Virtualbox.org, n.d. Chapter 6. Virtual Networking. Retrieved from https://www.virtualbox.org/manual/ch06.html#networkingmodes

Stackoverflow.com, 2014. Retrieved from http://stackoverflow.com/questions/24517593/why-hadoop-format-give-out-java-net-unknownhostexception-exception

Stackoverflow.com, 2015. Retrieved from http://stackoverflow.com/questions/19943766/hadoop-unable-to-load-native-hadoop-library-for-your-platform-warning