Hadoop Single Node Cluster Preparation

Case Study by Arunava Chakraborty
Overview

I was searching commands for easy to build Hadoop lab in local machine and found some command reference. Hence thought of presenting the same as a hands on lab.

In this test project we will do the following:

- Setup a hypervisor to run a linux virtual machine to host lab machine
- Build a linux appliance to build Hadoop lab
- Install and configure Java
- Install and configure a single node Hadoop instance

First up, here are some basic requirements to build our test bed:

- A personal computer or server of some form.
- A reasonably powerful x86 hardware (a recent Intel or AMD processor - an Intel-based Windows PC, Intel-based Mac or Intel-based Linux machine with at least 2 GB of RAM and 2 GB of Hard Drive space free.

Note: We are going to be run a full virtual computer on top of base operating system in computer, so we need to consider the performance impact, i.e. it could potentially slow our PC down a little while we are running the Hadoop VM under VirtualBox.
1. Download VM Tool and Virtual Appliances

First download the following two key components:

**Virtualbox:**

Download Virtualbox from:

https://www.virtualbox.org/wiki/Downloads

**Linux virtual appliance:**

This is a tiny Linux system “appliance” virtual machine we’ll use to install and run Hadoop on.

We will be importing this self configuring Linux appliance with Virtualbox to build the linux virtual machine (VM) we need to start from.

Download base linux virtual machine OVF (from TurnKey Linux):

http://mirror.turnkeylinux.org/turnkeylinux/images/ovf/

Look for turnkey-code 64 bit zip and save the file to a folder where you will setup your Hadoop test bed

- turnkey-core-13.0-wheezy-i386-ovf.zip

Note: this will expand to a folder called: turnkey-core-13.0-wheezy-i386
2. Install Virtual Machine

Install the Virtualbox Hypervisor:

The installation of Virtualbox is very simple, just locate the installer you downloaded, open it (i.e. double click on it), and follow the prompts.

Under Windows simply double click the download and it will lead you from there.

Under Linux and Mac OS X, you need to open the downloaded disk image or TAR file, and run the installer from within.

Follow the prompts, defaults will do what we need, you do not need to change anything during the install.

Simply double-click the base installer, follow the prompts and accept all the defaults, and in a few minutes you will have a full working version of Virtualbox installed and ready to run and import your Linux appliance.
Install and configure the base Linux VM:

The set of the Linux virtual machine is a little more detailed but the key steps are pretty straightforward.

From the main "File" menu select "Import Appliance"

- A new window will open titled "Appliance to import"
- Click on "Open appliance" button
- Navigate to the "turnkey-core-13.0-wheezy-i386" folder
- Select the file "turnkey-core-13.0-wheezy-i386.ovf" and click "Open"
- Click "Continue/Next"
- Click "Import"

![VirtualBox Manager](image)

We now need to change a few settings

- Right click on "vm" and select "Settings"
- Rename the VM from "vm" to Hadoop"
- Click on the "system" icon
- Change the "Base memory" from 256 MB to 1024 MB (1 GB)
- In the "Boot order" window unselect "Floppy" and "CD" (leave Hard Disk checked)
- Click on "OK" to save settings
Now you can start up your Hadoop VM.

Double click on the "Hadoop" VM listed as "Powered Off" to start it

Note: you can also single click on the Hadoop VM icon and the click START button

- The Hadoop VM will start up and auto-boot. Boot with first option in GRUB menu by pressing ENTER
You will be prompted for a new "Root Password"
set it to "hadoop" so it's easy to remember
It will ask you for the password twice to confirm you didn't make any typo's
• You are then asked to "Initialize Hub services"
• Press the TAB key to select "Skip" and press ENTER once

• You are then asked to install "Security updates"
• Press the TAB key to select "skip" and press ENTER once
• Your VM will then boot up and be running
• You will have a window displaying URL's you can use to connect to your new VM

Note: this is only your "base" linux OS, we have not installed Hadoop yet. **IP address will vary if different wireless is connected.**

Congratulations, you've successfully installed Virtualbox and imported and configured your Linux appliance.

To confirm you can now connect to your Hadoop virtual machine via a web browser from base operating system, make a note of the IP address displayed on the final screen when your Linux VM finishes booting (it will show up in the URL's on the final screen), and use a web browser to connect that ip address on port 12320 to the built in web shell, i.e if the IP address was 192.168.2.15 then connect to using firefox browser. **You might need to add new http address in security exception list in Firefox or IE browser as a trusted connection.**

http://192.168.2.15:12320

You will be presented with what looks like a terminal console. You can now login using the root user account and password, i.e.: root and hadoop

If this was successful you will now be logged in as the root user with a “#” prompt and you will see a screen similar to the following, and you will be at a prompt that looks like this:
Note: that once you are logged in as root, you are in fact the super user, so tread gently as you have the power to break the system)!!

The first thing we will do is setup a “group” for Hadoop with the following command:

```
root@core ~# addgroup hadoop
```

```
root@core ~# addgroup hadoop
Adding group `hadoop' (GID 1001) ... Done.
root@core ~# 
```

Now we need to add a user for Hadoop with the following command line. Use password as hadoop

```
root@core ~# adduser --ingroup hadoop hduser
```

```
Now add our Hadoop user “hduser” to the sudo group (so it can run commands as root):

```
root@core ~# adduser --ingroup hadoop hduser
Adding user 'hduser' ...
Adding new user 'hduser' (1000) with group 'hadoop' ...
Creating home directory `/home/hduser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hduser
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
```

Now we are going to generate Secure Shell “keys”:

```
root@core ~# ssh-keygen -t rsa -P ""
```
Now add our new public key to the known keys file:

```
root@core ~# cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Lab Case Study by Arunava Chakraborty
Now let's confirm that our new SSH keys work and we can login without entering a password.

The step is also needed to save your local machine's host key fingerprint to the hduser user's known_hosts file.

```
root@core ~# ssh localhost
```

What we've done now is connect to our own system using an SSH public key stored so we don't need to type in our account password – this allows Hadoop to run commands on the system without needing to know or enter the password.

Now exit from the login to your own server with this simple command line:

```
root@core ~# exit
```

You are now ready to proceed to download and install the Oracle Java development kit (JDK) version 7, and the core distribution of Hadoop - we'll be using version 1.2.1.
3. Download Java, Hadoop and FileZilla

Download Java

Search for jdk 7 for linux in Google and go to below url


As we are installing the Java JDK on a Debian based Linux distribution, we will download a 32-bit linux version. Accept license agreement and download jdk-7u79-linux-i586.tar.gz

<table>
<thead>
<tr>
<th>Product/File Description</th>
<th>File Size</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux x86</td>
<td>130.4 MB</td>
<td>jdk-7u79-linux-i586.rpm</td>
</tr>
<tr>
<td>Linux x86</td>
<td>147.8 MB</td>
<td>jdk-7u79-linux-i586.tar.gz</td>
</tr>
<tr>
<td>Linux x64</td>
<td>131.69 MB</td>
<td>jdk-7u79-linux-x64 rpm</td>
</tr>
<tr>
<td>Linux x64</td>
<td>146.4 MB</td>
<td>jdk-7u79-linux-x64.tar.gz</td>
</tr>
<tr>
<td>Mac OS X x64</td>
<td>196.89 MB</td>
<td>jdk-7u79-macosx-x64.dmg</td>
</tr>
<tr>
<td>Solaris x86 (SVR4 package)</td>
<td>140.79 MB</td>
<td>jdk-7u79-solaris-i586.tar.z</td>
</tr>
<tr>
<td>Solaris x86 (SVR4 package)</td>
<td>96.66 MB</td>
<td>jdk-7u79-solaris-i586.tar.gz</td>
</tr>
<tr>
<td>Solaris x64 (SVR4 package)</td>
<td>24.67 MB</td>
<td>jdk-7u79-solaris-x64.tar.z</td>
</tr>
<tr>
<td>Solaris x64 (SVR4 package)</td>
<td>16.38 MB</td>
<td>jdk-7u79-solaris-x64.tar.gz</td>
</tr>
<tr>
<td>Solaris SPARC (SVR4 package)</td>
<td>140 MB</td>
<td>jdk-7u79-solaris-sparc.tar.Z</td>
</tr>
<tr>
<td>Solaris SPARC</td>
<td>99.4 MB</td>
<td>jdk-7u79-solaris-sparc.tar.gz</td>
</tr>
<tr>
<td>Solaris SPARC 64-bit (SVR4 package)</td>
<td>24 MB</td>
<td>jdk-7u79-solaris-sparcv9.tar.z</td>
</tr>
<tr>
<td>Solaris SPARC 64-bit</td>
<td>18.4 MB</td>
<td>jdk-7u79-solaris-sparcv9.tar.gz</td>
</tr>
<tr>
<td>Windows x86</td>
<td>138.31 MB</td>
<td>jdk-7u79-windows-i586.exe</td>
</tr>
<tr>
<td>Windows x64</td>
<td>140.06 MB</td>
<td>jdk-7u79-windows-x64.exe</td>
</tr>
</tbody>
</table>

Download Hadoop

Go to below url and download hadoop-1.2.1.tar.gz

https://www.apache.org/dist/hadoop/core/hadoop-1.2.1/
Download and install SFTP client FileZilla


Transfer downloaded software in /root folder
4. Install Java and Hadoop

Install Java

root@core ~# pwd
=root
root@core ~# ls -l
total 213504
-rw-r--r-- 1 root root 63851630 Sep 14 03:27 hadoop-1.2.1.tar.gz
-rw-r--r-- 1 root root 154773078 Sep 14 03:26 jdk-7u79-linux-i586.tar.gz
root@core ~#

root@core ~# tar -zxvf jdk-7u79-linux-i586.tar.gz

Next we need to move it into the /usr/local directory:

root@core ~# mv jdk1.7.0_79 /usr/local/jdk-7-oracle

Note: we'll add the Java bin directory to our PATH environment variable in a few steps.

Install Hadoop

root@core ~# tar -zxvf hadoop-1.2.1.tar.gz

Now move it to the /usr/local directory with this command line:

root@core ~# mv hadoop-1.2.1 /usr/local

Next, create a softlink for /usr/local/hadoop with this command line:

root@core ~# ln -s /usr/local/hadoop-1.2.1 /usr/local/hadoop
Now we need to setup a couple of environment variables and update our command path.

To do this we need to edit our .bashrc (dot bash rc) file in the root users /home directory and add the following lines (cut and paste them to save typing them in):

```
export HADOOP_HOME=/usr/local/hadoop
export JAVA_HOME=/usr/local/jdk-7-oracle
export PATH=$PATH:$JAVA_HOME/bin
export PATH=$PATH:/usr/local/hadoop/bin
```

If you're familiar with Linux use your editor of choice. I'm a VI user myself, but if you're new to Linux you may way to use the nano editor. VI users will know their way around adding these lines to the .bashrc file. If you use the nano editor, add these extra lines just below the existing PATH setting so it looks like this:

```
Existing PATH setting:

    PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Add these lines below it:

```
export HADOOP_HOME=/usr/local/hadoop
export JAVA_HOME=/usr/local/jdk-7-oracle
export PATH=$PATH:$JAVA_HOME/bin
export PATH=$PATH:/usr/local/hadoop/bin
```

To put these changes into effect in our current shell we need to re-spawn a new shell with the following command:

```
root@core ~# exec bash
```
We can quickly check that our command shell’s PATH environment variable can now find the java and hadoop commands with the following commands.

Check we can find the java command - it should look like this (commands are in bold):

```
root@core ~# which java
/usr/local/jdk-7-oracle/bin/java
```

Next we should confirm the version of Java installed (1.7.0_79-b15):

```
root@core ~# java -version
java version "1.7.0_79"
Java(TM) SE Runtime Environment (build 1.7.0_79-b15)
Java HotSpot(TM) 64-Bit Client VM (build 24.79-b02, mixed mode)
```
5. Configure Hadoop as a single node instance

Hadoop Configuration

You're almost there! Next we need to make a directory for Hadoop to use for storage, which we’ll include in the configuration in the next few steps, change the directory permissions and ownership / group:

```
root@core hadoop/conf# mkdir -p /usr/local/hadoop/tmp
root@core hadoop/conf# chmod 750 /usr/local/hadoop/tmp
root@core hadoop/conf# chown -R hduser.hadoop /usr/local/hadoop/tmp
```

Now we need to make a couple changes to the Hadoop configuration and set it up as a single node instance.

First change into the Hadoop conf directory using this command line:

```
root@core ~# cd /usr/local/hadoop/conf
```

Now we need to make the following changes to the respective files. Use your preferred editor to add / edit the files listed below to include the following lines. You can cut and paste to save having to type it all in manually:

**File: core-site.xml**

```
<configuration>

hadoop.tmp.dir
/usr/local/hadoop/tmp

fs.default.name
hdfs://localhost:9000
</configuration>
```
File: mapred-site.xml

<!--?xml version="1.0"?-->
<!--?xml-stylesheet type="text/xsl" href="configuration.xsl"?-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  mapred.job.tracker
  localhost:9001

  dfs.data.dir
  /usr/local/hadoop/tmp/dfs/data
</configuration>

File: hdfs-site.xml

<!--?xml version="1.0"?-->
<!--?xml-stylesheet type="text/xsl" href="configuration.xsl"?-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  dfs.replication
  1
</configuration>

That's all the configuration changes done! Now for the next step!
Format the Hadoop Distributed File System (HDFS), with the following command:

```
root@core local/hadoop# hadoop namenode -format
```

**Output**

Warning: $HADOOP_HOME is deprecated.

```
15/09/14 04:16:31 INFO namenode.NameNode: STARTUP_MSG

************************************************************/

STARTUP_MSG: StartIng NameNode

STARTUP_MSG: host : core/127.0.1.1

STARTUP_MSG: args : [-format]

STARTUP_MSG: version : 1.2.1


STARTUP_MSG: Java : 1.7.0779

************************************************************/

```

15/09/14 04:16:32 INFO Util.GSet: Computing capacity for map BlockSMap

15/09/14 04:16:32 INFO Util.GSet: VM type : 32-bit

15/09/14 04:16:32 INFO Util.GSet: 2.0% max memory : 1013645312


15/09/14 04:16:32 INFO namenode.FSNamesystem: fsOwner:root

15/09/14 04:16:32 INFO namenode.FSNamesystem: Supergroup:supergroup

15/09/14 04:16:33 INFO namenode.FSNamesystem: IsPermissionEnabled:true

15/09/14 04:16:33 INFO namenode.FSNamesystem: dfs.block.1nvalidate.limit:100

15/09/14 04:16:33 INFO namenode.FSNamesystem: IsAccessTokenEnabled:false accessKeyUpdateInterval:0 min(s), accessToenLifetime:0 min(s)

15/09/14 04:16:33 INFO namenode.FSEditLog: dfs.namenode.edits.toleratIom.length : 0

15/09/14 04:16:33 INFO namenode.NameNode: Caching file names occurring more than 10 times

15/09/14 04:16:33 INFO common.Storage: Image file /tmp/hadoop-root/dfs/name/current/fsimage of size 110 bytes saved in 0 seconds


15/09/14 04:16:33 INFO namenode.FSEditLog: close Success: truncate to 4, editlog:/tmp/hadoop-root/dfs/name/current/edits

15/09/14 04:16:33 INFO common.Storage: Storage directory /tmp/hadoop-root/dfs/name has been Successfully formatted.

15/09/14 04:16:33 INFO namenode.NameNode: SHUTDOWN_MSG

---

Lab Case Study by Arunava Chakraborty
And that’s it – you’re all done! You can now start up your single node Hadoop cluster and check the core components are running as expected.

```
root@core local/hadoop# /usr/local/hadoop/bin/start-all.sh
```
6. Graphical Admin Console : webmin

webmin can be accessed from port 12321 using [https://192.168.2.15:12321/](https://192.168.2.15:12321/)

You must know what you are doing
7. Other VM Operation

Use TAB and ENTER. If clicked in VM, use host key (Right CTRL in VirtualBox) to get the mouse back in host operating system.

Root Shell in VM
Reboot from here
Shutdown
8. References

A very good video tutorial available in youtube for practicing Hadoop build using CentOS Linux 6.2 and Cloudera Hadoop installer CDH4 in rackspace public cloud.

https://www.youtube.com/watch?v=_mGIAOIGD6I

But this will incur cost in $ as shown below. But test server can be deleted after practice and credit card will be charged only for hours of usage.

Hands on, as shown in video, can be done installing same software in local machine using VM. But this will require Cloudera signup for package download and Linux install expertise with power user level.
Another good place to start is Hadoop PDFs/ HTMLs available in VM path below. You can refer these files by copying in local machine using FileZilla.